Administrator's Guide for ArcSight Platform 20.11

December 2020

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Copyright Notice

© Copyright 2021 Micro Focus or one of its affiliates

Confidential computer software. Valid license from Micro Focus required for possession, use or copying. The information contained herein is subject to change without notice.

The only warranties for Micro Focus products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein.

No portion of this product's documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's internal use, without the express written permission of Micro Focus.

Notwithstanding anything to the contrary in your license agreement for Micro Focus ArcSight software, you may reverse engineer and modify certain open source components of the software in accordance with the license terms for those particular components. See below for the applicable terms.

U.S. Governmental Rights. For purposes of your license to Micro Focus ArcSight software, “commercial computer software” is defined at FAR 2.101. If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation and other technical data subject to the terms of the Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulation (“FAR”) and its successors. If acquired by or on behalf of any agency within the Department of Defense (“DOD”), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of the Agreement as specified in 48 C.F.R. 227.7202-3 of the DOD FAR Supplement (“DFARS”) and its successors. This U.S. Government Rights Section 18.11 is in lieu of, and supersedes, any other FAR, DFARS, or other clause or provision that addresses government rights in computer software or technical data.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support-and-services/documentation

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online help.

About this Guide

Thursday, May 13, 2021

This Administrator’s Guide contains installation, user, and deployment guidance for the ArcSight software products and components that you deploy in the containerized platform. You can access the additional documents from the Micro Focus Product Documentation website.

Intended Audience

This guide provides information for IT administrators who are responsible for managing the ArcSight software and its environment.

Additional Documentation

ArcSight Platform documentation library includes the following resources:

- Administrator's Guide to ArcSight Platform 20.11, which provides concepts, use cases, and contextual help for the Dashboard and user management of the Fusion layer in ArcSight Platform.
- Technical Requirements for ArcSight Platform 20.11, which provides information about the hardware and software requirements for installing ArcSight Platform and the deployed capabilities.
- ArcSight Platform 20.11 Release Notes, which provides information about the latest release.

For the most recent version of this guide and other ArcSight documentation resources, visit the documentation site for ArcSight.

Contact Information

We want to hear your comments and suggestions about this book and the other documentation included with this product. You can use the comment on this topic link at the bottom of each page of the online documentation, or send an email to Documentation-Feedback@microfocus.com.

For specific product issues, contact Micro Focus Customer Care.

Introducing ArcSight Platform

ArcSight Platform (the Platform) enables you to deploy a combination of security, user, and entity solutions into a single cluster within the Container Deployment Foundation (CDF) environment. With CDF, you can add and remove product capabilities, as well as manage the workload across the installed nodes.

The Platform enables you to visualize, identify, and analyze potential threats by incorporating intelligence from the multiple layers of security sources that might be installed in your security environment.

These product capabilities might include the following:

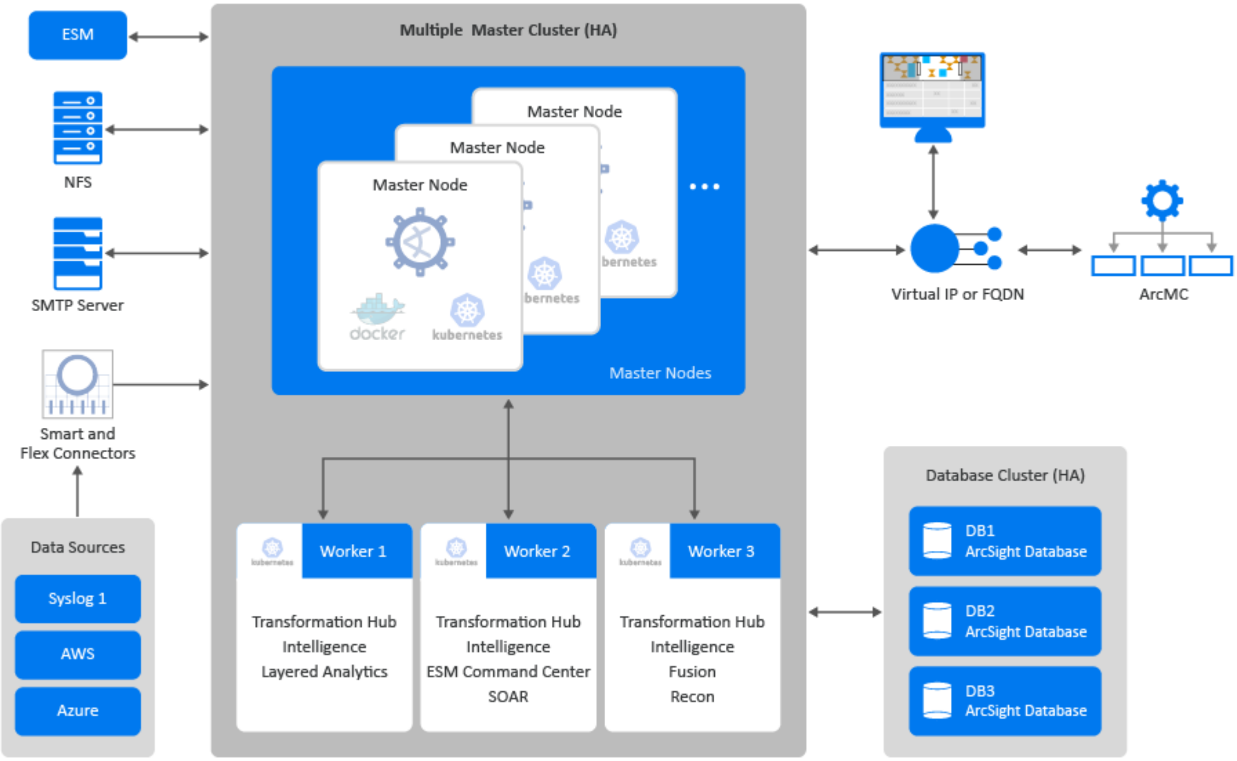
- Real-time event monitoring and correlation with data from ArcSight Enterprise Security Manager (ESM)
- Analyzing end-user behavior with ArcSight Intelligence
- Performing deep-dive investigations with ArcSight Recon
- Responding to and mitigating cyber attacks with ArcSight SOAR
- Coordinating and managing data streams with Transformation Hub

The Platform's SSO function ensures that users can navigate among the features in the Platform or launch applications from the Platform without having to log in for each product solution.

Understanding the Platform Architecture

The Platform includes three primary elements:

- The underlying CDF infrastructure
- The capabilities you deploy into the infrastructure
- The functions and applications that support the deployed capabilities



The following sections describe these three elements of the Platform architecture.

> Although you can also deploy NetIQ Identity Intelligence in this CDF-based environment, this *Administrator's Guide* does not provide instructions for deploying or managing that capability. For more information, see the Administrator's Guide to NetIQ Identity Intelligence.

Understanding the CDF Infrastructure

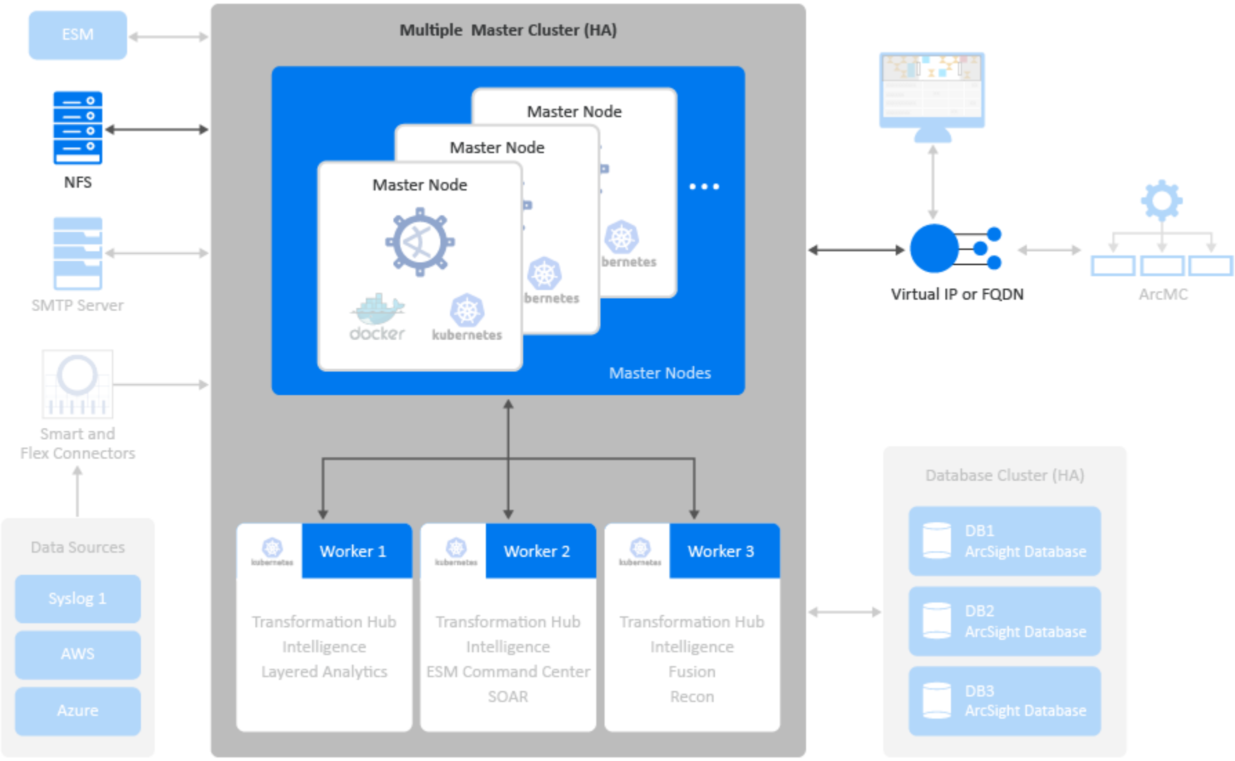
The Platform runs in the Container Deployment Foundation (CDF) infrastructure, which incorporates container management functions from Kubernetes and Docker. This containerized environment enables you to swiftly install and manage an integrated solution of ArcSight products in a single interface. The CDF has both a "CDF Installer" on the next page and a browser-based "CDF Management Portal" on the next page.

We provide two ways of using the installer function:

- An assisted process using the ArcSight Installation Tool
- A manual process

You will also need to install additional software and components to support your security solution. Your ArcSight environment might include the containerized capabilities, which are distributed across multiple host systems, plus servers for databases and the supporting products.

The number of hosts you need depends on several factors, such as the need for high availability and the size of workloads based on events per second.



The CDF architecture requires several components:

- "CDF Installer" below
- "CDF Management Portal" below
- "Kubernetes and Docker" below
- "Master Nodes" on the next page
- "Network File System" on the next page
- "Worker Nodes" on the next page
- "Virtual IP Address" on the next page

CDF Installer

You use the CDF installer for installing, configuring, and upgrading the CDF infrastructure. When using the ArcSight Installation Tool, the CDF installer is executed automatically in an embedded manner so that you need not use the CDF installer directly.

CDF Management Portal

The Management Portal enables you to manage and reconfigure your deployed environment after the installation process is complete. You can add or remove deployed capabilities and worker nodes, as well as manage license keys.

During installation, you specify the credentials for the administrator of the Management Portal. This administrator is not the same as the admin user that you are prompted to create the first time that you log in to the Platform after installation.

When you upgrade the Platform, you use the Management Portal to upgrade the deployed capabilities.

Kubernetes and Docker

Kubernetes automates deployment, scaling, maintenance, and management of the containerized capabilities across the cluster of host systems. Applications running in Kubernetes are defined as pods, which group containerized components. Kubernetes clusters use Docker containers as the pod components.

A **pod** consists of one or more containers that are guaranteed to be co-located on the host server and can share resources. Each pod in Kubernetes is assigned a unique IP address within the cluster, allowing applications to use ports without the risk of conflict.

Persistent services for a pod can be defined as a volume, such as a local disk directory or a network disk, and exposed by Kubernetes to the containers in the pod to use. A cluster relies upon an external Network File System (NFS) as its shared persistent storage. The clusters require master and worker nodes. For more information about the Platform pods, see Understanding Labels and Pods.

Master Nodes

The master nodes control the Kubernetes cluster, manage the workload on the worker nodes, and direct communication across the system. You should deploy three master nodes to ensure high availability. However, you can use the Platform with a single master node.

Network File System

The Network File System (NFS) stores some of the persistent data generated by Transformation Hub, Intelligence, and Fusion.

Worker Nodes

Worker nodes run the application components and perform the work in the Kubernetes cluster. For all highly available configurations, we recommend deploying a minimum of three dedicated worker nodes.

You can add and remove worker nodes from the cluster as needed. Scaling the cluster to perform more work requires additional worker nodes, all of which are managed by the master nodes. The workload assigned to each node depends on the labels assigned to them during deployment or reconfiguration after deployment.

Virtual IP Address

CDF supports high availability (HA) through load balancers and the Keepalived service. You can configure either external load balancers or Keepalived for high availability. If you have configured a virtual IP for a multi-master installation, the HA virtual IP address you defined bonds to one of the three master nodes.

If a master node fails, the virtual IP address is assigned to an active master node. This setup helps to provide high availability for the cluster.

When you configure a connection to the cluster, configure the connection to use the virtual IP so that it benefits from the HA capability. One exception to this recommendation is when you are configuring a connection to Transformation Hub's Kafka, in which case you can achieve better performance by configuring the Kafka connection to connect directly to the list of worker nodes where Kafka is deployed.

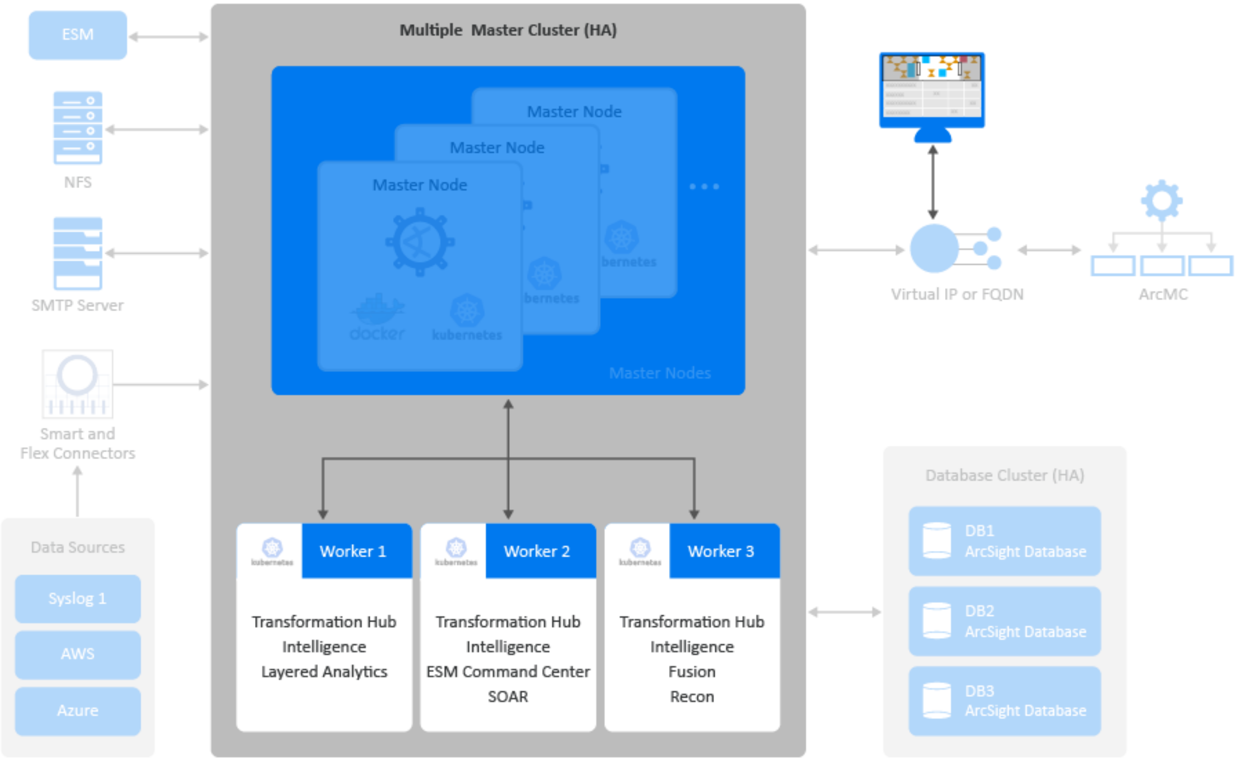
Deciding on the Capabilities to Deploy

The Platform infrastructure enables you to deploy a combination of container-based **capabilities**, which represent licensed products and functions that shape your ArcSight environment. Each release of the Platform supports a specific set of capabilities that you can deploy.

To perform appropriately, some capabilities that you deploy depend on the presence of additional capabilities. For example, most capabilities need the Fusion capability because it provides the user management functions in the Platform.

> The capabilities that can be deployed in the Platform are designed to automatically integrate with each other when deployed to the same cluster. You must deploy capabilities to the same cluster for them to operate in an integrated manner.

For a complete security, user, and entity solution, you might also need to integrate software and components that are not deployed within the Platform. For example, your solution might need a database for data storage and Micro Focus ArcSight SmartConnectors for data collection from various data sources.



You can deploy the following capabilities in the Platform:

- "ArcSight Management Center" below
- "ESM Command Center" below
- "Fusion" on the next page
- "Intelligence" on the next page
- "Layered Analytics" on page 14
- "Recon" on page 14
- "SOAR" on page 14
- "Transformation Hub " on page 15

For more shared capabilities, see "Understanding Labels and Pods" on page 584

ArcSight Management Center

ArcSight Management Center (ArcMC) is a centralized management tool that simplifies security policy configuration, deployment maintenance, and monitoring in an efficient and cost-effective manner.

ArcMC offers these key capabilities:

• **Management and Monitoring:** deliver the single management interface to administrate and monitor ArcSight managed nodes, such as Connector Appliances, Loggers, Connectors, Collectors, other ArcMCs, and Transformation Hub.

• **SmartConnector Hosting:** for the hardware appliance, as a platform to host and execute SmartConnectors.

ESM Command Center

ArcSight Command Center for Enterprise Security Manager (ESM Command Center) is a licensed product that provides widgets and dashboards that you can customize in the Dashboard feature for detecting threats to your enterprise. If you deploy "Intelligence" on the next page and "Layered Analytics" on page 14 in the same cluster as ESM Command Center, certain widgets will combine data from ESM and Intelligence to provide you greater insight into events and entity behavior.

With Transformation Hub deployed in the same cluster, ESM can receive event data for dashboarding and further correlation.

This capability requires the Fusion capability.

Fusion

Fusion provides the common elements needed for the products that you deploy in the Platform environment to ensure a unified solution experience: user management, the Dashboard, and other core services. The Dashboard enables you to visualize, identify, and analyze potential threats by incorporating intelligence from the multiple layers of security sources that might be installed in your security environment.

Fusion enables you to add users and groups, as well as manage their roles and permissions. Some capabilities provide the ability for users to set their preferences for some features, which is incorporated into the My Profile section of user management. For example, with Recon deployed, users can specify the default settings for Search.

Fusion services also support SSO configuration across the capabilities, high-capacity data management, and a search engine. Most capabilities require that Fusion be deployed in the same cluster.

Intelligence

ArcSight Intelligence is a licensed product that provides a market-leading analytics platform, using unsupervised online machine learning to identify unknown threats like insider threats or targeted outside attacks such as APTs.

These types of threats simply cannot be identified by searching for a known “bad signature.” Unsupervised machine learning gives threat hunters a high-quality set of leads to help them identify these elusive threats.

The analytics platform in ArcSight Intelligence uses:

- ArcSight SmartConnectors
- Supporting Active Directory/Authentication data
- Web proxy data
- Additional data sources

In addition, you can use FlexConnectors to pull ArcSight Intelligence analytical results and push them into ESM for higher accuracy correlation rules that leverage unsupervised learning anomalies, as well as correlation rule filtering using top risky entity lists.

If you deploy "ESM Command Center" on the previous page and "Layered Analytics" on the next page in the same cluster as the ArcSight Intelligence capability, certain widgets will combine data from ESM and ArcSight Intelligence to provide you greater insight into events and entity behavior.

This capability requires the Fusion and Transformation Hub capabilities, and the ArcSight Database.

Layered Analytics

Layered Analytics blends the analytics results from the "ESM Command Center" on page 12 and "Intelligence" on the previous page capabilities, thus providing multiple layers of useful data that can lead to actionable insights.

This capability requires the ESM Command Center and Intelligence capabilities.

Recon

ArcSight Recon is a licensed product that enables you to search, analyze, and visualize machine-generated data gathered from web sites, applications, sensors, and devices that make up your monitored network. Recon indexes the events from your data source so that you can view and search them.

The intuitive search language makes it easy to formulate queries. You can use the large set of dashboards and reports available in the Reports Portal to monitor and identify vulnerabilities and threats in your enterprise.

Recon integrates with "Transformation Hub " on the next page for processing raw events. Recon also can integrate with ESM to receive alerts and start the investigation process.

This capability requires the Fusion and Transformation Hub capabilities, and the ArcSight Database.

SOAR

ArcSight SOAR is a licensed Security Orchestration, Automation and Response Platform product that combines orchestration of both technology and people, automation, and incident management into a seamless experience.

SOAR enables you to connect the dots between people, process, and technology in SecOps with various and diverse forms of automation, analyst augmentation, and collaborative investigation and response. With 100+ integrations from different vendors, ArcSight SOAR provides a single pane of glass for security operations and speeds up the incident response process.

This capability requires Fusion, as well as the ESM Command Center or Recon capabilities.

Transformation Hub

Transformation Hub is a licensed product that lets you take advantage of scalable, high-throughput, multi-broker clusters for publishing and subscribing to event data. It coordinates and manages data streams, which enables your environment to scale, and opens events to third-party data solutions. Moreover, to reduce the computational overhead and workload on a syslog SmartConnector infrastructure, you can make use of Connectors in Transformation Hub (CTH) instead.

Transformation Hub ingests, enriches, normalizes, and then routes event data from data producers to connections between existing data lakes, analytics platforms, and other security technologies and the multiple systems within the Security Operations Center (SOC).

Transformation Hub can seamlessly broker data from any source and to any destination. Its architecture is based on Apache Kafka and it supports native Hadoop Distributed File System (HDFS) capabilities, enabling both the ArcSight Logger and ArcSight Recon technologies to push to HDFS for long-term, low-cost storage.

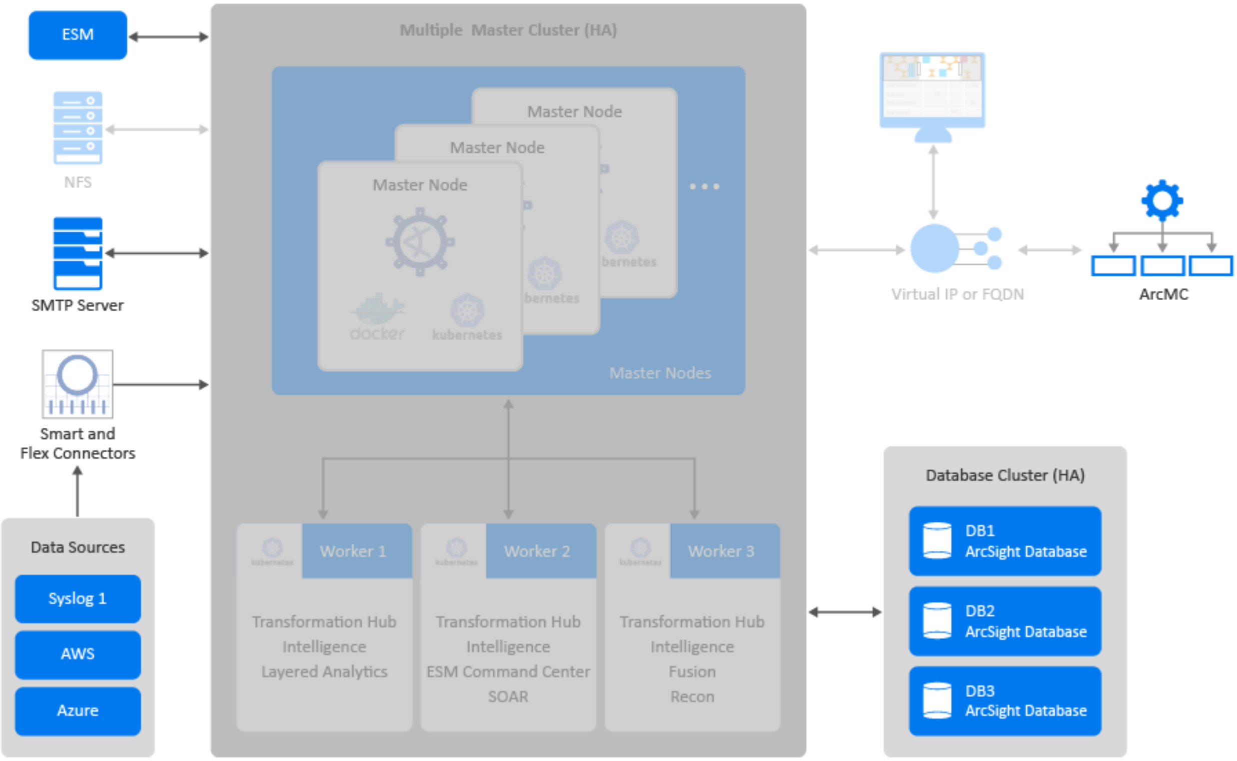
This architecture reduces the overall ArcSight infrastructure footprint, scales event ingestion using built-in capabilities, and greatly simplifies upgrades to newer Transformation Hub releases.

It also positions the platform to support an analytics streaming plug-in framework, supporting automated machine learning and artificial intelligence engines for data source onboarding, event enrichment, and detection and attribution of entities and actors.

Several capabilities require that Transformation Hub be deployed in the same cluster.

Understanding Related Components

The capabilities you deploy in the Platform depend on functions and applications installed in your environment. For example, Transformation Hub consumes data from a wide variety of collectors and connectors before passing that content to ESM and other products. Recon and Intelligence need the ArcSight Database to store their data.



- "ArcSight Management Center" below
- "Database" below
- "Data Sources" on the next page
- "Enterprise Security Manager" on the next page
- "SMTP Server" on the next page

ArcSight Management Center

ArcSight Management Center (ArcMC) serves as a centralized management interface to help you effectively administer and monitor Transformation Hub and the SmartConnectors. ArcMC communicates with the Platform by connecting to the virtual IP address or fully qualified domain name (FQDN) assigned to the primary master node in the cluster.

The Platform does not require ArcMC. However, ArcMC enhances the runtime management of Transformation Hub topics.

Database

The Arcsight Database stores all collected events and provides event searches and analysis capabilities. The database runs in its own cluster, usually on three or more nodes for high availability.

Data Sources

The deployed capabilities incorporate data from a variety of sources.

- **SmartConnectors** collect events from supported data sources. They then send these events to the Transformation Hub's Kafka cluster.
  - Event data includes entity information such as users, assets, and endpoints based on the event type.
  - When collecting data and sending it to Transformation Hub, the SmartConnector normalizes the values (such as severity, priority, and time zone) into the common format and normalizes the data structure into the common schema.
  - Next, the connectors filter and aggregate events to reduce the volume of events sent to the system.
  - You need to install and maintain connectors separately.
  - You can subscribe to the data Transformation Hub manages.
- Third-party collectors and connectors also provide data to the deployed capabilities.

Enterprise Security Manager

ArcSight Enterprise Security Manager (ESM) operates outside of the Platform CDF environment, but integrates with capabilities that operate within the Platform environment. For example, ESM shares SSO, event processing, and event search behavior with the Platform.

You can deploy the ESM Command Center capability to the Platform CDF environment to provide a more seamless user experience with other capabilities that integrate with the Platform Fusion capability, such as Intelligence and SOAR. When deployed in this manner, ESM Command Center integrates with ESM operating outside of the Platform CDF environment.

SMTP Server

The SMTP server allows the Platform to send notification messages to users.

Planning to Install and Deploy

This section describes the installation and deployment options, considerations, and caveats that you need to know for a successful deployment.

Checklist: Planning to Deploy the Platform

Use the following checklist to install and configure the Platform infrastructure. Perform the tasks in the listed order.

	Task	See
❑	1. Learn about the software and components that you need to install, deploy, and configure.	Deciding on the Capabilities to Deploy Understanding the CDF Infrastructure Understanding Related Components
❑	2. Decide how you want to configure your Platform environment.	ArcSight Platform Technical Requirements
❑	3. Ensure that the computers on which you are installing the Platform components meet the specified requirements.	ArcSight Platform Technical Requirements
❑	4. Review the knowledge and individuals needed to perform the installation processes.	Identifying the Installation Team
❑	5. Review the considerations for creating the Platform infrastructure.	Reviewing the Considerations and Best Practices
❑	6. Understand the security modes and their prerequisites needed for establishing communication between the infrastructure components.	Understanding Secure Communication Among Components
❑	7. Decide whether to use the ArcSight Installation Tool (on-premises only) or the manual process.	Choosing Your Installation Method

Identifying Your Installation Team

Your installation will require specific administration skills, and coordination with corporate IT departments, including the following:

- Linux operating system administration (including applying OS updates; configuring networks, firewalls, ports, and user access; and performing additional tasks)
- Familiarity with editing configuration files
- Running commands and scripts on one or more operating systems
- Familiarity with Micro Focus components
- Familiarity with Kafka processing and configuration

Your installation team will need the following roles and responsibilities to properly configure the infrastructure environment.

Role	Responsibility
Application admin	The person in this role must ensure successful execution of the entire installation including verification and post-installation tasks. This person must have a good understanding of the entire installation process, request support from other appropriate roles as needed, and complete the installation once the environment is ready for installation.
IT admin	The person in this role prepares physical or virtual machines as requested by the application administrator.
Network admin	The person in this role manages network-related configuration for your organization. This person needs to perform network configuration tasks as requested by the application administrator.
Storage admin	The person in this role plans and deploys all types of storage for your organization. This person needs to set up one or more NFS servers required by the CDF installation.

Reviewing the Considerations and Best Practices

Before starting the installation process, there are several decisions to be made to plan and prepare your infrastructure. Below are the considerations you need to consider, as well as an outline of steps you to follow during this planning and preparation process. We will explain details in later sections of this guide.

Consideration	Best Practices
Host Systems	• Provision cluster (master and worker node) host systems and operating environments, including OS, storage, network, and Virtual IP (VIP) if needed for high availability (HA). Note the IP addresses and FQDNs of these systems for use during product deployment. • You can install the cluster using a `sudo` USER with sufficient privileges, or, alternatively, you can install it using the `root` USERID. • Systems must not only meet minimum requirements for CPU cores, memory, and disk storage capacity, but also meet anticipated requirements for end-to-end events processing throughput. • Master and worker nodes can be deployed on virtual machines. However, since most of the processing occurs on worker nodes, if possible, you should deploy worker nodes on physical servers. • When using virtual environments, please ensure: ◦ Resources are reserved and not shared. ◦ The UUID and MAC addresses are static and do not change after a reboot or a VM move. Dynamic IP addresses will cause the Kubernetes cluster to fail. • All master and worker nodes must be installed in the same subnet. • If a master and worker are sharing a node, follow the higher-capacity worker node sizing guidelines. We do not recommend this configuration for production Transformation Hub environments.

Consideration	Best Practices
High Availability	• For high availability (HA) of master nodes on a multi-master installation, you must create a Virtual IP (VIP) which will be shared by all master nodes. Prior to installation, a VIP must not respond when pinged. • All master nodes should use the same hardware configuration, and all worker nodes should use the same hardware configuration (which is likely to be different from that of the master nodes). • For HA, exactly three master nodes, at least three worker nodes, and at least three database nodes should be used so that if one of each node type fails, the remaining nodes can continue to operate the system without downtime. This is the configuration illustrated in the diagram. You can use fewer nodes of each node type. However, this configuration will result in that node type not being HA. • For HA, use an NFS server that has HA capabilities so that it is not a single point of failure. • For master nodes, only 1 or 3 master nodes are allowed. • If you deploy a single master node, failure of the single master node could cause you to lose the ability to manage the entire cluster you recover until the single master node. In some extreme scenarios, failure of the single master node could cause the entire cluster to become unrecoverable, requiring a complete reinstall and reconfiguration. • It is not possible to add master nodes after the cluster has been initially deployed. You must decide before deploying the cluster whether to initially deploy multiple master nodes. Adding additional master nodes after deployment will require reinstalling the cluster, leading to downtime. • When the installer is configured to create more than one database node, the database fault tolerance will be set to one. This means the data in the database will be replicated so that one database node can fail and the system will continue to operate properly. Database storage utilization will double as a result of the data replication. In a failure scenario, the failed node should urgently be restored before there is a chance of another node failure, which will shut down the database to avoid additional problems. • If you configure the installer to create only a single database node, the database fault tolerance is set to zero because there is only a single node. Therefore, no other node will continue during a failure, and no data replication will occur in this scenario.
Storage	• Available from the Micro Focus support community, the CDF Deployment Disk Size Calculator spreadsheet determines your recommended disk storage requirements and other configuration settings based on throughput requirements. Download the spreadsheet to help determine your storage needs. • Create or use an existing external NFS storage environment with sufficient capacity for the throughput needed. Guidelines are provided below. • Determine the size and total throughput requirements of your environment using total EPS. For example, if there are 50K EPS inbound, and 100K EPS consumed, then the size would be 150K EPS. • Data compression is performed on the producer side (for example, in a Smart Connector).

Consideration	Best Practices
Scaling	• Adding more worker nodes is typically more effective than installing bigger and faster hardware because individual workloads on worker nodes are usually relatively small and some of them work better when there are fewer different workloads on the same node. Using more worker nodes also enables you to perform maintenance on your cluster nodes with minimal impact to your production environment. Adding more nodes also helps with predicting costs due to new hardware. • Unlike worker nodes, for the database it is typically more effective to use bigger and faster hardware than to increase the number of database nodes because the database technology can fully utilize larger hardware and this decreases the need for coordination between database nodes. With that said, for HA it is important to deploy enough database nodes to be resilient in case of a database node failure or individual node downtime for maintenance.
Network	• Although event data containing IPv6 content is supported, the cluster infrastructure is not supported on IPv6-only systems.
Security	• Determine a security mode (FIPS, TLS, Client Authentication) for communication between components. • "Understanding Secure Communication Among Components" on the next page Changing the security mode after installation might require downtime for uninstalling and re-installing the Transformation Hub.
Performance	• Kafka processing settings for Leader Acknowledgement (ACK) and TLS settings have a significant effect on throughput through the system. If ACK and TLS are both enabled, throughput performance might be degraded by a factor of 10 or more, requiring additional worker nodes to account for the processing overhead. • If SmartConnector is configured to send events to Transformation Hub in CEF format and the events are being stored in ArcSight Database, consider the potential performance effects of the CEF-to-Avro data transformation, and allow a 20% increase in CPU utilization. This will generally have a large impact only with very high EPS (250K+) rates. Consider configuring the SmartConnector to use the Avro event format instead, which avoids the need for this transformation.
Downloads and Licensing	• Ensure that you have access to the Micro Focus software download location. You will download installation packages to the Initial Master Node in the cluster. • Ensure that you have a valid Micro Focus license key for the software being installed.
Installing with Enterprise Security Manager	If you want to install the Platform and the ESM server in the same environment, specify during the Platform intallation a CDF API Server Port that does not use the same port as the ESM server (default 8443). For example, when using the Platform Install tool, the `example-install-config-esm_cmd_center-single-node.yaml` sets the master-api-ssl-port to port 7443.

Understanding Secure Communication Among Components

Determine the security mode for communication between your infrastructure components. The security mode of connected producers and consumers must be the same across all components.

> The secure communication described applies only in the context of the components that relate to the Micro Focus container-based application you are using, which is specified in that application's documentation.

When possible, configure the Micro Focus components with the security mode you intend to use *before* connecting them to additional ArcSight Platform products.

To enhance security, you can configure TLS Client Authentication between components that do not utilize client username and password authentication, such as producers and consumers connecting to Transformation Hub. With TLS Client Authentication enabled, the client and the server authenticate each other to ensure that both parties involved in the communication are trusted.

> Changing the Allow Plain Text, TLS Client Authentication, or FIPS settings after the deployment will necessitate system downtime because the component will need to be redeployed.

Micro Focus product documentation is available from the Micro Focus support community.

Product	Preparations Needed	TCP Ports	Supported Security Modes
ArcMC	• Be sure to use v2.9.5 or later. • Install ArcMC before the Platform installation.	• 443 • 32080	• TLS • FIPS • TLS Client Authentication

SmartConnectors and Collectors	• You can install and run SmartConnectors and ArcMC onboard connectors before you install the Platform. Or, you can install them after you deploy the Platform. • FIPS mode setup is not supported between SmartConnector v7.5 and the Platform. • Only TLS and TLS Client Authentication are supported. • FIPS mode *is* supported between Connectors v7.6 and later and the Platform.	• 9092 (Plain Text) • 9093 (TLS)	• TLS • FIPS (SC 7.6+ only) • TLS Client Authentication • Plain text
ArcSight ESM	• You can install and run ESM before you install the Platform. • Changing ESM from FIPS to TLS mode (or from TLS to FIPS) requires a redeployment of ESM.	• 9093 (TLS)	• TLS • FIPS • TLS Client Authentication
ArcSight Logger	• You can install and run Logger before you install the Platform.	• 9092 (Plain Text) • 9093 (TLS)	• TLS • FIPS • TLS Client Authentication • Plain text
ArcSight Database	• You install the ArcSight Database before the Platform.	• 9092 (Plain Text) • 9093 (TLS)	• Plain text • TLS • TLS Client Authentication • FIPS
NFS Server	• For optimal security, secure all NFS settings to allow only required hosts to connect to the NFS server.	• 2049	• Plain text
Web Browser	• By default, TLS is enabled.	• 443 • 5443 • 3000	• TLS

Leader Acknowledgements ("acks") and TLS Enablement: In general, enabling leader ACKs and TLS will result in significantly slower throughput rates, but greater fidelity in ensuring that subscribers receive events. For more information about Leader Acknowledgements, TLS enablement, and their effects on processing throughput, see the Kafka documentation.

Understanding FIPS

What is FIPS?

Federal Information Processing Standards (FIPS) is a set of rules and regulations defined by the United States government that specify the security requirements for data processing and communication between the components.

FIPS Standards

FIPS Publication	Standard
FIPS 140	Security Requirements for Cryptographic Modules Standard
FIPS 180	Secure Hash Standard
FIPS 186	Digital Signature Standard
FIPS 197	Advanced Encryption Standard (AES)
FIPS 198	Keyed-Hash Message Authentication Code (HMAC)
FIPS 199	Standards for Security Categorization of Federal Information and Information Systems
FIPS 200	Minimum Security Requirements for Federal Information and Information Systems
FIPS 201	Personal Identity Verification (PIV) of Federal Employees and Contractors
FIPS 202	Permutation-Based Hash and Extendable-Output Functions (SHA-3 Standard)

FIPS 140 - Security Requirements for Cryptographic Modules Standard

FIPS 140 is one of the standards of FIPS that governs the use of encryption and cryptographic services. FIPS 140 defines security rules and regulations for cryptographic modules to keep sensitive information secure.

According to the **Federal Information Security Management Act (FISMA)**, all the United States government agencies, United States government contractors, and third parties working for the federal agencies must adhere to the FIPS 140 standard.

For testing cryptographic modules, the two revised editions of FIPS 140 are given below:

FIPS 140 Edition	Purpose
FIPS 140-2	Includes changes in technology and standards defined by other standards bodies. Includes modifications based on comments from vendors, laboratories, and user communities.
FIPS 140-3	Aligns with standards defined by the **International Organization for Standardization (ISO)** and the **International Electrotechnical Commission (IEC)**

ArcSight Platform FIPS Mode

ArcSight Platform supports FIPS 140 and the FIPS 140 mode is active by default. ArcSight Platform establishes a secure communication between its components using FIPS validated cryptographic modules.

Currently, only some of the components in the ArcSight Platform architecture (and the components that integrate with it) can operate in the FIPS 140 mode.

These include all the components that directly handle event data (from edge ingestion to storage in the database) and the management and analytics capabilities listed below:

Component	Sub-components	Purpose	Enabling FIPS
Smart Connectors	All sub-components	Perform the edge ingestion of event data from the data sources and deliver the events to Transformation Hub.	- See the Smart Connectors section.

Component	Sub-components	Purpose	Enabling FIPS

Transformation Hub	`th-kafka`
	`th-schemaregistry`
	`th-routing-processor`
	`th-c2av-processor`
	`th-web-service`
	`th-cth`
		Receive events from SmartConnectors and make the events available to Transformation Hub consumers, such as the Database.	• For the sub-components listed that support FIPS mode, FIPS mode can be enabled during deployment.
			• When using the ArcSight Platform Installer tool, add the property `th-init-fips: true` to the `suite > config-params` section of your installation configuration `yaml` file.
			For example:
			`suite:`
			`  products: [fusion, esm, soar]`
			`  config-params:`
			`     th-init-fips: true`
			`    search-engine-replica: 0`
			• When performing the installation manually, configure the `Transformation Hub > Connections use FIPS encryption` option as described in the Configuring the Deployed Capabilities

Component	Sub-components	Purpose	Enabling FIPS
			section.
Database	All sub-components	Consume, store, and make the events from Transformation Hub available to other components to query or search the events efficiently.	- See the Setting FIPS Mode on the Database Server section.

Fusion	`fusion-single-sign-on`
	`fusion-user-management`
	`fusion-db-search-engine`
		Perform user management and authentication processes.
		Provide a database search engine API, which connects directly to the Database, but does not include the user interface components that call the search engine API.
			- For the sub-components listed that support FIPS mode, FIPS mode is always enabled with no option to disable.

Recon	`reporting-web-app`	Provides all the reporting capabilities and connects directly to the Database and Fusion authentication and user management pods.	- For the sub-components listed that support FIPS mode, FIPS mode is always enabled with no option to disable.

Component	Sub-components	Purpose	Enabling FIPS
Enterprise Security Manager (ESM)	All sub-components, except the ESM Command Center pods running in the containerized environment.		• See the Configuration Changes Related to FIPS section in the ArcSight ESM Guide.
ArcSight Management Center (ArcMC)	All sub-components		• See the ArcMC Administrator's Guide.

Intelligence	`h2`
	`interset-analytics`
	`interset-api`
	`interset-logstash`
	`searchmanager-api`
	`searchmanager-engine`
	`intelligence-arcsight-connector-api`
	`intelligence-tuning-api`
			• For the sub-components listed that support FIPS mode, FIPS mode is always enabled with no option to disable.

> The components that can not operate in the FIPS 140 mode use strong industry standard encryption to establish secure communication. However, our objective is to increase the coverage of components that can operate in the FIPS 140 mode.

For more information about each of the pods listed above, see Understanding Labels and Pods.

Understanding Kubernetes Network Subnets

Kubernetes automates the deployment of its management services and the pods associated with deployed capabilities to master and worker nodes. As part of this process, it allocates a unique IP address to each service and pod.

In order to do so, Kubernetes must be provided with a reserved range of private network IP addresses for its services (service-cidr parameter, default is 172.17.17.0/24) and a separate reserved range of private network IP addresses for pods (pod-cidr parameter, default is 172.16.0.0/16).

The two IP ranges must not overlap, must not be allocated to other systems in the network, and are provided to Kubernetes at install time by specifying a network subnet in Classless Inter-Domain Routing (CIDR) format. CIDR notation includes an IP address, a slash ('/') character, and a network prefix (a decimal number).

The minimum useful network prefix is /24 and the maximum useful network prefix is /8. The default value is 172.16.0.0/16. For example:

```
POD_CIDR=172.16.0.0/16
```

The pod-cidr IP range must contain an adequate number of IP addresses to accommodate the functions of all of the pods deployed to the cluster. Each node in the cluster is allocated a segment of the pod-cidr IP range for use by the pods that are deployed to that node as determined by the pod-cidr-subnetlen parameter.

The default value for pod-cidr-subnetlen is automatically computed depending on the value of pod-cidr, as described below. The default value of pod-cidr-subnetlen is expected to be adequate. However, if for some unexpected reason you find that pods on nodes run out of available IP addresses, you can set the pod-cidr-subnetlen parameter to a value that makes more IP addresses available to each node.

POD_CIDR Prefix	POD_CIDR_SUBNETLEN defaults	POD_CIDR_SUBNETLEN allowed values
/8 to /21	/24	/(POD_CIDR prefix + 3) to /27
/22 to /24	/(POD_CIDR prefix + 3)	/(POD_CIDR prefix + 3) to /27

Smaller prefix values indicate a larger number of available addresses. The minimum useful network prefix is /27 and the maximum useful network prefix is /12. The default value is 172.17.17.0/24.

Choosing Your Installation Method

You can install ArcSight products using one of the two methods below.

- Using ArcSight Platform Installer (On-premises Only)
- Manual Installation (On-premises or Cloud)

ArcSight Platform Installer (On-premises Only)

ArcSight Platform Installer significantly simplifies the installation and deployment experience using automation.

The installer has a prerequisites checker that verifies the OS, storage, network, and other settings are appropriate for the desired deployment. You can have the Installer tool adjust the prerequisite settings to ensure a successful deployment.

It is also capable of deploying containerized and database infrastructures in a simple, all-in-one node or in a highly-available multi-node configuration. It requires a minimum set of deployment configuration settings to describe the capabilities to deploy; master, worker, and database node host names; and login IDs and passwords. You can run the tool with this deployment configuration on a single node and it will automatically connect to all the nodes specified in the configuration in order to run the capabilities.

The download package includes a set of example deployment configurations. For example, a highly-available deployment configuration of Recon.

Consider the following when deciding if it is a good fit to use in your environment:

- ArcSight Platform Installer has not been tested on a dual-homed network (dual or redundant connections to a single Internet Service Provider), so be careful before using it in this scenario.
- ArcSight Platform Installer is only capable of installing to an on-premise environment.
- ArcSight Platform Installer disables the option to authorize the collection suite usage data.
- Passwordless ssh access will automatically be configured between the master node, where ArcSight Platform Installer is used and all other CDF and Database nodes, so ArcSight Platform Installer can automatically perform tasks on the nodes securely without requiring passwords to be retained.
- The ArcSight Platform Installer assumes that yum is already installed and configured on every node and the pre-check fails if it is not.
- If ArcSight Platform Installer is not a good fit for your environment, you can perform the installation manually.

> If you have any customizations on the operating system, you might want to prepare your machines with the prerequisites for CDF and Database and perform deployment and post deployment configuration using the ArcSight Platform Installer because, when ArcSight Platform Installer automatically configures the prerequisites, it might overwrite your customizations. Information for preparing your machines manually is availability:
> - Preparing Your Environment for Database
> - Preparing Your Environment for CDF

Manual Installation (On-premises or Cloud)

If you need to deploy to a cloud provider, such as Azure or AWS, or if ArcSight Platform Installer does not meet your needs, you can use a manual installation.

- On-premises (Ensure you prepare your environment before using this manual installation method.)
- Cloud deployment for Transformation Hub only

Creating an On-premises Deployment

This section discusses the process of preparing for and creating an on-premises deployment.

> The installation process validates the infrastructure environment before performing the installation, as well as after the installation has completed.

Checklist: Creating an On-premises Deployment

Use the following checklist to create an on-premises deployment of the Platform infrastructure. This process includes installing the CDF and deploying your chosen capabilities. Perform the tasks in the listed order.

	Task	See
❑	1. Complete the planning checklist.	Checklist: Planning to Deploy the Platform
❑	2. Prepare your on-premises environment for the CDF.	Preparing Your Environment
❑	3. (Conditional) For a guided deployment, use ArcSight Platform Installer.	Using ArcSight Platform Installer to Deploy
❑	4. (Conditional) For a manual deployment, install the CDF and the related components, and deploy the capabilities.	Performing a Manual Deployment
❑	5. Complete the deployment process.	Performing Post-deployment Configurations
❑	6. Integrate your platform into your environment.	Integrating the Platform Into Your Environment

Preparing Your Environment

The actual installation of container-based applications on properly configured infrastructure, as described later in the product Deployment Guides, is quick and straightforward. The most complex part of the installation process is the preparation of the hosts, storage, and networking infrastructure, which is described in this topic.

The installation process includes several milestones, and each milestone includes several interdependent steps. The installation process validates the infrastructure environment before performing application installation, as well as after the installation has completed.

> Before building your environment, ensure that the firewall is running on the CDF nodes.

Deploying ArcSight Platform and ESM on the Same Server

Micro Focus recommends that you install ArcSight Platform and ESM on different servers because this enables the use of ArcSight Platform high availability and provides the option to deploy additional capabilities in the future. However, if you do not need the additional capabilities or you plan to reinstall ArcSight Platform should you need those capabilities, you can install the ESM Command Center (and the required Fusion) capability of the Platform on the same server where you install Enterprise Security Manager.

> When you install the Platform, specify a CDF API Server Port that does not use the same port as the ESM server (default 8443). For more information about ArcSight Platform ports, see the Technical Requirements for ArcSight Platform. For example, when you use the ArcSight Platform installer, the `example-install-config-esm_cmd_center-single-node.yaml` sets the master-api-ssl-port to port 7443.

To deploy on the same server:

1. Install ESM.

   > ⚠ Always install ESM before you install the Platform.

2. Add the ESM https port in iptables using the following commands.

   a. To find your active zones, use the following command:

   ```
   firewall-cmd --get-active-zones
   ```

   b. To add the ESM port in iptables, use the following command. By default the port number is 8443.

```
firewall-cmd --zone=public --add-port=<port_number>/tcp --permanent
```

> This step enables you to access ESM externally (outside the firewall).

c. To reload the firewall so that the changes are applied, use the following command:

```
firewall-cmd --reload
```

3. Continue with the Platform preparation and deployment.

Configuring Proxy Settings

The cluster should have no access to the Internet and proxy settings (`http_proxy, https_proxy` and no_proxy) should not be set. However, if you need an Internet connection and you already specified a proxy server for http and https connection, then you must correctly configure `no_proxy`.

No Proxy Definitions

If you have the `http_proxy` or `https_proxy` set, then the `no_proxy` definitions must contain at least the following values:

```
no_proxy=localhost, 127.0.0.1, <all Master and Worker cluster node IP
addresses>,<all Master and Worker cluster node FQDNs>,<HA virtual IP
Address>,<FQDN for the HA Virtual IP address>
```

> Incorrect configuration of proxy settings is a common installation issue. To verify that proxy settings are configured properly, on all master and worker nodes, run the following command and ensure the output corresponds to the recommendations.
>
> ```
> echo $http_proxy, $https_proxy, $no_proxy
> ```

If the firewall is turned off, the install process generates a warning. To avoid this warning, set the CDF Install parameter `--auto-configure-firewall` to true.

Proxy Settings Example

> Although the text here is displayed with line breaks due to page limitations, there should be no line breaks in your actual proxy settings.

```
export http_proxy="http://web-proxy.http_example.net:8080"
```

```
export https_proxy="https://web-proxy.http_example.net:8080"
```

```
export no_
proxy="localhost,127.0.0.1,node1.swinfra.net,10.94.235.231,node2.swinfra.net,
10.94.235.232,node3.swinfra.net,10.94.235.233,node3.swinfra.net,10.94.235.233
,node4.swinfra.net,10.94.235.234,node5.swinfra.net,10.94.235.235,node6.swinfr
a.net,10.94.235.236,ha.swinfra.net 10.94.235.200"
```

> Optionally, in the above line, you can use `swinfra` to escape proxy for all hosts inside that domain.

Configuring DNS Settings

Ensure host name resolution through Domain Name Services (DNS) is working across all nodes in the cluster, including correct forward and reverse DNS lookups.

> Host name resolution **must not be** performed through `/etc/hosts` file settings.

- "Understanding the Use of a Fully Qualified Domain Name (FQDN)" below
- "Configuring Secure DNS" on the next page
- "Testing Forward and Reverse DNS Lookup" on the next page
- "Running the Commands" on page 39

Understanding the Use of a Fully Qualified Domain Name (FQDN)

All master and worker nodes must be configured with a Fully Qualified Domain Name (FQDN), and must be in the same subnet. Transformation Hub uses the host system FQDN as its Kafka `advertised.host.name`.

If the FQDN resolves successfully in the Network Address Translation (NAT) environment, Producers and Consumers will function correctly. If there are network-specific issues resolving FQDN through NAT, DNS will need to be updated to resolve these issues.

Configuration Notes

- Transformation Hub supports ingestion of event data that contains both IPv4 and IPv6 addresses. However, its infrastructure cannot be installed into an IPv6-only network.
- `localhost` must **not** resolve to an IPv6 address, for example, “`::1`”. The install process expects only IPv4 resolution to IP address 127.0.0.1. Any `::1` reference must be commented out in the `/etc/hosts` file.
- The Initial Master Node host name must not resolve to multiple IPv4 addresses, and this includes lookup in `/etc/hosts`.

Configuring Secure DNS

If Secure DNS is being used in the environment where the product is installed, the DNS must be configured so that the ACL allows connections from all of the following:

- Every machine in the Kubernetes cluster, master and worker nodes.
- The network address range of Kubernetes pods in Classless Inter-Domain Routing (CIDR) format. By default, this is 172.16.0.0/16.
- The network address range of Kubernetes services in Classless Inter-Domain Routing (CIDR) format. By default, this is 172.17.17.0/24.
- If the Database is being used, every machine in the Database cluster.

Testing Forward and Reverse DNS Lookup

Test that the forward and reverse lookup records for all servers were properly configured.

To test the forward lookup, run the commands on every master and worker node in the cluster and on every producer and consumer system, including:

- All master and worker nodes
- All ArcMC, Logger, and ESM hosts

Use the `nslookup` or `host` commands to verify your DNS configuration. (Do not use the `ping` command.) You must run the `nslookup` commands on every server specified in your `/etc/resolv.conf` file. Every server must be able to perform forward and reverse lookup properly and return the exact same results.

If you have a public DNS server specified in your `/etc/resolv.conf` file (such as the Google public DNS servers 8.8.8.8 or 8.8.4.4), you must remove this from your DNS configuration.

Running the Commands

Run the commands as follows. Expected sample output is shown below each command:

```
# hostname
```

For CentOS/RHEL 7.x or later, use `# hostnamectl`

```
mastern.yourcompany.com
```

```
# hostname -s
```

```
mastern.yourcompany.com
```

```
# hostname -f
```

```
mastern.yourcompany.com
```

```
# hostname -d
```

```
mastern.yourcompany.com
```

```
 # nslookup mastern.yourcompany.com
```

```
           Server:                      192.168.0.53
Address:                           192.168.0.53#53
Address:                             192.168.0.1
Name:                           mastern.example.com
```

```
 # nslookup mastern
```

```
Server:                                    192.168.0.53
Address:                                 192.168.0.53#53
Name:                                mastern.example.com
Address: 192.168.0.1
```

```
 # nslookup 192.168.0.1
```

```
Server:                                                       192.168.0.53
Address:                                                   192.168.0.53#53
1.0.168.192.in-addr.arpa name = mastern.example.com.
```

Creating the NFS Shares

NFS storage is used by all nodes in the Platform Kubernetes cluster to maintain state information about the infrastructure and to store other pertinent data.

- For high availability, the NFS server must run on a highly available device separate from the Kubernetes cluster nodes. This topic provides the information to manually configure the NFS share to be used by the Kubernetes cluster.
- If the service availability requirements of your environment do not require the NFS server to be highly available and if you plan to use ArcSight Platform Installer to automate the installation, you can also automate the configuration of the NFS server. To do so, use the NFS type `new` in the install configuration file, and skip this NFS server manual configuration topic.

> For optimal security, secure all NFS settings to allow only required hosts to connect to the NFS server.

- "Understanding NFS Prerequisites" below
- "Understanding NFS Directory Structure" on page 42
- "Exporting the NFS Configuration" on page 43
- "Testing NFS" on page 43
- "Securing NFS" on page 44

Understanding NFS Prerequisites

To ensure that your environment meets the prerequisites:

1. On the external NFS server, ensure ports 111, 2049, and 20048 are open.
2. Ensure the required packages `rpcbind` and `nfs-utils` are installed and the related services are enabled on the NFS server.
3. Check for prior installation.
4. Install any missing required packages.
5. Enable the required services by running the following commands:

   ```
   # systemctl enable rpcbind
   ```

```
# systemctl start rpcbind
# systemctl enable nfs-server
# systemctl start nfs-server
```

6. For the minimum required sizes for each of the NFS installation directories, see the "Network File System section" in the *Technical Requirements for ArcSight Platform*.

Checking for Prior Installation

To check for prior installation of these packages:

1. Set up the yum repository on your server.
2. Run the following command:

   ```
   # yum list installed <package name>
   ```

   This command returns an exit status code where:

   - 0 indicates the package is installed
   - 1 indicates the package is not installed (does not check whether the package is valid)

Installing a Missing Required Package

To install a missing required package, run the following command:

```
# yum -y install <package name>
```

Understanding NFS Directory Structure

To create the NFS directory structure:

1. Log in to the NFS server and create the following.

Item	Name	Specification	Example Command
GROUP	`arcsight`	GID of 1999	`# groupadd -g 1999 arcsight`
USER	`arcsight`	UID of 1999	`# useradd -u 1999 -g 1999 -u 1999 -d /opt/arcsight arcsight`
NFS root directory	`/opt/arcsight-nfs`	–	`# mkdir -p /opt/arcsight-nfs`

> If you have previously installed any version of CDF, you must remove all NFS shared directories from the NFS server before you proceed. To do this, run the following command for each directory: `rm -rf <path to shared directory>`

2. For each directory listed in the table below, run the following command to create each NFS shared directory.

```
# mkdir -p <path to shared directory>
```

For example:

```
mkdir -p /opt/arcsight-nfs/itom-vol
```

Directory	Mount Point Example
`<NFS_root_DIRECTORY>/itom-vol`	`/opt/arcsight-nfs/itom-vol`
`<NFS_root_DIRECTORY>/db-single-vol`	`/opt/arcsight-nfs/db-single-vol`
`<NFS_root_DIRECTORY>/db-backup-vol`	`/opt/arcsight-nfs/db-backup-vol`
`<NFS_root_DIRECTORY>/itom-logging-vol`	`/opt/arcsight-nfs/itom-logging-vol`
`<NFS_root_DIRECTORY>/arcsight-volume`	`/opt/arcsight-nfs/arcsight-volume`

3. The permission setting of each parent directory and each subdirectory must be recursively set. If it is not, run the following command to update the permissions:

```
# chmod -R <path to shared directory>
```

For example:

```
#chmod -R 755 /opt/arcsight-nfs
```

4. Set the ownership in this structure to UID 1999 and GID 1999. Change the directory to `/opt`, and then run the following command:

   ```
   # chown -R 1999:1999 <NFS_root_DIRECTORY>
   ```

   > If you use a UID/GID different than 1999/1999, then provide it during the CDF installation in the install script arguments `--system-group-id and --system-user-id`. In addition, if you are using NetApp with NFSv4 configuration, consider applying stickybits to al <NFS_root_directory> shares with: `# chmod g+s #chmod w+s`

Exporting the NFS Configuration

For every NFS volume, run the following set of commands on the External NFS server based on the IP address. You will need to export the NFS configuration with appropriate IPs in order for the NFS mount to work properly.

For every node in the cluster, you must update the configuration to grant the node access to the NFS volume shares. On the NFS server, edit the `etc/exports` file and add all the shared volumes to the file.

For example, this is a `/etc/exports` file entry for IP address 192.168.1.0 and for all of the volumes:

```
/opt/arcsight-nfs/itom-vol 192.168.1.0/24(rw,sync,anonuid=1999,anongid=1999,all_squash)
/opt/arcsight-nfs/db-single-vol 192.168.1.0/24(rw,sync,anonuid=1999,anongid=1999,all_
squash)
/opt/arcsight-nfs/db-backup-vol 192.168.1.0/24(rw,sync,anonuid=1999,anongid=1999,all_
squash)
/opt/arcsight-nfs/itom-logging-vol 192.168.1.0/24(rw,sync,anonuid=1999,anongid=1999,all_
squash)
/opt/arcsight-nfs/arcsight-volume 192.168.1.0/24(rw,sync,anonuid=1999,anongid=1999,all_
squash)
```

1. Save the `/etc/exports` file, then run the following command:

   ```
   exportfs -ra
   ```

2. Synchronize the time on the NFS server and the time on the other servers in the cluster.
3. If you add more NFS shared directories later, you must restart the NFS service.

Testing NFS

Test a mount to the NFS that might be used to determine the supported version.

1. Create a test directory by running the following command:

   ```
   # mkdir /mnt/nfstest
   ```

2. Create a test mount by running the following command:

   ```
   # mount -t nfs -o nfsvers=4 192.168.1.15:/opt/arcsight-nfs/arcsight-volume
   /mnt/nfstest
   ```

3. Confirm the command.
4. Remove the mount by running the following command:

   ```
   # umount /mnt/nfstest
   ```

Securing NFS

You must secure the NFS shared directories from external access. This section provides one method for ensuring security while maintaining access to master and worker nodes in the cluster. However, you can use a different approach to adequately secure NFS.

For example:

```
firewall-cmd --zone=public --add-port=111/tcp --permanent
firewall-cmd --zone=public --add-port=2049/tcp --permanent
firewall-cmd --zone=public --add-port=20048/tcp --permanent
```

To secure NFS:

1. Log in to the master node as root user.
2. Remove the firewall definition for all NFS ports using the following command:

   ```
   NFS_PORTS=('111/tcp' '111/udp' '2049/tcp' '20048/tcp')
   for port in "${NFS_PORTS[@]}"; do firewall-cmd --permanent --remove-port $port;
   done;
   ```

3. (Conditional) If you have deployed Intelligence by using scripts, remove all rich rules using the following command:

   ```
   firewall-cmd --list-rich-rules |xargs -I '{}' firewall-cmd --permanent --remove-
   rich-rule '{}'
   ```

4. (Conditional) If you want to expose NFS shares to other hosts such as other master and worker nodes, execute the following command:

   ```
   firewall-cmd --add-source="IP_address or CIDR expression of host or hosts" --
   zone="trusted" --permanent
   ```

5. Reload the new firewall configuration using the following command:

```
firewall-cmd --reload
```

6. Restart NFS using the following command:

```
exportfs -ra && systemctl restart rpcbind && systemctl restart nfs-server
```

7. Continue to"Creating the First System Admin User" on page 302

Disabling Swap Space

Disabling of swap space on all master and worker nodes is necessary to evenly distribute resources and not allocate swap space.

> This procedure does not apply to database nodes, because the database requires swap space. In the case where the database and Kubernetes master and worker nodes are co-located, such as an all-in-one single node deployment, you must enables swap because it is a hard requirement for the database installation. In such a single-node scenario, Kubernetes will operate properly with swap enabled because pod allocation is only to a single node, so swap does not affect the allocation logic.

To disable swap space:

1. Log on to the node.
2. Run the following command to disable the swap process.

```
# swapoff -a
```

3. Open the `/etc/fstab` file in a supported editor.
4. Comment out the lines that display `swap` as the disk type, then save the file. For example:

```
#/dev/mapper/centos_shcentos72x64-swap swap
```

Downloading the Installation Packages

You can use this procedure for an initial install and upgrade.

Follow the "Checklist: Upgrading Your Environment" on page 385 to ensure a successful upgrade.

To download the packages:

1. Launch a terminal session and log in to the primary master node as `root`.

   > If you select to install as a sudo user, log in to the primary master node as the non-root user.

2. In the ArcSight Platform release notes, *"Downloading and Installing the ArcSight Platform Capabilities section"* identify and access the files to download into a directory.
3. Unzip `cdf-2020.08.00153-x.x.x.x.zip` into a directory, which we'll refer to going forward as `{unzipped-cdf-dir}`.

   > Do not unzip under `/root` or any sub directory of it.

4. Move the ArcSight Metadata file into the `{unzipped-cdf-dir}/arcsight/metadata/` directory.

   > Do not untar the file. The filename must have the prefix `arcsight-installer-metadata`. Also, do not move signature files as it might cause warnings and errors from the installer script. Therefore, only copy ONLY.tar files you need based on your YAML file.

5. For each ArcSight product to install, move the corresponding image tar file into the `{unzipped-cdf-dir}/arcsight/images/` directory.

   > Do not untar the file. Also, do not move signature files as it might cause warnings and errors from the installer script. Therefore, only copy ONLY.tar files you need based on your YAML file.

   For example, if you deploy Fusion, Recon, SOAR, and Transformation Hub:

ArcSight Installer	`arcsight-installer-metadata-x.x.x.x.tar`
Fusion	`fusion-x.x.x.x.tar`
Recon	`recon-x.x.x.x.tar`
SOAR	`soar-x.x.x.x.tar`
Transformation Hub	`transformationhub-x.x.x.x.tar`

Installing with the sudo User Account

If you choose to run the Installer as a sudo user, the root user must first grant the non-root (`sudo`) user installation permission. The sudo user must have permission to execute scripts under temporary directory `/tmp` on all master and worker nodes.

There are two distinct file edits that need to be performed: first on the Initial Master Node only, and then on all remaining master and worker nodes. These file edits are detailed below. In addition, before installing CDF, the `CDF-updateRE.sh` script must be modified to install CDF as a sudo user.

- "Editing the sudoers File on the Initial Master Node" below
- "Editing the sudoers File on the Remaining Master and Worker Nodes" on the next page
- "Modifying the cdf-updateRE.sh Script" on page 49

Editing the sudoers File on the Initial Master Node

> ⚠ Make the following modifications **only on the Initial Master Node.**

First, log on to the initial master node as the root user. Then, using `visudo`, edit the `/etc/sudoers` file and add or modify the following lines.

> In the following commands you must ensure there is, at most, a single space character after each comma that delimits parameters. Otherwise, you might get an error similar to this when you attempt to save the file. `>>> /etc/sudoers: syntax error near line nn <<<`

To edit sudoers:

1. Add the following `Cmnd_Alias` line to the **command aliases** group in the `sudoers` file.

```
Cmnd_Alias CDFINSTALL = <CDF_installation_package_
directory>/installers/cdf/scripts/pre-check.sh, <CDF_installation_package_
directory>/install, <K8S_HOME>/uninstall.sh, /usr/bin/kubectl, /usr/bin/docker,
/usr/bin/mkdir, /bin/rm, /bin/su, /bin/chmod, /bin/tar, <K8S_
HOME>/scripts/uploadimages.sh, <K8S_HOME>/scripts/cdf-updateRE.sh, <K8S_
HOME>/bin/kube-status.sh, <K8S_HOME>/bin/kube-stop.sh, <K8S_HOME>/bin/kube-start.sh,
<K8S_HOME>/bin/kube-restart.sh, <K8S_HOME>/bin/env.sh, <K8S_HOME>/bin/kube-
common.sh, <K8S_HOME>/bin/kubelet-umount-action.sh, /bin/chown
```

> For an AWS installation, the `cdf-updateRE.sh` script has the path: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

> If you will be specifying an alternate `tmp` folder using the `--tmp-folder` parameter, ensure that you specify the correct path to `<tmp path>/scripts/pre-check.sh` in the `Cmnd_Alias` line.

- Replace the `<CDF_installation_package_directory>` with the directory where you unzipped the installation package. For example, `/tmp/cdf-2020.08.0xxx`.
- Replace `<K8S_HOME>` with the value defined from a command line. By default, `<K8S_HOME>` is `/opt/arcsight/kubernetes.`

2. Add the following lines to the **wheel users** group, replacing `<username>`with your `sudo` username.

   ```
   %wheel ALL=(ALL) ALL
   ```

   ```
   cdfuser ALL=NOPASSWD: CDFINSTALL
   ```

   ```
   Defaults: <username> !requiretty
   ```

   ```
   Defaults: root !requiretty
   ```

3. Locate the `secure_path` line in the `sudoers` file and ensure the following paths are present.

   ```
   Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
   ```

   By doing this, the `sudo` user can execute the `showmount, curl, ifconfig` and `unzip` commands when installing the CDF Installer.

4. Save the file.

Editing the sudoers File on the Remaining Master and Worker Nodes

> Make the following modifications only **on the remaining master and worker nodes.**

Log in to each master and worker node. Then, using `visudo`, edit the `/etc/sudoers` file and add or modify the following:

> In the following commands you must ensure there is, at most, a single space character after each comma that delimits parameters. Otherwise, you might get an error similar to this when you attempt to save the file. `>>> /etc/sudoers: syntax error near line nn <<<`

To edit sudoers:

1. Add the following `Cmnd_Alias` line to the **command aliases** group in the sudoers file.

   ```
   Cmnd_Alias CDFINSTALL = /tmp/pre-check.sh, /tmp/ITOM_Suite_Foundation_Node/install,
   <K8S_HOME>/uninstall.sh, /usr/bin/kubectl, /usr/bin/docker, /usr/bin/mkdir, /bin/rm,
   /bin/su, /bin/chmod, /bin/tar, <K8S_HOME>/scripts/uploadimages.sh, <K8S_
   HOME>/scripts/uploadimages.sh, <K8S_HOME>/scripts/cdf-updateRE.sh, <K8S_
   ```

```
HOME>/bin/kube-status.sh, <K8S_HOME>/bin/kube-stop.sh, <K8S_HOME>/bin/kube-start.sh,
<K8S_HOME>/bin/kube-restart.sh, <K8S_HOME>/bin/env.sh,<K8S_HOME>/bin/kube-common.sh,
<K8S_HOME>/bin/kubelet-umount-action.sh, <K8S_HOME>/scripts/uploadimages.sh,
/bin/chown, /usr/bin/cp
```

- Replace <K8S_HOME> which will be used from the command line. By default, <K8S_HOME> is /opt/arcsight/kubernetes.

2. Add the following lines to the **wheel users** group, replacing <username> with your sudo username.

```
%wheel ALL=(ALL) ALL
```

```
cdfuser ALL=NOPASSWD: CDFINSTALL
```

```
Defaults: <username>!requiretty
```

```
Defaults: root !requiretty
```

3. Locate the secure_path line in the sudoers file and ensure the following paths are present.

```
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
```

By doing this, the sudo user can execute the showmount, curl, ifconfig and unzip commands when installing the CDF Installer.

4. Save the file.
5. Repeat the process for each remaining master and worker node.

Modifying the cdf-updateRE.sh Script

In addition to the steps listed above, the following additional step is required for sudo user installation of CDF.

The cdf-updateRE.sh script is used in installation and other utility operations in CDF and CDF-based products (such as Transformation Hub). To install CDF as the sudo user, you must modify the script.

To modify the script:

1. In the location where you unzip the installer archive, modify the script <unzipped CDF directory>/scripts/cdf-updateRE.sh file in a text editor as follows.
   - Comment out the line containing the text exit 1
   - Add the following line inside the if block.

   ```
   export K8S_HOME=<install directory>
   ```

For Example

```
if [[ -z "${K8S_HOME}" ]]; then
echo "K8S_HOME not set. If running on fresh installation, please use new shell
session"
# exit 1
export K8S_HOME=/opt/arcsight/kubernetes
fi;
```

2. Save the file and then proceed to CDF installation as a sudo user.

Using ArcSight Platform Installer

You can use ArcSight Platform Installer to build your environment. ArcSight Platform Installer takes care of the end-to-end installation process, which starts from configuring the prerequisites to completing the post-installation configurations.

- "Using the Configuration Files" below
- "Understanding the Installation Commands" on the next page
- "Configuring the System Clock of the Database Node" on page 52
- Using ArcSight Platform Installer to Deploy

Using the Configuration Files

The installer uses the settings you place in the deployment configuration file in order to determine the capabilities to deploy, the machines to deploy to, and how to configure the capabilities.

The configuration file is in a .yaml format and the completed description of the format and options available is specified in the template file `<unzipped-cdf-dir>/arcsight/config/install-config-template.yaml`. You should not use this template file for deployment, but rather to identify all the settings to include in your configuration file.

The directory `<unzipped-cdf-dir>/arcsight/config/` contains multiple example deployment configuration .yaml files that describe a variety of deployment scenarios. Use these example files as a starting point for your specific deployment scenario.

For example, to deploy ESM Command Center and Transformation Hub in a high-availability environment, start with the `example-install-config-esm_and_transformation_hub-high_availability.yaml` file.

The example has placeholders for your specific environment, such as host names, so you will need to edit the example file before using it.

> The "suite > config-params" section of the example deployment configuration .yaml files include the internal ID of configuration properties that cannot be configured easily after installation. For a description of each property internal ID in the example deployment configuration .yaml files, see Configuring the Deployed Capabilities. After installation, you can easily configure most properties (those not in the example deployment configuration .yaml files) using the CDF Management Portal, where descriptions for all properties are supplied as tooltips.

Understanding the Installation Commands

This table provides information about the installation commands and their purpose.

> These instructions use the primary commands with defaults for the most straightforward installation experience. Additional options are available if needed and are explained when you run the command `./arcsight-install --help`.

Script	Purpose
`./arcsight-install -c /opt/my-install-config.yaml --cmd preinstall`	The preinstall command attempts to install automatically any missing operating system package dependencies using the yum command. Therefore, be sure yum is configured on all nodes to automatically be able to download the packages from a package repository. It runs checks on all hosts specified in the install config file and reports if they meet the requirements. It also modifies the configuration of all hosts specified in the install config file so each host meets the required system configuration for the components that will be installed on each host. Not all required system configurations can be handled by this command. The items that must be manually configured will be reported. It also installs or configures NFS as specified in the install config file.
`./arcsight-install -c /opt/my-install-config.yaml --cmd install`	The install command installs or configures the Database, Container Deployment Foundation (CDF) cluster, and ArcSight capabilities as specified in the install config file.
`./arcsight-install -c /opt/my-install-config.yaml --cmd postinstall`	The postinstall command performs the post-installation configurations.

Configuring the System Clock of the Database Node

A network time server must be available in your environment. chrony implements this protocol and is installed by default on some versions of RHEL and CentOS. Ensure that chrony is installed on every node. Click here for more information.

Using ArcSight Platform Installer to Deploy

ArcSight Platform Installer takes care of the prerequisites, software installations, and post-installation configurations.

> Before building your environment, ensure the firewall is running on the CDF nodes.

> ✓ To copy the metadata file and the images to their corresponding directories, see "Downloading the Installation Packages" on page 385.

To use the installer to deploy:

1. Launch a terminal session and log in to the master node as `root`.
2. Change to the following directory:

   ```
   cd <unzipped-cdf-dir>/arcsight/config/
   ```

3. Select an example install config file in the directory that most closely matches the deployment you need.

   > There is an explanation at the top of each example file and additional explanations are available in the `<unzipped-cdf-dir>/arcsight/config/` directory. Do not use the `install-config-template.yaml` file for your deployment, as it is for information purposes only.

4. Make a copy of the selected example file. For example, in these instructions, we will name the copy the following:

   ```
   /opt/my-install-config.yaml
   ```

5. Edit the following file as needed:

   ```
   /opt/my-install-config.yaml
   ```

   Each example install config file explains the minimal changes that must be made before performing the installation with the example file.

   > Depending on your specific deployment, you might need to make additional modifications that are not described in the example file. Additional explanations are available in the `<unzipped-cdf-dir>/arcsight/config/install-config-template.yaml` file.

6. Change to the following directory:

   ```
   <unzipped-cdf-dir>
   ```

7. Execute the command to check all the nodes and deploy all the prerequisites.

   ```
   ./arcsight-install -c /opt/my-install-config.yaml --cmd preinstall
   ```

> When you execute the script, the installer prompts you for the username and password you provided for each hostname specified. You need to provide this information only once for each hostname. The installer sets up secure passwordless ssh using certificates so executing commands later is seamless. Valid password specifications include: Length: between 8- 30 Can contain: letters, digits and special characters Valid special characters: _ ! % @ & Valid examples: 9badm1N_X, my6AsW@rd, mypasS_w0?d

8. Execute the command to install the Database, CDF, and ArcSight capabilities.

```
./arcsight-install -c /opt/my-install-config.yaml --cmd install
```

Database

If your install config file specifies to install the Database, the installer displays prompts for:

- Accept License Agreement
- Database admin password
- Database app admin password
- Database search username

> Be patient as the Database installation might take time to complete. The Database might need time to create indexes and complete setup tasks. The Database installation might appear to be complete; however, if you start the product before the Database installation is complete, you might experience errors and performance issues.

CDF and ArcSight Capabilities

Next, the installer displays prompts for:

- Accept License Agreement (again)

> If the installer discovers warnings while running a check of the node hardware configuration, a prompt appears asking you to confirm the warnings and continue.

- CDF admin password

> Be patient as the installation might take time to complete, depending on the number of suite products and cluster nodes being installed. For example, a small cluster might take 40 minutes or more to complete. You can monitor the progress of the installer in the terminal.

9. After the install command completes, run the pod command to check the pod status. Before continuing to the post-installation step, all pods must be in `Running` or `Completed` status.

   ```
   kubectl get pods -A
   ```

10. View additional cluster status, including logs (as needed).

    a. Log in to the CDF Management Portal using the CDF admin username and password you provided.

    b. Navigate to **Cluster** > **Dashboard**.

    c. In the Kubernetes Dashboard, select **Namespace arcsight-installer-***.

    d. Navigate to pods, then select the pod to inspect.

    e. To view the logs for the pod, click the **View Logs** icon in the upper-right corner of the UI.

    f. In the **Logs from** menu, select a different container to view relevant logs.

11. Execute the following command to perform the post-installation configurations.

    ```
    ./arcsight-install -c /opt/my-install-config.yaml --cmd postinstall
    ```

12. When you run this command, the installer displays the following prompt:

    ```
    Are you sure all arcsight pods are running and you want to continue? (y/N)
    ```

13. After ensuring that all the ArcSight pods are running, specify **y**.

    > If the Intelligence capability is deployed, the interset-api, searchmanager-api, and searchmanager-engine pods will not go into Running status until after the postinstall is complete, when the schema is created in the database. However, these pods do not interrupt the post-installation steps, so you can continue with the post-installation steps even if these specific pods are not yet in the Running status.

14. To make master and worker node labels manageable from the CDF Management Portal after installation, perform the following steps:

    a. Log in to the CDF Management Portal using the CDF admin username and password you provided.

    b. Navigate to Cluster > Nodes.

    c. In the Predefined Labels section, enter each of the labels applied to the master and worker nodes in the `/opt/my-install-config.yaml` file. Each label needs to be entered only once even though it might appear multiple times in your `/opt/my-install-config.yaml` file. The format to enter is `<label_name>:yes` (case-sensitive), for example `fusion:yes`, and then click the + icon to submit the label. After you submit the label, you will immediately see display on the relevant master and worker nodes in the Nodes section on the same page as configured in your `/opt/my-install-config.yaml` file.

d. Continue to the section "Performing Post-deployment Configurations" on page 297.

Performing a Manual Deployment

This section explains how to set up your deployment architecture for the Platform that runs on-premises, such as on a local network.

Checklist: Manually Installing the Platform Infrastructure

Use the following checklist to install and configure the Platform infrastructure. Perform the tasks in the listed order.

	Task	See
□	1. Complete the Planning Checklist.	Checklist: Planning to Deploy the Platform
□	2. Prepare your on-premises environment for the CDF.	Preparing Your Environment
□	3. (Conditional) To deploy Intelligence or Recon, install the ArcSight Database.	Installing the Database
□	4. Install the CDF.	Installing CDF
□	5. Deploy the Platform and capabilities.	Deploying ArcSight Platform and Capabilities
□	6. Configure the database.	Completing Database Setup
□	7. Complete the deployment process.	Performing Post-deployment Configurations

Installing the Database

This section provides information about configuring the database server and installing the ArcSight Database.

- "Preparing the Database Node for Installation" on the next page
- Configuring BIOS for Maximum Performance
- "Enabling Passwordless Communication" on page 60
- "Modifying the System Clock" on page 61
- "Configuring and Installing the Database Server" on page 61

Preparing the Database Node for Installation

To prepare the database:

1. Provision the server with at least 2 GB of swap space.

   > In case the pre-check on swap space fails after provisioned 2 GB on swap, a provision swap with 2.2 GB should solve the problem.

2. Add the following parameters to `/etc/sysctl.conf`.

Parameter	Description
`net.core.somaxconn = 1024`	Increases the number of incoming connections
`net.core.wmem_max = 16777216`	Sets the send socket buffer maximum size in bytes
`net.core.rmem_max = 16777216`	Sets the receive socket buffer maximum size in bytes
`net.core.wmem_default = 262144`	Sets the receive socket buffer default size in bytes
`net.core.rmem_default = 262144`	Controls the default size of receive buffers used by sockets
`net.core.netdev_max_backlog = 100000`	Increase the length of the network interface input queue
`net.ipv4.tcp_mem = 16777216 16777216 16777216`	
`net.ipv4.tcp_wmem = 8192 262144 8388608`	
`net.ipv4.tcp_rmem = 8192 262144 8388608`	
`net.ipv4.udp_mem = 16777216 16777216 16777216`	
`net.ipv4.udp_rmem_min = 16384`	
`net.ipv4.udp_wmem_min = 16384`	
`vm.swappiness = 1`	Defines the amount and frequency at which the kernel copies RAM contents to a swap space For more information, see Check for Swappiness.

3. Add the following parameters to `/etc/rc.local`.

   > The following commands assume that sdb is the data drive (i.e. /opt), and sda is the operating system/catalog drive.

Parameter	Description
`echo deadline > /sys/block/sdb/queue/scheduler`	Resolve FAIL (S0150)
`/sbin/blockdev --setra 4096 /dev/sdb`	Resolve FAIL (S0020) when database resides on `/dev/sdb`
`echo always > /sys/kernel/mm/transparent_hugepage/enabled`	
`cpupower frequency-set --governor performance`	Resolve WARN (S0140/S0141) (**CentOS only**)

4. To increase the process limit, add the following to `/etc/security/limits.d/20-nproc.conf`:

   ```
   * soft nproc 10240
   * hard nproc 10240
   * soft nofile 65536
   * hard nofile 65536
   * soft core unlimited
   * hard core unlimited
   ```

5. In `/etc/default/grub`, append line `GRUB_CMDLINE_LINUX` with `intel_idle.max_cstate=0 processor.max_cstate=1`. For example:

   ```
   GRUB_CMDLINE_LINUX="vconsole.keymap=us crashkernel=auto  vconsole.font=latarcyrheb-sun16 rhgb quiet intel_idle.max_cstate=0 processor.max_cstate=1"
   ```

   Execute the following command:

   ```
   grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

6. Use `iptables` to disable the firewall **WARN (N0010):**

   ```
   iptables -F
   iptables -t nat -F
   iptables -t mangle -F
   iptables -X
   systemctl mask firewalld
   systemctl disable firewalld
   systemctl stop firewalld
   ```

> The database requires several ports to be open on the local network. Micro Focus does not recommend that you place a firewall between nodes (all nodes should be behind a firewall), but if you must use a firewall between nodes, ensure that all the database ports are available. For more information, see Firewall Considerations.

7. Set SELinux to permissive mode in `/etc/selinux/config`.

   ```
   SELINUX=permissive
   ```

   For more information, see SELinux Configuration.

8. Reboot the system for your changes to take effect.

Configuring BIOS for Maximum Performance

Depending on your hardware, you might be able to access options to configure power and performance. Configure the system for maximum performance in the BIOS while the system is powering on. For example, for HPE hardware, the following setting is available.

```
System Configuration > BIOS/Platform Configuration (RBSU) > Power Management
> HPE Power Profile > Maximum Performance
```

Enabling Passwordless Communication

This section describes how to configure passwordless communication from the node1 server to all of the node servers in the cluster.

> You must repeat the authentication process for all nodes in the cluster.

To configure passwordless communication:

1. On the node1 server, run the `ssh-keygen` command:

   ```
   ssh-keygen -q -t rsa
   ```

2. Copy the key from node1 to all of the nodes, including node1, using the node IP address:

   ```
   ssh-copy-id -i ~/.ssh/id_rsa.pub root@11.111.111.111
   ```

   The system displays the key fingerprint and requests to authenticate with the node server.

3. Enter the required credentials for the node.
4. The operation is successful when the system displays the following message:

```
Number of key(s) added: 1
```

5. To verify successful key installation, run the following command from node1 to the target node to verify that node1 can successfully log in:

   ```
   ssh root@11.111.111.111
   ```

Modifying the System Clock

A network time server must be available. chrony implements this protocol and is installed by default on some versions of RHEL and CentOS. chrony must be installed on every node.

Verify the chrony configuration by using the command:

```
# chronyc tracking
```

To install chrony, start the chrony daemon, then verify operation with these commands:

```
# yum install chrony
# systemctl start chronyd
# systemctl enable chronyd
# chronyc tracking
```

Configuring and Installing the Database Server

> Before installing the database, ensure that you estimate the storage needed for the incoming EPS (event per second) and event size, and also evaluate the retention policy accordingly.

To install the database:

1. On the Database cluster node1 server, create a folder for the database installer.

   For example:

   ```
   mkdir /opt/arcsight-database
   ```

   > `/opt/arcsight-database` should not be under `/root` or `/opt/vertica`.

2. From the master node where you performed the Downloading Installation Packages steps, copy the following directory on the Database cluster node1 server:

   ```
   {unzipped-cdf-dir}/arcsight/database/db-installer_x.x.x-x.tar.gz file to the
   /opt/arcsight-database
   ```

3. To extract the installer file and place it in the correct directory, run the following commands:

   ```
   cd /opt/arcsight-database
   tar xvfz db-installer_x.x.x.x.tar.gz
   ```

4. Edit the `config/db_user.properties` file. The `hosts` property is required.

Property	Description
`hosts`	A comma separated list of the database servers in IPv4 format (for example, 1.1.1.1,1.1.1.2,1.1.1.3). If it is necessary to construct the cluster, avoid using local loopback (localhost, 127.0.0.1, etc.).

5. Install the database.

   ```
   ./db_installer install
   ```

6. When prompted, create the following users:
   Database administrator: Database administrator user account to be used during database deployment, configuration, upgrade, and debugging. For security reasons, the Platform deployed capabilities will not ask you for the credentials for this user.

   App admin user: A regular database user granted elevated permissions for performing operations on the database to manage the database, schema, and resource pools. The credentials for this user will need to be provided later in the CDF Management Portal when you are deploying capabilities.

   Search user: A regular database user with permissions restricted to event search operations. The credentials for this user will need to be provided later in the CDF Management Portal when you are deploying capabilities.

   > For a list of options that you can specify when installing the database, see Understanding the Database Installer Options.

7. Create the schema.

   ```
   ./db_installer create-schema
   ```

8. Monitor your database cluster status constantly. For more information, see Monitoring the Database.

- **Database nodes status:** Ensures all nodes are up
- **Database nodes storage status:** Ensures storage is sufficient

Installing CDF

Checklist: Preparing Your Environment for CDF

Use the following checklist to prepare your ArcSight Platform environment. Perform the tasks in the listed order.

	Task	See
☐	1. Verify your firewall settings.	Ensuring Your Firewall Settings
☐	2. Verify masquerade settings.	Enabling Masquerade Setting in Firewall
☐	3. Verify the system clock.	Modifying the System Clock
☐	4. Verify password settings.	Checking Password Authentication Settings
☐	5. Verify OS packages are installed.	Ensuring Required OS Packages Are Installed
☐	6. Verify algorithms.	Checking MAC and Cipher Algorithms
☐	7. Set system parameters	.Setting System Parameters (Network Bridging)
☐	8. Review example files.	Understanding Example Files
☐	9. Remove libraries.	Removing Libraries
☐	10. Configure settings.	Configuring Elasticsearch Settings

Preparing Your Environment

- "Checking Your Firewall Settings" on the next page
- "Enabling the Masquerade Setting in the Firewall" on the next page
- "Modifying the System Clock" on the next page
- "Checking Password Authentication Settings" on page 65

- "Ensuring That Required OS Packages Are Installed" on the next page
- "Checking MAC and Cipher Algorithms" on page 67
- "Setting System Parameters (Network Bridging)" on page 68
- "Understanding Example Files" on page 69
- "Removing Libraries to Prevent Ingress" on page 69
- "Configuring Elasticsearch Settings" on page 69

Checking Your Firewall Settings

Ensure that the firewalld.service is enabled and running on all nodes.

```
# systemctl start firewalld
```

```
# systemctl enable firewalld
```

Enabling the Masquerade Setting in the Firewall

You must enable the masquerade setting only when the firewall is enabled.

Run the following command on all master and worker nodes to check whether the masquerade setting is enabled:

```
# firewall-cmd --query-masquerade
```

- If the returned value is yes, the masquerade setting is enabled.
- If the returned value is no, run the following commands to enable the masquerade setting in the firewall.

```
# firewall-cmd --add-masquerade --permanent
# firewall-cmd --reload
```

Modifying the System Clock

A network time server must be available. chrony implements this protocol and is installed by default on some versions of RHEL and CentOS. chrony must be installed on every node.

Verify the chrony configuration by using the command:

```
# chronyc tracking
```

To install chrony, start the chrony daemon, then verify operation with these commands:

```
# yum install chrony
# systemctl start chronyd
# systemctl enable chronyd
# chronyc tracking
```

Checking Password Authentication Settings

If you use a user name and password authentication for adding cluster nodes during the installation, ensure that the PasswordAuthentication parameter in the `/etc/ssh/sshd_config` file is set to "yes."

There is no need to check the password authentication setting when you add the cluster nodes using a user name and key authentication.

To ensure the password authentication is enabled, perform the following steps on every master and worker node:

1. Log on to the cluster node.
2. Open the following file:

   ```
   /etc/ssh/sshd_config
   ```

3. Check if the parameter `PasswordAuthentication` is set to yes. If not, set the parameter to yes as below.

   ```
   PasswordAuthentication yes
   ```

4. Run the following command to restart the sshd service:

   ```
   systemctl restart sshd.service
   ```

Ensuring That Required OS Packages Are Installed

The packages listed in the following table are required on one or more node types, as shown here. These packages are available in the standard `yum` repositories.

Additional Information

- tar is required for tar images. If you do not have tar installed, the following error displays during installation:

```
2020-12-22T20:37:47.380684729-06:00 FATAL The metadata package
arcsight/metadata/arcsight-installer-metadata-20.11.0.16.tar does not have the correct
internal structure. Refer to /tmp/install.20201222203742.log file for detail information. If
need, please contact system administrator or Micro Focus support.
```

- Below are yum example lines including all the required packages for each node type.
  - **Master Nodes**

    ```
    # yum install conntrack-tools container-selinux curl device-mapper-libs
    httpd-tools java-1.8.0-openjdk libgcrypt libseccomp libtool-libs
    libtool-ltdl lvm2 net-tools nfs-utils rpcbind socat systemd-libs unzip
    bind-utils tar
    ```

  - **Worker Nodes**

    ```
    # yum install conntrack-tools container-selinux curl device-mapper-libs
    httpd-tools libgcrypt libseccomp libtool-libs libtool-ltdl lvm2 net-
    tools nfs-utils rpcbind socat systemd-libs unzip tar
    ```

  - **NFS**

    ```
    yum install nfs-utils rpcbind
    ```

Package Name	Required by Master Nodes?	Required by Worker Nodes?	Required by NFS Server?
`conntrack-tools`	Yes	Yes	No
`container-selinux` (package version 2.74 or later)	Yes	Yes	No
`curl`	Yes	Yes	No
`device-mapper-libs`	Yes	Yes	No
`httpd-tools`	Yes	Yes	No
`java-1.8.0-openjdk`	Yes	No	No
`libgcrypt`	Yes	Yes	No
`libseccomp`	Yes	Yes	No
`libtool-ltdl`	Yes	Yes	No
`lvm2`	Yes	Yes	No
`net-tools`	Yes	Yes	No

Package Name	Required by Master Nodes?	Required by Worker Nodes?	Required by NFS Server?
nfs-utils	Yes	Yes	Yes
rpcbind	Yes	Yes	Yes
socat	Yes	Yes	No
systemd-libs (version >= 219)	Yes	Yes	No
unzip	Yes	Yes	No
bind-utils	Yes	Yes	No

If bash-completion is not installed as a package on nodes, a warning is shown. However, the bash-completion package is not required.

To check for prior installation of any of these packages:

1. Set up the yum repository on your server.
2. Run this command:

```
# yum list installed <package name>
```

3. This command returns an exit status code where:

   0 indicates the package is installed

   1 indicates the package is not installed (does not check whether the package is valid)

To install a required package:

Run the following command:

```
# yum -y install <package name>
```

Checking MAC and Cipher Algorithms

Ensure that the /etc/ssh/sshd_config files on every master and worker nodes are configured with at least one of the following values, which lists all supported algorithms. Add only the algorithms that meet the security policy of your organization.

To verify configurations:

- For MAC algorithms:

```
hmac-sha1,hmac-sha2-256,hmac-sha2-512,hmac-sha1-96
```

- For Cipher algorithms:

```
3des-cbc,aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-
ctr,arcfour128,arcfour256,blowfish-cbc
```

For example, you could add the following lines to the `/etc/ssh/sshd_config` files on all master and worker nodes:

```
MACs hmac-sha2-256,hmac-sha2-512
Ciphers aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-ctr
```

Setting System Parameters (Network Bridging)

1. Log in to the node.
2. Run the following command:

   ```
   # echo -e "\nnet.bridge.bridge-nf-call-ip6tables=1\nnet.bridge.bridge-nf-call-
   iptables=1" >> /etc/sysctl.conf
   ```

3. Run the following commands:

   ```
   # modprobe br_netfilter && sysctl -p
   # echo -e '\nmodprobe br_netfilter && sysctl -p' >> /etc/rc.d/rc.local# chmod +x
   /etc/rc.d/rc.local
   ```

4. Open the following file in a text editor:

   ```
   /etc/sysctl.conf
   ```

5. (Conditional) If installing on RHEL or CentOS earlier than version 8.1, change the following if the line exists.

   ```
   net.ipv4.tcp_tw_recycle=1 to net.ipv4.tcp_tw_recycle=0
   ```

6. (Conditional) If installing on RHEL or CentOS 8.1 or later, remove or comment out this line, if it exists.

   ```
   net.ipv4.tcp_tw_recycle=
   ```

7. Save your changes and close the file.
8. Run this command to apply your updates to the node:

   ```
   # sysctl -p
   ```

Understanding Example Files

To view example files:

Example `sysctl.conf` file for RedHat/CentOS version 7.x:

```
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
net.ipv4.ip_forward=1
net.ipv4.tcp_tw_recycle=0
kernel.sem=50100 128256000 50100 2560
```

Example `sysctl.conf` file for RedHat/CentOS 8.1 or later:

```
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
net.ipv4.ip_forward=1
kernel.sem=50100 128256000 50100 2560
```

Removing Libraries to Prevent Ingress

You must remove any libraries that will prevent ingress from starting.

1. Run the following command:

   ```
   # yum remove rsh rsh-server vsftpd
   ```

2. Confirm the removal when prompted.

Configuring Elasticsearch Settings

> This procedure applies only when you are deploying the Intelligence capability.

To ensure the Elasticsearch pods run after deployment and the Elasticsearch cluster is accessible:

1. Launch a terminal session and log in to a worker node.
2. Change to the following directory:

   ```
   cd /etc/
   ```

3. In the `sysctl.conf` file, add the following:

```
vm.max_map_count=262144
```

4. Restart the node:

```
reboot
```

5. Repeat steps 1-4 on all worker nodes.

Configuring and Running CDF

After the installation packages have been downloaded, validated, and uncompressed in the download folder, you are ready to configure and run the CDF Installer. For a complete list of optional parameters, see CDF Installation CLI Commands.

To configure and run the CDF Installer:

1. Log in to one of the local master nodes where you downloaded and extracted the installation files as the root user. (In this document, the selected master node is referred to as the Initial Master Node. You initiate installations from the Initial Master Node.)

   > If you choose to install as a sudo user, log in to the master node as the non-root user.

2. Run the CDF Installer on the Initial Master Node with the following commands.

   > If you choose to install as a sudo user, execute this install command using the sudo command.

   > In the following commands, the *italicized* Docker parameters are optional, based on your network environment.

```
cd {unzipped-cdf-dir}
./install -m <path_to_a_metadata_file> --k8s-home <path_to_installation_
directory> --nfs-server <your_nfs_server_FQDN or IP Address> --nfs-folder
<itom_volume_folder> --docker-http-proxy <your_docker_http_proxy_value> -
-docker-https-proxy <your_docker_https_proxy_value> --docker-no-proxy
<your_docker_no_proxy_value> --ha-virtual-ip <your_HA_ip> --tmp-folder
<your_temp_folder>
```

3. You are prompted for a password, which will be used to log in to the CDF installer portal. For example:

```
cd /opt/arcsight/download/cdf-2020.08.xxxx
./install -m /opt/arcsight/download/cdf-
2020.08.xxxx/arcsight/metadata/arcsight-installer-metadata-
20.11.0.xxx.tar --k8s-home /opt/arcsight/kubernetes --docker-http-proxy
```

```
"http://web-proxy.example.com:8080" --docker-https-proxy "http://web-
proxy.example.com:8080" --docker-no-proxy "localhost,127.0.0.1,my-vmenv-
node1,my-vmenv-node1.example.com,example.com,216.3.128.12" --nfs-server
yourdomain-nfs.yourenterprise.net --nfs-folder /opt/arcsight-nfs/itom-vol
--ha-virtual-ip 216.3.128.12 --tmp-folder /opt/tmp
```

4. You might need to configure some additional parameters, depending on your organization’s OS, network, and storage configurations.

After the CDF Installer is configured and installed, you can use it to deploy one or more products or components into the cluster.

Deploying ArcSight Platform and Capabilities

Configuring and Deploying the Kubernetes Cluster

After you run the CDF Installer, complete the following steps to deploy your Kubernetes cluster.

To configure and deploy:

1. Browse to the Initial Master Node at:

   ```
   https://{master_FQDN or IP}:3000
   ```

2. Log in using *admin* userid and the password you specified during the platform installation. (This URL appears at the successful completion of the CDF installation shown earlier.)
3. On the Security Risk and Governance - Container Installer page, choose the CDF base product metadata version. Then, click **Next**.
4. On the End User License Agreement page, review the EULA and select the *'I agree…'* checkbox. You might optionally choose to have suite utilization information passed to Micro Focus. Then, click **Next**.
5. On the Capabilities page, choose the capabilities and products to install, and then click **Next**.

   > Some capabilities might require other capabilities as prerequisites. Such requirements are noted in the pull-down text associated with the capability. To show additional information associated with the product, click the **>** (greater than) arrow.

6. On the Database page, ensure the PostgreSQL High Availability box is *cleared*. This database is not used by capabilities in SODP.
7. Click **Next**.
8. On the Deployment Size page, choose a size for your deployment based on your planned implementation.

   - **Small Cluster:** Minimum of one worker node deployed (each node should have 4 cores, 16 GB memory, 50 GB disk)
   - **Medium Cluster:** Minimum of 1 worker node deployed (each node should have 8 cores, 32 GB memory, 100 GB disk)
   - **Large Cluster:** Minimum of 3 worker nodes deployed (each node should have 16 cores, 64 GB memory, 256 GB disk)

   > The installation will not proceed until the minimal hardware requirements for the deployment are met.

You can configure additional worker nodes, with each running on its own host system, in subsequent steps.

9. Select your appropriate deployment size, then click **Next**.
10. On the Connection page, an external hostname is automatically populated. This is resolved from the Virtual IP (VIP) specified earlier during the install of CDF (`--ha-virtual-ip parameter`), or the master node hostname if the `--ha-virtual-ip` parameter was not specified during CDF installation. Confirm the VIP is correct, then click **Next**.
11. On the Master High Availability page, if high availability (HA) is desired, select **Make master highly available** and add 2 additional master nodes. (CDF requires 3 master nodes to support high availability.) When complete, or if HA is not desired, click **Next**.
12. For High Availability clusters, the installer prompts you to add additional master nodes depending on your selected deployment size. On the Add Master Node page, specify the details of your first master node and then click **Save**. Repeat for any additional master nodes.

    Master node parameters include:

    - **Host:** FQDN (only) of node you are adding.
    - **Ignore Warnings:** If selected, the installer will ignore any warnings that occur during the pre-checks on the server. If deselected, the add node process will stop and a window will display any warning messages. We recommend that you start with Ignore Warnings cleared to view any warnings displayed. You might then evaluate whether to ignore or rectify any warnings, clear the warning dialog, then click Save again with the box selected to avoid stopping.
    - **User Name:** `root` or `sudo` user name.
    - **Verify Mode:** Choose the verification mode as *Password* or *Key-based*, and then either enter your password or upload a private key file. If you choose **Key-based**, you must first enter a username, then upload a private key file when connecting the node with a private key file.
    - **Device Type:** Select a device type for the master node from one of the following options.
        - Overlay 2: For production, Overlay 2 is recommended.
        - **Thinpool Device:** (Optional) Enter the Thinpool Device path, that you configured for the master node, if any. For example: `/dev/mapper/docker-thinpool`. You must have already set up the Docker thin pool for all cluster nodes that need to use thinpools.
    - **Container data:** Directory location of the container data.

- **flannel IFace:** (optional) Enter the flannel IFace value if the master node has more than one network adapter. This must be a single IPv4 address (or name of the existing interface) and will be used for Docker inter-host communication.

13. On the Add Node page, add the first worker node as required for your deployment by clicking on the **+** (Add) symbol in the box to the right. The current number of nodes is initially shown in red.
14. As you add worker nodes, each Node is then verified for system requirements. The node count progress bar on the Add Node page will progressively show the current number of verified worker nodes you have configured. This progress will continue until the necessary count is met. The progress bar will turn from red to green, which indicates you have reached the minimum number of worker nodes as shown selected in Step 7, above. You might add more Nodes than the minimum number.

    Check the **Allow suite workload to be deployed on the master node** to combine master/worker functionality on the same node (Not recommended for production).

15. On the Add Worker Node dialog, enter the required configuration information for the worker node, and then click **Save**. Repeat this process for each of the worker nodes you wish to add.

    Worker node parameters include:

    - **Type:** Default is based on the deployment size you selected earlier, and shows minimum system requirements in terms of CPU, memory, and storage.
    - **Skip Resource Check:** If your worker node does not meet minimum requirements, select **Skip resource check** to bypass minimum node requirement verification. (The progress bar on the **Add Node** page will still show the total of added worker nodes in green, but reflects that the resources of one or more of these have not been verified for minimum requirements.)
    - **Host:** FQDN (only) of node you are adding.

      When adding any worker node for Transformation Hub workload, on the **Add Node** page, **always** use the FQDN to specify the Node. **Do not use the IP address.**

    - **Ignore Warnings:** If selected, the installer will ignore any warnings that occur during the pre-checks on the server. If deselected, the add node process will stop and a window will display any warning messages. You might start with this deselected in order to view any warnings displayed. You may then evaluate whether to ignore or rectify any warnings, and then run the deployment again with the box selected to avoid stopping.
    - **User Name:** `root` or sudo user name.

- **Verify Mode:** Select a verification credential type: Password or Key-based. Then enter the actual credential.

Once all the required worker nodes have been added, click **Next**.

16. On the File Storage page, configure your NFS volumes.

    (For NFS parameter definitions, refer to the section "Configure an NFS Server environment".) For each NFS volume, do the following:

    - In **File Server**, enter the IP address or FQDN for the NFS server.
    - On the **Exported Path** drop-down, select the appropriate volume.
    - Click **Validate**.

    > All volumes must validate successfully to continue with the installation.

    > A *Self-hosted NFS* refers to the external NFS that you prepared when you configured an NFS server environment. Always choose this value for **File System Type.**

    The following volumes must be available on your NFS server.

CDF NFS Volume claim	Your NFS volume
`itom-vol-claim`	`{NFS_ROOT_DIRECTORY}/itom-vol`
`db-single-vol`	`{NFS_ROOT_DIRECTORY}/db-single-vol`
`db-backup-vol`	`{NFS_ROOT_DIRECTORY}/db-backup-vol`
`itom-logging-vol`	`{NFS_ROOT_DIRECTORY}/itom-logging-vol`
`arcsight-volume`	`{NFS_ROOT_DIRECTORY}/arcsight-volume`

17. Click **Next**.

> **Warning:** After you click **Next**, the infrastructure implementation will be deployed. *Please ensure that your infrastructure choices are adequate to your needs.* An incorrect or insufficient configuration may require a reinstall of all capabilities.

On the **Confirm** dialog, click **Yes** to start deploying master and worker nodes.

Uploading Images for the Capabilities

The **Check Image Availability** page lists the images that are currently loaded into the local Docker Registry from the originally-downloaded set of images. For an initial install, no images should be uploaded. You will upload the images now.

To upload the images to the local Docker Registry:

1. By this point, the images to be installed have already been downloaded from the Micro Focus software site, validated, and uncompressed. None of the files should require downloading at this point, so on the **Download Images** page, click **Next** to skip this step.

   Download Images

   Now that you made the selections, we will download all required container images from external servers.

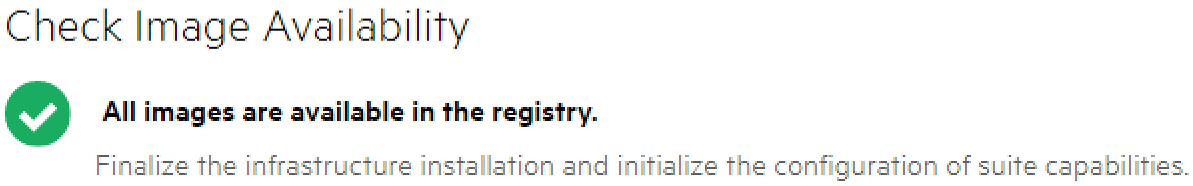
2. Log on to the Initial Master Node in a terminal session as the root or sudo user.
3. Run the following commands to upload the images to the local Docker Registry. Use the `-F <image file>` option on the command line multiple times for each image to upload. Adjust the `-c 2` option up to half of your CPU cores in order to increase the speed of the upload.

   > When running the upload images script, you will be prompted for the administrator password previously specified in "Configuring and Running CDF" on page 70.

   ```
   cd ${K8S_HOME}/scripts
   ```

   ```
   ./uploadimages.sh -c 2 -F {unzipped-cdf-dir}/arcsight/images/fusion-x.x.x.x.tar -F {unzipped-cdf-dir}/arcsight/images/recon-x.x.x.x.tar
   ```

4. The pre-deployment validation process verifies all environment prerequisites have been met before deploying.
5. To verify completion of the upload of all images, return to the CDF Management Portal’s Check Availability page, and click **Check Image Availability Again**. When the *All images are available in the registry.* message displays, all required component uploads are complete.

   Check Image Availability

   All images are available in the registry.

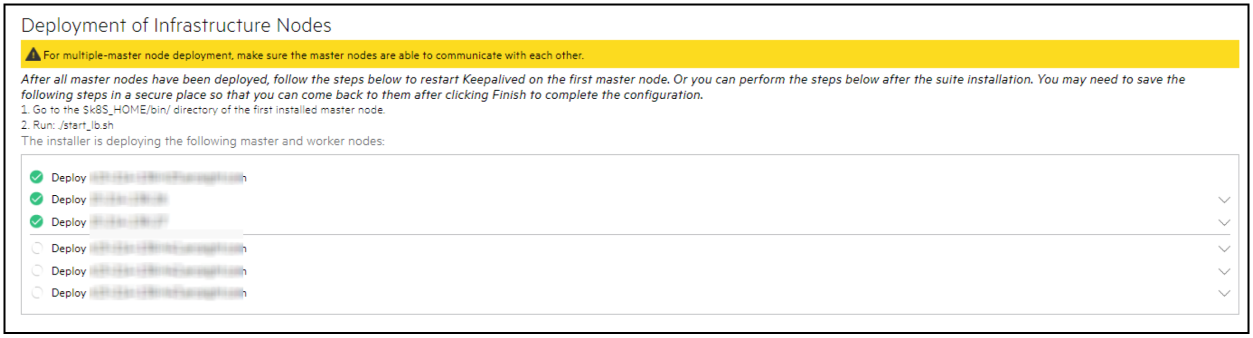
   Finalize the infrastructure installation and initialize the configuration of suite capabilities.

6. After verification, click **Next**.

Deploying Node Infrastructure and Services

Node Infrastructure

After the images are verified and you click **Next**, the node infrastructure is deployed. The **Deployment of Infrastructure Nodes** page will display progress.



Deployment of Infrastructure Nodes

For multiple-master node deployment, make sure the master nodes are able to communicate with each other.

After all master nodes have been deployed, follow the steps below to restart Keepalived on the first master node. Or you can perform the steps below after the suite installation. You may need to save the following steps in a secure place so that you can come back to them after clicking Finish to complete the configuration.

1. Go to the $k8S_HOME/bin/ directory of the first installed master node.
2. Run: ./start_lb.sh

The installer is deploying the following master and worker nodes:

✓ Deploy [redacted]
✓ Deploy [redacted]
✓ Deploy [redacted]
○ Deploy [redacted]
○ Deploy [redacted]
○ Deploy [redacted]

Please be patient. Wait for all master and worker nodes to be properly deployed (showing a green check icon). Depending on the speed of your network and node servers, this can take up to 15 minutes to complete.

> Clicking the **Retry** button will trigger additional communication with a problematic node, until the button converts to a spinning progress wheel. This indicates that the node deployment process is being started again. Until this occurs, refrain from clicking **Retry** again.

Monitoring Progress: You can monitor deployment progress on a node in the following ways:

- During installation, check the log on the node of interest, in `/tmp/install<timestamp>.log.` Run the command:

  ```
  tail -f <logfilename>
  ```

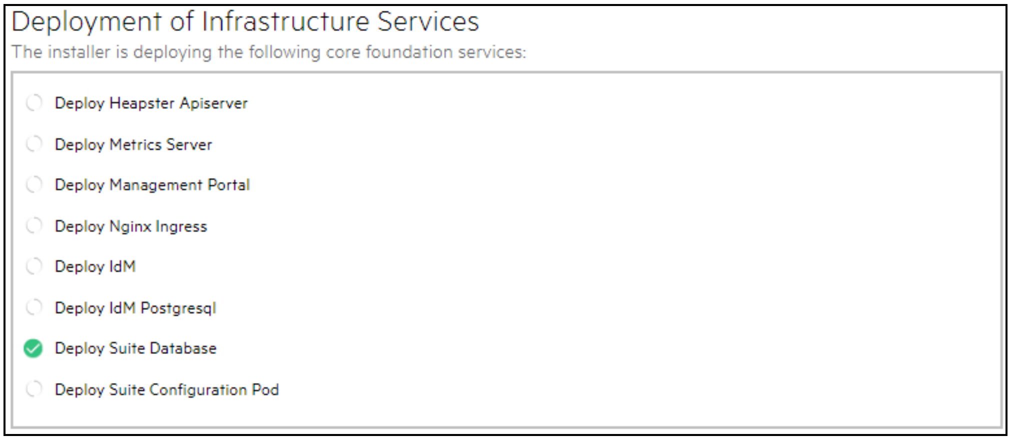
  - After installation has finished, the logs are copied to `${K8S_HOME}/log/scripts/install`
- You can watch the status of deployment pods with the command:

  ```
  kubectl get pods --namespace core -o wide | grep -i cdf-add-node
  ```

> The Initial Master Node is not reflected by its own `cdf-add-node` pod.

Infrastructure Services

Infrastructure services are then deployed. The **Deployment of Infrastructure Services** page shows the deployment progress.



Deployment of Infrastructure Services

The installer is deploying the following core foundation services:

○ Deploy Heapster Apiserver

○ Deploy Metrics Server

○ Deploy Management Portal

○ Deploy Nginx Ingress

○ Deploy IdM

○ Deploy IdM Postgresql

✓ Deploy Suite Database

○ Deploy Suite Configuration Pod

Please be patient. Wait for all services to be properly deployed (showing a green check icon). This can take up to 15 minutes to complete. Should any service show a red icon, then this process might have timed out. If this occurs, click the **Retry** icon to retry the deployment for that service.

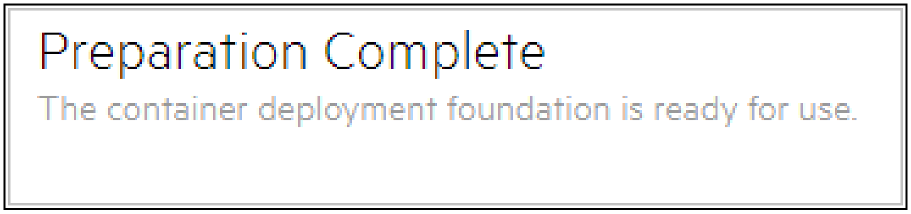
To monitor progress as pods are being deployed, on the Initial Master Node, run the command:

```
  watch 'kubectl get pods --all-namespaces'
```

> If you try to access the CDF Management Portal Web UI (port 3000) too quickly after this part of the install has finished, you might receive a ‘Bad Gateway’ error. Allow more time for the web UI to start (3 to 5 minutes) before retrying your login attempt.

After all services show a green check mark, click **Next**.

Once all nodes have been configured, and all services have been started on all nodes, the **Preparation Complete** page will be shown. You are ready to configure product-specific configuration properties.



Preparation Complete

The container deployment foundation is ready for use.

Click **Next** to configure the products and components of the deployment. You can now deploy ArcSight products.

Configuring the Deployed Capabilities

> Refer to *ArcSight Platform 20.11 Technical Requirements* "Hardware and Tuning Guidelines section" for your workload. It might specify additional settings beyond what is described below.

The deployed capabilities are ready to be configured and then deployed. The *Pre-Deployment Configuration* page displays to configure the products and capabilities chosen at the start of the installation process. This section explains the process of configuring deployed capabilities on a supported platform for both on-premises and cloud deployments.

Reviewing Settings That Must Be Set During Deployment

This section describes configuration settings that must be set during deployment. Additional settings can be modified after deployment by going to the CDF Management Portal.

> For more information, hover over the tooltips and set the values accordingly.

- "ArcSight Database" below
- "Transformation Hub" on the next page
- "Fusion" on the next page
- "Intelligence" on the next page

ArcSight Database

If you deployed the ArcSight database:

In the Transformation Hub tab, ensure the **# of CEF-to-Avro Stream Processor instances to start** is set to at least 1 or what is specified in *ArcSight Platform 20.11 Technical Requirements* for your workload.

In the Fusion tab, ensure you set these configuration settings for your environment:

- **Enable Database**
- **Use SSL for Database Connections**
- **Database Host**
- **Database Application Admin User Name**
- **Database Application Admin User Password**
- **Search User Name**
- **Search User Password**
- **Database Certificate(s)**

Transformation Hub

If you deployed the Transformation Hub:

In the Transformation Hub tab, ensure the following are set to the number of Kafka worker nodes designed into your deployment or what is specified in *ArcSight Platform 20.11 Technical Requirements* for your workload.

- **# of Kafka broker nodes in the Kafka cluster (th-kafka-count)**
- **# of ZooKeeper nodes in the ZooKeeper cluster (th-zookeeper-count)**
- **# of replicas assigned to each Kafka Topic (th-init-topicReplicationFactor)** (this setting must be set to 1 for a Single Worker deployment, and 2 for a 3-node environment)

In the Transformation Hub tab, configure the following security settings based on how you planned to secure communications as described in the Securing Communication Among Micro Focus Components section.

> FIPS and Client-Authentication are available during installation only.

- **Allow plain text (non-TLS) connections to Kafka (th-kafka-allow-plaintext)**
- **Connections use FIPS encryption (th-init-fips)**
- **Connection to Kafka uses TLS Client Authentication (th-init-client-auth)**

Fusion

If you deployed Fusion:

In the Fusion tab:

- If you have not deployed the database, set **Search Engine Replicas(search-engine-replica)** to 0, which disables the Search Engine so that it doesn't attempt to access a non-existent database. When the database is deployed, enable the Search Engine by setting this to 1.
- Modify the **Client ID** and **Client Secret** to a unique value for your environment.

Intelligence

If you deployed Intelligence:

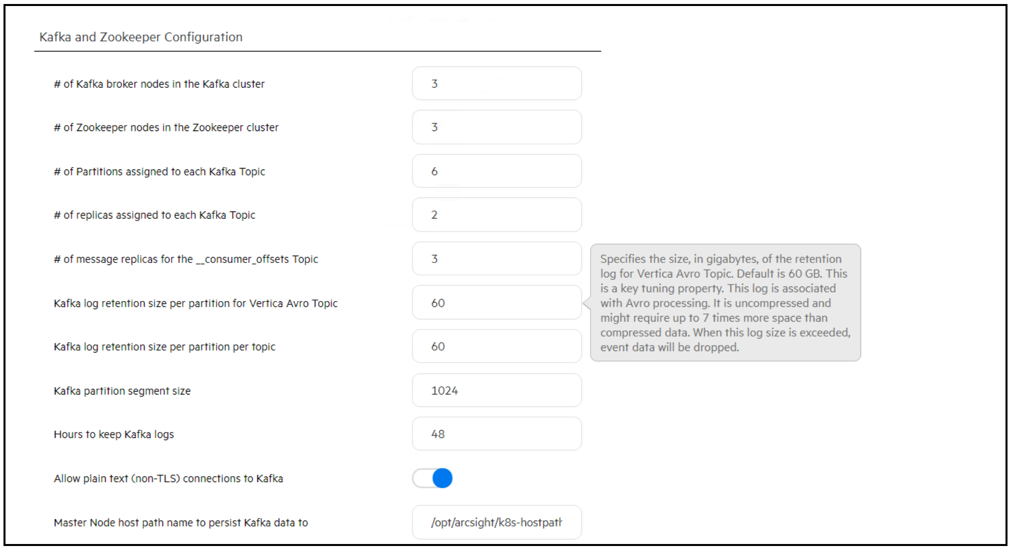
In the Intelligence tab, ensure you set these configuration settings for your environment:

- **Intelligence System Admin Email ID (interset-root-user)**
- **Number of Database Nodes (interset-vertica-number-of-nodes)**

> Be sure to change the passwords to a unique value for your environment.

- **HDFS NameNode (interset-hdfs-namenode)**
- **H2 Password (interset-h2-password)**
- **Elasticsearch Password (interset-elasticsearch-password)**
- **Analytics KeyStore Password (interset-analytics-keystore-password)**
- **Investigator KeyStore Password (interset-api-keystore-password)**
- **SearchManager KeyStore Password (searchmanager-api-keystore-password)**
- **Logstash KeyStore Password (interset-logstash-keystore-password)**
- **H2 KeyStore Password (interset-h2-keystore-password)**

> If the topic name specified for the Avro Event Topic field is not the default topic, then use Transformation Hub's Avro routing rules using ArcMC 2.96 or later to filter Avro events from the default topic. Create a routing rule with the source topic as th-arcsight-avro and destination topic as the topic name you have provided in the Avro Event Topic field. For more information, refer to the routing section in the ArcMC Administration Guide.
>
> For Intelligence **System Admin Email ID**, if Fusion is already a part of the cluster, ensure you specify the email ID of an existing System Admin user in Security, Risk & Governance. If you are deploying Fusion now, specify in the **System Admin Email ID** setting the email ID you intent to use. This user will be the default System Admin user of Intelligence.
>
> For **Analytics Configuration-Spark**, set the values based on the data load. For information about the values for Spark, see *ArcSight Platform 20.11 Technical Requirements* "Hardware and Tuning Guidelines section" for your workload.



Kafka and Zookeeper Configuration

# of Kafka broker nodes in the Kafka cluster	3
# of Zookeeper nodes in the Zookeeper cluster	3
# of Partitions assigned to each Kafka Topic	6
# of replicas assigned to each Kafka Topic	2
# of message replicas for the __consumer_offsets Topic	3
Kafka log retention size per partition for Vertica Avro Topic	60
Kafka log retention size per partition per topic	60
Kafka partition segment size	1024
Hours to keep Kafka logs	48
Allow plain text (non-TLS) connections to Kafka	●
Master Node host path name to persist Kafka data to	/opt/arcsight/k8s-hostpatł

Specifies the size, in gigabytes, of the retention log for Vertica Avro Topic. Default is 60 GB. This is a key tuning property. This log is associated with Avro processing. It is uncompressed and might require up to 7 times more space than compressed data. When this log size is exceeded, event data will be dropped.

Tuning Your Deployment

This topic contains information on:

- "Updating Event Topic Partition Number" below
- "Updating the CDF Hard Eviction Policy" on the next page

Updating Event Topic Partition Number

Refer to *ArcSight Platform 20.11 Technical Requirements* "Hardware and Tuning Guidelines section" to determine an appropriate event topic partition number for your workload.

The following steps are needed only when deploying Recon or Intelligence.

To update the topic partition number from the master node1, run the following commands:

1. Find the server ($ZK), running th-zookeeper-0:

   ```
   ZK=`kubectl get pods --all-namespaces -o wide|grep zookeeper-0|awk '{print $8}'`
   ```

2. Find NAMESPACE ($NS), for th-kafka-0:

   ```
   NS=`kubectl get pods --all-namespaces|grep kafka-0|awk '{print $1}'`
   ```

3. Update th-arcsight-avro topic partition number:

   ```
   kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --zookeeper
   $ZK:32181 --alter --topic th-arcsight-avro --partitions $number
   ```

   `$number` is the number used to calculate the partition size.

4. Update th-cef topic partition number:

   ```
   kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --zookeeper
   $ZK:32181 --alter --topic th-cef --partitions $number
   ```

5. Use the kafka manager to verify the partition number of th-cef topic and th-arcsight-avro topic have been updated to `$number.`

Updating the CDF Hard Eviction Policy

You need to update the Kubernetes hard eviction policy from 15% (default) to 100 GB to maximize disk usage.

To update the CDF Hard Eviction Policy, perform the following steps on each worker node, after deployment has been successfully completed.

> Please verify the operation is successfully executed on one work node first, then proceed on the next worker node.

> eviction-hard can either be defined as a percentage or a specific amount. The percentage or the specific amount will be determined by the volume storage.

To update the policy:

1. Run:

   ```
   cp /usr/lib/systemd/system/kubelet.service
   ```

   ```
   /usr/lib/systemd/system/kubelet.service.orig
   ```

   ```
   vim /usr/lib/systemd/system/kubelet.service
   ```

2. Behind the line:

   ```
   ExecStart=/usr/bin/kubelet \
   ```

3. Add line:

   ```
   --eviction-hard=memory.available<100Mi,nodefs.available<100Gi,imagefs.available<2Gi
   \
   ```

4. Run:

   ```
   systemctl daemon-reload and systemctl restart kubelet
   ```

5. To verify, run:

   ```
   systemctl status kubelet
   ```

   No error should be reported.

Labeling On-premises Worker Nodes

Labeling is a means for identifying application processing and qualifying the application as a candidate to run on a specific node. For example, labeling a node with the label `kafka:yes` indicates that a Kafka instance will run on that node. The labels tell Kubernetes the types of workloads that can run on a specific host system.

Immediately following deployment of your chosen capabilities, many of their associated pods will remain in a *Pending* state until you complete the labeling process. For example, the following Transformation Hub pods will be pending: `th-kafka`, `th-zookeeper`, `th-kafka-manager`, `th-web-service`, and `th-schemaregistry`.

When you finish labeling the nodes, Kubernetes immediately schedules and starts the label-dependent containers on the labeled nodes. The starting of services might take 15 minutes or more to complete.

> ⚠ Also, if you deploy a product using the YAML ARST installer method, NONE of the labels used by the deployed capability display in the UI. This is normal behavior and users should be aware of that. Only labels you create via the UI for manual application deployment are visible in the UI.

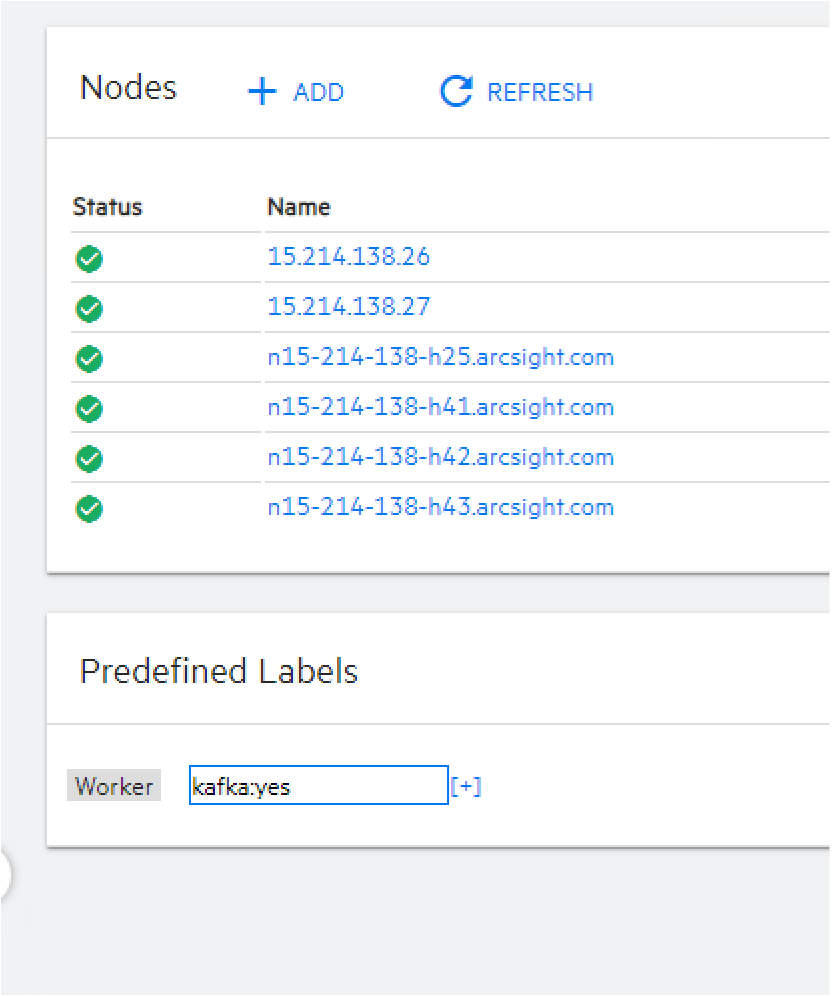
Label the Worker Nodes

You must first define the labels that you want to use, then assign them to each node. Labels are case-sensitive and must include the `:yes` text. To learn which labels apply to your deployed capabilities, see "Understanding Labels and Pods" on page 584.

> The `master:yes` and `Worker` labels are already predefined, and already applied to your nodes based on your installation. You will not need to take any action regarding these labels.

1. Log in to CDF Management Portal by clicking the link on the **Deployment status** (Configuration complete) page or browsing to (`https://<ha-address>:5443`), where:
   - *Ha-address:* represents the FQDN corresponding to the Virtual IP address provided during installation (`--ha-virtual-ip`) (or, for a single-master installation, the IP address of the master node).
   - *User Name:* admin
   - *Password:* Password that you created the first time that you logged in to the Management Portal.
2. Go to **CLUSTER** > **Nodes**.
3. In the text box for **Predefined Labels**, specify the label to add, and then click the **+** icon.

   For example, for Transformation Hub, create the `zk:yes` label. As you create the labels, the CDF Management Portal adds them to the **Predefined Labels** list to the left of the text box.
4. Repeat **Step 3** for each of the labels that you want to add to the list of predefined labels. Enter the text of the entire label, as shown here, including the `:yes` text.

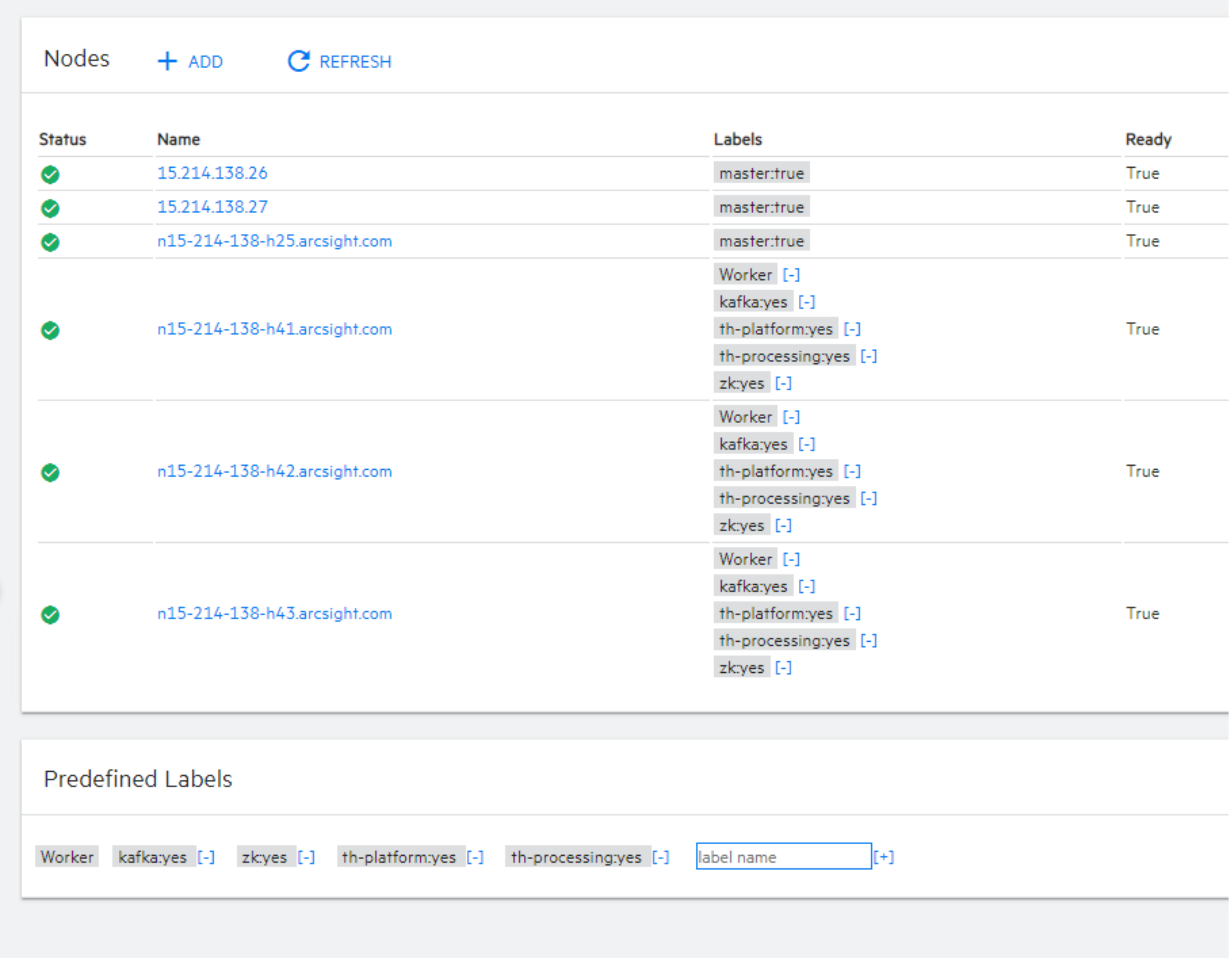


Status	Name
✓	15.214.138.26
✓	15.214.138.27
✓	n15-214-138-h25.arcsight.com
✓	n15-214-138-h41.arcsight.com
✓	n15-214-138-h42.arcsight.com
✓	n15-214-138-h43.arcsight.com

Predefined Labels

Worker kafka:yes [+]

5. Drag and drop each of the labels that you created to their corresponding worker nodes, based on your workload-sharing configuration. The corresponding components get deployed on the labeled worker nodes.



Nodes + ADD ⟳ REFRESH

Status	Name	Labels	Ready
✓	15.214.138.26	master:true	True
✓	15.214.138.27	master:true	True
✓	n15-214-138-h25.arcsight.com	master:true	True
✓	n15-214-138-h41.arcsight.com	Worker [-] kafka:yes [-] th-platform:yes [-] th-processing:yes [-] zk:yes [-]	True
✓	n15-214-138-h42.arcsight.com	Worker [-] kafka:yes [-] th-platform:yes [-] th-processing:yes [-] zk:yes [-]	True
✓	n15-214-138-h43.arcsight.com	Worker [-] kafka:yes [-] th-platform:yes [-] th-processing:yes [-] zk:yes [-]	True

Predefined Labels

Worker kafka:yes [-] zk:yes [-] th-platform:yes [-] th-processing:yes [-] label name [+]

6. Click **Refresh** to see the labels that you have applied to the nodes.

   After the nodes have been properly labeled, the status of the CDF pods in the **Configuration Complete** page displays as *Running* state.

7. To monitor the pod start up process, continue to "Checking the Deployment Status" on page 292.

8. (Conditional) To scale out the cluster, add more worker nodes to it.

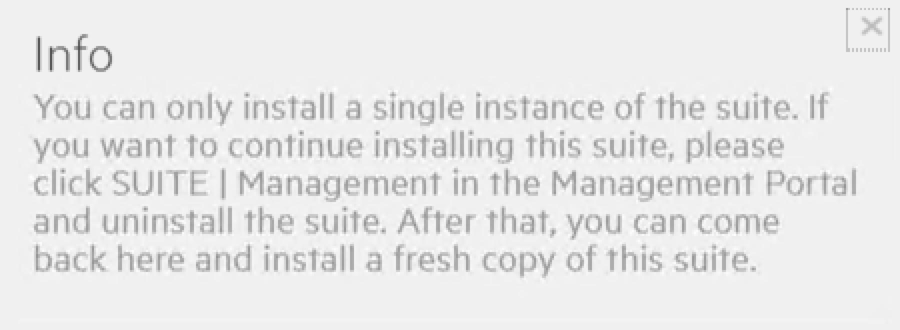
Checking the Deployment Status

When the **Configuration Complete** page displays, the pod deployment is finished.

- Pods that have not been labeled will remain in the *Pending* state until labeled.
- For a pod that is not in the *Running* state, you can find out more details on the pod by running the following command:

```
kubectl describe pod <pod name> -n <namespace>
```

The `Events` section in the output provides detailed information on the pod status.

> If the following error is displayed when attempting to log in to the CDF Management Portal on port 3000, this typically means that the CDF installation process has completed, port 3000 is no longer required, and has been closed. Instead of port 3000, log in to the Management Portal on port 5443.
>
> 
>
> Info
>
> You can only install a single instance of the suite. If you want to continue installing this suite, please click SUITE | Management in the Management Portal and uninstall the suite. After that, you can come back here and install a fresh copy of this suite.

Checking Cluster Status

To verify the success of the deployment, check the cluster status and make sure all pods are running.

> You might need to wait 10 minutes or more for all pods to be in a *Running* or *Completed* state.

To check cluster status:

1. Connect to the cluster by doing one of the following:
   - For an on-premises installation, log in to the initial master node.
   - For Azure, connect to the jump host.
   - For AWS, connect to the bastion.
2. Run the command:

   ```
   # kubectl get pods --all-namespaces
   ```

3. Review the output to determine the status of all pods.

Completing Database Setup

1. Log in to the database node1 as root:

   ```
   cd /opt/arcsight-database
   ```

2. Configure the Kafka scheduler for SSL.
3. Configure the schema registry server setting.

> ⚠ The cdf-updateRE command must be run on a master node, not on the DB.

```
/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh > issue_ca.crt
```

Use the file "issue_ca.crt" in the following command:

```
./schema_registry_setup <SCHEMA-REGISTRY-NODE-FQDN> issue_ca.crt
```

4. Create the Kafka scheduler using the following command. Specify one or more Transformation Hub nodes in a comma separated list. For high availability, we recommend specifying at least three nodes.

   If Kafka scheduler SSL was disabled in previous step, use port 9092:

   ```
   ./kafka_scheduler create <Transformation_Hub_Node_1_IP>:9092,<Transformation_Hub_
   Node_2_IP>:9092,<Transformation_Hub_Node_3_IP>:9092
   ```

   If Kafka scheduler SSL was enabled in previous step, use port 9093:

   ```
   ./kafka_scheduler create <Transformation_Hub_Node_1_IP>:9093,<Transformation_Hub_
   Node_2_IP>:9093,<Transformation_Hub_Node_3_IP>:9093
   ```

5. For a list of options you can specify when installing the scheduler, click here.
6. Check the Database status:

   ```
   ./db_installer status
   ```

7. Check the scheduler status, event-copy progress, and messages:

   ```
   ./kafka_scheduler status
   ./kafka_scheduler events
   ./kafka_scheduler messages
   ```

Creating a Cloud Deployment for Transformation Hub

This section discusses the process of preparing for and creating a cloud deployment.

Setting Up Your Deployment Architecture (Azure)

This section explains how to set up your deployment architecture for a Transformation Hub that runs on the Microsoft Azure cloud platform.

> For this release, only producers (SmartConnectors and Collectors) and consumers (Logger, ESM) and ArcMCs which are installed in the Azure cloud are supported with an Azure Transformation Hub.

Deployment Process Overview

The complete process of deploying Transformation Hub on Azure includes the following broad steps:

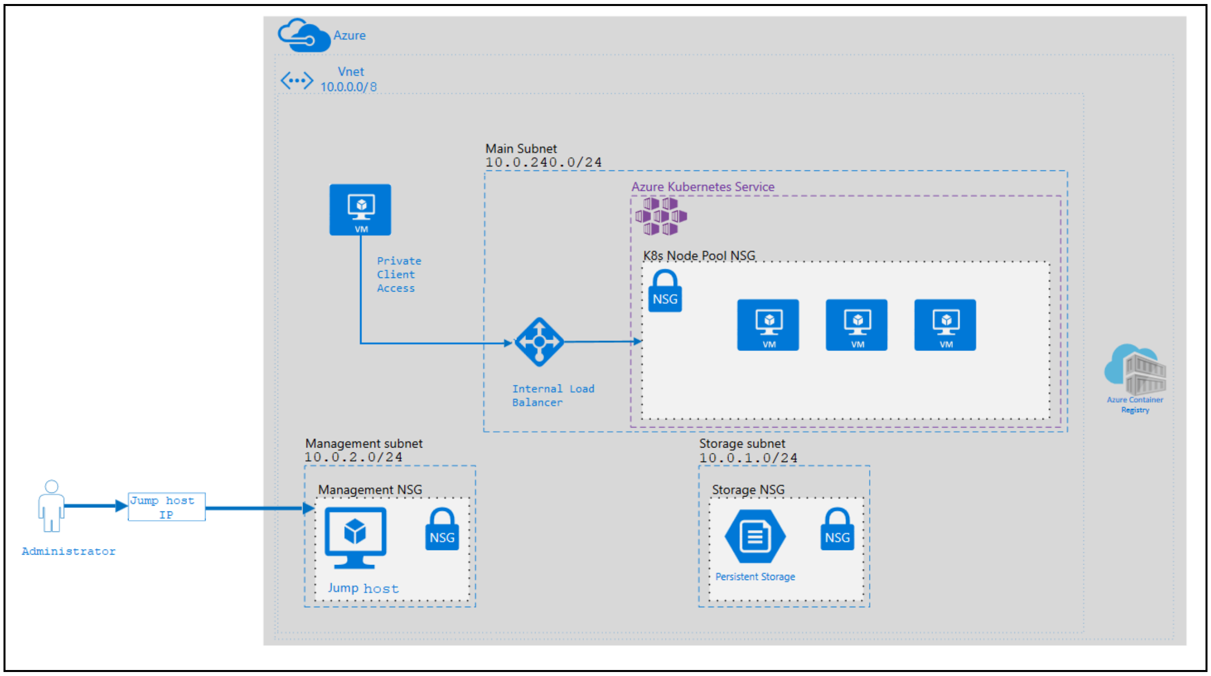
1. **Review Prerequisites:** Review the technical prerequisites and ensure that they are met before beginning the installation.
2. **Prepare the Azure Container Registry and Resource Group:** Create the Azure Container Registry (ACR) and the Azure resource group which will contain the deplpyment resources.
3. **Prepare the NFS Subnet:** The subnet is required for the NFS (Network File System) server and jump host.
4. **Prepare the Jump Host VM and Configure the Jump Host**: Create the jump host virtual machine and configure the jump host for co nnectivity to the cluster.
5. **Prepare the NFS Server:** Prepare the Network File System server. You have the choice of using a virtual machine for NFS, or, alternatively, you can use the Azure native NetApp service for NFS.
6. **Label Nodes:** Nodes in your cluster must be labeled to indicate their functionality.
7. **Upload Images:** Product images must be uploaded to the Azure Container Registry for installation.
8. **Install CDF:** The CDF installer script is run and CDF installed to Azure.

9. **Configure and Deploy the Kubernetes Cluster:** The Kubernetes cluster of AKS nodes is configured and deployed.
10. **Patch and Configure the Load Balancer:** For optimal operation, your load balancing capability must be patched with the latest updates.
11. **Configure and Deploy Transformation Hub:** Using the CDF Management Portal, configure and deploy Transformation Hub to run in the CDF-managed Kubernetes cluster.
12. **Manage Transformation Hub from the Management Center:** Configure the Management Center (ArcMC) to recognize and manage the Transformation Hub cluster.
13. **Integrate Transformation Hub with Other ArcSight Products:** Configure your SmartConnectors and Collectors as producers of events into Transformation Hub, as well as configure event Consumers such as Logger and ESM.

Each of these steps is explained in detail in the following sections. Most steps can be performed using either the Azure Portal or through the Azure Cloud Shell, and each method is explained (where possible).

Deployment Architecture

As a containerized application, Transformation Hub is deployed in the Azure environment created by CDF. The following diagram shows the completed Transformation Hub deployment infrastructure in the Azure environment.



As shown, the Transformation Hub Kubernetes nodes run as virtual machines in the Kubernetes node pool network security group (NSG) and under the Azure Kubernetes service. Secure administrator access to the nodes is from the jump host, which is included in the management NSG.

Azure Transformation Hub Deployment Prerequisites

In order to perform the installation of deploy Transformation Hub on Azure, the user requires an active Azure subscription, as well as the following:

- Permissions to create resource groups, an Azure Container Registry (ACR), and a service principal.
- OWNER rights on the created resource group.
- Permission to create Azure VMs and storage disks.
- *If using a NetApp NFS (network file system):* The Azure subscription needs to be granted access to the Azure NetApp Files service (details are described in the procedure).

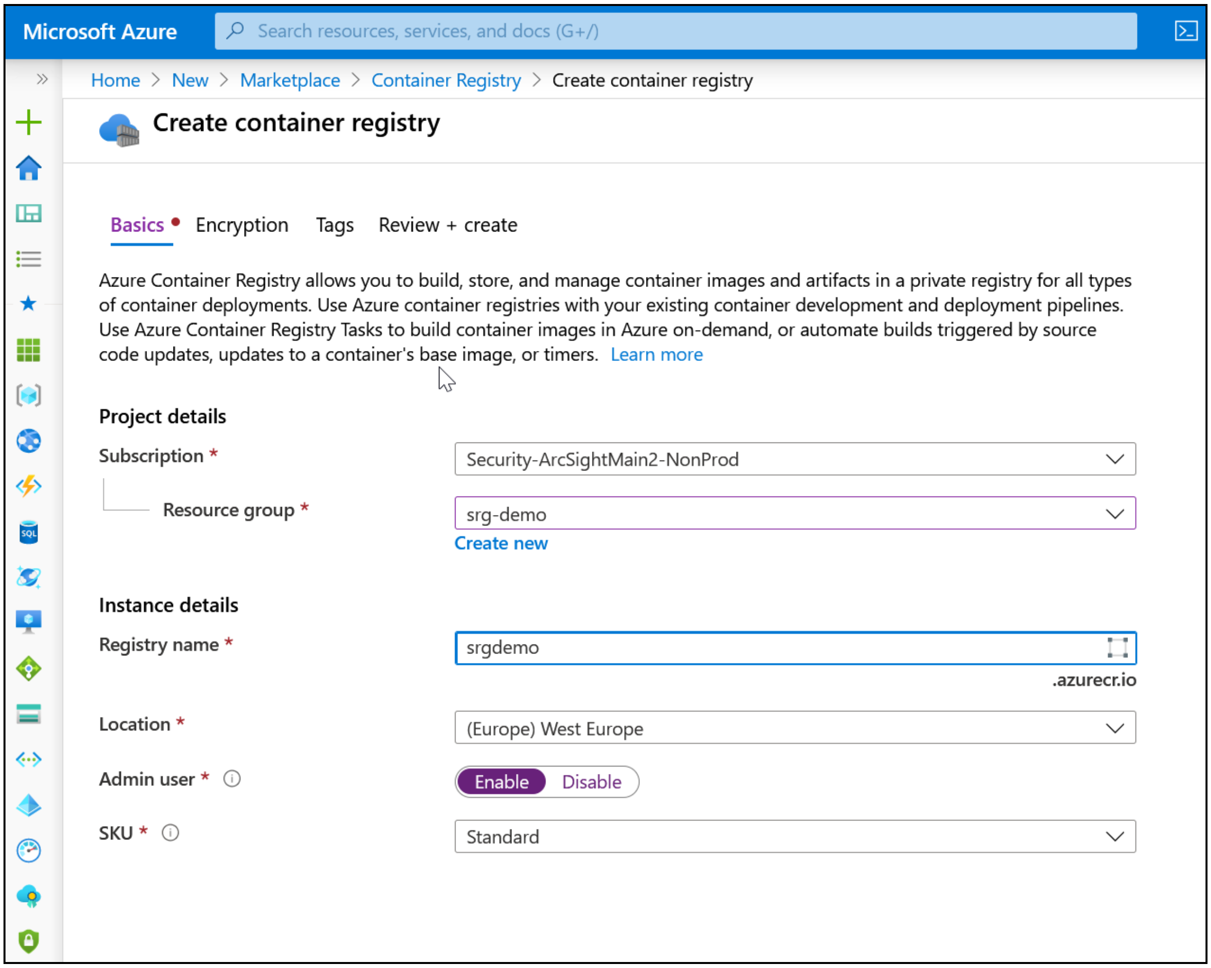
> Once you have installed and configured an Azure jump host, you can run all Azure Cloud Shell (`az cli`) commands from the jump host instead of the Azure Cloud Shell.

Next Step: Prepare the ACR and Resource Group Portal

Preparing the ACR and Resource Group

To prepare the ACR (Azure Container Registry) and resource group on the Azure Portal:

1. Log in to the Azure portal at (`https://portal.azure.com`)
2. Select an active Azure subscription and click **Create a Resource.**
3. In the search box (case-insensitive), enter *Container Registry* and click **create**. A screen similar to this is displayed.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > New > Marketplace > Container Registry > Create container registry

Create container registry

Basics • Encryption Tags Review + create

Azure Container Registry allows you to build, store, and manage container images and artifacts in a private registry for all types of container deployments. Use Azure container registries with your existing container development and deployment pipelines. Use Azure Container Registry Tasks to build container images in Azure on-demand, or automate builds triggered by source code updates, updates to a container's base image, or timers. Learn more

Project details

Subscription * Security-ArcSightMain2-NonProd

Resource group * srg-demo

Create new

Instance details

Registry name * srgdemo .azurecr.io

Location * (Europe) West Europe

Admin user * Enable Disable

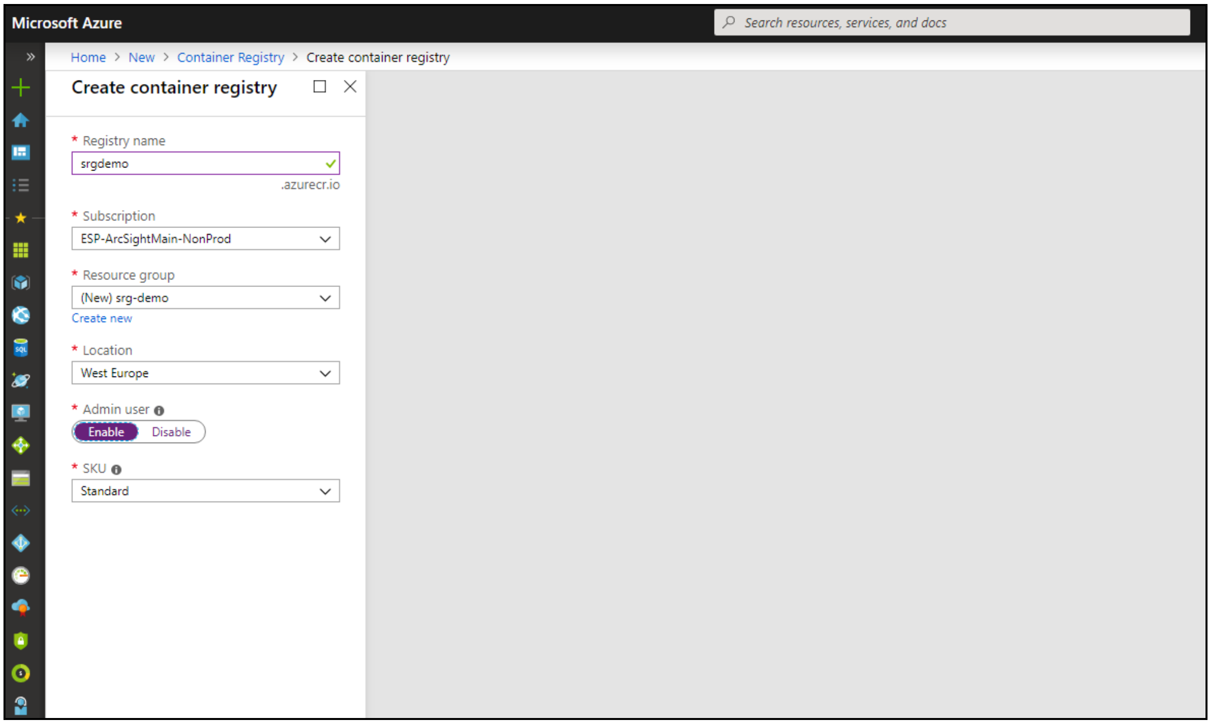
SKU * Standard

> Later in this guide, the steps above will be referred to as "Create a resource of type <some resource>." For these references, take the steps shown above to create the resource of the specified type.

4. Enter a value for **Registry Name**. Note this name for later reference.
5. For **Resource group**, click **Create New**, and in **Name**, enter a resource group name.
6. For **Location**, select a location with enough resources for your deployment.

> All other resources will need to use this location.

7. For **Admin user**, select *Enable*.
8. For **SKU**, choose the value you need (basic, medium, or standard).


Microsoft Azure
Search resources, services, and docs
Home > New > Container Registry > Create container registry
Create container registry
Create new
Enable
Disable

* Registry name: srgdemo .azurecr.io

* Subscription: ESP-ArcSightMain-NonProd

* Resource group: (New) srg-demo

* Location: West Europe

* Admin user: Enable

* SKU: Standard

9. Click **Create** to create the Azure Container Registry.

To prepare the ACR (Azure Container Registry) and resource group using the Azure Cloud Shell:

Required permissions:create an ACR, create a resource group

1. Open the Azure Cloud Shell (on the top right of the Azure Portal page). If necessary, confirm the creation of user storage.
2. Create the resource group by running the command:

   ```
   # az group create --name &lt;RESOURCE GROUP&gt; \--location
   &lt;LOCATION&gt;
   ```

   Parameters:

   `RESOURCE GROUP` is your group name, which will be used later for all other resources

   `LOCATION` is the location where resource group will be created. To get a list of all locations, run the command:

   ```
   # az account list-locations | jq ".[] | .name"
   ```

   For example: `# az group create |--name srg-demo --location westeurope`

3. Check the az command response. It should contain the text:

   `"provisioningState": "Succeeded"`

4. Create the Azure Container Registry (ACR) by running the command:

```
# az acr create -n &lt;your ACR name&gt; -g &lt;your resource group
name&gt; --admin-enabled "true" --sku "Standard"
```

> For example: `# az acr create \-n srgdemoACR \-g srg-demo \--admin-enabled "true" \--sku "Standard"`

5. Check the az command response. It should contain the text:

   `"provisioningState": "Succeeded"`

> In succeeding procedures, the az command response should contain the same text: `"provisioningState": "Succeeded"`

Next Step: "Preparing the Azure Kubernetes Service (AKS)" on the next page

Preparing the ACR and Resource Group Using the Azure Cloud Shell

Required permissions: create an ACR, create a resource group

1. Open the Azure Cloud Shell (on the top right of the Azure Portal page). If necessary, confirm the creation of user storage.
2. Create the resource group by running the command:

```
# az group create --name <RESOURCE GROUP> \--location <LOCATION>
```

Parameters:

`<RESOURCE GROUP>` is your group name, which will be used later for all other resources

`<LOCATION>` is the location where resource group will be created. To get a list of all locations, run the command:

```
# az account list-locations | jq ".[] | .name"
```

For example:

```
# az group create |--name srg-demo --location westeurope
```

9. Check the az command response. It should contain the text:

```
"provisioningState": "Succeeded"
```

10. Create the Azure Container Registry (ACR) by running the command:

```
# az acr create -n &lt;your ACR name&gt; -g &lt;your resource group name&gt; --
admin-enabled "true" --sku "Standard"
```

For example:

```
# az acr create \-n srgdemoACR \-g srg-demo \--admin-enabled "true" \--sku
"Standard"
```

11. Check the az command response. It should contain the text:

```
"provisioningState": "Succeeded"
```

> In succeeding procedures, the az command response should contain the same text: "provisioningState": "Succeeded"

Preparing the Azure Kubernetes Service (AKS)

Preparation of the AKS includes these sub-steps. Each is explained in the following sections.

Next Step: Creating the Service Principal ID for Kubernetes

Creating the Service Principal ID for Kubernetes

Required permissions: create service principal

To create the service principal ID:

1. In the Azure Cloud Shell, run the command:

```
# az ad sp create-for-rbac -n "PRINCIPAL ID NAME" --skip-assignment
```

For example:

```
# az ad sp create-for-rbac -n srgdemo-service-principal --skip-assignment
```

Example results:

```
{
   "appId":"52f25b66-2700-474d-a2a0-016f0b149e22",
   "displayName":"srgdemo-service-principal",
   "name":"http://srgdemo-service-principal",
   "password":"bf47aa85-9578-4d61-a8e9-ffafe5a1e22b",
   "tenant":"6002e264-31f7-43d3-a51e-9ed1ba9ca689"
}
```

Note the values for `password` and `appID`. These values will be used in the next step.

Next Step: Prepare the Virtual Network and AKS Subnet

Preparing the Virtual Network and AKS Subnet

Now you can prepare a virtual network with custom ranges and subnet for AKS. If you already have an existing virtual network with a subnet for AKS, you can skip this procedure.

All the created resources must be placed in the same virtual network to prevent performance issues caused by network latency; these resources include resource group, AKS cluster, jump host, and Azure NetApp Files (NFS).

To create the virtual network:

1. Run the following command:

```
# az network vnet create \
-g <RESOURCE_GROUP> \
-n <VNET_NAME> \
--address-prefix <VNET_CIDR> \
--subnet-name <SUBNET_NAME> \
--subnet-prefix <SUBNET_CIDR>
```

Parameters:

<RESOURCE_GROUP>: the name of the resource group created in step 1.1

<VNET_NAME>: The assigned name of this virtual network.

<VNET_CIDR>: The CIDR notation for this virtual network. For example, 10.1.0.0/16.

<SUBNET_NAME>: Name for this subnet for AKS.

<SUBNET_CIDR>: The CIDR notation for this subnet. For example, 10.1.1.0/24.

For example, this would create a virtual network demo-vnet, in resource group srg-demo, with range 10.1.0.0/16 and subnet aks-subnet with subnet range 10.1.1.0/24 :

```
# az network vnet create \
-g srg-demo \
-n demo-vnet \
--address-prefix 10.1.0.0/16 \
--subnet-name aks-subnet \
--subnet-prefix 10.1.1.0/24
```

Next Step: Create the AKS

Creating the Azure Kubernetes Service (AKS)

Required permissions: create Azure Kubernetes service; the user must be the OWNER of the resource group

To create the AKS:

1. Get the subnet ID which you want to use for AKS and store it to an environment variable:

```
# SUBNET_ID=$(az network vnet subnet show \
--resource-group <RESOURCE_GROUP> \
--vnet-name <VNET_NAME> \
--name <SUBNET_NAME> \
--query id -o tsv)
```

For example, to use the virtual network demo-vnet from the resource group srg-demo and subnet aks-subnet, you would run the following command:

```
# SUBNET_ID=$(az network vnet subnet show --resource-group srg-demo --vnet-
name demo-vnet --name aks-subnet --query id -o tsv)
```

2. Create the AKS in this subnet by running the command:

```
# az aks create \
-g <RESOURCE GROUP> \
-n <AKS NAME> \
-c <NUMBER OF NODES> \
--kubernetes-version <Kubernetes version> \
--generate-ssh-keys \
--node-vm-size <VM SIZE> \
--vm-set-type VirtualMachineScaleSet \
--service-principal "<SP APP ID>" \
--client-secret "<SP PASSWORD>" \
--load-balancer-sku basic \
--vnet-subnet-id $SUBNET_ID
```

where:

`<RESOURCE GROUP>` is your main resource group

`<AKS NAME>` is your AKS resource name

`<NUMBER OF NODES>` is the number of worker nodes

`<KUBERNETES VERSION>` is the version of Kubernetes cluster we want to create, which must be supported by your CDF version. You must be OWNER (or be OWNER of resource group) to be able to assign the virtual network to the AKS

`<VM SIZE>` for example, `Standard_D4s_v3`.

> For a production cluster, do not use a size less than `Standard_D8s_v3` with less than 32 GB of RAM.

For a list of possible VMs, run the command:

```
# az vm list-sizes -l <LOCATION> | jq ".[] | .name"
```

`<SP APP ID>` and `<SP_PASSWORD>` is the appID and password from the creation of the service principal ID.

Example command:

```
# az aks create \
-g "srg-demo" \
-n "srg-demo-aks" \
-c "3" \
--kubernetes-version 1.11.8 \
--generate-ssh-keys \
--node-vm-size "Standard_D4s_v3" \
--vm-set-type VirtualMachineScaleSets \
--service-principal "52f25b66-2700-474d-a2a0-016f0b149e22" \
--client-secret "bf47aa85-9578-4d61-a8e9-ffafe5a1e22b" \
--load-balancer-sku basic \
--vnet-subnet-id $SUBNET_ID
```

The `az aks create` command will generate private and public keys, which are stored in the `~/.ssh` directory. Download `id_rsa` to a secure network location. Later, this will be uploaded to the jump host and used to connect to AKS nodes from the jump host.

Next Step: Prepare the Subnet for the NFS Server and Jump Host

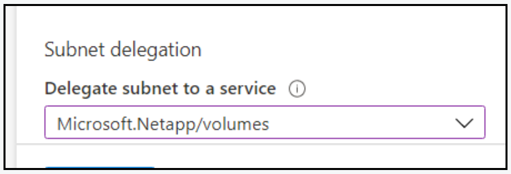
Preparing the Subnet for the NFS Server and Jump Host

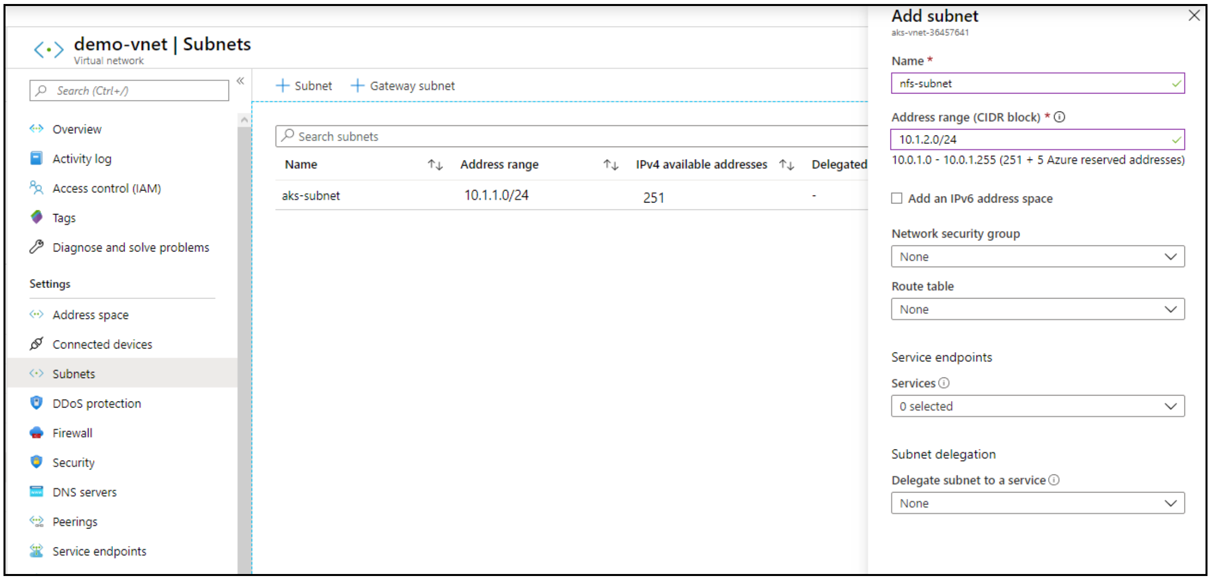
This step might be skipped if you already have a subnet prepared for the installation AKS.

In this section, you will prepare subnets for the NFS server and for the jump host. All of the created resources should be placed in the same vnet to prevent performance issues caused by network latency; such resources include resource group, AKS cluster, jumphost, Azure NetApp files, and so on.

To prepare the subnet for the NFS server and jump host:

1. Open the virtual network used for the installation AKS. In our example, this is called `demo-vnet` in resource group srg-demo.
2. On the **Virtual network** page, under **Settings**, select **Subnets**.
3. Click **+ Subnet.**
4. In **Name**, enter *nfs-subnet.*
5. In **Address range,** enter an address range based on the IP assigned by Azure. In our example, aks-subnet uses the address range `10.1.1.0/24`, so we use `10.1.2.0/24.`

> If you plan to use NetApp as an NFS service, under **Subnet delegation,** the subnet must be delegated to `Microsoft.NetApp/volumes.`
>
> 
>
> Subnet delegation
>
> Delegate subnet to a service: Microsoft.Netapp/volumes



Name	Address range	IPv4 available addresses	Delegated
aks-subnet	10.1.1.0/24	251	-

Add subnet
aks-vnet-36457641

Name *: nfs-subnet

Address range (CIDR block) *: 10.1.2.0/24
10.0.1.0 - 10.0.1.255 (251 + 5 Azure reserved addresses)

☐ Add an IPv6 address space

Network security group: None

Route table: None

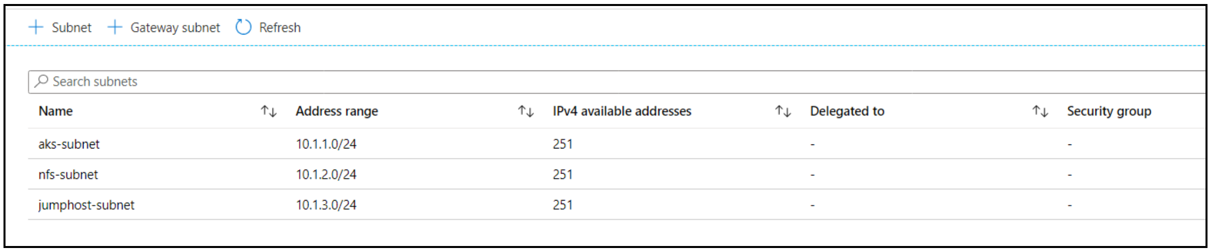
Service endpoints

Services: 0 selected

Subnet delegation

Delegate subnet to a service: None

6. Click **OK** to create the subnet.



Name	Address range	IPv4 available addresses	Delegated to	Security group
aks-subnet	10.1.1.0/24	251	-	-
nfs-subnet	10.1.2.0/24	251	-	-
jumphost-subnet	10.1.3.0/24	251	-	-

7. Repeat steps 3 and 4 to create a subnet for the jump host.
   - For name, use *jumphost-subnet.*
   - For address range we will use, for example, 10.1.3.0/24.

To prepare the subnet for the NFS server and jump host using the Azure Cloud Shell:

Required permissions:create subnets inside the AKS virtual network

1. Create the NFS subnet by running the command:

```
# az network vnet subnet create \
--address-prefixes <ADDRESS PREFIX> \
--name nfs-subnet \
-g <RESOURCE GROUP> \
--vnet-name <VIRTUAL NETWORK>
```

where:

- `<VIRTUAL NETWORK>` is the virtual network name where you want to create the subnet. We will use the virtual network created earlier.
- `<RESOURCE GROUP>` is the resource group where the virtual network is located.

> If you are using NetApp as an NFS service, add the argument `--delegations Microsoft.NetApp/volumes` to the above command.

> For example, the following command would create nfs-subnet in virtual network demo-vnet, from resource group srg-demo with range 10.1.2.0/24 : `# az network vnet subnet create --address-prefixes 10.1.2.0/24 --name nfs-subnet -g srg-demo --vnet-name demo-vnet`

2. Create the jumphost subnet by running a similar command to the one in Step 1, but with a different name and address prefix.

> For example, the following command would create the jumphost-subnet, in virtual network demo-vnet from resource group srg-demo with range 10.1.3.0/24 `# az network vnet subnet create --address-prefixes 10.1.3.0/24 --name jumphost-subnet -g srg-demo --vnet-name demo-vnet`

Next Step: "Preparing the Jump Host Virtual Machine" on the next page

Preparing the Subnet for the NFS Server and Jump Host Using the Azure Cloud Shell

Required permissions: create subnets inside the AKS virtual network

1. Create the NFS subnet by running the command:

```
# az network vnet subnet create \
--address-prefixes <ADDRESS PREFIX> \
--name nfs-subnet \
-g <RESOURCE GROUP> \
--vnet-name <VIRTUAL NETWORK>
```

where:

- `<VIRTUAL NETWORK>` is the virtual network name where you want to create the subnet. We will use the virtual network created earlier.
- `<RESOURCE GROUP>` is the resource group where the virtual network is located.

If you are using NetApp as an NFS service, add the argument `--delegations Microsoft.NetApp/volumes` to the above command.

For example, the following command would create nfs-subnet in virtual network demo-vnet, from resource group srg-demo with range 10.1.2.0/24 :

```
# az network vnet subnet create --address-prefixes 10.1.2.0/24 --name nfs-subnet -g srg-demo --vnet-name demo-vnet
```

2. Create the jumphost subnet by running a similar command to the one in Step 1, but with a different name and address prefix.

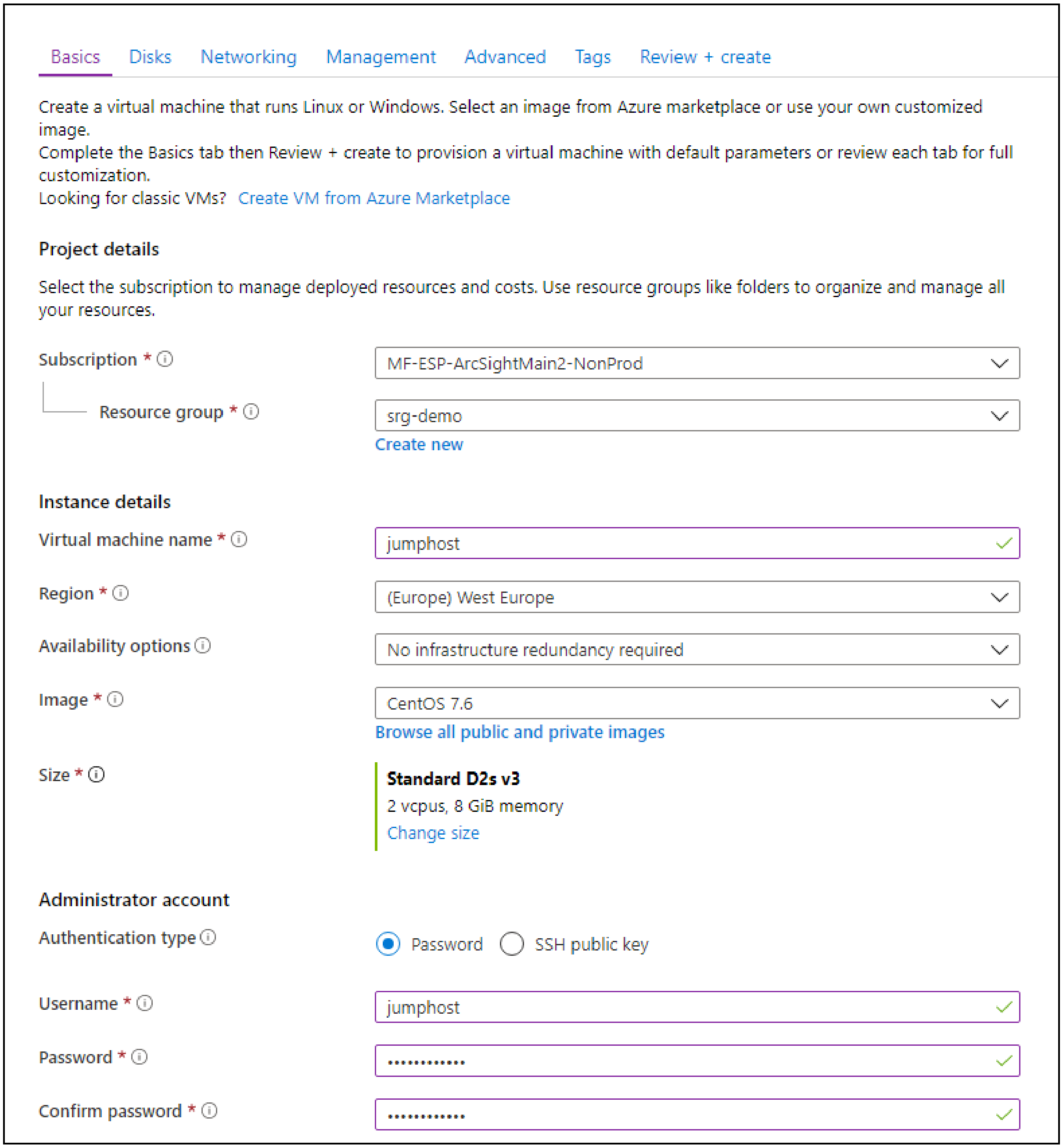
For example, the following command would create the jumphost-subnet, in virtual network demo-vnet from resource group srg-demo with range 10.1.3.0/24

```
# az network vnet subnet create --address-prefixes 10.1.3.0/24 --name jumphost-subnet -g srg-demo --vnet-name demo-vnet
```

Preparing the Jump Host Virtual Machine

To prepare the jump host VM using the Azure Portal:

1. Create resource of type *CentOS-based* and enter these values:
   - For **Resource group**, use the resource group you created for the ACR.
   - In **Virtual machine name**. assign a VM name.
   - For **Size**, leave at the default value.
   - Set the **Authentication type** to your preferences and supply the Administration account details accordingly. In our examples, we use the username/password authentication.



Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image.
Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.
Looking for classic VMs? Create VM from Azure Marketplace

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ	MF-ESP-ArcSightMain2-NonProd
Resource group * ⓘ	srg-demo Create new

Instance details

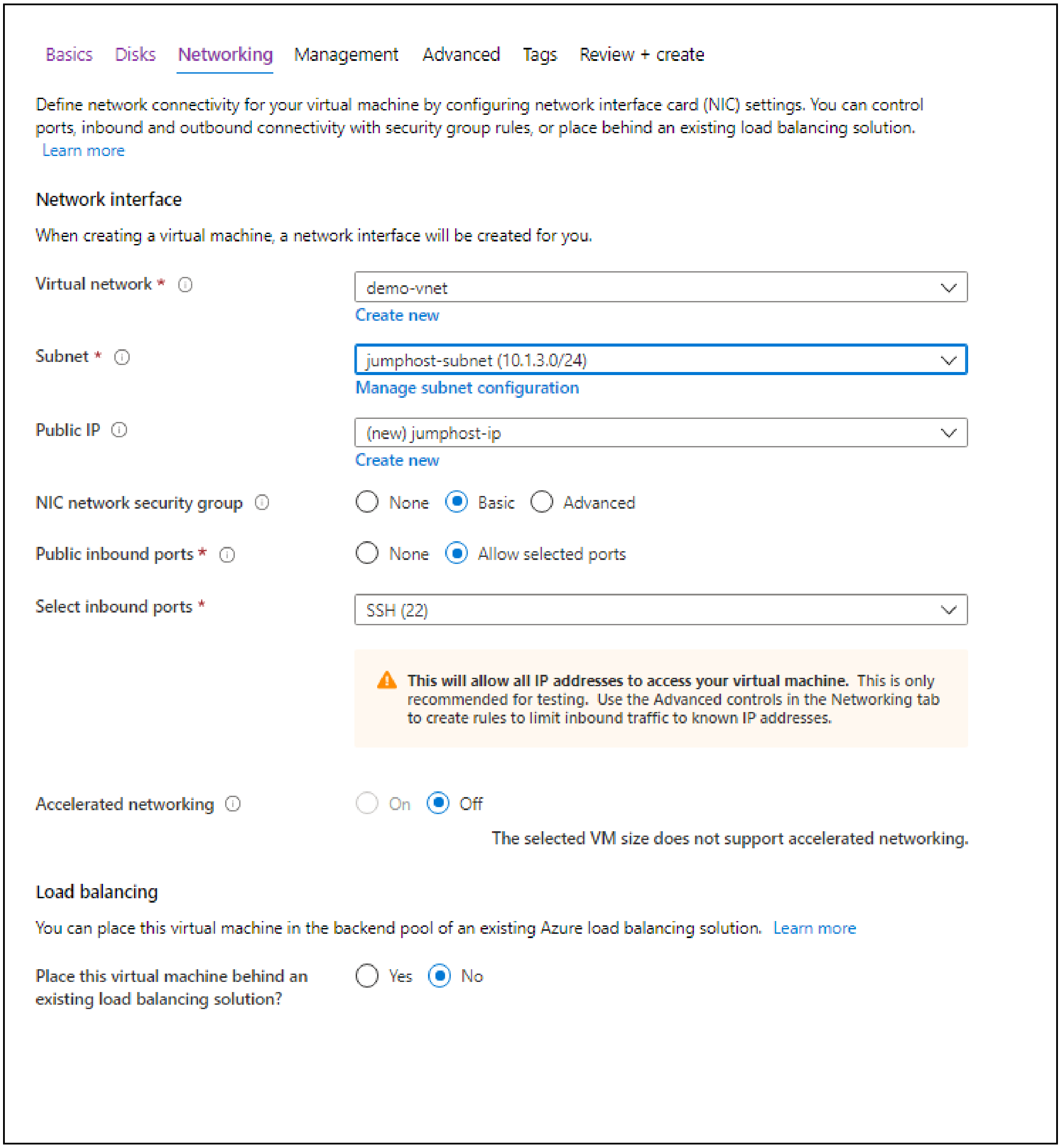
Virtual machine name * ⓘ	jumphost ✓
Region * ⓘ	(Europe) West Europe
Availability options ⓘ	No infrastructure redundancy required
Image * ⓘ	CentOS 7.6 Browse all public and private images
Size * ⓘ	**Standard D2s v3** 2 vcpus, 8 GiB memory Change size

Administrator account

Authentication type ⓘ	● Password ○ SSH public key
Username * ⓘ	jumphost ✓
Password * ⓘ	••••••••••••• ✓
Confirm password * ⓘ	••••••••••••• ✓

2. Click **Next: Disks.** No actions need to be taken on this page, so click **Next: Networking,** and then enter the following values:
   - For **Virtual network**, select the virtual network you created previously (its name has the format *demo-vnet*).
   - For **Subnet** select *jumphost-subnet.*
   - For **NIC network security group,** select *Basic* and *Allow SSH port to connect.*

For optimal security, remove this rule when the jumphost is not needed, or add more strict rules such IP filtering.



Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. Learn more

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network *	demo-vnet Create new
Subnet *	jumphost-subnet (10.1.3.0/24) Manage subnet configuration
Public IP	(new) jumphost-ip Create new
NIC network security group	○ None ● Basic ○ Advanced
Public inbound ports *	○ None ● Allow selected ports
Select inbound ports *	SSH (22)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Accelerated networking	○ On ● Off The selected VM size does not support accelerated networking.

Load balancing

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. Learn more

Place this virtual machine behind an existing load balancing solution?	○ Yes ● No

3. Click **Review + create.** Then, after validation, click **Create** to start the deployment.
4. When deployment completes, browse the to VM overview and note the jump host's Public IP address.

To prepare the jump host VM using the Azure Cloud Shell:

Required Permissions:create security groups, network interfaces, public IPs, and CentOS-based virtual machines

1. Set your main resource group name to an environment variable:

   ```
   # RESOURCE_GROUP=<your resource group name>
   ```

   For example: `# RESOURCE_GROUP=srg-demo`

2. Create a network security group for the jump host by running the following command:

   ```
   # az network nsg create \
   -g $RESOURCE_GROUP \
   -n jumphost-nsg
   ```

   For example:

   ```
   # az network nsg create \
   -g srg-demo \
   -n jumphost-nsg
   ```

3. Open the SSH port (if needed) by running this command:

   ```
   # az network nsg rule create \
   -g $RESOURCE_GROUP \
   -n ssh \
   --nsg-nam jumphost-nsg \
   --priority 1000 \
   --destination-port-ranges 22
   ```

   > Keep in mind the security risks of opening ports to the Internet and consider using a VPN, restricting access to the allowed source IP address, or limiting the amount of time that this port remains available.

4. Prepare the jump host public IP:

   ```
   # az network public-ip create \
   -n jumphost-PublicIP \
   -g $RESOURCE_GROUP \
   --allocation-method "Static" \
   --sku "Standard"
   ```

5. Get the subnet ID and store it in an environment variable for later usage.

   ```
   # SUBNET_ID=$(az network vnet list -g $RESOURCE_GROUP \
   | jq -r '.[] \
   | select(.name == "<your vnet name>") \
   ```

```
| .subnets[] | select(.name == "<your jumphost subnet>") \
| .id')
```

```
For example: # SUBNET_ID=$(az network vnet list -g $RESOURCE_GROUP \
| jq -r '.[] \
| select(.name == "demo-vnet") \
| .subnets[] \
| select(.name == "jumphost-subnet") \
| .id')
```

6. Create the network interface `jumphost-VMNic` in your resource group with public IP `jumphost-publicIP` with network security group `jumphost-nsg` by running the following command:

```
# az network nic create \
--name "jumphost-VMNic" \
--resource-group $RESOURCE_GROUP \
--public-ip-address "jumphost-PublicIP" \
--ip-forwarding  "true" \
--network-security-group "jumphost-nsg" \
--subnet $SUBNET_ID
```

7. Create the jump host VM by running the following command:

```
# az vm create \
--name "jumphost" \
--resource-group $RESOURCE_GROUP \
--image "OpenLogic:CentOS:7.7:latest" \
--size "Standard_D4s_v3" \
--public-ip-address-allocation "static" \
--nics "jumphost-VMNic" \
--admin-username jumphost  \
--admin-password myStrongPassword@!123
```

where:

- Size might be a smaller value. To get a list of supported sizes, run the command:

```
az vm list-sizes -l <LOCATION> | jq ".[] | .name"
```

- Image can be any supported CentOS. To get a list of CentOS images, run the command:

```
az vm image list -l <LOCATION> -f CentOS --all
```

Example result:

```
{
   "- Finished ..""fqdns":"",
   "id":"/subscriptions/af379ae8-90b3-4368-8fe7-
b6a55ab17720/resourceGroups/srg-
demo/providers/Microsoft.Compute/virtualMachines/jumphost",
   "location":"westeurope",
   "macAddress":"00-0D-3A-BD-08-42",
   "powerState":"VM running",
   "privateIpAddress":"10.0.2.4",
   "publicIpAddress":"51.124.17.183",
   "resourceGroup":"srg-demo",
   "zones":""
}
```

Use the VM Public IP address to permit SSH access to the jump host from outside. (The SSH port needs to be open if access is permitted from outside.)

Next Step: Configuring the Jump Host

Preparing the Jump Host VM Using the Azure Cloud Shell

Required Permissions: create security groups, network interfaces, public IPs, and CentOS-based virtual machines

1. Set your main resource group name to an environment variable:
   ```
   # RESOURCE_GROUP=<your resource group name>
   ```

For example:

```
# RESOURCE_GROUP=srg-demo
```

2. Create a network security group for the jump host by running the following command:
   ```
   # az network nsg create -g $RESOURCE_GROUP -n jumphost-nsg
   ```

For example:

```
# az network nsg create -g srg-demo -n jumphost-nsg
```

3. Open the SSH port (if needed) by running these commands:
   ```
   # az network nsg rule create -g $RESOURCE_GROUP -n ssh --nsg-nam jumphost-
   nsg --priority 1000 --destination-port-ranges 22
   ```

> Keep in mind the security risks of opening ports to the Internet and consider using a VPN, restricting access to the allowed source IP address, or limiting the amount of time that this port remains available.

4. Prepare the jump host public IP:
   ```
   # az network public-ip create -n jumphost-PublicIP -g $RESOURCE_GROUP --
   allocation-method "Static" --sku "Standard"
   ```
5. Get the subnet ID and store it in an environment variable for later usage.
   ```
   # SUBNET_ID=$(az network vnet list -g $RESOURCE_GROUP | jq -r '.[] |
   select(.name == "<your vnet name>") | .subnets[] | select(.name == "<your
   jumphost subnet>") | .id')
   ```

> For example:
> ```
> SUBNET_ID=$(az network vnet list -g $RESOURCE_GROUP | jq -r '.[] | select
> (.name == "demo-vnet") | .subnets[] | select(.name == "jumphost-subnet") |
> .id')
> ```

6. Create the network interface `jumphost-VMNic` in your resource group with public IP `jumphost-publicIP` with network security group `jumphost-nsg` by running the following command:
   ```
   # az network nic create --name "jumphost-VMNic" --resource-group
   $RESOURCE_GROUP --public-ip-address "jumphost-PublicIP" --ip-forwarding
   "true" --network-security-group "jumphost-nsg" --subnet $SUBNET_ID
   ```
7. Create the jump host VM by running the following command:
   ```
   az vm create --name "jumphost" --resource-group $RESOURCE_GROUP --image
   "OpenLogic:CentOS:7.7:latest" --size "Standard_D4s_v3" --public-ip-
   address-allocation "static" --nics "jumphost-VMNic" --admin-username
   jumphost --admin-password myStrongPassword@!123
   ```
   where:
   - Size might be a smaller value. To get a list of supported sizes, run the command:
     `az vm list-sizes -l <LOCATION> | jq ".[] | .name"`
   - Image can be any supported CentOS. To get a list of CentOS images, run the command:
     `az vm image list -l <LOCATION> -f CentOS --all`

Example result:

```
{- Finished ..

"fqdns": "",

"id": "/subscriptions/af379ae8-90b3-4368-8fe7-
b6a55ab17720/resourceGroups/srg-
demo/providers/Microsoft.Compute/virtualMachines/jumphost",

"location": "westeurope",

"macAddress": "00-0D-3A-BD-08-42",

"powerState": "VM running",

"privateIpAddress": "10.0.2.4",

"publicIpAddress": "51.124.17.183",

"resourceGroup": "srg-demo",

"zones": ""

}
```

Use the VM Public IP address to permit SSH access to the jump host from outside. (The SSH port needs to be open if access is permitted from outside.)

Configuring the Jump Host

1. Using the VM's public IP, SSH to the jump host VM and become `root`.
2. Run the following commands:

   ```
   # curl -LO https://storage.googleapis.com/kubernetes-
   release/release/$(curl -s https://storage.googleapis.com/kubernetes-
   release/release/stable.txt)/bin/linux/amd64/kubectl
   # chmod 755 kubectl
   # mv kubectl /bin
   # yum install epel-release -y
   # yum install jq -y
   ```

3. Install the Azure client for CentOS by running these commands:

   ```
   # rpm --import https://packages.microsoft.com/keys/microsoft.asc
   #sh -c 'echo -e "[azure-cli]\nname=Azure
   CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-
   cli\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/keys/mic
   rosoft.asc" > /etc/yum.repos.d/azure-cli.repo'
   # yum install azure-cli
   ```

4. Log in to your Azure account and follow the console instructions by running:

   ```
   # az login
   ```

5. Get your Kubernetes cluster credentials by running the following command:

   ```
   # az aks get-credentials --resource-group <your resource group name> --
   name <your kubernetes resource name>
   ```

For example:

```
# az aks get-credentials --resource-group srg-demo --name srg-demo-aks
```

6. Check if `kubectl` can access the cluster by running:

   ```
   # kubectl get nodes
   ```

Example output:

```
NAME                                STATUS   ROLES   AGE    VERSION
aks-agentpool-36457641-vmss000000   Ready    agent   137m   v1.13.11
aks-agentpool-36457641-vmss000001   Ready    agent   137m   v1.13.11
aks-agentpool-36457641-vmss000002   Ready    agent   137m   v1.13.11
```

Next Step: Configuring Remote Desktop Protocol

Configuring Remote Desktop Protocol (RDP) on Your Jump Host

Since RDP will be required for your jump host, the following configuration steps will be required:

- The installation of xRDP
- Installation of a preferred desktop environment (choice of XFCE, MATE, or GNOME)
- Opening of an RDP port on the jump host network security group (NSG)

> Consider the security risks of opening ports to the Internet. Use of a VPN, restricting access to the allowed source IP address, or limiting the amount of time that this port remains available can reduce these risks.

To configure RDP on your jump host:

1. Connect to the jump host and become root.
2. Install and enable xrdp. Run these commands:
   ```
   # yum install -y epel-release
   # yum install -y xrdp
   # systemctl enable xrdp
   # systemctl start xrdp
   ```
3. Check whether the firewall is running; if so, then open RDP port 3389/tcp by running these commands:
   ```
   # firewall-cmd --add-port=3389/tcp --permanent
   # firewall-cmd --reload
   ```
4. Install your preferred desktop environment (XFCE, MATE or GNOME). This example will use MATE. Run these commands:
   ```
   # yum install -y epel-release
   # yum groupinstall -y "MATE Desktop"
   ```
5. Wait for the install to complete, and then reboot the jump host.
6. Connect to the jump host and stay as a jump host user.
7. Create the Xclients file for the user, which will be used to log in. Run these commands:
   ```
   # echo "mate-session" > ~/.Xclients
   # chmod a+x ~/.Xclients
   ```
8. Do one of the following:
   1. On the Azure Portal, open the RDP port on the jump host network security group (NSG), and then proceed to Step 9, OR,
   2. Run the following command (after which, the procedure is complete)
      ```
      # az network nsg rule create -g <RESOURCE GROUP> -n rdp --nsg-nam
      jumphost-nsg --priority 1001 --destination-port-ranges 3389
      ```

   For example:
   ```
   # az network nsg rule create -g srg-demo -n rdp --nsg-nam jumphost-nsg --
   priority 1001 --destination-port-ranges 3389
   ```

> Find the jump host network security group (NSG) in your resource group. In our example, the NSG is called `jumphost-NSG`

9. In **Settings**, click **Inbound security rules.**
10. Click **Add** and then enter values for these settings:
    - In **Name**, enter a name for the NSG.
    - In **Destination port ranges** enter 3389.

> After the jump host is created and configured, all further `az cli` commands can be run from your jump host instead of using the Azure Cloud Shell.

Next Step: Prepare the NFS Server

Preparing the NFS Server

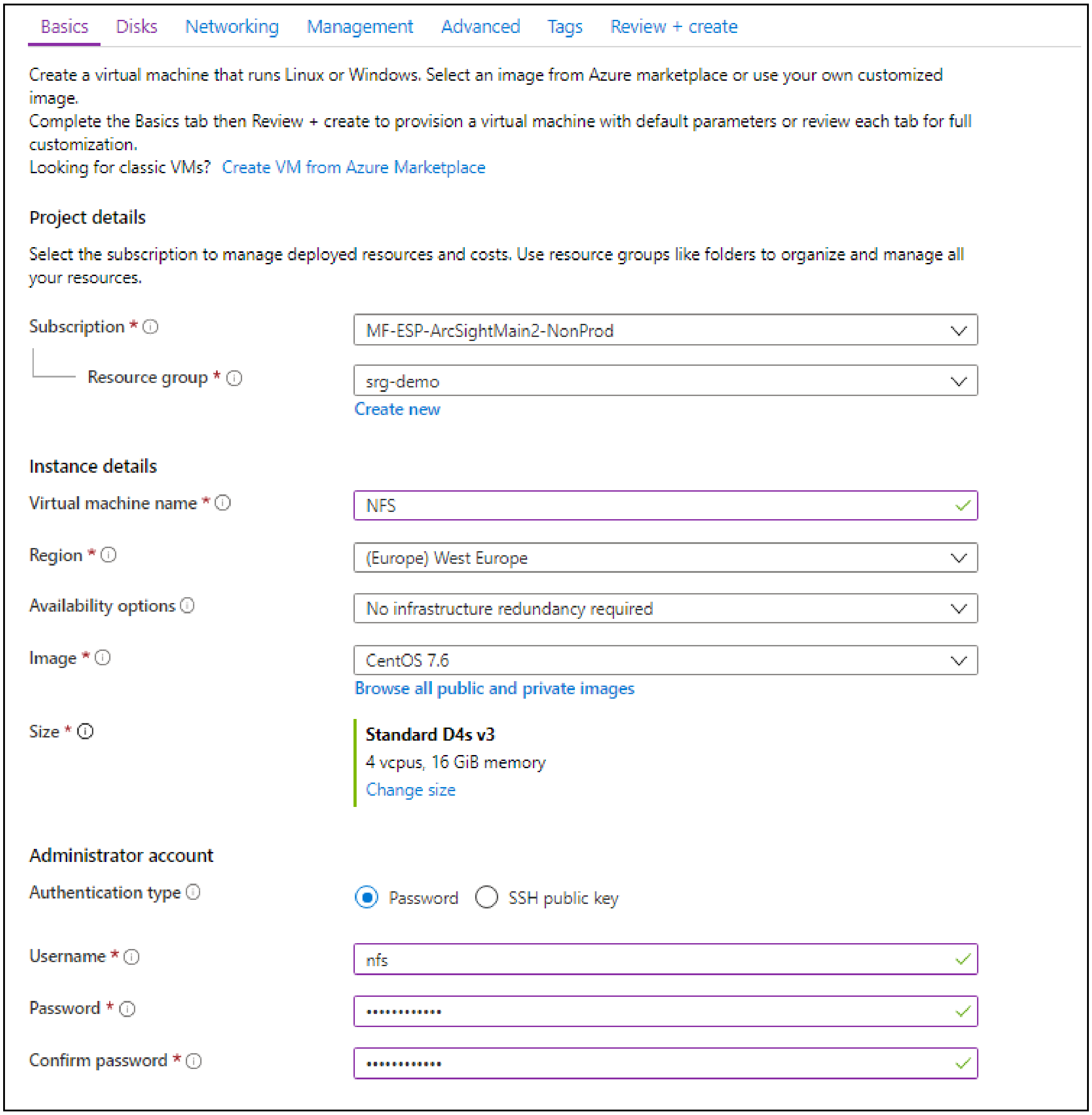
CDF requires an NFS server for operation. You can configure an NFS server using one of two methods:

- Configure a virtual machine which will be the host NFS server.
- Configure the native NetApp service provided by Azure to provision NFS shares.

Creating a Virtual Machine NFS Server

To create a VM NFS server using the Azure Portal:

1. Create a resource of type *CentOS-based.*
2. For **Resource group,** select your resource group.
3. In **Virtual machine name**, enter a VM name.
4. For **Size**, select **Change size.** In the popup, choose *D4s_V3,* and click **OK** to confirm. (You can select a different size according to your expected workload.)
5. Set the **Authentication type** to your preferences. Enter the **Administration account details** accordingly. In the examples given here, we use the username/password authentication.



Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image.
Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.
Looking for classic VMs? Create VM from Azure Marketplace

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	MF-ESP-ArcSightMain2-NonProd
Resource group *	srg-demo Create new

Instance details

Virtual machine name *	NFS ✓
Region *	(Europe) West Europe
Availability options	No infrastructure redundancy required
Image *	CentOS 7.6 Browse all public and private images
Size *	**Standard D4s v3** 4 vcpus, 16 GiB memory Change size

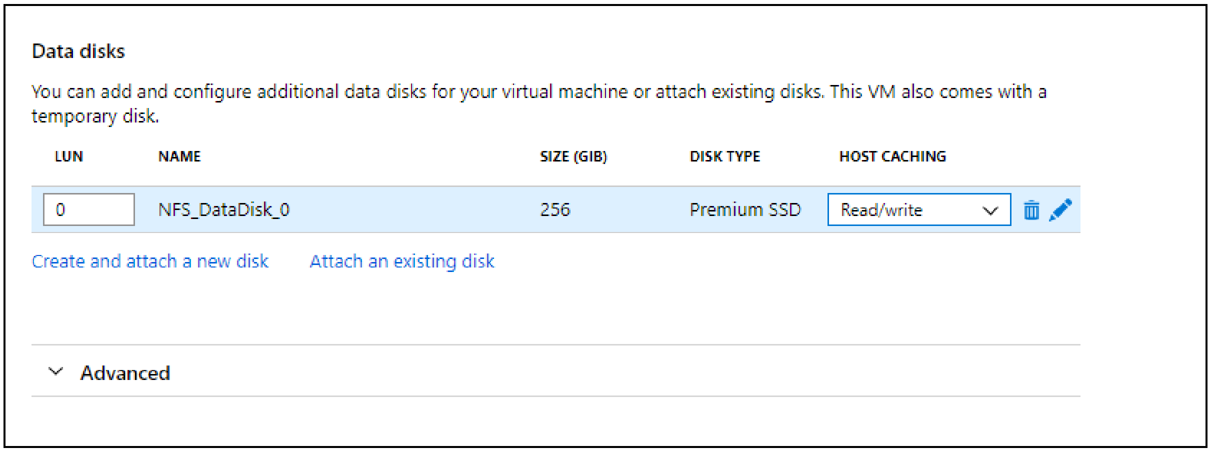
Administrator account

Authentication type	● Password ○ SSH public key
Username *	nfs ✓
Password *	•••••••••••• ✓
Confirm password *	•••••••••••• ✓

6. Click **Next: Disks**.

By default, the VM has a small (30GB) disk for the operating system and approximately the same size for the temporary disk. For NFS, we need to attach a new disk with IOPS 1100 or higher for better performance.

7. Click **Create.**
8. Attach a new disk and select a size that will meet your requirements. (IOPS should be 1100 or higher.)
9. After you add the disk, change the value in *Host Caching* to *Read/write.*

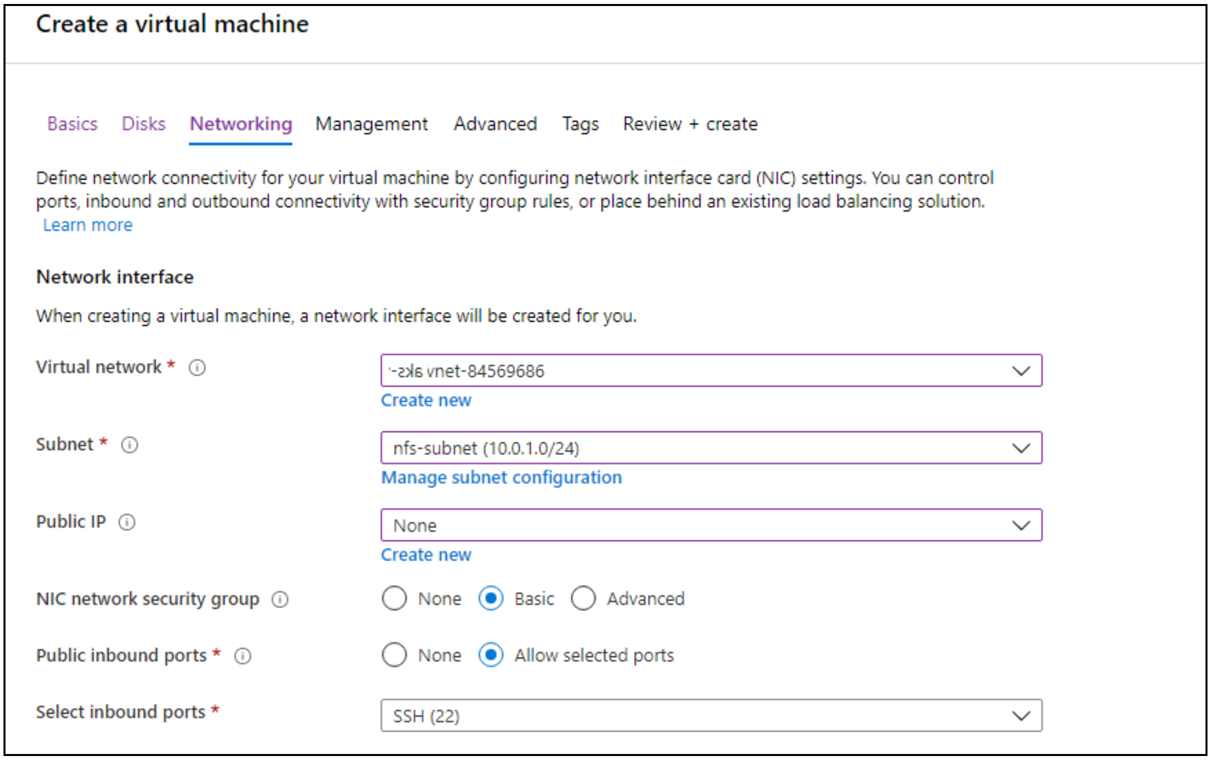


Data disks

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

LUN	NAME	SIZE (GIB)	DISK TYPE	HOST CACHING
0	NFS_DataDisk_0	256	Premium SSD	Read/write

10. Click **Next: Networking**
11. On the **Networking** tab, select values as follows:
    - **Virtual network**: select the virtual network you created earlier (for example, demo-vnet)
    - **Subnet:** select *nfs-subnet*
    - **NIC network security group:** select *Basic*
    - **Public Inbound Ports:** select None.



Create a virtual machine

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.
Learn more

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network *	'-ƨkɒ vnet-84569686 Create new
Subnet *	nfs-subnet (10.0.1.0/24) Manage subnet configuration
Public IP	None Create new
NIC network security group	○ None ● Basic ○ Advanced
Public inbound ports *	○ None ● Allow selected ports
Select inbound ports *	SSH (22)

11. Click **Review + create**.
12. When validation is passed, click **Create.**

To create a VM NFS server using the Azure Cloud Shell:

Required permissions: Create security groups, network interface and CentOS-based virtual machines

1. Set your main resource group name to an environment variable:
   ```
   # RESOURCE GROUP=<your resource group name>
   ```

For example:

```
# RESOURCE GROUP=srg-demo
```

2. Create a network security group for NFS by running this command:
   ```
   # az network nsg create -g $RESOURCE_GROUP -n nfs-nsg
   ```
3. Get the nfs-subnet ID and store it to an environment variable for later usage. (We will find subnet nfs-subnet in virtual network demo-vnet in resource group srg-demo.)
   ```
   # SUBNET_ID=$(az network vnet list -g $RESOURCE_GROUP | jq -r '.[] |
   select(.name == "<your_virtual_network_name>") | .subnets[] | select(.name
   == "<your_NFS_subnet>") | .id')
   ```
4. Create the network interface `nfs-VMNic` in the subnet from previous command in your resource using network security group nfs-nsg by running the following command:
   ```
   # az network nic create --name "nfs-VMNic" --resource-group $RESOURCE_
   GROUP --ip-forwarding "true" --network-security-group "nfs-nsg" --subnet
   $SUBNET_ID
   ```
5. Create the NFS VM by running this command:
   ```
   # az vm create --name "nfs" --resource-group $RESOURCE_GROUP --image
   "OpenLogic:CentOS:7.7:latest" --size "Standard_D4s_v3" --nics "nfs-VMNic"
   --data-disk-sizes-gb "256" --admin-username nfs --admin-password
   myStrongPassword@!123
   ```

Parameters:

`--size` is adjusted according to expected workload. To get a list of supported sizes, run the following command:

```
# az vm list-sizes -l <LOCATION> | jq ".[] | .name"
```

`--image` can be any supported CentOS. To get a list if CentOS images run the following command:

```
# az vm image list -l <LOCATION> -f CentOS --all
```

`--data-disk-sizes-gb` is specified according to workload. Use 256, 512, 1024 and so on.

Example result:

```
{
   "fqdns":"",
   "id":"/subscriptions/af379ae8-90b3-4368-8fe7-
b6a55ab17720/resourceGroups/srg-
demo/providers/Microsoft.Compute/virtualMachines/nfs",
   "location":"westeurope",
   "macAddress":"00-0D-3A-AA-E4-F7",
   "powerState":"VM running",
   "privateIpAddress":"10.1.2.4",
   "publicIpAddress":"",
   "resourceGroup":"srg-demo",
   "zones":""
}
```

The private IP will be used to access the NFS VM from the jumphost.

Next Step: Formatting the Disk on the NFS VM

Creating a VM NFS Server Using the Azure Cloud Shell

Required permissions: Create security groups, network interface and centos based virtual machines

1. Set your main resource group name to an environment variable:
   ```
   # RESOURCE GROUP=<your resource group name>
   ```

For example:
```
# RESOURCE GROUP=srg-demo
```

2. Create a network security group for NFS by running this command:
   ```
   # az network nsg create -g $RESOURCE_GROUP -n nfs-nsg
   ```
3. Get the nfs-subnet ID and store it to an environment variable for later usage. (We will find subnet nfs-subnet in virtual network demo-vnet in resource group srg-demo.)
   ```
   # SUBNET_ID=$(az network vnet list -g $RESOURCE_GROUP | jq -r '.[] |
   select(.name == "<your_virtual_network_name>") | .subnets[] | select(.name
   == "<your_NFS_subnet>") | .id')
   ```
4. Create the network interface `nfs-VMNic` in the subnet from previous command in your resource using network security group nfs-nsg by running the following command:
   ```
   # az network nic create --name "nfs-VMNic" --resource-group $RESOURCE_
   GROUP --ip-forwarding "true" --network-security-group "nfs-nsg" --subnet
   $SUBNET_ID
   ```
5. Create the NFS VM by running this command:
   ```
   # az vm create --name "nfs" --resource-group $RESOURCE_GROUP --image
   "OpenLogic:CentOS:7.7:latest" --size "Standard_D4s_v3" --nics "nfs-VMNic"
   --data-disk-sizes-gb "256" --admin-username nfs --admin-password
   myStrongPassword@!123
   ```

Parameters:

`--size` is adjusted according to expected workload. To get a list of supported sizes, run the following command:
```
# az vm list-sizes -l <LOCATION> | jq ".[] | .name"
```

`--image` can be any supported CentOS. To get a list if CentOS images run the following command:
```
# az vm image list -l <LOCATION> -f CentOS --all
```

`--data-disk-sizes-gb` is specified according to workload. Use 256, 512, 1024 and so on.

Example result:

```
{
"fqdns": "",
"id": "/subscriptions/af379ae8-90b3-4368-8fe7-b6a55ab17720/resourceGroups/srg-demo/providers/Microsoft.Compute/virtualMachines/nfs",
"location": "westeurope",
"macAddress": "00-0D-3A-AA-E4-F7",
"powerState": "VM running",
"privateIpAddress": "10.1.2.4",
"publicIpAddress": "",
"resourceGroup": "srg-demo",
"zones": ""
}
```

The private IP will be used to access the NFS VM from the jumphost.

Formatting the Disk on the NFS VM Using the Azure Cloud Shell

1. When your NFS VM deployment completes, determine its private IP address using the az command. Note the value for later usage.
2. From your jump host, SSH to the VM using its private IP address.

For example:

```
# ssh nfs@10.1.2.4
```

3. Log in using the user and password you specified earlier for the NFS VM.
4. Become root.
5. Find the device for the data disk by executing the command:
   `# fdisk -l`

> This will give you a list of existing disks. Usually the one added is named `/dev/sdc`.

6. Using `fdisk /dev/sdc`, create a new primary partition on whole device. Set it as type *83 - Linux.* Example commands are shown here:

```
# fdisk /dev/sdc
Welcome to fdisk (util-linux 2.23.2).

```

```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xc6a2cea5.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.

Command (m for help): n
Partition type:
p   primary (0 primary, 0 extended, 4 free)
e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-536870911, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-536870911, default 536870911):
Using default value 536870911
Partition 1 of type Linux and of size 256 GiB is set

Command (m for help): w
The partition table has been altered!
```

7. After saving the new partition table, run the command:
   `# mkfs.xfs /dev/sdc1`
8. Create a mountpoint. Run the command:
   `# mkdir /nfs`
9. Get the partition UUID. Run the command:
   `# blkid /dev/sdc1`

```
Example output:
/dev/sdc1: UUID="3696c212-1778-43d5-9d27-d9164686c327" TYPE="xfs"
```

10. In a text editor, open the file `/etc/fstab`. Add an entry to have this new partition mounted after restart. For example:
    `UUID=3696c212-1778-43d5-9d27-d9164686c327 /nfs xfs defaults 0 0`
11. Mount a new disk partition. Run the command:
    `# mount -a`
12. Verify it is properly mounted. Run the command:
    `# df -h`

Sample output:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2        30G  1.3G   29G   5% /
devtmpfs        7.9G     0  7.9G   0% /dev
tmpfs           7.9G     0  7.9G   0% /dev/shm
tmpfs           7.9G  9.0M  7.9G   1% /run
tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/sda1       497M   65M  433M  13% /boot
/dev/sdb1        99G   61M   94G   1% /mnt/resource <- Azure temporary drive
tmpfs           1.6G     0  1.6G   0% /run/user/0
/dev/sdc1       264G   33M  264G   1% /nfs          <- your new partition for
suite installation
```

Next Step: Prepare the NFS Server and Export Mountpoints

Preparing the NFS Server and Export Mountpoints

1. Check if `nfs-utils` is installed.

   ```
   # rpm -qa | grep nfs-utils
   ```

   Sample ouput: `nfs-utils-1.3.0-0.61.el7.x86_64`. The version might vary depending on your OS

2. If `nfs-utils` is not installed, install it by running the following command:

   ```
   # yum install -y nfs-utils
   ```

3. Configure NFS. Below is the suggested structure of the NFS volumes.

```
/nfs/itom-vol
/nfs/db-single-vol
/nfs/db-backup-vol
/nfs/itom-logging-vol
/nfs/arcsight-volume
```

4. For every NFS volume, run the following set of commands on the VM for NFS.

   ```
   mkdir -p /nfs/volume_name
   chown -R <uid>:<gid> /nfs/volume_name
   echo "/nfs/volume_name *(rw,sync,anonuid=<uid>,anongid=<gid>,all_
   squash)">>/etc/exports
   ```

   For example:

   ```
   mkdir -p /nfs/itom-vol
   chown -R 1999:1999 /nfs/itom-vol
   echo "/nfs/itom-vol *(rw,sync,anonuid=1999,anongid=1999,all_squash)">>/etc/exports
   ```

   > If you use a UID/GID different than 1999/1999, it must be provided during installation.

5. After configuring all 5 required volumes, run the following commands.

   ```
   # exportfs -ra
   # systemctl restart rpcbind
   # systemctl enable rpcbind
   # systemctl restart nfs-server
   # systemctl enable nfs-server
   ```

NFS configuration is now complete.

Next Step: Create and Attach the Data Disk to Nodes

Using NetApp as an NFS Server

You can use Azure's NetApp service as an NFS server. To request access to the service, see the Azure NetApp Files waitlist submission page. You must wait for an official confirmation email from the Azure NetApp Files team before continuing.

Next Step: Create the NetApp Account

Creating the NetApp Account

To create your NetApp account using the Azure Portal:

1. Create a resource of type *Azure NetApp Files.*
2. Choose a name and your subscription for the resource.

3. For **Resource group,** select your Kubernetes resource group where you have your virtual network and subnet for NFS.
4. For **Location**, select your resource group as in Step 3.
5. Click **Create** and wait for account creation.

To create your NetApp account using the Azure Cloud Shell:

1. Set your main resource group name to an environment variable:
   ```
   # RESOURCE_GROUP=<your resource group name>
   ```

For example:
```
# RESOURCE_GROUP=srg-demo
```

2. Create the NetApp account by running the following command:
   ```
   az netappfiles account create -g $RESOURCE_GROUP --name <ACCOUNT_NAME> -l
   <LOCATION>
   ```

For example:
```
# az netappfiles account create -g $RESOURCE_GROUP --name SrgDemoNetAppAdmin
-l westeurope
```

Parameters:

`<ACCOUNT NAME>` is your NetApp account name.

`<LOCATION>` is the same as for AKS

Next Step: Setting Up the Capacity Pool

Creating Your NetApp Account Using the Azure Cloud Shell

1. Set your main resource group name to an environment variable:

   ```
   # RESOURCE_GROUP=<your resource group name>
   ```

For example:

```
# RESOURCE_GROUP=srg-demo
```

2. Create the NetApp account by running the following command:

   ```
   az netappfiles account create -g $RESOURCE_GROUP --name <ACCOUNT_NAME> -l
   <LOCATION>
   ```

Parameters:

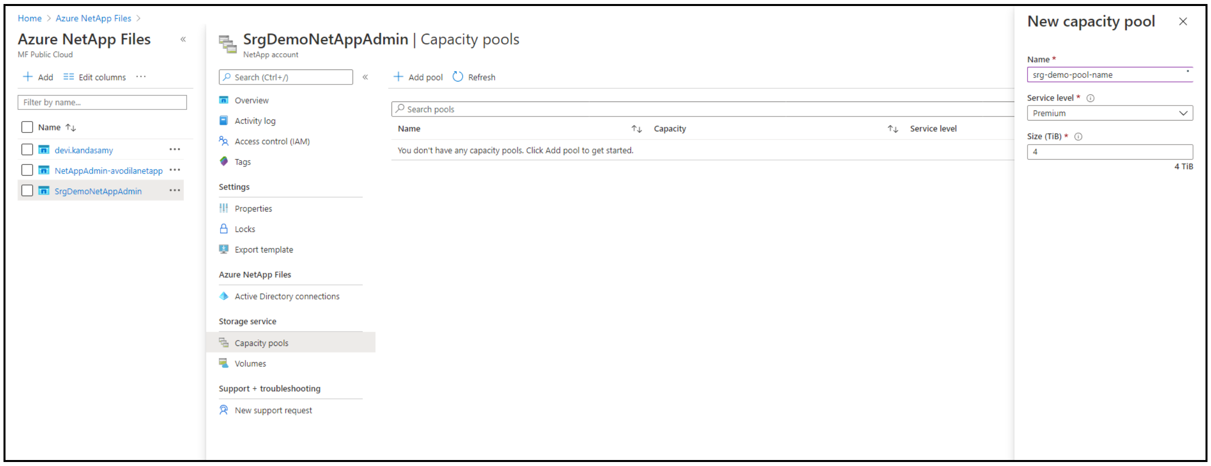
`<ACCOUNT NAME>` is your NetApp account name.

`<LOCATION>` is the same as for AKS

Setting Up the NetApp Capacity Pool

To set up the capacity pool using the Azure Portal:

1. In the **Azure NetApp Files** tab, browse to your NetApp account.
2. In **Storage Services**, select **Capacity Pools.**



You don't have any capacity pools. Click Add pool to get started.

Name * srg-demo-pool-name

Service level * Premium

Size (TiB) * 4

4 TiB

3. Click **+** and enter values for the following:
   - **Name:** enter a name for the pool.
   - **Service Level:** select a service level.
   - **Pool Size:** enter 4 (TB) for the pool size. (This is a service minimum.)

To set up the capacity pool using the Azure Cloud Shell:

1. Set your main resource group name to an environment variable:
   ```
   # RESOURCE_GROUP=<your resource group name>
   ```

For example:
```
# RESOURCE_GROUP=srg-demo
```

2. Run the command:
   ```
   # az netappfiles pool create -g $RESOURCE_GROUP --account-name <ACCOUNT_
   NAME> --name <POOL_NAME> -l <LOCATION> --size 4 --service-level premium
   ```

Parameters:

`<POOL_NAME>` is your new pool name

`<ACCOUNT_NAME>` is the NetApp account name specified in previous step

For example:
```
# az netappfiles pool create -g $RESOURCE_GROUP --account-name
SrgDemoNetAppAdmin --name srg-demo-pool-name -l westeurope --size 4 --
service-level premium
```

Next Step: Creating and Preparing the Volume

Setting Up the Capacity Pool Using the Azure Cloud Shell

1. Set your main resource group name to an environment variable:
   ```
   # RESOURCE_GROUP=<your resource group name>
   ```

For example:

```
# RESOURCE_GROUP=srg-demo
```

2. Run the command:
   ```
   # az netappfiles pool create -g $RESOURCE_GROUP --account-name <ACCOUNT_
   NAME> --name <POOL_NAME> -l <LOCATION> --size 4 --service-level premium
   ```

Parameters:

`<POOL_NAME>` is your new pool name

`<ACCOUNT_NAME>` is the NetApp account name specified in previous step

For example:

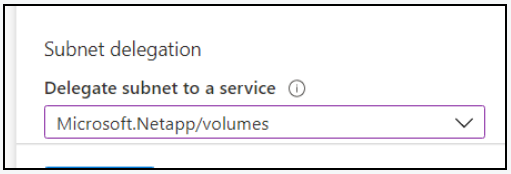
```
# az netappfiles pool create -g $RESOURCE_GROUP --account-name
SrgDemoNetAppAdmin --name srg-demo-pool-name -l westeurope --size 4 --
service-level premium
```

Creating and Preparing the Volume

To create and prepare the volume using the Azure Portal:

1. In your NetApp account resource, browse to **Storage service.**
2. Select **Volumes.**
3. Press **+ Add volume.**
4. Enter a name for the volume
5. Ensure that your volume is in the same virtual network as `aks-virtual-network` and `nfs-subnet.`
6. For **Subnet**, select *nfs-subnet.*

Under **Subnet delegation,** the subnet must be delegated to `Microsoft.NetApp/volumes`.



Subnet delegation

Delegate subnet to a service

Microsoft.Netapp/volumes

7. Click **Next:Protocol**
8. Ensure that the **Protocol** type is *NFS*.
9. Make sure **Version** is *NFSv3.*
10. In **Export policy section** , select the checkbox for *0.0.0.0/0 Read & Write.*
11. Enter the file path that will be used to create the export path for the volume
12. Click **Review + Create** at the bottom of the page. If you are satisfied with your settings, click **Create**.

To create and prepare the volume using the Azure Cloud Shell:

1. Set your main resource group name to an environment variable:

```
# RESOURCE GROUP=<your resource group name>
```

For example:

```
# RESOURCE GROUP=srg-demo
```

2. Run the command:

```
# az netappfiles volume create -g $RESOURCE_GROUP --account-name <ACCOUNT_
NAME> --pool-name <POOL_NAME> --name <VOLUME_ROOT> -l <LOCATION> --
service-level premium --usage-threshold <VOLUME_SIZE> --file-path <FILE_
PATH> --vnet <VIRTUAL_NETWORK> --subnet <NFS_SUBNET_NAME> --protocol-types
NFSv3
```

Parameters:

<ACCOUNT_NAME> is your netApp account name.

<POOL_NAME> is the capacity pool created on the previous step.

<VOLUME_ROOT> is your volume root name.

<LOCATION> is location of your NetApp.

<VOLUME_SIZE> size for NFS volume in GB.

<FILE_PATH> is the path to your volumes.

<VIRTUAL_NETWORK> the virtual network to which your subnets belong.

<NFS_SUBNET_NAME> is your subnet for NFS.

For example:

```
az netappfiles volume create -g $RESOURCE_GROUP --account-name
SrgDemoNetAppAdmin --pool-name srg-demo-pool-name --name volumename -l
westeurope --service-level premium --usage-threshold 100 --file-path "nfs" --
vnet demo-vnet --subnet nfs-subnet --protocol-types NFSv3
```

Next Step: Configure and Create the Volumes

Creating and Preparing the Volume Using the Azure Cloud Shell

1. Set your main resource group name to an environment variable:
   ```
   # RESOURCE GROUP=<your resource group name>
   ```

For example:

```
# RESOURCE GROUP=srg-demo
```

2. Run the command:
   ```
   # az netappfiles volume create -g $RESOURCE_GROUP --account-name <ACCOUNT_
   NAME> --pool-name <POOL_NAME> --name <VOLUME_ROOT> -l <LOCATION> --
   service-level premium --usage-threshold <VOLUME_SIZE> --file-path <FILE_
   PATH> --vnet <VIRTUAL_NETWORK> --subnet <NFS_SUBNET_NAME> --protocol-types
   NFSv3
   ```

Parameters:

`<ACCOUNT_NAME>` is your netApp account name.

`<POOL_NAME>` is the capacity pool created on the previous step.

`<VOLUME_ROOT>` is your volume root name.

`<LOCATION>` is location of your NetApp.

`<VOLUME_SIZE>` size for NFS volume in GB.

`<FILE_PATH>` is the path to your volumes.

`<VIRTUAL_NETWORK>` the virtual network to which your subnets belong.

`<NFS_SUBNET_NAME>` is your subnet for NFS.

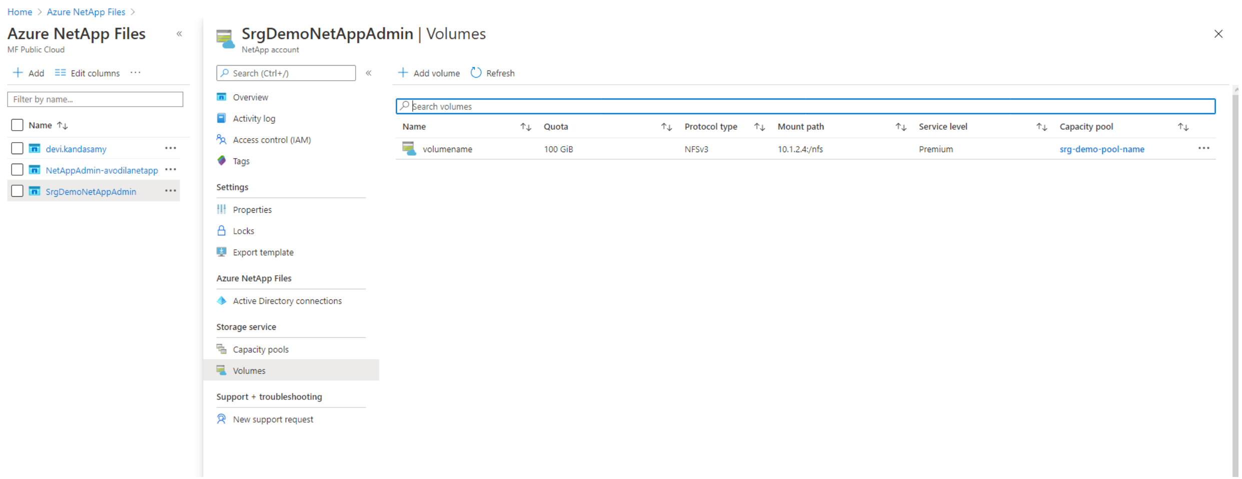
For example:

```
az netappfiles volume create -g $RESOURCE_GROUP --account-name
SrgDemoNetAppAdmin --pool-name srg-demo-pool-name --name volumename -l
westeurope --service-level premium --usage-threshold 100 --file-path "nfs" --
vnet demo-vnet --subnet nfs-subnet --protocol-types NFSv3
```

Configure and Create the Volumes

To configure and create the volumes:

1. Find your `<mount_path_ip>` and `<mount_path_file_name>` by navigating to your volume page. They were confirmed in the previous step after successful creation of the prepared volume. Note both of these parameters for later use.



Name	Quota	Protocol type	Mount path	Service level	Capacity pool
volumename	100 GiB	NFSv3	10.1.2.4:/nfs	Premium	srg-demo-pool-name

2. Log in to the jump host.
3. If not already present, install the NFS client by sudo:
   ```
   # yum install nfs-utils if not present
   ```
4. Get and unzip the file `cdf-deployer-<VERSION>.zip`
5. In the `scripts` folder, run the following command:
   ```
   # sudo ./createFileStore.sh <mount_path_ip> <mount_path_file_name>
   ```

Example:
```
sudo ./createFileStore.sh "10.1.2.4" "/nfs"
```

Next Step: Create and Attach the Data Disk to Nodes

Creating and Attaching the Data Disk to Nodes

By default, AKS nodes are created with a temporary data disk. Disk size depends on the `--node-vm-size` parameter and might not fit your needs.

In this section, you will prepare the Azure managed disk and attach it to the nodes which will host Transformation Hub. This process has three parts:

1. Creating the managed data disk.
2. Attaching the disk to the AKS node.
3. Formatting and mounting the attached disk.

Each of these procedures is explained below.

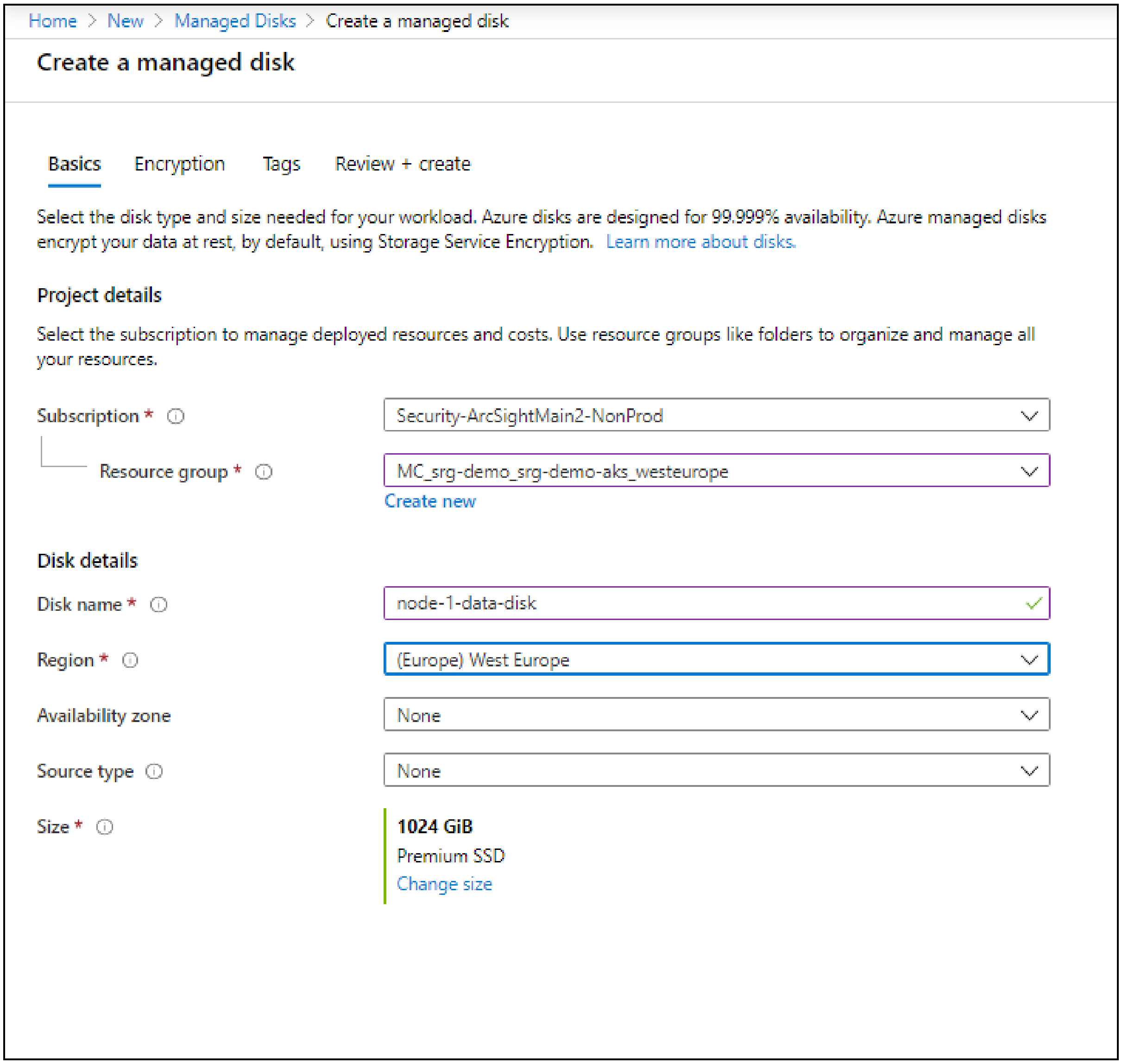
Next Step: Create the Managed Disk

Creating the Managed Data Disk

To create the managed data disk for a node using the Azure Portal:

1. Create a resource of type *Managed Disks.*

2. For **Resource group** select your AKS resource group. (the AKS resource group is named in the in format MC_<your_resource_group>_<aks_name>_<location>)
3. In **Disk name**, enter a name for the managed disk,
4. Select proper **Region** (based on the location you specified earlier).
5. Select your **Size** based on expected workload.



Create a managed disk

Select the disk type and size needed for your workload. Azure disks are designed for 99.999% availability. Azure managed disks encrypt your data at rest, by default, using Storage Service Encryption. Learn more about disks.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Security-ArcSightMain2-NonProd

Resource group * MC_srg-demo_srg-demo-aks_westeurope

Disk details

Disk name * node-1-data-disk

Region * (Europe) West Europe

Availability zone None

Source type None

Size * 1024 GiB
Premium SSD

6. Click **Review + create.** After validation, click **Create**.
7. When deployment finishes, click **Download** to get json with deployment results. Inside the archive is the file deployment.json. Note or copy the `primaryResourceId` value for later use in attaching this disk to the AKS node
8. Repeat steps 1 through 8 for each AKS node.

To create the managed data disk using the Azure Cloud Shell:

Required permissions: create disk

1. Get the AKS resource group and store it in an environment variable for later usage:
   ```
   # CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group <RESOURCE GROUP> -
   -name <AKS NAME> --query nodeResourceGroup -o tsv)
   ```

For example:

```
# CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group srg-demo --name srg-
demo-aks --query nodeResourceGroup -o tsv)
```

2. Create the managed disk by running the following command:
   ```
   # az disk create --name <DISK NAME> --resource-group $CLUSTER_RESOURCE_
   GROUP --size-gb <DISK SIZE>
   ```

For example:

```
# az disk create --name node-1-data-disk --resource-group $CLUSTER_RESOURCE_
GROUP --size-gb 1024
```

3. From the results, get the `id` value. It will be used later to attach the disk to the AKS node.
   The value will resemble the following:
   ```
   /subscriptions/af379ae8-90b3-4368-8fe7-b6a55ab17720/resourceGroups/MC_srg-
   demo_srg-demo-aks_westeurope/providers/Microsoft.Compute/disks/node-1-
   data-disk
   ```
4. Repeat the above steps for each expected AKS node.

Next Step: Attach the Disk to the AKS Node

Creating the Managed Data Disk Using the Azure Cloud Shell

Required permissions: create disk

1. Get the AKS resource group and store it in an environment variable for later usage:
   ```
   # CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group <RESOURCE GROUP> --name <AKS NAME> --query nodeResourceGroup -o tsv)
   ```

For example:

```
# CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group srg-demo --name srg-demo-aks --query nodeResourceGroup -o tsv)
```

2. Create the managed disk by running the following command:
   ```
   # az disk create --name <DISK NAME> --resource-group $CLUSTER_RESOURCE_GROUP --size-gb <DISK SIZE>
   ```

For example:

```
# az disk create --name node-1-data-disk --resource-group $CLUSTER_RESOURCE_GROUP --size-gb 1024
```

3. From the results, get the `id` value. It will be used later to attach the disk to the AKS node.

   The value will resemble the following:
   ```
   /subscriptions/af379ae8-90b3-4368-8fe7-b6a55ab17720/resourceGroups/MC_srg-demo_srg-demo-aks_westeurope/providers/Microsoft.Compute/disks/node-1-data-disk
   ```
4. Repeat the above steps for each expected AKS node.

Attaching the Disk to the AKS Node

1. Get the Virtual machine scale set and store it to an environment variable:
   ```
   # VMSS=$(az vmss list -g $CLUSTER_RESOURCE_GROUP | jq -r .[0].name)
   ```

> If you open a new session, run commands to set the `CLUSTER_RESOURCE_GROUP` environment variable first.

2. Attach the disk to instance by running the command:
   ```
   # az vmss disk attach --resource-group $CLUSTER_RESOURCE_GROUP --vmss-name $VMSS --instance-id <INDEX OF INSTANCE> --disk <DISK ID>
   ```

Parameters:

`<INDEX OF INSTANCE>` is number of node in cluster, starting from 0

`<DISK ID>` disk resource ID, obtained during creation.

> For example, the following command would attach the disk to the instance with an instance -id of 0.
> `# az vmss disk attach --resource-group $CLUSTER_RESOURCE_GROUP --vmss-name $VMSS --instance-id 0 --disk /subscriptions/af379ae8-90b3-4368-8fe7-b6a55ab17720/resourcegroups/MC_ srg- demo_ srg- demo- aks_ westeurope/providers/Microsoft.Compute/disks/node-1-data-disk`

3. Repeat steps 1-2 for the rest of the disks and node instances

Next Step: Format and Mount the Attached Disk

Formatting and Mounting the Attached Disk

1. Upload the `id_rsa` to your jump host.

> Use the `id_rsa` file you generated when creating the Azure Kubernetes Service.

2. Make `id_rsa` read only by running the command:
   `# chmod 400 id_rsa`
3. Get the node's private IP by running the command:
   `# kubectl get nodes -o wide`

Example output:

```
NAME                                STATUS   ROLES   AGE   VERSION
INTERNAL-IP   EXTERNAL-IP

aks-nodepool1-84569686-vmss000000   Ready    agent   79m   v1.15.10
10.240.0.4    <none>

aks-nodepool1-84569686-vmss000001   Ready    agent   79m   v1.15.10
10.240.0.5    <none>

aks-nodepool1-84569686-vmss000002   Ready    agent   79m   v1.15.10
10.240.0.6    <none>
```

4. Make an SSH connection from the jumphost to an AKS node (use the internal IP address). Successive commands will be executed on the AKS node to which you are connected.

Example command:

```
# ssh -i id_rsa azureuser@10.240.0.4
```

5. On the same AKS node, become `root`.
6. Find the device for the data disk by running the command:
   `# fdisk -l`

This will give you list of existing disks. Usually the one added is named `/dev/sdc`

7. Using `fdisk /dev/sdc`, create a new primary partition on the whole device and set it as type *83 - Linux.*

Example input and output:

```
# fdisk /dev/sdc
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xc6a2cea5.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.

Command (m for help): n
Partition type:
p   primary (0 primary, 0 extended, 4 free)
e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-536870911, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-536870911, default 536870911):
Using default value 536870911
Partition 1 of type Linux and of size 1024 GiB is set

Command (m for help): w
The partition table has been altered!
```

8. After saving the new partition table, create the file system by running the command:
   `mkfs.xfs /dev/sdc1`
9. Create the mountpoint:
   ` mkdir /opt/arcsight`
10. Get the partition UUID by running the command:
    `# blkid /dev/sdc1`

Example output:

```
/dev/sdc1: UUID="3696c212-1778-43d5-9d27-d9164686c327" TYPE="xfs"
```

11. Add an entry to the `/etc/fstab` file to have this new partition mounted after restart. For example:
    `UUID=3696c212-1778-43d5-9d27-d9164686c327 /opt/arcsight xfs defaults 0 0`
12. Mount a new disk partition by running the command:
    `# mount -a`
13. Verify it is properly mounted by running:
    `# df -h`

Example output:

```
Filesystem      Size  Used Avail Use% Mounted on

udev            7.9G     0  7.9G   0% /dev

tmpfs           1.6G  812K  1.6G   1% /run

/dev/sda1        97G  9.4G   88G  10% /              <- Azure temporary drive

tmpfs           7.9G     0  7.9G   0% /dev/shm

tmpfs           5.0M     0  5.0M   0% /run/lock

tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup

/dev/sda15      105M  3.6M  101M   4% /boot/efi

/dev/sdb1        32G   48M   30G   1% /mnt

tmpfs           7.9G   12K  7.9G   1% /var/lib/kubelet/pods/7194d3a7-cc84-
42bd-accb-30b09fcd1d27/volumes/kubernetes.io~secret/kube-proxy-token-cnxn8

overlay          97G  9.4G   88G  10%
/var/lib/docker/overlay2/3e04813889c25709c31206a48ee82fa67d677b76a6b1aab5e7d7
246b911a3bee/merged

shm              64M     0   64M   0%
/var/lib/docker/containers/bc0dd2ea23a9c0640e10ad4664addeb437f4ad4ac0830260ee
f942f70bcb0c0a/mounts/shm
```

```
overlay           97G  9.4G   88G  10%
/var/lib/docker/overlay2/b8290059f18b2f9d311395abcf12ccb377ed7107db5fa5fccc46
b6fc594e7da8/merged
```

```
tmpfs            1.6G     0  1.6G   0% /run/user/1000
```

```
/dev/sdc1        1.0T  1.1G 1023G   1% /opt/arcsight   <- your new partition
for Arcsight products
```

14. Repeat Steps 4 through 12 for all remaining nodes and their disks.

Next Step: Preparing a Private DNS Zone

Preparing a Private DNS Zone

Required permissions: create private DNS zone; a link to the virtual network is also needed,

1. Set your main resource group name to an environment variable, for example.:
   `# RESOURCE_GROUP=srg-demo`
2. Create the private-dns zone (for example, `arcsight.private.com`) in your resource group by running the command:
   ```
   # az network private-dns zone create -g $RESOURCE_GROUP -n
   arcsight.private.com
   ```

> You can use another zone name in place of `arcsight.private.com`, but you must use the same DNS suffix for the `--external-access-host` argument during CDF installation.

3. Link private-dns zone with your virtual network by running the command:
   ```
   # az network private-dns link vnet create \
   -g $RESOURCE_GROUP \
   -n DNSLink \
   -z arcsight.private.com \
   -v <your virtual network, such as demo-vnet> \
   -e false
   ```

Next Step: Assigning an IP Address to Private DNS

Assigning an IP Address to Private DNS

In this step you will assign an IP address from the aks-subnet range to the domain name (external access host).

1. Set your main resource group name to an environment variable; for example:
   `# RESOURCE_GROUP=srg-demo`

> Alternatively, use the resource group where your `vnet` and `private-dns` zone are located.

2. Get the address prefix by running the command:
   ```
   # az network vnet subnet show -g $RESOURCE_GROUP --vnet-name <your virtual
   ```

```
network> --name <subnet for AKS>| jq -r .addressPrefix
```

For example:

```
# az network vnet subnet show -g $RESOURCE_GROUP --vnet-name demo-vnet --
name aks-subnet | jq -r .addressPrefix
```

Example result:

```
10.1.1.0/24
```

You can select any IP from this range. The first N IP addresses are occupied by AKS nodes.

Example selection in this range: `10.1.1.101`

3. Assign the IP by running the following command:

```
# az network private-dns record-set a add-record -g $RESOURCE_GROUP -z
<PRIVATE DNS ZONE> -n <RECORD SET NAME> -a <EXTERNAL-IP>
```

Parameters

`<PRIVATE DNS ZONE>` is the `private-dns` zone created earlier (in our example it was `arcsight.private.com`)

`<RECORD SET NAME>` the name of the record set relative to the zone (in our example, `installer`)

`<EXTERNAL-IP>` IP must be from aks-subnet range

Example command:

```
# az network private-dns record-set a add-record -g $RESOURCE_GROUP -z
arcsight.private.com -n installer -a 10.1.1.101
```

This command will create `installer.arcsight.private.com` with the IP address 10.1.1.101.

4. Note the domain name and IP address. The domain name will be used as the `--external-access-host` parameter for CDF installation and the IP for patching the load balancer.

Next Step: Labeling Worker Nodes

Labeling AKS Nodes (Azure)

Labeling is a means for identifying application processing and qualifying the application as a candidate to run on a specific node. For example, labeling an Azure Kubernetes Service (AKS) node with the label `kafka=yes` specifies that a Kafka instance will run on that node.

Labels required for AKS nodes include the following:

Label	Purpose
`kafka=yes`	Run Kafka
`zk=yes`	Run ZooKeeper (Kafka management tool)
`th-processing=yes`	Process Transformation Hub data
`th-platform=yes`	Run Transformation Hub

To label your AKS nodes:

1. On your jump host, get a list of AKS nodes by running the following command:
   `# kubectl get nodes`

```
NAME                                STATUS   ROLES   AGE    VERSION
aks-agentpool-36457641-vmss000000   Ready    agent   137m   v1.13.11
aks-agentpool-36457641-vmss000001   Ready    agent   137m   v1.13.11
aks-agentpool-36457641-vmss000002   Ready    agent   137m   v1.13.11
```

2. Label the first AKS node by running the following command:
   `# kubectl label node <node-name> zk=yes kafka=yes th-processing=yes th-platform=yes`

> For example: `# kubectl label node aks-agentpool-36457641-vmss000000 zk=yes kafka=yes th-processing=yes th-platform=yes`

3. Repeat the command in step 2 for each additional node.

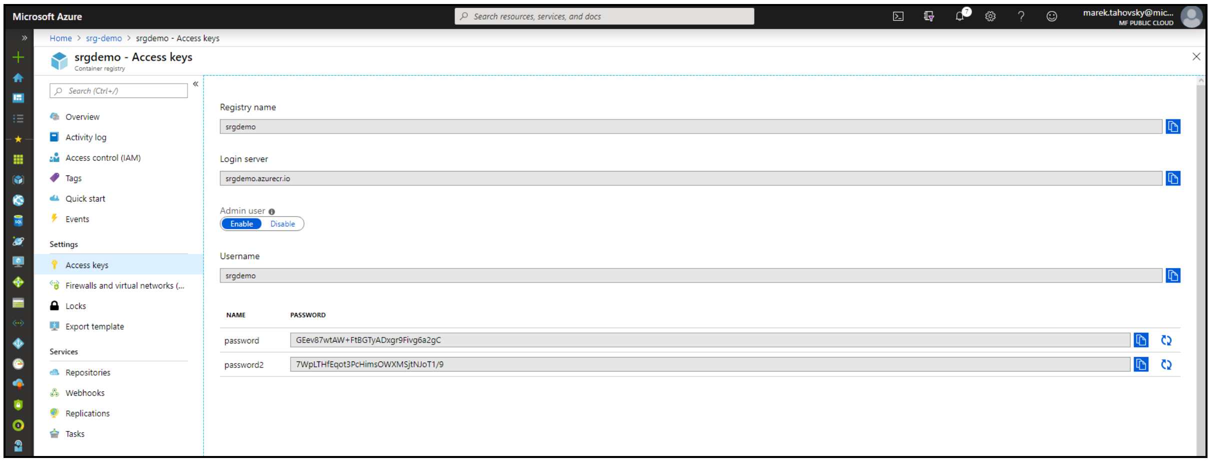
Next Step: Uploading Product Images to the ACR

Uploading the Product Images to the ACR

To upload the product images to the Azure Container Registry, you must first determine the ACR credentials, and then perform the upload.

To upload the product images to the ACR, using the Azure Portal to get credentials:

1. Download the product images and CDF deployer to a secure network location.
2. On the Azure Portal, open the Azure Container Registry.
3. In **Settings**, select the **Access keys** tab.
4. Take note of the values for **Login server, Username,** and **Password**.



srgdemo - Access keys
Container registry

Registry name
srgdemo

Login server
srgdemo.azurecr.io

Admin user
Enable Disable

Username
srgdemo

NAME	PASSWORD
password	GEev87wtAW+FtBGTyADxgr9Fivg6a2gC
password2	7WpLTHfEqot3PcHimsOWXMSjtNJoT1/9

5. Unzip the CDF deployer to a local directory (such as `/tmp`).

   For example:

   ```
   # cd /tmp
   # unzip cdf-deployer-<VERSION>.zip...
   ```

6. Change directory to the deployer scripts folder.

   For example:

   ```
   # cd cdf-deployer-<VERSION>/scripts/
   ```

7. Run the `uploadimages` script with credentials from the ACR by running the following command:

   ```
   # ./uploadimages.sh -o your-org-name -r <REGISTRY LOGIN SERVER> -u <USERNAME> -p
   <PASSWORD> -F /tmp/cdf-byok-images-<VERSION>.tar -c 2
   ```

   For example:

   ```
   ./uploadimages.sh -o your-org-name -r srgdemo.azurecr.io -u srgdemo -p
   GEev87wtAW+FtBGTyADxgr9Fivg6a2gC -F /tmp/cdf-byok-images-<VERSION>.tar -c 2...
   Upload completed in 1690 seconds.
   Upload-process successfully completed.
   ```

To upload product images to the ACR, using Azure commands to get the credentials:

1. Get the registry name and password by running these commands:

   ```
   # credentials=$(az acr credential show --name <your ACR name> -g <RESOURCE GROUP>)
   # echo $credentials | jq -r '.username'
   # echo $credentials | jq -r '.passwords[0].value'
   ```

   For example:

   ```
   # credentials=$(az acr credential show --name srgdemoACR -g srg-demo)
   # echo $credentials | jq -r '.username'
   # echo $credentials | jq -r '.passwords[0].value'
   ```

2. Determine the name of the registry login server by running the following command:

   ```
   # az acr show --name <REGISTRY NAME> -g <your ACR name> | jq -r '.loginServer'
   ```

   For example:

   ```
   # az acr show --name srgdemoACR -g srg-demo | jq -r '.loginServer'
   ```

3. Unzip the CDF deployer to a local directory (such as `/tmp`).

   For example:

   ```
   # cd /tmp
   # unzip cdf-deployer-<VERSION>.zip...
   ```

4. Change directory to the deployer scripts folder.

   For example:

   ```
   # cd cdf-deployer-<VERSION>/scripts/
   ```

5. Run the uploadimages script with credentials from the ACR by running the following command:

   ```
   # ./uploadimages.sh -o your-org-name -r <REGISTRY LOGIN SERVER> -u <USERNAME> -p
   <PASSWORD> -F /tmp/cdf-byok-images-<VERSION>.tar -c 2
   ```

   For example:

   ```
   # ./uploadimages.sh -o your-org-name -r srgdemo.azurecr.io -u srgdemo -p
   GEev87wtAW+FtBGTyADxgr9Fivg6a2gC -F /tmp/cdf-byok-images-<VERSION>.tar -c 2
   ...
   Upload completed in 1690 seconds.
   Upload-process successfully completed.
   ```

About Running `uploadimages.sh`

- Choose the value of the `-o` argument carefully, since it needs to be used during installation and for all `uploadimages.sh` calls.
- The `-c` argument indicates concurrent upload (maximum can be half of the CPU cores capacity), and can speed up the upload process.
- Uploading images is long process, and can take up to 60 minutes to complete. The exact time for completion depends largely on connectivity.
- See `uploadimages.sh --help` for more information.

Next Step: Uploading the JDBC Driver

Uploading Product Images to the ACR Using Azure Commands to get the Credentials

1. Get the registry name and password by running these commands:

   ```
   # credentials=$(az acr credential show --name <your ACR name> -g <RESOURCE
   GROUP>)
   # echo $credentials | jq -r '.username'
   # echo $credentials | jq -r '.passwords[0].value'
   ```

> For example:
> ```
> # credentials=$(az acr credential show --name srgdemoACR -g srg-demo)
> # echo $credentials | jq - r '.username'
> # echo $credentials | jq -r '.passwords[0].value'
> ```

1. Determine the name of the registry login server by running the following command:

   ```
   # az acr show --name <REGISTRY NAME> -g <your ACR name> | jq -r
   '.loginServer'
   ```

   > For example:
   > ```
   > # az acr show --name srgdemoACR -g srg-demo | jq -r '.loginServer'
   > ```

2. Unzip the CDF deployer to a local directory (such as /tmp). For example:

   ```
   # cd /tmp
   ```

   ```
   # unzip cdf-deployer-<VERSION>.zip...
   ```

3. Change directory to the deployer scripts folder. For example:

   ```
   # cd cdf-deployer-<VERSION>/scripts/
   ```

4. Run the uploadimages script with credentials from the ACR by running the following command:

   ```
   # ./uploadimages.sh -o your-org-name -r <REGISTRY LOGIN SERVER> -u <USERNAME> -p
   <PASSWORD> -F /tmp/cdf-byok-images-<VERSION>.tar -c 2
   ```

For example:

```
# ./uploadimages.sh -o your-org-name -r srgdemo.azurecr.io -u srgdemo -p
GEev87wtAW+FtBGTyADxgr9Fivg6a2gC -F /tmp/cdf-byok-images-<VERSION>.tar -c 2
```

```
...
```

```
Upload completed in 1690 seconds.
```

```
Upload-process successfully completed.
```

About Running `uploadimages.sh`

- Choose the value of the `-o` argument carefully, since it needs to be used during installation and for all `uploadimages.sh` calls.
- The `-c` argument indicates concurrent upload (maximum can be half of the CPU cores capacity), and can speed up the upload process.
- Uploading images is long process, and can take up to 60 minutes to complete. The exact time for completion depends largely on connectivity.
- See `uploadimages.sh --help` for more information

Uploading the JDBC Driver

In the deployer scripts directory, run `build_jdbc.sh.`

For example:

```
# cd /tmp/cdf-deployer-<VERSION>/scripts/
```

```
# ./build_jdbc.sh -o <your-org-name> -d /tmp/jdbc
```

```
...
```

```
# ./uploadimages.sh -o <your-org-name> -r srgdemo.azurecr.io -u srgdemo -p
GEev87wtAW+FtBGTyADxgr9Fivg6a2gC -d /tmp/jdbc
```

```
...
```

```
** Uploading suite images
```

```
Upload image [1/1] srgdemoacr.azurecr.io/your-org-name/jdbc-drivers-
container:1.0 ... OK
```

```
Upload completed in 248 seconds.
```

```
Upload-process successfully completed.
```

Installing CDF

1. Download the CDF deployer and the ArcSight metadata files to a secure network location.
2. SSH to your jump host and upload the CDF deployer and ArcSight metadata files to a directory on the jump host.
3. Unzip the deployer and run the installation. For example:

```
        # unzip cdf-deployer-<VERSION>.zip
```

```
        ...
```

```
        # cd cdf-deployer-<VERSION>/
```

```
       # ./install --nfs-server 10.1.2.4 --nfs-folder /nfs/itom-vol --registry-
orgname your-org-name --registry-url srgdemo.azurecr.io --registry-username
srgdemo --registry-password    GEev87wtAW+FtBGTyADxgr9Fivg6a2gC --external-
access-host installer.arcsight.private.com --noinfra --cloud-provider AZURE
```

The following arguments apply to the `install` command:

`--nfs-server`	Your NFS server private IP or NetApp end IP.
`--registry-url`	Login server (from the **Access keys** tab in your ACR resource)
`--registry-username`	Username (from the **Access keys** tab in your ACR resource)
`--registry-password`	Password (from the **Access keys** tab in your ACR resource)
`--registry-orgname`	Organization name. Use the same value as for the `-o` argument you specified during the uploading of your images to the ACR,
`--external-access-host`	DNS domain name configured earlier. For example, `installer.arcsight.private.com`. • If you use a different name for `private-dns` zone in the previous step, then change the value of `--external-access-host` to fit your DNS; that is, `installer.<YOUR NAME>` • You can verify this value on the Azure portal, in the main resource group, under **Private DNS Zone** resource.
`--cloud-provider`	Specifies the cloud provider when installing CDF on a cloud server. The allowed value of this parameter is AZURE.

For a complete list of optional parameters, see CDF Installation CLI Commands.

Patching the Load Balancer

Note: Before proceeding with annotating and patch the load balancer, execute the following command:

```
# kubectl get svc -A
```

In the resulting output, ensure that `iton-cdf-ingress-frontend-svc` has an IP address assigned.

If the command is processing for a long time, it indicates that Kubernetes is unable to create an internal load balancer and assign an IP to it. The usual cause is missing necessary rights. Refer to the Prerequisites section and make sure all prerequisites are met before proceeding any further.

To annotate and patch the load balancer:

1. On the jump host, run the following commands.

```
# kubectl annotate service -n core itom-cdf-ingress-frontend-svc
service.beta.kubernetes.io/azure-load-balancer-internal=true#
```

```
kubectl patch services itom-cdf-ingress-frontend-svc -p '{"spec":
{"type":"LoadBalancer","loadBalancerIP": "PUBLIC_IP"}}' -n core
```

Where `PUBLIC_IP` is the value of the public IP you assigned previously.

Configure the Kubernetes Cluster

After you install the CDF Installer, complete the following steps.

To configure your Kubernetes cluster:

1. RDP to the jumphost and browse to the cluster at your private DNS address at port 3000.

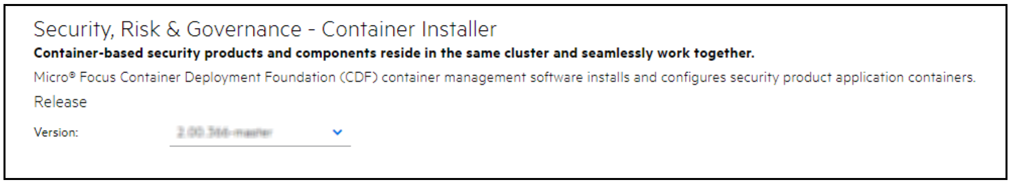
   For example:

   ```
   https://installer.private.arcsight.com:3000
   ```

2. Log in using *admin* userid and the password you specified during the CDF installation.
3. You will be prompted to upload the ArcSight installer metadata tar file:

   ```
   arcsight-installer-metadata-<version>.xx.tar
   ```

4. On the **Security Risk and Governance - Container Installer** page, choose the CDF base product metadata version.
5. Then, click **Next**.



Security, Risk & Governance - Container Installer

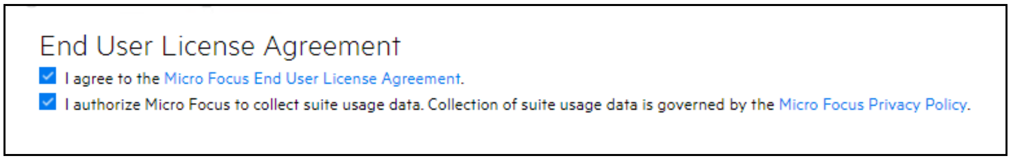
Container-based security products and components reside in the same cluster and seamlessly work together.

Micro® Focus Container Deployment Foundation (CDF) container management software installs and configures security product application containers.

Release

Version: 2.00.366-master

6. On the **End User License Agreement** page, review the EULA and select the *'I agree…'* checkbox. You might optionally choose to have suite utilization information passed to Micro Focus.
7. Then, click **Next**.



End User License Agreement

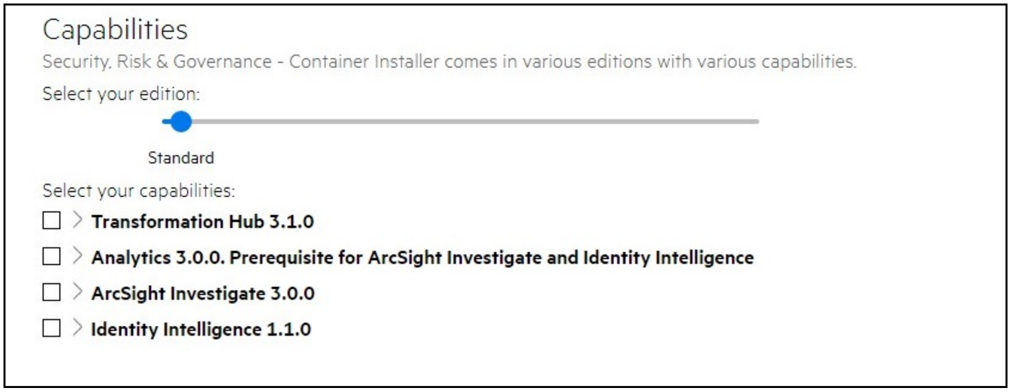
☑ I agree to the Micro Focus End User License Agreement.

☑ I authorize Micro Focus to collect suite usage data. Collection of suite usage data is governed by the Micro Focus Privacy Policy.

8. On the **Capabilities** page, choose the capabilities and/or products to be installed.
   a. To install Transformation Hub as a standalone install, make that your selection.

      > Other products might require Transformation Hub or other capabilities as prerequisites. Such requirements will be noted in the pull-down text associated with the capability.

   b. To show additional information associated with the product, click the **>** (greater than) arrow. When complete, click **Next.**



Capabilities

Security, Risk & Governance - Container Installer comes in various editions with various capabilities.

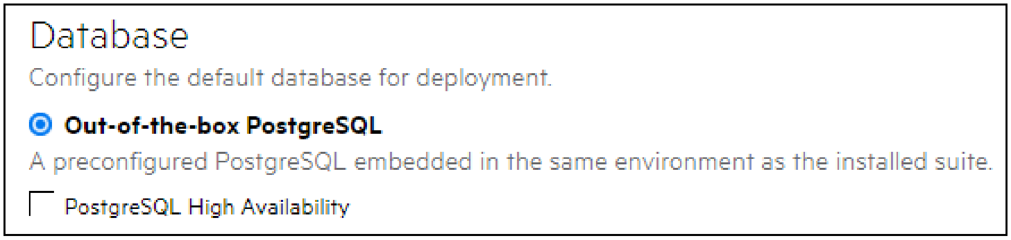
Select your edition:

Standard

Select your capabilities:

□ > Transformation Hub 3.1.0
□ > Analytics 3.0.0. Prerequisite for ArcSight Investigate and Identity Intelligence
□ > ArcSight Investigate 3.0.0
□ > Identity Intelligence 1.1.0

9. On the **Database** page, make sure the **PostgreSQL High Availability** box is *deselected*. This database is not used by capabilities in SODP.



Database

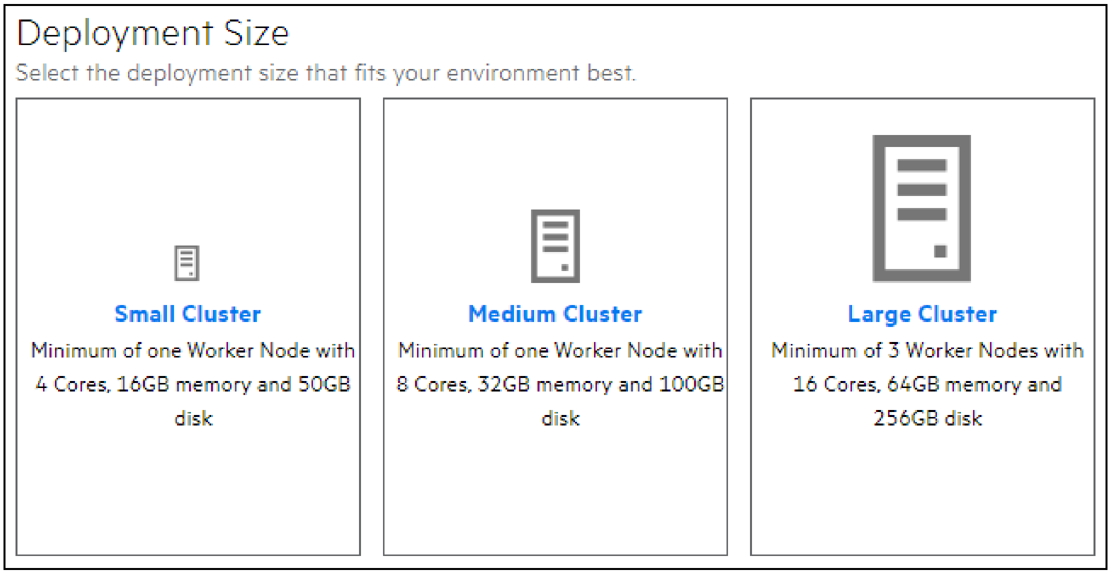
Configure the default database for deployment.

● Out-of-the-box PostgreSQL

A preconfigured PostgreSQL embedded in the same environment as the installed suite.

□ PostgreSQL High Availability

10. Click **Next.**
11. On the **Deployment Size** page, choose a size for your deployment based on your planned implementation.



Deployment Size

Select the deployment size that fits your environment best.

Small Cluster
Minimum of one Worker Node with 4 Cores, 16GB memory and 50GB disk

Medium Cluster
Minimum of one Worker Node with 8 Cores, 32GB memory and 100GB disk

Large Cluster
Minimum of 3 Worker Nodes with 16 Cores, 64GB memory and 256GB disk

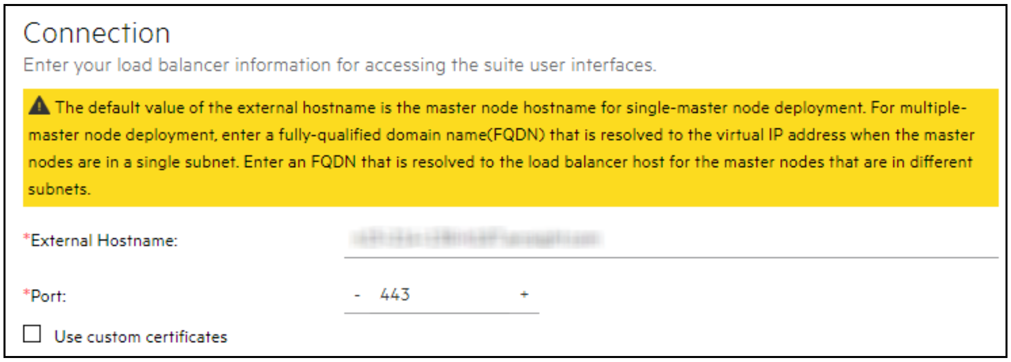
- **Small Cluster:** Minimum of one AKS node deployed (each node should have 4 cores, 16 GB memory, 50 GB disk)
- **Medium Cluster:** Minimum of 1 AKS node deployed (each node should have 8 cores, 32 GB memory, 100 GB disk)
- **Large Cluster:** Minimum of 3 AKS nodes deployed (each node should have 16 cores,64 GB memory, 256 GB disk)

> The installation will not proceed until the minimal hardware requirements for the deployment are met.

Additional nodes, with each running on their own host system, can be configured in subsequent steps.

Select your appropriate deployment size, and then click **Next**.

9. On the **Connection** page, an external hostname is automatically populated. This is resolved from the Virtual IP (VIP) specified earlier during the install of CDF (`--ha-virtual-ip parameter`), or the master node hostname if the `--ha-virtual-ip` parameter was not specified during CDF installation. Confirm the VIP is correct and then click **Next**.



Connection

Enter your load balancer information for accessing the suite user interfaces.

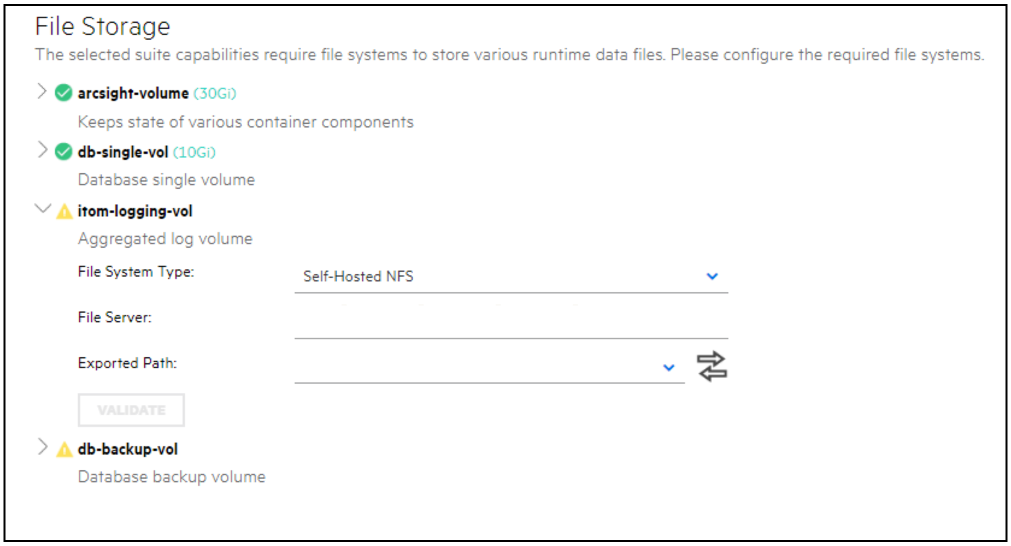
⚠ The default value of the external hostname is the master node hostname for single-master node deployment. For multiple-master node deployment, enter a fully-qualified domain name(FQDN) that is resolved to the virtual IP address when the master nodes are in a single subnet. Enter an FQDN that is resolved to the load balancer host for the master nodes that are in different subnets.

*External Hostname:	[redacted]
*Port:	- 443 +

☐ Use custom certificates

10. On the **File Storage** page, configure your NFS volumes. For each NFS volume, do the following:
    1. In **File Server**, enter the IP address or FQDN for the NFS server.
    2. On the **Exported Path** drop-down, select the appropriate volume.
    3. Click **Next**.

> All volumes must validate successfully to continue with the installation.



File Storage

The selected suite capabilities require file systems to store various runtime data files. Please configure the required file systems.

- arcsight-volume (30Gi)
  Keeps state of various container components
- db-single-vol (10Gi)
  Database single volume
- itom-logging-vol
  Aggregated log volume
  File System Type: Self-Hosted NFS
  File Server:
  Exported Path:
- db-backup-vol
  Database backup volume

> A *Self-hosted NFS* refers to the external NFS that you prepared when you configured an NFS server environment. Always choose this value for **File System Type.**

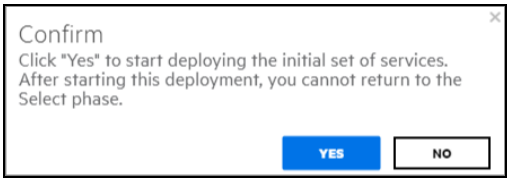
The following volumes must be available on your NFS server.

CDF NFS Volume claim	Your NFS volume
`arcsight-volume`	`<NFS_ROOT_FOLDER>/arcsight`
`itom-vol-claim`	`<NFS_ROOT_FOLDER>/itom_vol`
`db-single-vol`	`<NFS_ROOT_FOLDER>/db`
`itom-logging-vol`	`<NFS_ROOT_FOLDER>/logging`
`db-backup-vol`	`<NFS_ROOT_FOLDER>/db_backup`

11. Click **Validate**, and then click **Next**

> After you click **Next**, the infrastructure implementation will be deployed. *Please ensure that your infrastructure choices are adequate to your needs.* An incorrect or insufficient configuration might require a reinstall of all capabilities.

12. On the **Confirm** dialog, click **Yes** to start deploying nodes.



Confirm

Click "Yes" to start deploying the initial set of services. After starting this deployment, you cannot return to the Select phase.

Preparation Complete

Once all Nodes have been configured, and all services have been started on all nodes, the **Preparation Complete** page will be shown, meaning that the installation process is now ready to configure product-specific installation attributes.

Preparation Complete

The container deployment foundation is ready for use.

Click **Next** to configure the products and components of the deployment.

Configuring the Load Balancer

As part of load balancer configuration, to permit access to the 5443 port for TH deployment, the following needs to be added to the AKS load balancer:

- A health probe and load balancing rule for port 5443
- A health probe and load balancing rule for port 433

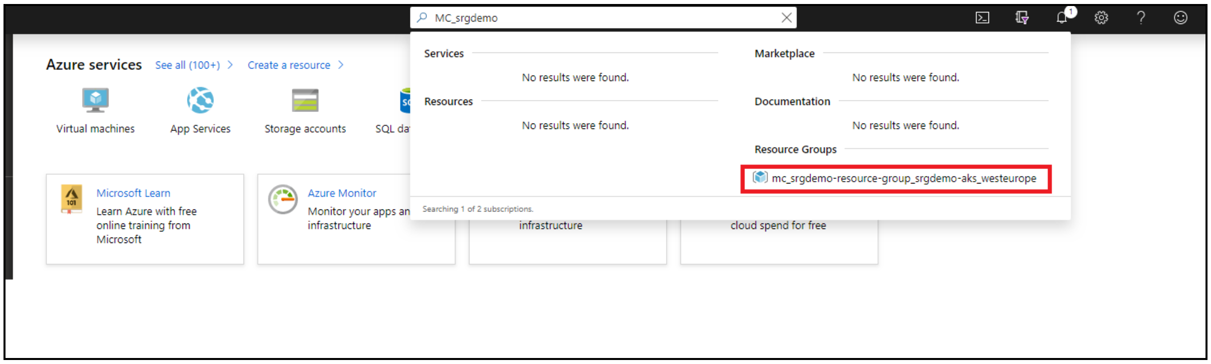
These steps are explained below.

To add a health probe for port 5443 using the Azure Portal:

1. On your jump host, run the following command to get the value of `nginx-ingress-controller-svc` for port 5443:
   ```
   kubectl get svc -n core | grep nginx-ingress-controller-svc
   ```

   > Example output, showing NodePort as 31249:
   > ```
   > nginx-ingress-controller-svc NodePort 10.0.146.63 5443:31249/TCP,5444:31036/TCP 21m
   > ```

2. Open the Azure Portal and locate the Azure Kubernetes resource group. (The AKS resource group name is in format MC_<your_resource_group>_<aks_name>_<location>.)



Services

No results were found.

Resources

No results were found.

Marketplace

No results were found.

Documentation

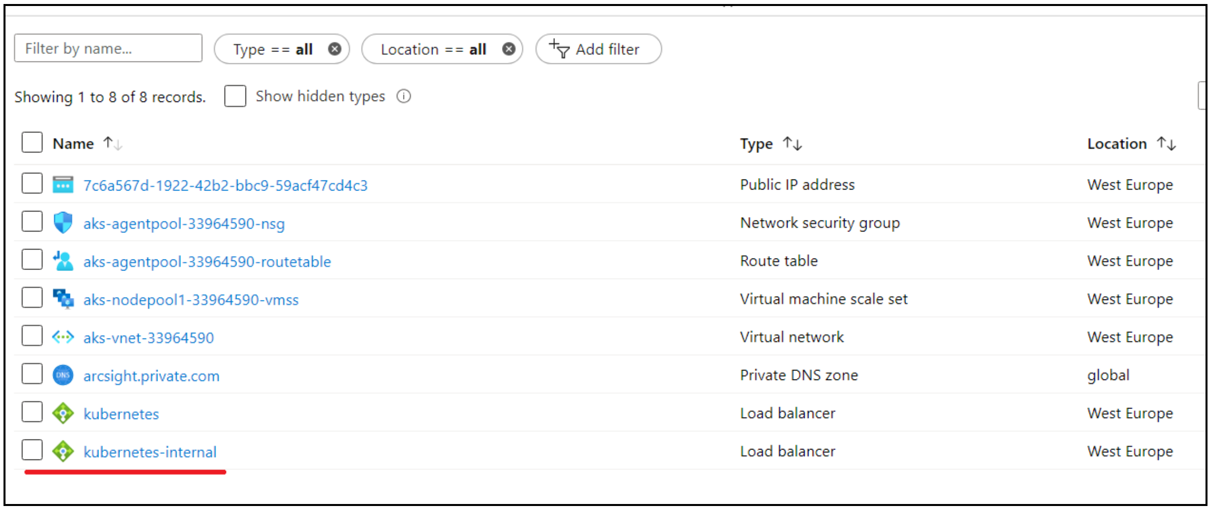
No results were found.

Resource Groups

mc_srgdemo-resource-group_srgdemo-aks_westeurope

Searching 1 of 2 subscriptions.

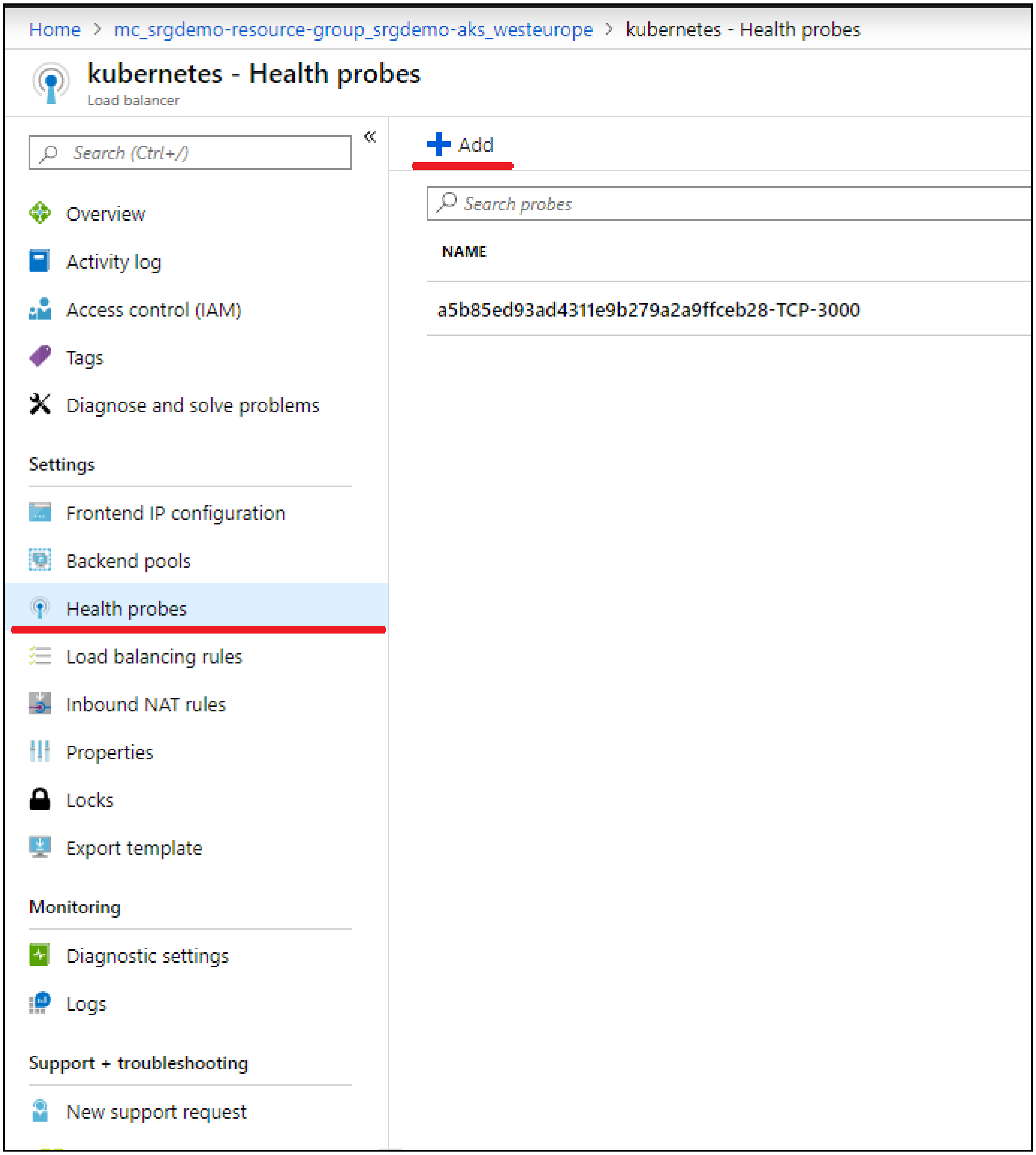
3. Open the Kubernetes resource group.
4. Find the Kubernetes load balancer and then open it.



Showing 1 to 8 of 8 records.

Name	Type	Location
7c6a567d-1922-42b2-bbc9-59acf47cd4c3	Public IP address	West Europe
aks-agentpool-33964590-nsg	Network security group	West Europe
aks-agentpool-33964590-routetable	Route table	West Europe
aks-nodepool1-33964590-vmss	Virtual machine scale set	West Europe
aks-vnet-33964590	Virtual network	West Europe
arcsight.private.com	Private DNS zone	global
kubernetes	Load balancer	West Europe
kubernetes-internal	Load balancer	West Europe

5. On the Kubernetes load balancer resource, click **Health probes.**
6. Add a health probe for 5443 using the value obtained for the service NodePort in step 1.



To add a health probe for port 5443 using the Azure Cloud Shell:

1. Get the AKS resource group and store it in an environment variable for later usage:

```
# CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group <RESOURCE GROUP> --name <AKS NAME> --query nodeResourceGroup -o tsv)
```

For example, for AKS `srg-demo-aks` from resource group `srg-demo`:

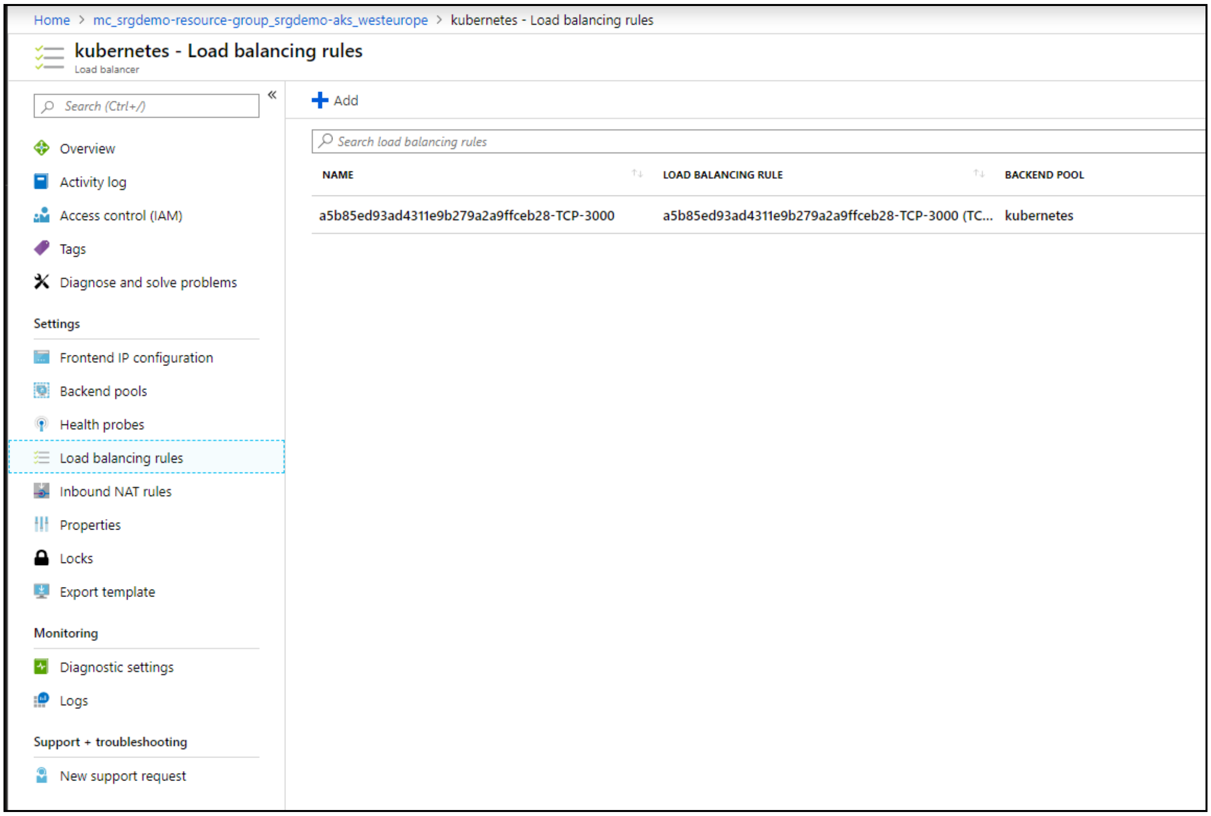
```
# CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group srg-demo --name srg-demo-aks --query nodeResourceGroup -o tsv)
```

2. Create the health probe by running the command:
   ```
   # az network lb probe create -g $CLUSTER_RESOURCE_GROUP --lb-name
   kubernetes-internal -n 5443-hp --protocol tcp --port <SERVICE PORT>
   ```

> Example:
> ```
> # az network lb probe create -g $CLUSTER_ RESOURCE_ GROUP --lb- name
> kubernetes-internal -n 5443-hp --protocol tcp --port 31249
> ```

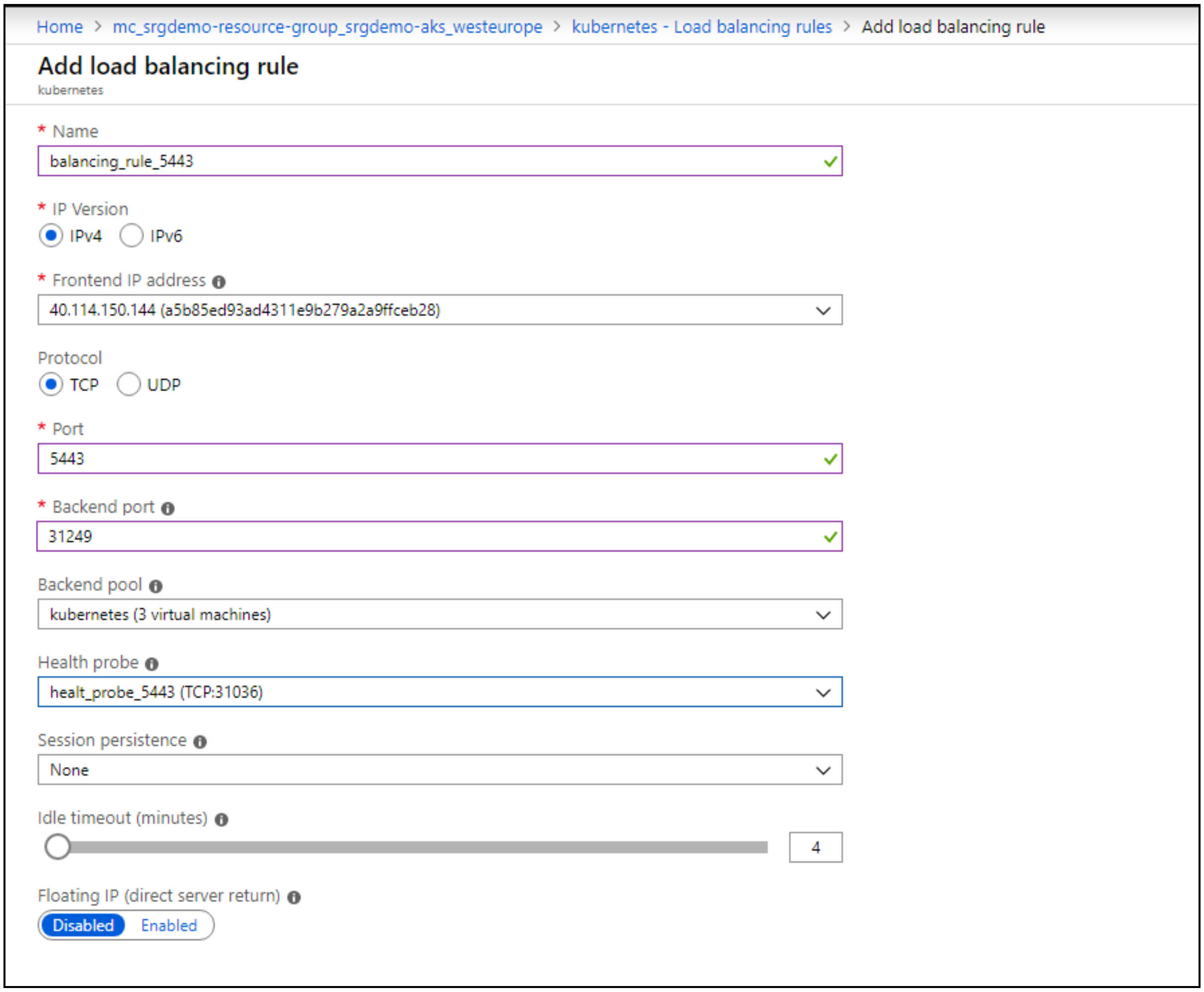
To add a load balancing rule for port 5443 using the Azure Portal:

1. Open the Kubernetes load balancer and click **Load balancing rules.**



NAME	LOAD BALANCING RULE	BACKEND POOL
a5b85ed93ad4311e9b279a2a9ffceb28-TCP-3000	a5b85ed93ad4311e9b279a2a9ffceb28-TCP-3000 (TC...	kubernetes

2. Add a rule for port 5443. The backend port is the value for `nginx-ingress-controller-svc` obtained previously and the health probe you just created.



Home > mc_srgdemo-resource-group_srgdemo-aks_westeurope > kubernetes - Load balancing rules > Add load balancing rule

Add load balancing rule

kubernetes

* Name

balancing_rule_5443 ✓

* IP Version

● IPv4 ○ IPv6

* Frontend IP address

40.114.150.144 (a5b85ed93ad4311e9b279a2a9ffceb28)

Protocol

● TCP ○ UDP

* Port

5443 ✓

* Backend port

31249 ✓

Backend pool

kubernetes (3 virtual machines)

Health probe

healt_probe_5443 (TCP:31036)

Session persistence

None

Idle timeout (minutes)

4

Floating IP (direct server return)

[✓ Disabled] Enabled

To add a load balancing rule for port 5443 using the Azure Cloud Shell:

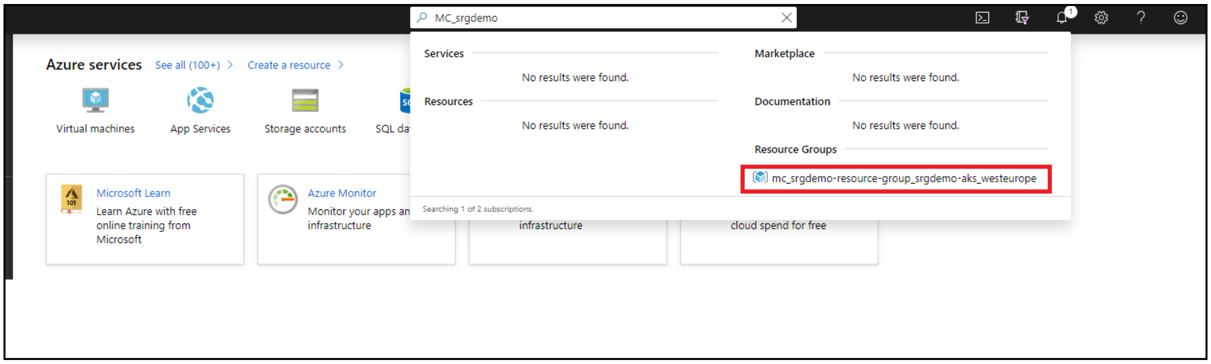
1. Run the following command:
   ```
   # az network lb rule create -g <AKS RESOURCE GROUP> --lb-name kubernetes-internal -n 5443-lb-rule --protocol Tcp --frontend-port 5443 --backend-port <SERVICE PORT> --probe-name 5443-hp --backend-pool-name kubernetes
   ```

For example:

```
# az network lb rule create -g mc_srg-demo_srg-demo-aks_westeurope --lb-name kubernetes-internal -n 5443-lb-rule --protocol Tcp --frontend-port 5443 --backend-port 31249 --probe-name 5443-hp --backend-pool-name kubernetes
```

To add a health probe for port 443 using the Azure Portal:

1. In the Azure portal, locate the Azure Kubernetes resource group. (The AKS resource group name is in format MC_<your_resource_group>_<aks_name>_<location>.)



Learn Azure with free online training from Microsoft

Monitor your apps an infrastructure

infrastructure

cloud spend for free

No results were found.

No results were found.

No results were found.

No results were found.

Searching 1 of 2 subscriptions.

2. Open the Kubernetes resource group.
3. On the Kubernetes load balancer resource, click **Health probes.**
4. Click **+ Add** for Kubernetes load balancer health probes and enter values for the following:
   - **Name:** assign a name to the probe.
   - **Protocol:** select TCP
   - **Port:** enter 443

To add a health probe for port 443 using the Azure Cloud Shell:

1. Run the following command:

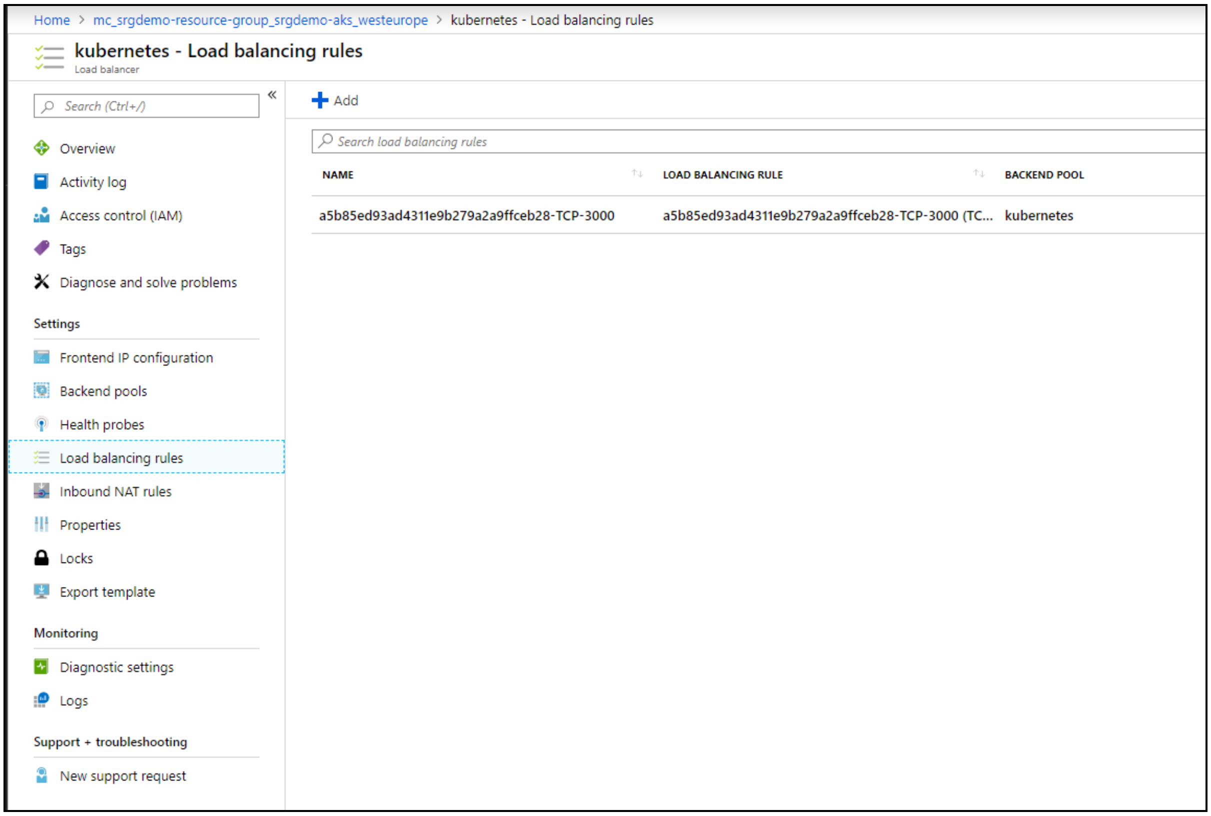
```
# az network lb probe create -g <AKS RESOURCE GROUP> --lb-name kubernetes-internal -n 443-hp --protocol tcp --port 443
```

For example:

```
# az network lb probe create -g mc_srg-demo_srg-demo-aks_westeurope --lb-name kubernetes-internal -n 443-hp --protocol tcp --port 443
```

To add a load balancing rule for port 443 using the Azure Portal:

1. Open the Kubernetes load balancing rule and click Load balancing rules.
2. Click **+ Add** for the Kubernetes load balancer load balancing rules and enter values for the following:
   - **Name:** assign a name to the probe.
   - **Port:** enter 443.
   - **Backend port:** enter 443.
   - **Health probe:** select the probe you previously created for port 443.
3. Open the Kubernetes resource group.


Home > mc_srgdemo-resource-group_srgdemo-aks_westeurope > kubernetes - Load balancing rules
kubernetes - Load balancing rules
Load balancer
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Frontend IP configuration
Backend pools
Health probes
Load balancing rules
Inbound NAT rules
Properties
Locks
Export template
Monitoring
Diagnostic settings
Logs
Support + troubleshooting
New support request
Add
Search load balancing rules

NAME	LOAD BALANCING RULE	BACKEND POOL
a5b85ed93ad4311e9b279a2a9ffceb28-TCP-3000	a5b85ed93ad4311e9b279a2a9ffceb28-TCP-3000 (TC...	kubernetes

To add a load balancing rule for port 443 using the Azure Cloud Shell:

1. Run the following command:

   ```
   # az network lb rule create -g <AKS RESOURCE GROUP> --lb-name kubernetes-internal -n 443-lb-rule --protocol Tcp --frontend-port 443 --backend-port 443 --probe-name 443-hp --backend-pool-name kubernetes
   ```

For example:

```
# az network lb rule create -g mc_srg-demo_srg-demo-aks_westeurope --lb-name kubernetes-internal -n 443-lb-rule --protocol Tcp --frontend-port 443 --backend-port 443 --probe-name 443-hp --backend-pool-name kubernetes
```

Opening the Management Portal

To open the CDF Management Portal:

1. RDP to the jump host desktop and open a browser.
2. Browse to :
   `https://<DNS_name>:5443.`

> Use the DNS name you configured earlier, such as `installer.private.access.com`.

Setting Up Your Deployment Architecture (Amazon Web Services)

This section explains how to set up your deployment architecture for a Transformation Hub that runs on the Amazon Web Services (AWS) cloud platform.

Understanding the Deployment Overview

The complete process of deploying Transformation Hub on AWS includes the following broad steps:

1. **Review prerequisites:** Review the technical prerequisites and ensure that they are met before beginning the installation.
2. **Create the Virtual Private Cloud (VPC):** Create and configure the AWS Virtual Private Cloud, including security groups and IAM roles.
3. **Configure Bastion:** Prepare the bastion host, which you will use for access to the AWS deployment environment.
4. **Download the installation packages and tools:** Download the required installation files and associated tools.
5. **Configure the Elastic File System (EFS):**Prepare the EFS instance used for the AWS deployment environment.
6. **Configure Elastic Kubernetes System (EKS):** Prepare the EKS control plane.
7. **Configure Worker Nodes:** Create and label the worker nodes, where application processing takes place.
8. **Upload Images to Elastic Container Registry:** Transfer the product images to the ECR.
9. **Configure Routing:** Prepare the Route 53 DNS routing.
10. **Bootstrap CDF:** Install CDF rudiments so that you can perform a complete installation after load balancer configuration.
11. **Configure Application Load Balancer (ALB):** Prepare the application load balancer.
12. **Install CDF:**Install the remaining CDF components and deploy the ArcSight Suite products.
13. **Post-installation configuration:** Configure access to the CDF management portal and access to re-configuration.

Each of these steps is explained in the following sections. Most steps can be performed using either the AWS web UI or through the AWS CLI tool, and each method is explained (where possible).

Next Step: The AWS worksheet

The AWS Deployment Worksheet

The process of setting up an AWS deployment environment will require configuration of many AWS resources. As a result, you will need convenient access to important details of these resources, such as resource names, IP addresses, settings for AWS entities, and so on, which you will determine during the setup process.

For ease of reference, it's strongly recommended that you print out and use the AWS worksheet to record the details of your configuration. The procedures given here assume you will be using the worksheet for reference and will note when particular details should be recorded.

Next Step: Verify Prerequisites

AWS Transformation Hub Deployment Prerequisites

In order to perform the installation of deploy Transformation Hub on AWS, the user requires an active AWS subscription, as well as the following:

IAM Policies

Installation of ArcSight Suite is performed under the local IAM user. If you do not have a local IAM user, ask your AWS administrator to create a user for you and assign these required IAM policies:

- ARST_BYOK_CustomPolicy
- ARST_RestrictedAPIs

Minimal Permissions

Access to various AWS resources is controlled by permissions assigned to the IAM user. For easier management, you can create a policy holding the minimal set of permissions required to complete tasks in this guide. The policy must contain the following permissions.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "route53:*",
            "iam:AddRoleToInstanceProfile",
            "iam:AttachRolePolicy",
            "iam:CreateAccessKey",
            "iam:CreateInstanceProfile",
            "iam:CreatePolicy",
            "iam:CreateRole",
            "iam:DeleteAccessKey",
            "iam:DeleteInstanceProfile",
            "iam:DeletePolicy",
            "iam:DeleteRole",
            "iam:DeleteRolePolicy",
            "iam:DetachRolePolicy",
            "iam:GenerateServiceLastAccessedDetails",
            "iam:GetAccessKeyLastUsed",
            "iam:GetAccountSummary",
            "iam:GetLoginProfile",
            "iam:GetPolicyVersion",
            "iam:GetRole",
            "iam:GetRolePolicy",
            "iam:GetServiceLastAccessedDetails",
            "iam:GetServiceLastAccessedDetailsWithEntities",
            "iam:GetUser",
            "iam:ListAccessKeys",
            "iam:ListAccountAliases",
            "iam:ListAttachedGroupPolicies",
            "iam:ListAttachedRolePolicies",
            "iam:ListAttachedUserPolicies",
            "iam:ListEntitiesForPolicy",
            "iam:ListGroupPolicies",
            "iam:ListGroups",
            "iam:ListGroupsForUser",
            "iam:ListInstanceProfiles",
            "iam:ListInstanceProfilesForRole",
            "iam:ListMFADevices",
            "iam:ListOpenIDConnectProviders",
            "iam:ListPolicies",
            "iam:ListPoliciesGrantingServiceAccess",
            "iam:ListPolicyVersions",
            "iam:ListRolePolicies",
            "iam:ListRoleTags",
            "iam:ListRoles",
            "iam:ListSAMLProviders",
            "iam:ListSSHPublicKeys",
            "iam:ListServerCertificates",
            "iam:ListServiceSpecificCredentials",
            "iam:ListSigningCertificates",
            "iam:ListUserPolicies",
            "iam:ListUserTags",
            "iam:ListUsers",
            "iam:ListVirtualMFADevices",
            "iam:PassRole",
            "iam:PutRolePolicy",
            "iam:RemoveRoleFromInstanceProfile",
            "iam:TagRole",
            "iam:TagUser",
            "iam:UntagRole",
            "iam:UntagUser",
            "iam:UpdateAccessKey",
            "iam:UpdateLoginProfile"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "acm:*",
            "autoscaling:*",
            "cloudformation:*",
            "ec2:*",
            "ecr:*",
            "eks:*",
            "elasticfilesystem:*",
            "elasticloadbalancing:*",
            "s3:CreateBucket",
            "s3:DeleteObject",
            "s3:GetObject",
            "s3:PutObject",
            "sns:ListSubscriptions",
            "sns:ListTopics",
            "ssm:DescribeActivations",
            "ssm:DescribeAssociation",
            "ssm:DescribeAssociationExecutionTargets",
            "ssm:DescribeAssociationExecutions",
            "ssm:DescribeAutomationExecutions",
            "ssm:DescribeAutomationStepExecutions",
            "ssm:DescribeAvailablePatches",
            "ssm:DescribeDocument",
            "ssm:DescribeDocumentParameters",
            "ssm:DescribeDocumentPermission",
            "ssm:DescribeEffectiveInstanceAssociations",
            "ssm:DescribeEffectivePatchesForPatchBaseline",
            "ssm:DescribeInstanceAssociationsStatus",
            "ssm:DescribeInstanceInformation",
            "ssm:DescribeInstancePatchStates",
            "ssm:DescribeInstancePatchStatesForPatchGroup",
            "ssm:DescribeInstancePatches",
            "ssm:DescribeInstanceProperties",
            "ssm:DescribeInventoryDeletions",
            "ssm:DescribeMaintenanceWindowExecutionTaskInvocations",
            "ssm:DescribeMaintenanceWindowExecutionTasks",
            "ssm:DescribeMaintenanceWindowExecutions",
            "ssm:DescribeMaintenanceWindowSchedule",
            "ssm:DescribeMaintenanceWindowTargets",
            "ssm:DescribeMaintenanceWindowTasks",
            "ssm:DescribeMaintenanceWindows",
            "ssm:DescribeMaintenanceWindowsForTarget",
            "ssm:DescribeOpsItems",
            "ssm:DescribeParameters",
            "ssm:DescribePatchBaselines",
            "ssm:DescribePatchGroupState",
            "ssm:DescribePatchGroups",
            "ssm:DescribePatchProperties",
            "ssm:DescribeSessions",
            "ssm:GetAutomationExecution",
            "ssm:GetCommandInvocation",
            "ssm:GetConnectionStatus",
            "ssm:GetDefaultPatchBaseline",
            "ssm:GetDeployablePatchSnapshotForInstance",
            "ssm:GetDocument",
            "ssm:GetInventory",
            "ssm:GetInventorySchema",
            "ssm:GetMaintenanceWindow",
            "ssm:GetMaintenanceWindowExecution",
            "ssm:GetMaintenanceWindowExecutionTask",
            "ssm:GetMaintenanceWindowExecutionTaskInvocation",
            "ssm:GetMaintenanceWindowTask",
            "ssm:GetManifest",
            "ssm:GetOpsItem",
            "ssm:GetOpsSummary",
            "ssm:GetParameter",
            "ssm:GetParameterHistory",
            "ssm:GetParameters",
            "ssm:GetParametersByPath",
            "ssm:GetPatchBaseline",
            "ssm:GetPatchBaselineForPatchGroup",
            "ssm:GetServiceSetting",
            "ssm:ListAssociationVersions",
            "ssm:ListAssociations",
            "ssm:ListCommandInvocations",
            "ssm:ListCommands",
            "ssm:ListComplianceItems",
            "ssm:ListComplianceSummaries",
            "ssm:ListDocumentVersions",
            "ssm:ListDocuments",
            "ssm:ListInstanceAssociations",
            "ssm:ListInventoryEntries",
            "ssm:ListResourceComplianceSummaries",
            "ssm:ListResourceDataSync",
            "ssm:ListTagsForResource",
            "ssm:PutConfigurePackageResult"
         ],
         "Resource":"*"
      }
   ]
}
```

Tools

The `AWS CLI` (v2) and `jq` tools must be installed on the local host.

`AWS CLI` is a unified tool to manage AWS services. If it is not already installed, then install and configure the `AWS CLI` (version 2) tool for your platform. All references to CLI in this guide refer to the `AWS CLI` version 2 interface.

- Amazon provides the instructions for installing AWS CLI.
- After installation, configure the `AWS CLI` to properly authenticate and connect to AWS as described in Configuring AWS CLI.

Most procedures for configuring AWS are supplied in both the AWS CLI and web UI versions.

`jq` is a lightweight and flexible open-source command-line JSON processor.

- You can download the `jq` binaries from the `jq` homepage.

Host Requirements

You can configure and use any local host which has Internet access for the initial steps in setting up your deployment environment. Later, you will create a bastion instance, and use the bastion to perform the installation, as well as to access the cluster after installation.

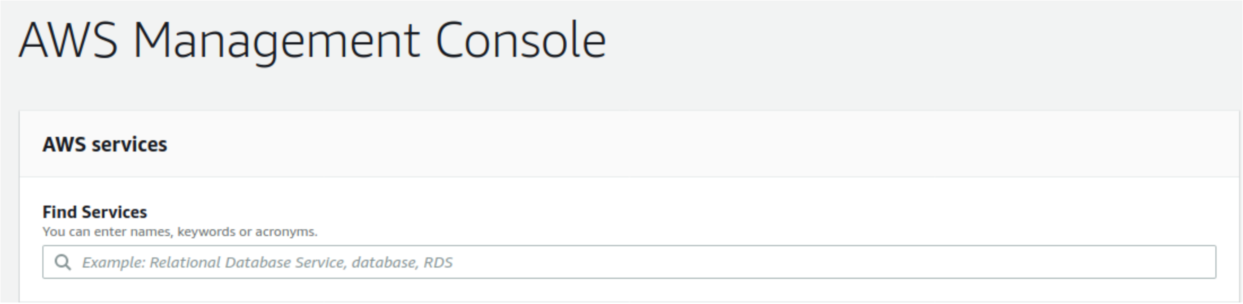
Next Step: Configuring AWS CLI

Configuring AWS CLI

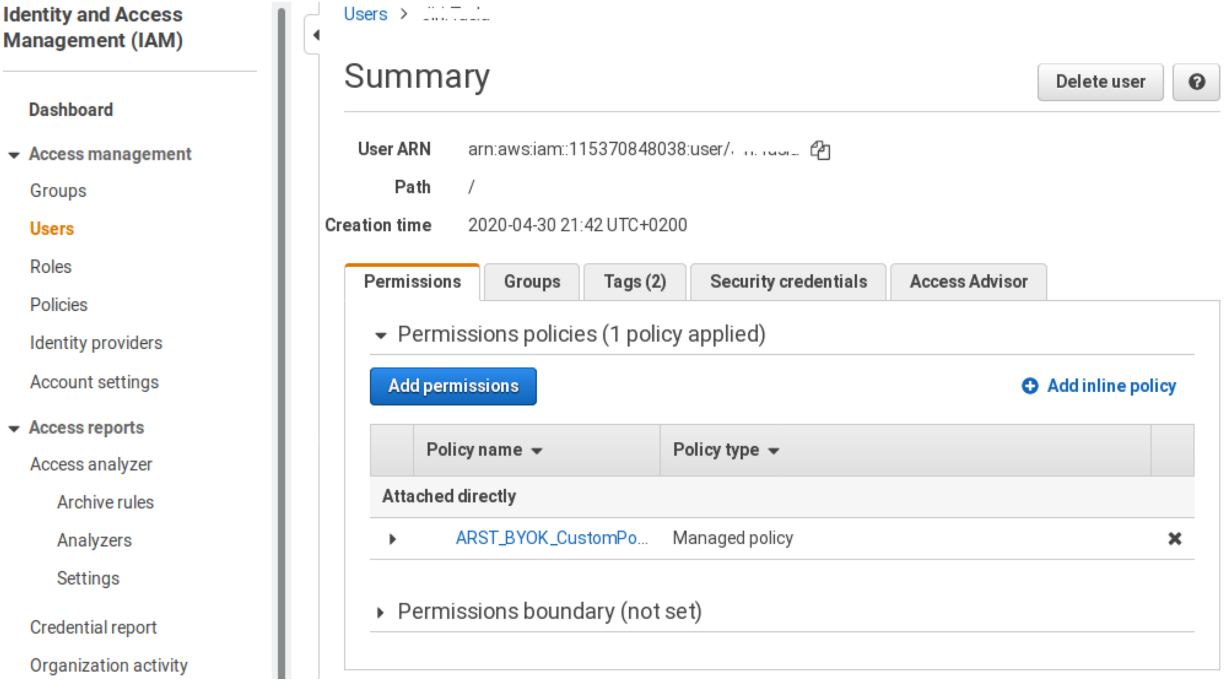
Follow these steps to configure AWS CLI.

First, create and retrieve the AWS access keys.

1. Log in to the AWS Management Console using your IAM account.


AWS Management Console
AWS services
Find Services
You can enter names, keywords or acronyms.
Example: Relational Database Service, database, RDS

2. Using the Find Services search box, browse to and open the IAM dashboard.
3. In the left navigation panel, choose **Users**.
4. Locate and open the user with which you are performing the installation.



User ARN arn:aws:iam::115370848038:user/[redacted]

Path /

Creation time 2020-04-30 21:42 UTC+0200

Policy name	Policy type
Attached directly	
ARST_BYOK_CustomPo...	Managed policy

5. Select the user in IAM and display the user summary.
6. On the **Security Credentials** tab, click **Create Access Key.**
7. Record both the Access key ID and secret access key, and download the `.csv` file for later use.

To configure the AWS CLI:

1. Launch the AWS CLI tool.
2. Run the following command:

```
# aws configure
```

3. Enter the following information:
   - **AWS Access Key ID:** Enter the access key ID that you previously recorded.
   - **AWS Secret Access Key:** Enter the secret access key that you previously recorded, or copy and paste the contents from the CSV file that you downloaded earlier.
   - **Default region name:** Enter the region in which you are installing the ArcSight Suite. If you are unsure, verify your current region from the information in the top right corner of AWS console. Click on the city name and find the region name in the list. For example, *eu-central-1.*]

     > Not all AWS regions support all required services. Your selected region must support the Elastic Container Service (ECS) and Elastic Kubernetes Service (EKS).

   - **Default output format:** Enter *json*.

Example input:

```
# aws configure
```

```
AWS Access Key ID [***************VPMC]: AKIA************VPMC↲
```

```
AWS Secret Access Key [***************wFap]: eZO**********************wFap↲
```

```
Default region name [eu-central-1]: eu-central-1↲
```

```
Default output format [json]: json↲
```

4. To verify configuration, in the AWS Cloud API, enter the command to view the default VPC description:

   ```
   # aws ec2 describe-vpcs
   ```

Example output:

```
{
    "Vpcs": [
        {
            "CidrBlock": "172.31.0.0/16",
            "DhcpOptionsId": "dopt-3a1efe53",
            "State": "available",
            "VpcId": "vpc-a71cfcce",
            "OwnerId": "115370848038",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-13d7337a",
                    "CidrBlock": "172.31.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
```

```
                    }
                }
            ],
            "IsDefault": true
        }
    ]
}
```

Next Step: Create the Virtual Private Cloud

Creating the VPC

1. In the AWS CLI, run the following command:
   ```
   # aws ec2 create-vpc \
   --cidr-block <CIDR allocated for new VPC> \
   | jq -r '.Vpc.VpcId'
   ```

The command will return the new VPC's VPC ID. Record the VPC ID and VPC CIDRto the AWS worksheet.

Sample input and output:

```
# aws ec2 create-vpc \
--cidr-block 10.0.0.0/16 \
| jq -r '.Vpc.VpcId'
```

```
vpc-0143197ca9bd9c117
```

Next Step: Add Tags to the VPC

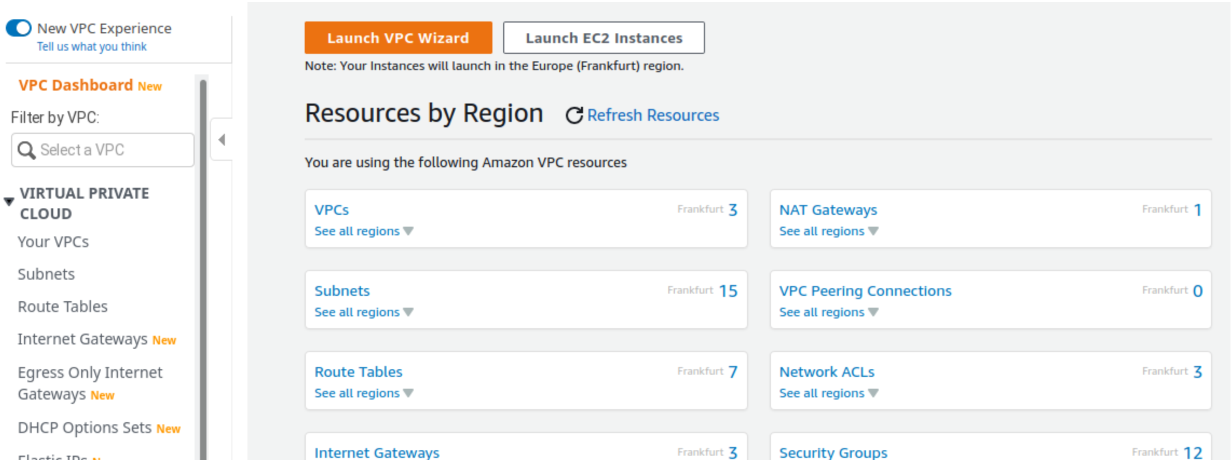
Tagging the VPC

The new VPC is required to have the following identifying tags:

- `Name=<vpc name>`: Name of the VPC, for easier identification.
- `kubernetes.io/cluster/<cluster name>=shared`: Cluster name required so that Kubernetes can join worker nodes to the cluster. (The EKS cluster must also be tagged with this tag later.)

To add tags to the VPC using the Web UI:

1. Using the Find Services search box, browse to the VPC Dashboard.



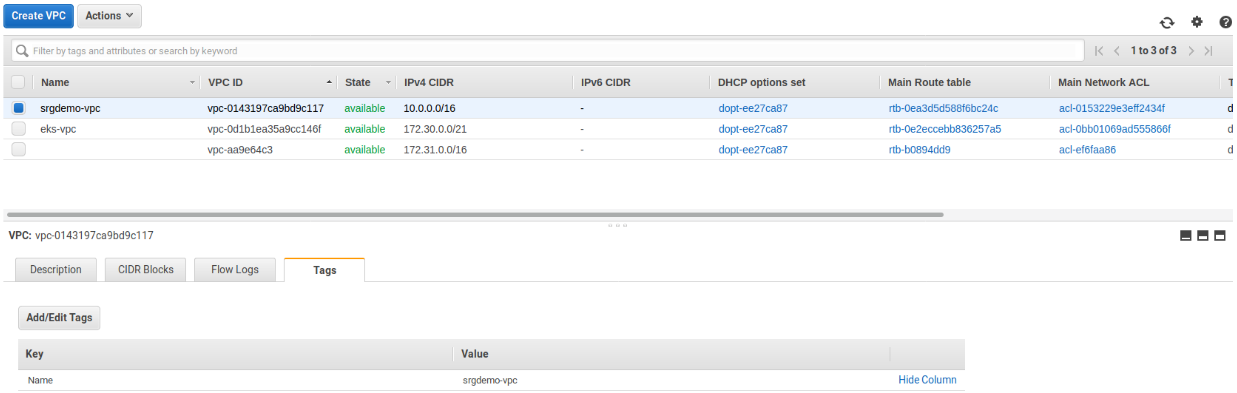
Note: Your Instances will launch in the Europe (Frankfurt) region.

Resources by Region

You are using the following Amazon VPC resources

VPCs	Frankfurt 3	NAT Gateways	Frankfurt 1
Subnets	Frankfurt 15	VPC Peering Connections	Frankfurt 0
Route Tables	Frankfurt 7	Network ACLs	Frankfurt 3
Internet Gateways	Frankfurt 3	Security Groups	Frankfurt 12

2. In the left navigation panel, under **Virtual Private Cloud,** click **Your VPCs**.

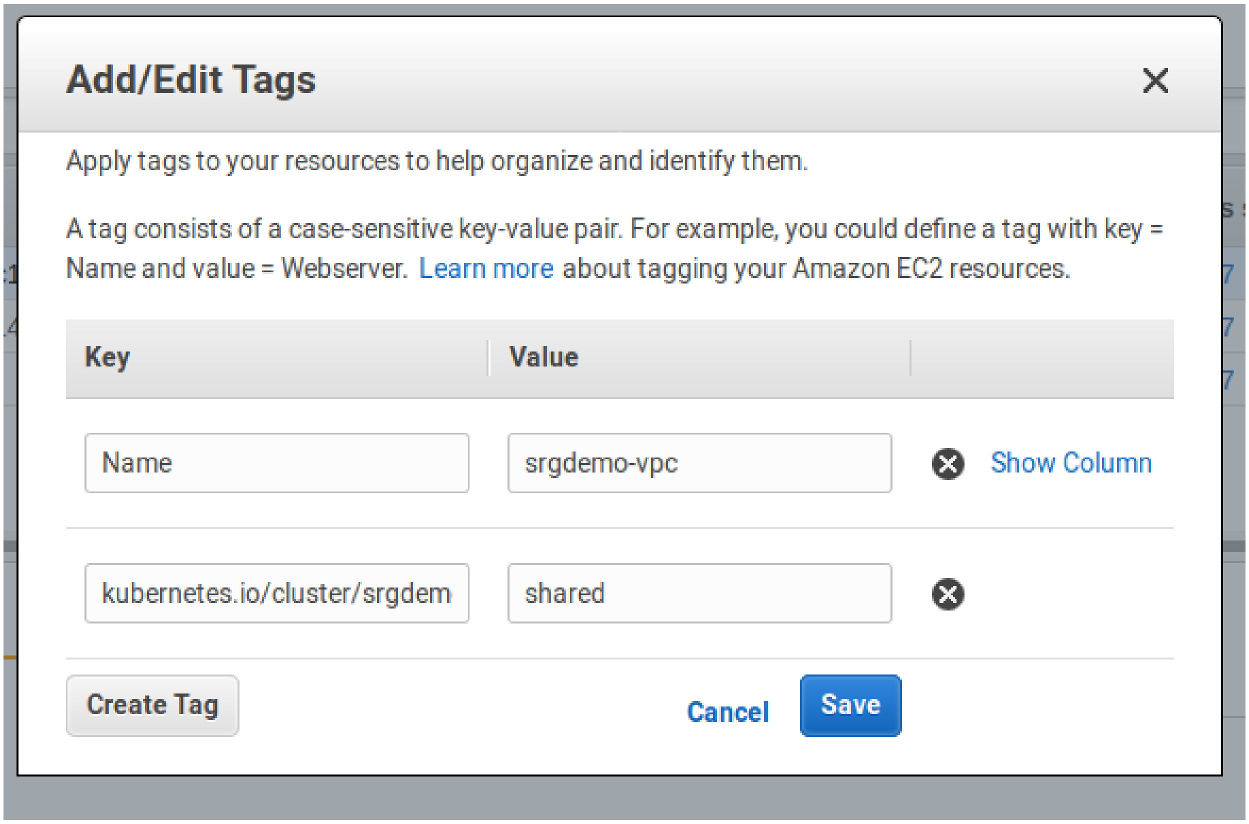


	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL	T
☑	srgdemo-vpc	vpc-0143197ca9bd9c117	available	10.0.0.0/16	-	dopt-ee27ca87	rtb-0ea3d5d588f6bc24c	acl-0153229e3eff2434f	d
□	eks-vpc	vpc-0d1b1ea35a9cc146f	available	172.30.0.0/21	-	dopt-ee27ca87	rtb-0e2eccebb836257a5	acl-0bb01069ad555866f	d
□		vpc-aa9e64c3	available	172.31.0.0/16	-	dopt-ee27ca87	rtb-b0894dd9	acl-ef6faa86	d

VPC: vpc-0143197ca9bd9c117

Key	Value
Name	srgdemo-vpc

3. On the VPC management page, select your VPC either by name or VPC ID.
4. At the bottom, select the **Tags** tab.
5. On the tag editor dialog, click **Create Tag** and then enter the key name `Name` and value of the VPC name for the `Name` , as described above. Click **Save**.



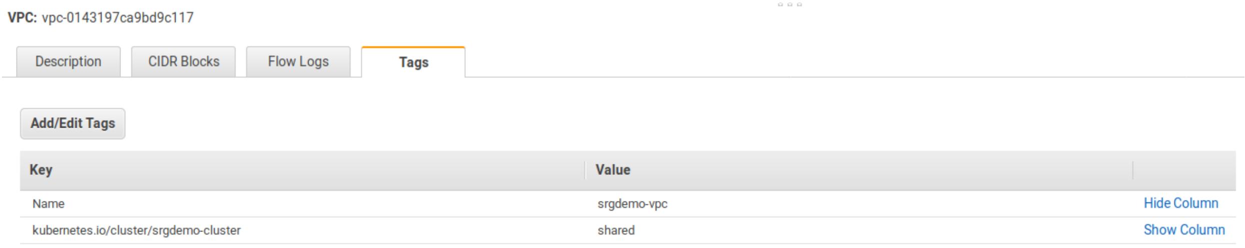
Add/Edit Tags

Apply tags to your resources to help organize and identify them.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key	Value	
Name	srgdemo-vpc	Show Column
kubernetes.io/cluster/srgdem	shared	

6. Click **Create Tag**, and then enter the tag for `kubernetes.io/cluster/<cluster name>` and the value `shared`.
7. Click **Save.**
8. The list of VPC tags is shown on the **Tags** tab.



VPC: vpc-0143197ca9bd9c117

Key	Value	
Name	srgdemo-vpc	Hide Column
kubernetes.io/cluster/srgdemo-cluster	shared	Show Column

To add tags to the VPC using the AWS CLI:

1. Run the following command:

```
# aws ec2 create-tags \
--resources <VpcId> \
--tags Key=Name,Value=<vpc name> Key=kubernetes.io/cluster/<cluster
name>,Value=shared
```

Parameters:

`<VpcId>`: The VPC ID.

`<vpc name>`: Assists in easier identification in the list.

`<cluster name`: Choose a name for your cluster and record it in the AWS worksheet. This value will be used later.

> This command has no output.

Example:

```
# aws ec2 create-tags \
--resources vpc-0143197ca9bd9c117 \
--tags Key=Name,Value=srgdemo-vpc Key=kubernetes.io/cluster/srgdemo-
cluster,Value=shared
```

To (optionally) verify assigned tags:

1. Run the command:
   ```
   # aws ec2 describe-tags \
   --filters "Name=resource-id,Values=<VPC ID>"
   ```

For example:

```
# aws ec2 describe-tags \
--filters "Name=resource-id,Values=vpc-0143197ca9bd9c117"
```

```
{
   "Tags":[
      {
         "Key":"Name",
         "ResourceId":"vpc-0143197ca9bd9c117",
         "ResourceType":"vpc",
         "Value":"srgdemo-vpc"
      },
      {
         "Key":"kubernetes.io/cluster/srgdemo-cluster",
         "ResourceId":"vpc-0143197ca9bd9c117",
         "ResourceType":"vpc",
         "Value":"shared"
      }
   ]
}
```

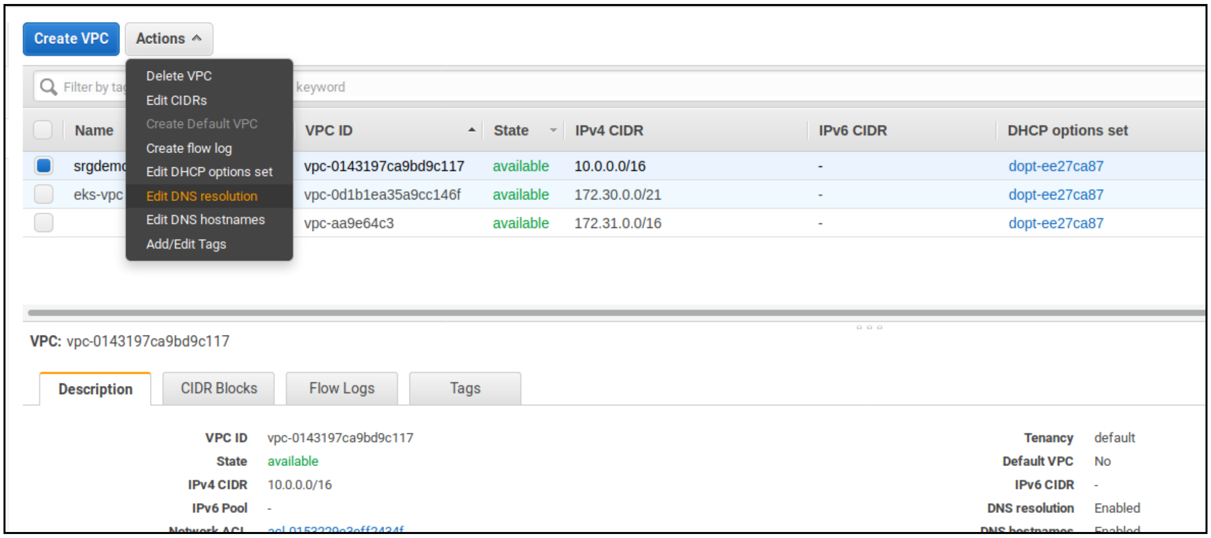
Next Step: Enable DNS

Enable DNS and Hostname Resolution

DNS support and hostname resolution should be enabled to make IP addresses more easily human-readable.

To enable DNS using the web UI:

1. Using the Find Services search tool, locate and browse to the VPC dashboard.
2. On the left navigation panel, under **Virtual Private Cloud,** click **Your VPCs.**
3. Select the checkbox corresponding to your VPC. Then, under **Actions**, select **Edit DNS resolution**.

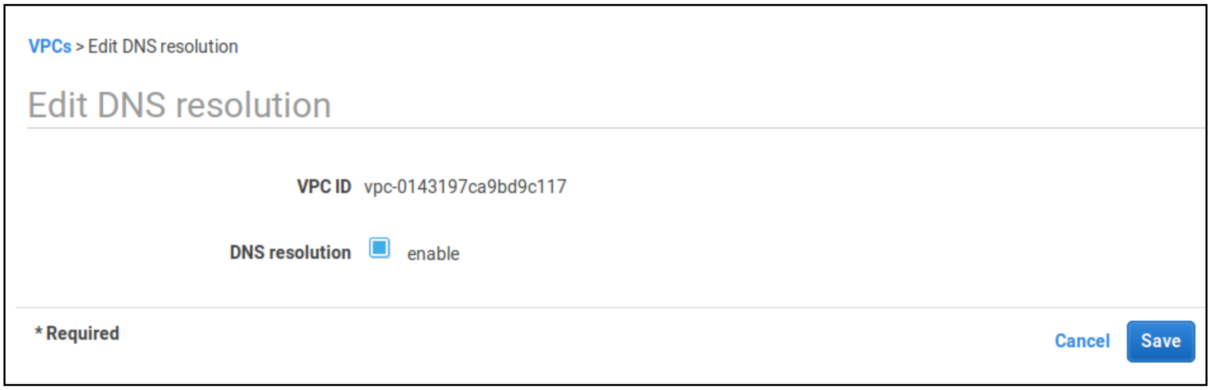


	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set
☑	srgdem	vpc-0143197ca9bd9c117	available	10.0.0.0/16	-	dopt-ee27ca87
□	eks-vpc	vpc-0d1b1ea35a9cc146f	available	172.30.0.0/21	-	dopt-ee27ca87
□		vpc-aa9e64c3	available	172.31.0.0/16	-	dopt-ee27ca87

VPC: vpc-0143197ca9bd9c117

VPC ID	vpc-0143197ca9bd9c117	Tenancy	default
State	available	Default VPC	No
IPv4 CIDR	10.0.0.0/16	IPv6 CIDR	-
IPv6 Pool	-	DNS resolution	Enabled
Network ACL	acl-0153229e3eff2434f	DNS hostnames	Enabled

4. On the **Edit DNS Resolution** page, for **DNS resolution**, select the **enable** checkbox.



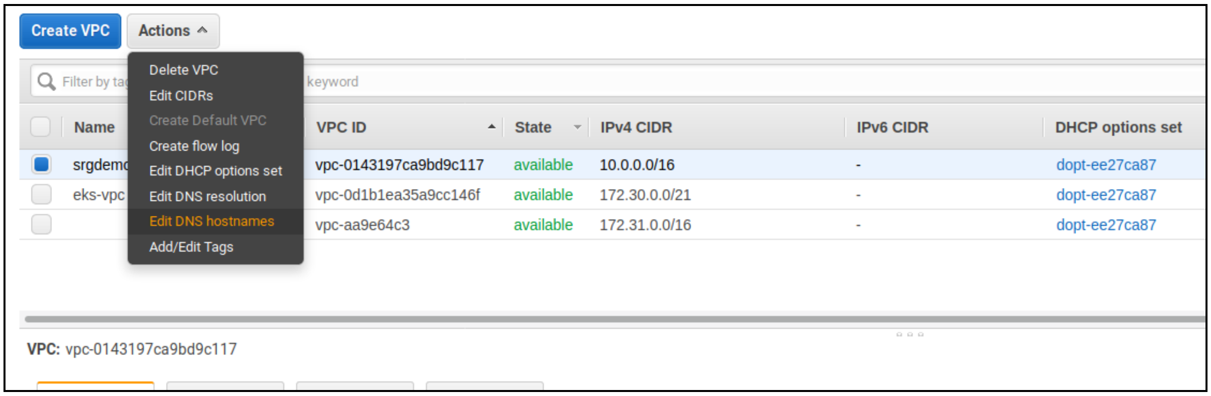
VPC ID vpc-0143197ca9bd9c117

DNS resolution ☑ enable

5. Click **Save**, then click **Close**.

To enable hostname resolution using the web UI:

1. Using the Find Services search tool, locate and browse to the VPC dashboard.
2. On the left navigation panel, under **Virtual Private Cloud**, click **Your VPCs.**
3. Select the checkbox corresponding to your VPC. Then, under **Actions**, select **Edit DNS hostnames**.



Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set
srgdem	vpc-0143197ca9bd9c117	available	10.0.0.0/16	-	dopt-ee27ca87
eks-vpc	vpc-0d1b1ea35a9cc146f	available	172.30.0.0/21	-	dopt-ee27ca87
	vpc-aa9e64c3	available	172.31.0.0/16	-	dopt-ee27ca87

VPC: vpc-0143197ca9bd9c117

4. On the **Edit DNS Hostnames** page, for **DNS hostnames**, select the **enable** checkbox.
5. Click Save, then click **Close**.

To enable DNS using the AWS CLI:

1. Execute the following commands *in order*, using the VPC ID of your created VPC:
```
# aws ec2 modify-vpc-attribute \
--vpc-id <VPC Id> \
--enable-dns-support

# aws ec2 modify-vpc-attribute\
 --vpc-id <VPC Id> \
--enable-dns-hostnames
```

These commands have no output.

```
For example:
# aws ec2 modify-vpc-attribute \
--vpc-id vpc-0143197ca9bd9c117 \
--enable-dns-support
# aws ec2 modify-vpc-attribute \
--vpc-id vpc-0143197ca9bd9c117 \
--enable-dns-hostnames
```

Next Step: Create the External IP Address

Creating the External (Public) IP Address (EIP)

The external IP address (EIP) is required for the NAT Gateway, used by the worker nodes, to access the Elastic Container Registry (ECR). In this step, you will create the EIP and then tag it.

1. Run the following command:
```
# aws ec2 allocate-address --domain vpc
```
2. Record the `AllocationId` value in the AWS worksheet.

Example output:

```
{
"PublicIp": "18.194.179.100",
"AllocationId": "eipalloc-004be822658206abe",
"PublicIpv4Pool": "amazon",
"NetworkBorderGroup": "eu-central-1",
"Domain": "vpc"
}
```

To tag the EIP:

1. Run the following command:
   ```
   # aws ec2 create-tags \
   --resources <Allocation Id> \
   --tags Key=Name,Value=<eip-name>
   ```

Parameters:

`<Allocation Id>`: Use the Allocation ID of the EIP.

`<eip-name>`: Assign an EIP name for easier identification.

Example:

```
# aws ec2 create-tags \
--resources eipalloc-004be822658206abe \
--tags Key=Name,Value=srgdemo-eip
```

Next Step: Creating the Subnets

Creating and Tagging the Subnets

Each availability zone requires one private and one public subnet for high availability. In this section, you will create one private and one public subnet for each of the three availability zones, for a total of six subnets.

Each subnet must meet the following criteria:

- Each subnet come from the VPC IP range.
- Subnets must not overlap one another.

All six subnets will be created in the same way. They will be distinguished based on the route table, internet gateway, and NAT gateway attachments.

Before proceeding, make sure you have completed your AWS worksheet with your subnet names, CIDRs, and availability zones.

To create a subnet:

1. Retrieve the availability zone names by running the following command:
   ```
   # aws ec2 describe-availability-zones \
   | jq -r '.AvailabilityZones[ ].ZoneName'
   ```

Example output:

```
eu-central-1a
```

```
eu-central-1b
```

```
eu-central-1c
```

2. Create the first subnet by running the following command, which will output the subnet ID:
   ```
   # aws ec2 create-subnet \
   --availability-zone <availability zone> \
   --cidr-block <CIDR> --vpc-id <VpcId> | jq -r '.Subnet.SubnetId'
   ```

For example:

```
# aws ec2 create-subnet \
--availability-zone eu-central-1a \
--cidr-block 10.0.1.0/24 \
--vpc-id vpc-0143197ca9bd9c117 | jq -r '.Subnet.SubnetId'
```

```
subnet-06a8caab19022c544
```

3. Repeat Step 2 for all rows from the subnet planning table in the AWS worksheet.

You should now tag the new subnets to differentiate between public and private subnets, as well as tag the private subnets for load balancing.

To tag the subnets:

1. Tag each **public** subnet by running the following command for each public subnet:
   ```
   # aws ec2 create-tags \
   --resources <public subnet id> \
   --tags Key=Name,Value=<subnet name>
   ```
2. Tag each **private** subnet by running this command for each private subnet:
   ```
   # aws ec2 create-tags \
   --resources <private subnet id> \
   --tags Key=Name,Value=<subnet name> Key=kubernetes.io/role/internal-
   elb,Value=1
   ```

Parameters:

`<public/private subnet id>`: The value from column **Subnet ID** in your planning table on the AWS worksheet.

`<public/private subnet name>`: The value from column **Subnet name** in your planning table on the AWS worksheet.

Example:

```
# aws ec2 create-tags \
--resources subnet-06a8caab19022c544 \
--tags Key=Name,Value=srgdemo-public-subnet-1
```

```
# aws ec2 create-tags \
--resources subnet-0fb2ebb5882c061f0 \
--tags Key=Name,Value=srgdemo-private-subnet-1
Key=kubernetes.io/role/internal-elb,Value=1
```

Next Step: Creating the Internet Gateway

Creating the Internet Gateway

The Internet gateway is the prerequisite for the NAT gateway, which will be created later.

To create the internet gateway and attach it to the VPC:

1. Run the following command:
   ```
   # aws ec2 create-internet-gateway
   ```

Example output:

```
{
   "InternetGateway":{
      "Tags":[

      ],
      "InternetGatewayId":"igw-0ddcfa7511fe10b43",
      "Attachments":[

      ]
   }
}
```

2. Record the value of `InternetGatewayId` in your AWS worksheet.
3. Optionally, you might tag the internet gateway by running the following command:
   ```
   # aws ec2 create-tags --resources <InternetGatewayId> --tags
   Key=Name,Value=<internet gateway name>
   ```
4. Attach the internet gateway to your previously-created VPC by running the following command (command has no output):

```
# aws ec2 attach-internet-gateway --internet-gateway-id
<InternetGatewayId> --vpc-id <VPC Id>
```

For example:

```
# aws ec2 attach-internet-gateway \
--internet-gateway-id igw-0ddcfa7511fe10b43 \
--vpc-id vpc-0143197ca9bd9c117
```

Next Step: Creating the NAT Gateway

Creating the NAT Gateway

The NAT gateway is required for worker nodes to connect to the Elastic Container Registry (ECR), which is used for downloading CDF and product images.

To create the NAT gateway:

1. Run the following command:

```
# aws ec2 create-nat-gateway \
--allocation-id <EIP allocation Id> \
--subnet-id <public subnet id>
```

Example input and output:

```
# aws ec2 create-nat-gateway \
--allocation-id eipalloc-004be822658206abe \
--subnet-id subnet-0c0ca63f2f793907d
```

```
{
   "NatGateway":{
      "CreateTime":"2020.08-20T20:53:01.000Z",
      "NatGatewayAddresses":[
         {
            "AllocationId":"eipalloc-004be822658206abe"
         }
      ],
      "NatGatewayId":"nat-013416dad7b7656ea",
      "State":"pending",
      "SubnetId":"subnet-0c0ca63f2f793907d",
      "VpcId":"vpc-0143197ca9bd9c117"
   }
}
```

2. Record the `NatGatewayId` value in your AWS worksheet.

Next Step: Creating the Route Tables

Creating the Route Tables

Route tables define the routing paths between resources in private and public subnets and the Internet.

1. Run the following command to create a route table and retrieve its ID:
   ```
   # aws ec2 create-route-table \
   --vpc-id <VpcId> | jq -r '.RouteTable.RouteTableId'
   ```
2. Run the command in Step 1 a second time, to create another route table and retrieve its ID.

Example input and output:

```
# aws ec2 create-route-table \
--vpc-id vpc-0143197ca9bd9c117 \
| jq -r '.RouteTable.RouteTableId'
```

```
rtb-0deda70daa09ca3bfw
```

3. Tag the first route table as private. Run the command:
   ```
   # aws ec2 create-tags --resources <route table ID> \
   --tags Key=Name,Value=<route table name indicating private>
   ```

Example:

```
# aws ec2 create-tags |
--resources rtb-0deda70daa09ca3bf \
--tags Key=Name,Value=srgdemo-private-route-table
```

4. Repeat Step 3 for the second route table, with the `--tags` value indicating `public` instead of `private`.

Next Step: Associating the Route Tables to Subnets

Associating the Route Tables to Subnets

The route tables will now need to be associated to the subnets you have created.

To associate the route tables to your public subnets:

1. Select one of your **public** subnets.
2. Associate the **public** route table to the selected **public** subnet by running the command:
   ```
   # aws ec2 associate-route-table \
   --route-table-id <public route table ID> \
   --subnet-id <public subnet ID>
   ```
3. Repeat the command in Step 2 for each of the other two **public** subnets.

To associate the route tables to your private subnets:

1. Select one of your **private** subnets.
2. Associate the **private** route table to the selected **private** subnet by running the command:
   ```
   # aws ec2 associate-route-table \
   --route-table-id <private route table ID> \
   --subnet-id <private subnet ID>
   ```
3. Repeat the command in Step 2 for each of the other two **private** subnets.

Example input and output:

```
# aws ec2 associate-route-table \
--route-table-id rtb-0deda70daa09ca3bf \
--subnet-id subnet-0fb2ebb5882c061f0
```

```
{
   "AssociationId":"rtbassoc-781d0d1a",
   "AssociationState":{
      "State":"associated"
   }
}
```

Next Step: Adding the NAT Gateway Route Path to the Private Route Table

Adding the NAT Gateway Route Path to the Private Route Table

1. Run the following command:
   ```
   # aws ec2 create-route \
   --route-table-id <private route table Id> \
   --destination-cidr-block "0.0.0.0/0" \
   --nat-gateway-id <NAT GW Id>
   ```
2. The command will return the creation status. A status of `true` indicates that the request succeeded.

Example input and output:

```
# aws ec2 create-route \
--route-table-id rtb-0deda70daa09ca3bf \
--destination-cidr-block "0.0.0.0/0" \
--nat-gateway-id nat-013416dad7b7656ea
```

```
{
   "Return":true
}
```

Next Step: Adding the Internet Gateway Route Path to the Public Routing Table

Adding the Internet Gateway Route Path to the Public Routing Table

1. Run the following command:

   ```
   # aws ec2 create-route \
   --route-table-id <public route table Id> \
   --destination-cidr-block "0.0.0.0/0" \
   --gateway-id <Internet Gateway Id>
   ```

Example:

```
# aws ec2 create-route \
--route-table-id rtb-0fa9f294a3743c9aa \
--destination-cidr-block "0.0.0.0/0" \
--gateway-id igw-0ddcfa7511fe10b43
```

Next Step: Creating Security Groups

Creating Security Groups

A *security group* is an AWS resource that acts as a firewall for the subnets. Every AWS resource must be assigned a security group so they will be network accessible. If a resource is assigned to multiple security groups, then all rules from all groups will be applied to the resource.

You will need to create two security groups, one for the bastion host and one for intra-VPC connectivity. The procedures are explained in the following sections.

Next Step: Creating the Security Group for the Bastion Host

Creating the Security Group for the Bastion Host

In order to connect to the bastion from the Internet and perform the configuration and installation tasks, you must open the connection on the default SSH port (port 22) from any address.

> Optionally, you can limit the access to the bastion by specifying your own public, static IP address while adding your own inbound rule as described below. Replace `0.0.0.0/0` with your own public IP address. If you choose to specify your own IP address, talk to your AWS infrastructure administrator before proceeding.

To create the security group for the bastion host:

1. Run the following command:
   ```
   # aws ec2 create-security-group \
   --description "Enables SSH Access to Bastion Hosts" \
   --group-name <group name> --vpc-id <VpcId>
   ```

Parameters:

`<group name>`: A descriptive security group name of your choice; in our examples we will use `srgdemo-bastion-sg`.

`<VpcId>`:The VPC ID of the VPC you created earlier.

Example:

```
# aws ec2 create-security-group \
--description "Enables SSH Access to Bastion Hosts" \
--group-name srgdemo-bastion-sg \
--vpc-id vpc-0143197ca9bd9c117
```

```
{
   "GroupId":"sg-00b5fcc4294d234f6"
}
```

2. Record the bastion security group ID in your AWS worksheet.

Adding the Inbound Rule

You will connect to the bastion using SSH on default port 22, so the newly-created security group needs to be opened to inbound connection on port 22 and the TCP protocol.

To add the inbound rule:

1. Open the security group to inbound connections on the default SSH port 22 (TCP) by running the following command:
   ```
   # aws ec2 authorize-security-group-ingress \
   --group-id <bastion security group ID> \
   --ip-permissions IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges='
   [{CidrIp=0.0.0.0/0,Description="SSH access; unlimited."}]'
   ```

Example:

```
# aws ec2 authorize-security-group-ingress \
--group-id sg-00b5fcc4294d234f6 \
--ip-permissions
```

2. Remove the default wide-open outbound rule by running the following command:
   ```
   # aws ec2 revoke-security-group-egress \
   --group-id <security group ID> \
   ```

```
    --protocol all \
    --port -1 \
    --cidr 0.0.0.0/0
```

While working from the bastion you will need to connect to various resources on the internet. Protocols and description for ports are shown in the following table:

Port	Protocol	Allowed CIDR	Description
80	TCP	`0.0.0.0/0`	HTTP
443	TCP	`0.0.0.0/0`	HTTPS

3. Add HTTP and HTTPS outbound rules by running the following command:

```
    # aws ec2 authorize-security-group-ingress \
    --group-id <bastion security group Id> \
    --ip-permissions
    IpProtocol=<protocol>,FromPort=<port>,ToPort=<port>,IpRanges='
    [{CidrIp=0.0.0.0/0,Description="<Description>"}]'
```

For `<port>`, `<protocol>`, and `<description>`: Use values from the table above.

Example:

```
# aws ec2 authorize-security-group-ingress \
--group-id sg-00b5fcc4294d234f6 \
--ip-permissions IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges='
[{CidrIp=0.0.0.0/0,Description="SSH access; unlimited."}]'
```

Next Step: Creating the Security Group for Intra-VPC Communication

Creating the Security Group for Intra-VPC Communication

This security group is dedicated to resources inside the VPC, and will allow unlimited communication between them. It will also allow outbound connection to the HTTP and HTTPS worldwide.

To create the security group for intra-VPC communication, use the same steps for creating the bastion's security group. Change the description to (for example) `<cluster name> intra VPC SG`.

- Note this name and ID to the AWS worksheet.
- Then, remove the default wide-open outbound rule. Repeat the process you performed for the bastion security group, of course using different security group ID.

Add inbound rule from itself

For communication between intra-VPC resources we will add a rule enabling all communication coming from this security group. To do this, use run a command (for example):

```
# aws ec2 authorize-security-group-ingress
--group-id <security group Id> \
--protocol all \
--port -1 \
--source-group <security group Id>
```

Parameters:

`<security group Id>`: Use the Id of security group just created; use the same value for both instances of the parameter.

Add HTTP and HTTPS outbound rules

For retrieving external resources (such as for CDF) and product images from the ECR, OS updates, and similar files, resources inside the VPC need to be able to connect to to HTTP/HTTPS on the internet.Repeat the process you performed for the bastion security group.

To add outbound rule to itself, run the command:

```
# aws ec2 authorize-security-group-egress \
--group-id <security group Id> \
--protocol all \
--port -1 \
--cidr <VPC CIDR>
```

`<security group Id>`: Use the Id of security group just created; use the same value for both instances of the parameter.

`<VPC CIDR>`: Use the same CIDR you used for creating the VPC.

Next Step: IAM Roles

IAM Roles

An *IAM role* is an IAM (AWS Identity and Access Management) entity that defines a set of permissions for making AWS service requests and manipulating various resources.

> Roles are shareable. Instead of creating new roles, you might use existing roles your organization has previously created. IAM is not region dependent, roles can be reused in all regions your organization uses.

You will create two roles: one for EKS (Elastic Kubernetes Service) and one for worker nodes, and assign them specific policies to define permissions.

Roles, policy names, and corresponding policy ARNs are shown in the following table:

Role	Policy Name	Policy ARN
EKS	AmazonEKSClusterPolicy	arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
EKS	AmazonEKSServicePolicy	arn:aws:iam::aws:policy/AmazonEKSServicePolicy
Worker Nodes	AmazonEKSWorkerNodePolicy	arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
Worker Nodes	AmazonEC2ContainerRegistryReadOnly	arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
Worker Nodes	AmazonEKS_CNI_Policy	arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy

EKS Policies

EKS requires the following policies to be granted:

- AmazonEKSClusterPolicy This policy provides Kubernetes the permissions it requires to manage resources on your behalf. Kubernetes requires EC2: CreateTags permissions to place identifying information on EC2 resources including but not limited to Instances, Security Groups, and Elastic Network Interfaces.
  ARN: `arn:aws:iam::aws:policy/AmazonEKSClusterPolicy`
- AmazonEKSServicePolicy This policy allows Amazon Elastic Container Service for Kubernetes to create and manage the necessary resources to operate EKS Clusters.
  ARN: `arn:aws:iam::aws:policy/AmazonEKSServicePolicy`

Worker Node Policies

For worker nodes in EKS, the following policies must be granted:

- AmazonEKSWorkerNodePolicy This policy allows Amazon EKS worker nodes to connect to Amazon EKS Clusters.
  ARN: `arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy`
- AmazonEC2ContainerRegistryReadOnly Provides read-only access to Amazon EC2 Container Registry repositories.
  ARN: `arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly`
- AmazonEKS_CNI_Policy This policy provides the Amazon VPC CNI Plugin (amazon-vpc-cni-k8s) the permissions it requires to modify the IP address configuration on your EKS worker nodes. This permission set enables the CNI to list, describe, and modify Elastic Network Interfaces on your behalf. For more information about the AWS VPC CNI Plugin, see the link here: https://github.com/aws/amazon-vpc-cni-k8s.
  ARN: `arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy`

Next Step: Creating the EKS Role

Creating the IAM Role for EKS

1. Run the following command:
   ```
   # aws iam create-role \
   --role-name <role name> \
   --assume-role-policy-document <role policy document>
   ```

Parameters:

`<role name>`: A name chosen for ease of reference; in our examples, we will use `srgdemo-eks-svc-role`.

`<role policy document>`: The location of a JSON document granting temporary security credentials to perform actions on resources and defining which resources are accessible. There is a ready-to-use document named `EksRolePolicyDocument.json` of the download package `aws-byok-installer-<version>.zip`, after unzipping, in the in the `objectdefs` folder. This document defines that the cluster can request temporary security credentials to `eks.amazonaws.com` only.

Example output:

```
{
  "Role": {
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "CreateDate": "2013-06-07T20:43:32.821Z",
    "RoleName": "Test-Role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

2. Record the ARN (Amazon Resource Name) value in your AWS worksheet.

Example input and output:

```
# aws iam create-role \
--role-name srgdemo-eks-svc-role \
--assume-role-policy-document file://./jsons/EksRolePolicyDocument.json
```

```
{
    "Role": {
        "Path": "/",
        "RoleName": "srgdemo-eks-svc-role",
        "RoleId": "AROARVXFDN4TOT5P3E3AQ",
        "Arn": "arn:aws:iam::115370848038:role/srgdemo-eks-svc-role",
        "CreateDate": "2020-05-18T12:10:48Z",
```

```
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "eks.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

Note the `Arn` value `*arn:aws:iam::115370848038:role/srgdemo-eks-svc-role*`.

3. Attach a policy to the EKS role by running the command:
   ```
   # aws iam attach-role-policy --role-name <role name> --policy-arn <policy arn>
   ```

Parameters:

`<role name>` is the role name you have chosen when creating a new role

`<policy arn>` is the policy ARN from the description above.

4. Repeat Step 3 for the next policy, changing the policy ARN to match.

Example commands:

```
# aws iam attach-role-policy
--role-name srgdemo-eks-svc-role \
--policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
```

```
# aws iam attach-role-policy \
--role-name srgdemo-eks-svc-role \
--policy-arn arn:aws:iam::aws:policy/AmazonEK
```

Next Step: Creating the Worker Node Role

Creating the Worker Node Role

1. Run the following command:
   ```
   # aws iam create-role \
   --role-name <role name> \
   --assume-role-policy-document <role policy document>
   ```

Parameters:

`<role name>`: A name chosen for ease of reference; in our examples, we will use srgdemo-eks-svc-role.

`<role policy document>`:The location of a JSON document granting temporary security credentials to perform actions on resources and defining which resources are accessible. The CDF installation package includes a ready-to-use document named `WorkerNodesRolePolicyDocument.json` in the downloadable package `aws-byok-installer-<version>.zip`, after unzipping, in the in the `objectdefs` folder This document defines that the cluster can request temporary security credentials to `eks.amazonaws.com` only.

Example output:

```
{
  "Role": {
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "CreateDate": "2013-06-07T20:43:32.821Z",
    "RoleName": "Test-Role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

2. Record the ARN (Amazon Resource Name) value in your AWS worksheet.

Example input and output:

```
# aws iam create-role \
--role-name srgdemo-workernodes-svc-role \
--assume-role-policy-document
file://./jsons/WorkerNodesRolePolicyDocument.json
```

```
{
    "Role": {
        "Path": "/",
        "RoleName": "srgdemo-workernodes-svc-role",
        "RoleId": "AROARVXFDN4TICMZYPKJ2",
        "Arn": "arn:aws:iam::115370848038:role/srgdemo-workernodes-svc-role",
        "CreateDate": "2020-05-19T16:20:11Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ec2.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

3. Attach a policy to the **worker node** role by running the following command:

   ```
   # aws iam attach-role-policy \
   --role-name <role name> \
   --policy-arn <policy arn>
   ```

Parameters:

`<role name>`:The role name you have chosen when creating a new role.

`<policy arn>`:The policy ARN from the description above.

4. Repeat Step 3 each policy, changing the policy ARN to match.

Example commands:

```
# aws iam attach-role-policy \
--role-name srgdemo-workernodes-svc-role \
--policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
```

```
# aws iam attach-role-policy \
--role-name srgdemo-workernodes-svc-role \
--policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
```

```
# aws iam attach-role-policy \
--role-name srgdemo-workernodes-svc-role \
--policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
```

Next Step: Creating and Configuring the Bastion

Creating and Configuring the Bastion

The *bastion* is the dedicated host which provides secure access to Linux instances located in the private and public subnets of your virtual private cloud (VPC). In this section, you will configure the bastion for your deployment environment.

Next Step: Creating the SSH Keypair

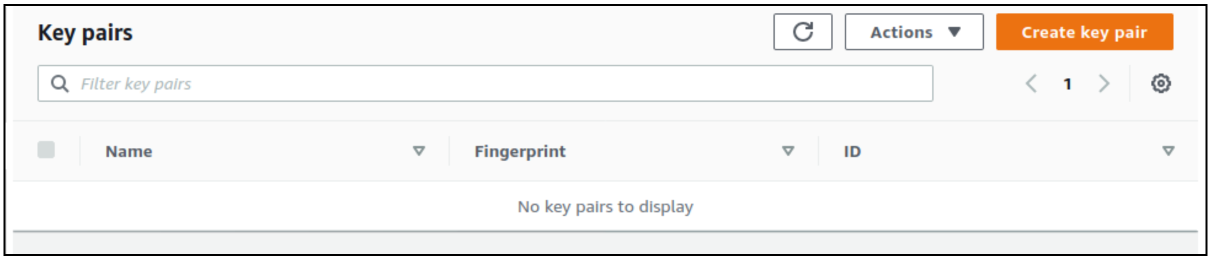
Creating the SSH Keypair

In order to connect to and perform tasks on the bastion, you will use SSH with keypair authentication. In this section, you will create a key pair and store its private value and fingerprint to local files.

The SSH keypair will be used later for instantiating worker nodes. Optionally, you can create a separate keypair for the bastion and for worker nodes. In that case, follow the steps described here, and give each keypair a distinct name.

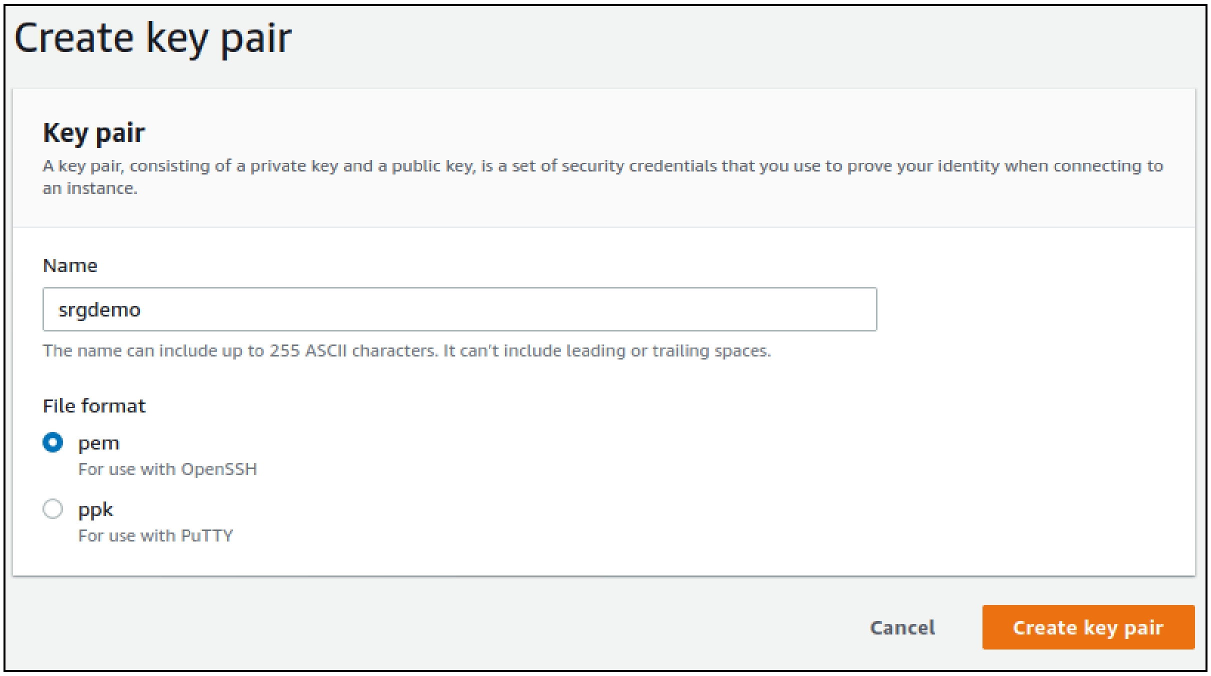
To create the keypair using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation pane, under **Network and Security**, select **Key Pairs**
3. On the **Key Pairs** management dialog, click **Create key pair.**



Name	Fingerprint	ID
No key pairs to display		

4. On the **Create Key Pair** page, enter values for the following:
   - **Name:** The key pair name will be later used for instantiating bastion as well as worker nodes. You will also use it as a CLI parameter when using an SSH client.
   - **File format:** Choose the format suitable for your client; check the description as shown.



Create key pair

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

srgdemo

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

File format

● pem
For use with OpenSSH

○ ppk
For use with PuTTY

5. Click **Create Key Pair.**
6. On the **Key pair** management dialog, save the private part to a secure location when prompted.

You *must* save the value now, and will not be able to save it later.

5. Optionally, save the key pair fingerprint to the same secure location. The optimal way to store this value is in the file named the same as the private part, exchanging the suffix. For example:
   `srgdemo.fingerprint.`
   You can later compare your locally-stored fingerprint value with the one presented on the table on the web UI.
6. To store the fingerprint value, copy the value in the **Fingerprint** column to a text document on your local machine.
7. Record the keypair name and keypair fingerprint to the AWS worksheet.

To create the SSH key pair using the CLI:

1. Enter the following commands:

```
# export KEYPAIR_NAME=<Key pair name>
```

```
# export KEYPAIR_CREATION=$(aws ec2 create-key-pair \
--key-name ${KEYPAIR_NAME}
```

```
# echo $KEYPAIR_CREATION | jq -r '.KeyMaterial' \
| sed "s/\\\\n/\n/g" > ~/.ssh/${KEYPAIR_NAME}.pem
```

```
# echo $KEYPAIR_CREATION \
| jq -r '.KeyFingerprint' > ~/.ssh/${KEYPAIR_NAME}.fingerprint
```

```
# chmod 400 ~/.ssh/${KEYPAIR_NAME}.pem
```

Replace the `<KEYPAIR_NAME>` value with your real key pair name. In our examples, we use `srgdemo.`

Example commands:

```
# export KEYPAIR_NAME=srgdemo
```

```
# export KEYPAIR_CREATION=$(aws ec2 create-key-pair \
--key-name ${KEYPAIR_NAME})
```

```
# echo $KEYPAIR_CREATION | jq -r '.KeyMaterial' \
| sed "s/\\\\n/\n/g" > ~/.ssh/${KEYPAIR_NAME}.pem
```

```
# echo $KEYPAIR_CREATION \
| jq -r '.KeyFingerprint' > ~/.ssh/${KEYPAIR_NAME}.fingerprint
```

```
# chmod 400 ~/.ssh/${KEYPAIR_NAME}.pem
```

Next Steps: Determining the Image ID

Determining AMI ID

You must determine the AMI (Amazon Machine Image) ID used for your bastion instance. You can select an OS image and its corresponding AMI from the AWS Marketplace, or you can determine an AMI by your region from the following table:

Region	CentOS 7	Amazon Linux
ap-northeast-1	ami-045f38c93733dd48d	ami-04b2d1589ab1d972c
ap-northeast-2	ami-06cf2a72dadf92410	ami-0be3e6f84d3b968cd
ap-south-1	ami-02e60be79e78fef21	ami-0b99c7725b9484f9e
ap-southeast-1	ami-0b4dd9d65556cac22	ami-0fb6b6f9e81056553
ap-southeast-2	ami-08bd00d7713a39e7d	ami-075caa3491def750b
ca-central-1	ami-033e6106180a626d0	ami-0a67d15f2858e33cb
eu-central-1	ami-04cf43aca3e6f3de3	ami-026d3b3672c6e7b66
eu-north-1	ami-5ee66f20	ami-8c169ef2
eu-west-1	ami-0ff760d16d9497662	ami-0862aabda3fb488b5
eu-west-2	ami-0eab3a90fc693af19	ami-0bdfa1adc3878cd23
eu-west-3	ami-0e1ab783dc9489f34	ami-05b93cd5a1b552734
sa-east-1	ami-0b8d86d4bf91850af	ami-0bb96001cf2299257
us-east-1	ami-02eac2c0129f6376b	ami-035b3c7efe6d061d5
us-east-2	ami-0f2b4fc905b0bd1f1	ami-02f706d959cedf892
us-west-1	ami-074e2d6769f445be5	ami-0fcdcdb074d2bac5f
us-west-2	ami-01ed306a12b7d1c96	ami-0f2176987ee50226e

You can also get new image IDs by running OS-based commands:

For CentOS Linux 7, run the following command:

```
# aws ec2 describe-images \
--filters "Name=name,Values=CentOS Linux 7 x86_64 HVM EBS ENA*" \
 "Name=architecture,Values=x86_64" "Name=virtualization-type,Values=hvm"
"Name=root-device-type,Values=ebs" "Name=owner-alias,Values=aws-marketplace"
```

```
\
| jq '.Images | sort_by(.CreationDate) | [last]'
```

For Amazon Linux, run the following command:

```
# aws ec2 describe-images \
--filters "Name=description,Values=Amazon Linux AMI *"
"Name=architecture,Values=x86_64" "Name=virtualization-type,Values=hvm"\
 "Name=root-device-type,Values=ebs" "Name=owner-alias,Values=aws-marketplace"
| jq '.Images | sort_by(.CreationDate) | [last]'
```

Record the `ImageId` value in the AWS worksheet.

Next Step: Selecting a Bastion Hardware Instance Type

Selecting a Bastion Hardware Instance Type

The type of host to use for your bastion depends on your deployment plans. Amazon offers a detailed list of EC2 Instance Types with hardware specifications for each. Select and prepare a host that balances and optimizes CPU, memory, storage, and pricing.

For purposes of examples here, we will assume `t2.medium` as your bastion instance type, which will be used only to perform a few configuration tasks and CDF bootstrap install. Your own environment needs might differ.

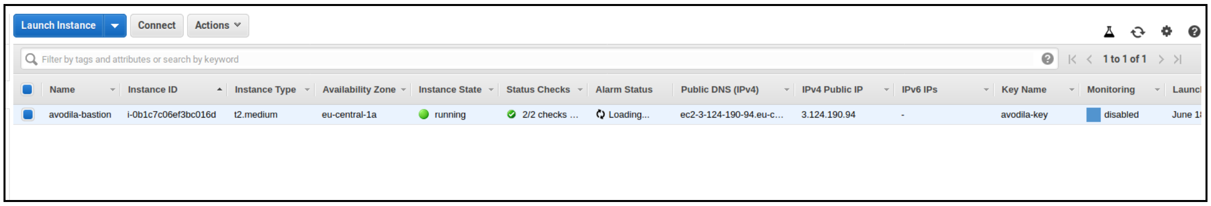
Once you've selected a bastion host, record its type in your AWS worksheet.

Next Step: Starting the Bastion Instance

Starting the Bastion Instance

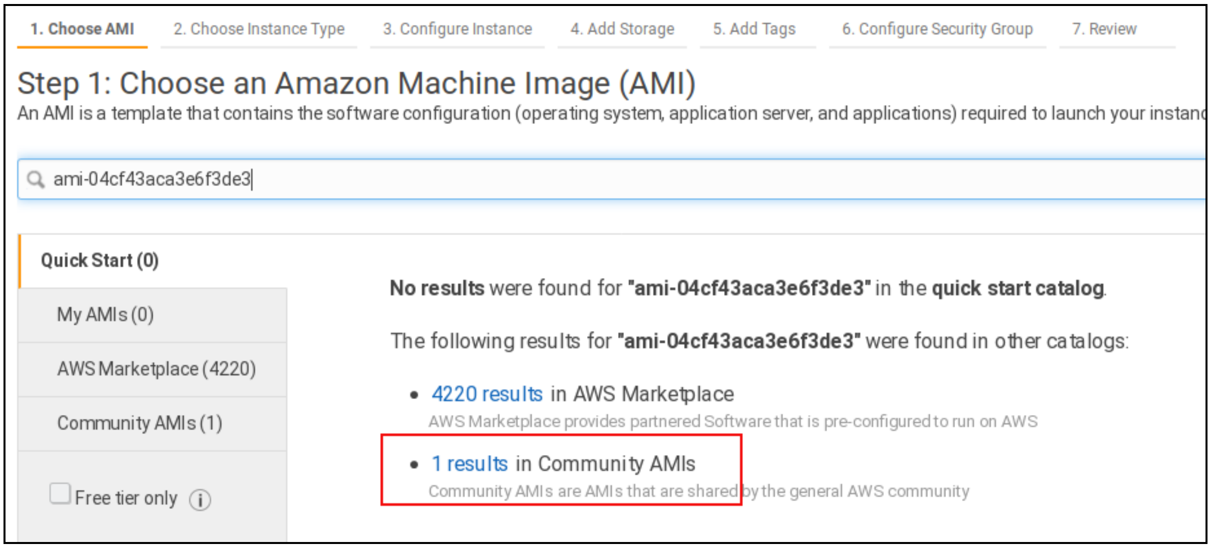
To start the bastion instance through the Web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation pane, under **INSTANCES**, click **Instances**.
3. On the Instances management dialog, click **Launch Instance** to start the wizard.



■	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launc
■	avodila-bastion	i-0b1c7c06ef3bc016d	t2.medium	eu-central-1a	● running	✓ 2/2 checks ...	Loading...	ec2-3-124-190-94.eu-c...	3.124.190.94	-	avodila-key	disabled	June 1

4. On the **Step 1: Choose an AMI** page, in the search box, enter your AMI ID (from your AWS worksheet) and press Enter. The page displays a single result in **Community AMIs.**



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instanc

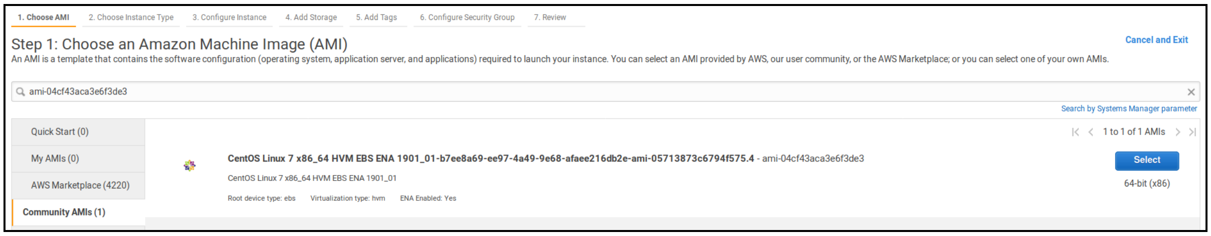
ami-04cf43aca3e6f3de3

No results were found for **"ami-04cf43aca3e6f3de3"** in the **quick start catalog**.

The following results for **"ami-04cf43aca3e6f3de3"** were found in other catalogs:

- 4220 results in AWS Marketplace
  AWS Marketplace provides partnered Software that is pre-configured to run on AWS
- 1 results in Community AMIs
  Community AMIs are AMIs that are shared by the general AWS community

5. Click the link for **1 results in Community AMIs.** The selected OS is displayed.
6. Click **Select**.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

ami-04cf43aca3e6f3de3

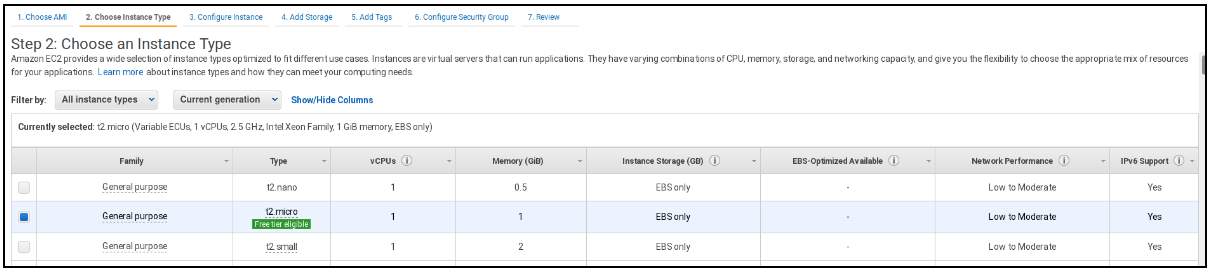
CentOS Linux 7 x86_64 HVM EBS ENA 1901_01-b7ee8a69-ee97-4a49-9e68-afaee216db2e-ami-05713873c6794f575.4 - ami-04cf43aca3e6f3de3

CentOS Linux 7 x86_64 HVM EBS ENA 1901_01

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

64-bit (x86)

7. On the **Step 2: Choose an Instance Type** page, on the Instance Type list, search for and select your own instance type (use the browser search function if needed). Then, click **Next: Configure Instance Details.**



Step 2: Choose an Instance Type

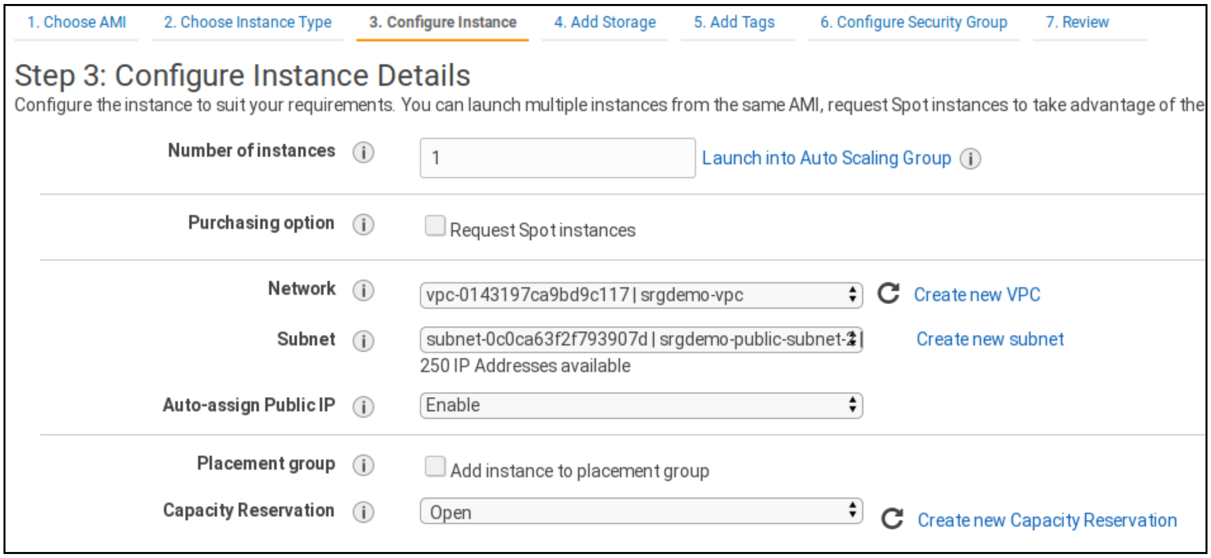
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes

8. On the **Step 3: Configure Instance Details** page, enter these values for the following settings:
   - **Network:** Choose your VPC.
   - **Subnet:** Choose one of your three public subnets.
   - **Auto-assign Public IP:** Enable this value.
9. Click **Next: Add Storage** .



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-0143197ca9bd9c117 | srgdemo-vpc — Create new VPC

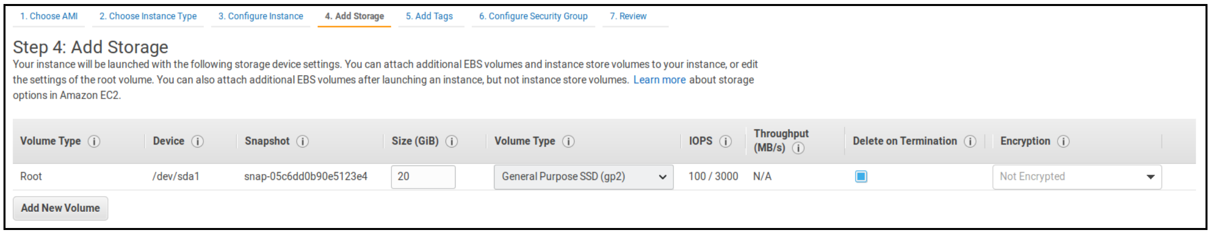
Subnet: subnet-0c0ca63f2f793907d | srgdemo-public-subnet-2 — Create new subnet
250 IP Addresses available

Auto-assign Public IP: Enable

Placement group: □ Add instance to placement group

Capacity Reservation: Open — Create new Capacity Reservation

10. On the **Step 4: Add Storage** page, set the root volume size according to your previously-decided needs. In this example, we assume that we will be using it only to upload product images, so we will set it to 20 GB.



Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-05c6dd0b90e5123e4	20	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

11. Enable **Delete on Termination.**

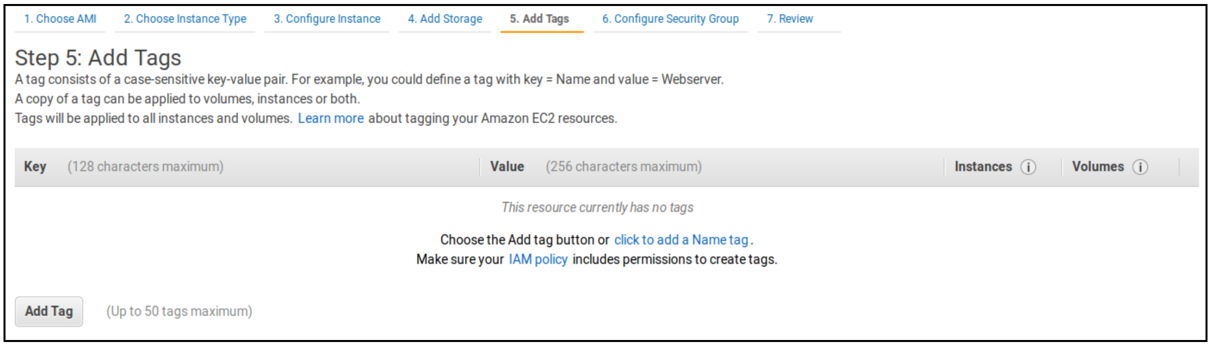
12. Click **Next: Add Tags**.

13. On the **Step 5: Add Tags** page, click **Add Tag.**

14. Enter and save a tag called *Name* with the value of your bastion name (for example, *srgdemo-bastion.*)

15. Optionally, you can add other tags as needed.

16. Click **Next: Configure Security Group.**



Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

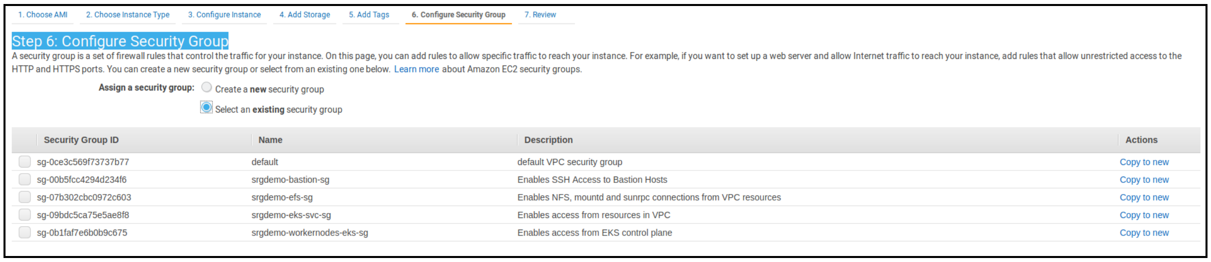
Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes

This resource currently has no tags

Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

(Up to 50 tags maximum)

17. On the **Step 6: Configure Security Group** page, under **Assign a security group**, choose **Select an existing security group**.
18. The list shows all security groups associated with your VPC. Select **both** the Bastion security group and Intra VPC security group. (Choose by name or ID from your AWS worksheet.)
19. Click **Review and Launch.**



Step 6: Configure Security Group

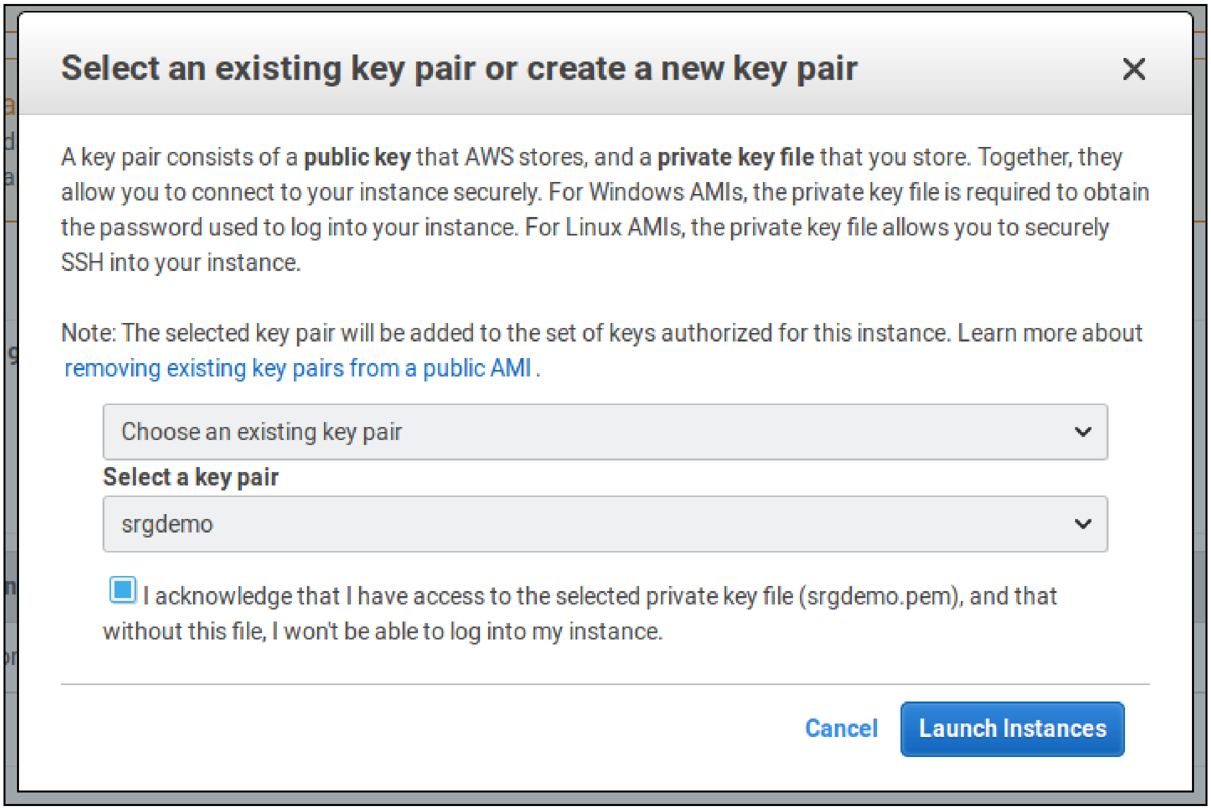
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ○ Create a **new** security group
● Select an **existing** security group

	Security Group ID	Name	Description	Actions
□	sg-0ce3c569f73737b77	default	default VPC security group	Copy to new
□	sg-00b5fcc4294d234f6	srgdemo-bastion-sg	Enables SSH Access to Bastion Hosts	Copy to new
□	sg-07b302cbc0972c603	srgdemo-efs-sg	Enables NFS, mountd and sunrpc connections from VPC resources	Copy to new
□	sg-09bdc5ca75e5ae8f8	srgdemo-eks-svc-sg	Enables access from resources in VPC	Copy to new
□	sg-0b1faf7e6b0b9c675	srgdemo-workernodes-eks-sg	Enables access from EKS control plane	Copy to new

20. On the **Step 7: Review Instance and Launch** page, review all parameters for correctness and fix if necessary. Then click **Launch**.
21. On the **Select an existing key pair...** dialog, pick **Choose an existing key pair** from the drop-down, and then select your previously-created key pair.

> **Important!** Confirm that you have the private part of your key pair accessible on your local host. Without the private part, your bastion will not be accessible through SSH.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

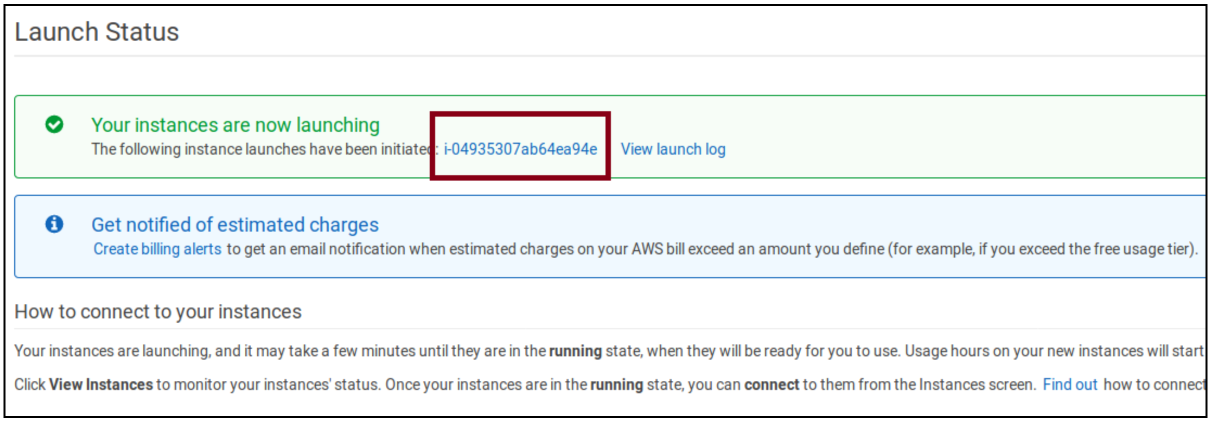
Choose an existing key pair

Select a key pair

srgdemo

☑ I acknowledge that I have access to the selected private key file (srgdemo.pem), and that without this file, I won't be able to log into my instance.

22. Click **Launch Instances**. The instance is launched and displayed.



Launch Status

Your instances are now launching
The following instance launches have been initiated: i-04935307ab64ea94e View launch log

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect

23. From on the **Launch Status** page, from the green box, copy your instance ID to your AWS worksheet under Notes.
24. Click **View Instances** to return to the **Instances management** page.

To start the bastion instance using the AWS CLI:

1. Run the following command:

```
# aws ec2 run-instances \
--image-id <Image Id> --count 1 \
--instance-type <Instance type> \
--key-name <Key pair name> \
--security-group-ids <security group Ids> \
--subnet-id <public subnet Id> \
--block-device-mappings <device mapping parameters> \
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=<bastion
instance name>}]' 'ResourceType=volume,Tags=[{Key=Name,Value=<bastion
instance volume name>}]' \
--associate-public-ip-address | jq '.Instances[].InstanceId'
```

Parameters:

`<Image Id>`: Your AMI ID.

`<Instance type>`: Your bastion type.

`<Key pair name>`: Name of the key pair previously created in SSH Keypair.

`<Security group IDs>`: IDs of the two security groups created by your AWS infrastructure administrators. Add **both** the Bastion Security group Id and Intra VPC Security group ID; separate entries with a single space character.

`<public subnet Id>`: ID of one of the three public subnets created by your AWS infrastructure administrators.

`<Device mapping parameters>`: See the example; used for changing root volume size. For more information about parameters and values, please run:
`aws ec2 run-instances help.`

`<bastion instance name>`: Name assigned to the bastion instance for easier identification.

`<bastion instance volume name>`: Name for the storage volume attached to the current bastion instance.

Example:

```
# aws ec2 run-instances --image-id ami-04cf43aca3e6f3de3 \
```

```
--count 1 --instance-type t2.medium \
```

```
--key-name srgdemo --security-group-ids sg-00b5fcc4294d234f6 sg-
0ce3c569f73737b77 \
```

```
--subnet-id subnet-0c0ca63f2f793907d \
```

```
--block-device-mappings DeviceName=/dev/sda1,Ebs={VolumeSize=10} \
```

```
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=srgdemo-
bastion}]' 'ResourceType=volume,Tags=[{Key=Name,Value=srgdemo-bastion-
volume}]' \
--associate-public-ip-address | jq '.Instances[].InstanceId'
```

2. The command returns an instance ID (for example, `i-06773a3ef6acd24f0`). Record your instance ID to the AWS worksheet.
3. Check the instance status by running the following command:
   ```
   # aws ec2 describe-instances \
   --instance-ids <Instance Id> | jq '.Reservations[].Instances[].State'
   ```

Example output (JSON):

```
{
   "Code":16,
   "Name":"running"
}
```

4. Repeat the check until the result shows *Name: running*. For example:
   ```
   # aws ec2 describe-instances \
   --instance-ids i-06773a3ef6acd24f0 | jq '.Reservations[].Instances
   [].State'
   ```
5. For easier identification of your bastion instance, tag it with a name by running the following command:
   ```
   # aws ec2 create-tags --resources <Instance Id> \
   --tags Key=Name,Value=<tag value>
   ```

For example:

```
# aws ec2 create-tags \
--resources i-06773a3ef6acd24f0 \
--tag Key=Name,Value=srgdemo-bastion
```

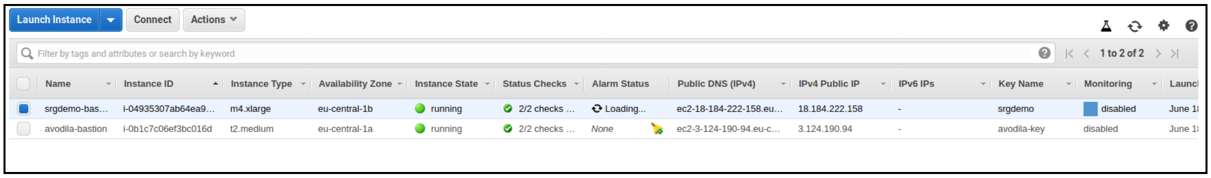
Next Step: Retrieving the Bastion Instance IP Address

Retrieving the Bastion Public IP

The bastion host provides the only access to the VPC and its resources. Therefore, you will need to know how to connect to it through its public IP address.

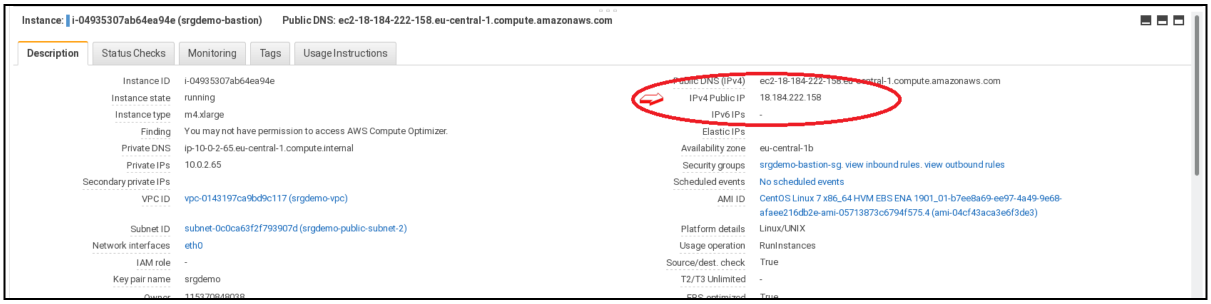
To retrieve the bastion public IP address using the Web UI:

1. On the **View Instances** page, select the bastion instance.



☐	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launc
☑	srgdemo-bas...	i-04935307ab64ea9...	m4.xlarge	eu-central-1b	● running	✓ 2/2 checks ...	Loading...	ec2-18-184-222-158.eu...	18.184.222.158	-	srgdemo	disabled	June 1
☐	avodila-bastion	i-0b1c7c06ef3bc016d	t2.medium	eu-central-1a	● running	✓ 2/2 checks ...	None	ec2-3-124-190-94.eu-c...	3.124.190.94	-	avodila-key	disabled	June 1

2. On the **Description** tab, locate the public IP address.



Instance: i-04935307ab64ea94e (srgdemo-bastion) Public DNS: ec2-18-184-222-158.eu-central-1.compute.amazonaws.com

Instance ID	i-04935307ab64ea94e	Public DNS (IPv4)	ec2-18-184-222-158.eu-central-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	18.184.222.158
Instance type	m4.xlarge	IPv6 IPs	-
Finding	You may not have permission to access AWS Compute Optimizer.	Elastic IPs	
Private DNS	ip-10-0-2-65.eu-central-1.compute.internal	Availability zone	eu-central-1b
Private IPs	10.0.2.65	Security groups	srgdemo-bastion-sg. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-0143197ca9bd9c117 (srgdemo-vpc)	AMI ID	CentOS Linux 7 x86_64 HVM EBS ENA 1901_01-b7ee8a69-ee97-4a49-9e68-afaee216db2e-ami-05713873c6794f575.4 (ami-04cf43aca3e6f3de3)
Subnet ID	subnet-0c0ca63f2f793907d (srgdemo-public-subnet-2)	Platform details	Linux/UNIX
Network interfaces	eth0	Usage operation	RunInstances
IAM role	-	Source/dest. check	True
Key pair name	srgdemo	T2/T3 Unlimited	-
Owner	115370848038	EBS optimized	True

3. Record the bastion public IP address in the AWS worksheet.

To retrieve the bastion public IP address using the CLI:

1. Run the following command:
   ```
   # aws ec2 describe-instances \
   --instance-ids <Instance Id> | jq -r '.Reservations[].Instances
   [].PublicIpAddress'
   ```
2. Record the bastion public IP in the AWS worksheet.

Example input and output:

```
# aws ec2 describe-instances \
--instance-ids i-06773a3ef6acd24f0 | jq -r '.Reservations[].Instances
[].PublicIpAddress'
```

```
18.184.151.208
```

Next Step: Connect to the Bastion and Download Software

Connecting to Bastion and Install Software Packages

Using the bastion's public IP address and the private part of your key pair, you will connect to the Bastion, install required tools, and perform several configuration tasks.

> In examples, we assume the keypair is stored in `~/.ssh`

To connect to Bastion and install required software packages:

1. Run the following command:
   ```
   # ssh -i ~/.ssh/<key pair name>.pem centos@<Bastion Public IP address>
   ```

For example:
```
# ssh -i ~/.ssh/srgdemo.pem centos@18.184.151.208
```

Installing `kubectl`

Next, you will need to install the kubectl tool for Kubernetes. AWS continually updates the `kubectl` version. You must use the version corresponding to the Kubernetes version used for the cluster (with a tolerance of one minor version).

Check the ITOM Platform: What's New page for the supported Kubernetes version for your version of CDF. Use only release and major version, for example, 1.15.10 would correspond to 1.15.

> You will be managing your EKS cluster with this version of `kubectl`, so its version must match the version required by CDF.

To install `kubectl`:

1. Record the required Kubernetes version in the AWS worksheet.

> In the following list of commands, find the URL for the `kubectl` version on the page Installing kubectl. Then replace the `curl -o kubectl...` command below with the correct command from that page.

2. If you have not installed the `epel` package, run the following command:
   ```
   # sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
   ```
3. Run the following commands to configure the bastion:
   ```
   # sudo yum install -y vim docker mc nfs-utils unzip jq htop ncdu nload nano xauth firefox
   # sudo groupadd docker
   # sudo usermod -a -G docker root
   ```

```
# sudo usermod -a -G docker centos
# sudo systemctl start docker
# sudo systemctl enable docker
# curl -o kubectl https://amazon-eks.s3.us-west-
2.amazonaws.com/1.15.11/2020-07-08/bin/linux/amd64/kubectl
# chmod +x ./kubectl
# sudo mv kubectl /usr/bin
# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
# unzip awscliv2.zip
# sudo ./aws/install -i /usr/local/aws -b /usr/local/bin
```

Configuring AWS CLI

To configure AWS CLI:

1. While connected to Bastion, run the following command:
   ```
   # aws configure
   ```

You will be prompted for the same security data, region and output format when you previously configured `AWS CLI` for the local host. Repeat the process outlined for AWS CLI configuration.

Next Steps: Download Installation Packages

Downloading Installation Tools and Packages

Download the installation packages for the CDF Installer and the product of your choice from the Micro Focus Entitlement Portal to a secure network location. After download, validate the digital signature of each file. You can store all the packages on you local computer, as most of the tasks could be performed on it.

For installation, you must have the following files (each package requires its corresponding `md5` file for authentication):

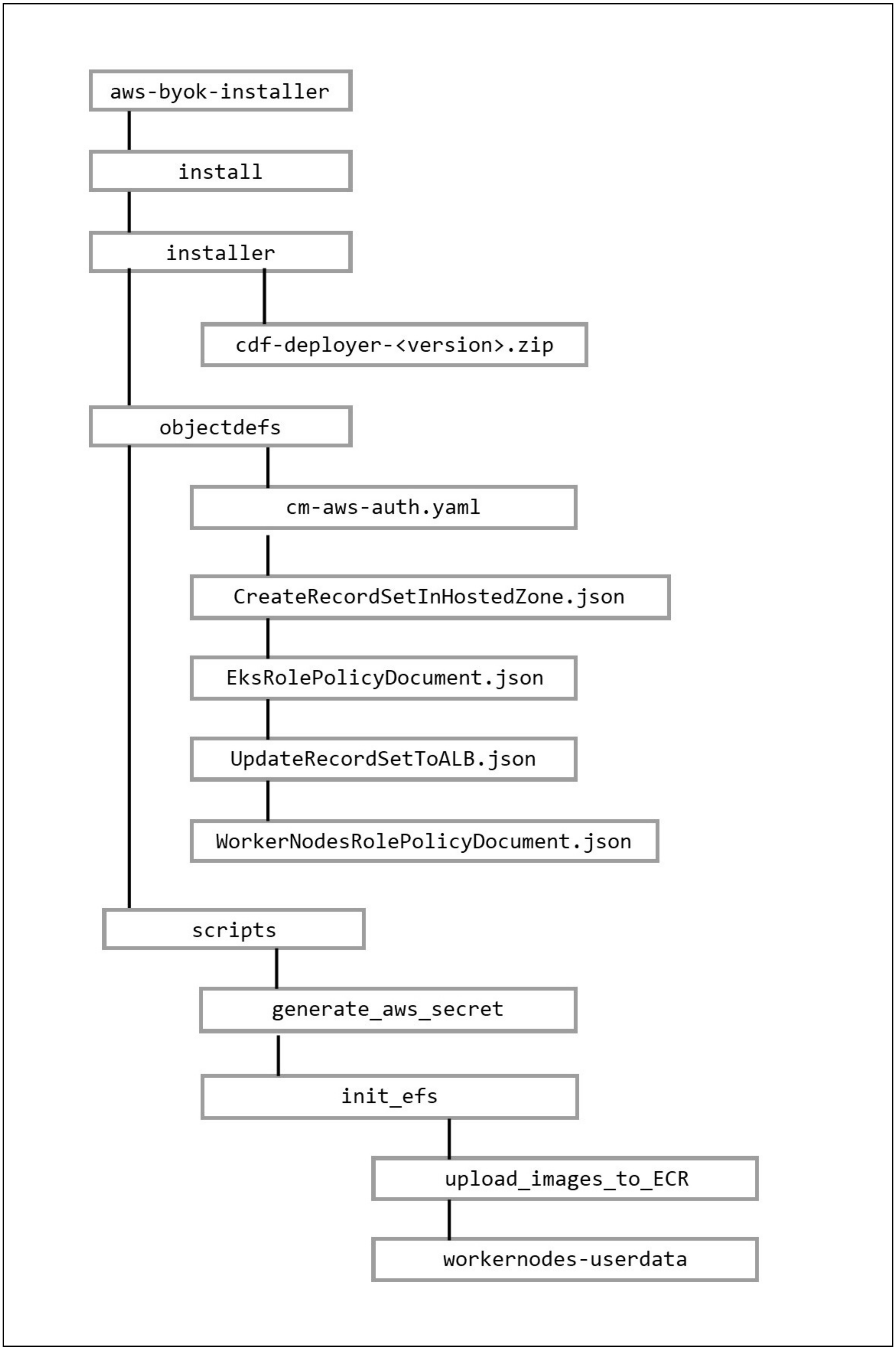
```
aws-byok-installer-<version>.zip/md5
```

```
cdf-byok-images-<version>.tar/md5
```

```
<product packages> tar/md5
```

Installation tools

The `aws-byok-installer-<version>.zip` archive contains utility scripts and some templates used during the deployment process. The structure of the `aws-byok-installer` archive is shown here:


aws-byok-installer
install
installer
cdf-deployer-<version>.zip
objectdefs
cm-aws-auth.yaml
CreateRecordSetInHostedZone.json
EksRolePolicyDocument.json
UpdateRecordSetToALB.json
WorkerNodesRolePolicyDocument.json
scripts
generate_aws_secret
init_efs
upload_images_to_ECR
workernodes-userdata

The `scripts` directory includes these scripts:

- `generate_aws_secrets`: Used to generate new Kubernetes Secrets for connecting from the cluster to the Elastic Container Registry (ECR). Generated credentials/secrets are valid only 12 hours after generation. For accessing the ECR after this timeframe, use this script according to Refresh the ECR credentials in the K8s.
- `init_efs`: Used to create the required folder structure on the Elastic File Storage (EFS) and assign correct ownership and permissions. You will use it when configuring EFS for the ArcSight Suite. Parameters for this script are discussed in the following sections. Execute this script without parameters to display the help.
- `upload_images_to_ECR`: Used for uploading the CDF and product images to the ECR to make them accessible to K8s. The script performs tasks in the background required specifically by the AWS ECR. Parameters for this script are detailed below. Execute this script without parameters to display the help.
- `workernodes-userdata` : Used indirectly for enabling worker nodes to join the Kubernetes cluster.

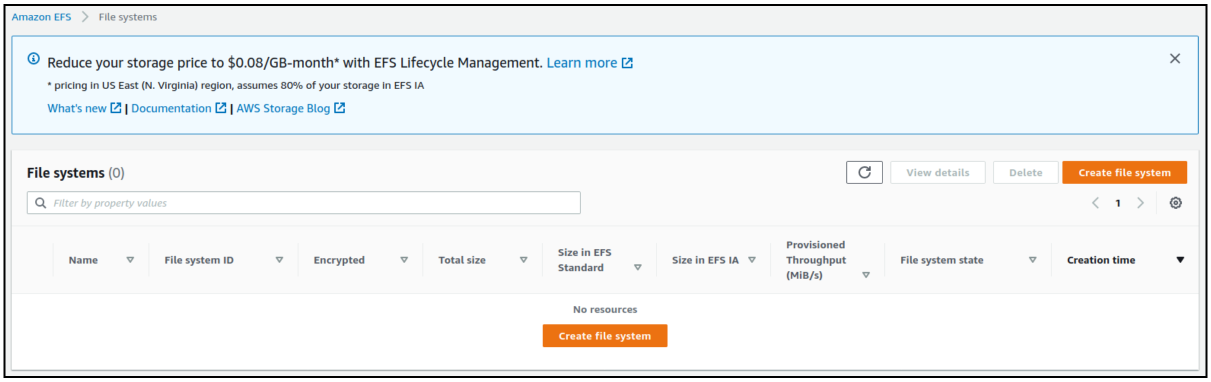
Next Step: Creating and Configuring EFS

Creating the Elastic File System (EFS)

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system. You can create an EFS through the web UI or the CLI.

To create an EFS using the web UI:

1. Using the Find Services search tool, locate and browse to the EFS dashboard.



Amazon EFS > File systems

Reduce your storage price to $0.08/GB-month* with EFS Lifecycle Management. Learn more

* pricing in US East (N. Virginia) region, assumes 80% of your storage in EFS IA

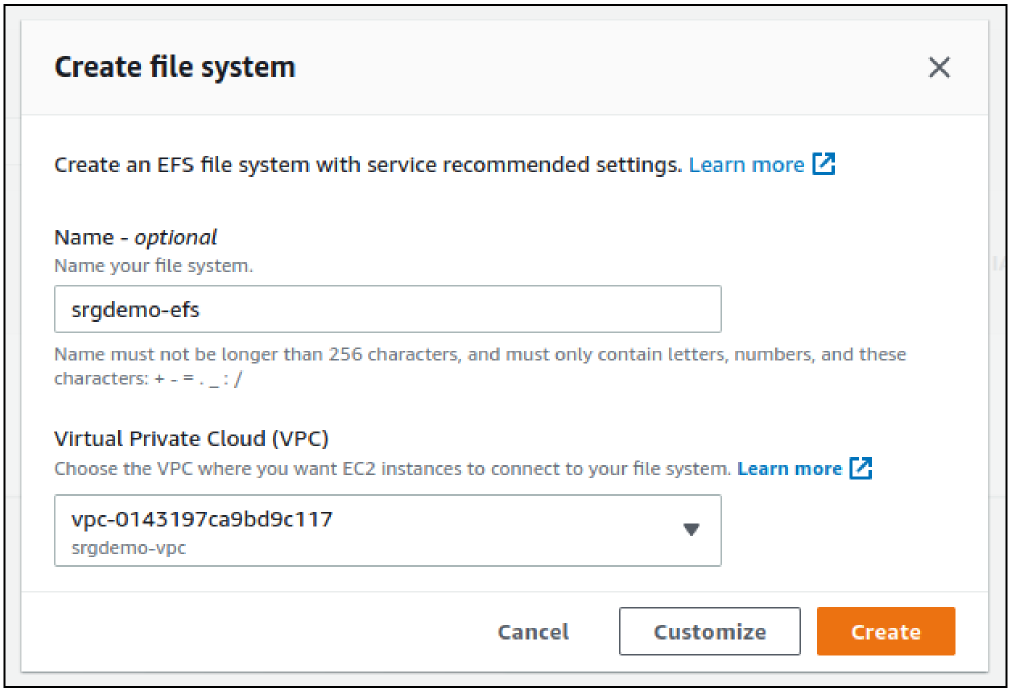
What's new | Documentation | AWS Storage Blog

File systems (0)

Name	File system ID	Encrypted	Total size	Size in EFS Standard	Size in EFS IA	Provisioned Throughput (MiB/s)	File system state	Creation time

No resources

2. Click **Create file system.**



Create file system

Create an EFS file system with service recommended settings. Learn more

Name - *optional*

Name your file system.

srgdemo-efs

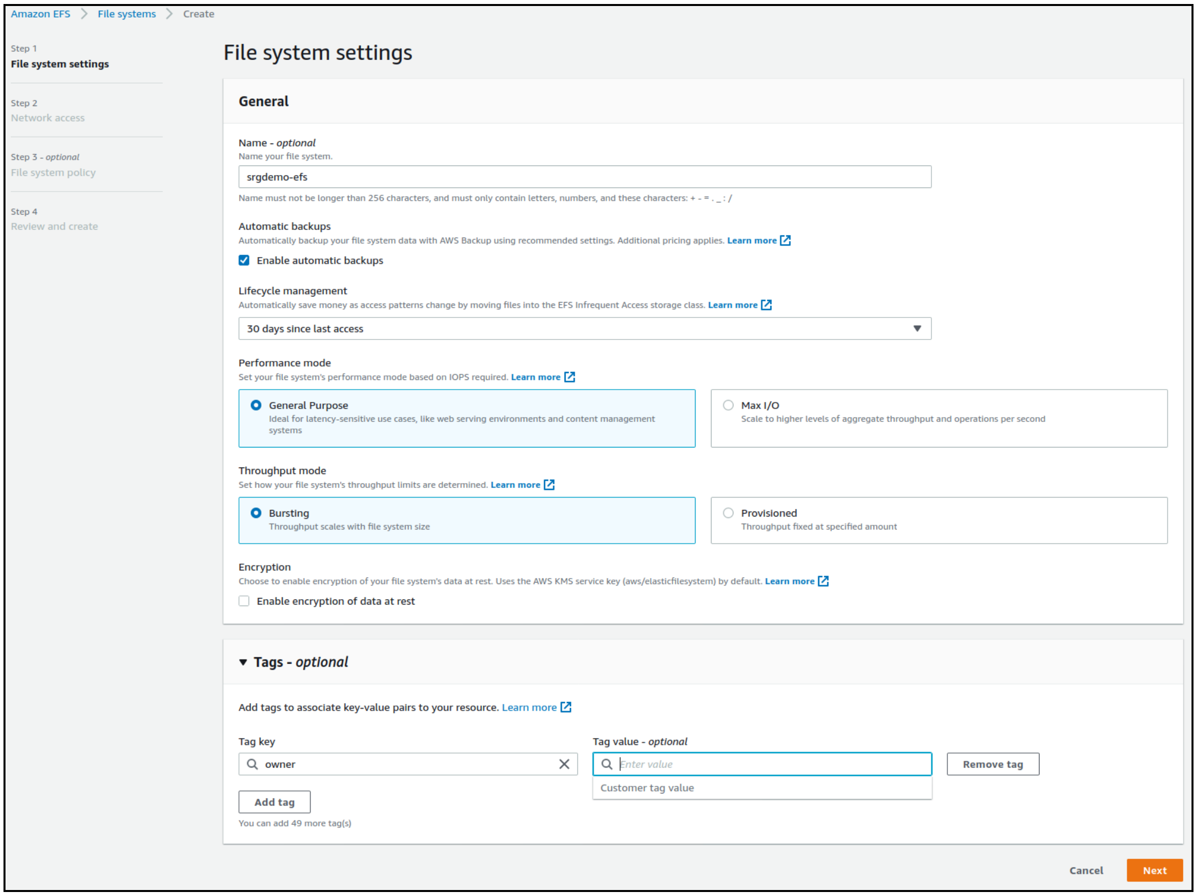
Name must not be longer than 256 characters, and must only contain letters, numbers, and these characters: + - = . _ : /

Virtual Private Cloud (VPC)

Choose the VPC where you want EC2 instances to connect to your file system. Learn more

vpc-0143197ca9bd9c117
srgdemo-vpc

3. On the **Create file system** dialog:
   a. In **Name**, enter a name for the EFS.
   b. Under **Virtual Private Cloud (VPC)**, select the VPC you recorded in your AWS worksheet.
   c. Click **Customize** to start the custom EFS wizard.
4. On the **File system settings** page:
   a. In **General**, deselect **Enable encryption of data at rest.**
   b. In **Tags**, add one or more identification tags as desired.
   c. Click **Next**.



File system settings

General

Name - *optional*
Name your file system.
srgdemo-efs
Name must not be longer than 256 characters, and must only contain letters, numbers, and these characters: + - = . _ : /

Automatic backups
Automatically backup your file system data with AWS Backup using recommended settings. Additional pricing applies. Learn more
☑ Enable automatic backups

Lifecycle management
Automatically save money as access patterns change by moving files into the EFS Infrequent Access storage class. Learn more
30 days since last access

Performance mode
Set your file system's performance mode based on IOPS required. Learn more
● General Purpose — Ideal for latency-sensitive use cases, like web serving environments and content management systems
○ Max I/O — Scale to higher levels of aggregate throughput and operations per second

Throughput mode
Set how your file system's throughput limits are determined. Learn more
● Bursting — Throughput scales with file system size
○ Provisioned — Throughput fixed at specified amount

Encryption
Choose to enable encryption of your file system's data at rest. Uses the AWS KMS service key (aws/elasticfilesystem) by default. Learn more
□ Enable encryption of data at rest

Tags - *optional*

Add tags to associate key-value pairs to your resource. Learn more

Tag key: owner
Tag value - *optional*: Enter value
Customer tag value

You can add 49 more tag(s)

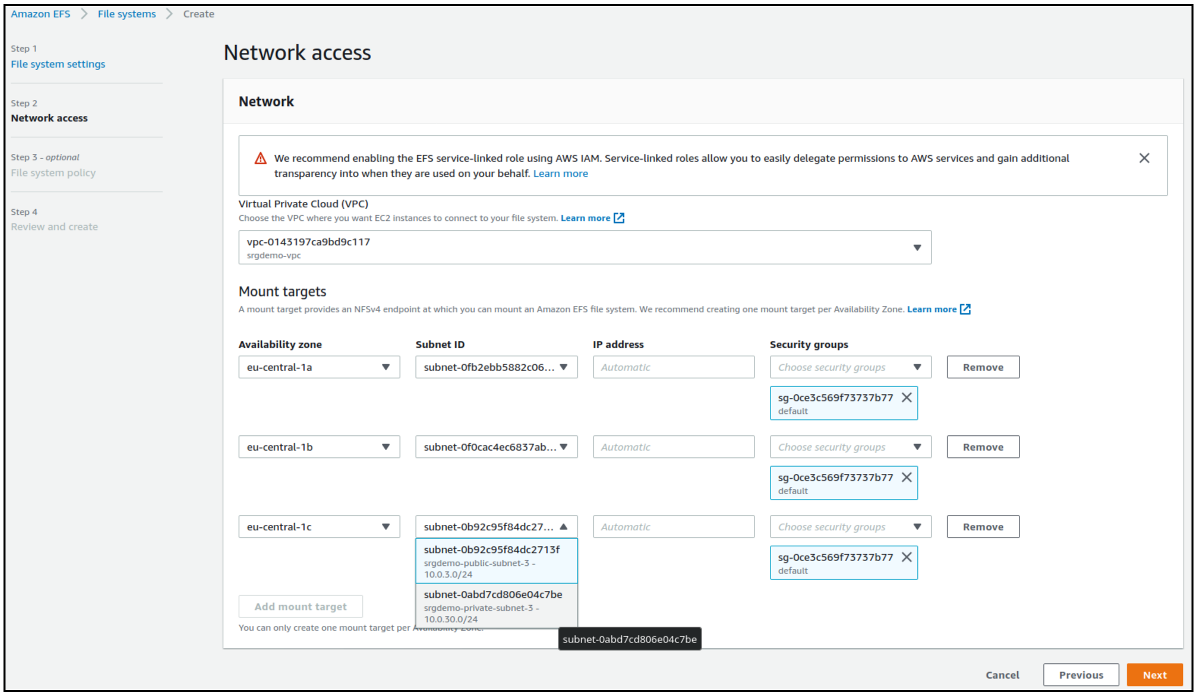
5. On the **Network Access** page:

   a. Under **Mount Targets**, for each target:

   b. Under **Subnet ID,** select the subnet ID of the private subnet.

   c. Change the **Security groups** by adding the Intra VPC security group to the dropdown, and remove any other security groups from the list.

   d. Click **Next**.



Amazon EFS > File systems > Create

Step 1
File system settings

Step 2
Network access

Step 3 - *optional*
File system policy

Step 4
Review and create

Network access

Network

We recommend enabling the EFS service-linked role using AWS IAM. Service-linked roles allow you to easily delegate permissions to AWS services and gain additional transparency into when they are used on your behalf. Learn more

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system. Learn more

vpc-0143197ca9bd9c117
srgdemo-vpc

Mount targets
A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. Learn more

Availability zone	Subnet ID	IP address	Security groups
eu-central-1a	subnet-0fb2ebb5882c06…	Automatic	sg-0ce3c569f73737b77 default
eu-central-1b	subnet-0f0cac4ec6837ab…	Automatic	sg-0ce3c569f73737b77 default
eu-central-1c	subnet-0b92c95f84dc27…	Automatic	sg-0ce3c569f73737b77 default

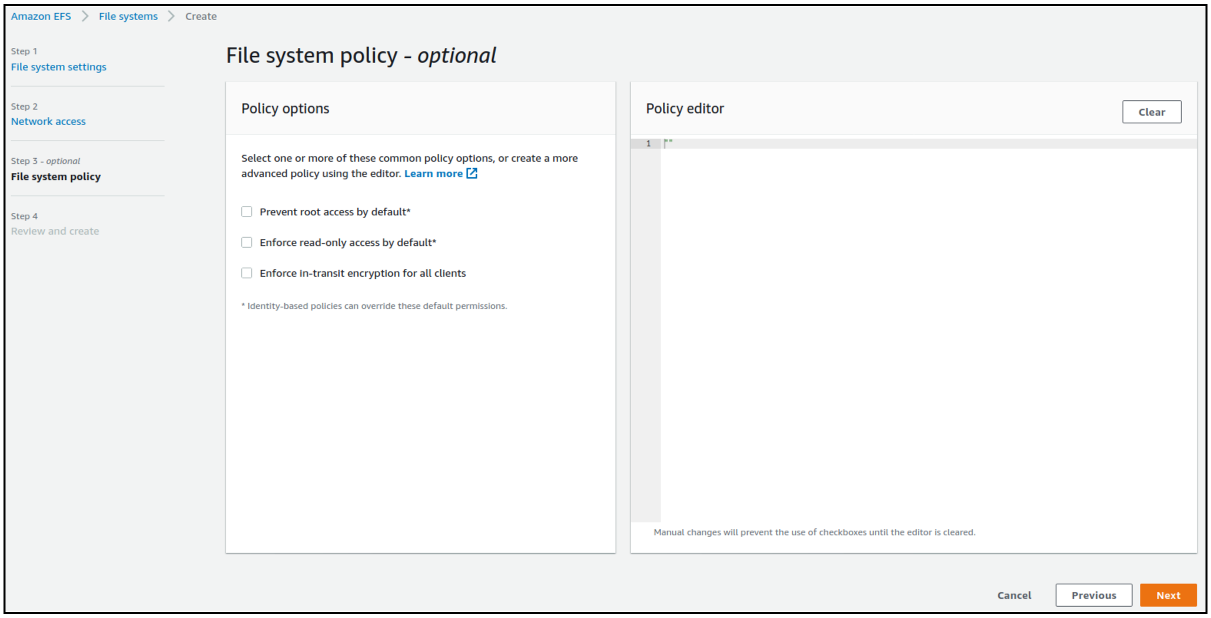
subnet-0b92c95f84dc2713f
srgdemo-public-subnet-3 - 10.0.3.0/24

subnet-0abd7cd806e04c7be
srgdemo-private-subnet-3 - 10.0.30.0/24

subnet-0abd7cd806e04c7be

You can only create one mount target per Availability Zone.

6. On the **File system policy - optional** page, leave all settings as is and click **Next**.



Amazon EFS > File systems > Create

Step 1
File system settings

Step 2
Network access

Step 3 - *optional*
File system policy

Step 4
Review and create

File system policy - *optional*

Policy options

Select one or more of these common policy options, or create a more advanced policy using the editor. Learn more

□ Prevent root access by default*

□ Enforce read-only access by default*

□ Enforce in-transit encryption for all clients

* Identity-based policies can override these default permissions.

Policy editor

Manual changes will prevent the use of checkboxes until the editor is cleared.

7. On the **Review and create** page, review all settings for accuracy, and then click **Create**. You are redirected to the EFS dashboard.
8. Record the File system ID in the AWS worksheet.

To create the EFS using the CLI:

1. Run the following command:

```
# aws efs create-file-system \
    --tags Key=Name,Value=<EFS name chosen for easy identification>
```

Example input and output:

```
#aws efs create-file-system \
--tags Key-Name, Value=srgdemo-efs
```

```
{
   "OwnerId":"115370848038",
   "CreationToken":"a53deaf5-ecd6-4dfa-9206-0e1a3db3e1d9",
   "FileSystemId":"fs-ebe456b3",
   "CreationTime":1589557528.0,
   "LifeCycleState":"creating",
   "Name":"srgdemo-efs",
   "NumberOfMountTargets":0,
   "SizeInBytes":{
      "Value":0,
      "ValueInIA":0,
      "ValueInStandard":0
   },
   "PerformanceMode":"generalPurpose",
   "Encrypted":false,
   "ThroughputMode":"bursting",
   "Tags":[
      {
         "Key":"Name",
         "Value":"srgdemo-efs"
      }
   ]
```

2. From the description, record the `FileSystemId` in the AWS worksheet.

3. Examine the value of `LifeCycleState`. Provisioning is complete when the value changes to `available`. (In the example, it has the value `creating`. To check the provisioning status while the process is running, run the following command:

   ```
   # aws efs describe-file-systems \
   --file-system-id <FileSystemId>
   ```

> Provisioning usually takes approximately 5 minutes.

Example input and output:

```
# aws efs describe-file-systems --file-system-id fs-ebe456b3
```

```
{
   "FileSystems":[
      {
         "OwnerId":"115370848038",
         "CreationToken":"a53deaf5-ecd6-4dfa-9206-0e1a3db3e1d9",
```

```
         "FileSystemId":"fs-ebe456b3",
         "CreationTime":1589557528.0,
         "LifeCycleState":"available",
         "Name":"srgdemo-efs",
         "NumberOfMountTargets":0,
         "SizeInBytes":{
            "Value":6144,
            "ValueInIA":0,
            "ValueInStandard":6144
         },
         "PerformanceMode":"generalPurpose",
         "Encrypted":false,
         "ThroughputMode":"bursting",
         "Tags":[
            {
               "Key":"Name",
               "Value":"srgdemo-efs"
            }
         ]
      }
   ]
}
```

Next Step: Create Mount Targets

Creating Mount Targets

A *mount target* connects the EFS to a specific subnet in the VPC. The instances contained in the VPC can mount the target using the NFS protocol and utilize NFS.

In this section, you will create mount targets between the newly-created EFS and all three private subnets.

To create a mount target in a private subnet:

1. Select one of your **private** subnets and run the following command:
   ```
   # aws efs create-mount-target \
   --file-system-id <FileSystemId> \
   --security-groups <Intra VPC Security group Id> \
   --subnet-id <private subnet Id>
   ```

Parameters:

`<FileSystemId>`: The file system ID of the EFS you just created

`<Intra VPC Security group Id>`: The ID of the Intra VPC security group you previously created.

> The command only accepts one subnet ID at a time. You must run this command separately for each private subnet which you are using for the cluster.

2. The command will respond with a mount target description. From the output, record the `MountTargetId` in your AWS worksheet.
3. Repeat Steps 1 and 2 for each of the other 2 private subnets (use the subnet IDs on your worksheet) and then record the values of `MountTargetId` for each in your AWS worksheet.

Example input and output:

```
# aws efs create-mount-target \
--file-system-id fs-ebe456b3 \
--security-groups sg-07b302cbc0972c603 \
--subnet-id subnet-0fb2ebb5882c061f0
```

```
{
   "OwnerId":"115370848038",
   "MountTargetId":"fsmt-63eaae3a",
   "FileSystemId":"fs-ebe456b3",
   "SubnetId":"subnet-0fb2ebb5882c061f0",
   "LifeCycleState":"creating",
   "IpAddress":"10.0.10.131",
   "NetworkInterfaceId":"eni-03ecba7e5eb46dc9f",
   "AvailabilityZoneId":"euc1-az2",
   "AvailabilityZoneName":"eu-central-1a"
}
```

To check the creation status of a mount target:

1. Run the following command:
   `# aws efs describe-mount-targets --mount-target-id <Mount target X Id>`
2. Record the value of `MountTargetId` in the AWS worksheet.

Immediately after creation, a mount target has a value for `LifeCycleState` value of `creating`. The transition to available usually takes approximately 3 minutes. To check the status, run the following command:
`# aws efs describe-mount-targets --mount-target-id <Mount target X Id>`

Example input and output:

```
# aws efs describe-mount-targets --mount-target-id fsmt-63eaae3a
```

```
{
   "OwnerId":"115370848038",
   "MountTargetId":"fsmt-63eaae3a",
   "FileSystemId":"fs-ebe456b3",
   "SubnetId":"subnet-0fb2ebb5882c061f0",
   "LifeCycleState":"creating",
   "IpAddress":"10.0.10.131",
   "NetworkInterfaceId":"eni-03ecba7e5eb46dc9f",
   "AvailabilityZoneId":"euc1-az2",
   "AvailabilityZoneName":"eu-central-1a"
}
```

Once all three mount targets are in the `available` state, you can proceed to the next step.

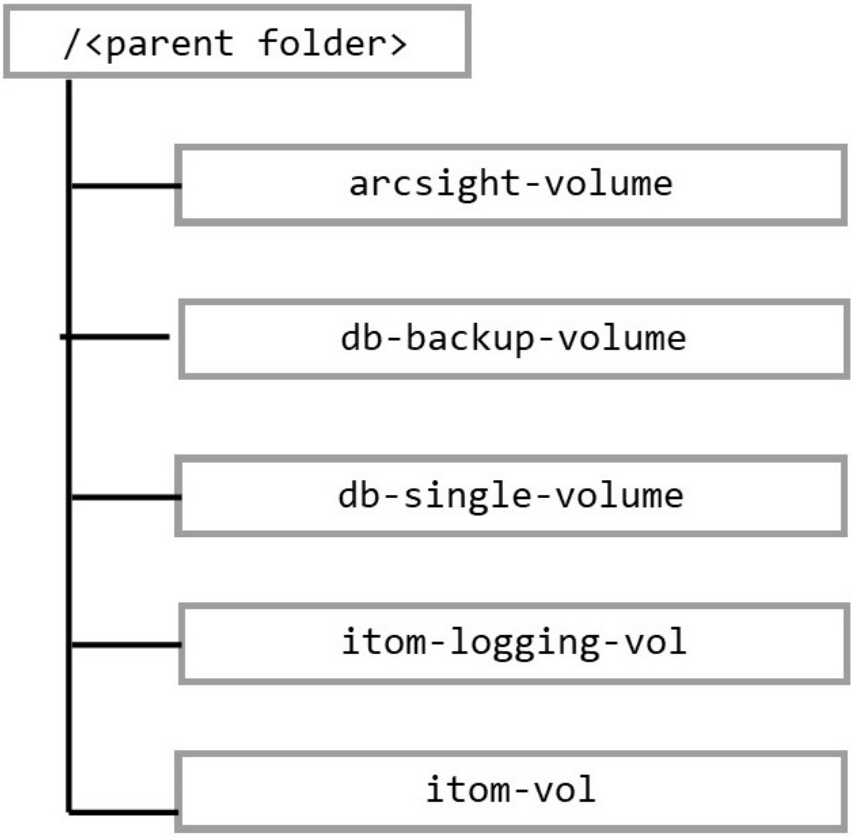
Next Step: Configuring EFS

Configuring EFS for the ArcSight Suite

CDF and the ArcSight suite require several separated folders for storing various types of information, such database files, log files, and runtime data. In this step, you will create the following folders:

- `arcsight-volume`
- `db-backup-vol`
- `db-single-vol`
- `itom-logging-vol`
- `itom-vol`

All of these folders are created in a parent folder from the filesystem, as follows:



Using different parent folders, you can use a single EFS for several different file systems (assuming they are in the same region and same VPC, and have the correct mount targets).

To configure EFS for ArcSight Suite:

1. Using an `scp` client, copy the `aws-byok-installer-<version>.zip` package to the bastion and unpack it.
2. For creating the folders and setting respective permissions, run the script `init_efs` from the `aws-byok-installer/scripts` directory.
3. Construct the filesystem FQDN. The filesystem FQDN should have the following format: `<FileSystemId>.efs.<Region>.amazonaws.com`

   Parameters:

`<FileSystemId>` : Previously created and recorded in the AWS worksheet.

`<Region>`:The ID of the region in which you have originally asked to create restricted resources.

4. Record the filesystem FQDN in the AWS worksheet.

> The FQDN will be used for initializing the folder structures; and will later be used during the Bootstrap CDF step and during CDF Web UI installation processes.

5. Execute the script:
   ```
   ./aws-byok-installer/scripts/init_efs \
   -p <Parent folder name> \
   -s <Filesystem FQDN>
   ```

Parameters:

:`<Parent folder name>` An optional parameter. If not specified, this value will be replaced with `Arcsight`. Record the chosen parent folder name to the AWS worksheet.

`<Filesystem FQDN>`: The filesystem FQDN you have just created.

Example:

```
# ./aws-byok-installer/scripts/init_efs \
-p srgdemo \
-s fs-ebe456b3.efs.eu-central-1.amazonaws.com
```

6. Verify the created folders correspond to the structure described above, with respect to your chosen parent folder.

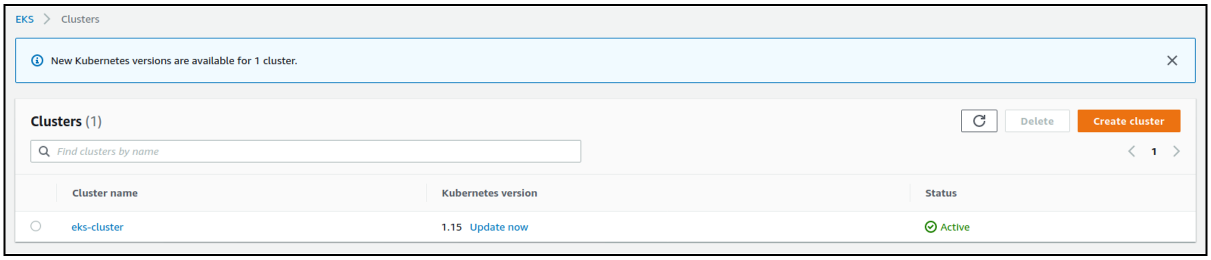
Next Step: Configure EKS

Configuring the Elastic Kubernetes Service (EKS)

Amazon Elastic Kubernetes Service (Amazon EKS) is a fully-managed Kubernetes service control plane. In this section you will set up your EKS cluster.

To configure EKS using the web UI:

1. Using the Find Services search tool, locate and browse to the EKS Dashboard.
2. Click **Create Cluster.**



New Kubernetes versions are available for 1 cluster.

Clusters (1)

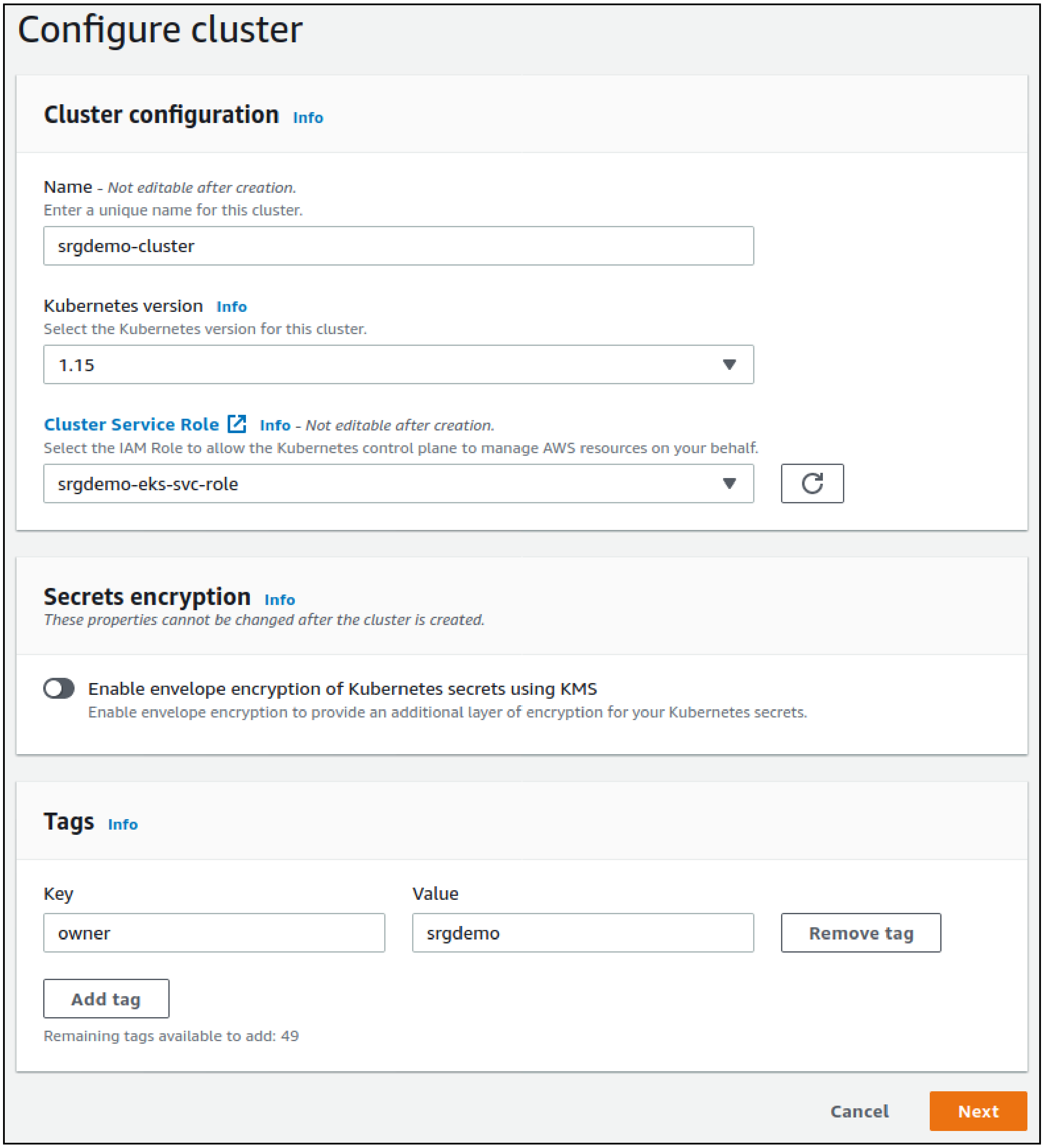
	Cluster name	Kubernetes version	Status
○	eks-cluster	1.15 Update now	Active

3. On the **Configure Cluster** page, enter values for the following:
   a. **Name**: Cluster name. Use the same value you passed in your resources creation request to your AWS infrastructure administrators, and recorded in the AWS worksheet. For example, `srgdemo-cluster`.

b. **Kubernetes**: Generally the version of Kubernetes to specify depends on the CDF requirements. Use the value recorded in the AWS worksheet.

a. **Cluster Service Role:** Select the role specified for cluster management.

b. **Tags:** Tags are optional, but you might add tags to identify the cluster.



Configure cluster

Cluster configuration Info

Name - *Not editable after creation.*
Enter a unique name for this cluster.
srgdemo-cluster

Kubernetes version Info
Select the Kubernetes version for this cluster.
1.15

Cluster Service Role Info - *Not editable after creation.*
Select the IAM Role to allow the Kubernetes control plane to manage AWS resources on your behalf.
srgdemo-eks-svc-role

Secrets encryption Info
These properties cannot be changed after the cluster is created.

○ Enable envelope encryption of Kubernetes secrets using KMS
Enable envelope encryption to provide an additional layer of encryption for your Kubernetes secrets.

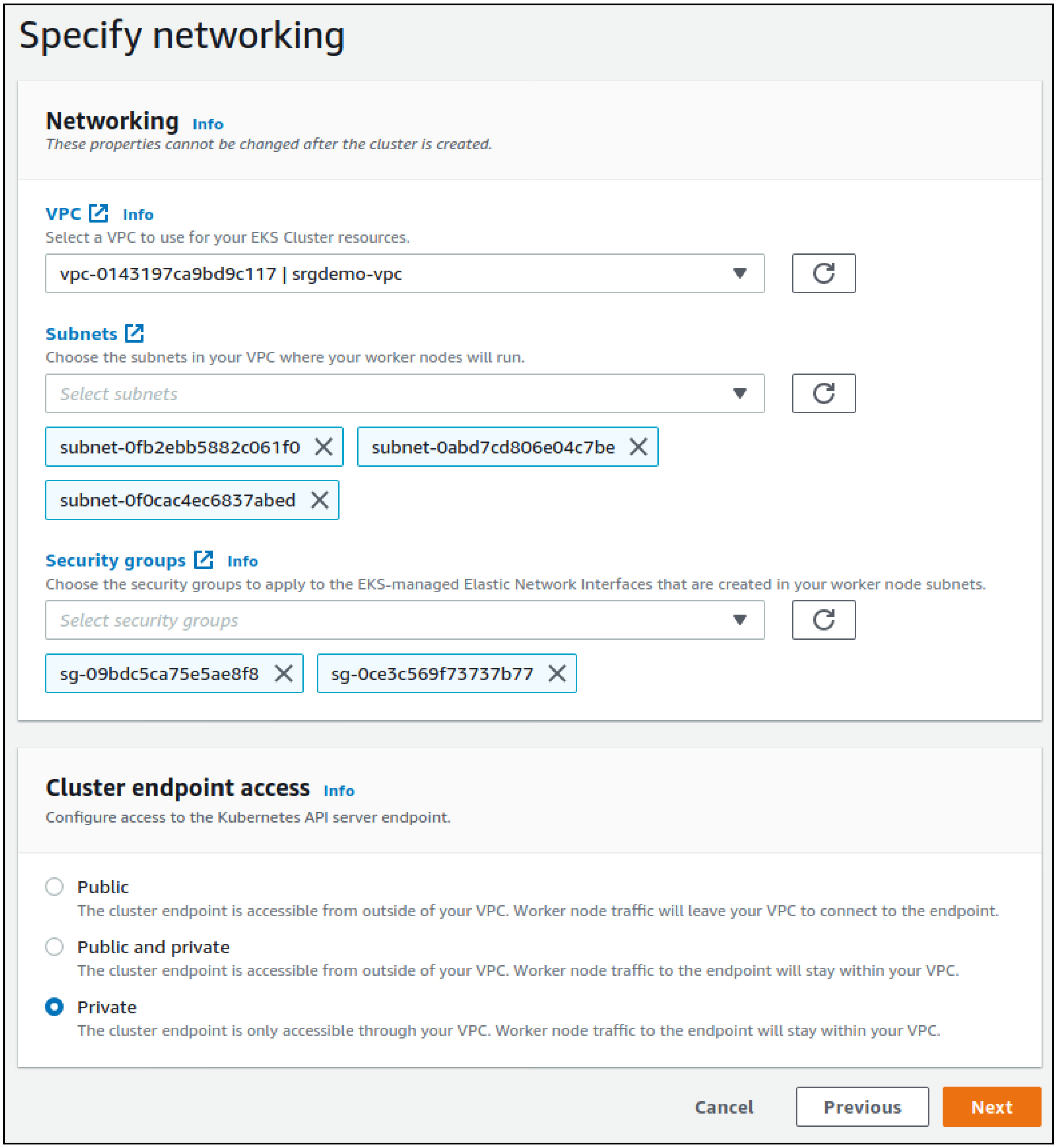
Tags Info

Key	Value
owner	srgdemo

Remaining tags available to add: 49

4. Click **Next.**

5. On the **Specify Networking** page, enter values for the following:
   a. **VPC:** Select your VPC from the dropdown.
   b. **Subnets:** Make sure **only** your private subnets are selected from the dropdown (subnet names are recorded in the AWS worksheet).
   c. **Security groups:** Add the Intra VPC security group named in the AWS worksheet.
   d. **Cluster endpoint access:** Select **Private** to keep the cluster isolated.



Specify networking

Networking

These properties cannot be changed after the cluster is created.

VPC

Select a VPC to use for your EKS Cluster resources.

vpc-0143197ca9bd9c117 | srgdemo-vpc

Subnets

Choose the subnets in your VPC where your worker nodes will run.

Select subnets

subnet-0fb2ebb5882c061f0

subnet-0abd7cd806e04c7be

subnet-0f0cac4ec6837abed

Security groups

Choose the security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your worker node subnets.

Select security groups

sg-09bdc5ca75e5ae8f8

sg-0ce3c569f73737b77

Cluster endpoint access

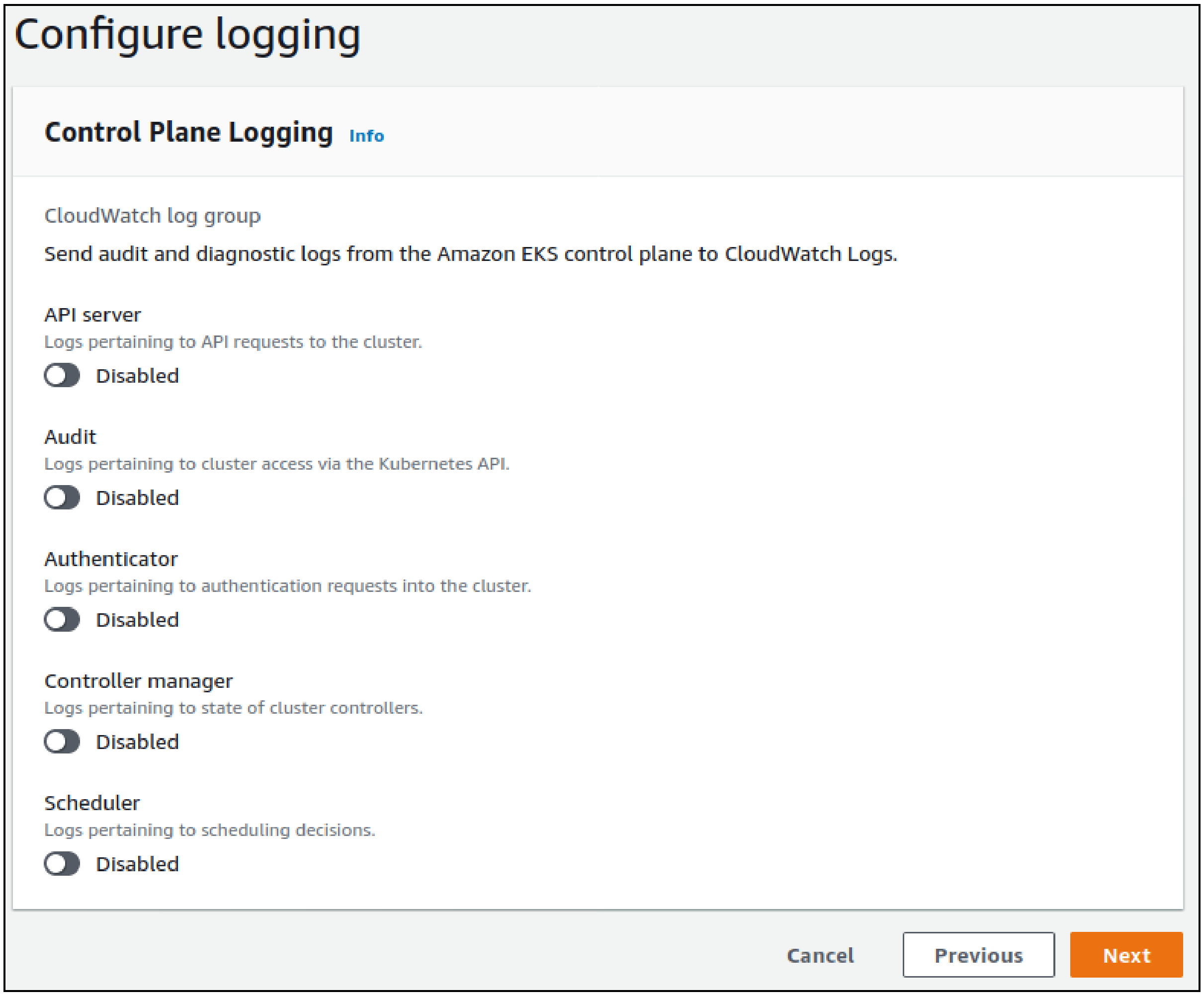
Configure access to the Kubernetes API server endpoint.

○ Public
The cluster endpoint is accessible from outside of your VPC. Worker node traffic will leave your VPC to connect to the endpoint.

○ Public and private
The cluster endpoint is accessible from outside of your VPC. Worker node traffic to the endpoint will stay within your VPC.

● Private
The cluster endpoint is only accessible through your VPC. Worker node traffic to the endpoint will stay within your VPC.

6. On the **Configure Logging** page, leave all values to default settings, and then click **Next**.



Configure logging

Control Plane Logging Info

CloudWatch log group

Send audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs.

API server
Logs pertaining to API requests to the cluster.
Disabled

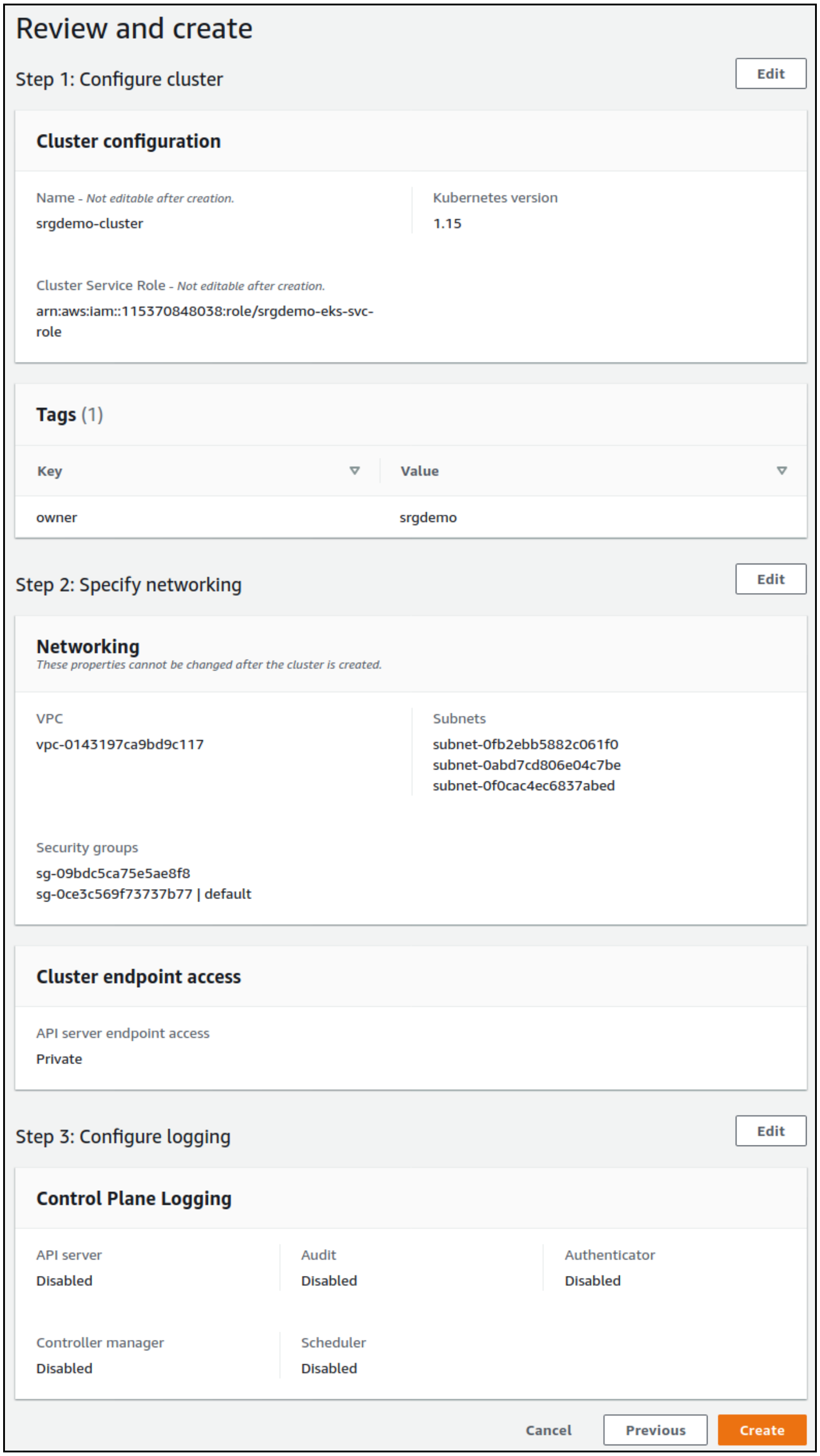
Audit
Logs pertaining to cluster access via the Kubernetes API.
Disabled

Authenticator
Logs pertaining to authentication requests into the cluster.
Disabled

Controller manager
Logs pertaining to state of cluster controllers.
Disabled

Scheduler
Logs pertaining to scheduling decisions.
Disabled

7. On the **Review and Create** page, check all settings for accuracy and then click **Create**. The cluster details are displayed.



Review and create

Step 1: Configure cluster

Cluster configuration

Name - *Not editable after creation.*
srgdemo-cluster

Kubernetes version
1.15

Cluster Service Role - *Not editable after creation.*
arn:aws:iam::115370848038:role/srgdemo-eks-svc-role

Tags (1)

Key	Value
owner	srgdemo

Step 2: Specify networking

Networking
These properties cannot be changed after the cluster is created.

VPC
vpc-0143197ca9bd9c117

Subnets
subnet-0fb2ebb5882c061f0
subnet-0abd7cd806e04c7be
subnet-0f0cac4ec6837abed

Security groups
sg-09bdc5ca75e5ae8f8
sg-0ce3c569f73737b77 | default

Cluster endpoint access

API server endpoint access
Private

Step 3: Configure logging

Control Plane Logging

API server
Disabled

Audit
Disabled

Authenticator
Disabled

Controller manager
Disabled

Scheduler
Disabled

The creation process usually takes approximately 20 minutes and the status will change to *Active* when complete. Click **Refresh** to refresh the creation status display.

To configure EKS using the CLI:

1. Run the following command:
   ```
   # aws eks create-cluster \
   --name <Cluster Name> \
   --role-arn <EKS role ARN> \
   --resources-vpc-config subnetIds=<private subnet
   Ids>,endpointPublicAccess=false,endpointPrivateAccess=true,securityGroupId
   s=<Intra VPC Security group Id> \
   --kubernetes-version <Kubernetes version>
   ```

Parameters:

`<Cluster Name>`: The cluster name you have chosen during VPC creation; check the AWS worksheet for the value.

`<EKS role>:` The IAM role ARN created by your AWS infrastructure administrators; check the AWS worksheet for the value.

`<private subnet Ids>:` Comma-separated IDs of private subnets created together with the VPC; these values are recored in the AWS worksheet.

`<Intra VPC Security group Id>`: The ID of the previously created security group

> The value for `resources-vpc-config` cannot contain spaces; it must be one string.

`<Kubernetes version>:` Use the value from the AWS worksheet.

2. Record the ARN value in your AWS worksheet.

Example input and output:

```
# aws eks create-cluster \
--name srgdemo-cluster \
--role-arn arn:aws:iam::115370848038:role/srgdemo-eks-svc-role \
--resources-vpc-config subnetIds=subnet-0fb2ebb5882c061f0,subnet-
0f0cac4ec6837abed,subnet-0abd7cd806e04c7be,\
endpointPublicAccess=false,endpointPrivateAccess=true,securityGroupIds=sg-
09bdc5ca75e5ae8f8 \
--kubernetes-version 1.17
```

```
{
   "cluster":{
      "name":"srgdemo-cluster",
      "arn":"arn:aws:eks:eu-central-1:115370848038:cluster/srgdemo-cluster",
      "createdAt":1589877429.005,
      "version":"1.15",
```

```
      "roleArn":"arn:aws:iam::115370848038:role/srgdemo-eks-svc-role",
      "resourcesVpcConfig":{
         "subnetIds":[
            "subnet-0fb2ebb5882c061f0",
            "subnet-0f0cac4ec6837abed",
            "subnet-0abd7cd806e04c7be"
         ],
         "securityGroupIds":[
            "sg-09bdc5ca75e5ae8f8"
         ],
         "vpcId":"vpc-0143197ca9bd9c117",
         "endpointPublicAccess":false,
         "endpointPrivateAccess":true,
         "publicAccessCidrs":[

         ]
      },
      "logging":{
         "clusterLogging":[
            {
               "types":[
                  "api",
                  "audit",
                  "authenticator",
                  "controllerManager",
                  "scheduler"
               ],
               "enabled":false
            }
         ]
      },
      "status":"CREATING",
      "certificateAuthority":{

      },
      "platformVersion":"eks.2",
      "tags":{

      }
   }
}
```

Cluster creation usually takes approximately 20 minutes. Check the cluster status by running the command:

```
# aws eks describe-cluster \
--name <Cluster Name> \
| jq '.cluster.status'
```

The output immediately after creation should state `CREATING.` Repeat the command until the output changes to `ACTIVE.`

> A newly created EKS might take up to 20 minutes to become `ACTIVE`.

Example:

```
# aws eks describe-cluster \
--name srgdemo-cluster \
| jq '.cluster.status'
```

Next Steps: Configure `kubectl`

Configuring the Kubernetes Client (`kubectl`)

Several Kubernetes configuration and diagnostic tasks using `kubectl` will be performed on the bastion. In order to do that, the `kubectl` utility needs be configured with bastion credentials.

To configure `kubectl`:

1. Connect to the bastion instance and run the following command:
   ```
   # aws eks update-kubeconfig \
   --name <Cluster Name>
   ```
2. The command will return:
   ```
   Updated context <eks cluster arn> in /home/centos/.kube/config
   ```

Example:

```
# aws eks update-kubeconfig \
--name srgdemo-cluster
```

```
Updated context arn:aws:eks:eu-central-1:115370848038:cluster/srgdemo-cluster
in /home/centos/.kube/config
```

3. On the bastion, check the Kubernetes service status by running:
   ```
   # kubectl get svc
   ```

Example output:

```
NAME         TYPE         CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE

kubernetes   ClusterIP    172.20.0.1    <none>         443/TCP    54m
```

The EKS control plane is now ready and accessible from the bastion.

Next Step: Applying the AWS Config Map

Applying the AWS ConfigMap to Enable Worker Nodes to Join the Cluster

The AWS ConfigMap needs to be applied so that the worker nodes can join your EKS cluster.

To apply the ConfigMap:

1. Connect to the bastion host.
2. Open the file the `cm-aws-auth.yaml` in any text editor. (The file is from the unpacked `aws-byok-installer-<version>.zip` located in the directory `aws-byok-installer-<version>/objectdefs`).
3. Replace the placeholder `${WORKERS_ROLE_ARN}` with the `Role ARN` value from your AWS worksheet, and then save your changes. The ConfigMap will then resemble the following example:

```
apiVersion: v1
kind: ConfigMap
metadata:
name: aws-auth
namespace: kube-system
data:
mapRoles: |
- rolearn: arn:aws:iam::115370848038:role/srgdemo-workernodes-svc-role
username: system:node:{{EC2PrivateDNSName}}
groups:
- system:bootstrappers
- system:nodes
```

4. On the bastion, run the following command:
   `# kubectl apply -f cm-aws-auth.yaml`
5. This command will output:
   `configmap/aws-auth created.`

Next Step: Create and Configure Worker Nodes

Creating and Configuring Worker Nodes

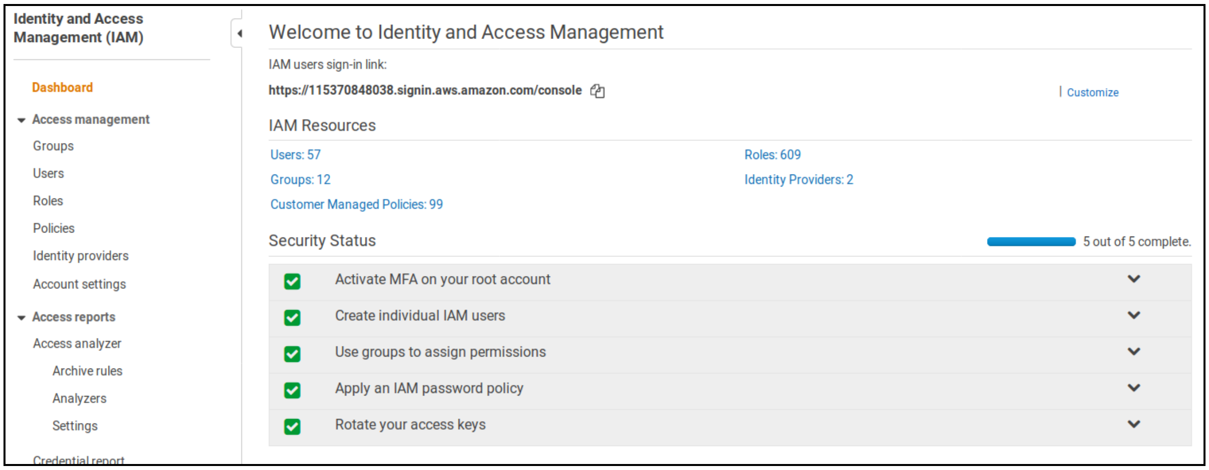
The *worker nodes* (EC2 nodes) are the Kubernetes nodes that will perform application processing. A cluster contains one or more Amazon EC2 nodes on which pods are scheduled. Amazon EKS nodes run in your AWS account and connect to your cluster's control plane through the cluster API server endpoint.

Next Step: Check for Worker Node Instance Profile

Checking for a Worker Node Instance Profile

To check for a worker node instance profile using the web UI:

1. Using the Find Services search tool, locate and browse to the IAM dashboard.
2. In the left navigation panel, under **Access management,** click **Roles** to get a list of existing roles.



Welcome to Identity and Access Management

IAM users sign-in link:

https://115370848038.signin.aws.amazon.com/console

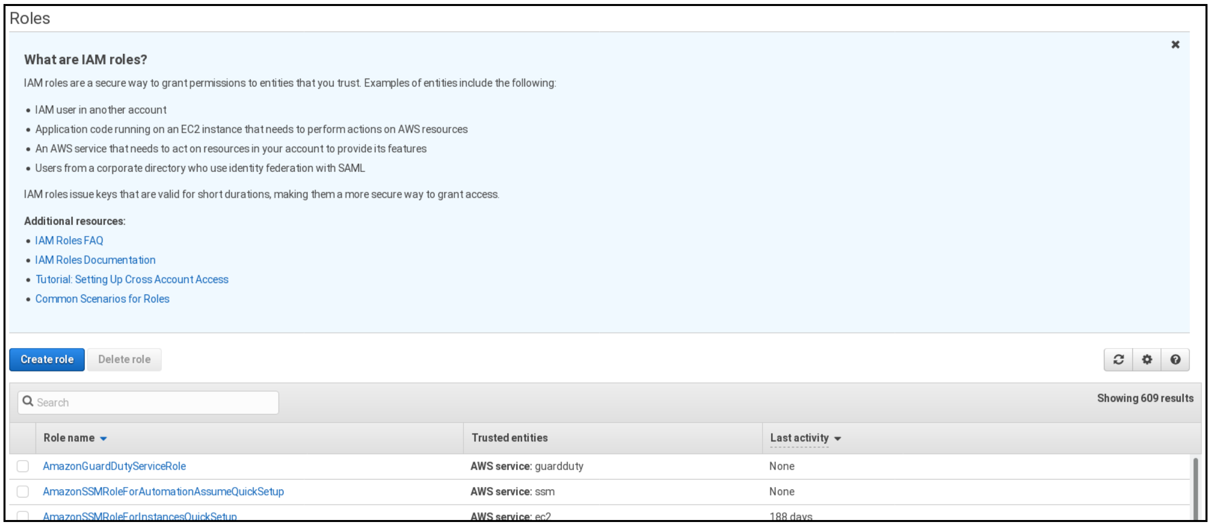
IAM Resources

Users: 57	Roles: 609
Groups: 12	Identity Providers: 2
Customer Managed Policies: 99	

Security Status — 5 out of 5 complete.

- ☑ Activate MFA on your root account
- ☑ Create individual IAM users
- ☑ Use groups to assign permissions
- ☑ Apply an IAM password policy
- ☑ Rotate your access keys

3. In the search box, enter the Worker Nodes role name (from the AWS worksheet) to filter it from the other roles.



Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

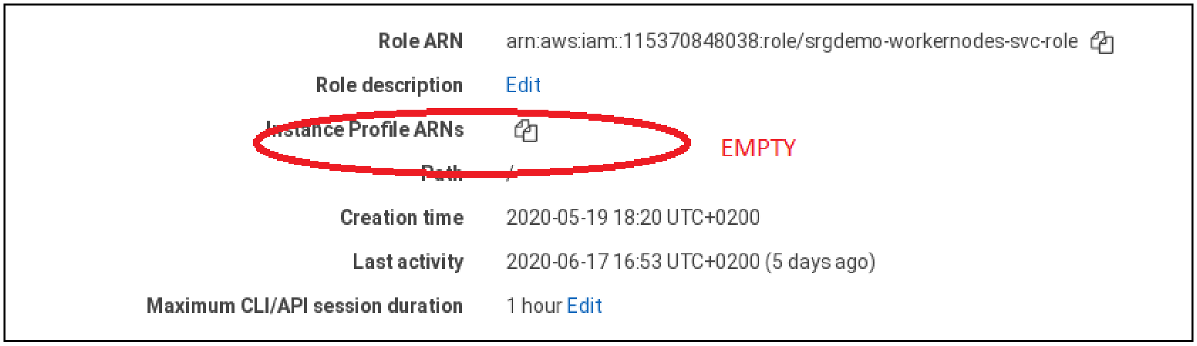
Additional resources:

- IAM Roles FAQ
- IAM Roles Documentation
- Tutorial: Setting Up Cross Account Access
- Common Scenarios for Roles

Showing 609 results

	Role name	Trusted entities	Last activity
□	AmazonGuardDutyServiceRole	**AWS service:** guardduty	None
□	AmazonSSMRoleForAutomationAssumeQuickSetup	**AWS service:** ssm	None
□	AmazonSSMRoleForInstancesQuickSetup	**AWS service:** ec2	188 days

4. Click on the role name to get its details and then check the row **Instance Profile ARNs.**



Role ARN	arn:aws:iam::115370848038:role/srgdemo-workernodes-svc-role
Role description	Edit
Instance Profile ARNs	
Path	/
Creation time	2020-05-19 18:20 UTC+0200
Last activity	2020-06-17 16:53 UTC+0200 (5 days ago)
Maximum CLI/API session duration	1 hour Edit

a. If no instance profile has been assigned to the role (that is, the row Instance Profile is empty, as illustrated here) then continue with creating an instance profile.

b. If the row **Instance Profile ARNs** is filled, record the value in the AWS worksheet.

> The Instance Profile creation guide only works on the command-line interface. It is not possible to create a separate instance profile without an assigned role.

To check for a worker node instance profile using the CLI:

1. Run the following command:
```
# aws iam list-instance-profiles-for-role \
--role-name <Workernodes role name from AWS worksheet>
```
2. Example:
```
# aws iam list-instance-profiles-for-role \
--role-name ARST-EKS-Workers-Custom-Role
```
3. The command will reply with one of the following cases.

   a. No instance profile for the role exists. Example output for this case:

```
{
   "InstanceProfiles":[

   ]
}
```

b. An instance profile exists for the role. Example output for this case:

```
{
   "InstanceProfiles":[
      {
         "Path":"/",
         "InstanceProfileName":"ARST-EKS-Workers-Custom-Role",
         "InstanceProfileId":"AIPARVXFDN4TBQBCRKX45",
         "Arn":"arn:aws:iam::115370848038:instance-profile/ARST-EKS-Workers-
Custom-Role",
         "CreateDate":"2020-06-16T05:57:59+00:00",
         "Roles":[
            {
               "Path":"/",
               "RoleName":"ARST-EKS-Workers-Custom-Role",
               "RoleId":"AROARVXFDN4TNRSAMVCVX",
               "Arn":"arn:aws:iam::115370848038:role/ARST-EKS-Workers-Custom-
Role",
               "CreateDate":"2020-06-16T05:57:58+00:00",
               "AssumeRolePolicyDocument":{
                  "Version":"2012-10-17",
                  "Statement":[
                     {
                        "Effect":"Allow",
                        "Principal":{
                           "Service":"ec2.amazonaws.com"
                        },
                        "Action":"sts:AssumeRole"
                     }
                  ]
               }
            }
         ]
      }
   ]
}
```

4. Do one of the following:
   - *If no instance profile exists for the role,* proceed with creating an instance profile, OR,

- *If the Instance profile already exists for the role,* record its name (`InstanceProfiles -> InstanceProfileName`) and ARN (`InstanceProfile -> Arn`) in the AWS worksheet, and then continue with the procedure to create a launch configuration.

Next Step: Create an Instance Profile

Creating an Instance Profile

1. Run the following command name:
   ```
   # aws iam create-instance-profile \
   --instance-profile-name <Workernodes Instance profile name>
   ```
2. Record your assigned workernodes instance profile name in the AWS worksheet. In our example we will use `srgdemo-workernodes-instance-profile`.
3. The command will return a description of the newly-created instance profile. For example:

```
{
   "InstanceProfile":{
      "InstanceProfileId":"AIPAJMBYC7DLSPEXAMPLE",
      "Roles":[

      ],
      "CreateDate":"2015-03-09T20:33:19.626Z",
      "InstanceProfileName":"Webserver",
      "Path":"/",
      "Arn":"arn:aws:iam::123456789012:instance-profile/Webserver"
   }
}
```

4. Record the Arn value in the AWS Worksheet as `Workernodes Instance profile ARN.`

To add the role to the instance profile:

1. Run the following command:
   ```
   # aws iam add-role-to-instance-profile \
   --instance-profile-name <Workernodes Instance profile name> \
   --role-name <Workernodes role name>
   ```

Parameters:

`<Workernodes Instance profile name>`: Use the instance profile name created above or by your AWS infrastructure administrators and recorded on the AWS worksheet; for example, `srgdemo-workernodes-instance-profile.`

`<Workernodes role name>:` Use the role name created by your AWS infrastructure administrators and recorded on the AWS worksheet; for example, `srgdemo-workernodes-svc-role.`

Example:

```
# aws iam add-role-to-instance-profile --instance-profile-name srgdemo-
workernodes-instance-profile --role-name srgdemo-workernodes-svc-role
```

The command has no output.

Next Step: Create and Configure a Launch Configuration

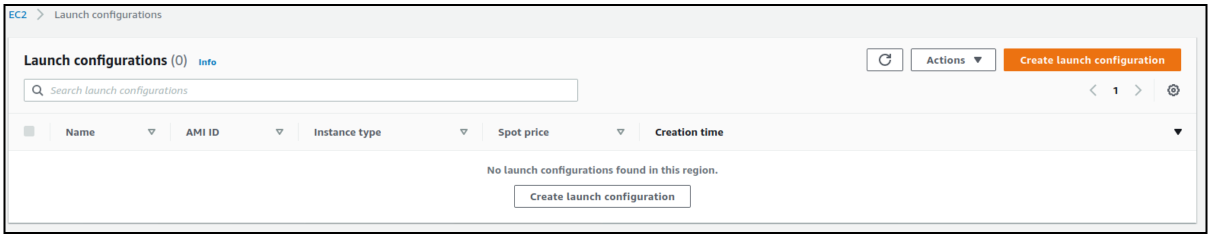
Creating and Configuring a Launch Configuration

A *launch configuration* is an instance configuration template that an Auto Scaling group uses to launch EC2 instances.

- Creating a launch configuration requires collecting some infrastructure data, specifically the Amazon Machine Image ID (AMI ID) and instance type.
- You can have more than one launch configuration created with different parameters, such as instance type or root volume size, and instantiate the auto-scaling groups from them.

To create a launch configuration using the Web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Auto Scaling**, select **Launch Configurations.**
3. On the **Launch Configurations** page, click **Create Launch Configuration.**



EC2 > Launch configurations

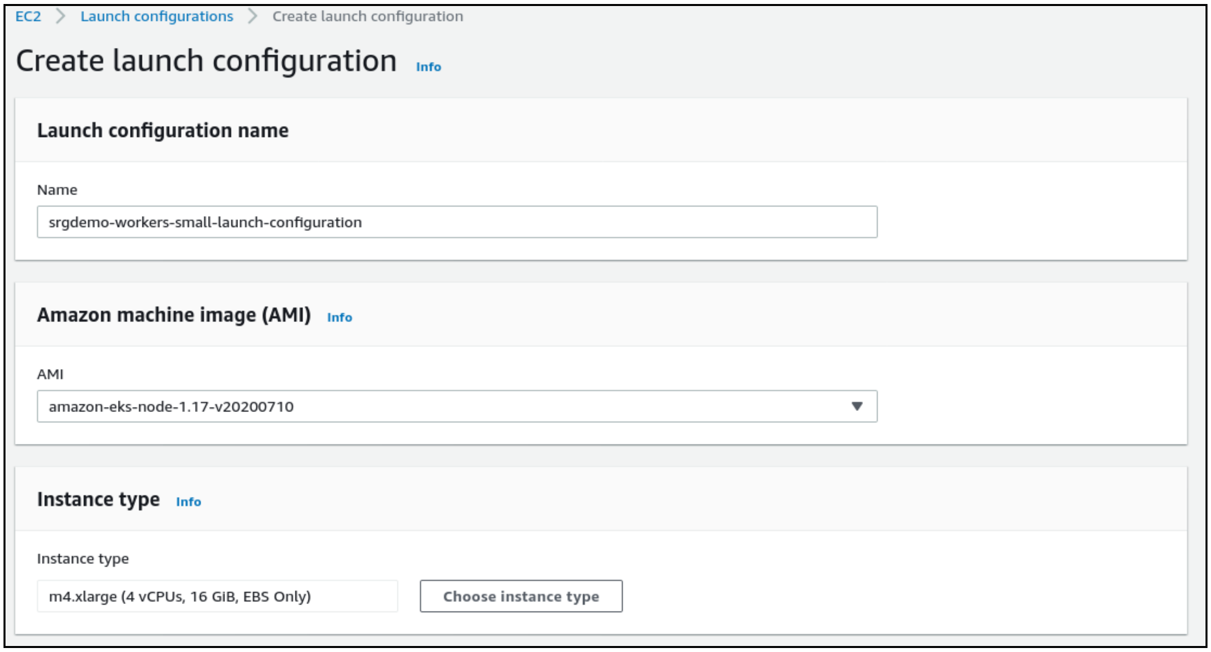
Launch configurations (0) Info

Search launch configurations

Name	AMI ID	Instance type	Spot price	Creation time

No launch configurations found in this region.

4. On the **Create Launch Configuration** page, enter values for the following:
   a. **Launch configuration name:** Specify your launch configuration name.
   b. **Virtual machines HW and OS:** Specify the virtual machine hardware and installed operating system.



Create launch configuration

Launch configuration name

Name: srgdemo-workers-small-launch-configuration

Amazon machine image (AMI)

AMI: amazon-eks-node-1.17-v20200710

Instance type

Instance type: m4.xlarge (4 vCPUs, 16 GiB, EBS Only)

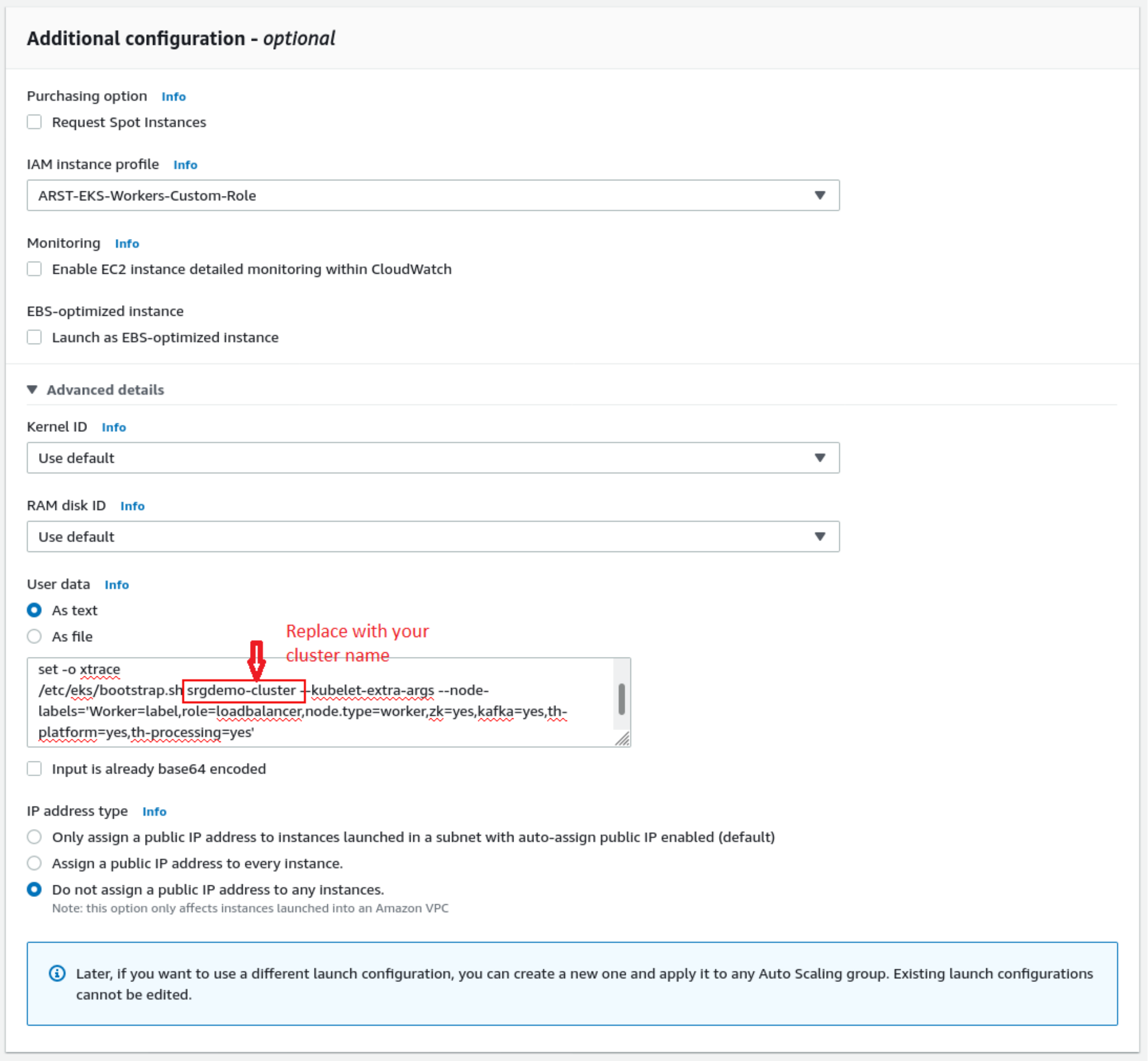
5. Under **Additional Configuration Details - optional**, enter values for the following (mandatory) settings:

   a. **IAM Instance profile:** Choose the instance profile name for the worker nodes from the AWS worksheet.

   b. **User data:** Leave as text, and then copy the contents of the script `workernodes-userdata` located in the `aws-byok-installer-<version>/scripts/ folder`. In the text area, replace the `<cluster name>` with your own cluster name from the AWS worksheet.

   c. **IP Address Type:** select **Do not assign a public IP address to any instances.**

> Also, notice the labels: `zk:yes, kafka:yes, th-platform:yes`, and `th-processing:yes`. Each created node will automatically receive the respective labels required for running Transformation Hub (for more information about labels, see Labeling Nodes).



Additional configuration - *optional*

Purchasing option Info
□ Request Spot Instances

IAM instance profile Info
ARST-EKS-Workers-Custom-Role

Monitoring Info
□ Enable EC2 instance detailed monitoring within CloudWatch

EBS-optimized instance
□ Launch as EBS-optimized instance

▼ Advanced details

Kernel ID Info
Use default

RAM disk ID Info
Use default

User data Info
● As text
○ As file

Replace with your cluster name

```
set -o xtrace
/etc/eks/bootstrap.sh srgdemo-cluster --kubelet-extra-args --node-labels='Worker=label,role=loadbalancer,node.type=worker,zk=yes,kafka=yes,th-platform=yes,th-processing=yes'
```

□ Input is already base64 encoded

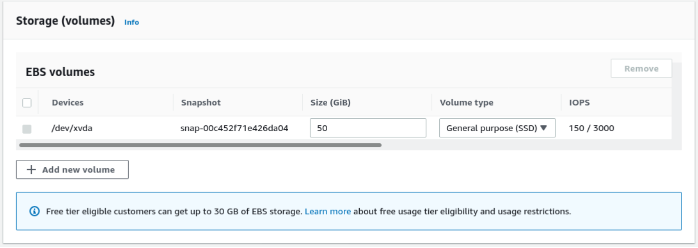
IP address type Info
○ Only assign a public IP address to instances launched in a subnet with auto-assign public IP enabled (default)
○ Assign a public IP address to every instance.
● Do not assign a public IP address to any instances.
Note: this option only affects instances launched into an Amazon VPC

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

6. Under **Storage**, configure the storage for each node. Enter the following settings:

   a. **Size (GiB):** Change the size to minimum 50GiB or more depending on your plans for products installation and load.

   b. Scroll to the right and select **Delete on termination.**



Storage (volumes) Info

EBS volumes

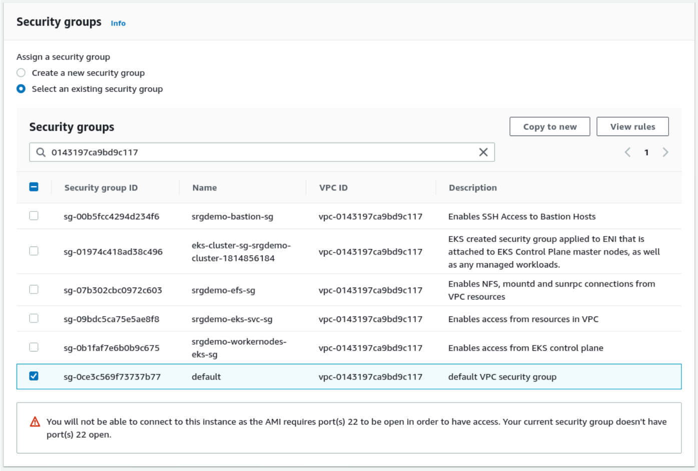
□	Devices	Snapshot	Size (GiB)	Volume type	IOPS
□	/dev/xvda	snap-00c452f71e426da04	50	General purpose (SSD) ▼	150 / 3000

Free tier eligible customers can get up to 30 GB of EBS storage. Learn more about free usage tier eligibility and usage restrictions.

7. Each resource will need to be accessible through the network, worker nodes must have the correct security groups assigned to them.

    a. Under **Security Groups**, choose **Select an existing security group.**

    b. Choose the security group created for intra-VPC communications, recorded in the AWS worksheet.



Security groups Info

Assign a security group

○ Create a new security group

● Select an existing security group

Security groups

0143197ca9bd9c117

☐	Security group ID	Name	VPC ID	Description
☐	sg-00b5fcc4294d234f6	srgdemo-bastion-sg	vpc-0143197ca9bd9c117	Enables SSH Access to Bastion Hosts
☐	sg-01974c418ad38c496	eks-cluster-sg-srgdemo-cluster-1814856184	vpc-0143197ca9bd9c117	EKS created security group applied to ENI that is attached to EKS Control Plane master nodes, as well as any managed workloads.
☐	sg-07b302cbc0972c603	srgdemo-efs-sg	vpc-0143197ca9bd9c117	Enables NFS, mountd and sunrpc connections from VPC resources
☐	sg-09bdc5ca75e5ae8f8	srgdemo-eks-svc-sg	vpc-0143197ca9bd9c117	Enables access from resources in VPC
☐	sg-0b1faf7e6b0b9c675	srgdemo-workernodes-eks-sg	vpc-0143197ca9bd9c117	Enables access from EKS control plane
☑	sg-0ce3c569f73737b77	default	vpc-0143197ca9bd9c117	default VPC security group

⚠ You will not be able to connect to this instance as the AMI requires port(s) 22 to be open in order to have access. Your current security group doesn't have port(s) 22 open.

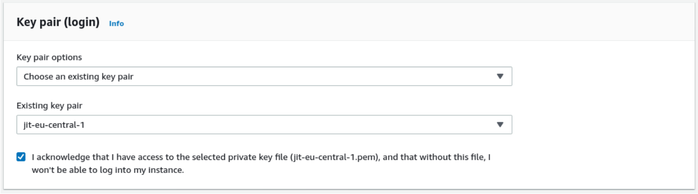
8. Create a key pair to allow cluster access as well as SSH access to the worker nodes.

9. Under **Key pair (login),** choose values for the following:

    a. **Key pair options:** Select Choose an existing key pair

    b. **Existing key pair:** Choose your key pair for which you own the private part

    c. **I acknowledge... :** this needs to be selected in order to proceed with launch configuration creation.



Key pair (login) Info

Key pair options

Choose an existing key pair

Existing key pair

jit-eu-central-1

☑ I acknowledge that I have access to the selected private key file (jit-eu-central-1.pem), and that without this file, I won't be able to log into my instance.

10. Click **Create launch configuration** to create the new launch configuration.

To create a launch configuration using the CLI:

1. Run the following command:

```
aws autoscaling create-launch-configuration \
 --launch-configuration-name <Launch Configuration name> \
 --image-id <Launch config AMI Id> \ --key-name <Key pair name> \
 --security-groups <Intra VPC Security group Id> \
 --instance-type <Instance type> \
 --block-device-mappings "<block device mapping>" \
 --iam-instance-profile <Workernodes Instance profile ARN> \
 --no-associate-public-ip-address \
    --user-data file://<user data filename>
```

Parameters:

`<Launch Configuration name>`: Choose a name which helps with easier identification; in our examples we will use srgdemo-workers-launch-config. Record the chosen value in your AWS worksheet.

`<Launch config AMI Id>`: Run the following command to get actual AMI ID.

```
# aws ec2 describe-images \--filters "Name=architecture,Values=x86_64"
"Name=name,Values=amazon-eks-node-1.17*" | jq '.Images | map(select
((.Description!=null) and (.Description | contains("GPU") | not) and
(.ImageLocation | contains("gpu") | not))) | sort_by(.CreationDate) | [last]'
```

> **Important!** Replace the value `1.17` with the same Kubernetes version number you used to create the EKS, and recorded as `Kubernetes version` in your AWS worksheet.**Retain the asterisk** at the end of the value for the name filter.

`<Key pair name>`: Use the key pair name from the AWS worksheet.

`<Intra VPC Security group Id>`: ID of Intra VPC security group recorded in the AWS worksheet.

`<Instance type>`: From Amazon EC2 Instance Types, choose your machine hardware configuration. Consider CPU, RAM, storage type, network performance, and price.
DO NOT USE the same type you used for the bastion.

`<block device mapping>`: Pass the value `DeviceName=<root device name>,Ebs={VolumeSize=<root volume size>,VolumeType=gp2,DeleteOnTermination=true}` where:

- `<root device name>`: In the description obtained for AMI ID above, locate the value for key `RootDeviceName.`
- `<root volume size>`: Size depends on planned installation size; 50GB for SMALL, 100GB for MEDIUM, 256GB for LARGE; sizes refer to the sizes from ArcSight Suite metadata definition. Size is in gigabytes.

`<instance profile ARN>`: use the instance profile ARN you created above or created by your AWS infrastructure administrators and recorded in your AWS worksheet.

`--user-data file://<user data filename>`: This file needs to be modified before executing the command. Copy the file `workernodes-userdata` located in `aws-byok-installer-<version>/scripts/` to the current working directory on the local host or on the bastion, and then edit. There is a parameter cluster name which needs to be replaced; use the value from the AWS worksheet.

Example of workernodes-userdata:

```
#!/bin/bash
```

```
set -o xtrace
```

```
/etc/eks/bootstrap.sh srgdemo-cluster --kubelet-extra-args --node-labels='Worker=label,role=loadbalancer,node.type=worker,zk:yes,kafka:yes,th-platform:yes,th-processing:yes'
```

Extending labels, as shown here, during the launch configuration creation will be important for scaling up the cluster later. However, if you do not assign labels automatically, you must manually add new labels every time you add a new node.

> You can also extend the labels assigned to the worker nodes by adding the Transformation Hub required labels. Use with care! Keep in mind that all worker nodes created from this launch configuration will automatically receive that set of labels. In some cases, this might be unwanted behavior.

Example:

```
# aws autoscaling create-launch-configuration \
--launch-configuration-name srgdemo-workers-launch-config \
--image-id ami-025291add34df213c \
--key-name srgdemo \
--security-groups sg-0ce3c569f73737b77 \
--instance-type m4.xlarge --block-device-mappings "DeviceName=/dev/xvda,Ebs={VolumeSize=15,VolumeType=gp2,DeleteOnTermination=true}" \
--iam-instance-profile arn:aws:iam::115370848038:instance-profile/srgdemo-workernodes-instance-profile \
--no-associate-public-ip-address --user-data file://workernodes-userdata
```

Next Step: Create the Auto Scaling Group

Creating the AWS Auto Scaling Group

AWS Auto Scaling enables you to build scaling plans that automate how groups of different resources respond to changes in demand. You can optimize availability, costs, or a balance of both.

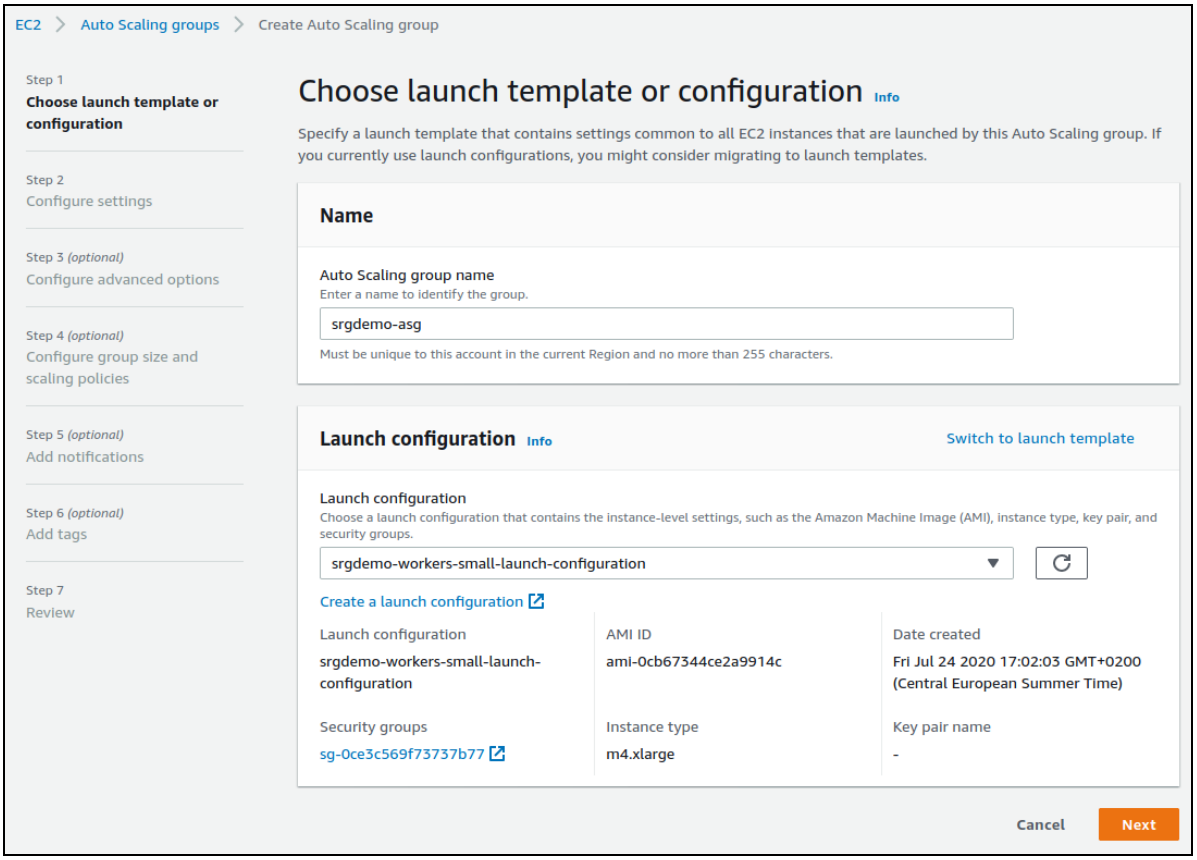
> Before proceeding, verify that the correct tag has been assigned to the VPC.

To create the Auto Scaling group using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation panel, under **Auto Scaling**, select **Auto Scaling Groups.**
3. On the landing page, click **Create Auto Scaling Group** to launch the creation wizard.

> If no Auto Scaling groups have been created yet, the introduction to Auto Scaling implementation in AWS is displayed.

4. On the **Choose Launch template or configuration** page, enter values for the following:
   a. **Name:** Enter a descriptive name for the Auto Scaling group.
   b. **Launch configuration:** Click **Switch to Launch configuration**, and then select the launch configuration you created previously.



EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template or configuration
Step 2
Configure settings
Step 3 (optional)
Configure advanced options
Step 4 (optional)
Configure group size and scaling policies
Step 5 (optional)
Add notifications
Step 6 (optional)
Add tags
Step 7
Review

Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

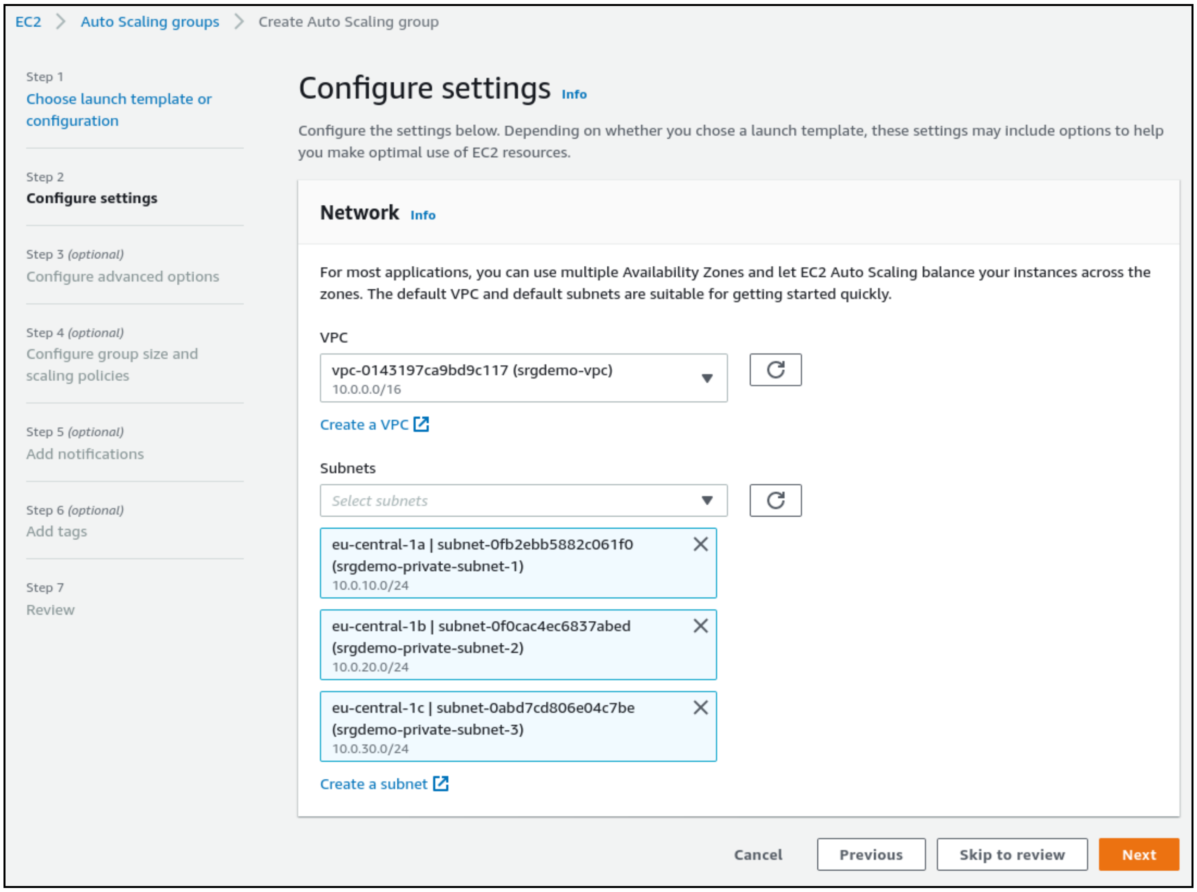
Auto Scaling group name
Enter a name to identify the group.
srgdemo-asg
Must be unique to this account in the current Region and no more than 255 characters.

Launch configuration Info — Switch to launch template

Launch configuration
Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
srgdemo-workers-small-launch-configuration
Create a launch configuration

Launch configuration	AMI ID	Date created
srgdemo-workers-small-launch-configuration	ami-0cb67344ce2a9914c	Fri Jul 24 2020 17:02:03 GMT+0200 (Central European Summer Time)
Security groups	**Instance type**	**Key pair name**
sg-0ce3c569f73737b77	m4.xlarge	-

5. Click **Next**.
6. On the **Configure settings** page, enter values for the following:
   a. **VPC:** Choose the previously-created VPC recorded in the AWS worksheet.
   b. **Subnets:** Select all 3 **private** subnets (recorded on the AWS worksheet).



EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template or configuration
Step 2
Configure settings
Step 3 (optional)
Configure advanced options
Step 4 (optional)
Configure group size and scaling policies
Step 5 (optional)
Add notifications
Step 6 (optional)
Add tags
Step 7
Review

Configure settings Info

Configure the settings below. Depending on whether you chose a launch template, these settings may include options to help you make optimal use of EC2 resources.

Network Info

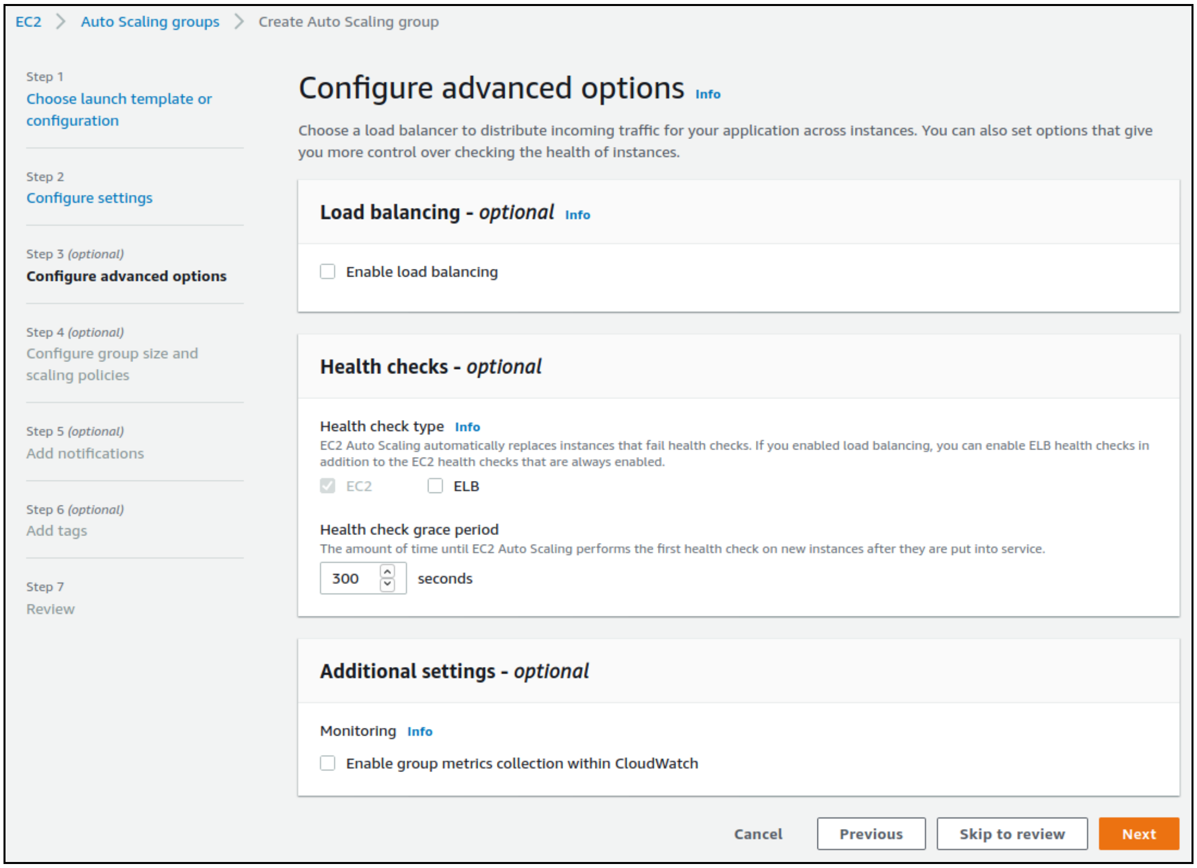
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC: vpc-0143197ca9bd9c117 (srgdemo-vpc) 10.0.0.0/16

Subnets: Select subnets

- eu-central-1a | subnet-0fb2ebb5882c061f0 (srgdemo-private-subnet-1) 10.0.10.0/24
- eu-central-1b | subnet-0f0cac4ec6837abed (srgdemo-private-subnet-2) 10.0.20.0/24
- eu-central-1c | subnet-0abd7cd806e04c7be (srgdemo-private-subnet-3) 10.0.30.0/24

7. Click **Next**.
8. On the **Configure advanced options** page, leave all values at default and click **Next**.



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template or configuration

Step 2
Configure settings

Step 3 *(optional)*
Configure advanced options

Step 4 *(optional)*
Configure group size and scaling policies

Step 5 *(optional)*
Add notifications

Step 6 *(optional)*
Add tags

Step 7
Review

Configure advanced options Info

Choose a load balancer to distribute incoming traffic for your application across instances. You can also set options that give you more control over checking the health of instances.

Load balancing - *optional* Info

□ Enable load balancing

Health checks - *optional*

Health check type Info
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

☑ EC2 □ ELB

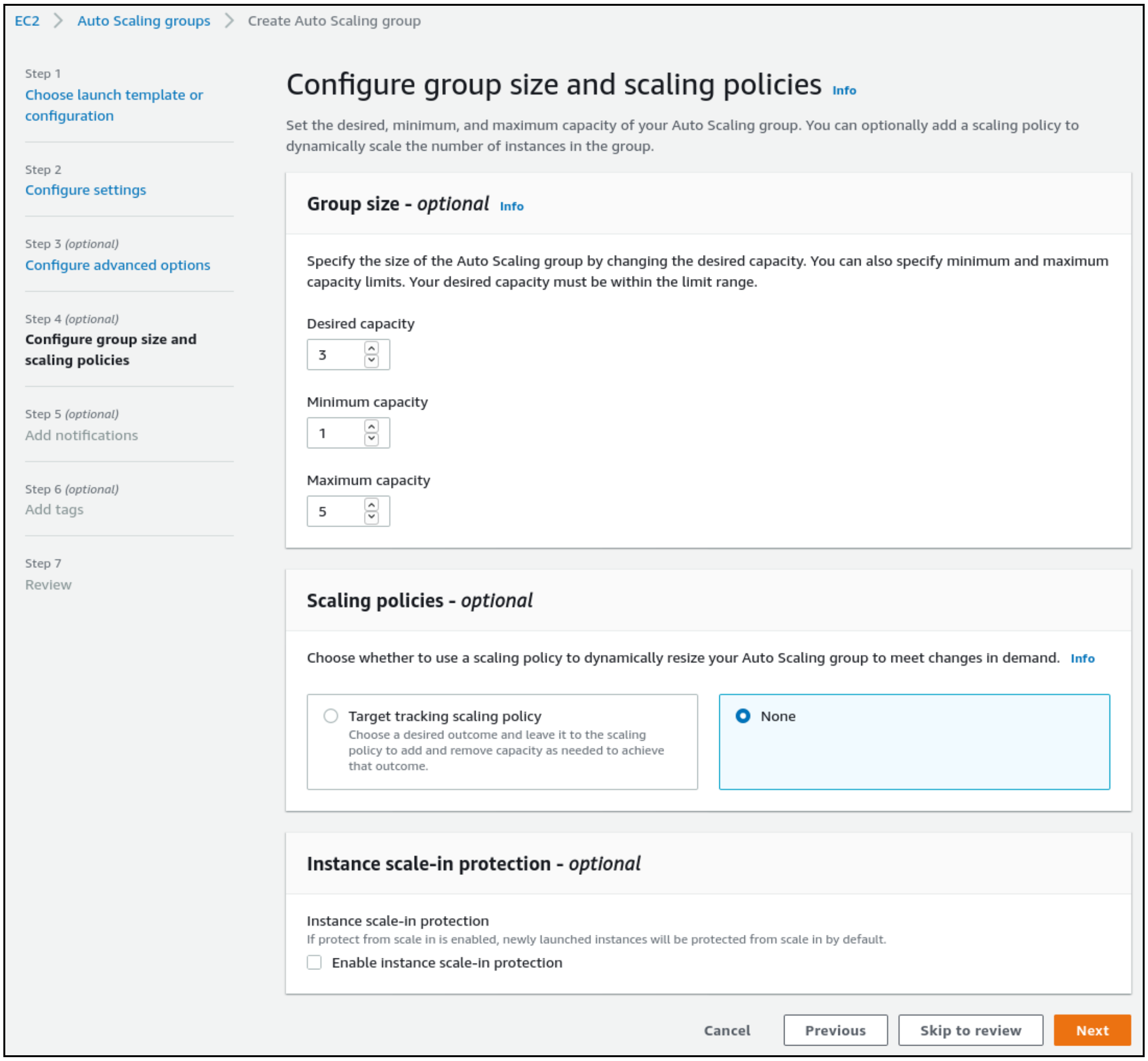
Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

300 seconds

Additional settings - *optional*

Monitoring Info
□ Enable group metrics collection within CloudWatch

9. On the **Configure group size and scaling policies** page, set values for **Desired capacity** and **Maximum capacity**. In the example shown here, there will initially be 3 nodes and enough space to instantiate 2 more by simply increasing the desired capacity. (For a production cluster, the minimum capacity setting should not be less than 2.)



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template or configuration
Step 2
Configure settings
Step 3 (optional)
Configure advanced options
Step 4 (optional)
Configure group size and scaling policies
Step 5 (optional)
Add notifications
Step 6 (optional)
Add tags
Step 7
Review

Configure group size and scaling policies Info

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

Group size - *optional* Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity: 3

Minimum capacity: 1

Maximum capacity: 5

Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

○ Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

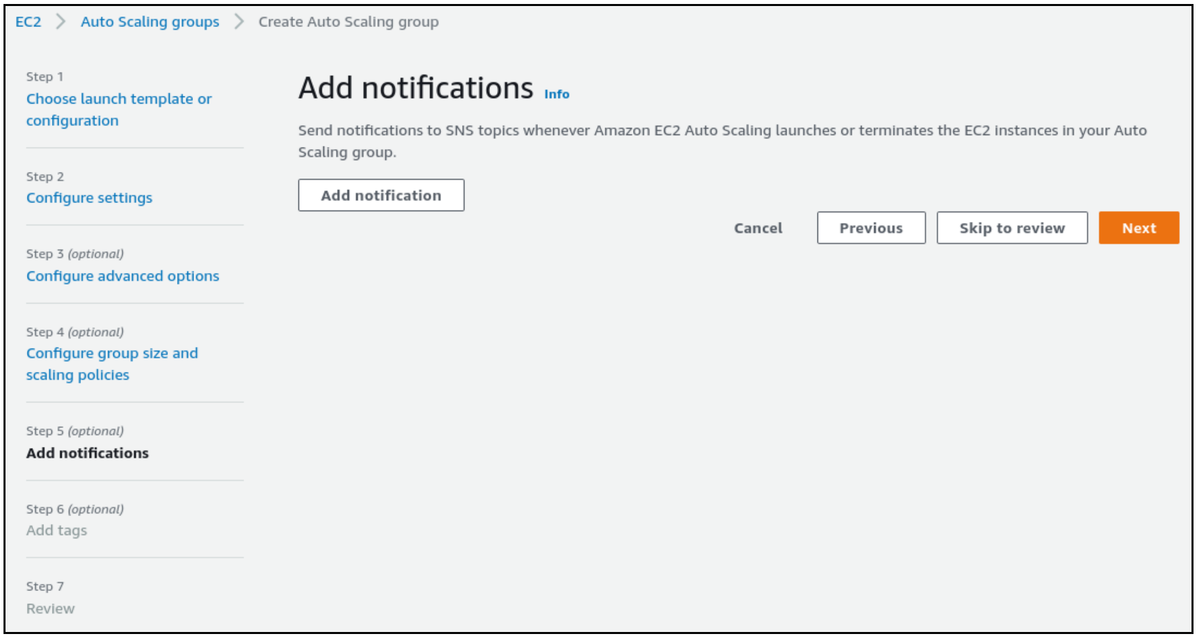
● None

Instance scale-in protection - *optional*

Instance scale-in protection
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

□ Enable instance scale-in protection

10. Click **Next**.
11. On the **Add Notifications** page, ignore all settings and click **Next**.



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template or configuration

Step 2
Configure settings

Step 3 *(optional)*
Configure advanced options

Step 4 *(optional)*
Configure group size and scaling policies

Step 5 *(optional)*
Add notifications

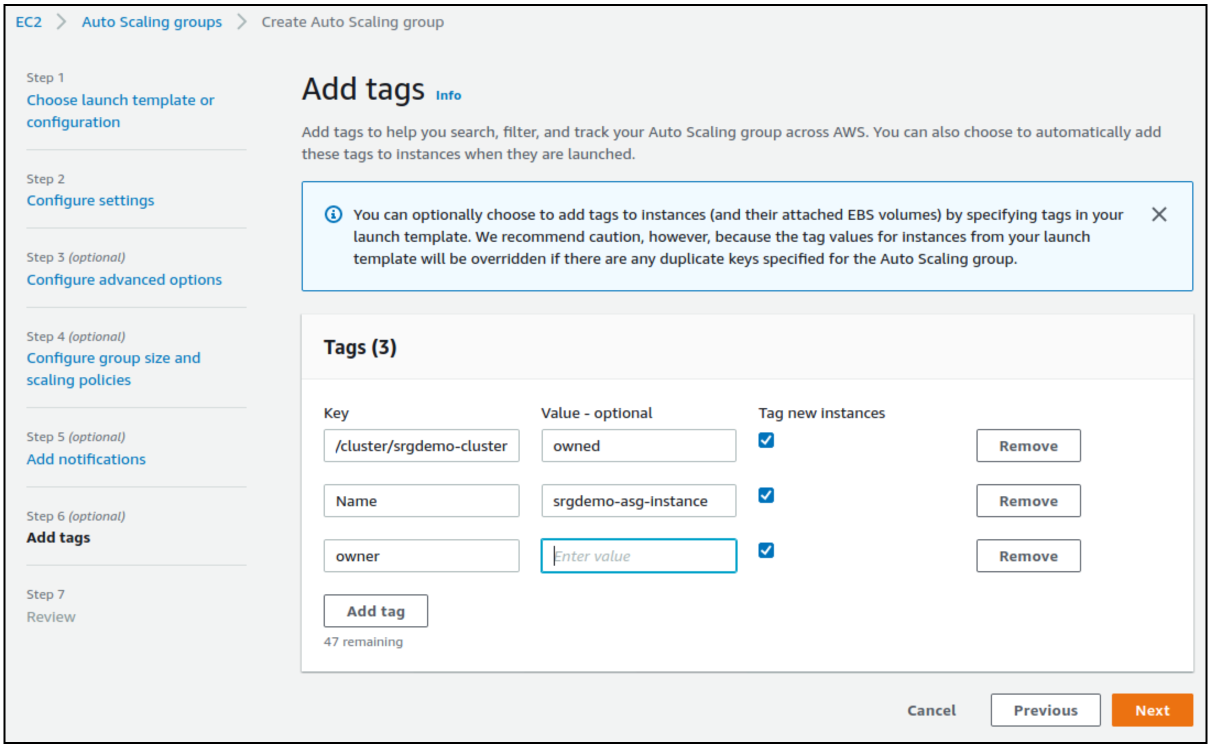
Step 6 *(optional)*
Add tags

Step 7
Review

Add notifications Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

12. On the **Add Tags** page, add tags as follows:
    a. Add the mandatory tag key: `kubernetes.io/cluster/<your cluster name>` with value: owned (replace `<your cluster name>` with your actual cluster name).
    b. Optionally, add these tags as desired:
        i. Key: `Name` with value: derived from the auto-scaling group name.
        ii. Key: `owner` with value: corresponding to you.
    c. For all new tags you add, select the **Tag new instances** checkbox so new instances are automatically tagged.
13. Click **Next**.



Add tags

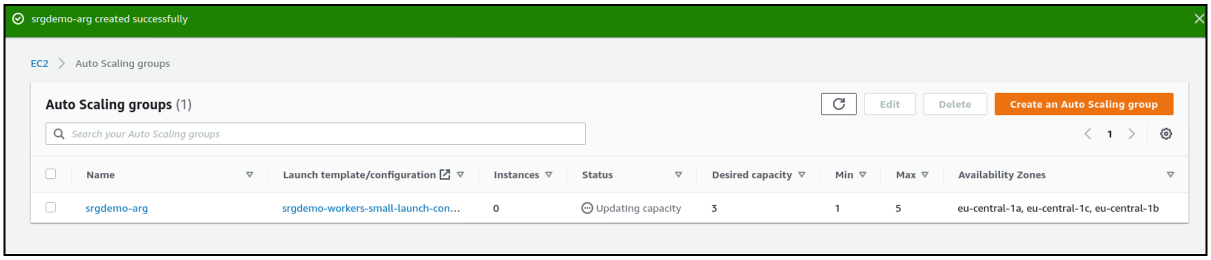
Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

ⓘ You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

Tags (3)

Key	Value - optional	Tag new instances
/cluster/srgdemo-cluster	owned	☑
Name	srgdemo-asg-instance	☑
owner	Enter value	☑

14. On the **Review** page, verify your settings, and then click **Create Auto Scaling Group.**

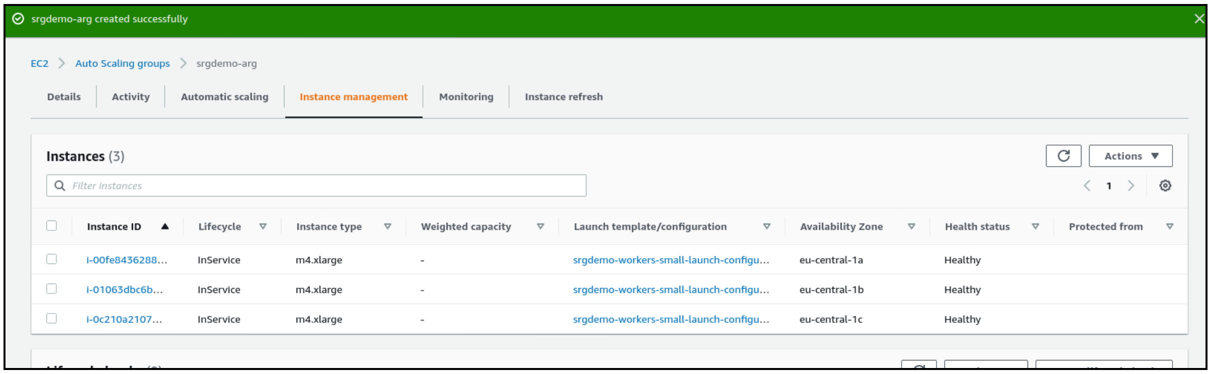

EC2 > Auto Scaling groups
Edit
Delete
Create an Auto Scaling group
Search your Auto Scaling groups
1

srgdemo-arg created successfully

Auto Scaling groups (1)

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
srgdemo-arg	srgdemo-workers-small-launch-con…	0	Updating capacity	3	1	5	eu-central-1a, eu-central-1c, eu-central-1b

15. For networking configurations, we will need to know the machine instance IDs in the new Auto Scaling group. On the **Auto Scaling Groups** management page, select your new group.



srgdemo-arg created successfully

Instances (3)

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/configuration	Availability Zone	Health status	Protected from
i-00fe8436288…	InService	m4.xlarge	-	srgdemo-workers-small-launch-configu…	eu-central-1a	Healthy	
i-01063dbc6b…	InService	m4.xlarge	-	srgdemo-workers-small-launch-configu…	eu-central-1b	Healthy	
i-0c210a2107…	InService	m4.xlarge	-	srgdemo-workers-small-launch-configu…	eu-central-1c	Healthy	

16. From the **Instances** tab, record all instance IDs in the AWS worksheet.

To create the Auto Scaling group using the CLI:

1. Run the following command:
   ```
   # aws autoscaling create-auto-scaling-group \
   --auto-scaling-group-name <Autoscaling group name> \
   --launch-configuration-name <Launch Configuration name> \
   --min-size <min size> \
   --desired-capacity <desired size> \
   --max-size <max size> \
   --tags "Key=kubernetes.io/cluster/<cluster name>,Value=owned"
   "Key=Name,Value=<auto scaling group name>" \
   --vpc-zone-identifier "<subnet Ids>"
   ```

Parameters:

`<Autoscaling group name>`: Choose a name which helps with easier identification; in this guide we will use srgdemo-autoscaling-group. Record the value in the AWS worksheet.

`<Launch Configuration name>`: Name of the launch configuration created above.

`<min size>`: The minimum size of the group ((for a production cluster, this should not be less than 2).

`<desired size>`: The number of Amazon EC2 instances that the Auto Scaling group attempts to maintain. This number must be greater than or equal to the minimum size of the group and less than or equal to the maximum size of the group. If you do not specify a desired capacity, the default is the minimum size of the group.

`<max size>`: The maximum size of the group.

`<Cluster Name>`: Use the cluster name from the AWS worksheet.

`<subnet Ids>`: A comma-separated list of private subnet IDs for your virtual private cloud (VPC); use values from from the AWS worksheet.

Example:

```
# aws autoscaling create-auto-scaling-group \
--auto-scaling-group-name srgdemo-autoscaling-group \
--launch-configuration-name srgdemo-workers-launch-config \
--min-size 1 \
--desired-capacity 3 \
--max-size 3 \
--tags "Key=kubernetes.io/cluster/srgdemo-cluster,Value=owned"
"Key=Name,Value=srgdemo-autoscaling-group" \
--vpc-zone-identifier "subnet-0fb2ebb5882c061f0,subnet-
0f0cac4ec6837abed,subnet-0abd7cd806e04c7be"
```

It can take approximately 5 minutes for nodes to be created and join the cluster.

Retrieve the instance IDs from auto-scaling group

For networking configurations, we will need to know the machine instances in the new Auto Scaling group.

To retrieve the instance IDs:

1. Run the command:
   ```
   # aws autoscaling describe-auto-scaling-instances\
   | jq -r '.AutoScalingInstances[] | select(.AutoScalingGroupName=="<Your
   new Auto Scaling group name>").InstanceId'
   ```
2. Record the returned IDs in the AWS worksheet.

Example command and output:

```
# aws autoscaling describe-auto-scaling-instances \
| jq -r '.AutoScalingInstances[] | select(.AutoScalingGroupName="srgdemo-
autoscaling-group").InstanceId'
```

```
i-05662f9ef84c182ca
```

```
i-07cfcd6716e9890b5
```

```
i-08d819b5ccabe83cb
```

After approximately 5 minutes, the nodes will be created and be joined to the cluster. You can then list all the worker nodes.

To list the worker nodes:

1. On the bastion host, run the following command:
   ```
   # kubectl get nodes
   ```

At this point, you should see all nodes listed in the Ready state with the expected Kubernetes version.

Next Step: Labeling Worker Nodes

Labeling Cloud (AWS) Worker Nodes

Labeling is a means for identifying application processing and qualifying the application as a candidate to run on a specific node. For example, labeling a node with the label `kafka=yes` specifies that a Kafka instance will run on that node.

- "Transformation Hub Labels" on the next page
- "Other Product Labels" on the next page

- "Labeling Worker Nodes" below
- For more information about labeling, see "Understanding Labels and Pods" on page 584

Transformation Hub Labels

Transformation Hub nodes on AWS require the following labels:

Label	Purpose
`kafka=yes`	Run Kafka
`zk=yes`	Run ZooKeeper (Kafka management tool)
`th-processing=yes`	Process Transformation Hub data
`th-platform=yes`	Run Transformation Hub

Other Product Labels

Other products require these labels:

Product	Required Labels
ArcSight ESM Command Center	`fusion=yes`
ArcSight Layered Analytics	`fusion=yes`
ArcSight Recon	`fusion=yes`
Fusion	`fusion=yes`

Intelligence	`fusion=yes`
	`interset=yes`
	`interset-namenode=yes`
	`interset-datanode=yes`

Labeling Worker Nodes

To label AWS worker nodes:

> You can skip this step if you have added all the required nodes to your launch configuration from which you have deployed your nodes.

1. Connect to the bastion.
2. Retrieve the list of nodes by running the command:

   ```
   # kubectl get nodes -o name | cut -d '/' -f 2
   ```

3. Run the following command once for each node:

   ```
   # kubectl label\
   --overwrite=true node <node name> zk=yes kafka=yes th-platform=yes th-processing=yes
   ```

   For example:

   ```
   # kubectl label \
   --overwrite=true node ip-10-0-10-83.eu-central-1.compute.internal zk=yes
   kafka=yes th-platform=yes th-processing=yes
   node/ip-10-0-10-83.eu-central-1.compute.internal labeled
   ```

4. Verify the labels by running the command:

   ```
   # kubectl get nodes --show-labels
   ```

Next Step: Upload Product Images to the ECR

Uploading Product Images to the ECR

The Amazon Elastic Container Registry (ECR) is a fully-managed Docker container registry. CDF and Kubernetes will search for product images to download from the ECR and instantiate them.

ECR is accessible from the internet and protected by username/password credentials. You can perform tasks in this chapter from a local computer or even from the bastion, as long as the AWS CLI has been configured.

Uploading images requires the script `upload_images_to_ECR` installed and located in `aws-byok-installer/scripts/` directory. This script parses the `manifest.json` description file and prepares the respective repositories in the ECR. Then the CDF script `uploadimages.sh` is called, which passes the correct parameters.

Uploading Image Requirements

In order to be able to upload images to the ECR, the following requirements must be met:

- You must be able to execute a bash script.
- The system used must have the following basic Linux/Unix utilities installed:
  - `cat`
  - `find`

  - awk
  - jq
  - pwd
  - unzip
  - tar
- You must have aws cli configured on your system.
- You have fulfilled all requirements for the CDF uploadimages.sh script.

To upload the product images to the ECR:

1. Verify that you have downloaded the product image files:
   a. aws-byok-installer.zip
   b. cdf-byok-images-<version>.tar
   c. transformationhub-<version>.tar
2. Run the following command:

```
# <path to upload script>/upload_images_to_ECR \
-d <images' folder> \
-F <product package> \
-o <organization> \
-y \
[-c <parallel uploads count>]\
[-uip <uploadimages.sh path>]
```

Parameters:

<path to upload script>: It is possible to execute the upload script from any folder. The recommendation is to have the current folder set to the one with downloaded images; then the path would resemble aws-byok-installer/scripts/

<images' folder>: Folder where all images in their subfolders are located; usually it is the folder where you have unpacked downloaded packages. Can be specified multiple times for situations where images are located in various folders.

<product package>: Path to the package file, for example, ./transformationhub-<version>.tar. Can be specified multiple times.

<organization>: Specifies the organization name (namespace) where the suite images are placed in the ECR. Record the chosen organization name in the AWS worksheet. There might be multiple repositories in the ECR which might be shared or overlap. Please pay special attention to specify the correct organization name.

> The organization name must be valid ASCII, and can be from 2 to 255 characters. It can only contain lowercase letters, numbers, dashes (-) and underscores (_).

`<parallel upload counts>`: Maximum allowed parallel uploads; this is limited based on the CPU cores. The parameter is optional; if not specified, defaults to 8.

`<uploadimages.sh path>`: Path to the original CDF `uploadimages.sh` script. Parameter is optional. When not specified, the `upload_images_to_ECR` script will try to locate it in the images' folder or in the unpacked cdf-deployer package which is part of of the `aws-byok-installer` package. Note that normally this package should not be unpacked.

> You must specify at least one image location, either in form of a folder (`-d` option) or as a file path (`-F` option, recommended). If the `-d` option is used, the image package must be unpacked before running the script.

Example:

```
./aws-byok-installer-2020.08.00153-20.11.0.599/scripts/upload_images_to_ECR \
-F ./transformationhub-3.4.0.1284-master.tar -o srgdemo \
-y \
-c 8
```

Please be patient and give the upload process time to complete. You can check the returned messages or check the log file in the directory where you are executing the upload script to determine successful upload. While the upload progresses, the repositories are created in the ECR, followed by image uploads to the repositories.

> Multiple suite images can be uploaded as a single command as long as each image package is prefaced with `-F`.

Next Step: Configuring Route 53 Routing

Configuring Route 53

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. This section will detail performing the creation of a publicly available CDF installation,and the CDF management portal, as well as suite reconfiguration. Your own business requirements might dictate a different secure configuration.

Next Step: Select a Public Hosted Zone and Create a Record Set

Selecting a Public Hosted Zone and Creating a Record Set

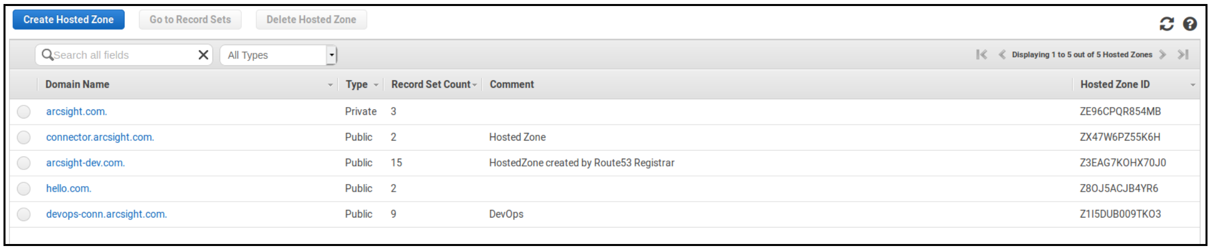
In Route 53, DNS records are organized in *hosted zones.* A hosted zone is analogous to a traditional DNS zone file; it represents a collection of records that can be managed together, belonging to a single parent domain name. All resource record sets within a hosted zone must have the hosted zone's domain name as a suffix.

In this section you will select a **public** hosted zone (which must be previously created by an AWS administrator) and create the record set.

> Do not use a private hosted zone.

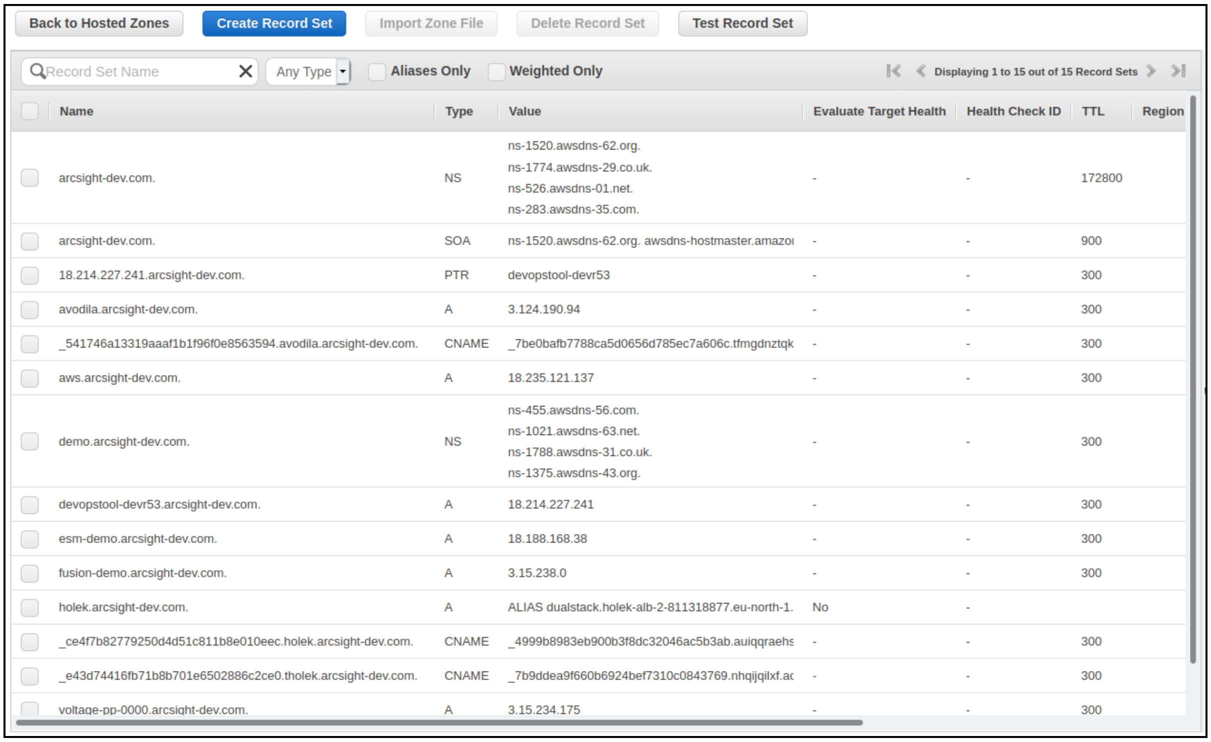
To select an existing public (not private) hosted zone and create the record set using the web UI:

1. Using the Find Services search tool, locate and browse to the Route 53 dashboard.
2. In the left navigation panel, select **Hosted Zones**. (Ignore any error message about insufficient permission.)
3. From the list of hosted zones, select a **public** zone. (For our example, we use `arcsight-dev.com`).



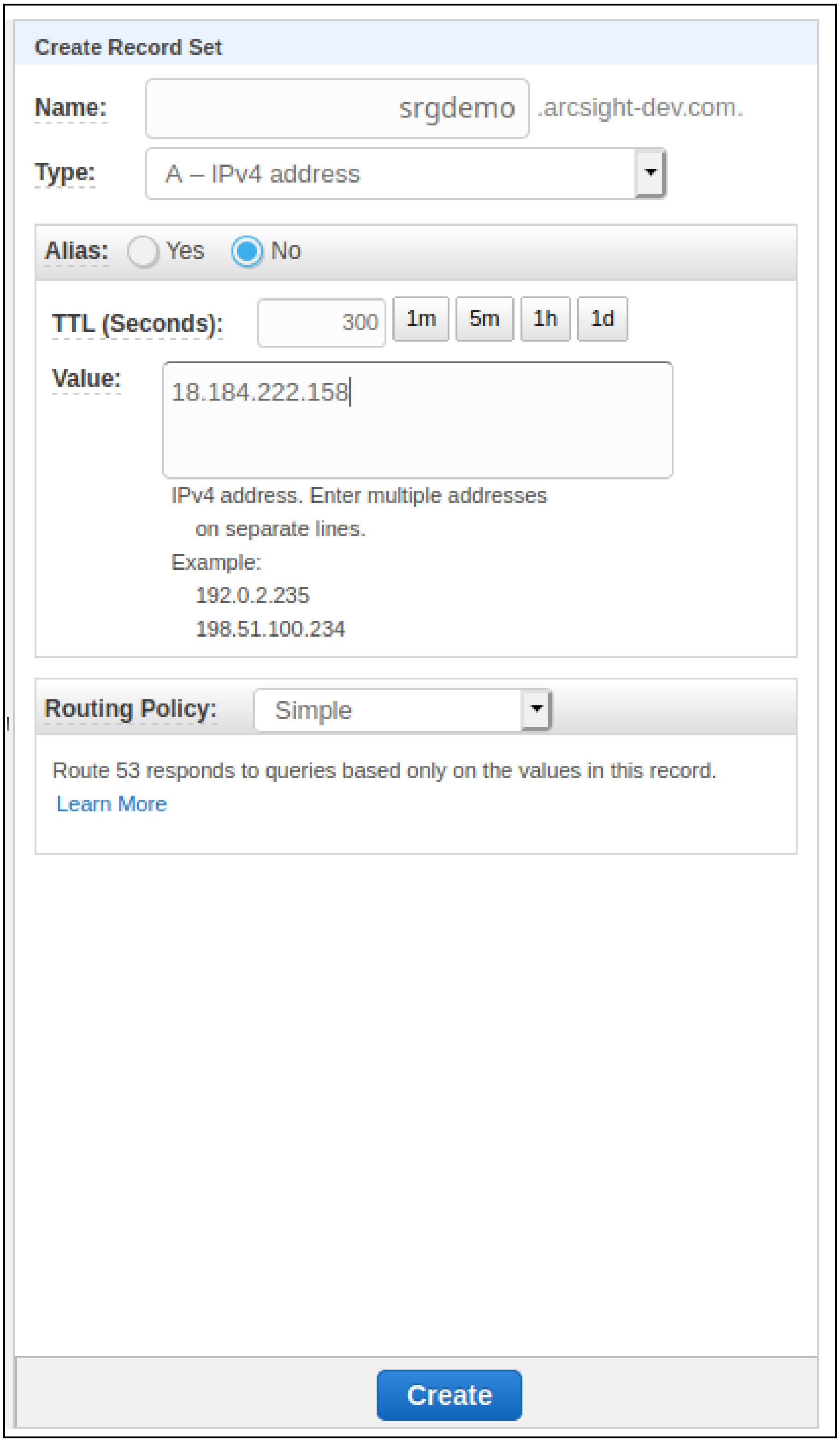
Domain Name	Type	Record Set Count	Comment	Hosted Zone ID
○ arcsight.com.	Private	3		ZE96CPQR854MB
○ connector.arcsight.com.	Public	2	Hosted Zone	ZX47W6PZ55K6H
○ arcsight-dev.com.	Public	15	HostedZone created by Route53 Registrar	Z3EAG7KOHX70J0
○ hello.com.	Public	2		Z8OJ5ACJB4YR6
○ devops-conn.arcsight.com.	Public	9	DevOps	Z1I5DUB009TKO3

4. Click on the public hosted zone domain name to list the record sets in the public zone.



Name	Type	Value	Evaluate Target Health	Health Check ID	TTL	Region
arcsight-dev.com.	NS	ns-1520.awsdns-62.org. ns-1774.awsdns-29.co.uk. ns-526.awsdns-01.net. ns-283.awsdns-35.com.	-	-	172800	
arcsight-dev.com.	SOA	ns-1520.awsdns-62.org. awsdns-hostmaster.amazoı	-	-	900	
18.214.227.241.arcsight-dev.com.	PTR	devopstool-devr53	-	-	300	
avodila.arcsight-dev.com.	A	3.124.190.94	-	-	300	
_541746a13319aaaf1b1f96f0e8563594.avodila.arcsight-dev.com.	CNAME	_7be0bafb7788ca5d0656d785ec7a606c.tfmgdnztqk	-	-	300	
aws.arcsight-dev.com.	A	18.235.121.137	-	-	300	
demo.arcsight-dev.com.	NS	ns-455.awsdns-56.com. ns-1021.awsdns-63.net. ns-1788.awsdns-31.co.uk. ns-1375.awsdns-43.org.	-	-	300	
devopstool-devr53.arcsight-dev.com.	A	18.214.227.241	-	-	300	
esm-demo.arcsight-dev.com.	A	18.188.168.38	-	-	300	
fusion-demo.arcsight-dev.com.	A	3.15.238.0	-	-	300	
holek.arcsight-dev.com.	A	ALIAS dualstack.holek-alb-2-811318877.eu-north-1.	No	-		
_ce4f7b82779250d4d51c811b8e010eec.holek.arcsight-dev.com.	CNAME	_4999b8983eb900b3f8dc32046ac5b3ab.auiqqraehs	-	-	300	
_e43d74416fb71b8b701e6502886c2ce0.tholek.arcsight-dev.com.	CNAME	_7b9ddea9f660b6924bef7310c0843769.nhqijqilxf.ac	-	-	300	
voltage-pp-0000.arcsight-dev.com.	A	3.15.234.175	-	-	300	

5. Click **Create Record Set** and enter or verify values for the following parameters:
   - **Name:** Choose a name for the subdomain. The record set FQDN will then be composed from this name and the public hosted zone domain name. In our example we will use `srgdemo`. Our installation will then be available at the URL: `srgdemo.arcsight-dev.com`.
     Record the record set FQDN in the AWS worksheet.
   - **Type:** *A - IPv4 address.*
   - **Alias:** *No*
   - **TTL:** Leave default.
   - **Value:** Enter your bastion's IP address.
   - **Routing Policy:** *Simple*



Create Record Set

Name: srgdemo .arcsight-dev.com.

Type: A – IPv4 address

Alias: ○ Yes ● No

TTL (Seconds): 300

Value: 18.184.222.158

IPv4 address. Enter multiple addresses on separate lines.

Example:

192.0.2.235

198.51.100.234

Routing Policy: Simple

Route 53 responds to queries based only on the values in this record.

Learn More

6. Click **Create.** The new record set is displayed on the list.

To select an existing public (not private) hosted zone and create a record set using the CLI:

1. Run the following command to select **public** hosted zones:
   ```
   # aws route53 list-hosted-zones \
   | jq -r '.HostedZones[] | select(.Config.PrivateZone==false) | "Id: " +
   .Id,"Name: " + .Name," " '
   ```

A list of hosted **public** zones is returned. For example:

```
Id:   /hostedzone/ZX47W6PZ55K6H
Name: connector.arcsight.com.
```

```
Id:   /hostedzone/Z3EAG7KOHX70J0
Name: arcsight-dev.com.
```

```
Id:   /hostedzone/Z8OJ5ACJB4YR6
Name: hello.com.
```

```
Id:   /hostedzone/Z1I5DUB009TKO3
Name: devops-conn.arcsight.com.
```

> The dots/period characters (.) at the end of each value are present intentionally. **Please do not remove them.**

2. Choose one of the **public** hosted zones. For example, we will use the **public** hosted zone (name shown includes a period):
   `arcsight-dev.com.`
   Record the chosen **public** hosted zone name and ID in the AWS worksheet under `Hosted zone name` and `Hosted zone Id` respectively.
3. Choose a subdomain in selected public hosted zone. For example, we will use `srgdemo`. Combining the subdomain and hosted zone name with a final period will give us the complete DNS name where our new cluster will be accessible.
   Example:
   `srgdemo.arcsight-dev.com.`
4. From the directory `aws-byok-installer-<version>/objectdefs/`, copy the supplied template `CreateRecordSetInHostedZone.json` to the working folder.
5. Open the template in a text editor and set values for the following placeholders:
   a. `<Record name>:` combine the name of the hosted zone, for example, `srgdemo` and Hosted zone name (for example, `arcsight-dev.com`) to create the DNS name and then append the dot character (.) `Example: srgdemo.arcsight-dev.com.`
   b. `<Record type>:` replace with a value of `A.`
   c. `<Record value>:` Use your bastion IP address.

> The placeholders in the template use syntax `<placeholder name>`, for example, `<Record name>`.

Example of modified JSON template (notice that the trailing period in the record name is mandatory):

```
{
   "Changes":[
      {
         "Action":"UPSERT",
         "ResourceRecordSet":{
            "Name":"srgdemo.arcsight-dev.com.",
            "Type":"A",
            "TTL":300,
            "ResourceRecords":[
               {
                  "Value":"3.120.237.11"
               }
            ]
         }
      }
   ]
}
```

6. Run the following command:

```
# aws route53 change-resource-record-sets \
--hosted-zone-id <Hosted zone Id> \
--change-batch file://CreateRecordSetInHostedZone.json
```

Parameters:

`<Hosted zone Id>`: Use the hosted zone Id retrieved above, for example: `/hostedzone/Z3EAG7KOHX70J0`

`--change-batch`: Replace the parameter here with your own modified instance of the JSON file `CreareRecordSetInHostedZone.json`.

7. The command returns a change request, for example:

```
{
   "ChangeInfo":{
      "Id":"/change/C04669622EJ7JNXG69KJO",
      "Status":"PENDING",
      "SubmittedAt":"2020-06-09T09:35:06.376000+00:00"
   }
}
```

Later the status, will change to `INSYNC`:

```
{
   "ChangeInfo":{
      "Id":"/change/C04669622EJ7JNXG69KJO",
      "Status":"INSYNC",
      "SubmittedAt":"2020-06-09T09:35:06.376000+00:00"
   }
}
```

Next Step: Creating and Validating the Route 53 Certificate

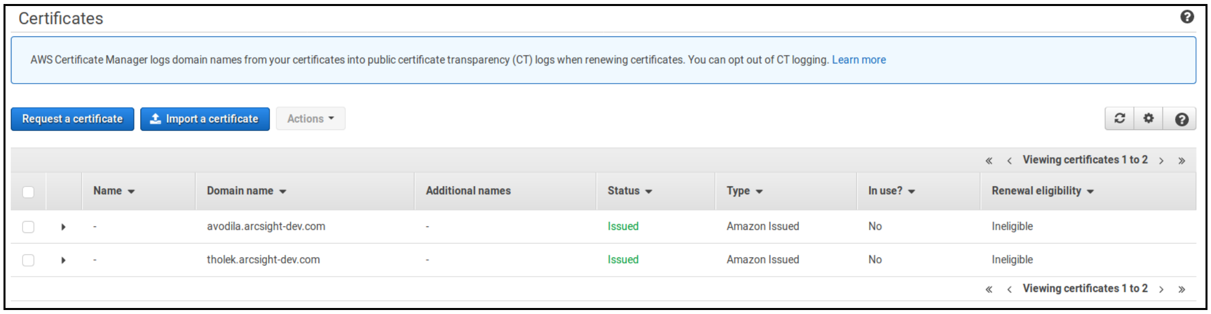
Creating and Validating the Certificate

A certificate is required for creating the Application Load Balancer (ALB). You can store your certificate in the Amazon Certificate Manager (ACM) or in the Identity and Access Management (IAM).

In our example, we will create and validate a new certificate in ACM, for the example domain registered previously, that is:
`srgdemo.arcsight-dev.com`.

To create and validate the certificate using the web UI:

1. Using the Find Services search tool, search for ACM, and locate and browse Certificate Manager.
2. On the **Certificates** page, click **Request a Certificate.**

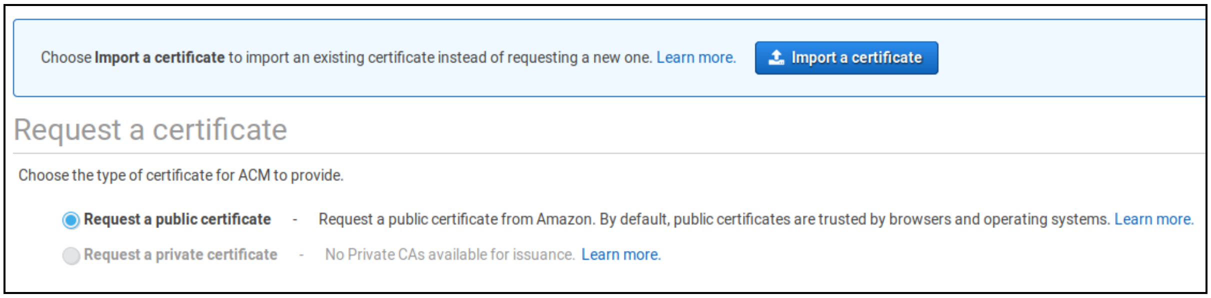


Certificates

AWS Certificate Manager logs domain names from your certificates into public certificate transparency (CT) logs when renewing certificates. You can opt out of CT logging. Learn more

	Name	Domain name	Additional names	Status	Type	In use?	Renewal eligibility
□	-	avodila.arcsight-dev.com	-	Issued	Amazon Issued	No	Ineligible
□	-	tholek.arcsight-dev.com	-	Issued	Amazon Issued	No	Ineligible

3. On the **Request a Certificate** page, click **Request a Certificate.**



Choose **Import a certificate** to import an existing certificate instead of requesting a new one. Learn more.

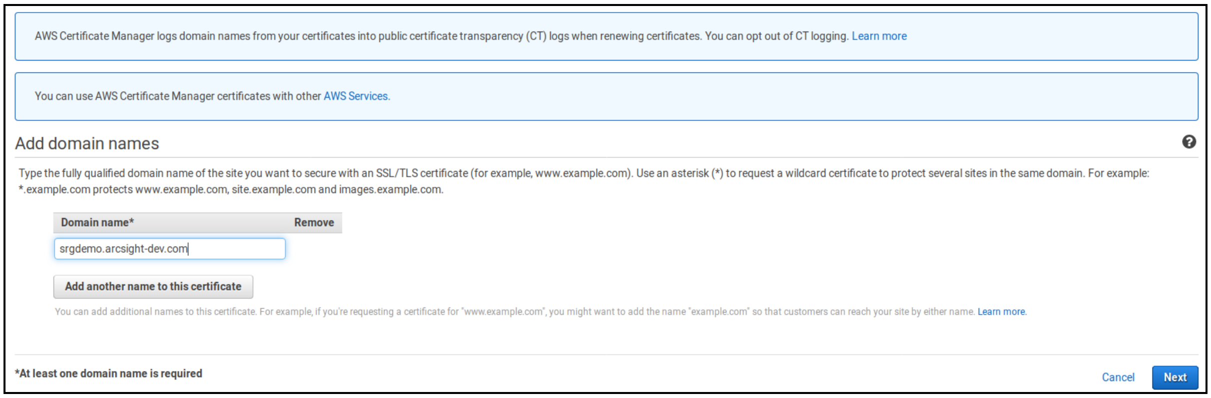
Request a certificate

Choose the type of certificate for ACM to provide.

● **Request a public certificate** - Request a public certificate from Amazon. By default, public certificates are trusted by browsers and operating systems. Learn more.

○ **Request a private certificate** - No Private CAs available for issuance. Learn more.

4. On the **Add domain names** page, in **Domain name**, enter the FQDN.



AWS Certificate Manager logs domain names from your certificates into public certificate transparency (CT) logs when renewing certificates. You can opt out of CT logging. Learn more

You can use AWS Certificate Manager certificates with other AWS Services.

Add domain names

Type the fully qualified domain name of the site you want to secure with an SSL/TLS certificate (for example, www.example.com). Use an asterisk (*) to request a wildcard certificate to protect several sites in the same domain. For example: *.example.com protects www.example.com, site.example.com and images.example.com.

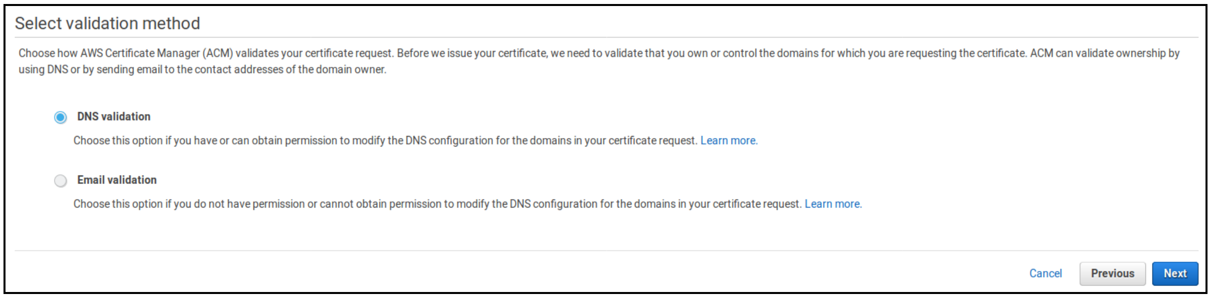
Domain name*	Remove
srgdemo.arcsight-dev.com	

Add another name to this certificate

You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name. Learn more.

*At least one domain name is required

5. Click **Next**.
6. On the **Select validation method** page, leave the validation method as *DNS validation.*



Select validation method

Choose how AWS Certificate Manager (ACM) validates your certificate request. Before we issue your certificate, we need to validate that you own or control the domains for which you are requesting the certificate. ACM can validate ownership by using DNS or by sending email to the contact addresses of the domain owner.

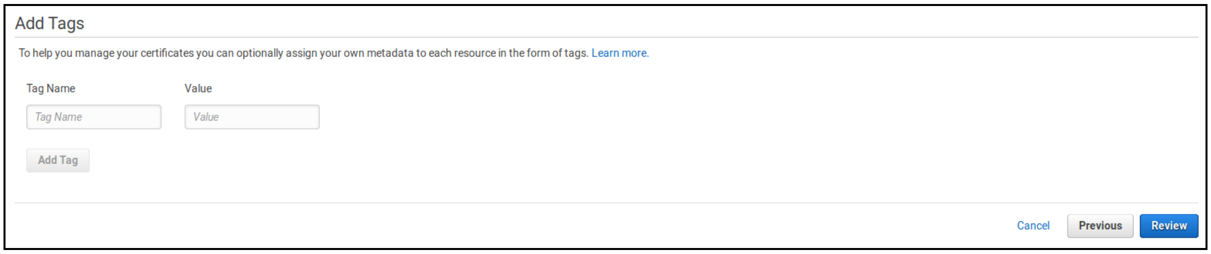
● DNS validation

Choose this option if you have or can obtain permission to modify the DNS configuration for the domains in your certificate request. Learn more.

○ Email validation

Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request. Learn more.

7. Click **Next**.
8. On the **Add tags** page, enter any tags as desired (tags are optional).

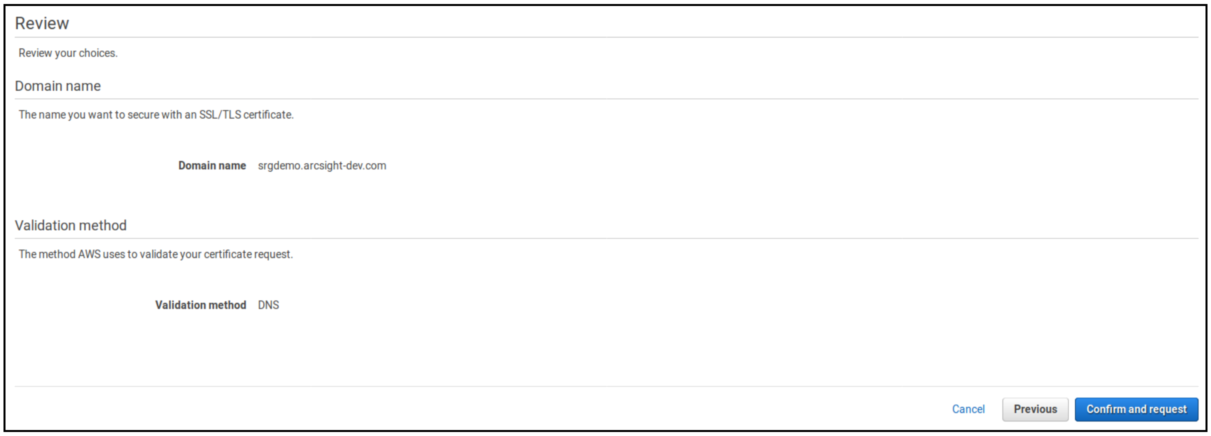


Add Tags

To help you manage your certificates you can optionally assign your own metadata to each resource in the form of tags. Learn more.

Tag Name	Value
Tag Name	Value

9. Click **Review**.
10. On the **Review** page, review your assigned settings. When ready, click **Confirm and Request.**



Review

Review your choices.

Domain name

The name you want to secure with an SSL/TLS certificate.

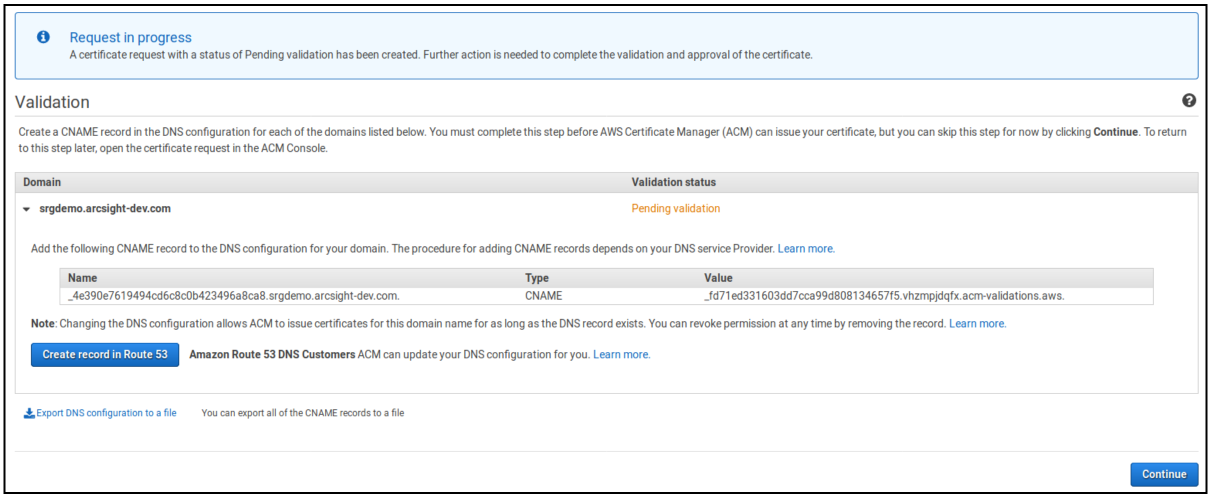
Domain name srgdemo.arcsight-dev.com

Validation method

The method AWS uses to validate your certificate request.

Validation method DNS

11. On the **Validation** page, expand the row with your new certificate. (In the screenshot, record the validation status with the value Pending validation.)



Request in progress

A certificate request with a status of Pending validation has been created. Further action is needed to complete the validation and approval of the certificate.

Validation

Create a CNAME record in the DNS configuration for each of the domains listed below. You must complete this step before AWS Certificate Manager (ACM) can issue your certificate, but you can skip this step for now by clicking **Continue**. To return to this step later, open the certificate request in the ACM Console.

Domain	Validation status
srgdemo.arcsight-dev.com	Pending validation

Add the following CNAME record to the DNS configuration for your domain. The procedure for adding CNAME records depends on your DNS service Provider. Learn more.

Name	Type	Value
_4e390e7619494cd6c8c0b423496a8ca8.srgdemo.arcsight-dev.com.	CNAME	_fd71ed331603dd7cca99d808134657f5.vhzmpjdqfx.acm-validations.aws.

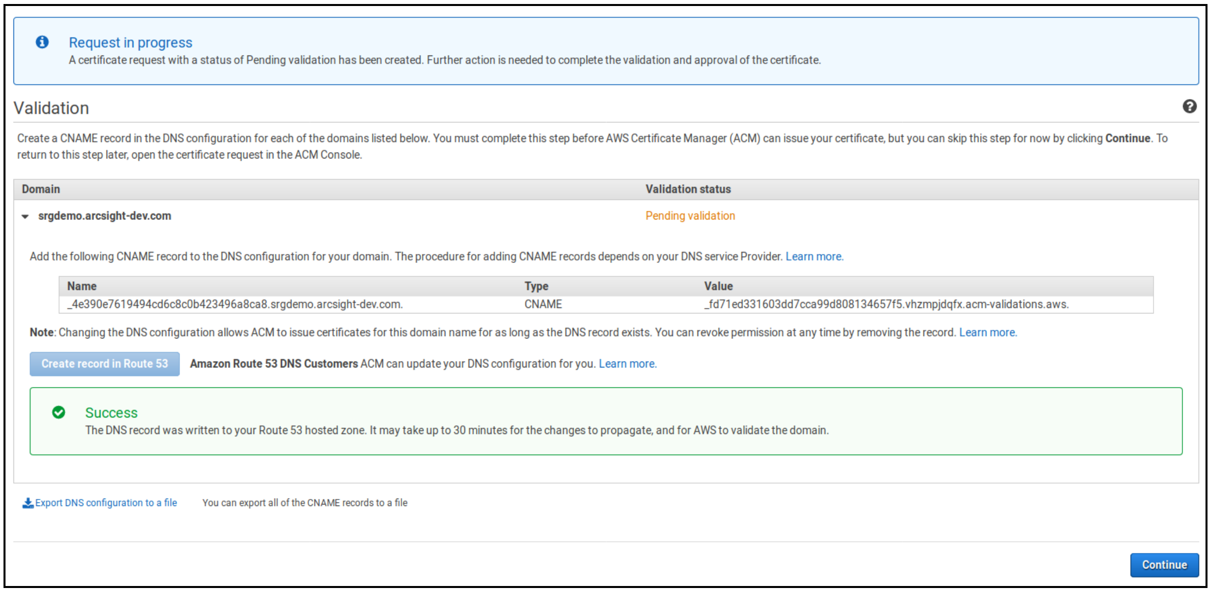
Note: Changing the DNS configuration allows ACM to issue certificates for this domain name for as long as the DNS record exists. You can revoke permission at any time by removing the record. Learn more.

Amazon Route 53 DNS Customers ACM can update your DNS configuration for you. Learn more.

You can export all of the CNAME records to a file

12. Perform the validation by clicking **Create record in Route 53.** Then, confirm the record set value in the pop-up window.

13. When the creation succeeds, click **Continue**. You will be returned to the certificates list. Your certificate should have a status of *Issued*; click **Refresh** to check the most recent status.



Request in progress

A certificate request with a status of Pending validation has been created. Further action is needed to complete the validation and approval of the certificate.

Validation

Create a CNAME record in the DNS configuration for each of the domains listed below. You must complete this step before AWS Certificate Manager (ACM) can issue your certificate, but you can skip this step for now by clicking **Continue**. To return to this step later, open the certificate request in the ACM Console.

Domain	Validation status
srgdemo.arcsight-dev.com	Pending validation

Add the following CNAME record to the DNS configuration for your domain. The procedure for adding CNAME records depends on your DNS service Provider. Learn more.

Name	Type	Value
_4e390e7619494cd6c8c0b423496a8ca8.srgdemo.arcsight-dev.com.	CNAME	_fd71ed331603dd7cca99d808134657f5.vhzmpjdqfx.acm-validations.aws.

Note: Changing the DNS configuration allows ACM to issue certificates for this domain name for as long as the DNS record exists. You can revoke permission at any time by removing the record. Learn more.

Amazon Route 53 DNS Customers ACM can update your DNS configuration for you. Learn more.

Success

The DNS record was written to your Route 53 hosted zone. It may take up to 30 minutes for the changes to propagate, and for AWS to validate the domain.

You can export all of the CNAME records to a file

To create and validate the certificate using the CLI:

1. Run the following command:

```
# aws acm request-certificate \
--domain-name "<Name in hosted zone>.<Hosted zone name>" \
--validation-method DNS \
--tags Key=owner,Value="<owner name>"
```

Parameters:

`<Name in hosted zone>`: the first part of the FQDN for which you created the Route 53 record set earlier (for example, srgdemo).

`<Hosted zone name>`: name of hosted zone we chose for creating the new record set (for example, arcsight-dev.com)

`<owner name>`: put the text of your choice; for our example, srgdemo.

> During the certificate request, the FQDN should **NOT** include the trailing period (.)

2. As a result of the certification request, the certificate ARN will be returned, as follows:

```
{ "CertificateArn": "<Certificate ARN>" }
```

3. Record this certificate ARN in the AWS worksheet.

Example input and output:

```
# aws acm request-certificate \
--domain-name "srgdemo.arcsight-dev.com" \
--validation-method DNS --tags Key=owner,Value="srgdemo"
```

```
{
"CertificateArn": "arn:aws:acm:eu-central-
1:115370848038:certificate/691ec232-98ff-45ed-8e69-1d15c0447538"
}
```

Certificate Validation

The created certificate is not valid until the validation process is completed. When we requested the certificate, we selected the DNS validation method. The specific record set must be created in the same hosted zone as the FQDN resides for which the certificate was issued.

For example, we created the certificate for FQDN `srgdemo.arcsight-dev.com`, so that validation record set needs to be created in the hosted zone named `arcsight-dev.com`.

To validate the certificate using the CLI:

1. Generate a comprehensive certificate description by running the following command:
   ```
   # aws acm describe-certificate --certificate-arn <Certificate ARN>
   ```

Example input and output with domain validation options:

```
# aws acm describe-certificate \
--certificate-arn arn:aws:acm:eu-central-1:115370848038:certificate/691ec232-
98ff-45ed-8e69-1d15c0447538
```

```
{
   "Certificate":{
      "CertificateArn":"arn:aws:acm:eu-central-
1:115370848038:certificate/691ec232-98ff-45ed-8e69-1d15c0447538",
      "DomainName":"srgdemo.arcsight-dev.com",
      "SubjectAlternativeNames":[
         "srgdemo.arcsight-dev.com"
      ],
      "DomainValidationOptions":[
         {
            "DomainName":"srgdemo.arcsight-dev.com",
            "ValidationDomain":"srgdemo.arcsight-dev.com",
            "ValidationStatus":"PENDING_VALIDATION",
            "ResourceRecord":{
               "Name":"_4e390e7619494cd6c8c0b423496a8ca8.srgdemo.arcsight-
dev.com.",
               "Type":"CNAME",
               "Value":"_fd71ed331603dd7cca99d808134657f5.vhzmpjdqfx.acm-
validations.aws."
            },
            "ValidationMethod":"DNS"
         }
      ],
      "Subject":"CN=srgdemo.arcsight-dev.com",
      "Issuer":"Amazon",
```

```
      "CreatedAt":"2020-06-10T12:50:59+02:00",
      "Status":"PENDING_VALIDATION",
      "KeyAlgorithm":"RSA-2048",
      "SignatureAlgorithm":"SHA256WITHRSA",
      "InUseBy":[

      ],
      "Type":"AMAZON_ISSUED",
      "KeyUsages":[

      ],
      "ExtendedKeyUsages":[

      ],
      "RenewalEligibility":"INELIGIBLE",
      "Options":{
         "CertificateTransparencyLoggingPreference":"ENABLED"
      }
   }
}
```

Currently, the certificate has status `PENDING_VALIDATION`. The secret information is located in the JSON output under: (JSON path) `Certificate -> DomainValidationOptions -> 0 -> ResourceRecord`

2. Copy the supplied template `CreateRecordSetInHostedZone.json` to a new file and replace the Name, Type, and Value placeholders with their values from the example above. Save the edited file with the new name as `ValidationRecordSet.json`.
3. Run the following command:
   ```
   # aws route53 change-resource-record-sets \
   --hosted-zone-id <Hosted zone Id> \
   --change-batch file://ValidationRecordSet.json
   ```

Parameters:

`<Hosted zone Id>`: Use the hosted zone ID recorded in the AWS worksheet. For example, `/hostedzone/Z3EAG7KOHX70J0`

`--change batch`: Replace the parameter here with your own modified instance of the JSON template.

4. After the process completes, the certificate will change its Status to `ISSUED` and in `DomainValidationOptions` you will see that the `ValidationStatus` has value `SUCCESS`.
5. Repeat the command to check the status change:
   ```
   # aws acm describe-certificate \
   --certificate-arn <Certificate ARN> \
   ```

```
| jq -r '"Validation Status: " + .Certificate.DomainValidationOptions
[0].ValidationStatus,"Certificate Status: " + .Certificate.Status'
```

Next Step: Bootstrapping CDF

Bootstrapping CDF

Bootstrapping CDF is a method of installing a few basic pods onto the Kubernetes cluster created previously (when you configured EKS and worker nodes).

During this process, the CDF bootstrap script does the following:

- Downloads Docker images from the ECR (Elastic Container Registry)
- Instantiates pods for various checks like the EFS space and structure created on it
- Creates `nginx` pods for use as a load balancer, and for allowing connections to the web installation process

After this CDF bootstrap process completes, you will need to configure some required networking settings, and then continue installation using the CDF web installation interface (on port 3000).

Preparing the CDF Deployer

The EKS and worker nodes you have configured are completely isolated from access from the internet, but they can access it if needed. As a result, the process of bootstrapping CDF must be performed from the bastion.

You have already copied the package `aws-byok-installer-<version>.zip` to the bastion and unpacked it during configuration of EFS. As a part of this package, the `cdf-deployer-<version>.zip` is supplied in the `aws-byok-installer-<version>/installer` directory.

To prepare the CDF deployer:

1. Unpack the `cdf-deployer-<version>.zip` archive by running the following command:
   ```
   # unzip ./aws-byok-installer-<version>/installer/cdf-deployer-<current
   version>.zip
   ```

This will create the directory `/aws-byok-installer-<version>/installer/cdf-deployer-<version>`.

Retrieve the ECR credentials

CDF needs the credentials to the ECR in order to be able to download images.

To retrieve the ECR credentials:

1. On the bastion, run the command:
   ```
   # ./aws-byok-installer-<version>/scripts/upload_images_to_ECR --get-ecr-credentials
   ```
2. The file `ecr_credentials` is created in the directory where the script was run, containing username, password and ECR URL.
3. Run the following command:
   ```
   # source ecr_credentials
   ```

⚠ The password retrieved here is only valid for 12 hours after creation.

To bootstrap CDF:

1. Change the working folder to `cdf-deployer-<version>` and run the following command:
   ```
   # ./install \
   --registry-url $ECR_URL \
   --registry-username $ECR_USER_NAME \
   --registry-password $ECR_USER_PASSWORD \
   -P <suite admin password> \
   --registry-orgname <orgname> \
   --nfs-server <Filesystem FQDN>
   --nfs-folder <CDF ITOM volume>
   --cloud-provider aws --external-access-host <RecordSet name>
   ```

Parameters:

Variables **`$ECR_URL, $ECR_USER_NAME`** and **`$ECR_USER_PASSWORD`** come from the `ecr_credentials` file which you sourced previously.

`<suite admin password>`: Choose a password 8 to 20 characters, with numbers, lowercase chars, uppercase chars and special characters. Exclude whitespace characters, such as space, newline, and so on.

`<orgname>` : Use the same value as for upload images; check the AWS worksheet for this value.

`<Filesystem FQDN>`: use the value from the AWS worksheet.

`<CDF ITOM volume>`: The directory on NFS/EFS where CDF starts installation into. The path combines from the parent directory as specified in Configure EFS for ArcSight Suite and predefined subfolder name. For example, `/srgdemo/itom-vol`.

`<RecordSet name>` : FQDN used for connecting to the CDF installation and management portal. Use the value the AWS worksheet.
Be sure to remove the trailing period from the FQDN.

Example:

```
# ./install --registry-url $ECR_URL \
--registry-username $ECR_USER_NAME \
--registry-password $ECR_USER_PASSWORD \
-P "Password@123" \
--registry-orgname srgdemo \
--nfs-server fs-ebe456b3.efs.eu-central-1.amazonaws.com \
--nfs-folder /srgdemo/itom-vol \
--cloud-provider aws \
--external-access-host srgdemo.arcsight-dev.com
```

After the CDF bootstrap completes, you will be prompted to log in at the following URL: `https://<external access host>:3000`

You will not be able to log in yet, as there are some network infrastructure resources still to prepare. You will perform the configuration process next.

Next Step: Configuring the Application Load Balancer

Configuring the Application Load Balancer (ALB)

A load balancer serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple availability zones. Balancing the load increases the availability of your application. AWS supports several types of load balancers: application, network, and (obsoleted) classic. In this section, you will configure an application load balancer (ALB)

The ALB needs to be configured with locations to balance requests; this is realized by target groups. During the installation process you will create target groups for various ports : 3000 (CDF installation), 5443 (CDF management portal), and 443 (ArcSight Suite configuration).

> Immediately after the core CDF bootstrap, only the installation portal is available on port 3000. The remaining two confgured will be created after the CDF UI installation process is completed.

Retrieving the CDF Ingress Service Node Port

To retrieve the CDF ingress service node port for 3000:

1. Run the following command on the bastion:
   `# kubectl get svc -n core | grep itom-cdf-ingress-frontend-svc`

Example output:

```
itom-cdf-ingress-frontend-svc LoadBalancer 172.20.150.202 <none>
3000:30058/TCP 18h
```

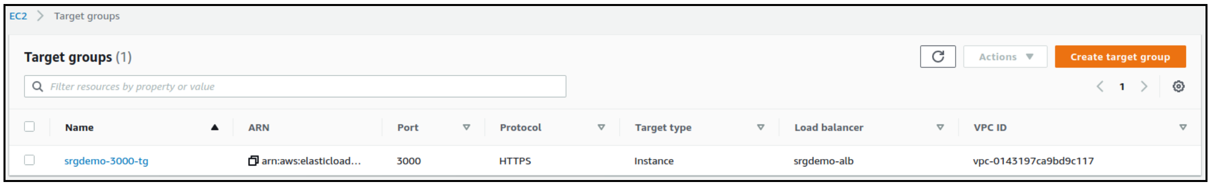
2. Record the highlighted port number in your AWS worksheet as `Node port for 3000.` In the example shown, the port number is 30058.

Next Step: Creating the Target Group for Port 3000

Creating the Target Group for Port 3000

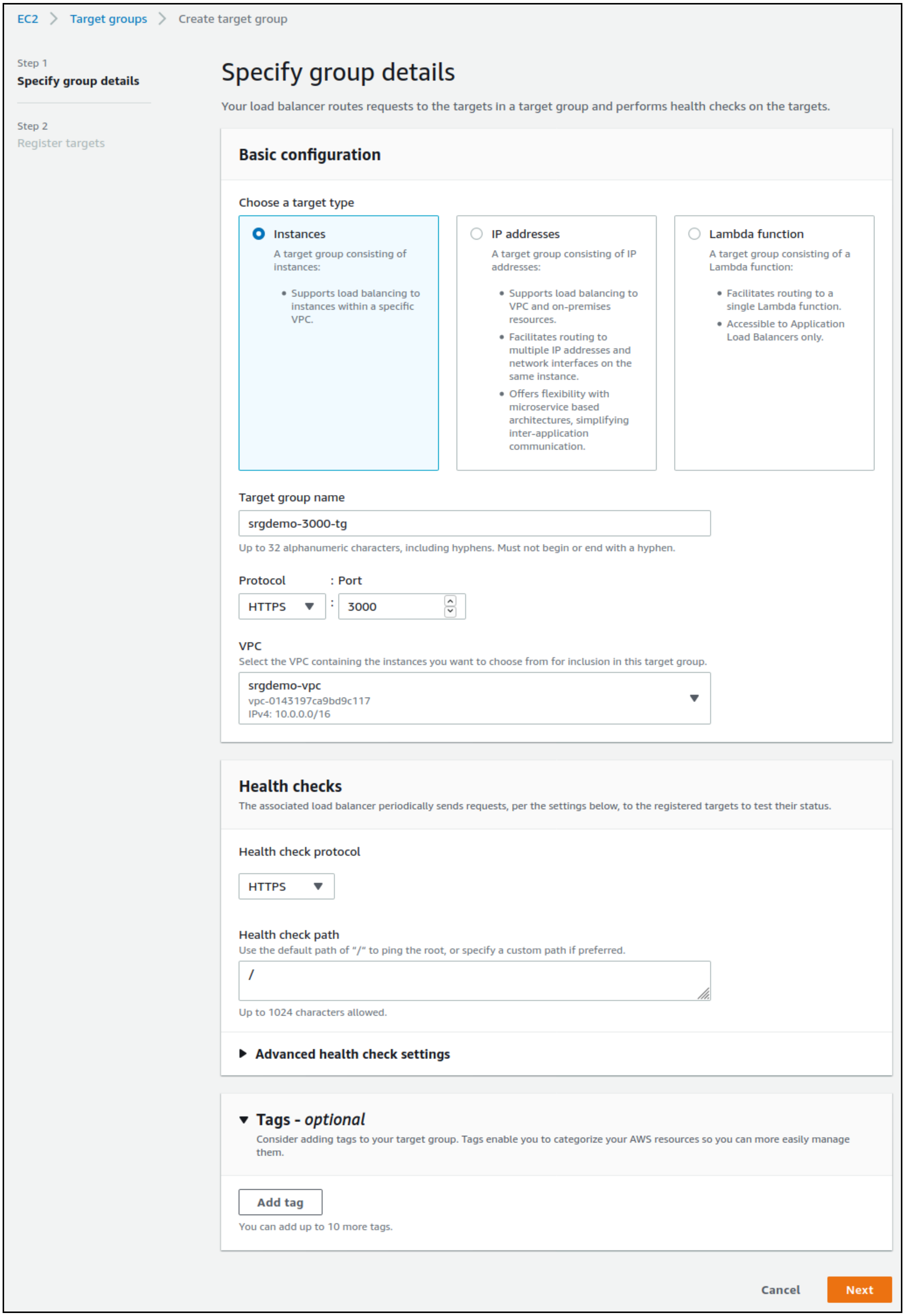
To create, tag, and add targets to the target group for port 3000 using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Target Groups**.
3. On the **Target Groups** management page, click **Create target group.**



□	Name ▲	ARN	Port	Protocol	Target type	Load balancer	VPC ID
□	srgdemo-3000-tg	arn:aws:elasticload...	3000	HTTPS	Instance	srgdemo-alb	vpc-0143197ca9bd9c117

4. On the **Specify group details** page, enter values for the following:
   a. Under **Choose a target type,** select **Instances**.
   b. **Target group name:** Choose a descriptive name for easier identification; for example *srgdemo-3000-tg*
   c. **Protocol:** change to HTTPS
   d. **Port:** enter 3000
   e. **VPC:** select your VPC
   f. **Tags:** Optionally, add descriptive tags as desired.



EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Choose a target type

● Instances

A target group consisting of instances:

- Supports load balancing to instances within a specific VPC.

○ IP addresses

A target group consisting of IP addresses:

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.

○ Lambda function

A target group consisting of a Lambda function:

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Target group name

srgdemo-3000-tg

Up to 32 alphanumeric characters, including hyphens. Must not begin or end with a hyphen.

Protocol : Port

HTTPS ▼ : 3000

VPC

Select the VPC containing the instances you want to choose from for inclusion in this target group.

srgdemo-vpc
vpc-0143197ca9bd9c117
IPv4: 10.0.0.0/16 ▼

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTPS ▼

Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

▶ **Advanced health check settings**

▼ Tags - *optional*

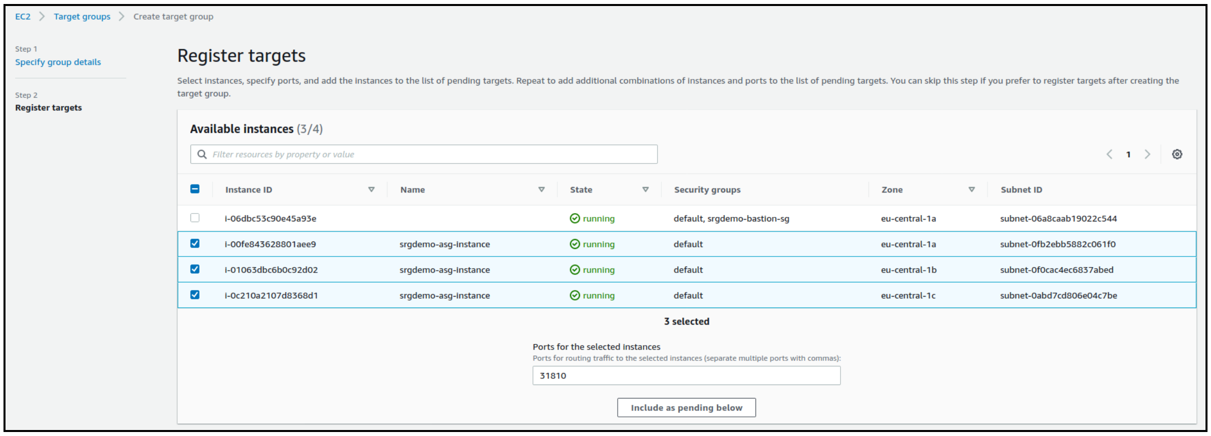
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add tag

You can add up to 10 more tags.

Cancel Next

5. Click **Next**.
6. On the **Register Targets** page, set values for the following:
   a. **Available instances:** Select your worker node instances, but **do not** select the bastion.
   b. **Ports:** For the selected instances, use the value you retrieved previously for the bastion as the corresponding node port for port 3000 and recorded in the AWS worksheet.


EC2 > Target groups > Create target group
Step 1
Specify group details
Step 2
Register targets

Register targets

Select instances, specify ports, and add the instances to the list of pending targets. Repeat to add additional combinations of instances and ports to the list of pending targets. You can skip this step if you prefer to register targets after creating the target group.

Available instances (3/4)

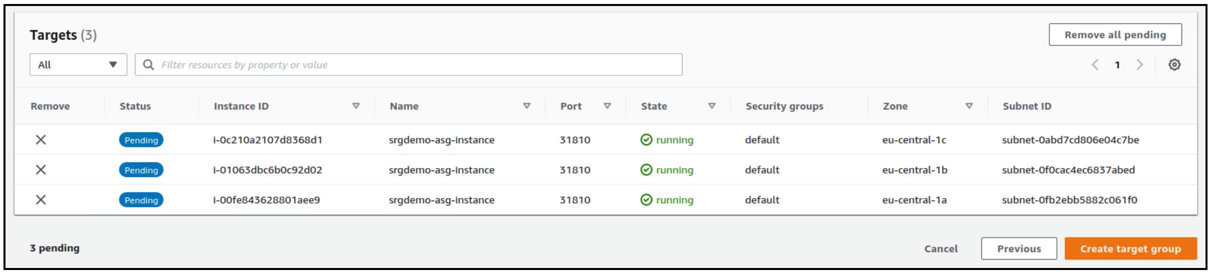
Filter resources by property or value
1

☐	Instance ID	Name	State	Security groups	Zone	Subnet ID
☐	i-06dbc53c90e45a93e		✓ running	default, srgdemo-bastion-sg	eu-central-1a	subnet-06a8caab19022c544
☑	i-00fe843628801aee9	srgdemo-asg-instance	✓ running	default	eu-central-1a	subnet-0fb2ebb5882c061f0
☑	i-01063dbc6b0c92d02	srgdemo-asg-instance	✓ running	default	eu-central-1b	subnet-0f0cac4ec6837abed
☑	i-0c210a2107d8368d1	srgdemo-asg-instance	✓ running	default	eu-central-1c	subnet-0abd7cd806e04c7be

3 selected

Ports for the selected instances
Ports for routing traffic to the selected instances (separate multiple ports with commas):
31810

7. Click **Include as pending below.** All selected instances will be added to the list of pending instances.


Targets (3)

Remove	Status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
✕	Pending	i-0c210a2107d8368d1	srgdemo-asg-instance	31810	✓ running	default	eu-central-1c	subnet-0abd7cd806e04c7be
✕	Pending	i-01063dbc6b0c92d02	srgdemo-asg-instance	31810	✓ running	default	eu-central-1b	subnet-0f0cac4ec6837abed
✕	Pending	i-00fe843628801aee9	srgdemo-asg-instance	31810	✓ running	default	eu-central-1a	subnet-0fb2ebb5882c061f0

3 pending

8. Click **Create target group.**
9. You will be redirected back to the target group management page.
10. From the list, select the newly created target group.
11. From the bottom of the page, record its ARN in the AWS worksheet.

To create, tag, and assign targets to the target group for port 3000 using the CLI:

1. Run the following command:

```
# aws elbv2 create-target-group \
--name <Target group 3000 Name> \
--protocol HTTPS \
--port 3000 \
--vpc-id <VPC ID> \
--health-check-protocol HTTPS \
--target-type instance
```

Parameters:

`<Target group 3000 Name>`: Choose some descriptive name such as `srgdemo-3000-tg`. Record the value in the AWS worksheet.

`<VPC ID>`: The ID of your VPC, as recorded in your AWS worksheet.

Example input and output:

```
# aws elbv2 create-target-group --name srgdemo-3000-tg --protocol HTTPS --
port 3000 --vpc-id vpc-0143197ca9bd9c117 --health-check-protocol HTTPS --
target-type instance
```

```
Target group for port 3000 description
```

```
{
```

```
"TargetGroupArn": "arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-3000-tg/c0684be94405b6b7",
```

```
"TargetGroupName": "srgdemo-3000-tg",
```

```
"Protocol": "HTTPS",
```

```
"Port": 3000,
```

```
"VpcId": "vpc-0143197ca9bd9c117",
```

```
"HealthCheckProtocol": "HTTPS",
```

```
"HealthCheckPort": "traffic-port",
```

```
"HealthCheckEnabled": true,
```

```
"HealthCheckIntervalSeconds": 30,
```

```
"HealthCheckTimeoutSeconds": 5,
```

```
"HealthyThresholdCount": 5,
```

```
"UnhealthyThresholdCount": 2,
```

```
"HealthCheckPath": "/",
```

```
"Matcher": {
```

```
"HttpCode": "200"
```

```
},
```

```
"TargetType": "instance"
```

```
}
```

```
]
```

```
}
```

From the output, record the value of `TargetGroupArn` in your AWS worksheet.

Tagging the Target Group (CLI)

Optionally, you can tag the target group for easier identification.

To tag the target group using the CLI:

1. Run the following command:
   ```
   # aws elbv2 add-tags \
   --resource-arns <Target group 3000 ARN> \
   --tags Key=owner,Value=<owner>
   ```

Example:

```
# aws elbv2 add-tags \
--resource-arns arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-3000-tg/c0684be94405b6b7 \
--tags Key=owner,Value=srgdemo
```

Adding Targets to the Target Group Using the CLI

To add targets to the target group:

1. Run the following command:
   ```
   # aws elbv2 register-targets \
   --target-group-arn <Target group 3000 ARN> \
   --targets Id="Instance 1 ID,Port=<Node port for 3000>" Id="Instance 2
   ID,Port=<Node port for 3000>" Id="Instance 3 ID,Port=<Node port for 3000>"
   ```

Parameters:

`<Instance x ID>`: Use the instance IDs you gathered for instances of the Auto Scaling group. Refer to the AWS worksheet for these values.

`<Node port for 3000>`: Use the port number for 3000 from your AWS worksheet.

Example:

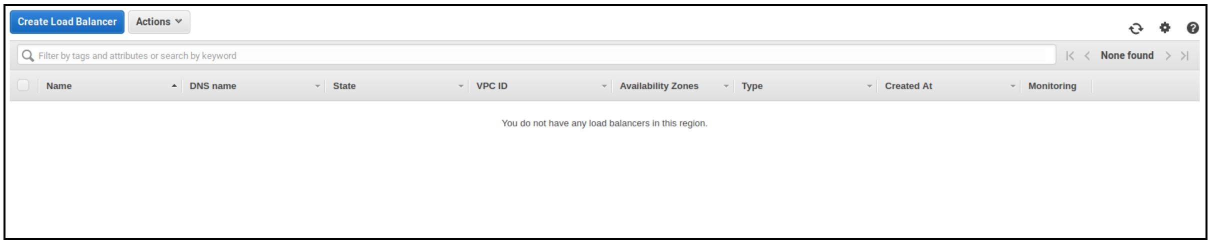
```
# aws elbv2 register-targets \
--target-group-arn arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-3000-tg/c0684be94405b6b7 \
--targets Id="i-05662f9ef84c182ca,Port=30058" Id="i-
07cfcd6716e9890b5,Port=30058" Id="i-08d819b5ccabe83cb,Port=30058"
```

Next Step: Creating the Application Load Balancer

Creating the Application Load Balancer

To create the ALB using the Web UI:

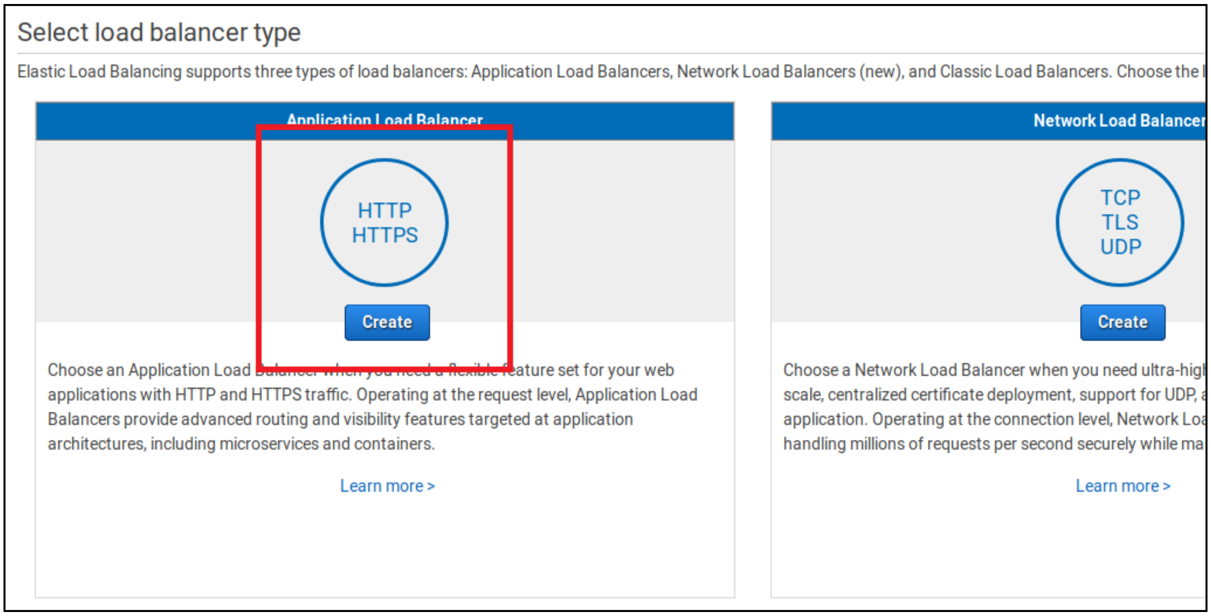
1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation panel, under **Load Balancing**, select **Load Balancers.**



Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring

You do not have any load balancers in this region.

3. On the **Load Balancing** management page, click **Create Load Balancer.**



Select load balancer type

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers (new), and Classic Load Balancers. Choose the l

Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

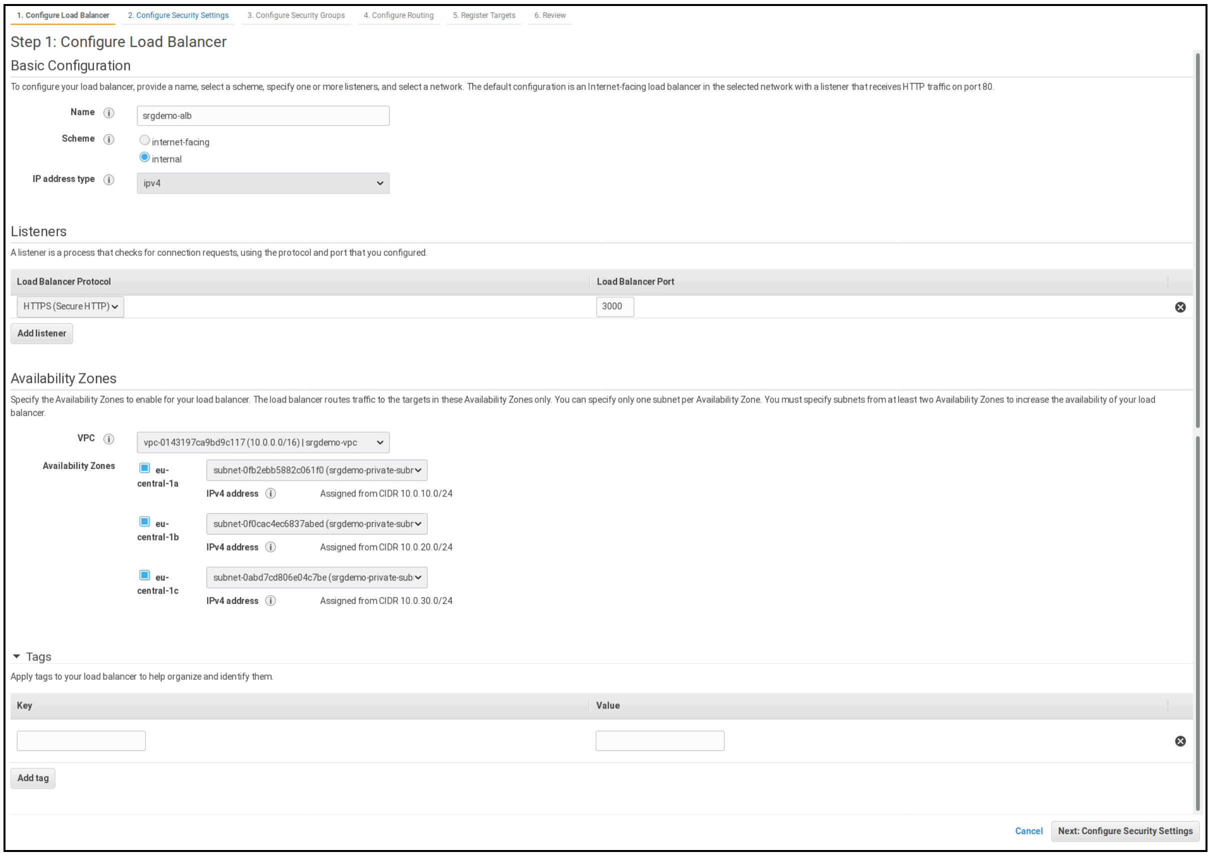
Learn more >

Network Load Balancer

Choose a Network Load Balancer when you need ultra-high scale, centralized certificate deployment, support for UDP, a application. Operating at the connection level, Network Loa handling millions of requests per second securely while ma

Learn more >

4. In the **Application Load Balancer** panel to the left, click **Create**.



1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name: srgdemo-alb

Scheme: ○ internet-facing ● internal

IP address type: ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTPS (Secure HTTP)	3000

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC: vpc-0143197ca9bd9c117 (10.0.0.0/16) | srgdemo-vpc

Availability Zones:

☑ eu-central-1a: subnet-0fb2ebb5882c061f0 (srgdemo-private-subr
IPv4 address: Assigned from CIDR 10.0.10.0/24

☑ eu-central-1b: subnet-0f0cac4ec6837abed (srgdemo-private-subr
IPv4 address: Assigned from CIDR 10.0.20.0/24

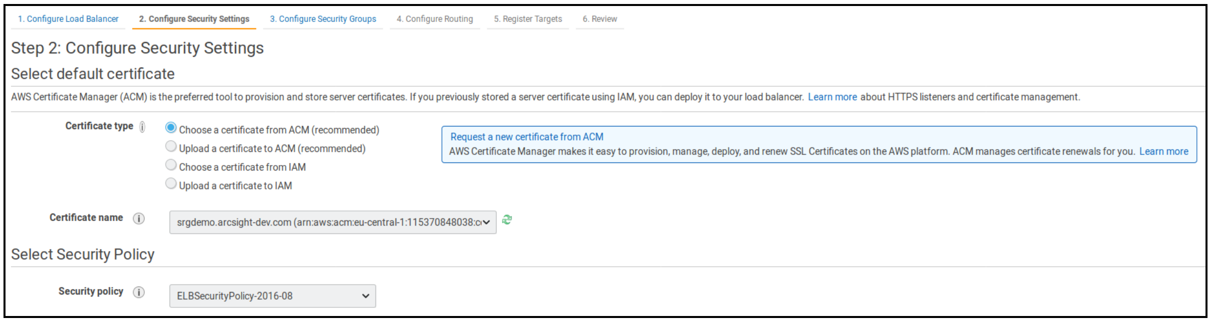
☑ eu-central-1c: subnet-0abd7cd806e04c7be (srgdemo-private-sub
IPv4 address: Assigned from CIDR 10.0.30.0/24

Tags

Apply tags to your load balancer to help organize and identify them.

Key	Value

5. On the **Configure Load Balancer** page, enter values for the following:
   a. **Name:** Choose an ALB name
   b. **Scheme:** Internal
   c. **Protocol:** HTTPS
   d. **Port:**3000
   e. **VPC:** Choose your VPC
   f. **Availability Zones:** Check all three zones, then in each, select corresponding private subnet.
   g. **Tags:** Optionally, enter any tags desired.
6. Click **Next: Configure Security Settings**.



1. Configure Load Balancer | 2. Configure Security Settings | 3. Configure Security Groups | 4. Configure Routing | 5. Register Targets | 6. Review

Step 2: Configure Security Settings

Select default certificate

AWS Certificate Manager (ACM) is the preferred tool to provision and store server certificates. If you previously stored a server certificate using IAM, you can deploy it to your load balancer. Learn more about HTTPS listeners and certificate management.

Certificate type
● Choose a certificate from ACM (recommended)
○ Upload a certificate to ACM (recommended)
○ Choose a certificate from IAM
○ Upload a certificate to IAM

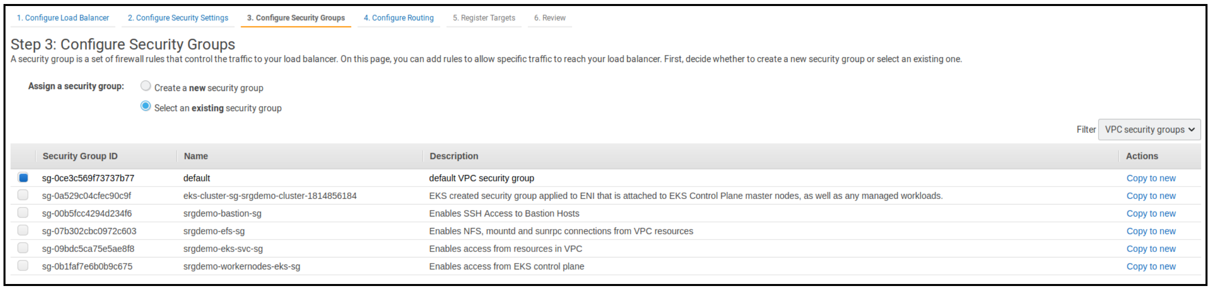
Request a new certificate from ACM
AWS Certificate Manager makes it easy to provision, manage, deploy, and renew SSL Certificates on the AWS platform. ACM manages certificate renewals for you. Learn more

Certificate name srgdemo.arcsight-dev.com (arn:aws:acm:eu-central-1:115370848038:c

Select Security Policy

Security policy ELBSecurityPolicy-2016-08

7. On the **Configure security settings** page, enter values for the following:
   a. **Certificate type:** Choose a certificate from ACM (recommended)
   b. **Certificate name:** Choose the certificate you have previously created
   c. **Security policy:** Leave unchanged.
8. Click **Next: Configure Security Groups.**



1. Configure Load Balancer | 2. Configure Security Settings | 3. Configure Security Groups | 4. Configure Routing | 5. Register Targets | 6. Review

Step 3: Configure Security Groups

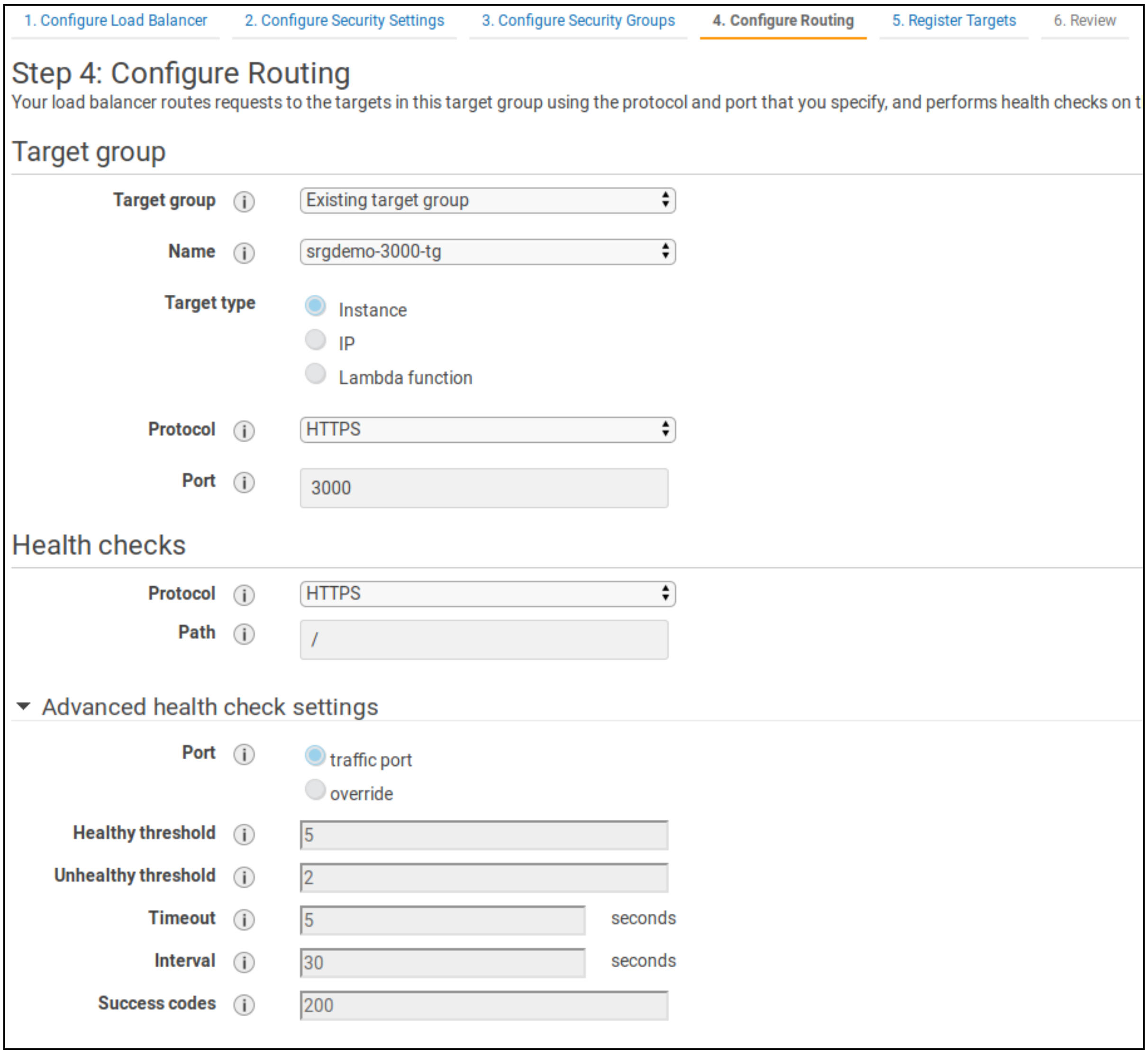
A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:
○ Create a **new** security group
● Select an **existing** security group

Filter VPC security groups

	Security Group ID	Name	Description	Actions
☑	sg-0ce3c569f73737b77	default	default VPC security group	Copy to new
□	sg-0a529c04cfec90c9f	eks-cluster-sg-srgdemo-cluster-1814856184	EKS created security group applied to ENI that is attached to EKS Control Plane master nodes, as well as any managed workloads.	Copy to new
□	sg-00b5fcc4294d234f6	srgdemo-bastion-sg	Enables SSH Access to Bastion Hosts	Copy to new
□	sg-07b302cbc0972c603	srgdemo-efs-sg	Enables NFS, mountd and sunrpc connections from VPC resources	Copy to new
□	sg-09bdc5ca75e5ae8f8	srgdemo-eks-svc-sg	Enables access from resources in VPC	Copy to new
□	sg-0b1faf7e6b0b9c675	srgdemo-workernodes-eks-sg	Enables access from EKS control plane	Copy to new

9. On the **Configure security groups** page, enter values for the following:
   a. **Assign a security group:** Leave on Select an existing security group
   b. In the list below, make sure only the Intra VPC security group created by your AWS infrastructure administrators (and recorded on the AWS worksheet) is selected.
10. Click **Next: Configure Routing**.


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on t

Target group

Target group: Existing target group

Name: srgdemo-3000-tg

Target type: ● Instance ○ IP ○ Lambda function

Protocol: HTTPS

Port: 3000

Health checks

Protocol: HTTPS

Path: /

Advanced health check settings

Port: ● traffic port ○ override

Healthy threshold: 5

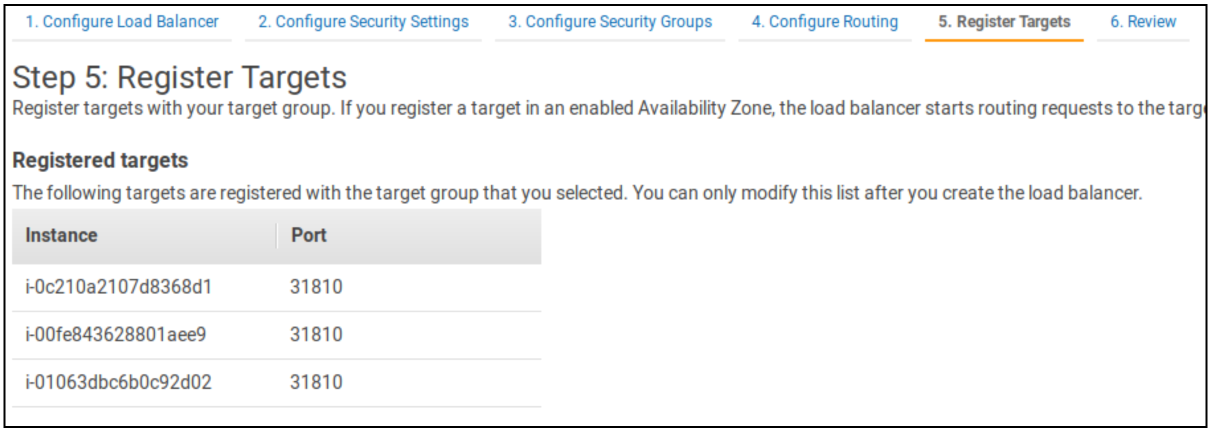
Unhealthy threshold: 2

Timeout: 5 seconds

Interval: 30 seconds

Success codes: 200

11. On the **Configure routing** page, enter values for the following:
    a. **Target group:** Change to **Existing target group.**
    b. **Name:** Choose the target group you have previously created.
12. Click **Next: Register** targets.


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targ

Registered targets

The following targets are registered with the target group that you selected. You can only modify this list after you create the load balancer.

Instance	Port
i-0c210a2107d8368d1	31810
i-00fe843628801aee9	31810
i-01063dbc6b0c92d02	31810

13. On the **Register targets** page, select (check) your instance IDs and node ports.
14. Click **Next: Review.**
15. On the **Review** page, verify all settings and then click **Create.** An ALB creation status will be displayed but will not be updated. (The page is not dynamic.) However, you might close the wizard.

To create the ALB using the CLI:

1. Run the following command:
```
# aws elbv2 create-load-balancer \
--name <ALB Name> \
--subnets <subnetIds> \
--security-groups <Intra VPC Security group Id> \
--scheme internal \
--type application \
--ip-address-type ipv4
```

Parameters:

`<ALB Name>`: Choose a name for easy application load balancer identification, and record it to the AWS worksheet.

`<subnet Ids>`: Use the space-separated IDs of all three private subnets in the VPC.

`<Intra VPC Security group Id>`: ID of the Intra VPC security group created previously and recorded in the AWS worksheet.

Example input and output:

```
#aws elbv2 create-load-balancer \
--name srgdemo-alb \
--subnets subnet-0fb2ebb5882c061f0 subnet-0f0cac4ec6837abed subnet-
0abd7cd806e04c7be \
--security-groups sg-0ce3c569f73737b77 \
--scheme internal \
--type application \
--ip-address-type ipv4
```

```
{
   "LoadBalancers":[
      {
         "LoadBalancerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:loadbalancer/app/srgdemo-alb/8718b24107ef591b",
         "DNSName":"internal-srgdemo-alb-505957021.eu-central-
1.elb.amazonaws.com",
         "CanonicalHostedZoneId":"Z215JYRZR1TBD5",
         "CreatedTime":"2020-06-15T09:16:12.480000+00:00",
         "LoadBalancerName":"srgdemo-alb",
         "Scheme":"internal",
         "VpcId":"vpc-0143197ca9bd9c117",
         "State":{
            "Code":"provisioning"
         },
         "Type":"application",
         "AvailabilityZones":[
            {
               "ZoneName":"eu-central-1c",
               "SubnetId":"subnet-0abd7cd806e04c7be",
               "LoadBalancerAddresses":[

               ]
            },
            {
               "ZoneName":"eu-central-1b",
               "SubnetId":"subnet-0f0cac4ec6837abed",
               "LoadBalancerAddresses":[

               ]
            },
            {
               "ZoneName":"eu-central-1a",
               "SubnetId":"subnet-0fb2ebb5882c061f0",
               "LoadBalancerAddresses":[

               ]
            }
         ],
         "SecurityGroups":[
            "sg-0ce3c569f73737b77"
         ],
         "IpAddressType":"ipv4"
      }
   ]
}
```

2. Record the `LoadBalancerArn`, `DNSName` and `CanonicalHostedZoneId` values in the AWS worksheet as **`ALB ARN,`** **`ALB DNS name`** and **`ALB Canonical hosted zone ID.`**
3. Creation of the ALB takes approximately 5 minutes. To check the ALB creation status, run the following command using the `ALB ARN` value:
   ```
   #aws elbv2 describe-load-balancers \
   --load-balancer-arns <LoadBalancerArn> \
   | jq -r '.LoadBalancers[0].State.Code'
   ```
4. Repeat Step 3 until the returned status changes to *Active*.

Do not proceed until the status has changed to Active.

Example

```
# aws elbv2 describe-load-balancers \
--load-balancer-arns arn:aws:elasticloadbalancing:eu-central-
1:115370848038:loadbalancer/app/srgdemo-alb/8718b24107ef591b \
| jq -r '.LoadBalancers[0].State.Code'
```

Adding the Listener for Port 3000

This action will connect the ALB to NLB 3000 through the target group. Incoming requests to the ALB on port 3000 will be directed to the Kubernetes `itom-cdf-ingress-frontend-svc` service.

To add the listener:

1. Run the following command:
   ```
   # aws elbv2 create-listener \
   --load-balancer-arn <ALB ARN> \
   --protocol HTTPS \
   --port 3000 \
   --certificates CertificateArn=<Certificate ARN> \
   --default-actions Type=forward,TargetGroupArn=<Target group 3000 ARN>
   ```

Parameters:

`<ALB ARN>`: Use ALB ARN recorded in the AWS worksheet.

`<Certificate ARN>`: Use certificate ARN recorded in the AWS worksheet.

`<Target group 3000 ARN>`: Use the target group for port 3000 ARN recorded in the AWS worksheet.

Example input and output:

```
# aws elbv2 create-listener \
--load-balancer-arn arn:aws:elasticloadbalancing:eu-central-
1:115370848038:loadbalancer/app/srgdemo-alb/8718b24107ef591b \
--protocol HTTPS --port 3000 --certificates CertificateArn=arn:aws:acm:eu-
central-1:115370848038:certificate/691ec232-98ff-45ed-8e69-1d15c0447538 \
```

```
--default-actions \
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-3000-tg/c0684be94405b6b7
```

```
"Listeners":[
   {
      "ListenerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:listener/app/srgdemo-alb/8718b24107ef591b/32a42e4edb52466b",
      "LoadBalancerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:loadbalancer/app/srgdemo-alb/8718b24107ef591b",
      "Port":3000,
      "Protocol":"HTTPS",
      "Certificates":[
         {
            "CertificateArn":"arn:aws:acm:eu-central-
1:115370848038:certificate/691ec232-98ff-45ed-8e69-1d15c0447538"
         }
      ],
      "SslPolicy":"ELBSecurityPolicy-2016-08",
      "DefaultActions":[
         {
            "Type":"forward",
            "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-3000-tg/c0684be94405b6b7",
            "ForwardConfig":{
               "TargetGroups":[
                  {
                     "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-
central-1:115370848038:targetgroup/srgdemo-3000-tg/c0684be94405b6b7",
                     "Weight":1
                  }
               ],
               "TargetGroupStickinessConfig":{
                  "Enabled":false
               }
            }
         }
      ]
   }
]
}
```

Next Step: Directing the Route 53 RecordSet to the ALB

Directing the Route 53 Record Set to the ALB

Although it is technically possible to connect to the ALB using its DNS name (such as `internal-srgdemo-alb-505957021.eu-central-1.elb.amazonaws.com`), this is not recommended for few reasons:

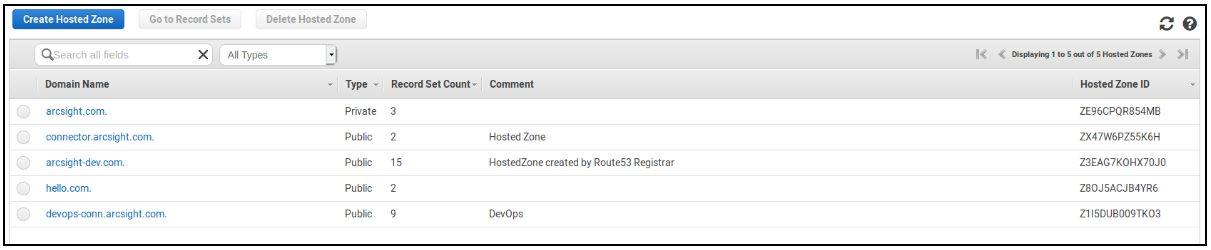
- The URL is hard to remember and a user is forced to bookmark it to use it.
- You are unable to create the certificate for this domain, so browsers will always warn a user about the insecure connection

Previously, we have created a record set in the Route 53 hosted zone and requested a certificate for the chosen domain name. You can now direct the record set to the application load balancer.

Using the Web UI

To direct the Route 53 record set to the ALB using the web UI:

1. Using the Find Services search tool, locate and browse to the Route 53 Dashboard.
2. In the left navigation panel, select **Hosted zones.** (Ignore any errors generated during this process.)
3. From the hosted zones list, select the same hosted zone as you chose for creating the new Route 53 record set. (Use the search box to search for the zone if necessary.)



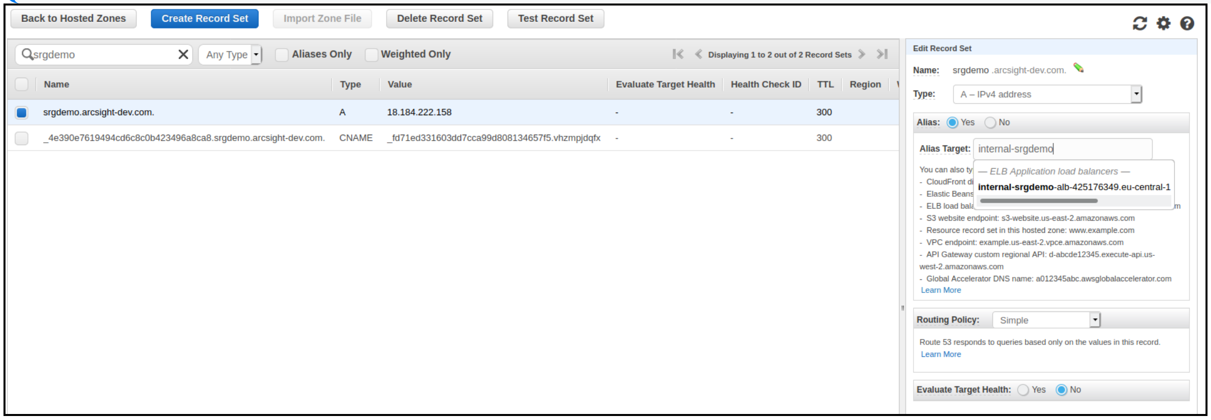
	Domain Name	Type	Record Set Count	Comment	Hosted Zone ID
○	arcsight.com.	Private	3		ZE96CPQR854MB
○	connector.arcsight.com.	Public	2	Hosted Zone	ZX47W6PZ55K6H
○	arcsight-dev.com.	Public	15	HostedZone created by Route53 Registrar	Z3EAG7KOHX70J0
○	hello.com.	Public	2		Z8OJ5ACJB4YR6
○	devops-conn.arcsight.com.	Public	9	DevOps	Z1I5DUB009TKO3

4. Select the main record set.

> The selected record set is the one that will be edited. The other one is verification for the certificate.

5. In the details pane on the right, set values for the following:
   a. **Alias:** Change to yes
   b. **Alias Target:** Start typing `internal-<domain name>`; for example, `internal-srgdemo`. The long list will be filtered and only your ALB will be displayed. Select it.

> Upon selection the word *dialstack* will be prepended to the name you entered. This can be ignored.



	Name	Type	Value	Evaluate Target Health	Health Check ID	TTL	Region
☑	srgdemo.arcsight-dev.com.	A	18.184.222.158	-	-	300	
□	_4e390e7619494cd6c8c0b423496a8ca8.srgdemo.arcsight-dev.com.	CNAME	_fd71ed331603dd7cca99d808134657f5.vhzmpjdqfx	-	-	300	

Edit Record Set

Name: srgdemo .arcsight-dev.com.

Type: A – IPv4 address

Alias: ● Yes ○ No

Alias Target: internal-srgdemo

— ELB Application load balancers —

internal-srgdemo-alb-425176349.eu-central-1

You can also ty
- CloudFront di
- Elastic Beans
- ELB load bala
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com
- VPC endpoint: example.us-east-2.vpce.amazonaws.com
- API Gateway custom regional API: d-abcde12345.execute-api.us-west-2.amazonaws.com
- Global Accelerator DNS name: a012345abc.awsglobalaccelerator.com

Routing Policy: Simple

Route 53 responds to queries based only on the values in this record.

Evaluate Target Health: ○ Yes ● No

6. Click **Save Record Set.**

Using the CLI

To direct the Route53 RecordSet to the ALB Using the CLI:

1. Copy the template `UpdateRecordSetToALB.json` and open the copy in a text editor. (This template is available in the `aws-byok-installer-<version>/objectdefs/` subfolder.).
2. Edit the following:
   a. **Record name:** Combine the subdomain (such as `srgdemo`) and hosted zone name (for example, `arcsight-dev.com`); you can also refer to the AWS worksheet for the RecordSet name.
   b. **ALB Canonical hosted zone ID:** Get the value from the AWS worksheet.
   c. **ALB DNS name:** Get the value from the AWS worksheet
3. Run the following command:
```
# aws route53 change-resource-record-sets \
--hosted-zone-id <Hosted zone Id> \
--change-batch file://UpdateRecordSetToALB.json
```

Describing Parameters

`<Hosted zone Id>`: Use the hosted zone ID retrieved previously for Route 53 hosted zone, refer to the AWS worksheet,`Hosted zone ID`. For example, `/hostedzone/Z3EAG7KOHX70J0`.

`<change-batch file>`: Replace the parameter with the name of your own modified instance of the linked JSON template.

Next Step: Installing CDF

Installing CDF and Products

With CDF bootstrapped, the next step in installing CDF and the ArcSight Suite is to connect to the CDF web installation UI, and then proceed through the installation wizard.

Accessing the CDF Installation UI

At the end of CDF bootstrap process, you were prompted to connect to the URL `https://<external access host>:3000`, which is part of the standard CDF installation procedure.

The CDF installation port 3000 is now accessible through the chosen Route 53 record set, but only within the VPC. The VPC and any resources inside it are isolated from access from the internet (except for the bastion host, which is accessible on port 22, the SSH port).

You cannot access the created DNS record outside the VPC, since that DNS record will resolve to one of the three private subnet IP addresses which are hidden (and, in our case, in a private A-class IP range).

There are several methods for connecting a browser to the CDF port 3000; these methods are discussed here.

Forwarding DISPLAY

Prerequisite: operating system capable of running X-server, such as *nix, linux, MacOS.

The easiest and fastest option for connection is to connect to the bastion using SSH with the -X or -Y switch. That will set the remote DISPLAY accordingly, so the process running remotely will render its UI on the local X-server. The bastion host you configured earlier has the Mozilla Firefox browser installed.

> The drawback of this method is that only one user can be connected and use the web browser, and the browser response might be quite slow. Any subsequent user will receive a message that the browser is already running, and results in significant lag while in the browser. However, the browser is only used for installation and configuration tasks, which are typically done once and by a single user, so the impact will likely be small.

To connect with this method, do the following:

1. Connect (SSH) to the bastion host with the additional parameters for dbus. Example command:
   ```
   ssh -i /{path to ssh key} /aws.pem -X centos@54.188.142.125 'firefox
   ```

```
https://srgdemo.arcsight-dev.com:3000'
```

2. Browse to the URL that CDF returned at the end of its CLI installation. For example:
   `https://srgdemo.arcsight-dev.com:3000`

Forwarding local ports

Prerequisite: Ability to execute SSH with command line switches, as well as the Web UI ability to edit the system file `/etc/hosts` or the corresponding file.

To connect with this method, connect to the bastion host, adding the `-L` parameter. Example:

```
# ssh -i .ssh/srgdemo.pem -L 3000:srgdemo.arcsight-dev.com:3000
centos@3.120.237.11
```

The `-L` parameter opens local port 3000 and connects each request to the `srgdemo.arcsight-dev.com port 3000` on the remote side. So the bastion will resolve `srgdemo.arcsight-dev.com` and opens a connection to it on port 3000.

The second part of this approach is to edit `/etc/hosts`, and add your domain to the line containing `localhost`. Example:

```
127.0.0.1 localhost srgdemo.arcsight-dev.com
```

Open your preferred browser and direct it to the address that CDF output at the end of its CLI installation. Here we will use the example:
`https://srgdemo.arcsight-dev.com:3000.`

CDF Web UI Installation

Once you have chosen your connection method and successfully connected to the CDF installation portal, perform the steps outlined to complete the CDF installation. During the usual installation process there are two steps where optional additional task or special handling might occur; during downloading images and file storage. These are explained in more detail here.

Downloading Images

Downloading images requires the CDF/K8s access to the ECR and checking the presence of respective Docker images there. If it has been more than 12 hours between the bootstrapping CDF and checking image availability the ECR credentials will expire, and you need to provide CDF/K8s new credentials.

Follow these procedures to refresh the ECR credentials: Refresh the ECR credentials in the K8s

File Storage

When setting the File Storage it is not possible to use the auto-discovery feature of remote mount points.

For File Server, supply the value of the `Filesystem FQDN` from the AWS worksheet.

Then click on the double-arrows and fill in the path to the volume. In our example it will be `/srgdemo/arcsight-volume`. This value was displayed as first the output of the `init_efs` script.

Installation finished

At the end of CDF installation, you are prompted to connect to the CDF management portal on the same host, this time using port 5443. Connection to port 5443 is not possible yet, as more network resources need to be configured.

To continue the appropriate flow, see "Post Installation Network Configuration" below

Post Installation Network Configuration

During the CDF web UI installation, new services providing CDF management and re-configuration were created. These services listen on different ports than the CDF installation UI. CDF management listens on port 5443 and re-configuration on port 443. In this section we will perform network configuration for these ports.

Get CDF Ingress Service Node Port for Port 5443

To get the ingress service node port for port 5443:

1. Run the following command:
   ```
   # kubectl get svc -n core | grep nginx-ingress-controller-svc
   ```

Example output:

```
nginx-ingress-controller-svc NodePort 172.20.26.194 <none>
5443:31704/TCP,5444:32558/TCP 170m
```

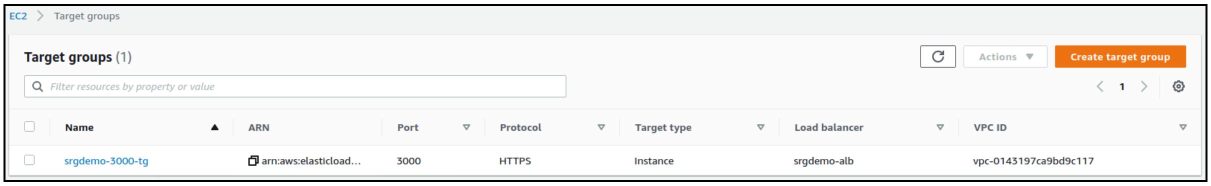
2. Record the ingress service node port for port 5443 (in the example, 31704) in the AWS worksheet.

Next Step: Create a Target Group for Port 5443

Creating a Target Group for Port 5443

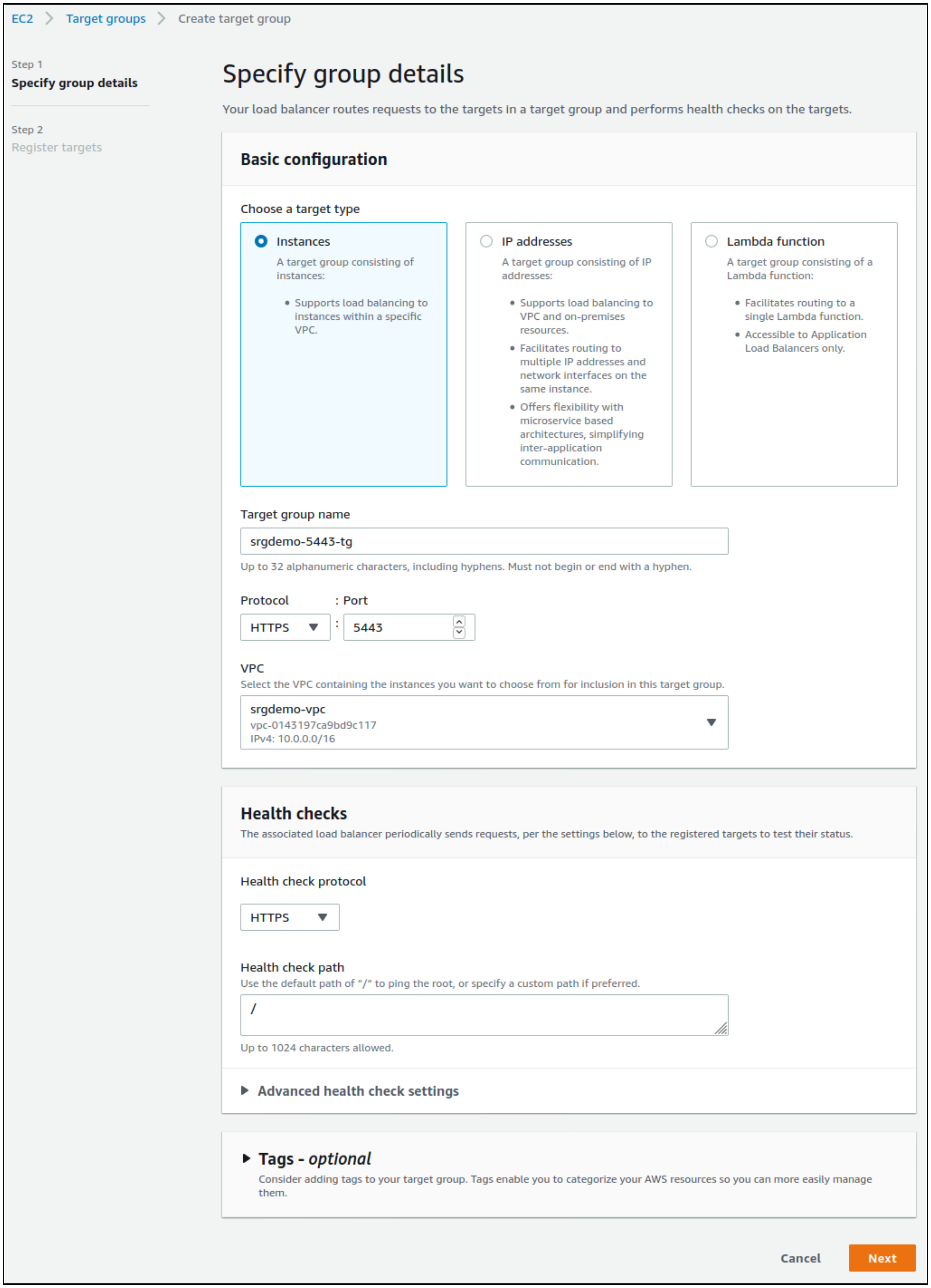
To create the target group for port 5443 using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Target Groups**.
3. On the **Target Groups** management page, click **Create target group.**



□	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
□	srgdemo-3000-tg	arn:aws:elasticload...	3000	HTTPS	Instance	srgdemo-alb	vpc-0143197ca9bd9c117

4. On the **Specify group details** page, enter values for the following:
   a. Under **Choose a target type,** select **Instances**.
   b. **Target group name:** Choose a descriptive name for easier identification; for example, *srgdemo-5443-tg*
   c. **Protocol:** change to HTTPS
   d. **Port:** 5443
   e. **VPC:** select your VPC
   f. **Tags:** Optionally, add descriptive tags as desired.



Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Choose a target type

● Instances

A target group consisting of instances:

- Supports load balancing to instances within a specific VPC.

○ IP addresses

A target group consisting of IP addresses:

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.

○ Lambda function

A target group consisting of a Lambda function:

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Target group name

srgdemo-5443-tg

Up to 32 alphanumeric characters, including hyphens. Must not begin or end with a hyphen.

Protocol : Port

HTTPS : 5443

VPC

Select the VPC containing the instances you want to choose from for inclusion in this target group.

srgdemo-vpc
vpc-0143197ca9bd9c117
IPv4: 10.0.0.0/16

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTPS

Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/

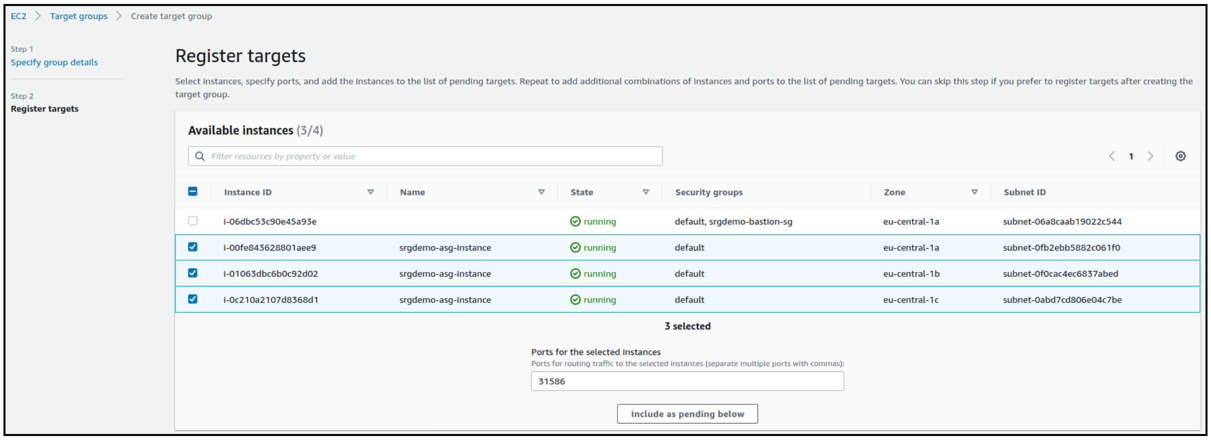
Up to 1024 characters allowed.

▶ Advanced health check settings

▶ Tags - *optional*

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

5. Click **Next**.
6. On the **Register Targets** page, set values for the following:
   a. **Available instances:** Select your instances; however, do not select the bastion.
   b. **Ports for the selected instances:** For the selected instances, use the value you retrieved previously and recorded on the AWS worksheet as `Node Port for Port 5443.`



EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Register targets

Select instances, specify ports, and add the instances to the list of pending targets. Repeat to add additional combinations of instances and ports to the list of pending targets. You can skip this step if you prefer to register targets after creating the target group.

Available instances (3/4)

Filter resources by property or value

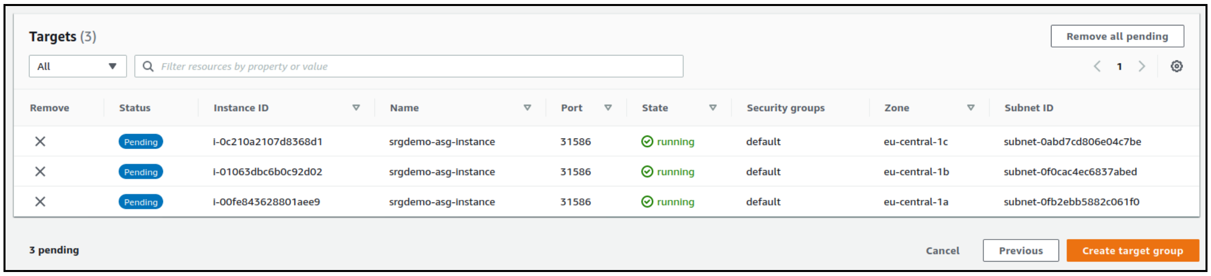
	Instance ID	Name	State	Security groups	Zone	Subnet ID
□	i-06dbc53c90e45a93e		✓ running	default, srgdemo-bastion-sg	eu-central-1a	subnet-06a8caab19022c544
☑	i-00fe843628801aee9	srgdemo-asg-instance	✓ running	default	eu-central-1a	subnet-0fb2ebb5882c061f0
☑	i-01063dbc6b0c92d02	srgdemo-asg-instance	✓ running	default	eu-central-1b	subnet-0f0cac4ec6837abed
☑	i-0c210a2107d8368d1	srgdemo-asg-instance	✓ running	default	eu-central-1c	subnet-0abd7cd806e04c7be

3 selected

Ports for the selected instances
Ports for routing traffic to the selected instances (separate multiple ports with commas):
31586

Include as pending below

7. Click **Include as pending below.** All marked instances will be added to the list of pending instances.



Targets (3)

Remove all pending

All

Filter resources by property or value

Remove	Status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
✕	Pending	i-0c210a2107d8368d1	srgdemo-asg-instance	31586	✓ running	default	eu-central-1c	subnet-0abd7cd806e04c7be
✕	Pending	i-01063dbc6b0c92d02	srgdemo-asg-instance	31586	✓ running	default	eu-central-1b	subnet-0f0cac4ec6837abed
✕	Pending	i-00fe843628801aee9	srgdemo-asg-instance	31586	✓ running	default	eu-central-1a	subnet-0fb2ebb5882c061f0

3 pending

Cancel Previous Create target group

8. Click **Create target group.**
9. You will be redirected back to the target group management page. From the list, select the newly created target group. From the bottom of the page, record its ARN in the AWS worksheet.

To create the target group for port 5443 using the CLI:

1. Run the following command:

```
# aws elbv2 create-target-group \
--name <Target group 5443 Name> \
--protocol HTTPS \
--port 5443 \
--vpc-id <VPC ID> \
--health-check-protocol HTTPS \
   --target-type instance
```

Parameters:

`<Target group 5443 Name>`: Choose some descriptive name, such as `srgdemo-5443-tg` and record the value in the AWS worksheet.

`<VPC ID>`: The ID of your VPC as recorded in your AWS worksheet.

Example input and output:

```
# aws elbv2 create-target-group \
   --name srgdemo-5443-tg \
   --protocol HTTPS \
   --port 5443 \
   --vpc-id vpc-0143197ca9bd9c117 \
   --health-check-protocol HTTPS \
   --target-type instance
```

```
{
      "TargetGroups":[
         {
            "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
   \1:115370848038:targetgroup/srgdemo-3000-tg/c0684be94405b6b7",
            "TargetGroupName":"srgdemo-5443-tg",
            "Protocol":"HTTPS",
            "Port":5443,
            "VpcId":"vpc-0143197ca9bd9c117",
            "HealthCheckProtocol":"HTTPS",
            "HealthCheckPort":"traffic-port",
            "HealthCheckEnabled":true,
            "HealthCheckIntervalSeconds":30,
            "HealthCheckTimeoutSeconds":5,
            "HealthyThresholdCount":5,
            "UnhealthyThresholdCount":2,
            "HealthCheckPath":"/",
            "Matcher":{
               "HttpCode":"200"
            },
            "TargetType":"instance"
         }
      ]
   }
```

2. From the output, record the value of `TargetGroupArn` in your AWS worksheet.

Tagging the Target Group (CLI)

Optionally, you can tag the target group for easier identification.

To tag the target group using the CLI:

1. Run the following command:
   ```
   # aws elbv2 add-tags \
   --resource-arns <Target group 5443 ARN> \
   --tags Key=owner,Value=<owner>
   ```

Example:

```
# aws elbv2 add-tags \
--resource-arns arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-5443-tg/c0684be94405b6b7 \
--tags Key=owner,Value=srgdemo
```

Next Step: Add Targets to the Target Group for Port 5443

Adding Targets to the Target Group for Port 5443

1. Run the following command:
   ```
   # aws elbv2 register-targets \
   --target-group-arn <Target group 5443 ARN> \
   --targets Id="Instance 1 ID,Port=<Node port for 5443>" Id="Instance 2
   ID,Port=<Node port for 5443>" Id="Instance 3 ID,Port=<Node port for 5443>"
   ```

Parameters:

`<Instance x ID>`: Use the instance IDs you gathered for instances of the Auto Scaling group. Refer to the AWS worksheet.

`<Target group 5443 ARN>`: ARN of the target group you just created.

`<Node port for 5443>`: Use the `node port number for 5443` from the AWS worksheet.

Example:

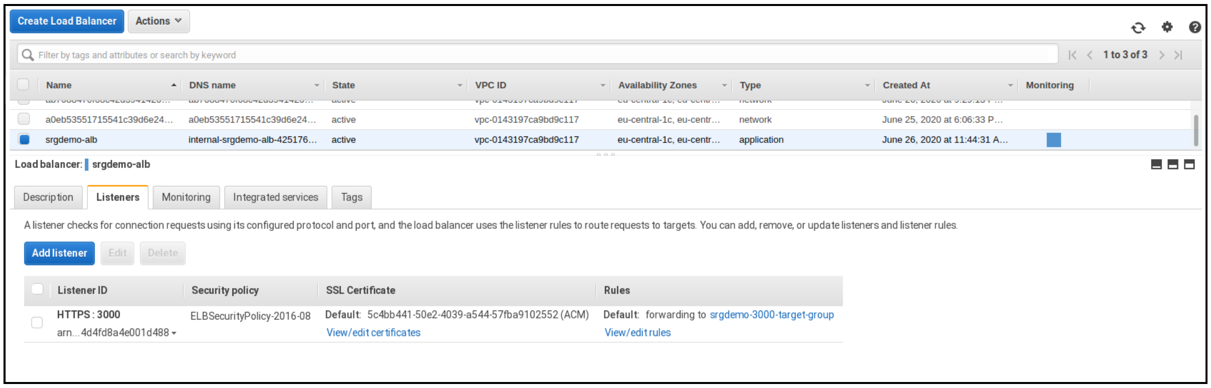
```
# aws elbv2 register-targets \
--target-group-arn arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-5443-tg/a096cb67c2f9144d \
--targets Id="i-05662f9ef84c182ca,Port=31704" Id="i-07cfcd6716e9890b5,Port=31704" Id="i-
08d819b5ccabe83cb,Port=31704"
```

Adding a Listener for Port 5443 to the ALB

Similarly to listener for port 3000, here we will create a path for requests on port 5443 to be routed to Kubernetes nginx-ingress-controller-svc service through respective target group created above.

To add a listener for port 5443 to the ALB using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Load Balancers.**
3. From the list of load balancers, select your previously created Application Load Balancer (ALB).



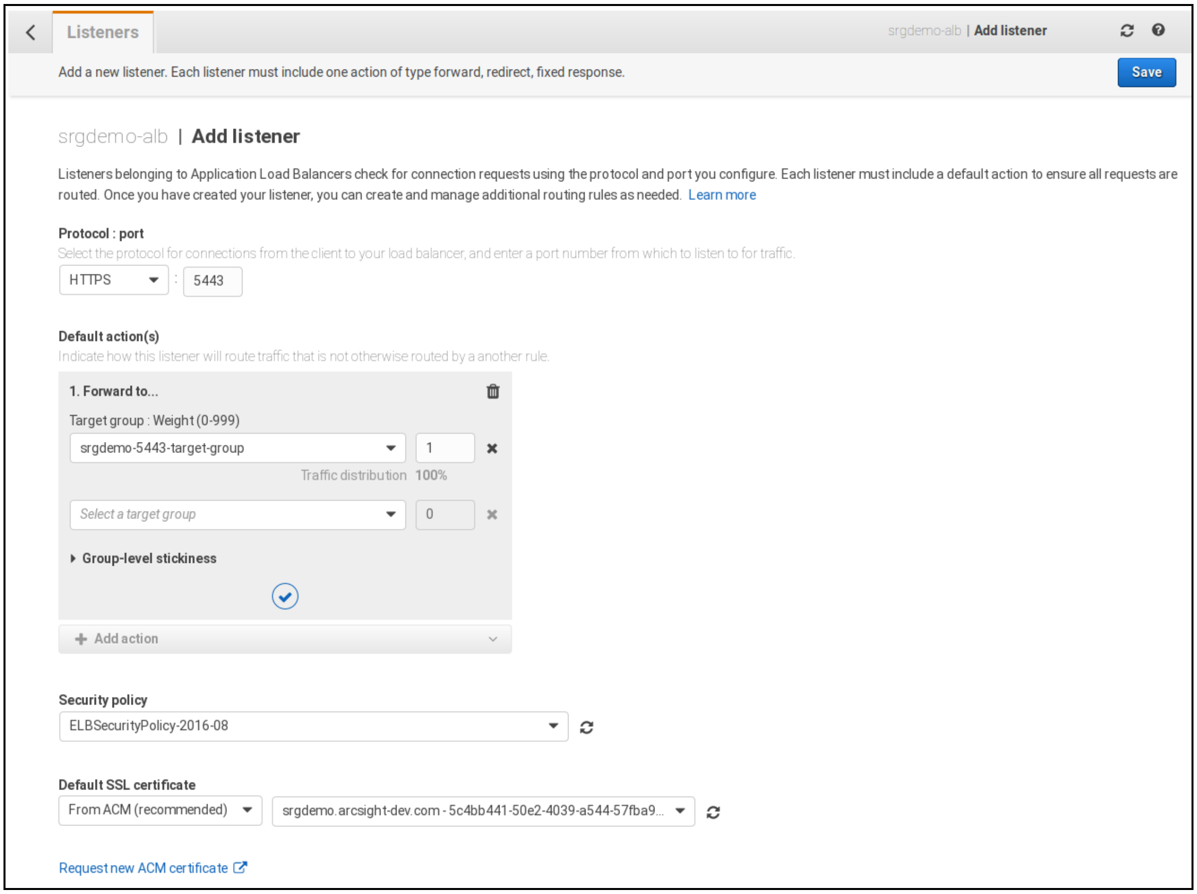
	Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
□	a0eb53551715541c39d6e24…	a0eb53551715541c39d6e24…	active	vpc-0143197ca9bd9c117	eu-central-1c, eu-centr…	network	June 25, 2020 at 6:06:33 P…	
☑	srgdemo-alb	internal-srgdemo-alb-425176…	active	vpc-0143197ca9bd9c117	eu-central-1c, eu-centr…	application	June 26, 2020 at 11:44:31 A…	

Load balancer: srgdemo-alb

A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener rules to route requests to targets. You can add, remove, or update listeners and listener rules.

	Listener ID	Security policy	SSL Certificate	Rules
□	HTTPS : 3000 arn...4d4fd8a4e001d488	ELBSecurityPolicy-2016-08	Default: 5c4bb441-50e2-4039-a544-57fba9102552 (ACM) View/edit certificates	Default: forwarding to srgdemo-3000-target-group View/edit rules

4. On the **Listeners** tab, click **Add Listener** and set values for the following:
   a. **Protocol: port:** Change to HTTPS and 5443
   b. **Default action(s):** Choose the action **Forward to...**, and then choose your target group for port 5443
   c. **Default SSL certificate:** Choose the SSL certificate you have created previously.



Add a new listener. Each listener must include one action of type forward, redirect, fixed response.

srgdemo-alb | **Add listener**

Listeners belonging to Application Load Balancers check for connection requests using the protocol and port you configure. Each listener must include a default action to ensure all requests are routed. Once you have created your listener, you can create and manage additional routing rules as needed. Learn more

Protocol : port
Select the protocol for connections from the client to your load balancer, and enter a port number from which to listen to for traffic.
HTTPS : 5443

Default action(s)
Indicate how this listener will route traffic that is not otherwise routed by a another rule.

1. Forward to...
Target group : Weight (0-999)
srgdemo-5443-target-group 1
Traffic distribution 100%
Select a target group 0

Group-level stickiness

Add action

Security policy
ELBSecurityPolicy-2016-08

Default SSL certificate
From ACM (recommended) srgdemo.arcsight-dev.com - 5c4bb441-50e2-4039-a544-57fba9...

Request new ACM certificate

5. Click **Save.**

To add a listener for port 5443 to the ALB using the CLI:

1. Run the following command:

```
# aws elbv2 create-listener \
--load-balancer-arn <ALB ARN> \
--protocol HTTPS \
--port 5443 \
--certificates CertificateArn=<Certificate ARN> \
--default-actionsType=forward,TargetGroupArn=<Target group 5443 ARN>
```

Parameters:

`<ALB ARN>`: Use the value of `ALB ARN` recorded in the AWS worksheet.

`<Certificate ARN>`: Use the value of `certificate ARN` recorded in the AWS worksheet.

`<Target group 5443 ARN>`: Use the value for `target group for port 5443 ARN` recorded in the AWS worksheet.

Example input and output:

```
# aws elbv2 create-listener \
--load-balancer-arn arn:aws:elasticloadbalancing:eu-central-
1:115370848038:loadbalancer/app/srgdemo-alb/8718b24107ef591b \
--protocol HTTPS --port 5443 \
--certificates CertificateArn=arn:aws:acm:eu-central-
1:115370848038:certificate/691ec232-98ff-45ed-8e69-1d15c0447538 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-5443-tg/a096cb67c2f9144d
```

```
Listener for port 5443 description
```

```
{
   "Listeners":[
      {
         "ListenerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:listener/app/srgdemo-alb/8718b24107ef591b/98e4aa47242b3d49",
         "LoadBalancerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:loadbalancer/app/srgdemo-alb/8718b24107ef591b",
         "Port":5443,
         "Protocol":"HTTPS",
         "Certificates":[
            {
               "CertificateArn":"arn:aws:acm:eu-central-
1:115370848038:certificate/691ec232-98ff-45ed-8e69-1d15c0447538"
            }
         ],
         "SslPolicy":"ELBSecurityPolicy-2016-08",
         "DefaultActions":[
            {
               "Type":"forward",
               "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-5443-tg/a096cb67c2f9144d",
               "ForwardConfig":{
                  "TargetGroups":[
                     {
                        "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-
central-1:115370848038:targetgroup/srgdemo-5443-tg/a096cb67c2f9144d",
                        "Weight":1
                     }
                  ],
                  "TargetGroupStickinessConfig":{
                     "Enabled":false
                  }
               }
            }
         ]
      }
```

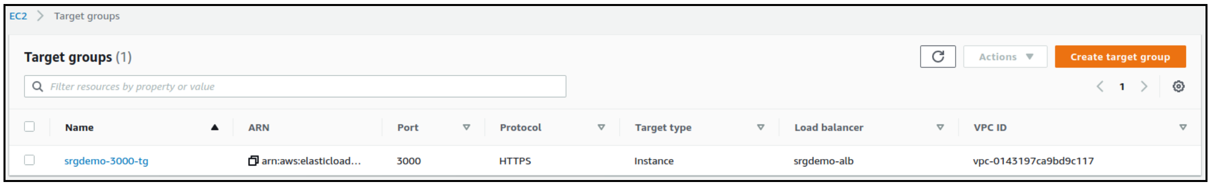
```
    ]
}
```

Next Step: Creating a Target Group for Port 443

Creating a Target Group for Port 443

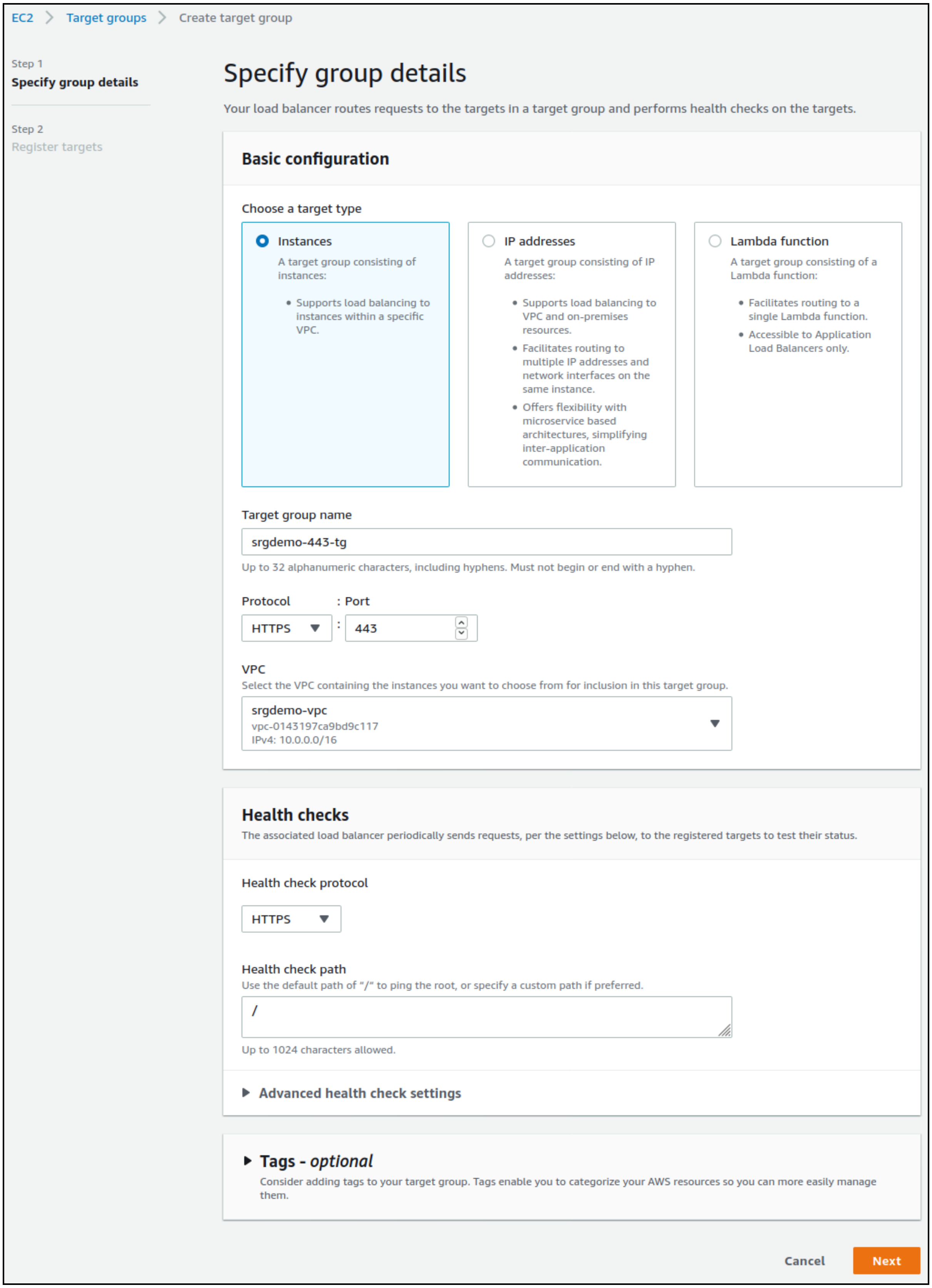
To create a target group for port 443 using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Target Groups**.
3. On the **Target Groups** management page, click **Create target group.**



□	Name ▲	ARN	Port	Protocol	Target type	Load balancer	VPC ID
□	srgdemo-3000-tg	arn:aws:elasticload...	3000	HTTPS	Instance	srgdemo-alb	vpc-0143197ca9bd9c117

4. On the **Specify group details** page, enter values for the following:
   a. Under **Choose a target type,** select **Instances**.
   b. **Target group name:** Choose a descriptive name for easier identification; for example *srgdemo-443-tg*
   c. **Protocol:** Change to HTTPS
   d. **Port:** Enter 443
   e. **VPC:** Select your VPC.
   f. **Tags:** Optionally, add descriptive tags as desired.



Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Choose a target type

● Instances
A target group consisting of instances:
- Supports load balancing to instances within a specific VPC.

○ IP addresses
A target group consisting of IP addresses:
- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.

○ Lambda function
A target group consisting of a Lambda function:
- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Target group name
srgdemo-443-tg
Up to 32 alphanumeric characters, including hyphens. Must not begin or end with a hyphen.

Protocol : Port
HTTPS : 443

VPC
Select the VPC containing the instances you want to choose from for inclusion in this target group.
srgdemo-vpc
vpc-0143197ca9bd9c117
IPv4: 10.0.0.0/16

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

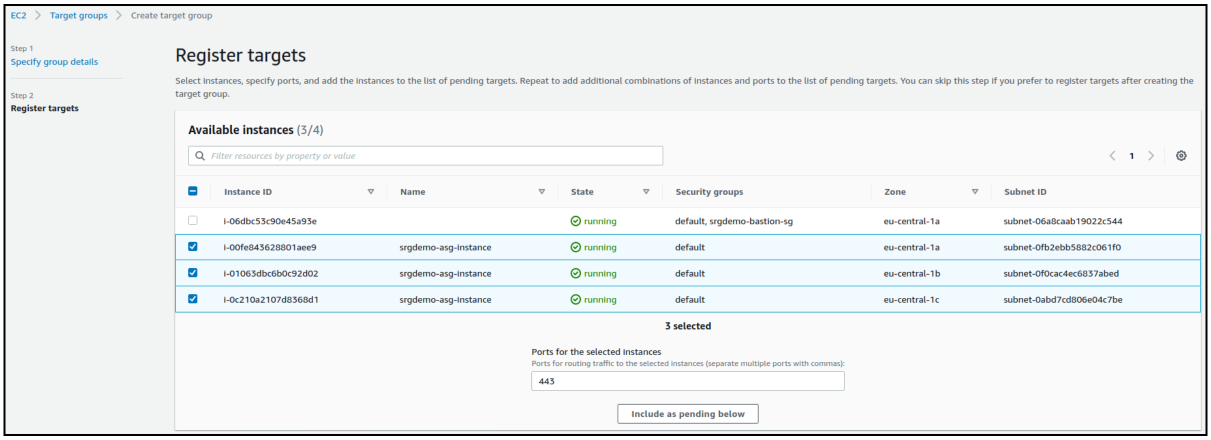
Health check protocol
HTTPS

Health check path
Use the default path of "/" to ping the root, or specify a custom path if preferred.
/
Up to 1024 characters allowed.

▶ Advanced health check settings

▶ Tags - *optional*
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

5. Click **Next**.
6. On the **Register Targets** page, set values for the following:
   a. **Available instances:** Select your instances; do not select the bastion.
   b. **Ports:** For the selected instances, use the value you retrieved previously and recorded on the AWS worksheet as the `Node Port for Port 443`.



EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Register targets

Select instances, specify ports, and add the instances to the list of pending targets. Repeat to add additional combinations of instances and ports to the list of pending targets. You can skip this step if you prefer to register targets after creating the target group.

Available instances (3/4)

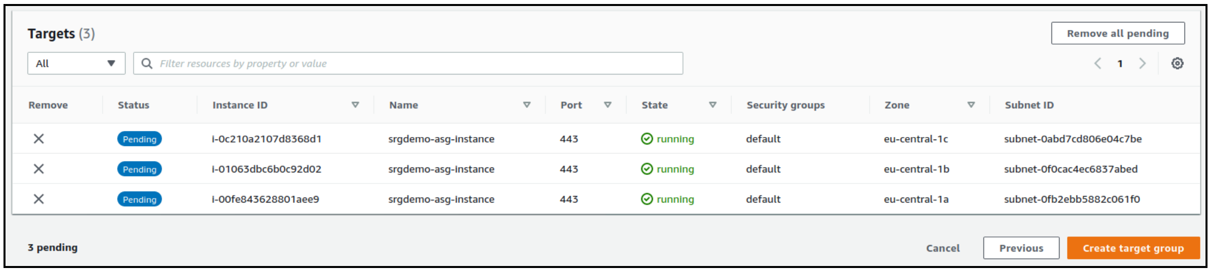
Filter resources by property or value

	Instance ID	Name	State	Security groups	Zone	Subnet ID
□	i-06dbc53c90e45a93e		✓ running	default, srgdemo-bastion-sg	eu-central-1a	subnet-06a8caab19022c544
☑	i-00fe843628801aee9	srgdemo-asg-instance	✓ running	default	eu-central-1a	subnet-0fb2ebb5882c061f0
☑	i-01063dbc6b0c92d02	srgdemo-asg-instance	✓ running	default	eu-central-1b	subnet-0f0cac4ec6837abed
☑	i-0c210a2107d8368d1	srgdemo-asg-instance	✓ running	default	eu-central-1c	subnet-0abd7cd806e04c7be

3 selected

Ports for the selected instances
Ports for routing traffic to the selected instances (separate multiple ports with commas):
443

7. Click **Include as pending below.** All marked instances will be added to the list of pending instances.



Targets (3)

Remove	Status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
✕	Pending	i-0c210a2107d8368d1	srgdemo-asg-instance	443	✓ running	default	eu-central-1c	subnet-0abd7cd806e04c7be
✕	Pending	i-01063dbc6b0c92d02	srgdemo-asg-instance	443	✓ running	default	eu-central-1b	subnet-0f0cac4ec6837abed
✕	Pending	i-00fe843628801aee9	srgdemo-asg-instance	443	✓ running	default	eu-central-1a	subnet-0fb2ebb5882c061f0

3 pending

8. Click **Create target group.**
9. You will be redirected back to the target group management page. From the list, select the newly created target group. From the bottom of the page, note its ARN in the AWS worksheet.

To create a target group for port 443 using the CLI:

1. Run the following command:

```
# aws elbv2 create-target-group \
--name <Target group 443 Name> \
--protocol HTTPS \
--port 443 --vpc-id <VPC ID> \
--health-check-protocol HTTPS \
--target-type instance
```

Parameters:

<Target group 443 Name>: Choose some descriptive name; such as srgdemo-443-tg and record the value in the AWS worksheet.

<VPC ID>: The ID of your VPC as recorded on your AWS worksheet.

Example input and output:

```
# aws elbv2 create-target-group \
   --name srgdemo-443-tg --protocol HTTPS \
   --port 443 --vpc-id vpc-0143197ca9bd9c117 \
   --health-check-protocol HTTPS \
   --target-type instance
```

```
{
      "TargetGroups":[
         {
            "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
   1:115370848038:targetgroup/srgdemo-443-tg/6d30f1c7be588bb6",
            "TargetGroupName":"srgdemo-443-tg",
            "Protocol":"HTTPS",
            "Port":443,
            "VpcId":"vpc-0143197ca9bd9c117",
            "HealthCheckProtocol":"HTTPS",
            "HealthCheckPort":"traffic-port",
            "HealthCheckEnabled":true,
            "HealthCheckIntervalSeconds":30,
            "HealthCheckTimeoutSeconds":5,
            "HealthyThresholdCount":5,
            "UnhealthyThresholdCount":2,
            "HealthCheckPath":"/",
            "Matcher":{
               "HttpCode":"200"
            },
            "TargetType":"instance"
         }
      ]
   }
```

2. From the output, record the value of TargetGroupArn in your AWS worksheet.

Tagging the Target Group (CLI)

Optionally, you can tag the target group for easier identification.

To tag the target group using the CLI:

1. Run the following command:
   ```
   # aws elbv2 add-tags \
   --resource-arns <Target group 443 ARN> \
   --tags Key=owner,Value=<owner>
   ```

Example:

```
# aws elbv2 add-tags \
--resource-arns arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-443-tg/c0684be94405b6b7 \
--tags Key=owner,Value=srgdemo
```

Next Step: Adding Targets to the Target Group for Port 443

Adding Targets to the Target Group

1. Run the following command:
   ```
   # aws elbv2 register-targets \
   --target-group-arn <Target group 443 ARN> \
   --targets Id="Instance 1 ID" \
   Id="Instance 2 ID" \
   Id="Instance 3 ID"
   ```

Parameters:

`<Instance x ID>`: Use the instance IDs you gathered for instances of the Auto Scaling group. Refer to AWS worksheet for values.

`<Target group 443 ARN>`: Use the ARN of the target group you just created.

`<Node port for 443>`: Use the node port number for 443 from the AWS worksheet.

Example:

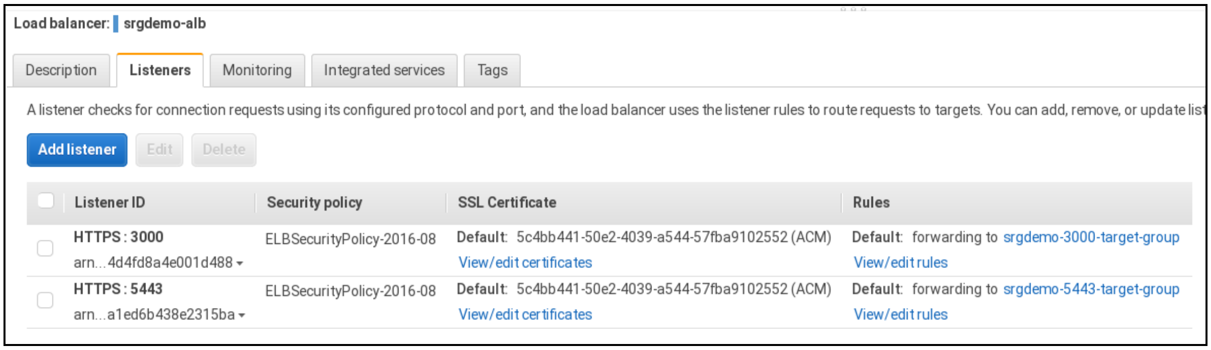
```
# aws elbv2 register-targets \
  --target-group-arn arn:aws:elasticloadbalancing:eu-central
1:115370848038:targetgroup/srgdemo-443-tg/6d30f1c7be588bb6 \
  --targets Id="i-05662f9ef84c182ca"\
  Id="i-07cfcd6716e9890b5" \
  Id="i-08d819b5ccabe83cb"
```

Adding a Listener for Port 443 to the ALB

This action will connect the ALB to port 443 through the target group. Then, incoming requests to the ALB on port 443 will be directed to the node instances.

To add a listener for port 443 using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Load Balancers.**
3. From the list of load balancers, select your previously-created Application Load Balancer (ALB).

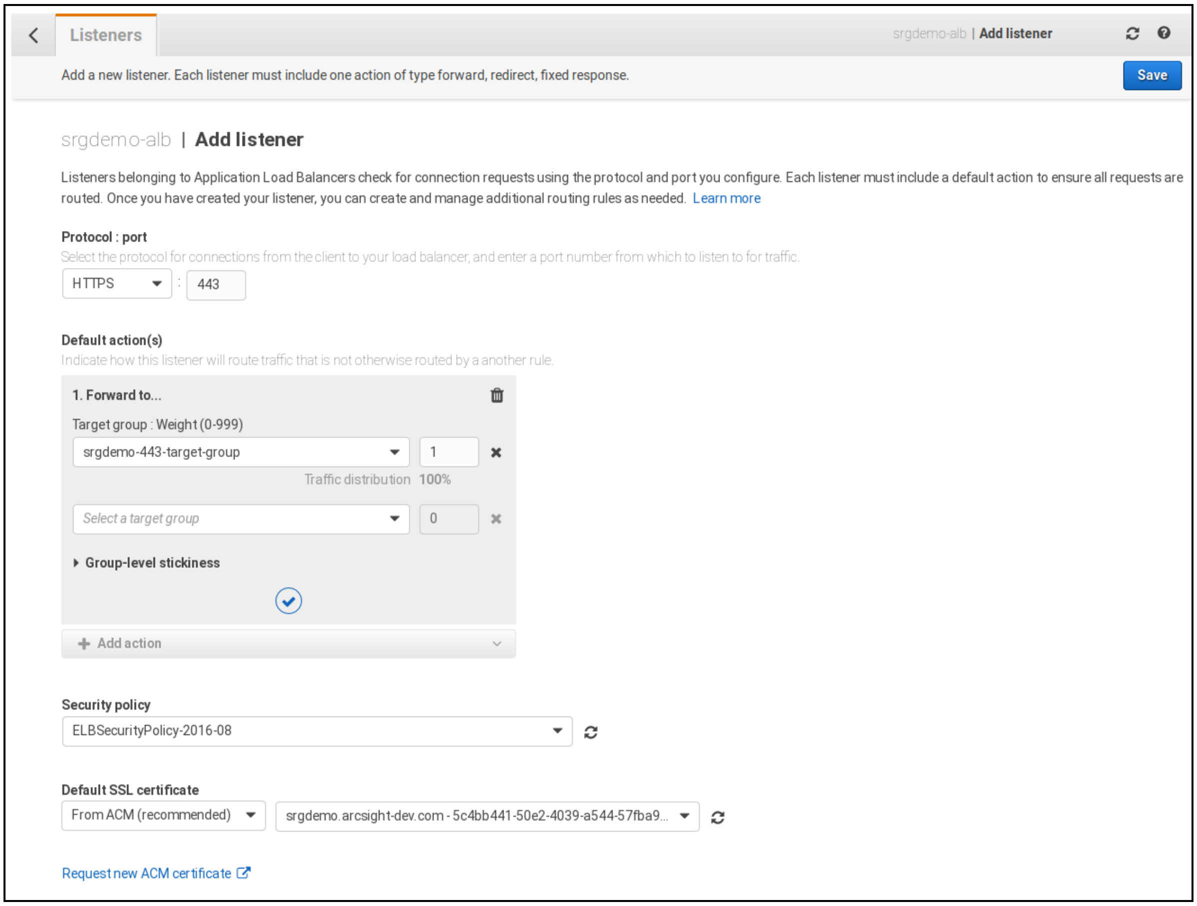


Load balancer: srgdemo-alb

A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener rules to route requests to targets. You can add, remove, or update list

Listener ID	Security policy	SSL Certificate	Rules
HTTPS : 3000 arn...4d4fd8a4e001d488	ELBSecurityPolicy-2016-08	Default: 5c4bb441-50e2-4039-a544-57fba9102552 (ACM) View/edit certificates	Default: forwarding to srgdemo-3000-target-group View/edit rules
HTTPS : 5443 arn...a1ed6b438e2315ba	ELBSecurityPolicy-2016-08	Default: 5c4bb441-50e2-4039-a544-57fba9102552 (ACM) View/edit certificates	Default: forwarding to srgdemo-5443-target-group View/edit rules

4. On the **Listeners** tab, click **Add Listener** and set values for the following:
   a. **Protocol: port:** Change to HTTPS and 443
   b. **Default action(s):** Choose the action Forward to... then choose your target group for port 443
   c. **Default SSL certificate:** Choose the SSL certificate you have created previously.



Add a new listener. Each listener must include one action of type forward, redirect, fixed response.

srgdemo-alb | **Add listener**

Listeners belonging to Application Load Balancers check for connection requests using the protocol and port you configure. Each listener must include a default action to ensure all requests are routed. Once you have created your listener, you can create and manage additional routing rules as needed. Learn more

Protocol : port

Select the protocol for connections from the client to your load balancer, and enter a port number from which to listen to for traffic.

HTTPS : 443

Default action(s)

Indicate how this listener will route traffic that is not otherwise routed by a another rule.

1. Forward to...

Target group : Weight (0-999)

srgdemo-443-target-group 1

Traffic distribution 100%

Select a target group 0

Group-level stickiness

Add action

Security policy

ELBSecurityPolicy-2016-08

Default SSL certificate

From ACM (recommended) srgdemo.arcsight-dev.com - 5c4bb441-50e2-4039-a544-57fba9...

Request new ACM certificate

5. Click **Save.**

To add a listener to port 443 using the CLI:

1. Run the following command:
```
# aws elbv2 create-listener \
--load-balancer-arn <ALB ARN> \
--protocol HTTPS --port 443 \
--certificates CertificateArn=<Certificate ARN> \
--default-actionsType=forward,TargetGroupArn=<Target group 443 ARN>
```

Parameters:

`<ALB ARN>`: use ALB ARN recorded in the AWS worksheet.

`<Certificate ARN>`: use certificate ARN recorded in the AWS worksheet.

`<Target group 443 ARN>`: use the target group for port 443 ARN recorded in the AWS worksheet.

Example input and output:

```
# aws elbv2 create-listener \
--load-balancer-arn arn:aws:elasticloadbalancing:eu-central-
```

```
1:115370848038:loadbalancer/app/srgdemo-alb/8718b24107ef591b \
--protocol HTTPS --port 443 --certificates CertificateArn=arn:aws:acm:eu-
central-1:115370848038:certificate/691ec232-98ff-45ed-8e69-1d15c0447538 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-443-tg/a096cb67c2f9144dv
```

```
{
   "Listeners":[
      {
         "ListenerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:listener/app/srgdemo-alb/8718b24107ef591b/66915d0da2adb8a9",
         "LoadBalancerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:loadbalancer/app/srgdemo-alb/8718b24107ef591b",
         "Port":443,
         "Protocol":"HTTPS",
         "Certificates":[
            {
               "CertificateArn":"arn:aws:acm:eu-central-
1:115370848038:certificate/691ec232-98ff-45ed-8e69-1d15c0447538"
            }
         ],
         "SslPolicy":"ELBSecurityPolicy-2016-08",
         "DefaultActions":[
            {
               "Type":"forward",
               "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370848038:targetgroup/srgdemo-443-tg/6d30f1c7be588bb6",
               "ForwardConfig":{
                  "TargetGroups":[
                     {
                        "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-
central-1:115370848038:targetgroup/srgdemo-443-tg/6d30f1c7be588bb6",
                        "Weight":1
                     }
                  ],
                  "TargetGroupStickinessConfig":{
                     "Enabled":false
                  }
               }
            }
         ]
      }
   ]
}
```

Next Step: You are now ready to deploy ArcSight Suite products using the CDF Management Portal. Proceed to Deploying Transformation Hub.

Next Steps

You are now ready to deploy ArcSight Suite products using the CDF Management Portal. Proceed to Deploying Transformation Hub.

AWS Configuration Worksheets

During the setup and configuration of your AWS deployment environment, use the following worksheets.

- "AWS Infrastructure Settings" below
- "Subnets" below
- "Security Groups" on the next page
- System and Environment Settings

AWS Infrastructure Settings

Region	
Vpc ID	
Vpc CIDR	
Vpc Name	
Cluster Name	
Public IP ID	
Public IP	
Internet GW ID	
NAT GW ID	
DNS Enabled (Y/N)	
Hostname Resolution Enabled (Y/N)	

Subnets

Availability Zone	*CIDR*	*Name*	*ID*	*Tagged Load Balancing *Y/N/NA*

Subnets

Availability Zone	*CIDR*	*Name*	*ID*	*Tagged Load Balancing *Y/N/NA*

Security Groups

Bastion Security Group Name	
Bastion Security Group ID	
Intra VPC Security Group Name	
Intra VPC Security Group ID	
IAM Roles	
EKS Role Name	
EKS Role ARN	
EKS Instance Profile Name	
EKS Instance Profile ARN	
Workernodes Role Name	
Workernodes Role ARN	
Workernodes Instance Profile Name	
Workernodes Instance Profile ARN	

System and Environment Settings

Bastion	Kubernetes Version	
	Key Pair Name	
	Key Pair Fingerprint	
	Image ID	

	Instance Type	
	Instance ID	
	Public IP Address	
EFS	EFS Name	
	FileSystemID	
	Filesystem FQDN	
	Mount Target 1 ID	
	Mount Target 2 ID	
	Mount Target 3 ID	
	Parent Folder Name	
EKS	Cluster ARN	
Worker Nodes	Launch Configuration Name	
	Launch Config AMI ID	
	Instance Tyoe	
	Autoscaling Group Name	
	Instance IDs	
ECR Registry Upload	Organization Name	
Route 53 Records	Name In Hosted Zone	
	Hosted Zone Name	
	Hosted Zone ID	
	RecordSet Name	
	Certificate ARN	
Networking	ALB Name	
	ALB ARN	
	ALB DNS Name	
	ALB Canonical Hosted Zone ID	
	Node Port for 3000	

	Target Group 3000 Name	
	Target Group 3000 ARN	
	Target Group 5443 Name	
	Target Group 5443 ARN	
	Target Group 443 Name	
	Target Group 443 ARN	

Deploying ArcSight Products

Tuning Your Deployment

This topic contains information on:

- "Updating Event Topic Partition Number" below
- "Updating the CDF Hard Eviction Policy" on the next page

Updating Event Topic Partition Number

Refer to *ArcSight Platform 20.11 Technical Requirements* "Hardware and Tuning Guidelines section" to determine an appropriate event topic partition number for your workload.

The following steps are needed only when deploying Recon or Intelligence.

To update the topic partition number from the master node1, run the following commands:

1. Find the server ($ZK), running th-zookeeper-0:

   ```
   ZK=`kubectl get pods --all-namespaces -o wide|grep zookeeper-0|awk '{print $8}'`
   ```

2. Find NAMESPACE ($NS), for th-kafka-0:

   ```
   NS=`kubectl get pods --all-namespaces|grep kafka-0|awk '{print $1}'`
   ```

3. Update th-arcsight-avro topic partition number:

   ```
   kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --zookeeper
   $ZK:32181 --alter --topic th-arcsight-avro --partitions $number
   ```

   > `$number` is the number used to calculate the partition size.

4. Update th-cef topic partition number:

   ```
   kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --zookeeper
   $ZK:32181 --alter --topic th-cef --partitions $number
   ```

5. Use the kafka manager to verify the partition number of th-cef topic and th-arcsight-avro topic have been updated to `$number.`

Updating the CDF Hard Eviction Policy

You need to update the Kubernetes hard eviction policy from 15% (default) to 100 GB to maximize disk usage.

To update the CDF Hard Eviction Policy, perform the following steps on each worker node, after deployment has been successfully completed.

> Please verify the operation is successfully executed on one work node first, then proceed on the next worker node.

> eviction-hard can either be defined as a percentage or a specific amount. The percentage or the specific amount will be determined by the volume storage.

To update the policy:

1. Run:

   ```
   cp /usr/lib/systemd/system/kubelet.service
   ```

   ```
   /usr/lib/systemd/system/kubelet.service.orig
   ```

   ```
   vim /usr/lib/systemd/system/kubelet.service
   ```

2. Behind the line:

   ```
   ExecStart=/usr/bin/kubelet \
   ```

3. Add line:

   ```
   --eviction-hard=memory.available<100Mi,nodefs.available<100Gi,imagefs.available<2Gi
   \
   ```

4. Run:

   ```
   systemctl daemon-reload and systemctl restart kubelet
   ```

5. To verify, run:

   ```
   systemctl status kubelet
   ```

   No error should be reported.

Configuring the Deployed Capabilities

> Refer to *ArcSight Platform 20.11 Technical Requirements* "Hardware and Tuning Guidelines section" for your workload. It might specify additional settings beyond what is described below.

The deployed capabilities are ready to be configured and then deployed. The *Pre-Deployment Configuration* page displays to configure the products and capabilities chosen at the start of the installation process. This section explains the process of configuring deployed capabilities on a supported platform for both on-premises and cloud deployments.

Reviewing Settings That Must Be Set During Deployment

This section describes configuration settings that must be set during deployment. Additional settings can be modified after deployment by going to the CDF Management Portal.

> For more information, hover over the tooltips and set the values accordingly.

- "ArcSight Database" on the next page
- "Transformation Hub" on the next page
- "Fusion" on page 291
- "Intelligence" on page 291

ArcSight Database

If you deployed the ArcSight database:

In the Transformation Hub tab, ensure the **# of CEF-to-Avro Stream Processor instances to start** is set to at least 1 or what is specified in *ArcSight Platform 20.11 Technical Requirements* for your workload.

In the Fusion tab, ensure you set these configuration settings for your environment:

- **Enable Database**
- **Use SSL for Database Connections**
- **Database Host**
- **Database Application Admin User Name**
- **Database Application Admin User Password**
- **Search User Name**
- **Search User Password**
- **Database Certificate(s)**

Transformation Hub

If you deployed the Transformation Hub:

In the Transformation Hub tab, ensure the following are set to the number of Kafka worker nodes designed into your deployment or what is specified in *ArcSight Platform 20.11 Technical Requirements* for your workload.

- **# of Kafka broker nodes in the Kafka cluster (th-kafka-count)**
- **# of ZooKeeper nodes in the ZooKeeper cluster (th-zookeeper-count)**
- **# of replicas assigned to each Kafka Topic (th-init-topicReplicationFactor)** (this setting must be set to 1 for a Single Worker deployment, and 2 for a 3-node environment)

In the Transformation Hub tab, configure the following security settings based on how you planned to secure communications as described in the Securing Communication Among Micro Focus Components section.

> FIPS and Client-Authentication are available during installation only.

- **Allow plain text (non-TLS) connections to Kafka (th-kafka-allow-plaintext)**
- **Connections use FIPS encryption (th-init-fips)**
- **Connection to Kafka uses TLS Client Authentication (th-init-client-auth)**

Fusion

If you deployed Fusion:

In the Fusion tab:

- If you have not deployed the database, set **Search Engine Replicas(search-engine-replica)** to 0, which disables the Search Engine so that it doesn't attempt to access a non-existent database. When the database is deployed, enable the Search Engine by setting this to 1.
- Modify the **Client ID** and **Client Secret** to a unique value for your environment.

Intelligence

If you deployed Intelligence:

In the Intelligence tab, ensure you set these configuration settings for your environment:

- **Intelligence System Admin Email ID (interset-root-user)**
- **Number of Database Nodes (interset-vertica-number-of-nodes)**

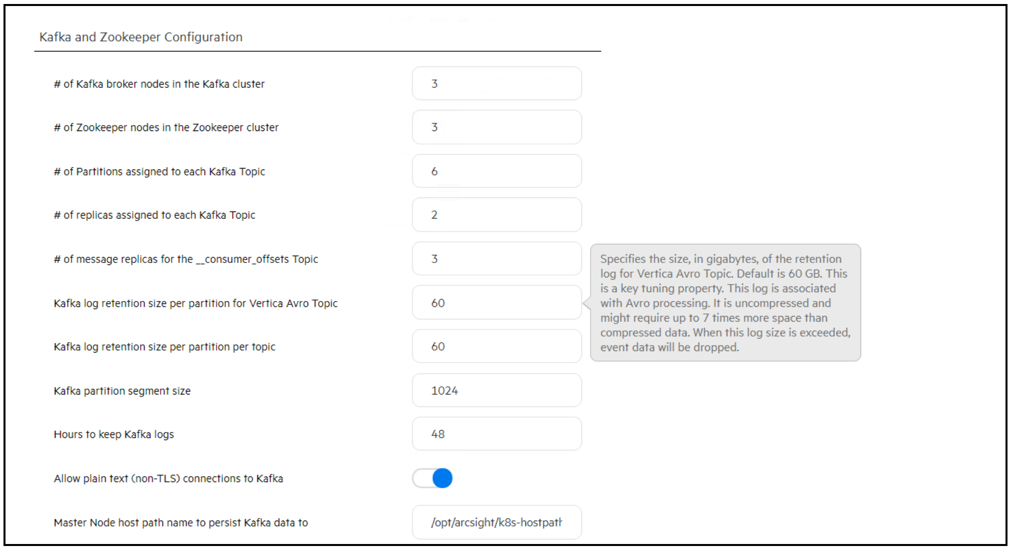
> Be sure to change the passwords to a unique value for your environment.

- **HDFS NameNode (interset-hdfs-namenode)**
- **H2 Password (interset-h2-password)**
- **Elasticsearch Password (interset-elasticsearch-password)**
- **Analytics KeyStore Password (interset-analytics-keystore-password)**
- **Investigator KeyStore Password (interset-api-keystore-password)**
- **SearchManager KeyStore Password (searchmanager-api-keystore-password)**
- **Logstash KeyStore Password (interset-logstash-keystore-password)**
- **H2 KeyStore Password (interset-h2-keystore-password)**

If the topic name specified for the Avro Event Topic field is not the default topic, then use Transformation Hub's Avro routing rules using ArcMC 2.96 or later to filter Avro events from the default topic. Create a routing rule with the source topic as th-arcsight-avro and destination topic as the topic name you have provided in the Avro Event Topic field. For more information, refer to the routing section in the ArcMC Administration Guide.

For Intelligence **System Admin Email ID**, if Fusion is already a part of the cluster, ensure you specify the email ID of an existing System Admin user in Security, Risk & Governance. If you are deploying Fusion now, specify in the **System Admin Email ID** setting the email ID you intent to use. This user will be the default System Admin user of Intelligence.

For **Analytics Configuration-Spark**, set the values based on the data load. For information about the values for Spark, see *ArcSight Platform 20.11 Technical Requirements* "Hardware and Tuning Guidelines section" for your workload.



Kafka and Zookeeper Configuration

# of Kafka broker nodes in the Kafka cluster	3
# of Zookeeper nodes in the Zookeeper cluster	3
# of Partitions assigned to each Kafka Topic	6
# of replicas assigned to each Kafka Topic	2
# of message replicas for the __consumer_offsets Topic	3
Kafka log retention size per partition for Vertica Avro Topic	60
Kafka log retention size per partition per topic	60
Kafka partition segment size	1024
Hours to keep Kafka logs	48
Allow plain text (non-TLS) connections to Kafka	●
Master Node host path name to persist Kafka data to	/opt/arcsight/k8s-hostpatl

Specifies the size, in gigabytes, of the retention log for Vertica Avro Topic. Default is 60 GB. This is a key tuning property. This log is associated with Avro processing. It is uncompressed and might require up to 7 times more space than compressed data. When this log size is exceeded, event data will be dropped.

Checking the Deployment Status

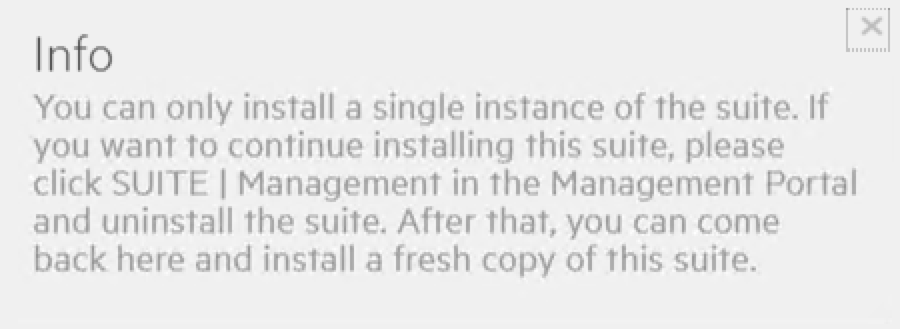
When the **Configuration Complete** page displays, the pod deployment is finished.

- Pods that have not been labeled will remain in the *Pending* state until labeled.
- For a pod that is not in the *Running* state, you can find out more details on the pod by running the following command:

```
    kubectl describe pod <pod name> -n <namespace>
```

The `Events` section in the output provides detailed information on the pod status.

If the following error is displayed when attempting to log in to the CDF Management Portal on port 3000, this typically means that the CDF installation process has completed, port 3000 is no longer required, and has been closed. Instead of port 3000, log in to the Management Portal on port 5443.



Info

You can only install a single instance of the suite. If you want to continue installing this suite, please click SUITE | Management in the Management Portal and uninstall the suite. After that, you can come back here and install a fresh copy of this suite.

Checking Cluster Status

To verify the success of the deployment, check the cluster status and make sure all pods are running.

You might need to wait 10 minutes or more for all pods to be in a *Running* or *Completed* state.

To check cluster status:

1. Connect to the cluster by doing one of the following:
   - For an on-premises installation, log in to the initial master node.
   - For Azure, connect to the jump host.
   - For AWS, connect to the bastion.
2. Run the command:

```
# kubectl get pods --all-namespaces
```

3. Review the output to determine the status of all pods.

Adding Capabilities to an Existing Cluster

You can deploy additional ArcSight capabilities to an existing ArcSight Platform Kubernetes cluster. Reusing an existing cluster reduces costs and system management effort compared to deploying these capabilities in a new cluster.

Understanding the Prerequisites and Considerations

> ✓ Be sure to review the capability-specific prerequisites listed here Configuring Elasticsearch Settings.

Before installing additional capabilities to an existing cluster, complete the following tasks.

- Ensure that your existing cluster has the supported version of the Platform required to deploy the additional capabilities. If your deployment does not have the supported version, you must upgrade the Platform using the instructions in Upgrading Your Environment. For information about the supported version of ArcSight Platform, see ArcSight Platform Technical Requirements.
- Recon and Intelligence both require the ArcSight Database.If you are adding these capabilities and your deployment does not already have the database, you will need to install the database using the instructions in this section.
- Check the system size of your existing ArcSight Platform Kubernetes cluster and, if applicable, ArcSight Database and ensure that it can handle the additional workload of the capabilities you want to add. If your existing cluster cannot handle the additional workload, scale the Kubernetes cluster or database as needed before deploying the additional capabilities. For information about system sizing of the Platform, see ArcSight Platform Technical Requirements.
- For Intelligence, configure SmartConnectors for data collection. For more information about data collection, see the SmartConnector User Guide and SmartConnector Configuration Guides.

Deploying Additional Capabilities to an Existing Cluster

> ✓ Be sure to review the capability-specific prerequisites listed here Configuring Elasticsearch Settings.

To deploy additional capabilities:

1. If you are adding Recon or Intelligence to your deployment, and you do not have a database deployed, see the Manually Install Database section and then continue.
2. Launch a terminal session and then log in to the master node as the `root` or as a `sudo` user.
3. Create a directory for additional capability image files to download in the next step. This directory must only contain the image files and nothing else.

   ```
   mkdir /tmp/download
   ```

4. Download the images for the capabilities to add. For more information about images, see Downloading ArcSight Platform Installation Files in ArcSight Platform Release Notes.
5. After download, validate the digital signature of each file. For a complete list of files and file versions to be downloaded, consult ArcSight Platform Release Notes.

   > Do not untar the files.

6. Change to the following directory.

   ```
   cd ${K8S_HOME}/scripts/
   ```

   For example:

   ```
   cd /opt/arcsight/kubernetes/scripts/
   ```

7. Run the following commands to upload the images to the local Docker Registry. Use the `-F <image file>` option on the command line multiple times for each image to upload. Adjust the `-c 2` option up to half of your CPU cores in order to increase the speed of the upload.

> You will be prompted for a password for the docker container registry-admin user. The registry-admin password is initially set to the same password as the admin user for the CDF Management Portal during installation when "Configuring and Running CDF" on page 70; however, later changing the CDF Management Portal admin password does not change the registry-admin password as it is managed separately.

```
./uploadimages.sh -c 2 -F /tmp/download/fusion-x.x.x.x.tar -F
/tmp/download/recon-x.x.x.x.tar
```

8. Log in to the CDF Management Portal with the following credentials:

   User name: admin

   Password: *<the password you provided during CDF installation>*

9. Click ⋮ and then click **Change**.
10. On the **Capabilities** page, select the additional capabilities to deploy.
11. Click the arrow next to the capability checkbox to view the description of the capabilities to deploy to determine if they require additional capabilities to be deployed. For example, ArcSight Recon requires Transformation Hub and Fusion.
12. Click **Next** until you reach the **Configure/Deploy** page.
13. See the Configuring the Deployed Capabilities section, and then return to this page to continue.
14. Click **Next**. On the **Configuration Complete** page, wait until the deployment is complete. The deployment process might take several minutes to complete.

    > Some of the pods in the **Configuration Complete** page might remain in a Pending state until the product labels are applied on worker nodes.

15. Continue with labeling the nodes.
16. If you deployed the database in the first step, see Completing the Database Setup section.
17. Continue to the Performing Post-deployment Configurations section.

Performing Post-deployment Configurations

This section provides information about the post-installation configurations you must perform.

Installing Your License Key

Transformation Hub, Intelligence, and Recon all require license keys. They contain a 90-day instant-on evaluation license, which will enable functionality for 90 days after installation. In order to continue working past the initial evaluation period, you will need to apply a valid license key. For more information about license keys, see the Understanding License Keys section.

> To ensure continuity of functionality and event flow, make sure you apply your product license **before** the evaluation license has expired.

To install your license:

1. Log in to the Management Portal (https://<ha-address>:5443).
2. Click **APPLICATION**.
3. Click **License**.

   > For more information about license management capabilities, see AutoPass License Management documentation.

4. Click **License > Install.**
5. Click **ADD FILE(S).**
6. Browse to the location of your license file.
7. Click **Next**.
8. Optionally, select the **I authorize Micro Focus to collect suite and product data...** checkbox to send usage data to Micro Focus to help improve the product.
9. Follow the prompts to apply your license.
10. Apply all of the licenses required for your deployed capabilities.
11. If you just installed a Transformation Hub license, restart each Kafka pod in the cluster, one at a time, as follows:

a. For each of the Kafka pod from 0 to x, restart the selected Kafka pod with the command:

```
# kubectl delete pod th-kafka-(x) -n arcsight-installer-XXX
```

b. Watch the logs and ensure the Kafka pod is up and running by running this command:

```
# kubectl logs th-kafka-(x) -n arcsight-installer-XXX
```

c. Once the selected broker node is up and running, only then proceed to restart the next node.

> You can also check the status of the restarted broker node using the Transformation Hub Kafka Manager.

Verifying the Transformation Hub License

To verify your license:

For each Kafka broker node, the license check result is logged both in the Kafka pod log and in the file:

```
/opt/arcsight/k8s-hostpath-volume/th/autopass/license.log.
```

If there is a valid license, the log includes the following text:

```
TH licensed capacity: <eps number>
```

If a license has not been installed, the following text displays instead:

```
ERROR: No valid license key was found. Please install a valid license key or contact
Micro Focus Customer Support for instructions on how to get one
```

Configuring the Database with HDFS for Intelligence

> Applies only when you deploy the Intelligence capability.

After deploying Intelligence, you must configure the database with HDFS for the database to receive the Intelligence Analytics results data from Spark through HDFS.

Prerequisites

For a manual deployment of Intelligence, ensure that you install firewall and open the firewall ports on the nodes before you proceed with configuring the database with HDFS:

1. Log in to a Kubernetes node labeled as intelligence-namenode:yes as a root user.
2. Execute the following commands to install and enable the firewall:

   ```
   yum -y install firewalld
   systemctl enable firewalld
   ```

3. Execute the following command to ensure NAT is configured:

   ```
   firewall-cmd --add-masquerade --permanent
   ```

4. (Conditional) Execute the following commands to open the firewall ports on the node labeled as intelligence-namenode:yes:

   ```
   firewall-cmd --permanent --add-port=30820/tcp
   firewall-cmd --permanent --add-port=30070/tcp
   ```

5. (Conditional) Execute the following commands to open the firewall ports on the node labeled as intelligence-datanode:yes.

   ```
   firewall-cmd --permanent --add-port=30210/tcp
   firewall-cmd --permanent --add-port=30010/tcp
   ```

6. Execute the following commands to avoid a firewall restart and to ensure that the Kubernetes services do not stop running on the node:

   ```
   iptables -I IN_public_allow -j ACCEPT -p tcp --dport 30820 -m conntrack -
   -ctstate NEW,UNTRACKED
   iptables -I IN_public_allow -j ACCEPT -p tcp --dport 30210 -m conntrack -
   -ctstate NEW,UNTRACKED
   iptables -I IN_public_allow -j ACCEPT -p tcp --dport 30070 -m conntrack -
   -ctstate NEW,UNTRACKED
   iptables -I IN_public_allow -j ACCEPT -p tcp --dport 30010 -m conntrack -
   -ctstate NEW,UNTRACKED
   ```

7. Repeat steps 1 to 5 on all nodes labeled as intelligence-datanode:yes.

To configure the database with HDFS:

1. Launch a terminal session and log in to a Kubernetes node as a `root` user.
2. Execute the following command to retrieve the namespace:

   ```
   export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
   ```

3. Execute the following command to retrieve the RPC port and the Web port:

   ```
   kubectl -n $NS get svc |grep hdfs-namenode
   ```

   An example of the output is:

   ```
   hdfs-namenode-svc ClusterIP None <none> 30820/TCP,30070/TCP 4h32m
   ```

   The first TCP port number (30820) is of the RPC port and the second TCP port number (30070) is of the Web port.

4. Log in to a database node as a root user.

5. Create the **/etc/hadoop/conf/** directory if it does not already exist.

6. Create the **core-site.xml** file if it does not already exist, then update the **fs.defaultFS** and **dfs.-namenode.http-address** properties along with the ports you retrieved in Step 3. Ensure that the `NAMENODE_HOST` value matches the hostname or IP address you provided in the **HDFS NameNode** field in the **CDF Management Portal** > **Configure/Deploy** > Intelligence.

   ```
   cat /etc/hadoop/conf/core-site.xml
   <configuration>
   <property>
   <name>fs.defaultFS</name>
   <value>hdfs://<NAMENODE_HOST>:<NAMENODE_RPC_PORT>/</value>
   </property>
   <property>
   <name>dfs.namenode.http-address</name>
   <value><NAMENODE_HOST>:<NAMENODE_WEB_PORT></value>
   </property>
   </configuration>
   ```

   For example:

   ```
   cat /etc/hadoop/conf/core-site.xml
   <configuration>
   <property>
   <name>fs.defaultFS</name>
   <value>hdfs://vlab012345.interset:30820/</value>
   </property>
   <property>
   <name>dfs.namenode.http-address</name>
   <value>vlab12345.interset:30070</value>
   </property>
   </configuration>
   ```

7. Create the **hdfs-site.xml** file as follows if it does not already exist:

```
<configuration>
</configuration>
```

8. Repeat steps 4 to 7 on all database nodes.
9. Verify whether the database and HDFS configuration is successful:
   a. Change to the following directory:

   ```
   cd /opt/vertica/bin/
   ```

   b. Log in as a dbadmin:

   ```
   su dbadmin
   ```

   c. Log in to vsql and specify the password when prompted:

   ```
   vsql
   [password prompt]
   ```

   d. (Optional) Clear the cache after configuring the database with HDFS:

   ```
   SELECT CLEAR_HDFS_CACHES();
   ```

   e. Execute the following commands:

   ```
   SELECT VERIFY_HADOOP_CONF_DIR();
   ```

   ```
   SELECT node_name, node_address, export_address FROM nodes;
   ```

   The expected output is:

   ```
   Welcome to vsql, the Vertica Analytic Database interactive terminal.

   Type: \h or \? for help with vsql commands
   \g or terminate with semicolon to execute query
   \q to quit

   dbadmin=> SELECT VERIFY_HADOOP_CONF_DIR();
   VERIFY_HADOOP_CONF_DIR
   --------------------------------------------------------------------
   -----
   Validation Success
   v_investigate_node0001: HadoopConfDir [/etc/hadoop/conf] is valid

   (1 row)
   ```

```
dbadmin=> SELECT node_name, node_address, export_address FROM nodes;
node_name | node_address | export_address
----------------------------------------------------------
v_investigate_node0001 | <IP1> | <IP1>
v_investigate_node0002 | <IP2> | <IP2>
v_investigate_node0003 | <IP3> | <IP3>
(3 rows)
```

Installing the Dashboard Widget for SOAR

> This only applies when ArcSight SOAR capability is deployed.

You must install the widget to display the SOAR data in the dashboard.

1. Download the `soar-widgets-n.n.n.n.tar` file from Micro Focus Downloads as an installation package.
2. Extract the contents of the tar file at the following location:

   ```
   /opt/arcsight-nfs/arcsight-volume/fusion/widget-store
   ```

Creating the First System Admin User

> This procedure applies only when you deploy a capability that requires Fusion.

To create the first user in the System Admin role:

1. Open a certified web browser.
2. Specify the following URL to log in to the application:

   ```
   https://<cdf_masternode_hostname or virtual_ip hostname>/mgmt
   ```

3. Specify the required information to create a System Admin user. (Important: It is strongly recommended that you use a valid email address for the user, so that it can be used to recover access to the account if the password is forgotten. There is no practical way to recover the account when the password is forgotten if the email address is not valid.)

   > If Intelligence is deployed, for the **Email** field, add the Intelligence System Admin Email ID you specified during the installation in the **Configure/Deploy** page > Intelligence > Intelligence **System Admin Email ID** field, or the email ID specified in the config YAML file, which was used for installation.

4. After the account is created, you will be prompted to log in with the credentials you just created.
5. (Optional) Log in to the application with the Email ID and password you just created.

Checklist: Performing Regular Maintenance

Use the following checklist to perform regular maintenance of the Platform infrastructure.

Frequency	Task	See
Every 1-3 Days	Check Health and Performance Dashboard for status or errors.	Using the Health and Performance Monitoring Dashboard
Every 1-3 Days	Check Kubernetes Dashboard for status and errors.	"Checking Kubernetes Dashboard for Status and Errors" on page 424
Every 1-3 Days	Check CDF Doctor for status and errors.	Using the CDF Doctor Utility
Every 90 Days	Reset expiring CDF Management Portal admin account password. he registry-admin password used when uploading capability images during system upgrade is initially set to the same password as the admin user for the CDF Management Portal during installation when Configuring and Installing CDF; however, later changing the CDF Management Portal admin password does not change the registry-admin password as it is managed separately. The registry-admin password does not automatically expire.	Resetting the CDF Administrator Password
Every 11 Months	Renew expiring CDF certificates (default expiration is 1 year).	"Maintaining Certificates" on page 425

Integrating the Platform Into Your Environment

Transformation Hub integrates with ArcSight SmartConnectors and Collectors, Logger, ESM, and ArcSight Recon. It is managed and monitored by ArcSight Management Center.

After you install and configure Transformation Hub you can use SmartConnectors and Collectors to produce and publish data to the Transformation Hub, and to subscribe to and consume that data with Logger, ESM, ArcSight Recon, Apache Hadoop, or your own custom consumer.

> Currently, cloud (Azure and AWS) clusters only support other ArcSight products which are in the Azure or AWS cloud. Integration with on-premises products is not supported for cloud-based Transformation Hub.

Transformation Hub supports both Common Event Format (CEF) versions, 0.1 and 1.0, as well as Avro and binary data formats. Transformation Hub third-party integration and other product features are explained in detail in the Transformation Hub Administrator's Guide.

Connecting to Your SMTP Server

> Applies only when you deploy a capability that requires Fusion.

To ensure that ESM for Fusion users receive email notifications, configure the connection to your SMTP server. For example, if you do not use SAML authentication, users will need notifications to help reset their forgotten passwords.

1. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.
2. Click **DEPLOYMENT**, and select **Deployments.**
3. Click the **Three Dots** ⋮ (Browse) on the far right and choose **Reconfigure**, under **FUSION > User Management Configuration**.
4. Input the following information, and click **SAVE:**
   - SMTP TLS Enabled
   - Fully qualified SMTP host name or IP Address

- SMTP port number
- SMTP USER name
- SMTP USER password
- Sender Mail Address or From Mail Address
- User session timeout in seconds

Configuring an External Identity Provider

Password-based authentication requires users to enter their User ID and Password when logging in.

You can select the built-in authentication or external authentication, such a SAML and LDAP.

Configuring LDAP Authentication

The identity provider (IDP) user and password has governance over the platform; therefore, the user must exist in both systems, but the password is validated only in LDAP.

1. Create at least one LDAP user to log in into the platform using LDAP authentication.
2. Log in to the CDF server and navigate to the SSO default configuration folder at:

   ```
   <arcsight_nfs_vol_path>/sso/default
   ```

where `<arcsight_nfs_vol_path>` is the NFS volume used for CDF installation; for example: `/opt/NFS_volume/arcsight-volume.`

3. Open the SSO configuration file (sso-configuration.properties), and review the LDAP parameters.

   ```
   ##### The following LDAP confgs are not utilized at this time
   # com.microfocus.sso.default.ldap.enabled = true
   # com.microfocus.sso.default.login.method = np-ldap
   # com.microfocus.sso.default.ldap.admin-dn = administrator@ospad.test
   # com.microfocus.sso.default.ldap.admin-pwd = password
   # com.microfocus.sso.default.ldap.host = 164.99.17.87
   # com.microfocus.sso.default.ldap.use-tls = false
   # com.microfocus.sso.default.ldap.port = 389
   #---- uncomment these if the LDAP server is Active Directory rather than eDirectory
   # com.microfocus.sso.default.ldap.dir-type = AD
   # com.microfocus.sso.default.as.naming-attr = sAMAccountName
   # com.microfocus.sso.default.as.users-container-dn = cn=Users,dc=ospad,dc=test
   ## uncomment these to configure URL when LDAP user forgets password
   # com.microfocus.sso.default.ldap.forgotten-pwd-url =
   ```

```
# com.microfocus.sso.default.ldap.login.forgotten-password-target = _blank
# com.microfocus.sso.default.ldap.login.forgotten-password-text-res-id =
# com.microfocus.sso.default.ldap.login.forgotten-password-title-res-id =
```

4. Update the SSO configuration file (sso-configuration.properties) for your LDAP log in method by commenting out these lines in the sso-configuration.properties file, and completing the information for your LDAP environment.

   ```
   com.microfocus.sso.default.ldap.enabled = true
   com.microfocus.sso.default.login.method = np-ldap
   com.microfocus.sso.default.ldap.admin-dn = provide your LDAP User DN here
   com.microfocus.sso.default.ldap.admin-pwd = provide your LDAP Admin password here
   com.microfocus.sso.default.ldap.host = provide your LDAP host here
   com.microfocus.sso.default.ldap.use-tls = provide your LDAP TLS setting here
   (true/false)
   com.microfocus.sso.default.ldap.port = provide your LDAP port here
   ```

5. For Active Directory rather than eDirectory:

   a. Update the SSO configuration file (sso-configuration.properties) to enable AD by uncommenting these lines in the sso-configuration.properties file, and completing the information for your LDAP environment.

      ```
      com.microfocus.sso.default.ldap.dir-type = AD
      com.microfocus.sso.default.as.naming-attr = provide your AD attribute here
      com.microfocus.sso.default.as.users-container-dn = provide your LDAP Base DN
      here
      ```

   b. Save the SSO configuration file (sso-configuration.properties).

6. For URL configuration when an LDAP user forgets the password:

   a. Update the SSO configuration file (sso-configuration.properties) to enable *forgot password* for the LDAP user by uncommenting these lines in the sso-configuration.properties file, and completing the information for your LDAP environment.

      ```
      com.microfocus.sso.default.ldap.forgotten-pwd-url = provide your LDAP url for
      forgotten password here
      com.microfocus.sso.default.ldap.login.forgotten-password-target = provide the
      target here
      com.microfocus.sso.default.ldap.login.forgotten-password-text-res-id = provide
      the text to be shown here
      com.microfocus.sso.default.ldap.login.forgotten-password-title-res-id = provide
      the title to be shown here
      ```

   b. Save the SSO configuration file (sso-configuration.properties).

7. Restart the fusion-single-sign-on pod.
   a. Get the fusion-single-sign-on pod information:

      ```
      kubectl get pods --all-namespaces | grep single-sign
      ```

   b. Restart the fusion-single-sign-on by deleting the currently running pod:

      ```
      kubectl delete pod fusion-single-sign-on-xxxxxxxxxx-xxxxx -n arcsight-installer-
      xxxxx
      ```

8. Log in using your LDAP credentials.

Configuring SAML Authentication

This section provides the steps to integrate SSO with an external SAML 2.0 IDP software, such as NetIQ Advanced Authentication.

> Fusion SSO and external SAML 2.0 IDP should be time-synchronized to the same NTP server. In the configuration UI, the session timeout must be set up with the same value that the external IDP has configured for user session timeouts.

- "Describing Information Regarding the Trusted Provider Metadata" below
- "Integrating with an External SAML Provider" on the next page

Describing Information Regarding the Trusted Provider Metadata

The metadata document for a trusted SAML provider with which a SSO defined provider interacts must be obtained in a provider-specific manner. While not all providers do so, many supply their metadata documents via URL.

After the trusted provider's metadata document (or the URL-accessible location of the document) is obtained, you must configure the SSO provider that will interact with the trusted provider's metadata.

In the document, modify the `<Metadata>` element within the `<AccessSettings>` element under either the `<TrustedIDP>` element or the `<TrustedSP>` element.

For example:

```
com.microfocus.sso.default.login.saml2.mapping-attr = email
```

The email attribute refers to the email attribute name from the SAML2 IDP.

Integrating with an External SAML Provider

To integrate with an external SAML provider:

1. On the NFS server, open the sso-configuration.properties file, located by default in the `<arcsight_nfs_vol_path>/sso/default` directory.

   `<arcsight_nfs_vol_path>` is the nfs volume used for CDF installation, for example: /opt/NFS_volume/arcsight-volume.

2. In the configuration directory, open the *sso-configuration.properties* file and add the following properties:
   - `com.microfocus.sso.default.login.method = saml2`
   - `com.microfocus.sso.default.saml2.enabled = true`

3. To specify the address where the IDP supplies its metadata document, complete one of the following actions:
   - Add the following property to the file: `com.microfocus.sso.default.login.saml2.metadata-url = <IDP SAML metadata URL>`
   - An example of a Keycloak server URL could be: https://<KeycloakServer>/auth/realms/<YourRealm>/protocol/saml/descriptor.

   > The IDP certificates need to be imported to the Fusion SSO keystore for HTTPS to work properly. See Step 5 for more details.

   - Alternatively, you can convert the metadata xml file to base64 string and set the following variable: `com.microfocus.sso.default.login.saml2.metadata = <base64 encoded metadata xml>`

4. Save the changes to the `sso-configuration.properties` file.

5. (Conditional) If you specified the metadata URL in Step 3, complete the following steps to import the IDP certificate to the SSO keystore:

   a. Copy the IDP certificate to the following location.

   ```
   arcsight_nfs_vol_path
   ```

   b. Get the pod information.

   ```
   kubectl get pods --all-namespaces | grep single-sign-on
   ```

   c. Open a terminal in the currently running pod:

   ```
   kubectl exec -it fusion-single-sign-on-xxxxxxxxxx-xxxxx -n arcsight-installer-
   xxxxx -c fusion-single-sign-on -- bash
   ```

d. Import the IDP certificate:

   i.
   ```
   cd /usr/local/tomcat/conf/default/
   ```

   ii.
   ```
   keytool -importcert -storepass $KEYSTORE_PASSWORD -destkeystore \
   sso.bcfks -alias AliasName -file CertificateFileName -storetype \
   BCFKS -providerclass \
   org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider \
   -providerpath /usr/local/openjdk-8/jre/lib/ext/bc-fips-1.0.2.jar
   ```

   - `CertificateFileName` represents the name of the certificate file that you want to import.
   - `AliasName` represents the new alias name that you want to assign to the certificate in the SSO keystore.

6. Restart the pod:

   a. Get the pod information.

   ```
   kubectl get pods --all-namespaces | grep fusion-single-sign-on
   ```

   b. Delete the current running pod.

   ```
   kubectl delete pod fusion-single-sign-on-xxxxxxxxxx-xxxxx -n arcsight-installer-
   xxxxx
   ```

7. Retrieve the Fusion SSO SAML service provider metadata from the server.

   ```
   https://EXTERNAL_ACCESS_HOST/osp/a/default/auth/saml2/spmetadata
   EXTERNAL_ACCESS_HOST is the hostname  of the server.
   ```

8. Use the SSO SAML service provider metadata to configure your IDP. For detailed instructions, see the IDP software documentation.

9. To establish a trust relationship between Fusion SSO and your IDP software, create certificates for your IDP software. For detailed instructions on how to create and import certificates in your IDP software, see the IDP software documentation.

Integrating Data and Users

The Fusion capability allows you to integrate users and data from ESM. With single sign-on (SSO) supported between Fusion and ESM, users can easily access the ArcSight Console, ArcSight Command Center, ESM Command Center, and REST APIs with the same login.

Understanding How ESM Users Access Fusion

Rather than manually adding users to Fusion, we recommend you create users in ESM first, and then import them into Fusion.

For the imported ESM users to log in to Fusion and be able to access ESM data, the following conditions apply:

- You must enable SSO access for ESM and Fusion users.
- Users must have an account in both ESM and Fusion.
- You must configure the External User ID and E-mail fields in the ESM accounts to comply with the *name@domain.com* format.
- Users must log in to Fusion with the **External User ID** from their ESM account.
- If your environment does not use SAML or LDAP authentication, ensure you have configured the SMTP server settings for Fusion. Users imported from ESM might need to set a password the first time they log in, which requires those users to initiate the Forgot Password function and receive an email notification.

Importing Users

You can import users that are already authorized. You need to have at least one role available in Fusion to assign to these users.

> Importing ESM users puts them ALL into the preselected fusion roles. You cannot downselect ESM users once you proceed. Only users with a filled e-mail address in ESM get imported.

1. In the ArcSight Console, ensure that the External User ID and E-mail fields for each account comply with the following format.

   ```
   name@domain.com
   ```

2. To log in to Fusion, use the following format.

   ```
   https://<cdf_masternode_hostname>
   ```

3. Click **ADMIN** > **Account Groups** > **Import Users**.
4. Select the role that you want to assign to the imported users.
5. Select **IMPORT USERS**.

As you add more users to ESM, you can run the import process again. Fusion ignores duplicates of user accounts that have been imported previously.

Enabling SSO With ESM

You must configure ESM to use **OSP Client Only Authentication**. If your ESM environment currently uses SAML or LDAP client authentication, you must delegate the Fusion SSO provider to connect to the SAML or LDAP client.

If you do not use SAML or LDAP authentication, you will need to "Connecting to Your SMTP Server" on page 304 to support forgotten password activity.

This procedure assumes you have ESM installed or upgraded.

1. Change the authentication settings for the ESM Manager service:

   a. On the ESM server, start the configuration wizard by entering the following.
   Command

   ```
   /opt/arcsight/manager/bin/
   ```

   Directory

   ```
   arcsight managersetup -i console
   ```

   b. Advance through the wizard until you reach the authentication settings.

   c. Select **OSP Client Only Authentication**, then click **Next**.

   d. To specify the host and port for the OSP server, use the following format:

   ```
   domain_name:port
   ```

   For example, Fusion by default installs OSP on port 443. So, when you are using Fusion, specify the format as:

   ```
    <fusion host>:443
   ```

   e. To specify the host and port for the ArcSight Command Center, use the following format:

   ```
   domain_name:port
   ```

   For example:

   ```
   <ESM Manager>:8443
   ```

   Typically, the host and port are the same as those for the ArcSight Manager.

   f. Specify a **Tenant Name for OSP**. If you are using a typical installer for Fusion, enter `default`.

   g. Click **Next** until you complete your changes in the wizard.

h. Restart the ESM Manager service using the following commands:

```
/etc/init.d/arcsight_services stop manager
```

```
/etc/init.d/arcsight_services start manager
```

2. Change the authentication settings for the ArcSight Console (the Console):

a. From the Console’s `/bin` directory, enter one of the following commands:
Windows

```
arcsight.bat consolesetup
```

Linux

```
 ./arcsight consolesetup
```

b. Advance through the wizard until you reach the authentication settings.

c. Select **OSP Client Only Authentication**.

d. Click **Next** until you complete your changes in the wizard.

3. To configure the SSO settings in the CDF Management Portal, complete the following steps:

a. Connect to the portal:

```
https://ESM_for_Fusion_server:5443
```

b. Log in with the credentials of the administrative user that you provided during installation.

c. Select **FUSION**.

d. Under **Single Sign-on Configuration**, specify the **Client ID** and **Client Secret**.

e. Under **ArcSight ESM Host Configuration**, verify the settings for the ESM host and port that were specified during deployment.

Using SSO with LDAP

To obtain a certificate from the Fusion server:

1. Run this command replacing <> with the correct information for your system:

```
echo | openssl s_client -showcerts -servername <fusion-fqdn.arcsight.com> -connect
<fusion-fqdn.arcsight.com>
```

2. Copy the certificate information from the begin to end (including dashes) cert.
    a. In a text editor, create a text file and paste the information in.
    b. Save the file.
    c. Copy the file to a host running ESM.
3. Ensure JAVA_HOME on the ESM console is pointing to the ArcSight JRE location before performing the certificate import.

Integrating Data from ESM

To view ESM data in the Dashboard, update the settings in the CDF Management Portal. The Fusion capability manages the Dashboard functions.

1. Open a new tab in a supported web browser.
2. Specify the URL for the CDF Management Portal:

    ```
    https://<cdf_masternode_hostname>:5443
    ```

    Use the fully qualified domain name of the host that you specified in the Connection step during the CDF configuration. Usually, this is the master node’s FQDN.
3. Log in to the CDF Management Portal with the credentials of the administrative user that you provided during installation.
4. Select **Reconfigure**.
5. On the **Configuration** page, select **FUSION**.
6. In the **ArcSight ESM Host Configuration** section, complete the following steps:
    a. For **ESM host**, specify the fully-qualified host name or IP address of the server that hosts ESM.
    b. For **ESM port**, specify the port associated with the **ESM host**. The default value is 8443.

Signing the External Communication Certificate with Your Certificate Authority

At the center of the Platform is a Kubernetes cluster where communication occurs between pods within the cluster and with non-containerized ArcSight components outside of the cluster. In order to ensure secure trusted communication between pods within the cluster and components outside of the cluster, encrypted communication with client certificate authentication is configured by default.

Signing a CSR for intermediate certificate is an alternative to providing intermediate key and intermediate certificate to the cluster. See "Maintaining Certificates" on page 425 for more information.

During installation, three self-signed Certificate Authorities (CA) are created automatically, two for signing certificates used exclusively for pod-to-pod communication within the cluster (RIC and RID CA) and the other for signing certificates for each pod that performs communication external to the cluster (RE CA). Only pods that perform external communication have a certificate that is signed by the external CA.

External cluster communication occurs not only with ArcSight components, but also with user web browsers and, in some cases, user clients of ArcSight APIs (such as REST API). By default, when the user connects to the cluster, they will be presented with a certificate that has been signed by the self-signed external CA. Since the external CA is self-signed, the user's connection will not automatically trust the certificate because it will not be verifiable using a certificate chain that is already in the user's trust store.

To give users confidence they are connecting to the trusted cluster, we recommend signing the certificates that are presented to the user with a CA that is trusted by the user's trust store. The only pods that perform external cluster communication with the user are the following (see "Understanding Labels and Pods" on page 584):

- nginx-ingress-controller - All user connections to ArcSight capability web UIs and APIs (default port 443) use this pod. This pod accepts the user's connection and then relays requests to other pods within the cluster using cluster internal communication, which utilizes the internal CA.
- CDF Management Portal - This is an administrator only portal (default port 5443) in the Platform used to manage the Kubernetes cluster and the capabilities running within it.

Signing the External Communication Certificate with Your Trusted Certificate Authority

In order to sign the external communication certificate with your trusted CA, you need to create a certificate signing request (CSR) from vault, take it to your organization, sign it, and return back signed CSR plus all the public chain of certificates used to sign it.

If you do not want to sign the CSR, as described below, you can provide intermediate certificate as described in Renewing External CA, which replaces steps 2-5.

1. Export the following access token dependencies, which you can remove later if not needed.

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json 2>/dev/null
| jq -r '.data.passphrase')
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core -o json
2>/dev/null | jq -r '.data."root.token"')
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -md sha256
-a -d -pass pass:"${PASSPHRASE}")
```

2. Ask vault to generate the CSR.

> <FQDN> is the FQDN of virtual IP address (also known as the External Access Hostname) for a multi-master type of installation or for a single master/single box use the FQDN of master/node.

```
/opt/arcsight/kubernetes/bin/vault write -tls-skip-verify -format=json \
-address=https://<FQDN>:8200 RE/intermediate/generate/internal  \
common_name="none-MF CDF RE CA on <FQDN>" \
| jq -r '.data.csr' > /tmp/pki_intermediate.csr
```

3. Use the CSR file to sign it with your certificate authority, and save the result into the intermediate.cert.pem.
For a basic example with openssl:

```
openssl ca -keyfile  your-rootca-sha256.key -cert your-rootca-sha256.crt
-config  your-openssl-configuration file -extensions v3_ca -notext -md
sha256 -in /tmp/pki_intermediate.csr -out intermediate.cert.pem
```

4. Import the file to vault.

```
/opt/arcsight/kubernetes/bin/vault write -tls-skip-verify -format=json \
RE/intermediate/set-signed certificate=@intermediate.cert.pem
```

5. Locate the RE_ca.crt (part of configuration map).
   a. Replace the certificate content in BEGIN and END of CERTIFICATE headers with the content of new certificate you imported to vault.
   b. Save your changes, and exit.
   c. See the issue commands below.

```
kubectl edit configmap -n core public-ca-certificates
kubectl edit configmap -n arcsight-installer-xxxx public-ca-certificates
```

6. Run the following command, which displays the output including the "private_key" and "certificate" data section.

> If you need to set additional SANs (Subject Alternative Names) in the generated certificate, use following syntax: common_ name=YOUR_ EXTERNAL_ ACCESS_ HOST alt_ names=arcsight.com,pkiadmin@arcsight.com. The list of parameters accepted while requesting certificate to be generated can be found in Vault documentation - PKI engine - Generate Certificate - Parameters.

```
/opt/arcsight/kubernetes/bin/vault write -tls-skip-verify -format=json \
RE/issue/coretech common_name=YOUR_EXTERNAL_ACCESS_HOST
```

a. For creating the nginx.KEY, locate the "private_key" data section and copy all the contents from `-----BEGIN RSA PRIVATE KEY-----` until the end of `-----END RSA PRIVATE KEY-----` into the nginx.KEY file.

b. For creating the nginx.CRT, locate the "certificate" data section and copy all the contents from `-----BEGIN CERTIFICATE-----` until the end of `-----END CERTIFICATE-----` into the nginx.CRT file.

c. Lastly, run the following command to replace all "\n" with CR:
sed -i 's/\\n/<enter>/g' nginx.KEY ;
sed -i 's/\\n/<enter>/g' nginx,.CRT

7. Use the command below to apply a new nginx key and certificates.

> The "dry-run" flag here is for creating yaml definition but not submitting it to kubernetes for creation - this part is done after pipe with apply command and namespace clarification

```
kubectl create secret generic "nginx-investigate-secret" --from-
file=tls.crt=./nginx.CRT \
--from-file=tls.key=./nginx.KEY --dry-run -o yaml \
| kubectl --namespace="arcsight-installer-xxxxx" apply -f -
```

```
kubectl create secret generic "nginx-default-secret" --from-
file=tls.crt=./nginx.CRT \
--from-file=tls.key=./nginx.KEY --dry-run -o yaml \
| kubectl --namespace="core" apply -f -
```

8. Get the pod information.

```
kubectl get pods --all-namespaces | grep single-sign-on
```

9. Open a terminal in the currently running pod.

```
kubectl exec -it fusion-single-sign-on-xxxxxxxxxx-xxxxx -n arcsight-installer-xxxxx
-c hercules-osp -- bash
```

10. Import the new RE certificate.

```
cd /usr/local/tomcat/conf/default/
```

```
keytool -importcert -storepass $KEYSTORE_PASSWORD -destkeystore \
sso.bcfks -alias updatedreca -file /vault-crt/trustedCAs/RE_ca.crt -storetype \
BCFKS -providerclass \
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider \
-providerpath /usr/local/openjdk-8/jre/lib/ext/bc-fips-1.0.2.jar
```

11. Restart pods of `fusion-single-sign-on-xxxxx` and `fusion-user-management-xxxxx`. Also, any other pod that uses the CA certificates needs to be restarted in order to new certificate to take effect, e.g: reporting (if applies).
12. Make sure to import the certificate to your browser's trust store for proper functionality of both Management and Fusion portals.

Restarting Pods

Pods of deployed capabilities that rely on the external certificates for communication will continue to use the existing certificate until they are restarted.

To restart pods:

1. The pods need to be gently scaled to 0 and later scaled back to original values. E.g. for transformation-hub, note the current amount of replicas by issuing (where here and now on xxxxx should correspond to unique suffix of your deployment):

```
kubectl get replicaset -n arcsight-installer-xxxxx
```

> The unique name of th-schemaregistry-xxxxxx and th-web-service-xxxxxx replicasets and amount of replicas you are currently running for them.

2. Scale them down to zero, by commands below. Make sure to repeat above commands to confirm they were scaled to zero and not running in terminating state.

```
kubectl scale --replicas=0 statefulset/th-kafka -n arcsight-installer-
xxxxx
kubectl scale --replicas=0 replicaset/th-schemaregistry-xxxxxx -n
arcsight-installer-xxxxx
kubectl scale --replicas=0 replicaset/th-web-service-xxxxxx -n arcsight-
installer-xxxxx
```

3. You can repeat commands afterward and increase replicas value to the normal state from before the procedure. Other capabilities also have pods that rely on certificates:

```
kubectl scale --replicas=x replicaset/itom-data-ingestion-loader-
xxxxxxxxx -n arcsight-installer-xxxxx
```

```
kubectl scale --replicas=x replicaset/fusion-db-search-engine-xxx -n
arcsight-installer-xxxxx
```

Apply the mentioned scaling procedure to all in exactly the same way.

Configuring ArcMC to Manage a Transformation Hub

ArcMC serves as the management UI for Transformation Hub. In order for ArcMC to manage a Transformation Hub, the Transformation Hub must be added as a managed host to ArcMC.

This process will include these steps, which are explained below:

- Retrieve the ArcMC certificate from your ArcMC
- Configure the CDF cluster with ArcMC details
- Retrieve the CDF certificate
- Configure ArcMC

Azure Integration - Additional Steps

For proper integration with Azure Transformation Hub, you must perform the following additional procedures for the ArcSight product (ArcMC, SmartConnector, CTH, Logger, or ESM) you are integrating.

- Edits to the `/etc/hosts` file
- Peering
- Configure health probes and load-balancing rules for ports 32080, 9093, and 9092. (For ESM, this applies only to port 9093.)

Each of these is detailed below.

Edit the `/etc/hosts` File

You must add each instance to the product's `/etc/hosts` file.

1. Open the product's `/etc/hosts` file in a text editor.
2. Add the internal IP and FQDN for each instance in the Azure Kubernetes service. (The instance IP and FQDN can be obtained by opening the AKS resource group and then opening the aks-nodepool virtual machine scale set.)
3. Save the changes to the file.

Example additions to `/etc/hosts:`

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
```

```
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

```
10.1.1.4 aks-nodepool1-12400006-vmss000000
```

```
10.1.1.5 aks-nodepool1-12400006-vmss000001
```

```
10.1.1.6 aks-nodepool1-12400006-vmss000002
```

Peering

If the Azure product and Azure Transformation Hub are on different VLANs, you must configure peering between the two VLANs. An example is provided in the section Peering Virtual Networks.

Health Probes

You must configure health probes and load balancing rules for ports 32080, 9093, and, optionally, 9092. Configuring health probes for these ports is detailed in Configuring Port Rules for Product Integration. Use the same procedure as outlined for port 443.

AWS Integration - AWS ArcMC with AWS Transformation Hub

To intergrate an AWS Transformation Hub with ArcMC, you will need to take the following additional steps, which must be performed for the following ports:

- 443
- 32080
- 32081 (Schema Registry port)
- 32101 - 32150 (used for CTH creation)

For each of the ports listed here, do the following:

1. Create a target group for the port.

> When creating Target Groups for the CTH ports (32101-32150), you will need to edit the Health Checks as follows:
> On the **Advanced Health Check Settings** tab, change the Success codes value from 200 (default) to: 200, 401, 404.
> Then, create the Target Group. Note that before a CTH is deployed, the health status for the Target Groups for this port range will be *unhealthy*. After a CTH is deployed, the health status will

> change to *healthy*.

2. Register all of the worker nodes in your cluster to the target group.
3. Add a listener to the ALB for the port using the defaults for all parameters.
4. Ensure that your ArcMC is added to the the AWS Transformation Hub cluster Security Group, with rules allowing access to the above ports.

To retrieve the ArcMC certificate:

1. Log into ArcMC.
2. Click **Administration > System Admin > SSL Server Certificate > Generate Certificate.**
3. On the **Enter Certificate Settings** dialog, enter the required settings. In **Hostname**, your certificate settings must match the FQDN of your ArcMC.
4. Click **Generate Certificate.**
5. Once the certificate is generated, click **View Certificate** and copy the full content from `--BEGIN cert to END cert--` to the clipboard.

Configure the CDF cluster:

1. Log in to the CDF management portal.
2. Select **Deployment > Deployments.**
3. Click **...** (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
3. Scroll down to the Management Center Configuration section. Then, enter values as described for the following:
   - **Username:** `admin`
   - Password: Use your Transformation Hub password.
   - Enter the ArcMC hostname and port 443 (for example, `arcmc.example.com:443`). If ArcMC was installed as a non-`root` user, enter port 9000 instead.
   - **ArcMC certificates:** Paste the text of the generated server certificates you copied to the clipboard as described above.
4. Click **Save**. Web services pods in the cluster will be restarted

To retrieve the CDF certificate:

1. On the initial master node of the cluster, run the following:
```
# ${K8S_HOME}/scripts/cdf-updateRE.sh
```

2. Copy the contents of this certificate, from `--BEGIN cert to END cert--`, to the clipboard for use in the next procedure, Configuring ArcMC.

To configure ArcMC:

> If Azure ArcMC and Azure Transformation Hub are in different VLANs, then before configuring, enable peering between the 2 products as described in Peering Virtual Networks. Otherwise, begin with Step 1.

1. Log in to the ArcMC.
2. Click **Node Management > View All Nodes.**
3. In the navigation bar, click Default (or the ArcMC location where you wish to add Transformation Hub). Then click **Add Host,** and enter the following values:
    - **Type:** Select Transformation Hub Containerized
    - **Hostname:** Virtual IP of the Transformation Hub for an HA environment, or master node hostname for any single-master node environment.
    - **Port**: 32080
    - **Cluster Port:** 443
    - **Cluster Username:** admin
    - **Cluster Password:** <use CDF Management Portal password>
    - **Cluster Certificate:** Paste the contents of the CDF certificate you copied earlier.
4. Click **Add**. The Transformation Hub is added as a managed host.

Configuring Security Mode for Smart Connectors with Transformation Hub Destinations

Follow these instructions to configure a security mode for SmartConnectors with destinations on an SSL secured Transformation Hub destinations.

These procedures are provided with the following assumptions:

- You use the default password. See the appendix for FIPS Compliant SmartConnectors in the *SmartConnector User Guide* on the Micro Focus support community to set a non-default password.
- You are on the Linux platform. For Windows platforms, use backslashes (\) when entering commands instead of the forward slashes given here.

- You are using a command window to enter Windows commands. Do not use Windows PowerShell.

Azure Integration - Additional Steps

For proper integration with Azure Transformation Hub, you must perform the following additional procedures for the ArcSight product (ArcMC, SmartConnector, CTH, Logger, or ESM) you are integrating.

- Edits to the `/etc/hosts` file
- Peering
- Configure health probes and load-balancing rules for ports 32080, 9093, and 9092. (For ESM, this applies only to port 9093.)

Each of these is detailed below.

Edit the `/etc/hosts` File

You must add each instance to the product's `/etc/hosts` file.

1. Open the product's `/etc/hosts` file in a text editor.
2. Add the internal IP and FQDN for each instance in the Azure Kubernetes service. (The instance IP and FQDN can be obtained by opening the AKS resource group and then opening the aks-nodepool virtual machine scale set.)
3. Save the changes to the file.

Example additions to `/etc/hosts:`

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
```

```
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

```
10.1.1.4 aks-nodepool1-12400006-vmss000000
```

```
10.1.1.5 aks-nodepool1-12400006-vmss000001
```

```
10.1.1.6 aks-nodepool1-12400006-vmss000002
```

Peering

If the Azure product and Azure Transformation Hub are on different VLANs, you must configure peering between the two VLANs. An example is provided in the section Peering

Virtual Networks.

Health Probes

You must configure health probes and load balancing rules for ports 32080, 9093, and, optionally, 9092. Configuring health probes for these ports is detailed in Configuring Port Rules for Product Integration. Use the same procedure as outlined for port 443.

AWS Integration - Additional Steps

For proper integration with AWS Transformation Hub, you must perform the following additional procedures for the ArcSight product (SmartConnectors, Logger, or ESM) you are integrating.

For configuring SmartConnectors, Loggers, and ESMs: obtain the cluster worker node (Kafka broker node) host names, using one of the following procedures:

1. From the bastion host, run the following command:
   ```
   # kubectl get nodes
   ```
2. Copy the node host names.

OR

1. In the AWS UI, go to your Auto Scaling Group.
2. Click on the **Instance Management** tab to see the instance IDs.
3. Click the first instance ID to view the details of the corresponding instance.
4. Note the Private DNS name for the instance.
5. Repeat steps 3-4 for each instance ID.

After determining these values, ensure that your SmartConnector, Logger, or ESM instance is added to the the AWS TH cluster Security Group with rules allowing access to the ports 32080, 9093, and optionally 9092.

You can then follow the integration procedures below.

Configuring a SmartConnector with a Transformation Hub Destination without Client Authentication in non-FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub destination without client authentication in non-FIPS mode. This is the default security mode configuration when installing Transformation Hub.

On the SmartConnector Server

1. Prepare the SmartConnector:
   - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Disabled**.
   - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Disabled**.
2. Navigate to the connector's `current` directory, for example:
   ```
   # cd <install dir>/current
   ```
3. Set the environment variables for the static values used by keytool, for example:
   ```
   # export CURRENT=<full path to this "current" folder>
   # export TH=<Transformation Hub hostname>_<Transformation Hub port>
   # export STORES=${CURRENT}/user/agent/stores
   # export CA_CERT=ca.cert.pem
   # export STORE_PASSWD=changeit
   ```
   On Windows platforms:
   ```
   # set CURRENT=<full path to this "current" folder>
   # set TH=<Transformation Hub hostname>_<Transformation Hub port>
   # set STORES=%CURRENT%\user\agent\stores
   # set STORE_PASSWD=changeit
   ```
4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist, for example:
   ```
   mkdir -p ${STORES}
   ```
   On Windows platforms:
   ```
   mkdir -p %STORES%
   ```

On the Transformation Hub:

Create a `${CA_CERT}` file with the content of the root CA certificate as follows:

1. Set the environment:
   ```
   # export CA_CERT=/tmp/ca.cert.pem
   ```

2. Create a certificate:

   ```
   # ${k8s-home}/scripts/cdf-updateRE.sh > ${CA_CERT}
   ```

> For a cloud installation, the `cdf-updateRE.sh` script has a different path:
> AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
> Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

3. Copy this file from the Transformation Hub to the connector `STORES` directory.

On the Connector:

1. Import the CA certificate to the trust store in the `${CURRENT}` folder; for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -
   keystore ${STORES}/${TH}.truststore.jks -storepass ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\%CA_CERT% -alias CARoot -
   keystore %STORES%\%TH%.truststore.jks -storepass %STORE_PASSWD%
   ```

2. When prompted, enter `yes` to trust the certificate.
3. Note the trust store path:

   ```
   # echo ${STORES}/${TH}.truststore.jks
   ```

   On Windows platforms:

   ```
   # echo %STORES%\%TH%.truststore.jks
   ```

4. Navigate to the `bin` directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

   ```
   # cd <installation dir>/current/bin
   # ./runagentsetup.sh
   ```

   On Windows platforms:

   ```
   # cd <installation dir>\current\bin
   # runagentsetup.bat
   ```

5. Set **Use SSL/TLS** to **true**.
6. Set **Use SSL/TLS Authentication** to **false**.
7. When completing the Transformation Hub destination fields, use the value from Step 3 for the trust store path and the password used in Step 4 for the trust store password.
8. Cleanup. Delete the certificate file, for example:

Caution: The following file should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

```
# rm ${STORES}/${CA_CERT}
```

On Windows platforms:

```
# del %\STORES%\%CA_CERT%
```

Configuring a SmartConnector with a Transformation Hub Destination with Client Authentication in FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub (TH) destination with client authentication in FIPS mode.

You will need to supply an intermediate certificate and key.

Step 1: On the SmartConnector Server

1. Prepare the connector:
   - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Enabled**.
   - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.
2. Navigate to the connector's `current` directory, for example:
   ```
   # cd <install dir>/current
   ```
3. Apply the following workaround for a Java keytool issue:
   a. Create a new file, `agent.security`, in `<install dir>/current/user/agent` (or in Windows platforms, `<install dir>\current\user\agent`).

   b. Add the following content to the new file and then save it:
   ```
   security.provider.1=org.bouncycastle.jcajce.provider
   .BouncyCastleFipsProvider
   security.provider.2=com.sun.net.ssl.internal.ssl.Provider BCFIPS
   ```

```
security.provider.3=sun.security.provider.Sun
```

c. Move the `lib/agent/fips/bcprov-jdk14-119.jar` file to the `current` directory.

4. Set the environment variables for static values used by keytool:

```
# export CURRENT=<full path to this "current" folder>
# export BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
Djava.security.egd=file:/dev/urandom -J-
Djava.ext.dirs=${CURRENT}/lib/agent/fips -J-
Djava.security.properties=${CURRENT}/user/agent/agent.security"
# export TH=<Transformation Hub hostname>_<Transformation Hub port>
# export STORES=${CURRENT}/user/agent/stores
# export STORE_PASSWD=changeit
# export TH_HOST=<TH master host name>
# export CA_CERT=ca.cert.pem
# export INTERMEDIATE_CA_CRT=intermediate.cert.pem
# export FIPS_CA_TMP=/opt/fips_ca_tmp
```

On Windows platforms:

```
# set CURRENT=<full path to this "current" folder>
# set BC_OPTS=-storetype BCFKS -providername BCFIPS
-J-Djava.ext.dirs=%CURRENT%\jre\lib\ext;%CURRENT%\lib\agent\fips
-J-Djava.security.properties=%CURRENT%\user\agent\agent.security
# set TH=<Transformation Hub hostname>_<Transformation Hub port>
# set STORES=%CURRENT%\user\agent\stores
# set STORE_PASSWD=changeit
# set TH_HOST=<TH master host name>
# set CA_CERT=C:\Temp\ca.cert.pem
# set INTERMEDIATE_CA_CRT=C:\Temp\intermediate.cert.pem
# set FIPS_CA_TMP=\opt\fips_ca_tmp
```

5. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist, for example:

```
# mkdir -p ${STORES}
```

On Windows platforms:

```
# mkdir -p %STORES%
```

6. Create the connector key pair, for example (the connector `FQDN`, `OU`, `O`, `L`, `ST`, and `C` values must be changed for your company and location):

```
# jre/bin/keytool ${BC_OPTS} -genkeypair -alias ${TH} -keystore
${STORES}/${TH}.keystore.bcfips -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 365
```

On Windows platforms:

```
# jre\bin\keytool %BC_OPTS% -genkeypair -alias %TH% -keystore
%STORES%\%TH%.keystore.bcfips -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF ,L=Sunnyvale,ST=CA,C=US" -validity 365
```

If the command fails, set BC_OPTS as follows and create the connector key pair:

```
# export BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
Djava.security.egd=file:/dev/urandom -providerpath $
{CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
```

For Connector 8.0 or earlier, use `bc-fips-1.0.0.jar` in the above command.

When prompted, enter the password. Note the password; you will need it again in a later step. Press **Enter** to use the same password for the key. If you want to match the default value in the properties file, use the password changeit.

7. List the key store entries. There should be one private key.

   ```
   # jre/bin/keytool ${BC_OPTS} -list -keystore
   ${STORES}/${TH}.keystore.bcfips -storepass ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -list -keystore
   %STORES%\%TH%.keystore.bcfips -storepass %STORE_PASSWD%
   ```

8. Create a Certificate Signing Request (CSR), for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -certreq -alias ${TH} -keystore
   ${STORES}/${TH}.keystore.bcfips -file ${STORES}/${TH}-cert-req -storepass
   ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -certreq -alias %TH% -keystore
   %STORES%\%TH%.keystore.bcfips -file %STORES%\%TH%-cert-req -storepass
   %STORE_PASSWD%
   ```

Step 2: On the Transformation Hub Server

1. When Transformation Hub is first installed, it is configured to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an

intermediate certificate and key pair. Copy them to /tmp with the following names:

```
/tmp/intermediate.cert.pem

/tmp/intermediate.key.pem

/tmp/ca.cert.pem
```

Use the following command to update the certificate on the Transformation Hub:

```
# /opt/arcsight/kubernetes/scripts/cdf-updateRE.sh write --re-
key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-
ca=/tmp/ca.cert.pem
```

> For a cloud installation, the `cdf-updateRE.sh` script has a different path:
> AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
> Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

> After the new certificate is imported to the Transformation Hub, the Transformation Hub will need to be uninstalled and then re-installed with FIPS and Client Authentication enabled.

3. Run the following commands:

```
# export CA_CERT=/tmp/ca.cert.pem

# export INTERMEDIATE_CA_CRT=/tmp/intermediate.cert.pem

# export INTERMEDIATE_CA_KEY=/tmp/intermediate.key.pem

# export FIPS_CA_TMP=/opt/fips_ca_tmp

# export TH=<Transformation Hub hostname>_<Transformation Hub port>
```

4. Create a temporary location on the Transformation Hub master node:

```
# mkdir $FIPS_CA_TMP
```

Step 3: On the Connector Server

Copy the `${STORES}/${TH}-cert-req` file (`%STORES%\%TH%-cert-req` on Windows platforms) from the connector to the Transformation Hub directory created above, `/opt/fips_ca_tmp`.

Step 4: On the Transformation Hub Server

Create the signed certificate, for example:

```
@ /bin/openssl x509 -req -CA ${INTERMEDIATE_CA_CRT} -CAkey ${INTERMEDIATE_CA_
KEY} -in ${TH}-cert-req -out ${FIPS_CA_TMP}/${TH}-cert-signed -days 365 -
CAcreateserial -sha256
```

Step 5: On the Connector Server

1. Copy the `${TH}-cert-signed` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the `%TH%-cert-signed` certificate to the connector's `%STORES%` directory.)
2. Copy the `ca.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
3. Copy the `intermediate.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
4. Import the CA certificate to the trust store, for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${CA_CERT} -alias
   CARoot -keystore ${STORES}/${TH}.truststore.bcfips -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%CA_CERT% -alias
   CARoot -keystore %STORES%\%TH%.truststore.bcfips -storepass %STORE_
   PASSWD%
   ```

5. Import the intermediate certificate to the trust store, for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${INTERMEDIATE_
   CA_CRT} -alias INTCARoot -keystore ${STORES}/${TH}.truststore.bcfips -
   storepass ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%INTERMEDIATE_
   CA_CRT% -aliasINTCARoot -keystore %STORES%\%TH%.truststore.bcfips -
   storepass %STORE_PASSWD%
   ```

6. Import the CA certificate to the key store, for example:

   ```
   jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${CA_CERT} -alias
   CARoot -keystore ${STORES}/${TH}.keystore.bcfips -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%CA_CERT% -alias
   CARoot -keystore %STORES%\%TH%.keystore.bcfips -storepass %STORE_
   PASSWD%
   ```

7. When prompted, enter **yes** to trust the certificate.
8. Import the intermediate certificate to the key store, for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${INTERMEDIATE_
   CA_CRT} -alias INTCARoot -keystore ${STORES}/${TH}.keystore.bcfips -
   storepass ${STORE_PASSWD}
   ```

   When completed successfully, this command will return the message, `Certificate reply was installed in keystore`.

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%INTERMEDIATE_
   CA_CRT% -alias
   ```

   ```
   # INTCARoot -keystore %STORES%\%TH%.keystore.bcfips -storepass %STORE_
   PASSWD%
   ```

   If the command fails, set `BC_OPTS` as follows and create the connector key pair:

   ```
   # export BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
   Djava.security.egd=file:/dev/urandom -providerpath $
   {CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -providerclass
   org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
   ```

   For Connector 8.0 or earlier, use `bc-fips-1.0.0.jar` in the above command.

9. Import the signed certificate to the key store, for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${TH}-cert-signed
   -alias ${TH} -keystore ${STORES}/${TH}.keystore.bcfips -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%TH%-cert-signed
   -alias %TH% -keystore %STORES%\%TH%.keystore.bcfips -storepass %STORE_
   PASSWD%
   ```

   When successfully complete, this command will return the message, *Certificate reply was installed in keystore.*

10. Navigate to the `bin` directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

    ```
    # cd <installation dir>/current/bin
    # ./runagentsetup.sh
    ```

    On Windows platforms:

```
# cd <installation dir>\current\bin
# runagentsetup.bat
```

a. When completing the Transformation Hub destination fields, use the same values as in Step 8 for the path and password.

b. Set **Use SSL/TLS** to **true**.

c. Set **Use SSL/TLS Authentication** to **true**.

d. Set keystore path to:
`${STORES}/${TH}.keystore.bcflips`

e. Set truststore path to:
`${STORES}/${TH}.keystore.bcflips`

11. Cleanup. Delete the following files:

> **Caution:** The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

```
# rm ${STORES}/${INTERMEDIATE_CA_CRT}
# rm ${STORES}/intermediate.key.pem
# rm ${STORES}/${TH}-cert-signed
# rm ${STORES}/${TH}-cert-req
```

On Windows platforms:

```
# del %STORES%\intermediate.cert.pem
# del %STORES%\intermediate.key.pem
# del %STORES%\%TH%-cert-signed
# del %STORES%\%TH%-cert-req
```

12. Move the `bcprov-jdk14-119.jar` file back to the `lib/agent/fips` directory (or `lib\agent\fips` on Windows platforms).

Step 6: On the Transformation Hub Server

To clean up the Transformation Hub server, delete the temporary folder where the certificate was signed and the certificate and key files in `/tmp`.

> **Caution:** The temporary certificate folder should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

SmartConnectors on AWS

When configuring a SmartConnector in AWS, for TLS with FIPS and client authentication enabled, the keytool keypair creation command might fail or appear to hang if the available entropy on the connector instance is less that 1000. It has been found that AWS instances installed with a minimum OS will have an entropy availability of only about 60.

You can check this by installing the rng-tools package and then running the a `cat` command on the connector host:

```
# yum install rng-tools -y
```

```
# cat /proc/sys/kernel/random/entropy_avail
```

If the available entropy needs to be increased, enable the `rngd` service at boot and start the `rngd` service with the following commands:

```
# systemctl enable rngd.service
```

```
# systemctl start rngd.service
```

> The `rngd` service will check and feed random data from the hardware device to kernel entropy pool automatically.

Then run this command again to check available entropy:

```
# cat /proc/sys/kernel/random/entropy_avail
```

After increasing the available entropy, the keytool `create keypair` command will run normally.

Configuring a SmartConnector with Transformation Hub Destination with Client Authentication in Non-FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub (TH) destination with client authentication, in non-FIPS mode.

> You will need to supply an intermediate certificate and key.

On the SmartConnector Server

1. Prepare the SmartConnector:
   - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Disabled**.
   - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Disabled**.
2. Navigate to the connector's `current` directory, for example:
   ```
   # cd <install dir>/current
   ```
   On Windows platforms:
   ```
   # cd <install dir>\current
   ```
3. Set the environment variables for the static values used by keytool, for example:
   ```
   # export CURRENT=<full path to this "current" folder>
   # export TH=<th hostname>_<th port>
   # export STORES=${CURRENT}/user/agent/stores
   # export STORE_PASSWD=changeit
   # export TH_HOST=<TH master host name>
   # export CA_CERT=ca.cert.pem
   # export INTERMEDIATE_CA_CRT=intermediate.cert.pem
   export CERT_CA_TMP=/opt/cert_ca_tmp
   ```
   On Windows platforms:
   ```
   # set CURRENT=<full path to this "current" folder>
   # set TH=<th hostname>_<th port>
   # set STORES=%CURRENT%\user\agent\stores
   # set STORE_PASSWD=changeit
   # set TH_HOST=<TH master host name>
   # set CA_CERT=C:\Temp\ca.cert.pem
   # set INTERMEDIATE_CA_CRT=C:\Temp\intermediate.cert.pem
   # set CERT_CA_TMP=\opt\cert_ca_tmp
   ```
4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist, for example:
   ```
   mkdir -p ${STORES}
   ```
   On Windows platforms:

```
# mkdir -p %STORES%
```

5. Create the connector key pair, for example:

   ```
   # jre/bin/keytool -genkeypair -alias ${TH} -keystore
   ${STORES}/${TH}.keystore.jks -dname "cn=<Connector
   FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 365
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -genkeypair -alias %TH% -keystore
   %STORES%\%TH%.keystore.jks -dname "cn=<Connector
   FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 365
   ```

   When prompted, enter the password. Note the password; you will need it again in a later step. Press Enter to use the same password for the key.

6. List the key store entries. There should be one private key.

   ```
   # jre/bin/keytool -list -keystore ${STORES}/${TH}.keystore.jks -storepass
   ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -list -keystore %STORES%\%TH%.keystore.jks -storepass
   %STORE_PASSWD%
   ```

7. Create a Certificate Signing Request (CSR), for example:

   ```
   # jre/bin/keytool -certreq -alias ${TH} -keystore
   ${STORES}/${TH}.keystore.jks -file ${STORES}/${TH}-cert-req -storepass
   ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -certreq -alias %TH% -keystore
   %STORES%\%TH%.keystore.jks -file %STORES%\%TH%-cert-req -storepass
   %STORE_PASSWD%
   ```

On the Transformation Hub Server

1. When Transformation Hub is first installed, it's setup to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an intermediate certificate and key pair. Copy them to `/tmp` with the following names:

   ```
   /tmp/intermediate.cert.pem
   /tmp/intermediate.key.pem
   /tmp/ca.cert.pem
   ```

   Use the following command to add them to Transformation Hub:

```
# /opt/arcsight/kubernetes/scripts/cdf-updateRE.sh write --re
key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-
ca=/tmp/ca.cert.pem
```

> AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
> Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

> After the new certificate is imported to the Transformation Hub, the Transformation Hub will need to be uninstalled and then re-installed with FIPS and Client Authentication enabled.

2. Run the following commands:

   ```
   # export CA_CERT=/tmp/ca.cert.pem
   # export INTERMEDIATE_CA_CRT=/tmp/intermediate.cert.pem
   # export INTERMEDIATE_CA_KEY=/tmp/intermediate.key.pem
   # export CERT_CA_TMP=/opt/cert_ca_tmp
   # export TH=<Transformation Hub hostname>_<Transformation Hub port>
   ```

3. Create a temporary location on the Transformation Hub master node:

   ```
   # mkdir $CERT_CA_TMP
   ```

On the Connector Server

Copy the `${STORES}/${TH}-cert-req` file (`%STORES%\%TH%-cert-req` on Windows platforms) from the connector to the Transformation Hub directory created above.

On the Transformation Hub Server

Create the signed certificate, for example:

```
/bin/openssl x509 -req -CA ${INTERMEDIATE_CA_CRT} -CAkey ${INTERMEDIATE_CA_
KEY} -in ${TH}-cert-req -out ${CERT_CA_TMP}/${TH} -cert-signed-days 365 -
CAcreateserial -sha256
```

On the Connector Server

1. Copy the `${TH}-cert-signed` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the `%TH%-cert-signed` certificate to the connector's `%STORES%` directory.)
2. Copy the `ca.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)

3. Copy the `intermediate.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
4. Import the CA certificate to the trust store, for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot
   -keystore ${STORES}/${TH}.truststore.jks -storepass ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\%CA_CERT% -alias CARoot
   -keystore %STORES%\%TH%.truststore.jks -storepass %STORE_PASSWD%
   ```

5. Import the intermediate certificate to the trust store, for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${INTERMEDIATE_CA_CRT} -
   alias
   ```

   ```
   # INTCARoot -keystore ${STORES}/${TH}.truststore.jks -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\%INTERMEDIATE_CA_CRT% -
   aliasINTCARoot -keystore %STORES%\%TH%.truststore.jks -storepass
   %STORE_PASSWD%
   ```

6. When prompted, enter **yes** to trust the certificate.
7. Import the CA certificate to the key store, for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -
   keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\${CA_CERT} -alias CARoot -
   keystore %STORES%\%TH%.keystore.jks -storepass %STORE_PASSWD%
   ```

8. Import the intermediate certificate to the key store, for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${INTERMEDIATE_CA_CRT} -
   alias
   ```

   ```
   # INTCARoot -keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\%INTERMEDIATE_CA_CRT% -
   alias INTCARoot -keystore %STORES%\%TH%.keystore.jks -storepass %STORE_
   PASSWD%
   ```

When successfully completed, this command will return the message, *Certificate reply was installed in keystore*.

9. When prompted, enter **yes** to trust the certificate.

10. Import the signed certificate to the key store, for example:

    ```
    # jre/bin/keytool -importcert -file ${STORES}/${TH}-cert-signed -alias
    ${TH} -keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_PASSWD}
    ```

    On Windows platforms:

    ```
    # jre\bin\keytool -importcert -file %STORES%\%TH%-cert-signed -alias
    %TH% -keystore %STORES%\%TH%.keystore.jks -storepass %STORE_PASSWD%
    ```

    When successfully complete, this command will return the message, `Certificate reply was installed in keystore`.

11. Note the key store and trust store paths:

    ```
    #echo ${STORES}/${TH}.truststore.jks
    # echo ${STORES}/${TH}.keystore.jks
    ```

    On Windows platforms:

    ```
    # echo %STORES%\%TH%.truststore.jks
    # echo %STORES%\%TH%.keystore.jks
    ```

12. Navigate to the `bin` directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

    ```
    # cd <installation dir>/current/bin
    # ./runagentsetup.sh
    ```

    On Windows platforms:

    ```
    # cd <installation dir>\current\bin
    # runagentsetup.bat
    ```

    a. When completing the Transformation Hub destination fields, use the same values as in Step 8 for the path and password.

    b. Set **Use SSL/TLS** to **true**.

    c. Set **Use SSL/TLS Authentication** to **true**.

13. Cleanup. Delete the following files:

    Caution: The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

    ```
    # rm ${STORES}/${INTERMEDIATE_CA_CRT}
    # rm ${STORES}/intermediate.key.pem
    ```

```
# rm ${STORES}/${TH}-cert-signed
# rm ${STORES}/${TH}-cert-req
```

On Windows platforms:

```
# del %STORES%\intermediate.cert.pem
# del %STORES%\intermediate.key.pem
# del %STORES%\%TH%-cert-signed
# del %STORES%\%TH%-cert-req
```

On the Transformation Hub Server

To clean up the Transformation Hub server, delete the temporary folder where the certificate was signed and the certificate and key files in `/tmp`.

> **Caution:** The temporary certificate folder should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

Configuring a SmartConnector with a Transformation Hub Destination without Client Authentication in FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub destination without client authentication in FIPS mode.

On the SmartConnector Server

1. Prepare the SmartConnector:

   - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Enabled**.

   - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and then **Set FIPS Mode** to **Enabled**.

2. Navigate to the connector's `current` directory, for example:

   ```
   # cd <install dir>/current
   ```

3. Set the environment variables for the static values used by keytool, for example:

   ```
   # export CURRENT=<full path to this "current" folder>
   ```

```
# export BC_OPTS="-storetype BCFKS -providername BCFIPS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar
-J-Djava.security.egd=file:/dev/urandom"
```

> For Connector 8.0, use `bc-fips-1.0.0.jar` in the above command.

```
# export TH=<Transformation Hub hostname>_<Transformation Hub port>
# export STORES=${CURRENT}/user/agent/stores
# export STORE_PASSWD=changeit
# export CA_CERT=ca.cert.pem
```

On Windows platforms:

```
# set CURRENT=<full path to this "current" folder>
# set BC_OPTS="-storetype BCFKS -providername BCFIPS
-J-Djava.ext.dirs=%CURRENT%\jre\lib\ext;%CURRENT%\lib\agent\fips
-J-Djava.security.properties=%CURRENT%\user\agent\agent.security"
# set TH=<Transformation Hub hostname>_<Transformation Hub port>
# set STORES=%CURRENT%\user\agent\stores
# set STORE_PASSWD=changeit
```

4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist, for example:

   ```
   # mkdir -p ${STORES}
   ```

   On Windows platforms:

   ```
   # mkdir -p %STORES%
   ```

5. Create a `ca.cert.pem` file with the contents of the root CA certificate with the following command:

   ```
   # ${k8s-home}/scripts/cdf-updateRE.sh > /tmp/ca.cert.pm
   ```

   > For a cloud installation, the `cdf-updateRE.sh` script has a different path:
   > AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
   > Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

6. Copy the just-created `ca.cert.pem` file from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)

7. Import the CA certificate to the trust store, for example:

```
# jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${CA_CERT} -alias
CARoot -keystore ${STORES}/${TH}.truststore.bcfips -storepass ${STORE_
PASSWD}
```

On Windows platforms:

```
# jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%CA_CERT% -alias
CARoot -keystore %STORES%\%TH%.truststore.bcfips -storepass %STORE_
PASSWD%
```

8. When prompted, enter **yes** to trust the certificate.
9. Note the trust store path:

```
# echo ${STORES}/${TH}.truststore.bcfips
```

On Windows platforms:

```
# echo %STORES%\%TH%.truststore.bcfips
```

10. Navigate to the `bin` directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

```
# cd <installation dir>/current/bin
# ./runagentsetup.sh
```

On Windows platforms:

```
cd <installation dir>\current\bin
runagentsetup.bat
```

   a. When completing the Transformation Hub destination fields, use the value from Step 7 for the trust store path and the password used in Step 6 for the trust store password.
   b. Set **Use SSL/TLS** to **true**.
   c. Set **Use SSL/TLS Authentication** to **false**.

11. Cleanup. Delete the certificate file, for example:

> **Caution:** The following file should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

```
# rm ${STORES}/${CA_CERT}
```

On Windows platforms:

```
# del %\STORES%\ca.cert.pem
```

Configuring Logger as a Transformation Hub Consumer

The procedure for configuring a Logger as a Transformation Hub consumer will depend on whether the Logger will be using SSL/TLS.

Prerequisite: Import RH CA Certificate

Prior to configuring a Logger as a Transformation Hub consumer, you manually import the Realm External CA (RE CA) certificate exported from the TH clusterThe RE certificate does not change if the Transformation Hub is restarted or redeployed. Any newly generated certificates after restart are trusted by this RE CA, enabling the receiver to continue accepting events.

To configure Transformation Hub using TLS, the user is only required to create the TH receiver and import the files as described below; Authentication and Sign in are done automatically. Meanwhile, for Client Authentication, you must set up two way authentication between the Container Deployment Foundation as described in Transformation Hub Authentication.

Step 1: Obtain Transformation Hub RE Certificate

1. On the Transformation Hub , run the following command to retrieve the Transformation Hub RE certificate:

```
/opt/kubernetes/scripts/cdf-updateRE.sh > /tmp/RE.crt
```

> For a cloud installation, the `cdf-updateRE.sh` script has a different path:
> AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
> Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

2. Copy the `/tmp/RE.crt` obtained from step 1 to the Logger in the directory `/tmp;`.

Step 2: Set the environment on the Logger

1. On the Logger, set the `ARCSIGHT_HOME` environment variable:

- Appliance:
  ```
  # export ARCSIGHT_HOME=/opt/arcsight/logger
  ```
- Software:
  ```
  # export ARCSIGHT_HOME=[logger install directory]/current/arcsight/logger
  ```

For existing Kafka receivers only:

1. In the Logger SSH console, look for any previous TH certificates from Logger receiver trust store running the script available at:
   - Appliance:
     `/opt/arcsight/logger/bin/scripts/keytool_util.sh.`
   - Software:
     `# [Install dir]/current/arcsight/logger/bin/scripts/keytool_util.sh`

   > **Caution:**`<verisignserverca>` uses a 1000-bit RSA key which is considered a security risk.

2. Delete the TH certificates from the previous step in the Logger receiver trust store running the script available at:

```
# /opt/arcsight/logger/bin/scripts/keytool_util.sh
```

Make sure to execute the command as it follows: `./keytool_util.sh receiver delete [alias]`

Step 3: Import the RE Certificate into Logger

1. In the Logger SSH console, import the new TH RE certificate using the `RE.crt` file copied from TH running the script available at:

   `# /opt/arcsight/logger/bin/scripts/keytool_util.sh`.

   Make sure to execute the command as it follows:
   `# ./keytool_util.sh receiver importcert [certificate]`

2. Confirm TH FQDN is settled in Logger DNS before creating Kafka receivers in SSL mode in Logger.

For existing Kafka receivers:

On the Logger, restart the receiver processes available at:

- Appliance:
  `# /opt/arcsight/logger/bin/loggerd restart receivers`
- Software:
  `# [install dir]/current/arcsight/logger/bin/loggerd restart receivers`

Azure Integration - Additional Steps

For proper integration with Azure Transformation Hub, you must perform the following additional procedures for the ArcSight product (ArcMC, SmartConnector, CTH, Logger, or ESM) you are integrating.

- Edits to the `/etc/hosts` file
- Peering
- Configure health probes and load-balancing rules for ports 32080, 9093, and 9092. (For ESM, this applies only to port 9093.)

Each of these is detailed below.

Edit the `/etc/hosts` File

You must add each instance to the product's `/etc/hosts` file.

1. Open the product's `/etc/hosts` file in a text editor.
2. Add the internal IP and FQDN for each instance in the Azure Kubernetes service. (The instance IP and FQDN can be obtained by opening the AKS resource group and then opening the aks-nodepool virtual machine scale set.)
3. Save the changes to the file.

Example additions to `/etc/hosts`:

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
```

```
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

```
10.1.1.4 aks-nodepool1-12400006-vmss000000
```

```
10.1.1.5 aks-nodepool1-12400006-vmss000001
```

```
10.1.1.6 aks-nodepool1-12400006-vmss000002
```

Peering

If the Azure product and Azure Transformation Hub are on different VLANs, you must configure peering between the two VLANs. An example is provided in the section Peering Virtual Networks.

Health Probes

You must configure health probes and load balancing rules for ports 32080, 9093, and, optionally, 9092. Configuring health probes for these ports is detailed in Configuring Port Rules for Product Integration. Use the same procedure as outlined for port 443.

AWS Integration - Additional Steps

For proper integration with AWS Transformation Hub, you must perform the following additional procedures for the ArcSight product (SmartConnectors, Logger, or ESM) you are integrating.

For configuring SmartConnectors, Loggers, and ESMs: obtain the cluster worker node (Kafka broker node) host names, using one of the following procedures:

1. From the bastion host, run the following command:
   ```
   # kubectl get nodes
   ```
2. Copy the node host names.

OR

1. In the AWS UI, go to your Auto Scaling Group.
2. Click on the **Instance Management** tab to see the instance IDs.
3. Click the first instance ID to view the details of the corresponding instance.
4. Note the Private DNS name for the instance.
5. Repeat steps 3-4 for each instance ID.

After determining these values, ensure that your SmartConnector, Logger, or ESM instance is added to the the AWS TH cluster Security Group with rules allowing access to the ports 32080, 9093, and optionally 9092.

You can then follow the integration procedures below.

To configure a SmartConnector as a Transformation Hub consumer (not using SSL/TLS):

1. Log in to Logger.
2. Select **Configuration > Receivers > Add.**
3. In the **Add Receiver** dialog, enter the following:
   - **Name:** Enter a unique name for the new receiver.
   - **Type:** Transformation Hub Receiver

4. Select and edit the Transformation Hub Receiver and enter the following parameters:
   - **Transformation Hub host(s) and port:** {Kafka broker Host IP 1}:9092, {Kafka broker Host IP 2}:9092, {Kafka broker Host IP 3}:9092
   - **Event Topic List:** th-cef (If additional topics are needed, enter multiple topics with a comma-separated list.)
   - **Retrieve event from earliest offset:** true
   - **Consumer Group (Logger Pool):** Logger Pool
   - **Use SSL/TLS:** false
   - **Use Client Authentication:** false
   - **Enable:** Checked

To configure a Logger as a Transformation Hub consumer (using SSL/TLS):

1. Log in to Logger.
2. Select **Configuration > Receivers > Add.**
3. In the **Add Receiver** dialog, enter the following:
   - **Name:** Transformation Hub Receiver
   - **Type:** Transformation Hub Receiver
4. Select and edit the Transformation Hub Receiver and enter the following parameters:
   - **Transformation Hub host(s) and port:** {Kafka broker Host IP 1}:9093, {Kafka broker Host IP 2}:9093, {Kafka broker Host IP 3}:9093
   - **Event Topic List:** th-cef (You can enter multiple topics with a comma-separated list.)
   - **Retrieve event from earliest offset:** true
   - **Consumer Group (Logger Pool):** Logger Pool
   - **Use SSL/TLS:** true
   - **Use Client Authentication:** false
   - **Enable:** Checked

To configure a Logger as a Transformation Hub consumer (using SSL/TLS with Client Authentication):

1. Log in to Logger.
2. Select **Configuration > Receivers > Add.**

3. In the **Add Receiver** dialog, enter the following:
   - **Name:** Transformation Hub Receiver
   - **Type:** Transformation Hub Receiver
4. Select and edit the Transformation Hub Receiver and enter the following parameters:
   - **Transformation Hub host(s) and port:** {Kafka broker Host IP 1}:9093, {Kafka broker Host IP 2}:9093, {Kafka broker Host IP 3}:9093
   - **Event Topic List:** th-cef (You can enter multiple topics with a comma-separated list.)
   - **Retrieve event from earliest offset:** true
   - **Consumer Group (Logger Pool):** Logger Pool
   - **Use SSL/TLS:** true
   - **Use Client Authentication:** true
   - **Enable:** Checked

Troubleshooting

The following troubleshooting tips might be useful in diagnosing Logger integration issues.

Error Message	Issue
IP Address th1.example.com is not a valid address	Use IP addresses in Receiver configuration, not host names.
There was a problem contacting Transformation Hub: Timeout expired while fetching topic metadata, please check the receiver configuration	Logger can’t communicate with Transformation Hub because of routing or network issues.
The specified Event Topic (`th-<topicname>`) is not valid	You are specifying an incorrect or non-existent the topic name.

This process is explained in more detail in the Logger Administrator's Guide, available from the Micro Focus support community.

Configuring ESM as a Transformation Hub Consumer

This procedure describes how to configure ESM as a Transformation Hub consumer with client authentication using a User (intermediate) certificate.

Azure Integration - Additional Steps

For proper integration with Azure Transformation Hub, you must perform the following additional procedures for the ArcSight product (ArcMC, SmartConnector, CTH, Logger, or ESM) you are integrating.

- Edits to the `/etc/hosts` file
- Peering
- Configure health probes and load-balancing rules for ports 32080, 9093, and 9092. (For ESM, this applies only to port 9093.)

Each of these is detailed below.

Edit the `/etc/hosts` File

You must add each instance to the product's `/etc/hosts` file.

1. Open the product's `/etc/hosts` file in a text editor.
2. Add the internal IP and FQDN for each instance in the Azure Kubernetes service. (The instance IP and FQDN can be obtained by opening the AKS resource group and then opening the aks-nodepool virtual machine scale set.)
3. Save the changes to the file.

Example additions to `/etc/hosts:`

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
```

```
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

```
10.1.1.4 aks-nodepool1-12400006-vmss000000
```

```
10.1.1.5 aks-nodepool1-12400006-vmss000001
```

```
10.1.1.6 aks-nodepool1-12400006-vmss000002
```

Peering

If the Azure product and Azure Transformation Hub are on different VLANs, you must configure peering between the two VLANs. An example is provided in the section Peering Virtual Networks.

Health Probes

You must configure health probes and load balancing rules for ports 32080, 9093, and, optionally, 9092. Configuring health probes for these ports is detailed in Configuring Port Rules for Product Integration. Use the same procedure as outlined for port 443.

> Some of the commands shown here will require `root` user privileges.

AWS Integration - Additional Steps

For proper integration with AWS Transformation Hub, you must perform the following additional procedures for the ArcSight product (SmartConnectors, Logger, or ESM) you are integrating.

For configuring SmartConnectors, Loggers, and ESMs: obtain the cluster worker node (Kafka broker node) host names, using one of the following procedures:

1. From the bastion host, run the following command:
   ```
   # kubectl get nodes
   ```
2. Copy the node host names.

OR

1. In the AWS UI, go to your Auto Scaling Group.
2. Click on the **Instance Management** tab to see the instance IDs.
3. Click the first instance ID to view the details of the corresponding instance.
4. Note the Private DNS name for the instance.
5. Repeat steps 3-4 for each instance ID.

After determining these values, ensure that your SmartConnector, Logger, or ESM instance is added to the the AWS TH cluster Security Group with rules allowing access to the ports 32080, 9093, and optionally 9092.

You can then follow the integration procedures below.

To configure ESM as a Transformation Hub consumer:

1. On Transformation Hub, run the command:

```
   # ${K8S_HOME}/scripts/cdf-updateRE.sh write --re-key={path to intermediate
certificate}/intermediate.key.pem --re-crt={path to intermediate
certificate}/intermediate.cert.pem --re-ca={path to intermediate
certificate}/ca.cert.pem
```

2. On an ESM host which has not been configured as a Transformation Hub consumer, switch to the manager directory:
   `# cd /opt/arcsight/manager`
3. Run each of these commands, one at a time. When prompted by the keytool for a password, enter the ESM password.

```
   # touch config/client.properties
```

```
   # bin/arcsight keytool -store clientkeys -storepasswd -storepass ""
```

```
   # bin/arcsight keytool -store clientkeys -keypasswd -keypass "" -alias
services-cn
```

```
   # bin/arcsight changepassword -f config/client.properties -p
ssl.keystore.password
```

4. Import the intermediate certificates to the ESM client keystore.
5. Run these commands:

```
   # bin/arcsight keytool -store clientcerts -importcert -file
/tmp/ca.cert.pem -alias thcert
```

```
   # bin/arcsight keytool -store clientkeys -importcert -file
/tmp/intermediate.cert.pem -alias thintcert
```

```
   # bin/arcsight keytool -store clientcerts -importcert -file
/tmp/intermediate.cert.pem -alias thintcert
```

```
   # /etc/init.d/arcsight_services stop manager
```

```
   # bin/arcsight keytool -store clientkeys -genkeypair -dname "cn=<your
CN>,ou=<your OU>, o=<your org short name>, c=<your country>" -keyalg rsa -
keysize 2048 -alias th -startdate -1d -validity 366
```

```
   # bin/arcsight keytool -certreq -store clientkeys -alias th -file
thkey.csr
```

6. Generate a certificate signing request (`.csr` file) so the Transformation Hub can sign a client certificate.

7. Copy the `.csr` file to the Transformation Hub initial master node.
8. On the Transformation Hub Initial Master Node, run the command:

```
# openssl x509 -req -CA /opt/intermediate_cert_files/intermediate.cert.pem -
CAkey /opt/intermediate_cert_files/intermediate.key.pem -in /opt/thkey.csr -
out /opt/signedTHkey.crt -days 3650 -CAcreateserial -sha256
```

9. Copy the signed certificate to `/tmp` on the ESM host.
10. On the ESM host import the signed client certificate into the client keystore so it can be used to authenticate to Transformation Hub. Run these commands:

```
# /opt/arcsight/manager/bin/arcsight keytool -store clientkeys -alias th -
importcert -file /tmp/signedTHkey.crt -trustcacerts
```

11. Start the manager configuration wizard:

```
# /opt/arcsight/manager/bin/arcsight managersetup
```

> If on a host without X Window access, run the `managersetup` command with the -i parameter. Consult the ESM documentation for more information regarding the `managersetup` command.

9. Proceed through the wizard for adding the Transformation hub to the ESM, until the dialog is displayed that prompts for a connection to Transformation Hub. On the dialog, under **"ESM can consume events from a Transformation Hub…"**, enter *Yes*. Then enter then the following parameters. (This will put an entry in the Manager `cacerts` file, displayed as `ebcaroot`):

Host:Port(s): `th-broker-hostname1:9093,th-broker-hostname2:9093,th-broker-hostname3:9093`

> **Note:** You must use host names, not IP addresses. In addition, ESM does not support non-TLS port 9092.

Topics to read from: `th-binary_esm` and Avro topics.

Path to Transformation Hub root cert:{leave this empty}

8. On the ESM, restart the ESM Manager:

```
   # /etc/init.d/arcsight_services stop manager
```

```
   # /etc/init.d/arcsight_services start manager
```

Understanding How Data is Produced and Consumed

Transformation Hub's publish-subscribe messaging system uses SmartConnectors and Collectors to produce event data, and supports Logger, Recon, and ESM, as well as Apache Hadoop and other third-party consumers.

While Transformation Hub can support a very high event flow (millions of events per second), the event rate for each producer and consumer will generally be much smaller (tens of thousands of events per second). Actual event flow will depend on your specific implementation and tuning applied, as well as server resources available, such as memory and CPU.

Producing Events with SmartConnectors

SmartConnectors can publish events to Transformation Hub topics. In order to publish events, you must configure your SmartConnectors to use the Transformation Hub destination. To send events to multiple topics, you can configure multiple concurrent destinations with the same Transformation Hub using different topics.

Once configured with a Transformation Hub destination, the SmartConnector sends events to Transformation Hub's Kafka cluster, which can then further distribute events to real-time analysis and data warehousing systems. Other applications, including Recon, ESM, Logger, and any third-party application that supports retrieving data from Kafka can receive them, for example, Apache Hadoop.

Transformation Hub balances incoming events between nodes, by distributing them evenly between the partitions in the configured topic.

Acknowledgments ("acks") ensure that Transformation Hub has received the event before the SmartConnector removes it from its local queue. You can disable acknowledgments, require acknowledgment only from the primary replica, or require every replica to acknowledge the event. (Acknowledgments do not indicate that consumers, such as Logger, have received the event data, only that Transformation Hub itself has.)

> Performance impact due to leader acks is a known Kafka behavior. Exact details of the impact will depend on your specific configuration, but could reduce the event rate by half or more.

Supported SmartConnector versions encode their own IP address as meta-data in the Kafka message for consumers that require that information such as Logger Device Groups.

- For information on supported SmartConnector versions, see the *SODP Support Matrix.*
- For more information about SmartConnectors and how to configure a Transformation Hub destination, refer to the CEF Destinations chapter of the *SmartConnector User's Guide*.

Micro Focus documentation is available for download from the Micro Focus support community.

Consuming Events with ESM

ESM agents are the consumers for Transformation Hub's publish-subscribe messaging system. An ESM agent can connect to Transformation Hub and consume all events in binary or Avro format for the topics to which it is subscribed.

Additionally, ESM provides data monitors to monitor Transformation Hub health.

- For information on supported versions of ESM and SmartConnectors, see the *SODP Support Matrix.*
- For instructions on configuring a supported version of ESM as a consumer, see the *ESM Administrator's Guide.*

Consuming Events with Logger

To subscribe to Transformation Hub topics with Logger, you must configure a receiver on a supported Logger version to receive the Transformation Hub events. Logger's Transformation Hub receivers are consumers for Transformation Hub's publish-subscribe messaging system. They receive events in Common Event Format (CEF) from Transformation Hub topics. A Logger Transformation Hub receiver connects to Transformation Hub and consumes all events for the topics it subscribes to.

When configuring a Logger Transformation Hub receiver, specify the worker node FQDNs, topics to consume from, and consumer group name. You can configure multiple Loggers to consume from the same topic as a part of a consumer group.

For more information about Logger and how to configure a Transformation Hub receiver, refer to the *Logger Administrator's Guide,* available for download from the Micro Focus support community.

> Kafka consumers can take up to 24 hours for the broker nodes to balance the partitions among the consumers. Check the Transformation Hub Kafka Manager **Consumers** page to confirm all consumers are consuming from the topic.

Sending Transformation Hub Data to Logger

For a Logger to be able to consume Transformation Hub events, the Logger must have a Transformation Hub receiver configured with the Transformation Hub worker nodes, consumer group, and event topic list. SmartConnectors that send data to Transformation Hub must have a Transformation Hub destination.

A group of Loggers, called a pool, can be configured to receive and distribute events between themselves. This works similarly to the Logger pool created by using the Logger Smart Message Pool destination on SmartConnectors. The difference is that when the SmartConnectors have a Logger Smart Message Pool destination, the event load is balanced by each SmartConnector, but when the SmartConnectors have a Transformation Hub destination, the event load is balanced by the Loggers.

Additional Loggers can be added to the pool simply by configuring the same Transformation Hub worker nodes, consumer group, and event topic list in the new Logger's Transformation Hub receivers, without having to reconfigure either the existing Loggers or any SmartConnectors.

The events retrieved by the Logger pool are distributed among the Loggers in the pool. If one Logger is down, new events are rebalanced among existing Loggers. When a Logger is added or removed from the Consumer Group, the event load is distributed across the pool of Loggers.

To send events from a group of SmartConnectors to a pool of Loggers, configure their Transformation Hub destinations to send events to the topic from which the Logger pool is consuming.

To configure Logger to subscribe to event data from specific SmartConnectors, you can do either of the following:

- Configure all the SmartConnectors to publish events to the same topic. Configure the Logger's Transformation Hub receiver to subscribe to this event topic.
- Configure each SmartConnector to publish events to different topics and then configure the Transformation Hub receiver on the Logger to subscribe to multiple event topics.

> ✓ **Tip:** Loggers in the same Logger pool do not consume the same events, since they are in the same Consumer Group. In high availability situations, you need events to be stored on two different Loggers. To store the same events on two Loggers, configure the Loggers to have different Consumer Group names, but subscribe them to the same event topic.

The number of Loggers in a Logger pool is restricted by the number of event topic partitions configured on the Container Deployment Foundation. For example, if there are only five partitions configured, only five Loggers will receive the events. If you have more than five Loggers configured in the same Consumer Group, some Loggers will not normally receive events, but will be available as hot spares. When adding receivers, be sure to increase the number of event topic partitions. See Managing Topics for more information.

Procedure to Send Transformation Hub Data to Logger

1. Configure the SmartConnector:
   - Set up a SmartConnector to publish to a particular Transformation Hub topic. Connectors can only send to a single topic for each destination. Additional destinations need to be configured if each event needs to go to multiple topics. Note the number of partitions in the topic.
   - For more information about SmartConnectors and how to configure a Transformation Hub destination, refer to the CEF Destinations chapter of the SmartConnector User's Guide, available for download from the Micro Focus support community.
2. Configure Logger:
   - Create a Transformation Hub receiver on each Logger in the Logger pool.
   - Configure each receiver to subscribe to the topics to which the SmartConnectors are publishing data. To subscribe to multiple topics, indicate the topics by specifying them in the Event Topic List parameter (a list of comma-separated values) while configuring the Transformation Hub receiver.
   - Configure each receiver to be in the same Consumer Group.

Example Setup with Multiple Loggers in a Pool

You can set up your Logger pools to subscribe to events from a particular device type, such as "Firewall." To do this, you would:

1. In ArcMC, create a Kafka topic named *Firewall*.
2. Configure all the SmartConnectors that handle firewall events to publish these events to topic "Firewall."
3. Configure the Loggers in the Logger pool:
   - Create a Transformation Hub Receiver on each Logger in the pool.
   - Configure the receivers to subscribe to the event topic “Firewall,” and include them in the “Logger_Firewall” Consumer Group.

Once the configuration is set up properly, the Logger pool will subscribe to device type *Firewall*.

> This example assumes that the Transformation Hub is being managed by an ArcSight Management Center for topic creation. Topics can also be managed through the Kafka Manager UI.

Consuming Events with Third-Party Applications

Transformation Hub is designed with support for third-party tools. You can create a standard Kafka consumer and configure it to subscribe to Transformation Hub topics. By doing this you can pull Transformation Hub events into your own data lake.

> Custom consumers must use Kafka client libraries of version 0.11 or later.

- All Transformation Hub nodes, consumers, and producers must be properly configured for forward and reverse DNS lookup, and be time-synchronized, using a time server such as NTP.
- Events are sent in standard CEF (CEF text) and binary (exclusively for ESM consumption). Any software application that can consume from Kafka and understand CEF text can process events.
- You can set up multiple consumer groups, and each group will get a copy of every event. Therefore you can have Logger and Apache Hadoop configured to consume from the same topic and each will get a copy of every event. This enables fanning out multiple copies of events without reconfiguring SmartConnectors or using additional CPU or network resources for them.

Consuming Transformation Hub Events with Apache Hadoop

Apache Hadoop is a software framework that enables the distributed processing of large data sets across clusters of computers. You can send Transformation Hub events to Hadoop by using Apache Flume.

This section describes how to set up the Apache Flume agent to transfer Common Event Format (CEF) events from an Transformation Hub Kafka cluster to Hadoop Distributed File System (HDFS).

Architecture for Kafka to Hadoop Data Transfer

Apache Flume uses a source module to read a Kafka topic containing CEF events, and it then transfers the events using a memory channel, and persists them to HDFS using a sink module. The CEF files are stored on HDFS by time, in a year/month/day/hour directory structure.

Using Apache Flume to Transfer Events to Hadoop

One of the applications you could use to transfer Transformation Hub events into your data lake is Apache Flume. Flume is designed to push data from many sources to the various storage systems in the Hadoop ecosystem, such as HDFS and HBase. This section describes how to use

Apache Flume as a data transfer channel to transfer events from Transformation Hub to Apache Hadoop or other storage systems.

Prerequisites

- Transformation Hub installed: Consult the *Micro Focus Transformation Hub Deployment Guide*.
- Flume installed: For information on how to install and configure Flume, refer to the Flume documentation.
- Storage system installed: Refer to your storage system documentation.

Procedure

Flume is controlled by an agent configuration file. You must configure Transformation Hub as the source agent, your storage system as the sink agent, and ZooKeeper as the channel agent in this file.

To configure Transformation Hub as the source:

Edit the agent configuration file to include the required properties, as in the table below. Configure other properties as needed for your environment.

Required Kafka Source Configuration

Property	Description
`type`	Set to `org.apache.flume.source.kafka.KafkaSource`.
`topic`	The Event Topic from which this source reads messages. Flume supports only one topic per source.

To configure the sink:

The required configuration varies. Refer to the Flume documentation for details on your storage system. The section Consuming Events with Apache Flume provides an example of how to configure Apache Hadoop as the sink.

Setting Up Flume to Connect with Hadoop

In the simplest deployment model, you need to deploy the Apache Flume agent on a Hadoop node server to pull events, and send them to Hadoop Distributed File System (HDFS).

Hadoop must be installed before you can connect it with Flume. If you do not already have your own Hadoop deployment, you can deploy Hadoop on a Red Hat Enterprise Linux 7.2 host. For more information, see Setting Up Hadoop.

For a detailed discussion of connecting Apache Flume with Hadoop, consult the Apache online documentation.

Sample Flume Configuration File

Before starting Apache Flume, create a configuration file based on the template below.

The configuration file should reside in `bin/flume/conf/`. This file is called *kafka.conf* in our example. You can name your own configuration file whatever is appropriate.

```
####################################################
#Sample Flume/Kafka configuration file
####################################################
#defines Kafka Source, Channel, and Destination aliases
tier1.sources = source1
tier1.channels = channel1
tier1.sinks = sink1
#Kafka source configuration
tier1.sources.source1.type = org.apache.flume.source.kafka.KafkaSource
tier1.sources.source1.kafka.bootstrap.servers= kafkaIP1:9092, kafkaIP2:9092,…
tier1.sources.source1.kafka.topics = th-cef
tier1.sources.source1.kafka.consumer.group.id = flume
tier1.sources.source1.channels = channel1
tier1.sources.source1.interceptors = i1
tier1.sources.source1.interceptors.i1.type = timestamp
tier1.sources.source1.kafka.consumer.timeout.ms = 150
tier1.sources.source1.kafka.consumer.batchsize = 100
#Kafka Channel configuration
tier1.channels.channel1.type = memory
tier1.channels.channel1.capacity = 10000
tier1.channels.channel1.transactionCapacity = 1000
#Kafka Sink (destination) configuration
```

```
tier1.sinks.sink1.type = hdfs

tier1.sinks.sink1.channel = channel1

tier1.sinks.sink1.hdfs.path = hdfs://localhost:9000/opt/\

hadoop/cefEvents/year=%y/month=%m/day=%d

tier1.sinks.sink1.hdfs.rollInterval = 360

tier1.sinks.sink1.hdfs.rollSize = 0

tier1.sinks.sink1.hdfs.rollCount = 0

tier1.sinks.sink1.hdfs.fileType = DataStream

tier1.sinks.sink1.hdfs.filePrefix = cefEvents

tier1.sinks.sink1.hdfs.fileSuffix = .cef

tier1.sinks.sink1.hdfs.batchSize = 100

tier1.sinks.sink1.hdfs.timeZone = UTC
```

Setting Up Hadoop

This is an overview of the steps necessary to install Apache Hadoop 2.7.2 and set up a one-node cluster. For more information, refer to the Hadoop documentation for your version.

To install Hadoop:

1. Be sure that your environment meets the operating system and Java prerequisites for Hadoop.
2. Add a user named 'hadoop'.
3. Download and unpack Hadoop.
4. Configure Hadoop for pseudo-distributed operation.
   - Set the environment variables.
   - Set up passphraseless SSH.
   - Optionally, set up Yarn. (You will not need Yarn if you want to use Hadoop only storage and not for processing.)
   - Edit the Hadoop configuration files to set up a core location, a Hadoop Distributed File System (HDFS) location, a replication value, a NameNode and a DataNode.
   - Format the Name node.
5. Start the Hadoop server using the tools provided.

6. Access Hadoop Services in a browser and login as the user "hadoop".
7. Execute the following commands to create the Hadoop cefEvents directory:
   ```
   hadoop fs -mkdir /opt
   hadoop fs -mkdir /opt/hadoop
   hadoop fs -mkdir /opt/hadoop/cefEvents
   ```
8. Execute the following commands to grant permissions for Apache Flume to write to this HDFS
   ```
   hadoop fs -chmod 777 -R /opt/hadoop
   hadoop fs -ls
   ```
9. Execute the following command to check Hadoop system status:
   ```
   hadoop dfsadmin -report
   ```
10. Execute the following command to view the files transferred by Flume to Hadoop.
   ```
   hadoop fs -ls -R /
   ```

Connectors in Transformation Hub (CTH)

To reduce the computational overhead and workload on a syslog SmartConnector infrastructure, you can make use of Connectors in Transformation Hub (CTH) instead.

CTH Functionality

Operationally, Micro Focus SmartConnectors hold two main responsibilities:

- **Collection**: A SmartConnector collects data from various sources.
- **Processing**: A SmartConnector processes the collected data into enriched security event data and posts them to a destination.

With CTH, the two functions of SmartConnector are handled in a slightly different manner. CTH takes advantage of the massive scalability of the robust Transformation Hub streaming architecture by moving the computationally intensive processing step directly to Transformation Hub.

- **Collection:** The collection step is performed by a dedicated Collector component, which gathers raw syslog data and publishes it to a dedicated syslog topic in Transformation Hub. As the name suggests, a Collector is a lightweight component responsible solely for collecting syslog data and passing it along to a dedicated CTH topic. A Collector is deployed on a VM or server using ArcMC.
- **Processing:** The CTH component reads the data from the Collector destination, and then parses, normalizes, enriches, and filters this data. It posts the data to a dedicated Transformation Hub topic for availability to any desired consumer. CTHs are deployed as Kubernetes pods within the CDF infrastructure.

CTH includes the majority of the functionality of ArcSight syslog connectors, except for data collection, which is handled by the lightweight Collector component instead. For more information about CTH configuration, consult the ArcSight Syslog Connector User Guide.

Advantages of CTH

CTH has the following advantages over traditional SmartConnector architecture.

- Hardware consolidation in the data collection layer where Collectors are deployed, due to the logical separation of collection and processing. A single data feed from a Collector can replace multiple SmartConnector feeds.
- Improved stability, easy horizontal scalability, and improved load balancing as data flows increase with time or fluctuate during operations.
- Ease of deployment, since CTHs are deployed with a single click in the ArcMC management console.
- Raw syslog data is now available in the CTH topic and can be shared with any desired consumer.
- CTH supports FIPS mode.

Limitations of CTH

- CTH presently supports the processing of syslog data only.
- Upgrades to CTH are performed by upgrading Transformation Hub, rather than by upgrading CTH itself.

Deploying and Managing CTH

Installation and management of CTH is performed on a managed Transformation Hub though the ArcMC management console. Consult the ArcMC Administrator’s Guide for instructions on how to deploy and manage CTH.

Destination Topics

Collectors should only be configured with the `th-syslog` topic as a destination (and no other destinations).

Valid routing topic destinations for CTH include the following:

- `th-cef`
- `th-binary_esm`
- `th-cef-other`
- `mf-event-cef-esmfiltered`
- `th-arcsight-avro`
- `mf-event-avro-esmfiltered`

In addition, custom CTH source and destination topics might be configured on Transformation Hub. (Custom topics might only be created for CEF data.)

Collector/CTH Supported Security Modes

Collector destinations can support the following security modes:

- Plain text (no security mode selected)
- FIPS only
- TLS only

Collector security mode can be set during Instant Deployment in the ArcMC console. See the ArcMC Administrator's Guide for more information.

CTH source and destinations can support the following security modes:

- TLS + Client Authentication (default setting)
- FIPS + Client Authentication (automatically set when enabling FIPS mode in Transformation Hub.
- Plain text (no security mode selected)
- TLS only
- FIPS only

If desired, CTH's plain text, TLS-only, and FIPS-only modes can be set in ArcMC after deployment.

Configuring Consumers and Producers for High Availability

Configure the Transformation Hub Kafka cluster endpoint to avoid single points of failure in both the producers sending data to Transformation Hub (such as SmartConnectors), and the consumers subscribing to data from the Transformation Hub (such as Logger and ESM).

For Producers

Configure the **Initial Host:Port(s)** parameter field in the Transformation Hub Destination to include all Kafka broker (worker) nodes as a comma-separated list.

Provide all Kafka broker (worker) nodes for a producer and a consumer configuration to avoid a single point of failure. For example, broker_hostname1:9093, broker_hostname2:9093, broker_hostname3:9093.

For more information about how Kafka handles this using bootstrap.servers, see the kafka documentation here.

For Consumers

Configure the **Transformation Hub host(s) and port** parameter field in the Receiver to include all Kafka cluster nodes as a comma-separated list.

For more information about how Kafka handles this using bootstrap servers, click here.

Understanding Data Compression

Transformation Hub compression settings affect data in two general places, communication and storage. Specifically, this refers to data stored on disk, in Kafka topic partitions, and data that is in transit.

- All external producers such as connectors, collectors, and internal producers, like routing and CEF2Avro processors, compress data before sending it.
- For data in transit, data compression is controlled by the producer's configuration.

Data Consumers

There is no property that controls data compression on consumers. Consumers read metadata from each message, which indicates the correct decompression algorithm to use. Since this is evaluated on a message-by-message basis, the consumer's behavior does not depend on which topic it is consuming from. A single topic might contain messages which have been compressed with different compression algorithms (also referred to as compression types or codecs).

Data Storage (Data at Rest)

The algorithm used to compress stored data is determined by the topic configuration. All Transformation Hub topics, except `th-arcsight-avro,` currently use the default compression type, which is the same as that used by `producer`. This configuration choice means the topic will retain the original compression algorithm set by the producer. By leaving this as producer-defined, there is flexibility for the producer to send either compressed (using any supported codec) or uncompressed data.

The `th-arcsight-avro` topic is an exception because the database scheduler reads from this topic, but does not yet have support for reading messages encoded with the ZStandard (zstd) compression algorithm. Therefore, there is a specific, out-of-the-box value for this topic, to insure that the database scheduler can read it, no matter what over-the-wire compression was used.

Topic	Compression Type	Transformation Hub Version
All topics except `th-arcsight-avro`	producer (default)	3.4.0 and earlier
`th-arcsight-avro`	gzip	3.4.0 and 3.3.0
`th-arcsight-avro`	uncompressed	3.2.0 and earlier

Configuring Compression

There are two places in the Kafka architecture where compression can be configured: the producer and the topic.

- Producer-level compression is set on the producer; for example, in SmartConnector Transformation Hub destination parameters. For producers that reside inside TH, such as routing and stream processors, the compression algorithm is configured on the Transformation Hub configuration page, during deployment.
- Topic-level compression can be set with Kafka Manager (using **Topic > Update Config Menu**); however, it is strongly recommended that settings be left at default values.

Compression Types

While Kafka supports a handful of compression types, Transformation Hub implements only two types: gzip and zstd.

- **gzip:** By default, gzip is used for Transformation Hub routing and stream processors, as well as for SmartConnectors. This is for backward compatibility and might change in a future release.
- **zstd:** Testing has shown that zstd uses less bandwidth, storage, and CPU resources than gzip. For bandwidth constrained networks, higher EPS is typically seen when using zstd; however actual results are unique to each environment. Third-party Java producers should use kafka-clients version 2.1.0 or later, for zstd support. ArcSight consumers compatible with zstd include Logger 7.0, ESM 7.2, IDI 1.1, or later.

Pushing JKS files from ArcMC

You can push JKS (Java Keystore) files to multiple managed SmartConnectors in ArcMC. First, you will upload the files to a file repository in ArcMC, and then push them out to their destination SmartConnectors. You must then configure and enable the Kafka destination on all SmartConnectors.

To upload the Java Keystore files:

1. Prepare the .jks files you want to push and store them in a secure network location.
2. In ArcMC, click **Administration > Repositories > New Repository.**
3. In **Name, Display Name**, and **Item Display Name**, enter KAFKA_JKS
4. Enter other required details as needed, and then click **Save**.
5. Click **Upload to Repository.**
6. Follow the prompts in the upload wizard and browse to the first .jks file. Make sure to choose the individual file option.
7. Upload as many files as needed by repeating the upload wizard.

To push the files to multiple SmartConnectors:

1. In ArcMC, browse to the file repository for the `.jks` files.
2. Click the **Upload** arrow.
3. Follow the prompts in the wizard and select your destination SmartConnectors.
4. The files are pushed to the managed SmartConnectors and stored in the designated SmartConnector folder.

To configure the Kafka destination on all SmartConnectors:

In ArcMC, click **Node Management > Connectors** tab.

1. Select the SmartConnectors to be configured.
2. Choose **Add a destination** and pick the Kafka destination type.
3. Add the destination details along with the `.jks` path and password, and save the changes.

Integrating Intelligence with ESM

To enable ESM to receive the analysed entities and alerts information from Intelligence, you need to install and configure the ArcSight REST FlexConnectors.

The REST FlexConnector provides a configurable method to collect events from Intelligence and send them to ESM. Intelligence’s Alerts and Entities APIs serve as the REST API endpoints from which the REST FlexConnectors collect data.

The REST FlexConnectors use the OAuth2 authentication to get permission to receive events from Intelligence. The events collected by the FlexConnectors are in JSON format.

With the help of one JSON parser file each for Alert data and Entities data, these events are converted into a format that can be understood and received by ESM.

Using the JSON Parser Files

The parser file that is used for alerts data is **alerts.jsonparser.properties**.

```
trigger.node.location=/data
token.count=14
token[0].name=alertId
token[0].type=String
token[0].location=alertId

token[1].name=datasource
token[1].type=String
token[1].location=datasource

token[2].name=alertTime
token[2].type=Long
token[2].location=timestamp

token[3].name=risk
token[3].type=Integer
token[3].location=risk

token[4].name=contribution
token[4].type=Integer
token[4].location=contribution

token[5].name=significance
token[5].type=String
token[5].location=significance

token[6].name=threat
token[6].type=String
token[6].location=templates/threat

token[7].name=family
token[7].type=String
#token[7].format=__uri()
token[7].location=templates/family

token[8].name=teaser
token[8].type=String
token[8].location=templates/teaser

token[9].name=alert
```

```
token[9].type=String
token[9].location=templates/alert

token[10].name=anomalyTypes
token[10].type=String
token[10].location=anomalyTypes

token[11].name=numAnomalies
token[11].type=Integer
token[11].location=numAnomalies

token[12].name=category
token[12].type=String
token[12].location=category

token[13].name=scrollId
token[13].type=String
token[13].location=/scrollId

#(End Of Token Definitions)

#tokens

event.externalId=alertId
event.deviceCustomNumber1=risk
event.deviceCustomNumber1Label=__stringConstant("RiskScore")
event.deviceCustomNumber2=contribution
event.deviceCustomNumber2Label=__stringConstant("Contribution")
event.deviceCustomNumber3=__safeToLong(__regexToken(alert,.?risk=([^\\s]+)
.*))
event.deviceCustomNumber3Label=__stringConstant("Entity Risk Score")

event.fileName=__regexToken(alert,.?entity name="([^"]+)".*)
event.fileHash=__regexToken(alert,.?hash="([^"]+)".*)
event.fileType=__regexToken(alert,.?type="([^"]+)".*)

event.message=alert
event.reason=teaser
event.aggregatedEventCount=numAnomalies
event.deviceEventCategory=category
event.deviceReceiptTime=__createLocalTimeStampFromSecondsSinceEpoch
(alertTime)

#tags
#event.destinationUserId=id
#event.deviceCustomString5=tags
#event.destinationUserName=otherName
#event.deviceCustomString2=source
```

```
#event.message=desc

#Other Mappings
event.name=family
event.deviceEventClassId=threat
event.deviceVendor=__stringConstant("Micro Focus")
event.deviceProduct=__stringConstant("Interset")
event.deviceSeverity=significance

#Agent Severity
severity.map.veryhigh.if.deviceSeverity=9,10
severity.map.high.if.deviceSeverity=7,8
severity.map.medium.if.deviceSeverity=4,5,6
severity.map.low.if.deviceSeverity=2,3
severity.map.verylow.if.deviceSeverity=0,1

#Conditional mappings
conditionalmap.count=1

conditionalmap[0].field=event.fileType
conditionalmap[0].mappings.count=3

conditionalmap[0].mappings[0].values=user
conditionalmap[0].mappings[0].event.destinationUserName=__regexToken
(alert,.?entity name="([^"]+)".*)

conditionalmap[0].mappings[1].values=ip
conditionalmap[0].mappings[1].event.destinationAddress=__
regexTokenAsAddress(alert,.?entity name="([^"]+)".*)

conditionalmap[0].mappings[2].values=machine
conditionalmap[0].mappings[2].event.destinationHostName=__regexToken
(alert,.?entity name="([^"]+)".*)
```

The parser file that is used for entities data is **entities.jsonparser.properties**.

```
trigger.node.location=/data

token.count=12

token[0].name=entityHash
token[0].type=String
token[0].location=entityHash

token[1].name=entityType
token[1].type=String
token[1].location=entityType
```

```
token[2].name=entityName
token[2].type=String
token[2].location=entityName

token[3].name=risk
token[3].type=Integer
token[3].location=risk

token[4].name=riskChange
token[4].type=Integer
token[4].location=riskChange

token[5].name=storyCount
token[5].type=Integer
token[5].location=storyCount

token[6].name=lastActivity
token[6].type=String
token[6].location=lastActivity

token[7].name=tags
token[7].type=String
token[7].format=__uri()
token[7].location=tags

token[8].name=otherName
token[8].type=String
token[8].location=../../tags/name

token[9].name=source
token[9].type=String
token[9].location=../source

token[10].name=desc
token[10].type=String
token[10].location=../tags/description

token[11].name=scrollId
token[11].type=String
token[11].location=/scrollId

#(End Of Token Definitions)

#tokens

event.fileHash=entityHash
event.fileType=entityType
event.fileName=entityName
event.deviceCustomNumber1=risk
event.deviceCustomNumber1Label=__stringConstant("RiskScore")
event.deviceCustomNumber2=riskChange
event.deviceCustomNumber2Label=__stringConstant("RiskChange")
event.deviceCustomString3=lastActivity
event.deviceCustomString3Label=__stringConstant("LastActivity")
#event.deviceCustomDate1=lastActivity
#__parseMutableTimeStampSilently(start)

#tags
#event.destinationUserId=id
event.deviceCustomString5=tags
#event.destinationUserName=otherName
#event.deviceCustomString2=source
#event.message=desc

#nextUrl?
event.deviceCustomString6=scrollId

#Other Mappings
event.name=__stringConstant("Interset Risky User Information")
event.deviceEventClassId=__stringConstant("IRU")
event.deviceVendor=__stringConstant("Micro Focus")
event.deviceProduct=__stringConstant("Interset")
event.deviceSeverity=2

#Agent Severity
severity.map.low.if.deviceSeverity=2
```

Installing and Configuring the FlexConnectors

You need to install two REST FlexConnectors: one to collect and parse the Alerts data, and another to collect and parse the Entities data.

Prerequisites

Complete the following steps before you begin with the REST FlexConnector installation and configuration:

1. Create the `OAuth2.properties` file for using the OAuth2 authentication with Intelligence as follows and save it in the desired location (For example, `C:\Users\Administrator\Desktop\`):

   ```
   client_id= <The Client ID field value present in the FUSION tab in the
   ```

```
CDF Management Portal>
client_secret=<The Client Secret field value present in the FUSION tab
in the CDF Management Portal>
redirect_uri=http://localhost:8081/oauth2callback
auth_url=https://<FQDN of ArcSight Platform Virtual IP for HA or single
master node>/osp/a/default/auth/oauth2/grant
token_url=https://<FQDN of ArcSight Platform Virtual IP for HA or
single master node>/osp/a/default/auth/oauth2/grant
scope=
timestamp_format_of_api_vendor=
```

2. Do the following to register the callback URL in OSP. The callback URL is the URL where the OSP directs the user after a successful authentication.

   a. Launch a terminal session and log in to the node where NFS is present.

   b. Change to the following directory:

   ```
   cd <NFS_root_DIRECTORY>/arcsight-volume/sso/default/WEB-
   INF/conf/current/default/services/
   ```

   c. Execute the following command to open the authcfg.xml:

   ```
   vi authcfg.xml
   ```

   d. Add the following within `<RedirectUrlList></RedirectUrlList>`:

   `http://localhost:8081/oauth2callback`

   e. Execute the following commands to restart OSP by deleting the fusion-single-sign-on container:

   ```
   kubectl get pods --all-namespaces|grep osp
   kubectl delete pod <hercules-osp-xxxxxxxxxx-xxxxx> -n <arcsight-
   installer-xxxxx>
   ```

Install and Configure the REST FlexConnector

To install and configure a REST FlexConnector, see ArcSight FlexConnector REST Developer Guide.

Ensure the following when you install and configure the REST FlexConnector:

- Select **ArcSight FlexConnector REST** as the **Connector Type**.
- When adding the parameters information, specify the following:
  - For the Configuration File field, specify only alerts if the FlexConnector is for collecting and parsing alerts data, else specify only entities if the FlexConnector is for collecting and

parsing entities data.

- For the **Events URL** field, specify **https: //<ip address or hostname of Intelligence>/interset/api/search/0/alerts?sort=timestamp&sortOrder=desc&riskSort =maximum** if the FlexConnector is for collecting and parsing alerts data, else specify **https: //<ip address or hostname of Intelligence>/interset/api/search/0/topRisky?count=100** if the FlexConnector is for collecting and parsing entities data.
- For the **Authentication Type** field, select **OAuth2**.
- For the **OAuth2 Client Properties File** field, browse to the location where you have created and saved the **OAuth2.properties** file, then select the file.

- Import the OSP Certificate in the REST FlexConnector.
- When configuring the destination, select either ArcSight Manager (encrypted) or Transformation Hub as the destination. For more information, see SmartConnector User Guide. When adding the parameters information, specify the following if you have selected Transformation Hub as the destination:
  - For the **Content type** field, select **ESM**.
  - For the **Topic (hover for recommendations)** field, specify either **th-binary_esm** or avro topics.
  - For the **For ESM topic**, **the ESM version** field, select **7.2.x** or above versions.

Importing the OSP Certificate in the REST FlexConnector

To import the OSP certificate in the REST FlexConnector:

1. Launch a terminal session and log in to any of the Kubernetes nodes.
2. Execute the following command:

   ```
   kubect exec -it th-kafka -n <namespace> bash
   ```

3. Navigate to the following directory where the `issue_ca.crt` certificate file is present. This certificate is the OSP Issuer Certificate (CA).

   ```
   cd /vault-crt/RE
   ```

4. Copy the contents of the `issue_ca.crt` file in a new file, name the file as `issue_ca.cer`, and save it in the desired location (for example, C:\Users\<user_name>\Desktop\).
5. Do the following to import the OSP CA certificate to the FlexConnector truststore cacerts:
   a. Open a command window and navigate to the following location:

      ```
      cd $ARCSIGHT_HOME\current\jre\bin\
      ```

   b. Execute the following command:

```
keytool -importcert -file "<location_of_issue_ca.cer>\issue_ca.cer" -
keystore "$ARCSIGHT_
HOME\current\jre\lib\security\cacerts" -storepass changeit
```

c. When you run this command, you are prompted to provide your input for the following message: “Trust this certificate [no]:” Specify Yes.

Performing FlexConnector Post-Installation Tasks

After you install and configure the FlexConnector and before you run the FlexConnector, copy the desired JSON parser files in the **ARCSIGHT_HOME\user\agent\ flexagent** location.

Installing ESM and Configuring Transformation Hub with ESM

Installing ESM

To install ESM and ArcSight Console to leverage Intelligence entities and alerts information, see Installation Guide for ESM.

Configuring Transformation Hub with ESM

To configure Transformation Hub with ESM, see Configuring ESM as a Transformation Hub Consumer

Sending Data to Transformation Hub From Intelligence

Perform the following steps to start the SmartConnector agent so that it can send the entities and alerts information from Intelligence to the configured topic.

1. Navigate to:

   ```
   cd $ARCSIGHT_HOME\current\bin\
   ```

2. Execute the following command:

   ```
   ./arcsight agents
   ```

Viewing the Intelligence Entities and Alerts Information in the ArcSight Console

Perform the following steps to view the Intelligence entities and alerts information in the ArcSight Console:

1. Download the **Interset_Sample_Content.arb.zip** file from the Micro Focus Marketplace. and save it in a desired location (For example, C:/Desktop/Interset_Sample_Content.arb.zip).
2. Log in to the ArcSight Console.
3. Click the **Packages** tab in the left pane, then click **Import**.
4. Browse to location of the **Interset_Sample_Content.arb.zip** file.
5. Click **Install**. The installation process starts.
6. After the installation is successful, click the **Resources** tab in the left pane.
7. Navigate to **Active Channels** > **Shared** > **All Active Channels** > **Interset**.
8. Double-click **Interset** or **Interset Anomalies** to view the Intelligence entities and alerts information.
9. Navigate to **Dashboards** > **Shared** > **All Dashboards** > **Interset**.
10. Double-click **Interset Overview** to view a summary of the Intelligence entities and alerts information.

Integrating SOAR with ESM

SOAR integrates with ESM to log and forward detailed reporting on every single incident to facilitate prioritization and investigation of alerts as well as the remediation of incidents.

SOAR ingests correlated events from ESM and converts them into an alert. When an alert is generated, a new incident is created on SOAR's Incident Management Service Desk. Analyst can then investigate the incident and take remedial actions.

The ESM and SOAR integrations presents following capabilities to:

- Ingest Correlated Alerts
- Retrieve Base Events
- Create Case
- Update Case
- Search Cases

- Get Case Details
- Query Active List
- Add Entries to Active List
- Delete Entries from Active List

The bidirectional integration of ESM and SOAR requires configuration at both the platforms.

Understanding the Prerequisites for ESM and SOAR Integration

Complete following steps before you begin the ESM and SOAR integration:

- Network traffic from ESM to SOAR towards port 32200/tcp must be allowed. Arcsight SOAR listener for correlated event data (alert) is accessible from this reverse proxy port.
- Port 8443/tcp must be open at ESM to allow HTTPS traffic. SOAR connects with the ESM REST APIs on this port.
- A user account for SOAR must be configured to connect with ESM API.
- Enable the parameter **ArcSightListenerEnabled** on SOAR at **Configuration** > **Parameters** before configuring the forwarding destination on connector.

  > If the parameter **ArcSightListenerEnabled** is not enabled, an error message is displayed as destination not reachable.

- To forward the correlation events, a forwarding connector must be installed on ESM.

  To install a forwarding connector:

1. **Create a forwarding connector**: To create a forwarding connector, see Create Forwarding Connector and Forward Correlation Events. For this example, you can create the forwarding connector for ESM and SOAR integration, with following values:
   - User ID: forwardATAR
   - User Type: Forwarding Connector
2. **Install and configure the forwarding connector package**: Install the forwarding connector package on ESM. Then complete the following steps for configuration:

   To configure forwarding connector:

   Run the following command:

   ```
   $ cd /opt/arcsight/MicroFocus_
   ArcsightSmartConnectors/SuperConnector/current/bin (directory $
   ./runagentsetup.sh
   ```

   To add localconnector cert:

a. Run the following command:

   For ArcSight Home:

   ```
   /opt/arcsight/MicroFocus_
   ArcsightSmartConnectors/SuperConnector/current
   ```

   For JAVA_HOME:

   ```
   /opt/arcsight/MicroFocus_
   ArcsightSmartConnectors/SuperConnector/current
   ```

b. In the **Conector Setup Wizard**, select the **Add a Connector** option, then the **ArcSight Forwarding Connector (Enhanced)** option to configure the connector as a forwarding connector.

c. Enter the parameter details as follows:

   - **ArcSight Source Manager Host Name[localhost]**: <Specify local host IP>
   - **ArcSight Source Manager Port**: 8443
   - **ArcSight Source Manager User Name**: <Specify the user name that you have created for ESM>
   - **ArcSight Source Manager Password**: <Specify the password that you have created>

d. Select **Yes**, if the values are correct.

To configure forwarding connector for forwarding events from ESM to SOAR:

a. To setup ArcSight Agent, run the following command:

   For ArcSight Home:

   ```
   /opt/arcsight/MicroFocus_ArcsightSmartConnectors/SuperConnect
   ```

   For JAVA_HOME:

   ```
   /opt/arcsight/MicroFocus_ArcsightSmartConnectors/SuperConnector/
   ```

b. In the **Connector Setup Wizard**, select **Modify Connector** > **Modify connector parameters** > **Add destination options**, then select **CEF Syslog** option as the type of destination.

c. Specify the parameter details as follows:

   - **IP/Host**: < Specify ArcSight Platform FQDN corresponding to the Virtual IP address provided during installation for HA or, for a single-master installation, the IP address of the master node>
   - **Port**: 32200
   - **Protocol**: Raw TCP

- **Forwarder**: False
- **ArcSight Source Manager Password**: <Specify the password that you have created>

d. Select **Yes**, if the values are correct.

Configuring ESM for Integration

The SOAR and ESM integration requires configuration on ESM. To ingest data, you must create an active list on ESM and configure the rules to forward events to this list. The rules define the type of event that is forwarded to SOAR for investigation. After the active list is added and the rule is configured, SOAR monitors the events from ESM, and creates respective alerts.

To configure ESM for Integration:

1. Log in to ArcSight Console.
2. Create a new active list with name **ATAR Rule Name List**.
3. Add the rule names to the newly created **ATAR Rule Name List**.
4. Create a **Pre-persistence** rule on ArcSight Console. To process and forward alerts to SOAR, you must create the Pre-persistence rule with following conditions:

   a. Select the forwarding connector user that you have created, as the owner of this rule.

      For example, to assign the forwarding connector user **forwardATAR**, as the owner of the **Pre-persistence** rule, navigate to **Inspect/Edit** window. Click **Attributes** tab of the **Rule:ATAR Integration Rule**. Under **Assign** tab, specify **forwardATAR, admin** as **Owner**.

   b. Set **Action** for this rule to add a key value to event data before sending the data to SOAR.

      For example, to add a key value to event data, before sending the data to SOAR, navigate to the **Rule:ATAR Integration Rule** of the **Inspect/Edit** window.

      i. Click the **Conditions** tab. In **Edit** tab, click **event1** below the **Event conditions**.

      ii. Click **& AND**. Set **Type = Correlation** and **InActiveList("/All Active Lists/Public/ATAR/ATAR Rule Names)**.

      iii. Click **Actions** tab and select **On Every Event [Active]** option.

      iv. Click **Set Event Field Actions** and set **oldFileHash** = <**some_random_string**>.

5. Create a web user account on ArcSight Console, with following details:

   Login

   - **User ID**: atarapi
   - **User Type**: Web User

User

- **Last Name**: API Access
- **First Name**: ATAR

This user account enables SOAR’s access to ESM’s REST API.

6. Set permission to read all potential base events triggering correlations to the web user.
   a. Navigate to **Filter:FetchBaseEventsFilter** window and click **Filter** tab.
   b. In the **Edit** tab, click **Event conditions**. Click **{} Event** and set **Type =Base**.
7. Add a filter as **ATAR Filter**.
   a. Navigate the **Inspect/Edit** window and click **Filter: ATARFilter** tab. Click **Filter** tab and go to **Edit**.
   b. Click **Event conditions** and go to **{} Event 1**. Click **& AND** and set **Type = Correlation**. Set **oldFileHash** = <**some_random_string**>.
8. Add the ATAR Filter to **ACL**.
   a. Navigate to **ACL Editor [/All Users/Custom User Groups/ATAR]** window and click the **Events** tab.
   b. Select **Filter** in the **Resource** field on top of the window and click **Add**.

Configuring SOAR For Integration

The ESM and SOAR integration requires some configurations at SOAR. A web user account is created at SOAR to connect with ESM. This user account is used to read, write and access the active list at ESM. This web user created at SOAR is also responsible for accessing all of the required events, including the base events in ESM. To listen to the events, ESM is configured as an alert source on SOAR. After ESM is configured as alert source, SOAR can pull the events from ESM and convert them into alerts for investigation purpose.

Configuring Credential

To support ESM and SOAR integration, a web user account must be created on SOAR to communicate with ESM. This user account is used by SOAR to fetch and/or update events and invoke other supported actions.

To configure credentials on SOAR:

1. Navigate to **Configuration** > **Credentials** on SOAR.
2. Click **+Create Credential** to view the **Credential Editor** window.
3. Enter the following values in the **Credential Editor** window:

a. **For Internal Credential:**
   - **Type**: Internal credential
   - **Name**: <Display name of credential set>

     For example: ArcSight ESM Credentials
   - **Username**: <Web user created for SOAR on ESM>
   - **ESM Password**: <Password of the user created for SOAR on ESM>
   - **Private Key**: <Empty>

b. **For Credential Store:**
   - **Type**: External credential
   - **Name**: <Name of the credential with pull path of the safe on store>

Configuring ESM As Alert Source

The active list on ESM has correlated events that is passed to SOAR. The SOAR then converts these events to alerts and performs investigation and response procedures. To receive alerts on SOAR, ESM must be configured as an alert source to SOAR.

To configure ESM as Alert Source on SOAR :

1. Navigate to **Configuration** > **Alert Source** on SOAR.
2. Click **Create Alert Source Configuration** and enter the following values in the **Create Alert Source Configuration** window:
   - **Name**:< Display name of ESM Alert Source on SOAR >
   - **Type**: Micro Focus ArcSight ESM
   - **Address**: <Address of the ESM Manager>

     For example, you can specify address of the ESM Manager as https://192.168.5.5:8443).
   - **Key**: <Specify the name of the key that you have defined in Pre-persistence rule definition>
   - **Allowed IP Addresses**: <Specify the IP addresses of the ESM Manager and CDF container’s gateway>

     Any data not originating from these IP addresses is discarded by SOAR listener.
   - **Alert Severity**: <Specify the alert severity values mapping, with SOAR incident severity>
   - **Configuration**: Specify the following parameter:

Parameter Name	Parameter Description	Parameter Usage
CEF field **[severity.field]**	Used as severity value when mapping severity value to SOAR incident severity. You can set this parameter for priority, severity, flexString1 and flexNumber 1	severity.field=priority
CEF-extension **[severity.field]**	Used as rule name value.	
Scope fields: **[src]**	a. The value of scope field is extracted from correlated event. src:NETWORK_ ADDRESS:OFFENDER, dst:NETWORK_ ADDRESS:IMPACT, request:URL:OFFENDE R fields are always extracted by default. b. This parameter can also specify additional fields to be extracted:	a. (field1:CATEGORY:ROLE, (field2:CATEGORY:ROLE, …) CATEGORY is any EMAIL_ADDRESS, HASH, HOST, MAC_ADDRESS, NETWORK_ADDRESS, COMPUTER_ NAME, UNKNOWN, URL, USERNAME, PROCESS ROLE is any OFFENDER, IMPACT, RELATED b. correlated.scope=s_ user:USERNAME:OFFENDER, dvc:NETWORK_ADDRESS:RELATED correlated.scope=src:NETWORK_ ADDRESS:OFFENDER, dst:NETWORK_ ADDRESS:IMPACT, request:URL:OFFENDER

Parameter Name	Parameter Description	Parameter Usage
Additional Scope Field: **[baseevent.scope]**	These valuse are extracted from base events (field1:CATEGORY:ROLE) and use JSON pointer notation. See the correlated.scope property for Category and Role values details. This parameter can specify additional fields to be extracted, and will not override the default behavior.	Example: baseevent.scope=/device/address:NETWORK_ADDRESS:RELATED # baseevent.scope=
[cache.reusing.duration]	Used to configure how far (in minutes) into the past this enrichment is checked.	cache.reusing.duration=20
enable/disable **[enable.baseevent.activity]**	Used to enable/disable base events activity in the incident timeline.	enable.baseevent.activity=false

3. Click **Test** to test the integration. A **Test Alert Source** pop up is displayed to confirm that you have entered the valid credentials and address.
4. Click **Save** to complete the ESM and SOAR integration.
5. Navigate to **Configuration** > **Parameters** and set the value of **ArcSightListenerEnabled** to **true**.

Configuring ESM as Integration

ESM must be configured on the SOAR as an integration. This integration seamlessly maps the incoming ESM correlated events into SOAR alerts.

To configure ESM as Integration:

1. Navigate to **Configuration** > **Integrations** on SOAR.
2. Click **+Create Integration** to view the **Configuration** window.
3. Enter the following values in the **Configuration** window:
   - **Name**: <Display name of ESM integration on SOAR>
   - **Type**: Micro Focus ArcSight ESM
   - **Address**: <Address of the ESM Manager>

For example, you can specify the address of the ESM Manager as: https://192.168.5.5:8443

- **Configuration**: #proxy.id=5422
- **Credential**: <Name of the credential set created>

  For example, ArcSight ESM Credentials
- **Trust Invalid SSL Certificates**: <Select this option if server certificate is self-signed or not recognized by browsers>
- **Require Approval From**: <Select users from list that can provide approval before executing actions on this integration>
- **Notify**: <Select users to be notified when SOAR performs an action on this integration>

4. Click **Test** to test the integration. A **Test Alert Source** pop up is displayed to confirm that you have entered the valid credentials and address.
5. Click **Save** to complete the integration.

Tuning ESM and SOAR Integration

The ESM and SOAR integration can be customized as per your requirements. Following parameter values can be tuned to suit your environment:

Consult with ArcSight SOAR Field Engineering Team if tuning is required.

Parameter Name	Parameter Description	Default Value
ArcSightAutoEnrichEnabled	Enable ArcSight auto-enrichment with base-event data	False
ArcSightListenerEnabled	Enable Arcsight Listener	False
ArcSightListenerKeyField	ArcSight listener key field for alert source identification	oldFileHash
ArcSightListenerPort	Arcsight listenet port	9090
ArcSightListenerProtocol	ArcSight listener protocol	tcp
ArcSightListenerThreadPoolCoreSize	ArcSight listener thread pool core pool size (0 = unlimited)	0

Parameter Name	Parameter Description	Default Value
ArcSightListenerThreadPoolKeepAlive	ArcSight listener thread pool keep-alive seconds (ignored if core pool size = 0)	60
ArcSightListenerThreadPoolMaxSize	ArcSight listener thread pool maximum size (ignored if core pool size = 0)	20
ArcSightListenerThreadPoolQueueCapacity	ArcSight listener thread pool queue capacity (ignored if core pool size = 0)	1000

Upgrading Your Environment

This section provides information about upgrading.

Checklist: Upgrading Your Environment

Follow the steps listed below to ensure a successful upgrade.

	Task	See
❑	1. Download the installation packages.	"Downloading the Installation Packages" below
❑	2. Stop the Event Ingestion during the upgrade.	"Stopping Event Ingestion During the Upgrade" on page 387
❑	3. Upgrade the database.	"Upgrading the Database" on page 390
❑	4. Upgrade CDF.	"Upgrading CDF " on page 391
❑	5. Upgrade deployed capabilities.	"Upgrading Deployed Capabilities" on page 399
❑	6. Start the Event Ingestion.	"Starting Event Ingestion" on page 402

Downloading the Installation Packages

You can use this procedure for an initial install and upgrade.

Follow the "Checklist: Upgrading Your Environment" on the previous page to ensure a successful upgrade.

To download the packages:

1. Launch a terminal session and log in to the primary master node as `root`.

   > If you select to install as a sudo user, log in to the primary master node as the non-root user.

2. In the ArcSight Platform release notes, *"Downloading and Installing the ArcSight Platform Capabilities section"* identify and access the files to download into a directory.
3. Unzip `cdf-2020.08.00153-x.x.x.x.zip` into a directory, which we'll refer to going forward as `{unzipped-cdf-dir}`.

   > Do not unzip under `/root` or any sub directory of it.

4. Move the ArcSight Metadata file into the `{unzipped-cdf-dir}/arcsight/metadata/` directory.

   > Do not untar the file. The filename must have the prefix `arcsight-installer-metadata`. Also, do not move signature files as it might cause warnings and errors from the installer script. Therefore, only copy ONLY.tar files you need based on your YAML file.

5. For each ArcSight product to install, move the corresponding image tar file into the `{unzipped-cdf-dir}/arcsight/images/` directory.

   > Do not untar the file. Also, do not move signature files as it might cause warnings and errors from the installer script. Therefore, only copy ONLY.tar files you need based on your YAML file.

   For example, if you deploy Fusion, Recon, SOAR, and Transformation Hub:

ArcSight Installer	`arcsight-installer-metadata-x.x.x.x.tar`
Fusion	`fusion-x.x.x.x.tar`
Recon	`recon-x.x.x.x.tar`
SOAR	`soar-x.x.x.x.tar`
Transformation Hub	`transformationhub-x.x.x.x.tar`

Stopping Event Ingestion During the Upgrade

This step is required only when you have deployed Transformation Hub.

Follow the "Checklist: Upgrading Your Environment" on page 385 to ensure a successful upgrade.

As part of the process, you must stop the event ingestion temporarily during the upgrade while the event schema is updated.

Ensure to perform these steps in a timely manner so that producers caches can hold up on what they are ingesting but yet not delivering to Transformation Hub.

	Task	See
☐	1. Stop all Avro event producers.	"Stopping CEF-to-Avro Producers" below
☐	2. Drain the Avro event queue.	"Draining the Avro Event Queue" on the next page
☐	3. Resetting the offset record for Avro topics.	"Resetting the Offset Record for Avro Topics" on the next page

Stopping CEF-to-Avro Producers

To stop CEF-to-Avro producer:

1. Browse to the management portal.

   ```
   https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443
   ```

2. Click **DEPLOYMENT**, and select **Deployments.**

3. Click the **Three Dots** ⋮ (Browse) on the far right and choose **Reconfigure**, under **Transformation Hub > Stream Processors and Routers** perform the following actions:

4. Note down the value of **# of CEF-to-Avro Stream Processor instances to start**, and then change it to 0.
5. Click **SAVE**.
6. Use kafka Manager to check `th-arcsight-avro` topic.
7. Wait for the value of **Produce Message/Sec** to become 0.

Draining the Avro Event Queue

To monitor that consumers have drained their Avro event queue:

1. If you deployed Database, perform the following steps to monitor the database.
   a. From the Database cluster node1, change to the `/opt/arcsight-database` directory.

      ```
      cd /opt/arcsight-database
      ```

   b. Run the following command to monitor database EPS ingestion.

      ```
      ./kafka_scheduler events
      ```

   c. Check the output of **Event Copy Status for (th-arcsight-avro)** topic.
   d. Wait until the **end_reason** field displays `END_OF_STREAM`.
2. If you deployed ESM or Intelligence, monitor those consumers as well.

Resetting the Offset Record for Avro Topics

Because the event schema has changed for the current release, perform the following steps for each:

- Avro topics
- th-arcsight-avro
- mf-event-avro-esmfiltered
- And others

To reset the offset record:

1. Log in to the pod:

   ```
   kubectl exec th-kafka-0 -n arcsight-installer-xyzab -it bin/bash
   ```

> The `arcsight-installer-xyzab` value changes across deployments so you need the correct name. You can check the name with kubectl get pods commands and using the namespaces of the th-kafka-n pod.

2. Navigate to the location of the `kafka-delete-records.sh` script.

```
cd /usr/bin
```

3. Generate the json file that carries the partitions metadata.

   If Kafka scheduler SSL was disabled, use port 9092:

```
./kafka-run-class kafka.tools.GetOffsetShell --broker-list
server.example.com:9092,server.example.com:9092,server.example.com:9092 --topic th-
arcsight-avro | sed -re 's/(.*):(.*):(.*)/{"topic": "\1", "partition": \2,
"offset": \-1},/'>>/tmp/test.json
```

   If Kafka scheduler SSL was enabled, use port 9093:

```
./kafka-run-class kafka.tools.GetOffsetShell --broker-list
server.example.com:9092,server.example.com:9092,server.example.com:9092 --topic th-
arcsight-avro | sed -re 's/(.*):(.*):(.*)/{"topic": "\1", "partition": \2,
"offset": \-1},/'>>/tmp/test.json
```

4. Reset the offset to the latest record (for consumers to start fresh).

   If Kafka scheduler SSL was disabled, use port 9092:

```
./kafka-delete-records --bootstrap-server
server.example.com:9092,server.example.com:9092,server.example.com:9092 --offset-
json-file /tmp/test.json
```

   If Kafka scheduler SSL was enabled, use port 9093:

```
./kafka-delete-records --bootstrap-server
server.example.com:9093,server.example.com:9093,server.example.com:9093 --offset-
json-file /tmp/test.json
```

5. Ensure the following two command outputs match. If the output of the two commands match, the topic offsets are reset.

   If Kafka scheduler SSL was disabled, use port 9092:

```
#The following command shows the low watermark offset
./kafka-run-class kafka.tools.GetOffsetShell --broker-list
server.example.com:9092,server.example.com:9092,server.example.com:9092 --topic th-
arcsight-avro --time -2
#The following command shows the high watermark offset
./kafka-run-class kafka.tools.GetOffsetShell --broker-list
server.example.com:9092,server.example.com:9092,server.example.com:9092 --topic th-
arcsight-avro --time -1
```

   If Kafka scheduler SSL was enabled, use port 9093:

```
#The following command shows the low watermark offset
./kafka-run-class kafka.tools.GetOffsetShell --broker-list
server.example.com:9093,server.example.com:9093,server.example.com:9093 --topic th-
arcsight-avro --time -2
#The following command shows the high watermark offset
./kafka-run-class kafka.tools.GetOffsetShell --broker-list
server.example.com:9093,server.example.com:9093,server.example.com:9093 --topic th-
arcsight-avro --time -1
```

6. Now you are ready to upgrade Transformation Hub.

Upgrading the Database

This procedure is only required when you have a deployed ArcSight Database and need to upgrade it.

The upgrade process is irreversible, make sure to backup the database. Also, be patient as the Database upgrade might take time to complete. The Database might need time to create indexes and complete upgrade tasks. The Database upgrade might appear to be complete; however, if you start the product before the Database upgrade is complete, you might experience errors and performance issues.

Follow the "Checklist: Upgrading Your Environment" on page 385 to ensure a successful upgrade.

Performing the Database Upgrade

To upgrade the ArcSight Database:

1. Log in to the master node where you downloaded the files.
2. Copy the `{unzipped-cdf-dir}/arcsight/database/db-installer_x.x.x-x.tar.gz` file to the Database cluster node 1.
3. Log in to Database cluster node 1.
4. Create a directory to extract the `db-installer_x.x.x-x.tar.gz` file into. We will refer to this directory as `{unzipped-db-installer-dir}`.

   Do not use the directories `/root`, `/opt/vertica`, or the existing database installer directory (default is `/opt/arcsight-database`). The files in `/opt/arcsight-database` will be upgraded by the database upgrade tool.

5. Change to the directory.

   ```
   cd {unzipped-db-installer-dir}
   ```

6. Extract (untar) the `db-installer_x.x.x-x.tar.gz` file into the directory using the following command:

   ```
   tar xvfz  db-installer_x.x.x-x.tar.gz
   ```

7. Execute the following command to start the upgrade.

   ```
   ./db_upgrade -c upgrade-utilities
   ```

   The output of the command will look similar to the following:

   ```
   Upgrade related changes cannot be rolled back, do you want to continue
   with the upgrade (Y/N): y
   Starting upgrade...
   ********************* Start of Database Upgrade ******************
   Enter previous installed location (/opt/install-db):/opt/arcsight-
   database
   …
   ********* Start of Database Upgrade to x.x.x *********
   Pre Upgrade Check for DB Event_vx.x.x Schema
   DB will be upgraded to Event_vx.x.x Schema
   Create event quality table and create event quality crontab ...
   event quality table has been created successfully.
   Upgrading schema ...
   …
   Schema has been upgraded successfully.
   Version specific upgrade methods
   ********************* Database Upgraded Complete. Version is x.x.x
   ******************
   ```

8. Run.

   ```
   ./db_upgrade -c upgrade-db-rpm
   ```

9. (Optional) Start firewall service.

Upgrading CDF

Follow the "Checklist: Upgrading Your Environment" on page 385 to ensure a successful upgrade.

As part of the process, you must upgrade CDF, the following options available.

- "Upgrading CDF Automatically" below
- "Upgrading CDF Manually" on page 394
- "Upgrading CDF BYOK on Azure" on page 395

We recommend using the automatic installation, as it is easy to use. However, if the automatic installation method does not met your needs, you can upgrade manually.

> Please note, if you installed your environment with theignore-swap flag before, swap space needs to be disabled before you start the upgrade. Otherwise, the upgrad will fail, with first master not starting up. Please refer to "Disabling Swap Space" on page 45.

Upgrading CDF Automatically

The automated upgrade of CDF is performed using a single command and requires no interaction until completion of each phase. Typically, each automated upgrade phase takes around 1 hour for a cluster with 3 master nodes and 3 worker nodes.

- "Preparing the Upgrade Manager " below
- "Configuring Passwordless Communication" on the next page
- "Downloading the Upgrade File " on the next page
- "Performing the CDF Automatic Upgrade" on page 394
- "Removing the Auto-upgrade Temporary Directory from UM" on page 394

Preparing the Upgrade Manager

Automatic upgrade should be run from a host (for purposes of these instructions, known as the upgrade manager).

The upgrade manager (UM) may be one of the following host types:

- One of the cluster nodes
- A host outside the cluster (a secure network location)

> The following uses the cluster master node1 as an example.

Configuring Passwordless Communication

You must configure passwordless SSH communication between the UM and all the nodes in the cluster.

To configure passwordless communication:

1. Run the following command on the UM to generate key pair.

   ```
   ssh-keygen -t rsa
   ```

2. Run the following command on the UM to copy the generated public key to every node of your cluster.

   ```
   ssh-copy-id -i ~/.ssh/id_rsa.pub root@<node_fqdn_or_ip>
   ```

Downloading the Upgrade File

Download the upgrade files for CDF to a download directory (referred to as `<download_directory>`) on the UM.

There are three directories involved in the auto-upgrade process:

1. An auto-upgrade directory `/tmp/autoUpgrade` will be auto generated on the UM. It will store the upgrade process steps and logs.
2. A backup directory `/tmp/CDF_202005_upgrade` will be auto generated on every node. (approximate size 1.5 GB)
3. A working directory will be auto generated on the UM and every node at the location provided by the `- d` parameter The upgrade package will be copied to this directory. (approximate size 9 GB). The directory will be automatically deleted after the upgrade.

   > The working directory can be created manually on UM and every node and then passed as -d parameter to the auto-upgrade script. If you are a non-root user on the nodes inside the cluster, make sure you have permission to this directory.

Performing the CDF Automatic Upgrade

To upgrade automatically:

1. Log in to the master node where you downloaded the upgrade files.
2. Change to directory.

   ```
   {unzipped-cdf-dir}
   ```

3. Run the following.

   ```
   ./autoUpgrade.sh -d /path/to/workinig_directory -n {any_cluster_node_adress_or_ip}
   ```

   For example:

   ```
   ./autoUpgrade.sh -d /tmp/upgrade -n yourdomain-masternode1.yourenterprise.net
   ```

Removing the Auto-upgrade Temporary Directory from UM

The auto-upgrade temporary directory contains the upgrade steps and logs.

To upgrade another cluster from the same UM, remove that directory using the following.

```
rm -rf /tmp/autoUpgrade
```

Upgrading CDF Manually

Beginning with the master node1, upgrade your CDF infrastructure on every node of the cluster.

To run the following process *on each node:*

1. Run the following.

   ```
   mkdir /tmp/upgrade-download
   ```

2. From the "Downloading the Installation Packages" on page 385 section, copy the CDF bits.

   ```
   cdf-2020.08.xxxx.zip to /tmp/upgrade-download
   ```

3. Unzip the upgrade package by running these commands.

   ```
   cd /tmp/upgrade-download
   unzip cdf-2020.08.xxxx.zip
   ```

4. Run the following commands on each node (follow this pattern: master1, master2, master3, to worker1, worker2, worker3, etc.).

   ```
   cd /tmp/upgrade-download/cdf-2020.08.xxxx
   ```

   ```
   ./upgrade.sh -i
   ```

5. On the initial master node1, run the following commands to upgrade CDF components.

   ```
   cd /tmp/upgrade-download/cdf-2020.08.xxxx
   ```

   ```
   ./upgrade.sh -u
   ```

6. Clean the unused docker images by running the following commands on all nodes (masters and workers). This can be executed simultaneously.

   ```
   cd /tmp/upgrade-download/cdf-2020.08.xxxx
   ```

   ```
   ./upgrade.sh -c
   ```

7. Verify the cluster status. First, check the CDF version on each node by running the command:

   ```
   cat ${K8S_HOME}/version.txt>> 2020.08.xxxx
   ```

8. Check the status of CDF on each node by running these commands.

   ```
   cd ${K8S_HOME}/bin
   ./kube-status.sh
   ```

Upgrading CDF BYOK on Azure

To upgrade your CDF BYOK infrastructure on Azure, use the following procedure.

Prerequisites

- You must be able to access the jumphost VM as root in the Azure cluster.
  You can perform all steps from jumphost. For more information, see "Preparing the Jump Host Virtual Machine" on page 101.
- The Kubernetes command-line tool (kubectl) must be installed and connected to your cluster on jumphost.

It is possible to perform the upgrade from another machine, but you must have kubectl connected to your cluster and proxy settings must be disabled.

To Upgrade CDF BYOK on Azure

1. Run the following command to ensure that all pods are running:

   ```
   kubectl get pods -A
   ```

2. From the "Downloading the Installation Packages" on page 385 section, copy the CDF deployer bits `cdf-deployer-2020.08.xxxx.zip` to `/tmp/upgrade-download`.
3. Unzip the deployer package by running these commands.

   ```
   cd /tmp/upgrade-download
   unzip cdf-2020.08.xxxx.zip
   ```

4. Upload new images to Azure Container Registry (ACR).
   a. Go to the Azure management portal and open ACR, and click **Access keys > Login Server**. A username and password is required to upload images. For more information, see "Uploading the Product Images to the ACR" on page 135.
   b. Change to the deployer scripts directory:

      ```
      cd cdf-deployer-2020.08.xxxx/scripts/
      ```

   c. Run `uploadimages.sh` with the credentials from ACR:

      ```
      ./uploadimages.sh -o <your-org-name> -r <login-server> -u <username> -
      p <password> -F /tmp/cdf-byok-images-<VERSION>.tar -c 4
      ...
      Upload completed in 1690 seconds.
      Upload-process successfully completed.
      ```

      > The -o argument for orgnamemust be the same as the one used for the original installation.You can check your orgname with the following by command: kubectl get cm -n core base-configmap -o yaml | grep REGISTRY_ORGNAME:For more information, see `uploadimages.sh --help`.

5. Run the upgrade using the following steps.
   a. Ensure all PODs in the core namespaces are **Running** or **Completed**.

      ```
      kubectl get pods -n core
      ```

      Example output:

      ```
      cdf-apiserver-7965dcf689-4qvkx                        2/2     Running
          0           145m
      fluentd-7q4dw                                         2/2     Running
          0           136m
      fluentd-kkf2p                                         2/2     Running
          0           136m
      ```

```
fluentd-mwqh8                                          2/2     Running
    0          136m
idm-77b4f9fbfb-cfwkg                                   2/2     Running
    0          136m
idm-77b4f9fbfb-g5pcb                                   2/2     Running
    0          136m
itom-cdf-deployer-2020.05-2.2-2.3-3.1-tncp8            0/1
Completed   0          137m
itom-cdf-deployer-xg6cw                                0/1
Completed   0          147m
itom-cdf-ingress-frontend-56c9987b7-bvrsn              2/2     Running
    0          145m
itom-cdf-ingress-frontend-56c9987b7-n8tbc              2/2     Running
    0          145m
itom-logrotate-deployment-6cf9546f8b-rbcvs             1/1     Running
    0          136m
itom-postgresql-default-77479dfbff-t87tv               2/2     Running
    0          137m
itom-vault-6f558dc6cc-bz52l                            1/1     Running
    0          146m
kubernetes-vault-67f8698568-csd54                      1/1     Running
    0          145m
mng-portal-7cfc584db5-hcmjf                            2/2     Running
    0          133m
nginx-ingress-controller-6f6d4c95b9-7fhbs              2/2     Running
    0          133m
nginx-ingress-controller-6f6d4c95b9-nv2zw              2/2     Running
    0          133m
suite-conf-pod-arcsight-installer-86c9687b69-kctjz     2/2     Running
    0          132m
suite-db-68bfc4fbd5-v6nvm                              2/2     Running
    0          145m
suite-installer-frontend-6f49f88797-msb7j              2/2     Running
    0          145m
```

b. Change to the to deployer directory:

```
cd cdf-deployer-2020.08.xxxx/
```

c. Run the upgrade process :

```
./upgrade.sh -u
```

Example output:

```
***********************************************************************
*************
WARNING: This step is used to upgrade CDF components to 2020.08
release.
```

```
The upgrade process is irreversible. You can NOT roll back.
Make sure that all nodes in your cluster are in Ready status.
Make sure that all Pods and Services are Running.


**********************************************************************
*************
Do you want to continue (Y/N): Y

** Pre-checking before upgrade ...

** Prerequisite tasks for components upgrade... (Step 1/3)
Checking CDF endpoints status before upgrade...
Checking helm2 deployment status ...
Setting BYOK environment values ...
Copying itom-cdf-alias.sh to /etc/profile.d/ ...

** Updating Kubernetes RBAC ... (Step 2/3)
RBAC update successfully.

** Configure and start the cdf-deployer ... (Step 3/3)
Creating resources from YAML: /home/jumphost/cdf-deployer-
2020.08.00153-20.11.0.631/objectdefs/itom-cdf-deployer-upgrade.yaml
Waiting for CDF components upgrade process complete ...
....................................................
CDF components upgrade process completed.
Successfully completed CDF components upgrade process.
```

d. At the end of upgrade, ensure all pods are **Running** or **Completed**:

```
kubectl get pods -A
```

6. Fix your Azure load balancing rules after the upgrade.
   The upgrade recreated resources where theload balancing rules ware mapped. You need recreate all the health probe and load balancing rules.

   a. Find IP assigned to your external access host for the CDF by pinging it from the jumphost:

   ```
   ping installer.arcsight.private.com
   PING installer.arcsight.private.com (10.1.1.101) 56(84) bytes of data.
   ```

   If you do not know your hostname, you can get it by command

   ```
   kubectl get cm -n core base-configmap -o yaml | grep EXTERNAL_
   ACCESS_HOST:
   ```

   b. Patch the load balancer service.
   For more information, see "Patching the Load Balancer" on page 140

```
kubectl patch services -n core itom-cdf-ingress-frontend-svc -p '
{"spec":{"type":"LoadBalancer","loadBalancerIP": "PUBLIC_IP"}}'
```

> Replace the placeholder `PUBLIC_IP` with the IP assigned to your external access host.

c. After successfully patching the service, continue with creating health probe and load balancer rules for port 5443 and 443. For more information, see "Configuring the Load Balancer" on page 145.

7. Upgrade your deployed capabilities.
   Upgrading your deployed capabilities is almost the same process as for on-premises installation except for the following differences:

   - Images are uploaded to Azure Container Registry (not to local running registry)
   - The upgrade is being performed from jumphost not from master node.

   For more information, see Upgrading Deployed Capabilities.

Upgrading Deployed Capabilities

Follow the "Checklist: Upgrading Your Environment" on page 385 to ensure a successful upgrade.

> To upgrade Azure, click here for the installation steps.

As part of the process, you must upgrade your deployed capabilities using the CDF Management Portal.

1. "To accept the certificate:" below
2. "To upgrade deployed capabilities:" on the next page

Accepting the Certificate

To accept the certificate:

1. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.
2. Click **DEPLOYMENT**, and select **Deployments.**

3. Click the **Three Dots** ⋮ (Browse) on the far right and choose **Reconfigure**.
4. Accept the certificate.

Upgrading Deployed Capabilities

To upgrade deployed capabilities:

1. Log in to the master node where you downloaded the upgrade files.
2. Change to the following directory.

   ```
   cd ${K8S_HOME}/scripts
   ```

3. Run the following commands to upload the images to the local Docker Registry. Use the `-F <image file>` option on the command line multiple times for each image to upload. Adjust the `-c 2` option up to half of your CPU cores in order to increase the speed of the upload.

   > You will be prompted for a password for the docker container registry-admin user. The registry-admin password is initially set to the same password as the admin user for the CDF Management Portal during installation when "Configuring and Running CDF" on page 70; however, later changing the CDF Management Portal admin password does not change the registry-admin password as it is managed separately.

   ```
   ./uploadimages.sh -c 2 -F {unzipped-cdf-dir}/arcsight/images/fusion-
   x.x.x.x.tar -F {unzipped-cdf-dir}/arcsight/images/recon-x.x.x.x.tar
   ```

4. Add new metadata.

   > Make sure to copy the `arcsight-installer-metadata-x.x.x.tar` to the system where your web browser is running before performing the process below.

5. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.

   a. Click **DEPLOYMENT > Metadata** and click **+ Add**.

   b. Select `arcsight-installer-metadata-x.x.x.xx.tar` from your system. The new metadata is added to the system.

6. Start the upgrade process.

   a. Go to **DEPLOYMENT > Deployments**. Notice the number **1** in the red circle in the Update column.

> Minor version changes do not display like regular updates. (For example: 20.11.0.15 -> 20.11.0.16.)

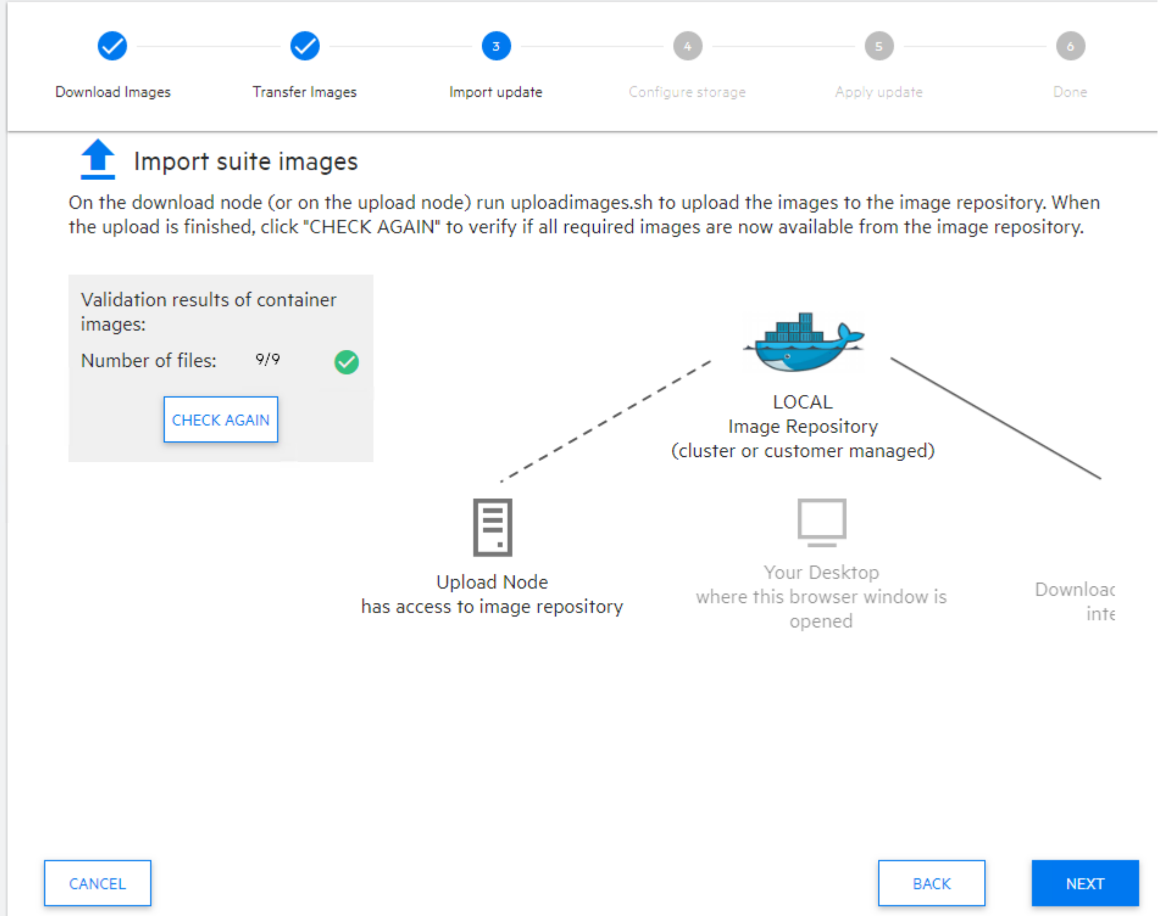
   b. Click the red circle and select your recently added metadata to initiate the upgrade.

7. From the **Update to** page, click **NEXT** until you reach the **Import suite images** page.

> When prompted to download or transfer images, you can simply click Next to skip the steps. You performed these steps earlier.

8. Ensure that the validation results of container images show a complete number of files.

> When you arrive at the Import suite images page, the images should already be imported, as you performed these steps earlier.



Import suite images

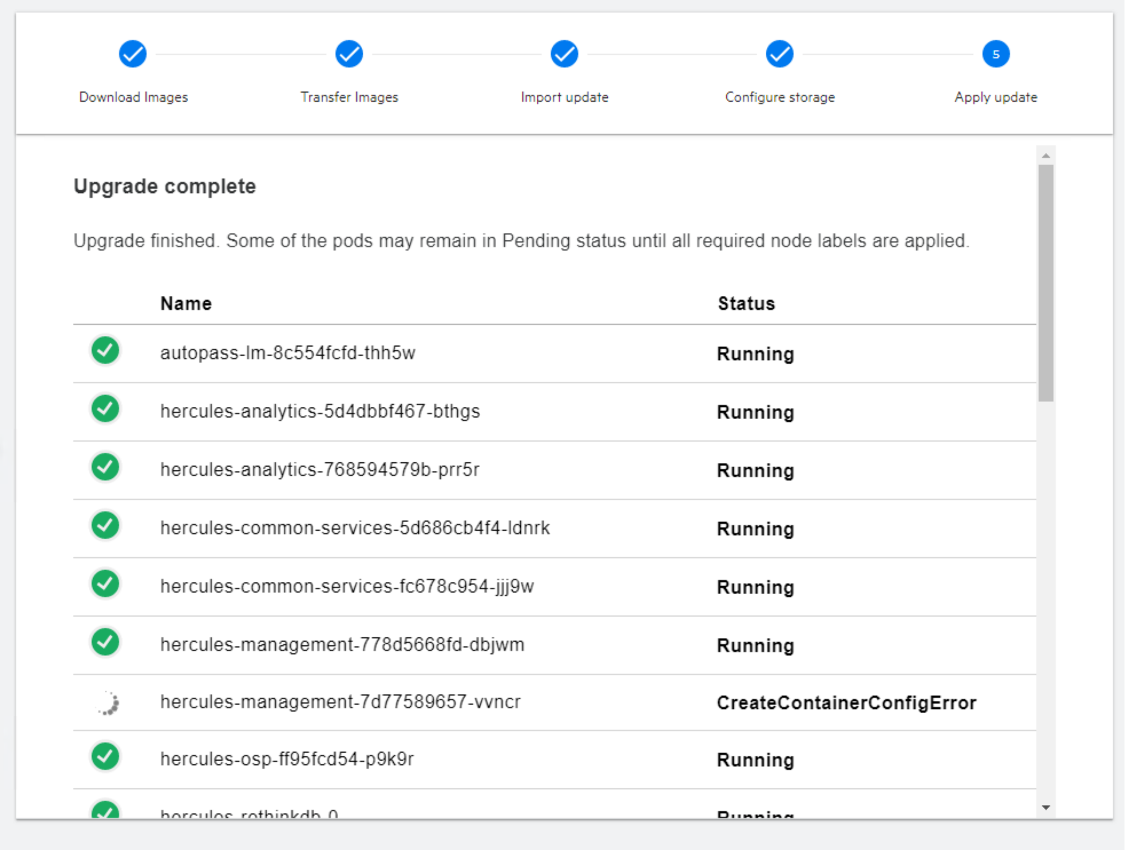
On the download node (or on the upload node) run uploadimages.sh to upload the images to the image repository. When the upload is finished, click "CHECK AGAIN" to verify if all required images are now available from the image repository.

Validation results of container images:

Number of files: 9/9 ✓

CHECK AGAIN
LOCAL
Image Repository
(cluster or customer managed)
Upload Node
has access to image repository
Your Desktop
where this browser window is opened
Downloac
inte
CANCEL
BACK
NEXT

9. Click **NEXT** until you reach the **Upgrade Complete** page.



Upgrade complete

Upgrade finished. Some of the pods may remain in Pending status until all required node labels are applied.

	Name	Status
✓	autopass-lm-8c554fcfd-thh5w	Running
✓	hercules-analytics-5d4dbbf467-bthgs	Running
✓	hercules-analytics-768594579b-prr5r	Running
✓	hercules-common-services-5d686cb4f4-ldnrk	Running
✓	hercules-common-services-fc678c954-jjj9w	Running
✓	hercules-management-778d5668fd-dbjwm	Running
	hercules-management-7d77589657-vvncr	CreateContainerConfigError
✓	hercules-osp-ff95fcd54-p9k9r	Running
✓	hercules-rethinkdb-0	Running

Starting Event Ingestion

This step is required only when you have deployed Transformation Hub.

First, follow the "Checklist: Upgrading Your Environment" on page 385 to ensure a successful upgrade. Secondly, follow the checklist below to start the event ingestion.

	Task	See
☐	1. (Conditional) If you deployed the database, recreate the database Avro event consumers.	"Recreating the Database Avro Event Consumers" on the next page

❑	2. Start all the Avro Event Producers.	"Starting the CEF-to-Avro Events Producer" on the next page SmartConnectors that are configured to send events in Avro format

Recreating the Database Avro Event Consumers

1. Log in to the database node1 as root and change to the /opt/arcsight-database directory:

   ```
   cd /opt/arcsight-database
   ```

2. Delete the existing Kafka scheduler because it needs to be recreated after upgrade.

   ```
   ./kafka_scheduler delete
   ```

3. Recreate the Kafka scheduler using the following command. Specify one or more Transformation Hub nodes in a comma separated list. For high availability, we recommend specifying at least three nodes.

   If Kafka scheduler SSL was disabled, use port 9092:

   ```
   ./kafka_scheduler create <Transformation_Hub_Node_1_IP>:9092,<Transformation_Hub_
   Node_2_IP>:9092,<Transformation_Hub_Node_3_IP>:9092
   ```

   If Kafka scheduler SSL was enabled, use port 9093:

   ```
   ./kafka_scheduler create <Transformation_Hub_Node_1_IP>:9093,<Transformation_Hub_
   Node_2_IP>:9093,<Transformation_Hub_Node_3_IP>:9093
   ```

4. Complete the Database setup.

   a. Log in to the database:

      ```
      # ./db_installer start-db
      ```

   b. Configure the schema registry server setting.

      ```
      /opt/arcsight/kubernetes/scripts/cdf-updateRE.sh > issue_ca.crt
      ```

   c. Use the file "issue_ca.crt" in the following command:

      ```
      ./schema_registry_setup <SCHEMA-REGISTRY-NODE-FQDN>
      /opt/arcsight/kubernetes/ssl/issue_ca.crt
      ```

For example:

```
# ./kafka_scheduler create 192.168.1.1:9092
```

d. Confirm the error message is not occuring:

```
./kafka_scheduler messages
```

Starting the CEF-to-Avro Events Producer

1. Browse to the management portal.

   ```
   https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443
   ```

2. Click **DEPLOYMENT**, then select **Deployments**.
3. Go to **Deployment > Deployments > Three dots** ⋮ **> Reconfigure**.
   a. Under **Transformation Hub > Stream** change the value of **# of CEF-to-Avro Stream Processor instances to start** back to its original number.
   b. Click **SAVE**.
4. From the Kafka Manager, monitor EPS to `th-arcsight-avro` is increasing, i.e. not 0 anymore.

Maintaining the Platform and Deployed Capabilities

This section describes maintaining platform capabilities.

Changing ArcSight Platform Configuration Properties

> ⚠ Reconfiguring properties causes the capabilities related to the property to stop and restart and this might cause operations underway to fail. Therefore, ensure that effected capabilities that cannot be easily retried are not running when you reconfigure any of these properties. For example, check the pod logs to see what operations are underway.

To change ArcSight Platform configuration properties:

1. In the CDF Management Portal, select **Deployment > Deployments.**
2. Click **...** (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
3. Update configuration properties as needed.
4. Click **Save.**

All services in the cluster affected by the configuration change will be restarted (in a rolling manner) across the cluster nodes.

Configuring Log Levels

You can configure the log level as desired for troubleshooting purposes.

To change the log level:

1. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.
2. Click **DEPLOYMENT**, and select **Deployments.**
3. Click the **Three Dots** ⋮ (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.

4. Under the appropriate capability tabs there are log level configuration options for each component. Select the appropriate value to update the Log Levels. The change goes into effect automatically.

Using REST APIs

User interfaces use REST APIs to manage and access data and configuration information. You can also access the APIs directly, if needed. For example, you might want to update a particular user’s dashboard.

> For SSO access to the REST APIs, specify the values for Client ID and Client Secret, in the Single Sign-on Configuration section.

Name	REST API Endpoint Documentation
Dashboard Metadata	`https://{master_FQDN or IP}/metadata/rest-api-docs`

Understanding License Keys

- "Transformation Hub License" below
- "Recon License" on the next page
- "Intelligence License" on page 408
- "SOAR License Check" on page 409
- "Verifying Expired License Keys" on page 409

Transformation Hub License

Transformation Hub ships with a 90-day instant-on evaluation license, which will enable functionality for 90 days after installation. In order for Transformation Hub to continue working past the initial evaluation period, you will need to apply a valid license key to Transformation Hub. A Transformation Hub license key, as well as a legacy ArcMC ADP license key, can be used for licensing Transformation Hub.

> To ensure continuity of functionality and event flow, make sure you apply your product license **before** the evaluation license has expired.

Recon License

This section explains the features, warnings and capacity of the Recon license.

Instant on License

Recon includes an instant on license for 90 days, after this license expires, you will not be able to use the product.

Installing a term or permanent license will overwrite the instant on license.

Moving Median Events per Second (MMEPS)

MMEPS is tracked every day at GTM+0 hours, even if the license is expired or removed.

MMEPS Calculation:

1. Calculate Events Per Day (EPD): Events Per Day is the total number of events ingested into database in a twenty-four hour period (for day #1 we calculate the EPD based from the time we install Product component short name until GTM+0 hours). The time frame is based on GTM+0 hours starting at 00:00:00 and ending at 23:59:59, regardless of any local times that might be in use.
2. Calculate Sustained EPS (SEPS): Sustained EPS is the “constant” Events Per Second that the system sustained within the twenty-four hour period(for day #1 we calculate the EPD based from the time we install Recon until GTM+0 hours). It normalizes peaks and valleys and gives a better indication of use. The formula used for this calculation is (EPD/ ((60*60)*24)).
3. Calculate last 45 days moving median (MMEPS): Utilizing the SEPS information recorded per day, a moving median EPS value will be identified. The Median value is calculated using last 45 day data set, and shifting the calculation window one day every twenty-four hours after the first 45 days. The official clock for calculation purposes is defined by GTM+0 hours starting at 00:00:00 to 23:59:59 regardless of local time.

Actual Calculation:

Day 1: MMEPS = SEPS of day 1

Day 2: MMEPS = AVG(SEPS of day 1 and 2)

Day 3 until last 45 days: MMEPS = median value of SEPS of day 1...45

Warnings

A warning message will be displayed in the following scenarios:

- Within thirty days before license expiration (term license or instant on license), you will receive a warning message after login indicating the license expiration date.
- Recon will be tracking EPS every twenty four hours after installation, or when a new license is installed after the previous one expired.
- If the current calculated MMEPS exceeds license EPS capacity then there will be a warning indicating that license EPS capacity has been exceeded.
- If there are many events in Transformation Hub, and data ingestion to database is higher than license EPS (an EPS exceed warning will be temporarily displayed until data ingestion rate normalizes).

If any of the following conditions are met you will be redirected to an invalid license page and won't be able to use the product:

- Instant on license expires.
- Term license expires.
- No license for Recon is present.

> In order to revert this issue, install a valid license.

License Capacity

If a term or permanent license is installed, it will automatically overwrite the instant on license. License capacity will not be cumulative in this case.

If multiple licenses are installed, (term or permanent), capacity will be cumulative. Expiration date will be determined by whichever license expires first.

License Cache Performance

In this release we cache the license for one hour, it will be generated when refreshing or navigating to a different page. If users delete or add another license, these changes will be reflected after one hour.

Intelligence License

Intelligence comes with a trial license. Install a valid license before the trial license expires or if the trial license policy has been violated. Purchase the relevant license based on the number of users you want Intelligence to run analytics on.

The license policy is violated when the number of users exceeds the maximum limit. Renew your license before its validity expires or if the license policy has been violated.

The database comes with an inbuilt license. The storage capacity for the license is 976 PB. Renew your database license if the storage capacity exceeds 976 PB.

SOAR License Check

ArcSight SOAR is provided to ESM and Recon customers with no additional cost. While operating, SOAR checks for an active ESM or Recon license on the environment.

Verifying Expired License Keys

You can check if the license has expired with the following command:

```
# kubectl logs hercules-search-#########-##### -n arcsight-installer-##### -c hercules-search
```

If the license has expired, then the following message displays.

```
"<errorMessage>No license is found in Memory ..."
```

Creating Widgets for the Dashboard

The license for your deployed application also grants you access to the **Widget Software Development Kit** (the Widget SDK), which you can download to your local production or test environment. The Widget SDK enables you to build new widgets or modify existing widgets for deployed applications.

- "Using the Widget SDK" below
- "Considerations for Updating the Widget Store" on the next page

Using the Widget SDK

The Widget SDK requires nodejs 12.7.0, at a minimum, which comes with yarn version 1.16.0.

1. Extract the contents of the `widget-sdk-n.n.n.tgz` file to your developer workstation.
2. Follow the steps in the Getting Started section of the included *ReadMe*.
3. After you compile the new or modified widget, add it to the widget store for use in the Dashboard.
4. (Optional) To allow additional Fusion users to incorporate your custom widget into their environment, submit the widget to the ArcSight Marketplace.

Considerations for Updating the Widget Store

Review the following considerations before modifying or creating new widgets:

- Widgets provided with a deployed application are included in the default widget store directory.

  ```
  /opt/arcsight-nfs/arcsight-volume/fusion/widget-store
  ```

- Each new widget must have a unique name.
- You cannot edit an out-of-the-box widget. However, you can use the widget as a template for creating a new one. To prevent the modified widget from being erased or overwritten by a product upgrade, give the widget a non-default name.

Restarting Nodes in the Cluster

If you need to restart or shut down any node in the cluster, you must stop the Kubernetes and the databases services running on the node.

If you do not stop the services running on the node, the database on the node might get corrupted and the Kubernetes pods will not start after the restart.

To restart nodes manually:

1. (Conditional) If the node contains CDF, perform the following from the master node and worker node:
   a. Log in to the node you need to restart as the root user.
   b. Change to the following directory:

      ```
      cd <K8S_HOME>/bin/
      ```

      ```
      For example: /opt/arcsight/kubernetes/bin
      ```

   c. Execute the following command to stop the Kubernetes services:

      ```
      kube-stop.sh
      ```

   d. Execute the following command to unmount Kubernetes volumes:

      ```
      kubelet-umount-action.sh
      ```

2. (Conditional) If the node contains the database, do the following:
   a. Log in to the node as a database administrator.
   b. Execute the following command to stop the database services:

      ```
      /opt/vertica/bin/admintools -t stop_db -p <database_password> -d
      investigate --force
      ```

3. Restart the node:

   ```
   reboot
   ```

4. (Conditional) If restart fails, perform a hard reboot of the node.
5. (Conditional) After the node restarts, do the following if the node contains the database:
   a. Log in to the node as a database administrator.
   b. Execute the following command to start the database services:

      ```
      /opt/vertica/bin/admintools -t start_db -p <database_password> -d
      investigate --force
      ```

6. (Conditional) After the node restarts, do the following if the node contains CDF:
   a. Log in to the node as root.
   b. Change to the following directory:

      ```
      cd <K8S_HOME>/bin/
      ```

      ```
      For example: /opt/arcsight/kubernetes/bin
      ```

   c. Check whether all Kubernetes services are running:

      ```
      kube-status.sh
      ```

   d. (Conditional) If any of the services is not running, start the service:

      ```
      kube-start.sh
      ```

Migrating the NFS Server to a New Location

The process given here explains how to migrate your NFS server and paths to another location (including changing paths within the same NFS server). During the move, some of exported path pods from the core namespace will incur downtime as they are scaled to zero or temporarily removed. The CDF Management Portal (and all of its features) will not be available during such downtime.

Data will be moved transferred by copying first, so the original location should remain as a backup until the procedure is complete and the cluster successfully operates, with started back pods with new paths and the new NFS server.

This procedure will be executed on your primary master node, with access to the`kubectl` command and the contents of `/opt/arcsight/kubernetes`

The procedures make usage of the `volume_admin.sh` script located in `/opt/arcsight/kubernetes/scripts`

Usage: `./volume_admin.sh <Operation> <Persistent Volume> <Options>`

Where options include:

`reconfigure`: Reconfigure a persistent volume

`search`: Find persistent volume consumers

Preparation

1. Verify that all pods are running correctly with the following command:
   ```
   # kubectl get pods --all-namespaces -o wide | awk -F " *|/" '($3!=$4 ||
   $5!="Running") && $5!="Completed" {print $0}'
   ```
2. Verify status of CDF installation with the following command:
   ```
   # /opt/arcsight/kubernetes/bin/kube-status.sh
   ```
3. Prepare the new NFS volumes with the same permission set as the existing volumes.

- If you are using a software-controlled NFS, make sure the export policy is configured in the correct order. For example, for NetApp NFS, the RO/RW Access rules are None, Superuser Security types are None, User ID to which anonymous users are mapped equals 1999 (or whatever value you used during initial install).
- For using NFSv4 and later versions, make sure ID mapping (configured in (`/etc/idmapd.conf`) on both the NFS server and all NFS clients (that is, your cluster nodes) use the same domain.
- Verify that UID/GID is correct by manually mounting new NFS mount points and touching a file. Permission should be the same as for touching the file on the old NFS mount points.
- Note that for any changes on the NFS Server to take effect, all pending mounts of mountpoints should be closed.

4. Get an overview of persistent volumes for your installation with the following command:

```
# kubectl get pv
```

Migration Procedures

The recommended order in which migration should be executed on your persistent volumes is as follows:

1. itom-logging
2. arcsight-installer-xxxxx-db-backup-vol
3. itom-vol
4. db-single
5. arcsight-installer-xxxxx-arcsight-volume

In any of the following commands, `<old_nfs_mount>` and `<new_nfs_mount>` refer to manually-mounted NFS for copying or maintenance procedures, and `<new_nfs_path>` refers to the real path on the NFS server of the mountpoint for the PV change command.

> If any PV change fails, roll back any changes to the old NFS location until the issue is resolved. **Do not leave your cluster in a change-pending state.**

Migrate PV itom-logging

1. Determine the services using the itom-logging PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-logging
   ```

> **Note**: For fluentd, the YAML definition will include an NFSpath. You will need to mount it on a temporary mount to delete (and later to create) it with the following command:
> ```
> #   kubectl   delete   - f   /<old_ nfs_ mount>/itom/itom_ vol/suite-install/yamlContent/itom-fluentd.yaml
> ```

2. Scale down other services by running these commands:
   ```
   # kubectl scale --replicas=0 -n core deployment/idm
   # kubectl scale --replicas=0 -n core deployment/itom-logrotate-deployment
   ```
3. Verify all pods of interest are deleted by running this command:
   ```
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   ```
4. Verify that consumers have been removed from the PV users list:
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-logging
   ```
5. Copy NFS data to new mountpoint:
   ```
   # cp -rfp /mnt/<old_nfs_mount>/itom/logging /mnt/<new_nfs_
   mount>/itom/logging
   ```

6. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
   ```
   # ls -l /mnt/<old_nfs_mount>/itom/logging
   # ls -l /mnt/<new_nfs_mount>/itom/logging
   ```
7. Authorize the PV change by running this command:
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure itom-
   logging -t nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/itom/logging
   ```
8. Verify the new NFS path in the configuration by running the following command:
   ```
   # kubectl get pv itom-logging -o yaml
   ```
9. For the previous command, locate the `nfs:` section of the output. It should list the new server and volume.
10. Repeat all the commands you used to scale down or destroy the pods to scale all replicas up or start up related daemonsets.
11. Recreate the daemonset from the YAML with these commands. (Note that this will be still old path until itom_vol PV is migrated.)
    ```
    # kubectl create -f /<old_nfs_mount>/itom/itom_vol/suite-
    install/yamlContent/itom-fluentd.yaml
    # kubectl scale --replicas=<value> -n core deployment/idm
    # kubectl scale --replicas=<value> -n core deployment/itom-logrotate-
    deployment
    ```
12. Verify that consumers have been restored with this command:
    ```
    # /opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-logging
    ```
13. Verify pods are all running:
    ```
    # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
    awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
    ```
14. If all pods are running, verify CDF status:
    ```
    # /opt/arcsight/kubernetes/bin/kube-status.sh
    ```

Migrate PV arcsight-installer-xxxxx-db-backup-vol

> Some additional checks are omitted from this procedure, but should be run as in the procedure above, to make sure no discrepancies arise.

1. Determine the services using the arcsight-installer-xxxxx-db-backup-vol PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-
   installer-xxxxx-db-backup-vol
   ```
2. Scale down the necessary deployments:
   ```
   # kubectl scale --replicas=0 deployment/itom-pg-backup -n arcsight-
   installer-xxxxx
   ```

3. Verify that consumers have been removed:
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-
   installer-xxxxx-db-backup-vol
   ```
4. Copy the NFS data to a new mountpoint:
   ```
   # cp -rfp /mnt/<old_nfs_mount>/itom/db_backup /mnt/<new_nfs_
   mount>/itom/db_backup
   ```
5. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
   ```
   # ls -l /mnt/<old_nfs_mount>/itom/logging
   # ls -l /mnt/<new_nfs_mount>/itom/logging
   ```
6. Authorize the PV change:
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure arcsight-
   installer-xxxxx-db-backup-vol -t nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_
   path>/itom/db_backup
   ```
7. Repeat all the commands you used to scale down or destroy the pods to scale all replicas up or start up related daemonsets. (To restore path services, use this command: `# kubectl create -f <PATH>`)
   ```
   # kubectl scale --replicas=<value> deployment/itom-pg-backup -n arcsight-
   installer-xxxxx
   ```
8. Verify consumers have been restored with this command:
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-
   installer-xxxxx-db-backup-vol
   ```
9. Verify pods are all running:
   ```
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   ```
10. If all pods are running, verify CDF status:
    ```
    # /opt/arcsight/kubernetes/bin/kube-status.sh
    ```

Migrate PV itom-vol

1. Determine the services using the itom-vol PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-vol
   ```
2. Delete the YAML-based daemonsets by running these commands:
   ```
   # kubectl delete -f /<old_nfs_mount>/itom/itom_vol/suite-install/yamlContent/kube-registry.yaml
   # kubectl delete -f /<old_nfs_mount>/itom/itom_vol/suite-install/yamlContent/itom-fluentd.yaml
   ```
3. Scale down deployments with these commands. (Note: Make sure you have noted original number of replicas for each deployment.)
   ```
   # kubectl scale --replicas=0 -n core deployment/cdf-apiserver
   # kubectl scale --replicas=0 -n core deployment/idm
   # kubectl scale --replicas=0 -n core deployment/itom-vault
   # kubectl scale --replicas=0 -n core deployment/mng-portal
   # kubectl scale --replicas=0 -n core deployment/kube-registry
   # kubectl scale --replicas=0 -n core deployment/suite-conf-pod-arcsight-installer
   # kubectl scale --replicas=0 -n core deployment/suite-db
   # kubectl scale --replicas=0 -n core deployment/suite-installer-frontend
   ```

Note: Any consumer jobs displayed during the listing are just temporary one-time actions and can be deleted by `kubectl delete pod -n core <job_name>`

4. Verify if all Pods are deleted and not in terminating state by running this command:
   ```
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide | awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   ```
5. After make sure PV consumers list is returned empty:
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-vol
   ```
6. Copy the NFS data to a new mountpoint:
   ```
   # cp -rfp /mnt/<old_nfs_mount>/itom/itom_vol /mnt/<new_nfs_mount>/itom/itom_vol
   ```
7. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
   ```
   # ls -l /mnt/<old_nfs_mount>/itom/logging
   # ls -l /mnt/<new_nfs_mount>/itom/logging
   ```
8. Authorize PV change:
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure itom-vol -t nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/itom/itom_vol
   ```
9. Repeat all the commands you used to scale down or destroy the pods to scale all replicas up or start up related daemonsets.
   ```
   #kubectl scale --replicas=<value> -n core deployment/cdf-apiserver
   ```

```
kubectl scale --replicas=<value> -n core deployment/idm
kubectl scale --replicas=<value> -n core deployment/itom-vault
kubectl scale --replicas=<value> -n core deployment/mng-portal
kubectl scale --replicas=<value> -n core deployment/kube-registry
kubectl scale --replicas=<value> -n core deployment/suite-conf-pod-
arcsight-installer
kubectl scale --replicas=<value> -n core deployment/suite-db
kubectl scale --replicas=<value> -n core deployment/suite-installer-
frontend
kubectl create -f /<new_nfs_mount>/itom/itom_vol/suite-
install/yamlContent/kube-registry.yaml
kubectl create -f /<new_nfs_mount>/itom/itom_vol/suite-
install/yamlContent/itom-fluentd.yaml
```

10. To restore path services, use this command:

    ```
    # kubectl create -f <PATH>
    ```

11. Verify consumers have been restored with this command:

    ```
    # /opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-vol
    ```

12. Verify pods are all running:

    ```
    # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
    awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
    ```

13. If all pods are running, verify CDF status:

    ```
    # /opt/arcsight/kubernetes/bin/kube-status.sh
    ```

Migrate PV db-single

1. Determine the services using the db-single PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search db-single
   ```
2. Scale down the necessary deployments:
   ```
   # kubectl scale --replicas=0 -n core deployment/itom-postgresql-default
   ```
3. Verify pods are not stuck in terminating state, and afterward no consumers are displayed:
   ```
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   #/opt/arcsight/kubernetes/scripts/volume_admin.sh search db-single
   ```
4. Copy the NFS data to a new mountpoint:
   ```
   # cp -rfp /mnt/<old_nfs_mount>/itom/db /mnt/<new_nfs_mount>/itom/db
   ```
5. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
   ```
   # ls -l /mnt/<old_nfs_mount>/itom/logging
   # ls -l /mnt/<new_nfs_mount>/itom/logging
   ```
6. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
   ```
   # ls -l /mnt/<old_nfs_mount>/itom/logging
   # ls -l /mnt/<new_nfs_mount>/itom/logging
   ```
7. Authorize the PV change by running this command:
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure db-single -
   t nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/itom/db
   ```
8. Repeat all the commands you used to scale down or destroy the pods to scale all replicas up.
9. Verify consumers have been restored with this command:
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search db-single
   ```
10. Verify pods are all running:
    ```
    # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
    awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
    ```
11. If all pods are running, verify CDF status:
    ```
    # /opt/arcsight/kubernetes/bin/kube-status.sh
    ```

Migrate PV arcsight-installer-xxxxx-arcsight-volume

1. Determine the services using the db-single PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-
   ```

```
installer-xxxxx-arcsight-volume
```

2. Scale down the necessary deployments with the following commands, **in the listed order**. (Your list may vary depending on your Transformation Hub configuration). Note that between each scaledown command, you will run a `get pods` command as shown to make sure the scaledown has finished successfully, before proceeding to the next consumer.

   ```
   # kubectl scale --replicas=0 -n arcsight-installer-xxxxx deployment/th-
   kafka-manager
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   # kubectl scale --replicas=0 -n arcsight-installer-xxxxx deployment/th-
   schemaregistry
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   # kubectl scale --replicas=0 -n arcsight-installer-xxxxx deployment/th-
   web-service
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   # kubectl scale --replicas=0 -n arcsight-installer-xxxxx sts/th-routing-
   processor-group1
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   # kubectl scale --replicas=0 -n arcsight-installer-xxxxx
   deployment/autopass-lm
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   ```

> Note: Scaling down can take some time. Please be patient, as this is normal behavior.

3. Run these commands in the listed order:

   ```
   # kubectl scale --replicas=0 -n arcsight-installer-xxxxx sts/th-kafka
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   #kubectl scale --replicas=0 -n arcsight-installer-xxxxx sts/th-zookeeper
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   ```

4. Verify that no consumers are displayed for the PV by running the following command:

   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-installer-xxxxx-
   arcsight-volume
   ```

5. Copy the NFS data to a new mountpoint:

   ```
   # cp -rfp /mnt/<old_nfs_mount>/itom/db /mnt/<new_nfs_mount>/itom/db
   ```

6. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:

   ```
   # ls -l /mnt/<old_nfs_mount>/arcsight
   # ls -l /mnt/<new_nfs_mount>/arcsight
   ```

7. Authorize the PV change:

   ```
   # /opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure arcsight-installer-xxxxx-arcsight-volume -t nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/arcsight
   ```

8. Authorize PV change and verify the new server and volume are listed under “`nfs:`” section in the configuration:

   ```
   # opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure arcsight-installer-xxxxx-arcsight-volume -t nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/arcsighT
   # kubectl get pv arcsight-installer-xxxxx-arcsight-volume -o yaml
   ```

9. Run the scale up commands in the order shown. (After each scaleup, you will run the get pods command as shown to make sure nothing is in the crashing state.)

   ```
   # kubectl scale --replicas=<value> -n arcsight-installer-xxxxx deployment/autopass-lm
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide | awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   # kubectl scale --replicas=<value> -n arcsight-installer-xxxxx sts/th-zookeeper
   #/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide | awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   # kubectl scale --replicas=<value> -n arcsight-installer-xxxxx sts/th-kafka
   # /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide | awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   ```

10. When all th-zookeeper and th-kafka nodes are in the running state, run these commands to scale up the rest of the PV consumers. (Note that this list may vary depending on your configuration):

    ```
    # kubectl scale --replicas=<value> -n arcsight-installer-xxxxx deployment/th-kafka-manager
    # kubectl scale --replicas=<value> -n arcsight-installer-xxxxx deployment/th-schemaregistry
    # kubectl scale --replicas=<value> -n arcsight-installer-xxxxx deployment/th-web-service
    # kubectl scale --replicas=<value> -n arcsight-installer-xxxxx sts/th-routing-processor-group1
    ```

11. Log into Kafka manager and verify topic assignment between brokers, and if all brokers are up and running.

Managing the CDF Infrastructure

> For information about CDF infrastructure, see " Understanding the CDF Infrastructure" on page 8

This section provides information about managing the CDF infrastructure.

- "Accessing the CDF Management Portal" below
- "Adding Additional Worker Nodes to the Cluster" on page 423
- "Maintaining Certificates" on page 425
- "Managing CDF Management Portal Access" below
- "Configuring Flannel Memory" on page 428
- "The CDF Doctor Utility" on page 429

Accessing the CDF Management Portal

The CDF management portal enables management, deployment, and configuration of CDF and CDF-based products.

To open the management portal for an on-premises installation:

1. Browse to `https://<ha-address>:5443`.
2. Enter the username *admin* and the password where *Ha-address:* FQDN corresponding to the Virtual IP address provided during installation (`--ha-virtual-ip`) (or, for a single-master installation, the IP address of the master node).

To open the management portal for an Azure-based cluster:

1. On the jump host, browse to `http://<private_DNS>:5443`.
2. Enter the username *admin* and the password.

To open the management portal for an AWS-based cluster:

1. On the bastion host, use either the forwarding display or forwarding local ports methods to browse to `http://<ALB DNS name>:5443`.
2. Enter the username *admin* and the password.

Managing CDF Management Portal Access

At times, you may be unable to log in to the CDF Management Portal using admin rights. When this situation occurs, you can unlock the user's account or reset the user's password.

Resetting the CDF Administrator Password

You can reset the administrator password on a CDF installation.

1. Browse to CDF Management Portal.
2. Log in using admin USERID and the password you specified during the platform installation in the command line argument. (This URL is displayed at the successful completion of the CDF installation shown earlier.)
3. In the left navigation page, click **IDM Administration.**
4. In the main panel, click **SRG**.
5. In the left navigation bar, click **Users**.
6. In the list of users on the right, select *Admin* and click **Edit**.
7. In the bottom right, click **Remove Password.**
8. Click **Add Password.**
9. Enter a new admin password, and then click **Save**.

Unlocking the CDF Management Portal User Account

To unlock the account:

1. Access the following.

   ```
   # [root@n15-214-140-h240 opt]# kubectl exec -it $(kubectl get pod -n core -ocustom-
   columns=NAME:.metadata.name |grep idm|head -1) -n core sh
   ```

2. Change the default container name to `idm`.
3. Run the following commands to see all containers in this pod.

   ```
   # kubectl describe pod/idm-798f7bc6f6-2mqhk -n core
   ```

   ```
   sh-4.4# sh /idmtools/idm-installer-tools/idm.sh databaseUser unlockUser -org
   Provider -name admin
   ```

4. The following output is displayed:
   `INFO User admin is unlocked successfully.`

Resetting the User's Password

To reset the account password:

1. Run the following command to access the `idm` pod:
   ```
   # kubectl exec -it $(kubectl get pod -n core -ocustom-
   columns=NAME:.metadata.name |grep idm|head -1) -n core sh
   ```
2. Run the following command to reset the password to a temporary value. (Replace `<new_tmp_password>` with your new temporary password.)

```
# sh /idmtools/idm-installer-tools/idm.sh databaseUser resetPassword -org
Provider -name "admin" -plainPwd "<new_tmp_password>"
```

> If the user account is locked due to too many failed login attempts, run unlock, as described above in "Unlocking the CDF Management Portal User account"

3. Log into the CDF Management Portal with the new temporary password, and then set the new non-temporary password on the password reset page.
4. Log in to the CDF Management Portal with the new password.

Adding Additional Worker Nodes to the Cluster

To scale out the cluster for increased events processing and analytics computing power, you can add worker nodes to it. You can add the worker nodes either before deploying Intelligence or after deploying it.

> If you are yet to deploy Intelligence and need to add additional worker nodes, then consider the following:
>
> - When deploying Intelligence in a new cluster, add the worker nodes during the deployment. The following procedure is not applicable.
> - When deploying Intelligence in an existing cluster, add the worker nodes before starting the deployment.

To add worker nodes to a cluster:

1. Launch the CDF Management Portal on port 5443.
2. Log in with the following credentials:

   User name: admin

   Password: *<the password you provided during CDF installation>*

3. Click **Cluster** > **Nodes**.
4. Click **+ ADD**.
5. In the **Add Worker Node** dialog box, specify the required configuration information and click **ADD**.
6. Repeat Steps 4 and 5 to add more worker nodes.
7. (Conditional) If you are yet to deploy Intelligence in an existing cluster, skip the remaining steps and proceed to deploying Intelligence in an existing cluster.
8. In **Predefined Labels**, specify a label in the text box and click the + icon. Repeat this step to add more labels.

> For more information about labeling nodes, see Labeling the Nodes.

9. Drag and drop each of the labels you added to the corresponding worker nodes based on your workload sharing configuration. The corresponding components get deployed on the corresponding worker nodes.
10. Click **Refresh** to see the labels you applied to the nodes.
11. Click ⋮ and then click **Reconfigure**.
12. Depending on the capabilities for which you have assigned labels to the worker nodes, click the relevant tabs and reconfigure the properties.
13. Click **Save**.
14. Verify that all the pods are in the Running state:
    a. Launch a terminal session and log in to the master node as the root user.
    b. Execute the following command:

```
kubectl get pods --all-namespaces -o wide
```

Checking Kubernetes Dashboard for Status and Errors

1. Log in to the CDF Management Portal.
2. Navigate to **Cluster** > **Dashboard** to access the Kubernetes Dashboard.
3. In Kubernetes Dashboard change Namespace to arcsight-installer-*.
4. Navigate to **Workloads** > **Pods**.
5. View the status of pods. For more information about each pod, see Understanding Labels and Pods.
6. Clicking a pod reveals more status details of that pod.
    a. Logs for the pod can be viewed by clicking on View Logs button in the right side of the blue banner near the top.
    b. Each pod may contain multiple containers, so when viewing logs, be sure to use the Logs from <container> to view the logs for the specific container you need to view.
    c. Logging levels can be modified as described at here.

Maintaining Certificates

Certificates and their Certificate Authority (CA) have an expiration date; therefore, they need to be renewed prior to expiring in order for the cluster to operate properly.

To better understand the CAs in the cluster, see "Signing the External Communication Certificate with Your Certificate Authority" on page 313.

> In this section, `${K8S path}` refers to:
> On-premises: `/opt/arcsight/kubernetes`
> Cloud: `${cdf-deployer path}`

- "Viewing the CA Validity Dates" below
- "Renewing Internal CAs" below
- "Renewing External CAs" on the next page
- "Renewing External Certificate of Management Portal and Fusion Single-Sign-On Portal " on the next page

Viewing the CA Validity Dates

- Internal CA (RIC and RID CA) is reported in the beginning of each kube-status run with time/date and days remaining till expiration.
- To view the external CA (RE CA) validity dates, execute the following command on the primary master node. Or, if deployed to the Cloud, execute the command on the jump host.

```
# ${K8S path}/scripts/cdf-updateRE.sh read | openssl x509 -noout -issuer -subject -dates
```

Renewing Internal CAs

> This information for is for pod communication within the cluster and not for certificates used for external pod communication.

> To check if your Internal Certificate Authority is close to expiration, login into CDF Management Portal, which will show a warning if less than 30 days are left till expiration. Alternativel, you can run the `kube-status.sh` script from `/opt/arcsight/kubernetes/bin` (installation path by default). Expiration date will be reported as the first line in the script output.

To renew internal CAs and dependent certificates:

1. Execute `renewCert`. This action also distributes renewed CA between the nodes.

   ```
   # ${K8S path}/scripts/renewCert --renew -t -V 730
   ```

2. Follow the on-screen prompts to generate new certificates, distribute them between the nodes using `scp`, and apply certificates by restarting nodes one by one.

Renewing External CAs

> This procedure updates the certificates used by the CDF Management Portal as well as ArcSight capabilities. Changing the certificate by way of the CDF Management Portal, Administration > Certificate, only changes the certificate used by the CDF Management Portal.

To renew external CAs, request that your PKI team generates an intermediate certificate and matching key. Be sure to obtain any higher root certificate authority or a whole chain if more that one level used. If you cannot get a key from your PKI team, see "Signing the External Communication Certificate with Your Certificate Authority" on page 313.

1. Execute `cdf-updateRE.sh`.

   ```
   # ${K8S path}/scripts/cdf-updateRE.sh write --re-key={New Intermediate Key
   Name}.pem --re-crt={New Intermediate Certificate Name}.crt
   ```

   > If your intermediate certificate is signed by higher root certificate authority provide a chain of root CA certificate and intermediate certificate concatenated in one file (keeping the headers) to the "re-crt" parameter. Make sure intermediate certificate is last in the file.

2. Pods of the deployed ArcSight capabilities that perform external communication continue to use the certificates generated by the platform on the pod start up until the pod is restarted.

   > To understand the pods that perform external communication, see "Understanding Labels and Pods" on page 584.

Renewing External Certificate of Management Portal and Fusion Single-Sign-On Portal

> Management port 5443 and Single-Sign-On port 443. Please also, note the nginx.CRT and nginx.KEY files need to be created by the user.

To renew certificate for portals:

1. Export the following access token dependencies, which you can remove later if not needed to invoke vault anymore.

```
# export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
# export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core -o json
2>/dev/null | jq -r '.data."root.token"')
```

```
# export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -md
sha256 -a -d -pass pass:"${PASSPHRASE}")
```

2. Request Vault to generate the nginx certificate for your external access host.

```
# /opt/arcsight/kubernetes/bin/vault write -tls-skip-verify -format=json \
```

```
# RE/issue/coretech common_name=<FQDN>
```

where <FQDN> is the fully qualified domain name of the Virtual IP address (the so-called "External Access Hostname"). For a multi-master type of installation or for the single master/single worker installation, use the FQDN of the master node. If you need to set additional parameters in the generated certificate, use this syntax: `common_name=<FQDN> country=<country> province=<state> locality=<city or town> organization=<orgname>`

> The full list of parameters accepted while generating CSR can be found in the Vault documentation - "PKI engine - Generate Intermediate - Parameters"

Save the output results into nginx.CRT and nginx.KEY files accordingly. When copying the KEY and CERT into nginx.CRT and nginx.KEY, replace all "\n" with CR. Please note the nginx.CRT and nginx.KEY files need to be created by the user.

2. Apply them by running the following commands.

```
# kubectl create secret generic "nginx-investigate-secret" --from-
file=tls.crt=./nginx.CRT \
```

```
--from-file=tls.key=./nginx.KEY --dry-run -o yaml \
```

```
| kubectl --namespace="arcsight-instaler-xxxxx" apply -f -
```

```
# kubectl create secret generic "nginx-default-secret" --from-
file=tls.crt=./nginx.CRT \
```

```
--from-file=tls.key=./nginx.KEY --dry-run -o yaml \
```

```
| kubectl --namespace="core" apply -f -
```

3. Get the pod information.

```
# kubectl get pods --all-namespaces | grep single-sign-on
```

4. Open a bash terminal in the currently running pod.

   ```
   # kubectl exec -it fusion-single-sign-on-xxxxxxxxxx-xxxxx -n arcsight-installer-
   xxxxx -c hercules-osp -- bash
   ```

5. Import the new RE certificate. If you receive message about existing alias, try another name.

   ```
   # cd /usr/local/tomcat/conf/default/
   ```

   ```
   # keytool -importcert -storepass $KEYSTORE_PASSWORD -destkeystore \
   ```

   ```
   sso.bcfks -alias updatedreca -file /vault-crt/trustedCAs/RE_ca.crt -storetype \
   ```

   ```
   BCFKS -providerclass \
   ```

   ```
   org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider \
   ```

   ```
   # -providerpath /usr/local/openjdk-8/jre/lib/ext/bc-fips-1.0.2.jar
   ```

6. Restart the pods fusion-single-sign-on-xxxxx and fusion-user-management-xxxxx.
7. Make sure to import the certificate to your browser's trust store for proper functionality of both Management and Fusion portals.

Configuring Flannel Memory

> Applies only if you installed Intelligence in a new cluster.

In some cases, after flannel pods have been running continuously for some time, the Kafka Manager pod (and others) might terminate abruptly. To prevent this issue, you must modify the flannel file.

1. Execute the following commands to back up the existing **yaml** file:

   ```
   cp $
   {K8S_HOME}/objectdefs/flannel.yaml ${K8S_HOME}
   /objectdefs/flannel.yaml.orig
   ```

2. Do one of the following to modify the flannel **yaml** file:
   - In the **vi ${K8S_HOME}/objectdefs/flannel.yaml** file, change both the request and limits memory to 250Mi.
   - Execute the command: `sed -i s/50Mi/250Mi/g ${K8S_HOME}/objectdefs/flannel.yaml`.
3. Execute the following command to delete the existing flannel file:

```
kubectl delete -f ${K8S_HOME}/objectdefs/flannel.yaml
```

4. Execute the following command to create a new flannel **yaml** file:

   ```
   kubectl create -f ${K8S_HOME}/objectdefs/flannel.yaml
   ```

5. Execute the following command on each flannel pod to verify the change:

   ```
   kubectl get pod $f_pod -n kube-system -o yaml | grep -A6 resources|grep
   memory
   ```

6. Ensure that the memory value is set to 250Mi.

The CDF Doctor Utility

The CDF Doctor utility can be used to check for and repair issues discovered with an on-premises CDF installation.

CDF Doctor is located at `{K8S_HOME}/tools/cdf-doctor`.

> The CDF Doctor is only supported for on-premises CDF installations.

Running CDF Doctor

For maximum visibility into issues, run CDF Doctor on each problematic node.

To run the CDF Doctor on a problematic node:

1. Enter the following commands:

   ```
   # cd $K8S_HOME/tools/cdf-doctor/
   # ./cdf-doctor cluster check
   ```

2. When prompted for login credentials:

   a. For username use `admin`

   b. For password, use your password for the CDF management portal (that is, at `https://<your high availability FQDN>:5443`)

> You can run CDF Doctor on a failed master node by adding the `--master` parameter to the `cluster check` run command.

Types of Diagnostic Checks

When run, CDF Doctor will perform an array of diagnostic checks by default. Some checks permit CDF Doctor to repair an issue as soon as it is detected. Default diagnostic checks run by CDF Doctor will check the following components.

- CDF components
- Native components, such docker and kubelet (On-Premises CDF only)
- Kube-system (`etcd`)

Default diagnostic checks are compatible with all CDF versions 2020.08 and later.

Component	Checks...
Docker	Docker status
`kubelet`	• `kubelet` status • whether policy is loaded when SELinux is enforcing • whether `kubelet` runtime data directory is missing • whether `kubelet` certificate files's permission is incorrect • whether `kubelet` certificate is expired • whether swap is off • whether swap is enabled
`Etcd`	`etcd` status
`cdf-apiserver`	`cdf-apiserver` status
`dashboard`	`dashboard` status
`db-backup`	`db-backup` status
`idm`	`idm` status
`mng-portal`	`mng-portal` status
`nginx-ingress`	`nginx-ingress-controller` status
`node`	cluster node status
`pv`	persistent volume status
`registry`	registry status
`suite-config`	FQDN in `suite-conf-cm-` FQDN in `suite-conf-ing-` FQDN in `suite-conf-pod-`

Component	Checks...
`suite-frontend-ingress`	`suite-frontend-ingress` status
`suite-frontend-ui`	`suite-frontend-ui` status
`suite-update`	• FQDN in `suite-upgrade-cm-` • FQDN in `suite-upgrade-ing-`
Vault	• Vault component status • whether node NTP service is enable and synced • cluster nodes time difference • suite metadata folder permission • whether vault policy incorrect (automated fix) • whether pullsecret exists • whether can login to registry • whether registry contains jdbc image • FQDN in `nginx-ingress-controller` deployment • FQDN in idm deployment and ingress • whether suite parameter file is missing • PV info in suite parameter file (fix provided after user confirmation) • FQDN in ingress • FQDN in mng-portal deployment and ingress • FQDN in frontend-ingress deployment • check FQDN in suite-installer-frontend deployment • FQDN in itom-k8s-dashboard deployment and ingress • FQDN in itom-pg-backup-config configmap • FQDN in itom-ingress-pg-backup ingress

Dump File

The dump file provides a quick way to gather information about nodes where CDF is deployed. The file can be used quickly gather and encrypt information to provide for support investigation of issues.

To generate a dump file with check results, run CDF Doctor with the `--encrypt-password` parameter. You can also provide a username and password to get additional dump data from the CDF Management Portal.

The dump file contains the following information:

File Section	Description
OS commands output	Refer to `$K8S_HOME/tools/support-tool/conf/supportdump.config`
Directory content	Refer to `$K8S_HOME/tools/support-tool/conf/supportdump.config`
Files content	Refer to `$K8S_HOME/tools/support-tool/conf/supportdump.config`
Kube-Info	• Docker version and installation status • `kubelet` version and Installation status • Current node infomation Current node information: • Docker containers on current node • Docker images on current node's docker runtime Cluster info: • namespace, pv, pvc, nodes, deployment, service,pod,ingress Pod container information: • pod name • pod namespace • node pod is running on • images pod uses Suite info: • manage portal accessibility • selected features Deployment information • Docker journal logs • Docker images details collected from Docker inspection • cluster dump info collected from `kubectl` cluster dump • pod description • suite-db data • suite metadata

Managing the Database

This section provides information about managing the database.

Setting FIPS Mode on the Database Server

To enable the FIPS mode, you should set the operating system in FIPS mode.

Enabling FIPS Mode

1. Run the commands given below:

   ```
   yum install dracut-fips
   ```

   ```
   yum install dracut-fips-aesni
   ```

   ```
   rpm -q prelink && sed -i '/^PRELINKING/s,yes,no,' /etc/sysconfig/prelink
   ```

   Ignore the error if prelink was not installed.

   ```
   rpm>mv -v /boot/initramfs-$(uname -r).img{,.bak}
   ```

   ```
   rpm>dracut
   ```

   ```
   grubby --update-kernel=$(grubby --default-kernel) --args=fips=1
   ```

   ```
   uuid=$(findmnt -no uuid /boot)
   ```

   ```
   [[ -n $uuid ]] && grubby --update-kernel=$(grubby --default-kernel) \
   ```

   ```
   --args=boot=UUID=${uuid}
   ```

   ```
   reboot
   ```

2. To verify whether the FIPS mode is enabled, run the following command:

   ```
   sysctl crypto.fips_enabled
   ```

   Expected Result:

   ```
   crypto.fips_enabled = 1
   ```

Disabling FIPS Mode

1. Run the commands given below:

   ```
   yum remove dracut-fips
   ```

   ```
   dracut --force
   ```

   ```
   grubby --update-kernel=$(grubby --default-kernel) --remove-args=fips=1
   ```

   ```
   reboot
   ```

2. To verify whether the FIPS mode is disabled, run the following command:

   ```
   sysctl crypto.fips_enabled
   ```

   Expected Result:

   ```
   crypto.fips_enabled = 0
   ```

Backing Up and Restoring the Database

This section discusses backing up and restoring the Database.

- "Preparing the Database Backup Host" below
- "Backing Up the Database" on page 440
- "Managing Database Backups" on page 445
- "Restoring the Database" on page 445

Preparing the Database Backup Host

Micro Focus recommends each backup host have space for at least twice the node footprint size. Consider your long-term backup storage needs.

- "Understanding Considerations" below
- "Estimating the Required Storage Space" below
- "Setting Up Password-less SSH" on the next page
- "Preparing Backup Configuration File" on page 436

Understanding Considerations

Consider the following when backing up and restoring the Database:

- The backup process can consume additional storage. The amount of space that the backup consumes depends on the size of your catalog and any objects you drop during the backup. The backup process releases this storage after the backup is complete.
- You can only restore backups to the same version of Database. For example, you cannot back up Database 9.1.0 and restore it to Database 9.2.0.
- For optimal network performance, each node should have its own backup host.
- Use one directory on each node to store successive backups.
- You can save backups to the local folder on the node or to a remote server.
- You can perform backups on ext3, ext4, and NFS file systems.

Estimating the Required Storage Space

If you are using a single backup location, you can use the following Database operation to estimate the required storage space for the Database cluster.

```
dbadmin=> select sum(used_bytes) as total_used_bytes from v_monitor.storage_containers;
```

```
total_used_bytes
------------------
5717700329
(1 row)
```

If you are using multiple backup locations, one per node, use the following Database operation to estimate the required storage space:

```
dbadmin=> select node_name, sum(used_bytes) as total_used_bytes from v_monitor.storage_
containers group by node_name;
```

```
node_name        |   total_used_bytes
------------------------+--------------------
 v_investigate_node0002 | 1906279083
 v_investigate_node0003 | 1905384292
 v_investigate_node0001 | 1906036954
(3 rows)
```

Remote backup hosts must have SSH access, and you must configure password-less SSH from node1 in order for the database administrator to access the hosts.

If one host is the backup destination for multiple nodes, increase the maximum SSH connections on the backup host by increasing the `MaxStartups` parameter in `/etc/ssh/sshd_config`. The `MaxStartups` number should be greater than the number of nodes in the cluster.

Setting Up Password-less SSH

To set up password-less SSH:

1. Log in to the backup server.
2. Create user `$dbadmin`.`$dbadmin` is the administrator for the cluster.
3. Ensure that `$dbadmin` has write permission to the dedicated directory where you will store the backup.
4. Log in to node1 as `root`.
5. Change to the Database administrator:

   ```
   # su -l $dbadmin
   ```

6. Setup password-less SSH for all backup servers:

```
# ssh-keygen -t rsa
```

```
# ssh-copy-id -i ~/.ssh/id_rsa.pub $dbadmin@$back_up_server_ip
```

Preparing Backup Configuration File

Database includes sample configuration files that you can copy, edit, and deploy for your various *vbr* tasks.

The Database automatically installs these files at:

```
/opt/vertica/share/vbr/example_configs.
```

For more information, see Sample VBR .ini Files.

The default number of restore points (`restorePointLimit`) is 52, assuming a weekly backup for one year. Using multiple restore points gives you the option to recover from one of several backups. For example, if you specify 3, you have 1 current backup and 3 backup archives.

We use `backup_restore_full_external.ini` as an example.

```
# su - $dbadmin
```

```
# cp /opt/vertica/share/vbr/example_configs/backup_restore_full_external.ini db_
backup.ini
```

```
# vi db_backup.ini
```

You must save a copy of `db_backup.ini` for future tasks.

The following is an example for reference only `.v_investigate_node000*` is hard coded. `dbName = investigate` is hard coded.

```
# cat db_backup.ini
```

```
; This sample vbr configuration file shows full or object backup and restore to a
separate remote backup-host for each respective database host.
```

```
; Section headings are enclosed by square brackets.
```

```
; Comments have leading semicolons (;) or pound signs (#).
```

```
; An equal sign separates options and values.
```

```
; Specify arguments marked '!!Mandatory!!' explicitly.
; All commented parameters are set to their default value.
; ------------------------------------------- ;
;;; BASIC PARAMETERS ;;;
; ------------------------------------------- ;
[Mapping]
; !!Mandatory!! This section defines what host and directory will store the backup for each node.
; node_name = backup_host:backup_dir
; In this "parallel backup" configuration, each node backs up to a distinct external host.
; To backup all database nodes to a single external host, use that single hostname/IP address in each entry below.
v_investigate_node0001 = 192.168.1.1:/opt/dbadmin/backups
v_investigate_node0002 = 192.168.1.2:/opt/dbadmin/backups
v_investigate_node0003 = 192.168.1.3:/opt/dbadmin/backups
[Misc]
; !!Recommended!! Snapshot name.  Object and full backups should always have different snapshot names.
; Backups with the same snapshotName form a time sequence limited by restorePointLimit.
; SnapshotName is used for naming archives in the backup directory, and for monitoring and troubleshooting.
; Valid characters: a-z A-Z 0-9 - _
snapshotName = Vertica_backup_09_09_2019
[Database]
; !!Recommended!! If you have more than one database defined on this Vertica cluster, use this parameter to specify which database to backup/restore.
dbName = investigate
; If this parameter is True, vbr prompts the user for the database password every time.
; If False, specify the location of password config file in 'passwordFile' parameter in [Misc] section.
dbPromptForPassword = True
; ------------------------------------------- ;
```

```
;;; ADVANCED PARAMETERS ;;;
; ------------------------------------------ ;
[Misc]
; The temp directory location on all database hosts.
; The directory must be readable and writeable by the dbadmin, and must implement POSIX style fcntl lockf locking.
tempDir = /tmp
; How many times to retry operations if some error occurs.
retryCount = 2
; Specifies the number of seconds to wait between backup retry attempts, if a failure occurs.
retryDelay = 1
; Specifies the number of historical backups to retain in addition to the most recent backup.
; 1 current + n historical backups
restorePointLimit = 52
; Full path to the password configuration file
; Store this file in directory readable only by the dbadmin
; (no default)
; passwordFile = /path/to/vbr/pw.txt
; When enabled, Vertica confirms that the specified backup locations contain
; sufficient free space and inodes to allow a successful backup. If a backup
; location has insufficient resources, Vertica displays an error message explaining the shortage and
; cancels the backup. If Vertica cannot determine the amount of available space
; or number of inodes in the backupDir, it displays a warning and continues
; with the backup.
enableFreeSpaceCheck = True
; When performing a backup, replication, or copycluster, specifies the maximum
; acceptable difference, in seconds, between the current epoch and the backup epoch.
; If the time between the current epoch and the backup epoch exceeds the value
; specified in this parameter, Vertica displays an error message.
SnapshotEpochLagFailureThreshold = 3600
[Transmission]
; Specifies the default port number for the rsync protocol.
port_rsync = 50000
; Total bandwidth limit for all backup connections in KBPS, 0 for unlimited. Vertica distributes
; this bandwidth evenly among the number of connections set in concurrency_backup.
total_bwlimit_backup = 0
; The maximum number of backup TCP rsync connection threads per node.
; Optimum settings depend on your particular environment.
; For best performance, experiment with values between 2 and 16.
concurrency_backup = 2
; The total bandwidth limit for all restore connections in KBPS, 0 for unlimited
total_bwlimit_restore = 0
; The maximum number of restore TCP rsync connection threads per node.
; Optimum settings depend on your particular environment.
; For best performance, experiment with values between 2 and 16.
concurrency_restore = 2
[Database]
; Vertica user name for vbr to connect to the database.
; This setting is rarely needed since dbUser is normally identical to the database administrator
dbUser = $dbadmin
```

Backing Up the Database

The $dbadmin user must perform the backup.

- "Understanding Available Options" below
- "Backing Up the Database" below
- "Backing Up the Database Incrementally" on page 443
- "Verifying the Integrity of the Backup" on page 443

Understanding Available Options

The following options are available for the backup configuration file:

- The default for the number of restore points is 52, assuming a weekly backup for one year. Using multiple restore points gives you the option to recover from one of several backups. For example, if you specify 3, you have 1 current backup and 3 backup archives. The Database stores the value you enter as the `restorePointLimit` parameter in the `vbr` configuration file.
- To avoid prompting in the future, the backup configuration can save the `$dbadmin` password.
- Advanced options allow additional security measures, but Micro Focus recommends using the default options.
- To back up the Database incrementally, see "Backing Up the Database Incrementally" on page 443.

Backing Up the Database

1. Log in to cluster node1 as `root`.
2. Generate a backup configuration file.

   > The configuration file is required for all future backup and restore operations.

   For each node, you must specify the backup host. The host can be either the local computer or a remote host.

   For each backup host, you must specify the directory where you want to store the backup. Following is an example configuration:

   ```
   # su -l $dbadmin
   ```

   ```
   # /opt/vertica/bin/vbr --setupconfig
   ```

```
Number of restore points: 52
Specify objects:
Object restore mode (coexist, createOrReplace or create): createOrReplace
Vertica user name: $dbadmin
Save password to avoid runtime prompt? [y/n]: n
Node v_investigate_node0001
Backup host name: <Backup_Host_1_IP>
Backup directory: /opt/vertica/backup1
Node v_investigate_node0002
Backup host name: <Backup_Host_2_IP>
Backup directory: /opt/vertica/backup2
Node v_investigate_node0003
Backup host name: <Backup_Host_3_IP>
Backup directory: /opt/vertica/backup3
Change advanced settings? [y/n]: n
Config file name: db_backup.ini
Saved vbr config to db_backup.ini.
The db_backup.ini file is created in /home/$dbadmin.
# cat ./db_backup.ini
[Misc]
snapshotName = vertica_backup
restorePointLimit = 52
objectRestoreMode = createOrReplace
[Database]
dbName = Recon
dbUser = analyst
dbPromptForPassword = True
[Transmission]
[Mapping]
```

```
v_investigate_node0001 = <Backup_Host_1_IP>:/opt/vertica/backup1
```

```
v_investigate_node0002 = <Backup_Host_2_IP>:/opt/vertica/backup2
```

```
v_investigate_node0003 = <Backup_Host_3_IP>:/opt/vertica/backup3
```

3. Initialize the backup locations:

```
                       # /opt/vertica/bin/vbr --task init --config-file db_
backup.ini
```

4. To ensure you do not lose events during the backup, stop the Kafka scheduler:

```
# exit
```

```
# cd /root/install-vertica
```

```
./kafka_scheduler stop
```

5. Back up data:

```
# su -l $dbadmin
```

```
# /opt/vertica/bin/vbr --task backup --config-file db_backup.ini
```

```
Starting backup of database Recon.
```

```
Participating nodes: v_investigate_node0001.
```

```
Enter vertica password:
```

```
Snapshotting database.
```

```
Snapshot complete.
```

```
Approximate bytes to copy: 270383427 of 270383427 total.
```

```
[==================================================] 100%
```

```
Copying backup metadata.
```

```
Finalizing backup.
```

```
Backup complete!
```

6. Verify the backup files were written to the backup locations:

```
# ssh [BACKUP HOST 1 IP] ls /opt/vertica/backup1
```

```
backup_manifest
```

```
Objects
```

```
Snapshots
```

```
# ssh [BACKUP HOST 2 IP] ls /opt/vertica/backup2
backup_manifest
Objects
Snapshots
# ssh [BACKUP HOST 3 IP] ls /opt/vertica/backup3
backup_manifest
Objects
Snapshots
```

Backing Up the Database Incrementally

Incremental backups use the same setup as a full backup and only back up what changed from the previous full backup. When you perform a full backup using the same configuration file, subsequent backups are incremental.

When you start an incremental backup, the `vbr` tool displays a backup size that is a portion of the total backup size. This portion represents the delta changes that will be backed up during the incremental backup.

Run the following command to perform an incremental backup:

```
# /opt/vertica/bin/vbr --task backup --config-file db_backup.ini
```

Verifying the Integrity of the Backup

Use the `full-check` option to verify the integrity of the Database backup.

The option reports the following:

- Incomplete restore points
- Damaged restore points
- Missing backup files
- Unreferenced files

To verify the backup integrity, run the following command:

```
# /opt/vertica/bin/vbr --task full-check --config-file db_backup.ini
```

The output is similar to the following:

```
Checking backup consistency.

List all snapshots in backup location:

Snapshot name and restore point: backup_snapshot_20180116_172347, nodes:['v_investigate_node0001'].

Snapshot name and restore point: backup_snapshot_20180116_172253, nodes:['v_investigate_node0001'].

Snapshot name and restore point: backup_snapshot_20180116_172236, nodes:['v_investigate_node0001'].

Snapshot name and restore point: backup_snapshot_20180116_172310, nodes:['v_investigate_node0001'].

Snapshot name and restore point: backup_snapshot_20180116_172158, nodes:['v_investigate_node0001'].

Snapshots that have missing objects(hint: use 'vbr --task remove' to delete these snapshots):

Backup locations have 0 unreferenced objects

Backup locations have 0 missing objects

Backup consistency check complete.
```

Managing Database Backups

This section describes how to view and delete backups.

- "Viewing Available Backups" below
- "Deleting a Backup" below

Viewing Available Backups

To view available backups:

Run the following command:

```
# /opt/vertica/bin/vbr --task listbackup --config-file db_backup.ini
```

The output is similar to the following:

```
backup backup_type  epoch   objects   nodes(hosts) file_system_type
```

```
vertica_backup_20180104_142326 full 29  v_investigate_node0001(10.12.57.27) [Linux]
```

The backup name includes the backup timestamp.

Deleting a Backup

To delete a backup:

Run the following command:

```
# /opt/vertica/bin/vbr --task remove --config-file /backup/db_backup.ini --archive 
20180104_142326
```

The output is similar to the following:

```
# 20180104_142326 is the backup timestamp
```

```
Removing restore points: 20180104_142326
```

```
Remove complete!
```

Restoring the Database

The `$dbadmin` user must perform the restore.

To restore the database:

1. Build a cluster that is identical to the original cluster.
2. Log in to node1 and stop the database:

   ```
   # cd <Vertica_Installation_Directory>
   ```

   ```
   # ./vertica_installer stop-db
   ```

3. Change to the `$dbadmin` user:

   ```
   # su -l $dbadmin
   ```

4. Copy `db_backup.ini` to `/home/$dbadmin`.
5. Restore the backup data:

   ```
   # /opt/vertica/bin/vbr --task restore --config-file db_backup.ini
   ```

   The output should be similar to the following:

   ```
   Starting full restore of database Recon.

   Participating nodes: v_investigate_node0001, v_investigate_node0002, v_investigate_
   node0003.

   Restoring from restore point: Recon_backup_20180110_010826

   Determining what data to restore from backup.

   [==================================================] 100%

   Approximate bytes to copy: 2246248425 of 2246250258 total.

   Syncing data from backup to cluster nodes.

   [==================================================] 100%

   Restoring catalog.

   Restore complete!
   ```

6. Start the database:

   ```
   # exit
   ```

   ```
   # ./vertica_installer start-db
   ```

   The output should be similar to the following:

   ```
   Starting nodes:

   v_investigate_node0001 (127.0.0.1)
   ```

```
Starting Vertica on all nodes. Please wait, databases with a large catalog may take a while to initialize.

Node Status: v_investigate_node0001: (DOWN)

Node Status: v_investigate_node0001: (DOWN)

Node Status: v_investigate_node0001: (DOWN)

Node Status: v_investigate_node0001: (DOWN)

Node Status: v_investigate_node0001: (UP)

Database Recon started successfully
```

7. Start the Kafka scheduler:

```
# cd /root/install-vertica

# ./kafka_scheduler start
```

Monitoring the Database

You can monitor the Database by using commands, or the out-of-the-box Health and Performance Monitoring dashboard included in the component. The out-of-the-box Health and Performance Monitoring dashboard is not applicable for Intelligence.

- "Understanding Database Watchdog" below
- "Monitoring Database Status" on the next page
- "Monitoring Scheduler Status" on the next page
- "Using the Health and Performance Monitoring Dashboard" on the next page
- "Modifying the Storage Utilization Threshold" on page 449

Understanding Database Watchdog

Database includes a watchdog, which is configured as a cron job to automatically run once an hour to monitor the database and perform the following operations:

- When the database disk usage exceeds storage utilization threshold (default is 95%), watchdog will start to incrementally purge the oldest event data until utilization is under threshold.
- When it detects a database cluster node is in down state, it will try to restart the node.
- Rebalance data across nodes when a node with the maximum disk utilization percentage is more than 10 percentage points greater than the node with the minimum disk utilization

percentage. When the imbalance is less than 10 percentage but greater than 5 percentage points, the watchdog will log an error but not automatically attempt to balance.

- Create database event ingestion process (Kafka Scheduler) if it is missing.
- Start database event ingestion process (Kafka Scheduler) if it is stopped.

Monitoring Database Status

Monitor the database status by using the following command:

```
/opt/arcsight-database/db_installer status
```

Monitoring Scheduler Status

Monitor the scheduler's status by using the following command:

```
/opt/arcsight-database/kafka_scheduler status
```

Using the Health and Performance Monitoring Dashboard

You can also monitor the status of the database by using the out-of-the-box Health and Performance Monitoring dashboard included in the component. The dashboard includes the following widgets.

Database Event Ingestion Timeline

The Database Event Ingestion Timeline widget represents the rate of event ingestion into the database. This widget measures when the database receives the event data.

As a SOC Manager or an IT Administrator you want to monitor the event ingestion rate into the database. Due to differences in how quickly an event from different sources arrive at the database for storage, the moment when a database stores an event differs from when the event occurred. In this widget, you can monitor when the database receives the event data.

In the Database Event Ingestion widget, you can set the Upper and Medial Threshold values. Yellow represents the EPS values occurring in between the Medial and Upper Thresholds, and red represents the values occurring above the Upper Threshold. Green represents the EPS values occurring below the Medial Threshold.

Database Storage Utilization

The Database Storage Utilization widget displays storage utilization data related to the Database nodes.

As a SOC Manager or an IT Administrator you can see the available and used space in the Database nodes. This information appears as a group of Catalog and Data in the widget’s bar graph. The widget allows you to set the Upper and Medial Threshold values. By default, yellow represents the values occurring in between the Medial and Upper Thresholds, and red represents the values occurring above the Upper Threshold. Green represents the values occurring below the Medial Threshold.

To help SOC Managers and IT Administrators ensure that disk use does not overload the database nodes, the Database Storage Utilization widget displays storage utilization data for up to five database nodes. In general, most administrators keep disk usage below 60 percent per node, thus ensuring space for temporary activity required by some query execution operators.

If the database cluster has more than five nodes in the cluster, you might specify the nodes with the least amount of free space available. In this way, you can monitor the nodes at most risk of running out space. For each node, you can compare the percent and quantity of space used to the total amount. You can also monitor the throughput and latency of the database per second.

Modifying the Storage Utilization Threshold

To modify the threshold:

1. Log in to database cluster node1 as root.
2. Change the database installer directory:

   ```
   cd /opt/arcsight-database
   ```

3. Change the storage threshold value:

   ```
   vi db.properties
   ```

   ```
   STORAGE_THRESHHOLD= <new value>
   ```

To optimize storage utilization using data retention policies, see Managing Your Stored Data.

Backing Up Recon Management and Search Data Stores

Micro Focus recommends using a backup location not under the following directory.

```
/opt/arcsight
```

Use a local folder on the system or a remote location. This procedure uses the `/opt/Recon/backup` directory as an example.

- "Backing Up the Data Stores" below
- "Restoring Recon Management and Search Data Stores" below

Backing Up the Data Stores

To back up the data stores:

1. To prohibit database access, undeploy Recon.
2. SSH to the Kubernetes cluster master node1.
3. Run the following commands:

   ```
   # cd /opt/arcsight/volumes/Recon/
   ```

   ```
   # mkdir -p /opt/Recon/backup
   ```

   ```
   # cp –R * /opt/Recon/backup
   ```

   ```
   # diff -r -s /opt/Recon/backup/mgmt /opt/arcsight/volumes/Recon/mgmt
   ```

   ```
   # diff -r -s /opt/Recon/backup/search /opt/arcsight/volumes/Recon/search
   ```

   If you do not receive a message that states that the files are identical, repeat the commands.
4. Redeploy Recon to resume operations.
5. Before you resume Recon operations, ensure that the pods are in Running status:

   ```
   # kubectl get pods --all-namespaces | grep Recon
   ```

Restoring Recon Management and Search Data Stores

When restoring the Recon management and search data stores, retain the original directory structure.

```
/opt/arcsight/volumes/Recon/
```

The management data store will be restored to the following directory.

```
/opt/arcsight/volumes/Recon/mgmt/db/
```

The search data store will be restored to the following directory.

```
/opt/arcsight/volumes/Recon/search
```

To restore the data stores:

1. Ensure that you have a valid backup of the data stores.
2. To prohibit access to the database, undeploy Recon.
3. SSH to the Kubernetes master node, and then run the following commands.

   ```
   # cd /opt/Recon/backup
   ```

   ```
   # cp –R search/* /opt/arcsight/volumes/Recon/search
   ```

   ```
   Reply yes to overwrite files and folders.
   ```

   ```
   # cd /opt/arcsight/volumes/Recon/mgmt/db/
   ```

   ```
   # rm - rf h2.lock.db
   ```

   ```
   # cp /opt/Recon/backup/mgmt/db/h2.mv.db .
   ```

   ```
   Reply yes to overwrite files and folders.
   ```

   ```
   # diff -r -s /opt/arcsight/volumes/Recon/mgmt/db/h2.mv.db
   /opt/Recon/backup/mgmt/db/h2.mv.db
   ```

   ```
   # diff -r -s /opt/Recon/backup/search /opt/arcsight/volumes/Recon/search
   ```

   You should receive a message stating that all files are identical. If they are not identical, repeat the procedure.

4. Change the permission of the Recon directory:

   ```
   # chown 1999:1999 -R /opt/arcsight/volumes/Recon/
   ```

5. Redeploy Recon to resume operations.
6. Before you resume Recon operations, ensure that the pods are in Running status:

   ```
   # kubectl get pods --all-namespaces | grep Recon
   ```

Rebooting Database Cluster

1. Log in to Database node 1.

   ```
   cd /opt/arcsight-database
   ```

2. Run the following command:

   ```
   ./db_installer stop-db
   ```

3. Reboot all cluster nodes.
4. Log in to Database node 1.

   ```
   cd /opt/arcsight-database
   ```

5. Run the following command:

   ```
   ./db_installer start-db
   ```

Managing Recon

This section provides guidance for managing Recon functions and features within the deployment.

Making Searches Case-insensitive

By default, Search queries are case-sensitive for full-text searches and field-based ones. You can modify the database to make Search insensitive to case.

As the dbadmin user in the ArcSight Database, execute the following command:

```
 – ALTER DATABASE investigate set DefaultSessionLocale = ‘en_
US@colstrength=secondary’
```

> Case-insensitive searches tend to slow Search performance.

Integrating the Platform Into Your Environment

Transformation Hub integrates with ArcSight SmartConnectors and Collectors, Logger, ESM, and ArcSight Recon. It is managed and monitored by ArcSight Management Center.

After you install and configure Transformation Hub you can use SmartConnectors and Collectors to produce and publish data to the Transformation Hub, and to subscribe to and consume that data with Logger, ESM, ArcSight Recon, Apache Hadoop, or your own custom consumer.

> Currently, cloud (Azure and AWS) clusters only support other ArcSight products which are in the Azure or AWS cloud. Integration with on-premises products is not supported for cloud-based Transformation Hub.

Transformation Hub supports both Common Event Format (CEF) versions, 0.1 and 1.0, as well as Avro and binary data formats. Transformation Hub third-party integration and other product features are explained in detail in the Transformation Hub Administrator's Guide.

Configuring Security Mode for Smart Connectors with Transformation Hub Destinations

Follow these instructions to configure a security mode for SmartConnectors with destinations on an SSL secured Transformation Hub destinations.

These procedures are provided with the following assumptions:

- You use the default password. See the appendix for FIPS Compliant SmartConnectors in the *SmartConnector User Guide* on the Micro Focus support community to set a non-default password.
- You are on the Linux platform. For Windows platforms, use backslashes (\) when entering commands instead of the forward slashes given here.
- You are using a command window to enter Windows commands. Do not use Windows PowerShell.

Azure Integration - Additional Steps

For proper integration with Azure Transformation Hub, you must perform the following additional procedures for the ArcSight product (ArcMC, SmartConnector, CTH, Logger, or ESM) you are integrating.

- Edits to the `/etc/hosts` file
- Peering
- Configure health probes and load-balancing rules for ports 32080, 9093, and 9092. (For ESM, this applies only to port 9093.)

Each of these is detailed below.

Edit the `/etc/hosts` File

You must add each instance to the product's `/etc/hosts` file.

1. Open the product's `/etc/hosts` file in a text editor.
2. Add the internal IP and FQDN for each instance in the Azure Kubernetes service. (The instance IP and FQDN can be obtained by opening the AKS resource group and then opening the aks-nodepool virtual machine scale set.)
3. Save the changes to the file.

Example additions to `/etc/hosts`:

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
```

```
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

```
10.1.1.4 aks-nodepool1-12400006-vmss000000
```

```
10.1.1.5 aks-nodepool1-12400006-vmss000001
```

```
10.1.1.6 aks-nodepool1-12400006-vmss000002
```

Peering

If the Azure product and Azure Transformation Hub are on different VLANs, you must configure peering between the two VLANs. An example is provided in the section Peering Virtual Networks.

Health Probes

You must configure health probes and load balancing rules for ports 32080, 9093, and, optionally, 9092. Configuring health probes for these ports is detailed in Configuring Port Rules for Product Integration. Use the same procedure as outlined for port 443.

AWS Integration - Additional Steps

For proper integration with AWS Transformation Hub, you must perform the following additional procedures for the ArcSight product (SmartConnectors, Logger, or ESM) you are integrating.

For configuring SmartConnectors, Loggers, and ESMs: obtain the cluster worker node (Kafka broker node) host names, using one of the following procedures:

1. From the bastion host, run the following command:
   ```
   # kubectl get nodes
   ```
2. Copy the node host names.

OR

1. In the AWS UI, go to your Auto Scaling Group.
2. Click on the **Instance Management** tab to see the instance IDs.
3. Click the first instance ID to view the details of the corresponding instance.
4. Note the Private DNS name for the instance.
5. Repeat steps 3-4 for each instance ID.

After determining these values, ensure that your SmartConnector, Logger, or ESM instance is added to the the AWS TH cluster Security Group with rules allowing access to the ports 32080, 9093, and optionally 9092.

You can then follow the integration procedures below.

Configuring a SmartConnector with a Transformation Hub Destination with Client Authentication in FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub (TH) destination with client authentication in FIPS mode.

> You will need to supply an intermediate certificate and key.

Step 1: On the SmartConnector Server

1. Prepare the connector:
   - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Enabled**.
   - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.
2. Navigate to the connector's `current` directory, for example:

   ```
   # cd <install dir>/current
   ```
3. Apply the following workaround for a Java keytool issue:
   a. Create a new file, `agent.security`, in `<install dir>/current/user/agent` (or in Windows platforms, `<install dir>\current\user\agent`).

   b. Add the following content to the new file and then save it:

   ```
   security.provider.1=org.bouncycastle.jcajce.provider
   .BouncyCastleFipsProvider
   security.provider.2=com.sun.net.ssl.internal.ssl.Provider BCFIPS
   security.provider.3=sun.security.provider.Sun
   ```

   c. Move the `lib/agent/fips/bcprov-jdk14-119.jar` file to the `current` directory.
4. Set the environment variables for static values used by keytool:

   ```
   # export CURRENT=<full path to this "current" folder>
   # export BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
   Djava.security.egd=file:/dev/urandom -J-
   Djava.ext.dirs=${CURRENT}/lib/agent/fips -J-
   Djava.security.properties=${CURRENT}/user/agent/agent.security"
   # export TH=<Transformation Hub hostname>_<Transformation Hub port>
   # export STORES=${CURRENT}/user/agent/stores
   # export STORE_PASSWD=changeit
   ```

```
# export TH_HOST=<TH master host name>
# export CA_CERT=ca.cert.pem
# export INTERMEDIATE_CA_CRT=intermediate.cert.pem
# export FIPS_CA_TMP=/opt/fips_ca_tmp
```

On Windows platforms:

```
# set CURRENT=<full path to this "current" folder>
# set BC_OPTS=-storetype BCFKS -providername BCFIPS
-J-Djava.ext.dirs=%CURRENT%\jre\lib\ext;%CURRENT%\lib\agent\fips
-J-Djava.security.properties=%CURRENT%\user\agent\agent.security
# set TH=<Transformation Hub hostname>_<Transformation Hub port>
# set STORES=%CURRENT%\user\agent\stores
# set STORE_PASSWD=changeit
# set TH_HOST=<TH master host name>
# set CA_CERT=C:\Temp\ca.cert.pem
# set INTERMEDIATE_CA_CRT=C:\Temp\intermediate.cert.pem
# set FIPS_CA_TMP=\opt\fips_ca_tmp
```

5. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist, for example:

```
# mkdir -p ${STORES}
```

On Windows platforms:

```
# mkdir -p %STORES%
```

6. Create the connector key pair, for example (the connector `FQDN`, `OU`, `O`, `L`, `ST`, and `C` values must be changed for your company and location):

```
# jre/bin/keytool ${BC_OPTS} -genkeypair -alias ${TH} -keystore
${STORES}/${TH}.keystore.bcfips -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 365
```

On Windows platforms:

```
# jre\bin\keytool %BC_OPTS% -genkeypair -alias %TH% -keystore
%STORES%\%TH%.keystore.bcfips -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF ,L=Sunnyvale,ST=CA,C=US" -validity 365
```

If the command fails, set `BC_OPTS` as follows and create the connector key pair:

```
# export BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
Djava.security.egd=file:/dev/urandom -providerpath $
{CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
```

> For Connector 8.0 or earlier, use `bc-fips-1.0.0.jar` in the above command.

When prompted, enter the password. Note the password; you will need it again in a later step. Press **Enter** to use the same password for the key. If you want to match the default value in the properties file, use the password `changeit`.

7. List the key store entries. There should be one private key.

```
# jre/bin/keytool ${BC_OPTS} -list -keystore
${STORES}/${TH}.keystore.bcfips -storepass ${STORE_PASSWD}
```

On Windows platforms:

```
# jre\bin\keytool %BC_OPTS% -list -keystore
%STORES%\%TH%.keystore.bcfips -storepass %STORE_PASSWD%
```

8. Create a Certificate Signing Request (CSR), for example:

```
# jre/bin/keytool ${BC_OPTS} -certreq -alias ${TH} -keystore
${STORES}/${TH}.keystore.bcfips -file ${STORES}/${TH}-cert-req -storepass
${STORE_PASSWD}
```

On Windows platforms:

```
# jre\bin\keytool %BC_OPTS% -certreq -alias %TH% -keystore
%STORES%\%TH%.keystore.bcfips -file %STORES%\%TH%-cert-req -storepass
%STORE_PASSWD%
```

Step 2: On the Transformation Hub Server

1. When Transformation Hub is first installed, it is configured to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an intermediate certificate and key pair. Copy them to `/tmp` with the following names:

`/tmp/intermediate.cert.pem`

`/tmp/intermediate.key.pem`

`/tmp/ca.cert.pem`

Use the following command to update the certificate on the Transformation Hub:

```
# /opt/arcsight/kubernetes/scripts/cdf-updateRE.sh write --re-
key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-
ca=/tmp/ca.cert.pem
```

> For a cloud installation, the `cdf-updateRE.sh` script has a different path:
> AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
> Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

> After the new certificate is imported to the Transformation Hub, the Transformation Hub will need to be uninstalled and then re-installed with FIPS and Client Authentication enabled.

3. Run the following commands:

   ```
   # export CA_CERT=/tmp/ca.cert.pem
   # export INTERMEDIATE_CA_CRT=/tmp/intermediate.cert.pem
   # export INTERMEDIATE_CA_KEY=/tmp/intermediate.key.pem
   # export FIPS_CA_TMP=/opt/fips_ca_tmp
   # export TH=<Transformation Hub hostname>_<Transformation Hub port>
   ```

4. Create a temporary location on the Transformation Hub master node:

   ```
   # mkdir $FIPS_CA_TMP
   ```

Step 3: On the Connector Server

Copy the `${STORES}/${TH}-cert-req` file (`%STORES%\%TH%-cert-req` on Windows platforms) from the connector to the Transformation Hub directory created above, `/opt/fips_ca_tmp`.

Step 4: On the Transformation Hub Server

Create the signed certificate, for example:

```
@ /bin/openssl x509 -req -CA ${INTERMEDIATE_CA_CRT} -CAkey ${INTERMEDIATE_CA_
KEY} -in ${TH}-cert-req -out ${FIPS_CA_TMP}/${TH}-cert-signed -days 365 -
CAcreateserial -sha256
```

Step 5: On the Connector Server

1. Copy the `${TH}-cert-signed` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the `%TH%-cert-signed` certificate to the connector's `%STORES%` directory.)
2. Copy the `ca.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
3. Copy the `intermediate.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
4. Import the CA certificate to the trust store, for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${CA_CERT} -alias
   CARoot -keystore ${STORES}/${TH}.truststore.bcfips -storepass ${STORE_
   PASSWD}
   ```

On Windows platforms:

```
# jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%CA_CERT% -alias
CARoot -keystore %STORES%\%TH%.truststore.bcfips -storepass %STORE_
PASSWD%
```

5. Import the intermediate certificate to the trust store, for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${INTERMEDIATE_
   CA_CRT} -alias INTCARoot -keystore ${STORES}/${TH}.truststore.bcfips -
   storepass ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%INTERMEDIATE_
   CA_CRT% -aliasINTCARoot -keystore %STORES%\%TH%.truststore.bcfips -
   storepass %STORE_PASSWD%
   ```

6. Import the CA certificate to the key store, for example:

   ```
   jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${CA_CERT} -alias
   CARoot -keystore ${STORES}/${TH}.keystore.bcfips -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%CA_CERT% -alias
   CARoot -keystore %STORES%\%TH%.keystore.bcfips -storepass %STORE_
   PASSWD%
   ```

7. When prompted, enter **yes** to trust the certificate.
8. Import the intermediate certificate to the key store, for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${INTERMEDIATE_
   CA_CRT} -alias INTCARoot -keystore ${STORES}/${TH}.keystore.bcfips -
   storepass ${STORE_PASSWD}
   ```

   When completed successfully, this command will return the message, `Certificate reply was installed in keystore`.

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%INTERMEDIATE_
   CA_CRT% -alias

   # INTCARoot -keystore %STORES%\%TH%.keystore.bcfips -storepass %STORE_
   PASSWD%
   ```

   If the command fails, set `BC_OPTS` as follows and create the connector key pair:

   ```
   # export BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
   Djava.security.egd=file:/dev/urandom -providerpath $
   ```

```
{CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
```

For Connector 8.0 or earlier, use `bc-fips-1.0.0.jar` in the above command.

9. Import the signed certificate to the key store, for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${TH}-cert-signed
   -alias ${TH} -keystore ${STORES}/${TH}.keystore.bcfips -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%TH%-cert-signed
   -alias %TH% -keystore %STORES%\%TH%.keystore.bcfips -storepass %STORE_
   PASSWD%
   ```

   When successfully complete, this command will return the message, *Certificate reply was installed in keystore.*

10. Navigate to the `bin` directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

    ```
    # cd <installation dir>/current/bin
    # ./runagentsetup.sh
    ```

    On Windows platforms:

    ```
    # cd <installation dir>\current\bin
    # runagentsetup.bat
    ```

    a. When completing the Transformation Hub destination fields, use the same values as in Step 8 for the path and password.

    b. Set **Use SSL/TLS** to **true**.

    c. Set **Use SSL/TLS Authentication** to **true**.

    d. Set keystore path to:
    `${STORES}/${TH}.keystore.bcflips`

    e. Set truststore path to:
    `${STORES}/${TH}.keystore.bcflips`

11. Cleanup. Delete the following files:

    Caution: The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

    ```
    # rm ${STORES}/${INTERMEDIATE_CA_CRT}
    # rm ${STORES}/intermediate.key.pem
    ```

```
# rm ${STORES}/${TH}-cert-signed
# rm ${STORES}/${TH}-cert-req
```

On Windows platforms:

```
# del %STORES%\intermediate.cert.pem
# del %STORES%\intermediate.key.pem
# del %STORES%\%TH%-cert-signed
# del %STORES%\%TH%-cert-req
```

12. Move the `bcprov-jdk14-119.jar` file back to the `lib/agent/fips` directory (or `lib\agent\fips` on Windows platforms).

Step 6: On the Transformation Hub Server

To clean up the Transformation Hub server, delete the temporary folder where the certificate was signed and the certificate and key files in `/tmp`.

Caution: The temporary certificate folder should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

SmartConnectors on AWS

When configuring a SmartConnector in AWS, for TLS with FIPS and client authentication enabled, the keytool keypair creation command might fail or appear to hang if the available entropy on the connector instance is less that 1000. It has been found that AWS instances installed with a minimum OS will have an entropy availability of only about 60.

You can check this by installing the rng-tools package and then running the a `cat` command on the connector host:

```
# yum install rng-tools -y
```

```
# cat /proc/sys/kernel/random/entropy_avail
```

If the available entropy needs to be increased, enable the `rngd` service at boot and start the `rngd` service with the following commands:

```
# systemctl enable rngd.service
```

```
# systemctl start rngd.service
```

The `rngd` service will check and feed random data from the hardware device to kernel entropy pool automatically.

Then run this command again to check available entropy:

```
# cat /proc/sys/kernel/random/entropy_avail
```

After increasing the available entropy, the keytool `create keypair` command will run normally.

Configuring a SmartConnector with Transformation Hub Destination with Client Authentication in Non-FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub (TH) destination with client authentication, in non-FIPS mode.

> You will need to supply an intermediate certificate and key.

On the SmartConnector Server

1. Prepare the SmartConnector:
   - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Disabled**.
   - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Disabled**.
2. Navigate to the connector's `current` directory, for example:

   ```
   # cd <install dir>/current
   ```

   On Windows platforms:

   ```
   # cd <install dir>\current
   ```
3. Set the environment variables for the static values used by keytool, for example:

   ```
   # export CURRENT=<full path to this "current" folder>
   # export TH=<th hostname>_<th port>
   # export STORES=${CURRENT}/user/agent/stores
   # export STORE_PASSWD=changeit
   # export TH_HOST=<TH master host name>
   # export CA_CERT=ca.cert.pem
   # export INTERMEDIATE_CA_CRT=intermediate.cert.pem
   export CERT_CA_TMP=/opt/cert_ca_tmp
   ```

   On Windows platforms:

   ```
   # set CURRENT=<full path to this "current" folder>
   # set TH=<th hostname>_<th port>
   ```

```
# set STORES=%CURRENT%\user\agent\stores
# set STORE_PASSWD=changeit
# set TH_HOST=<TH master host name>
# set CA_CERT=C:\Temp\ca.cert.pem
# set INTERMEDIATE_CA_CRT=C:\Temp\intermediate.cert.pem
# set CERT_CA_TMP=\opt\cert_ca_tmp
```

4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist, for example:

   ```
   mkdir -p ${STORES}
   ```

   On Windows platforms:

   ```
   # mkdir -p %STORES%
   ```

5. Create the connector key pair, for example:

   ```
   # jre/bin/keytool -genkeypair -alias ${TH} -keystore
   ${STORES}/${TH}.keystore.jks -dname "cn=<Connector
   FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 365
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -genkeypair -alias %TH% -keystore
   %STORES%\%TH%.keystore.jks -dname "cn=<Connector
   FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 365
   ```

   When prompted, enter the password. Note the password; you will need it again in a later step. Press Enter to use the same password for the key.

6. List the key store entries. There should be one private key.

   ```
   # jre/bin/keytool -list -keystore ${STORES}/${TH}.keystore.jks -storepass
   ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -list -keystore %STORES%\%TH%.keystore.jks -storepass
   %STORE_PASSWD%
   ```

7. Create a Certificate Signing Request (CSR), for example:

   ```
   # jre/bin/keytool -certreq -alias ${TH} -keystore
   ${STORES}/${TH}.keystore.jks -file ${STORES}/${TH}-cert-req -storepass
   ${STORE_PASSWD}
   ```

   On Windows platforms:

```
# jre\bin\keytool -certreq -alias %TH% -keystore
%STORES%\%TH%.keystore.jks -file %STORES%\%TH%-cert-req -storepass
%STORE_PASSWD%
```

On the Transformation Hub Server

1. When Transformation Hub is first installed, it's setup to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an intermediate certificate and key pair. Copy them to `/tmp` with the following names:

   ```
   /tmp/intermediate.cert.pem
   /tmp/intermediate.key.pem
   /tmp/ca.cert.pem
   ```

   Use the following command to add them to Transformation Hub:

   ```
   # /opt/arcsight/kubernetes/scripts/cdf-updateRE.sh write --re
   key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-
   ca=/tmp/ca.cert.pem
   ```

   > AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
   > Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

   > After the new certificate is imported to the Transformation Hub, the Transformation Hub will need to be uninstalled and then re-installed with FIPS and Client Authentication enabled.

2. Run the following commands:

   ```
   # export CA_CERT=/tmp/ca.cert.pem
   # export INTERMEDIATE_CA_CRT=/tmp/intermediate.cert.pem
   # export INTERMEDIATE_CA_KEY=/tmp/intermediate.key.pem
   # export CERT_CA_TMP=/opt/cert_ca_tmp
   # export TH=<Transformation Hub hostname>_<Transformation Hub port>
   ```

3. Create a temporary location on the Transformation Hub master node:

   ```
   # mkdir $CERT_CA_TMP
   ```

On the Connector Server

Copy the `${STORES}/${TH}-cert-req` file (`%STORES%\%TH%-cert-req` on Windows platforms) from the connector to the Transformation Hub directory created above.

On the Transformation Hub Server

Create the signed certificate, for example:

```
/bin/openssl x509 -req -CA ${INTERMEDIATE_CA_CRT} -CAkey ${INTERMEDIATE_CA_
KEY} -in ${TH}-cert-req -out ${CERT_CA_TMP}/${TH} -cert-signed-days 365 -
CAcreateserial -sha256
```

On the Connector Server

1. Copy the `${TH}-cert-signed` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the `%TH%-cert-signed` certificate to the connector's `%STORES%` directory.)
2. Copy the `ca.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
3. Copy the `intermediate.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
4. Import the CA certificate to the trust store, for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot
   -keystore ${STORES}/${TH}.truststore.jks -storepass ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\%CA_CERT% -alias CARoot
   -keystore %STORES%\%TH%.truststore.jks -storepass %STORE_PASSWD%
   ```

5. Import the intermediate certificate to the trust store, for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${INTERMEDIATE_CA_CRT} -
   alias
   ```

   ```
   # INTCARoot -keystore ${STORES}/${TH}.truststore.jks -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\%INTERMEDIATE_CA_CRT% -
   aliasINTCARoot -keystore %STORES%\%TH%.truststore.jks -storepass
   %STORE_PASSWD%
   ```

6. When prompted, enter **yes** to trust the certificate.
7. Import the CA certificate to the key store, for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -
   keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\${CA_CERT} -alias CARoot -
   keystore %STORES%\%TH%.keystore.jks -storepass %STORE_PASSWD%
   ```

8. Import the intermediate certificate to the key store, for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${INTERMEDIATE_CA_CRT} -
   alias
   ```

   ```
   # INTCARoot -keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\%INTERMEDIATE_CA_CRT% -
   alias INTCARoot -keystore %STORES%\%TH%.keystore.jks -storepass %STORE_
   PASSWD%
   ```

   When successfully completed, this command will return the message, *Certificate reply was installed in keystore*.

9. When prompted, enter **yes** to trust the certificate.

10. Import the signed certificate to the key store, for example:

    ```
    # jre/bin/keytool -importcert -file ${STORES}/${TH}-cert-signed -alias
    ${TH} -keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_PASSWD}
    ```

    On Windows platforms:

    ```
    # jre\bin\keytool -importcert -file %STORES%\%TH%-cert-signed -alias
    %TH% -keystore %STORES%\%TH%.keystore.jks -storepass %STORE_PASSWD%
    ```

    When successfully complete, this command will return the message, `Certificate reply was installed in keystore`.

11. Note the key store and trust store paths:

    ```
    #echo ${STORES}/${TH}.truststore.jks
    # echo ${STORES}/${TH}.keystore.jks
    ```

    On Windows platforms:

    ```
    # echo %STORES%\%TH%.truststore.jks
    # echo %STORES%\%TH%.keystore.jks
    ```

12. Navigate to the `bin` directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

    ```
    # cd <installation dir>/current/bin
    # ./runagentsetup.sh
    ```

    On Windows platforms:

    ```
    # cd <installation dir>\current\bin
    # runagentsetup.bat
    ```

a. When completing the Transformation Hub destination fields, use the same values as in Step 8 for the path and password.

b. Set **Use SSL/TLS** to **true**.

c. Set **Use SSL/TLS Authentication** to **true**.

13. Cleanup. Delete the following files:

> **Caution:** The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

```
# rm ${STORES}/${INTERMEDIATE_CA_CRT}
# rm ${STORES}/intermediate.key.pem
# rm ${STORES}/${TH}-cert-signed
# rm ${STORES}/${TH}-cert-req
```

On Windows platforms:

```
# del %STORES%\intermediate.cert.pem
# del %STORES%\intermediate.key.pem
# del %STORES%\%TH%-cert-signed
# del %STORES%\%TH%-cert-req
```

On the Transformation Hub Server

To clean up the Transformation Hub server, delete the temporary folder where the certificate was signed and the certificate and key files in `/tmp`.

> **Caution:** The temporary certificate folder should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

Configuring a SmartConnector with a Transformation Hub Destination without Client Authentication in FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub destination without client authentication in FIPS mode.

On the SmartConnector Server

1. Prepare the SmartConnector:

   - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set**

FIPS mode. Set to **Enabled**.

- **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and then **Set FIPS Mode** to **Enabled**.

2. Navigate to the connector's `current` directory, for example:

   ```
   # cd <install dir>/current
   ```

3. Set the environment variables for the static values used by keytool, for example:

   ```
   # export CURRENT=<full path to this "current" folder>
   ```

   ```
   # export BC_OPTS="-storetype BCFKS -providername BCFIPS -providerclass
   org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
   ${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar
   -J-Djava.security.egd=file:/dev/urandom"
   ```

   > For Connector 8.0, use `bc-fips-1.0.0.jar` in the above command.

   ```
   # export TH=<Transformation Hub hostname>_<Transformation Hub port>
   # export STORES=${CURRENT}/user/agent/stores
   # export STORE_PASSWD=changeit
   # export CA_CERT=ca.cert.pem
   ```

   On Windows platforms:

   ```
   # set CURRENT=<full path to this "current" folder>
   ```

   ```
   # set BC_OPTS="-storetype BCFKS -providername BCFIPS
   -J-Djava.ext.dirs=%CURRENT%\jre\lib\ext;%CURRENT%\lib\agent\fips
   -J-Djava.security.properties=%CURRENT%\user\agent\agent.security"
   ```

   ```
   # set TH=<Transformation Hub hostname>_<Transformation Hub port>
   # set STORES=%CURRENT%\user\agent\stores
   # set STORE_PASSWD=changeit
   ```

4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist, for example:

   ```
   # mkdir -p ${STORES}
   ```

   On Windows platforms:

   ```
   # mkdir -p %STORES%
   ```

5. Create a `ca.cert.pem` file with the contents of the root CA certificate with the following command:

   ```
   # ${k8s-home}/scripts/cdf-updateRE.sh > /tmp/ca.cert.pm
   ```

> For a cloud installation, the `cdf-updateRE.sh` script has a different path:
> AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
> Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

6. Copy the just-created `ca.cert.pem` file from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)

7. Import the CA certificate to the trust store, for example:

   ```
   # jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${CA_CERT} -alias
   CARoot -keystore ${STORES}/${TH}.truststore.bcfips -storepass ${STORE_
   PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%CA_CERT% -alias
   CARoot -keystore %STORES%\%TH%.truststore.bcfips -storepass %STORE_
   PASSWD%
   ```

8. When prompted, enter **yes** to trust the certificate.

9. Note the trust store path:

   ```
   # echo ${STORES}/${TH}.truststore.bcfips
   ```

   On Windows platforms:

   ```
   # echo %STORES%\%TH%.truststore.bcfips
   ```

10. Navigate to the `bin` directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

    ```
    # cd <installation dir>/current/bin
    # ./runagentsetup.sh
    ```

    On Windows platforms:

    ```
    cd <installation dir>\current\bin
    runagentsetup.bat
    ```

    a. When completing the Transformation Hub destination fields, use the value from Step 7 for the trust store path and the password used in Step 6 for the trust store password.

    b. Set **Use SSL/TLS** to **true**.

    c. Set **Use SSL/TLS Authentication** to **false**.

11. Cleanup. Delete the certificate file, for example:

Caution: The following file should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

```
# rm ${STORES}/${CA_CERT}
```

On Windows platforms:

```
# del %\STORES%\ca.cert.pem
```

Configuring a SmartConnector with a Transformation Hub Destination without Client Authentication in non-FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub destination without client authentication in non-FIPS mode. This is the default security mode configuration when installing Transformation Hub.

On the SmartConnector Server

1. Prepare the SmartConnector:
   - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Disabled**.
   - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Disabled**.
2. Navigate to the connector's `current` directory, for example:
   ```
   # cd <install dir>/current
   ```
3. Set the environment variables for the static values used by keytool, for example:
   ```
   # export CURRENT=<full path to this "current" folder>
   # export TH=<Transformation Hub hostname>_<Transformation Hub port>
   # export STORES=${CURRENT}/user/agent/stores
   # export CA_CERT=ca.cert.pem
   # export STORE_PASSWD=changeit
   ```

   On Windows platforms:

   ```
   # set CURRENT=<full path to this "current" folder>
   # set TH=<Transformation Hub hostname>_<Transformation Hub port>
   ```

```
# set STORES=%CURRENT%\user\agent\stores
# set STORE_PASSWD=changeit
```

4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist, for example:

   ```
   mkdir -p ${STORES}
   ```

   On Windows platforms:

   ```
   mkdir -p %STORES%
   ```

On the Transformation Hub:

Create a `${CA_CERT}` file with the content of the root CA certificate as follows:

1. Set the environment:

   ```
   # export CA_CERT=/tmp/ca.cert.pem
   ```

2. Create a certificate:

   ```
   # ${k8s-home}/scripts/cdf-updateRE.sh > ${CA_CERT}
   ```

> For a cloud installation, the `cdf-updateRE.sh` script has a different path:
> AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
> Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

3. Copy this file from the Transformation Hub to the connector `STORES` directory.

On the Connector:

1. Import the CA certificate to the trust store in the `${CURRENT}` folder; for example:

   ```
   # jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -
   keystore ${STORES}/${TH}.truststore.jks -storepass ${STORE_PASSWD}
   ```

   On Windows platforms:

   ```
   # jre\bin\keytool -importcert -file %STORES%\%CA_CERT% -alias CARoot -
   keystore %STORES%\%TH%.truststore.jks -storepass %STORE_PASSWD%
   ```

2. When prompted, enter yes to trust the certificate.
3. Note the trust store path:

   ```
   # echo ${STORES}/${TH}.truststore.jks
   ```

   On Windows platforms:

```
# echo %STORES%\%TH%.truststore.jks
```

4. Navigate to the bin directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

   ```
   # cd <installation dir>/current/bin
   # ./runagentsetup.sh
   ```

   On Windows platforms:

   ```
   # cd <installation dir>\current\bin
   # runagentsetup.bat
   ```

5. Set **Use SSL/TLS** to **true**.
6. Set **Use SSL/TLS Authentication** to **false**.
7. When completing the Transformation Hub destination fields, use the value from Step 3 for the trust store path and the password used in Step 4 for the trust store password.
8. Cleanup. Delete the certificate file, for example:

   > **Caution:** The following file should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

   ```
   # rm ${STORES}/${CA_CERT}
   ```

   On Windows platforms:

   ```
   # del %\STORES%\%CA_CERT%
   ```

Configuring Logger as a Transformation Hub Consumer

The procedure for configuring a Logger as a Transformation Hub consumer will depend on whether the Logger will be using SSL/TLS.

Prerequisite: Import RH CA Certificate

Prior to configuring a Logger as a Transformation Hub consumer, you manually import the Realm External CA (RE CA) certificate exported from the TH clusterThe RE certificate does not change if the Transformation Hub is restarted or redeployed. Any newly generated certificates after restart are trusted by this RE CA, enabling the receiver to continue accepting events.

To configure Transformation Hub using TLS, the user is only required to create the TH receiver and import the files as described below; Authentication and Sign in are done automatically.

Meanwhile, for Client Authentication, you must set up two way authentication between the Container Deployment Foundation as described in Transformation Hub Authentication.

Step 1: Obtain Transformation Hub RE Certificate

1. On the Transformation Hub , run the following command to retrieve the Transformation Hub RE certificate:

```
/opt/kubernetes/scripts/cdf-updateRE.sh > /tmp/RE.crt
```

> For a cloud installation, the `cdf-updateRE.sh` script has a different path:
> AWS: `aws-byok-installer/installer/cdf-deployer-<build version>/scripts/cdf-updateRE.sh`
> Azure: `cdf-deployer-<build version>/scripts/cdf-updateRE.sh`

2. Copy the `/tmp/RE.crt` obtained from step 1 to the Logger in the directory `/tmp;`.

Step 2: Set the environment on the Logger

1. On the Logger, set the `ARCSIGHT_HOME` environment variable:

- Appliance:
  `# export ARCSIGHT_HOME=/opt/arcsight/logger`
- Software:
  `# export ARCSIGHT_HOME=[logger install directory]/current/arcsight/logger`

For existing Kafka receivers only:

1. In the Logger SSH console, look for any previous TH certificates from Logger receiver trust store running the script available at:
   - Appliance:
     `/opt/arcsight/logger/bin/scripts/keytool_util.sh.`
   - Software:
     `# [Install dir]/current/arcsight/logger/bin/scripts/keytool_util.sh`

   > **Caution:**`<verisignserverca>` uses a 1000-bit RSA key which is considered a security risk.

2. Delete the TH certificates from the previous step in the Logger receiver trust store running the script available at:

```
# /opt/arcsight/logger/bin/scripts/keytool_util.sh
```

Make sure to execute the command as it follows: `./keytool_util.sh receiver delete [alias]`

Step 3: Import the RE Certificate into Logger

1. In the Logger SSH console, import the new TH RE certificate using the `RE.crt` file copied from TH running the script available at:

   `# /opt/arcsight/logger/bin/scripts/keytool_util.sh`.

   Make sure to execute the command as it follows:

   ```
   # ./keytool_util.sh receiver importcert [certificate]
   ```

2. Confirm TH FQDN is settled in Logger DNS before creating Kafka receivers in SSL mode in Logger.

For existing Kafka receivers:

On the Logger, restart the receiver processes available at:

- Appliance:
  ```
  # /opt/arcsight/logger/bin/loggerd restart receivers
  ```
- Software:
  ```
  # [install dir]/current/arcsight/logger/bin/loggerd restart receivers
  ```

Azure Integration - Additional Steps

For proper integration with Azure Transformation Hub, you must perform the following additional procedures for the ArcSight product (ArcMC, SmartConnector, CTH, Logger, or ESM) you are integrating.

- Edits to the `/etc/hosts` file
- Peering
- Configure health probes and load-balancing rules for ports 32080, 9093, and 9092. (For ESM, this applies only to port 9093.)

Each of these is detailed below.

Edit the `/etc/hosts` File

You must add each instance to the product's `/etc/hosts` file.

1. Open the product's `/etc/hosts` file in a text editor.
2. Add the internal IP and FQDN for each instance in the Azure Kubernetes service. (The instance IP and FQDN can be obtained by opening the AKS resource group and then opening the aks-nodepool virtual machine scale set.)
3. Save the changes to the file.

Example additions to `/etc/hosts`:

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
```

```
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

```
10.1.1.4 aks-nodepool1-12400006-vmss000000
```

```
10.1.1.5 aks-nodepool1-12400006-vmss000001
```

```
10.1.1.6 aks-nodepool1-12400006-vmss000002
```

Peering

If the Azure product and Azure Transformation Hub are on different VLANs, you must configure peering between the two VLANs. An example is provided in the section Peering Virtual Networks.

Health Probes

You must configure health probes and load balancing rules for ports 32080, 9093, and, optionally, 9092. Configuring health probes for these ports is detailed in Configuring Port Rules for Product Integration. Use the same procedure as outlined for port 443.

AWS Integration - Additional Steps

For proper integration with AWS Transformation Hub, you must perform the following additional procedures for the ArcSight product (SmartConnectors, Logger, or ESM) you are integrating.

For configuring SmartConnectors, Loggers, and ESMs: obtain the cluster worker node (Kafka broker node) host names, using one of the following procedures:

1. From the bastion host, run the following command:
   ```
   # kubectl get nodes
   ```
2. Copy the node host names.

OR

1. In the AWS UI, go to your Auto Scaling Group.
2. Click on the **Instance Management** tab to see the instance IDs.
3. Click the first instance ID to view the details of the corresponding instance.
4. Note the Private DNS name for the instance.
5. Repeat steps 3-4 for each instance ID.

After determining these values, ensure that your SmartConnector, Logger, or ESM instance is added to the the AWS TH cluster Security Group with rules allowing access to the ports 32080, 9093, and optionally 9092.

You can then follow the integration procedures below.

To configure a SmartConnector as a Transformation Hub consumer (not using SSL/TLS):

1. Log in to Logger.
2. Select **Configuration > Receivers > Add.**
3. In the **Add Receiver** dialog, enter the following:
   - **Name:** Enter a unique name for the new receiver.
   - **Type:** Transformation Hub Receiver
4. Select and edit the Transformation Hub Receiver and enter the following parameters:
   - **Transformation Hub host(s) and port:** {Kafka broker Host IP 1}:9092, {Kafka broker Host IP 2}:9092, {Kafka broker Host IP 3}:9092
   - **Event Topic List:** th-cef (If additional topics are needed, enter multiple topics with a comma-separated list.)
   - **Retrieve event from earliest offset:** true
   - **Consumer Group (Logger Pool):** Logger Pool
   - **Use SSL/TLS:** false
   - **Use Client Authentication:** false
   - **Enable:** Checked

To configure a Logger as a Transformation Hub consumer (using SSL/TLS):

1. Log in to Logger.
2. Select **Configuration > Receivers > Add.**
3. In the **Add Receiver** dialog, enter the following:
   - **Name:** Transformation Hub Receiver
   - **Type:** Transformation Hub Receiver
4. Select and edit the Transformation Hub Receiver and enter the following parameters:
   - **Transformation Hub host(s) and port:** {Kafka broker Host IP 1}:9093, {Kafka broker Host IP 2}:9093, {Kafka broker Host IP 3}:9093
   - **Event Topic List:** th-cef (You can enter multiple topics with a comma-separated list.)
   - **Retrieve event from earliest offset:** true

- **Consumer Group (Logger Pool):** Logger Pool
- **Use SSL/TLS:** true
- **Use Client Authentication:** false
- **Enable:** Checked

To configure a Logger as a Transformation Hub consumer (using SSL/TLS with Client Authentication):

1. Log in to Logger.
2. Select **Configuration > Receivers > Add.**
3. In the **Add Receiver** dialog, enter the following:
   - **Name:** Transformation Hub Receiver
   - **Type:** Transformation Hub Receiver
4. Select and edit the Transformation Hub Receiver and enter the following parameters:
   - **Transformation Hub host(s) and port:** {Kafka broker Host IP 1}:9093, {Kafka broker Host IP 2}:9093, {Kafka broker Host IP 3}:9093
   - **Event Topic List:** th-cef (You can enter multiple topics with a comma-separated list.)
   - **Retrieve event from earliest offset:** true
   - **Consumer Group (Logger Pool):** Logger Pool
   - **Use SSL/TLS:** true
   - **Use Client Authentication:** true
   - **Enable:** Checked

Troubleshooting

The following troubleshooting tips might be useful in diagnosing Logger integration issues.

Error Message	Issue
IP Address th1.example.com is not a valid address	Use IP addresses in Receiver configuration, not host names.
There was a problem contacting Transformation Hub: Timeout expired while fetching topic metadata, please check the receiver configuration	Logger can’t communicate with Transformation Hub because of routing or network issues.
The specified Event Topic (`th-<topicname>`) is not valid	You are specifying an incorrect or non-existent the topic name.

> This process is explained in more detail in the Logger Administrator's Guide, available from the Micro Focus support community.

Configuring ESM as a Transformation Hub Consumer

This procedure describes how to configure ESM as a Transformation Hub consumer with client authentication using a User (intermediate) certificate.

Azure Integration - Additional Steps

For proper integration with Azure Transformation Hub, you must perform the following additional procedures for the ArcSight product (ArcMC, SmartConnector, CTH, Logger, or ESM) you are integrating.

- Edits to the `/etc/hosts` file
- Peering
- Configure health probes and load-balancing rules for ports 32080, 9093, and 9092. (For ESM, this applies only to port 9093.)

Each of these is detailed below.

Edit the `/etc/hosts` File

You must add each instance to the product's `/etc/hosts` file.

1. Open the product's `/etc/hosts` file in a text editor.
2. Add the internal IP and FQDN for each instance in the Azure Kubernetes service. (The instance IP and FQDN can be obtained by opening the AKS resource group and then opening the aks-nodepool virtual machine scale set.)
3. Save the changes to the file.

Example additions to `/etc/hosts`:

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
```

```
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

```
10.1.1.4 aks-nodepool1-12400006-vmss000000
```

```
10.1.1.5 aks-nodepool1-12400006-vmss000001
```

```
10.1.1.6 aks-nodepool1-12400006-vmss000002
```

Peering

If the Azure product and Azure Transformation Hub are on different VLANs, you must configure peering between the two VLANs. An example is provided in the section Peering Virtual Networks.

Health Probes

You must configure health probes and load balancing rules for ports 32080, 9093, and, optionally, 9092. Configuring health probes for these ports is detailed in Configuring Port Rules for Product Integration. Use the same procedure as outlined for port 443.

> Some of the commands shown here will require `root` user privileges.

AWS Integration - Additional Steps

For proper integration with AWS Transformation Hub, you must perform the following additional procedures for the ArcSight product (SmartConnectors, Logger, or ESM) you are integrating.

For configuring SmartConnectors, Loggers, and ESMs: obtain the cluster worker node (Kafka broker node) host names, using one of the following procedures:

1. From the bastion host, run the following command:
   ```
   # kubectl get nodes
   ```
2. Copy the node host names.

OR

1. In the AWS UI, go to your Auto Scaling Group.
2. Click on the **Instance Management** tab to see the instance IDs.
3. Click the first instance ID to view the details of the corresponding instance.
4. Note the Private DNS name for the instance.
5. Repeat steps 3-4 for each instance ID.

After determining these values, ensure that your SmartConnector, Logger, or ESM instance is added to the the AWS TH cluster Security Group with rules allowing access to the ports 32080, 9093, and optionally 9092.

You can then follow the integration procedures below.

To configure ESM as a Transformation Hub consumer:

1. On Transformation Hub, run the command:

```
   # ${K8S_HOME}/scripts/cdf-updateRE.sh write --re-key={path to intermediate
certificate}/intermediate.key.pem --re-crt={path to intermediate
certificate}/intermediate.cert.pem --re-ca={path to intermediate
certificate}/ca.cert.pem
```

2. On an ESM host which has not been configured as a Transformation Hub consumer, switch to the manager directory:
   `# cd /opt/arcsight/manager`
3. Run each of these commands, one at a time. When prompted by the keytool for a password, enter the ESM password.

```
   # touch config/client.properties
```

```
   # bin/arcsight keytool -store clientkeys -storepasswd -storepass ""
```

```
   # bin/arcsight keytool -store clientkeys -keypasswd -keypass "" -alias
services-cn
```

```
   # bin/arcsight changepassword -f config/client.properties -p
ssl.keystore.password
```

4. Import the intermediate certificates to the ESM client keystore.
5. Run these commands:

```
   # bin/arcsight keytool -store clientcerts -importcert -file
/tmp/ca.cert.pem -alias thcert
```

```
   # bin/arcsight keytool -store clientkeys -importcert -file
/tmp/intermediate.cert.pem -alias thintcert
```

```
   # bin/arcsight keytool -store clientcerts -importcert -file
/tmp/intermediate.cert.pem -alias thintcert
```

```
   # /etc/init.d/arcsight_services stop manager
```

```
   # bin/arcsight keytool -store clientkeys -genkeypair -dname "cn=<your
CN>,ou=<your OU>, o=<your org short name>, c=<your country>" -keyalg rsa -
keysize 2048 -alias th -startdate -1d -validity 366
```

```
   # bin/arcsight keytool -certreq -store clientkeys -alias th -file
thkey.csr
```

6. Generate a certificate signing request (`.csr` file) so the Transformation Hub can sign a client certificate.

7. Copy the `.csr` file to the Transformation Hub initial master node.
8. On the Transformation Hub Initial Master Node, run the command:

```
# openssl x509 -req -CA /opt/intermediate_cert_files/intermediate.cert.pem -
CAkey /opt/intermediate_cert_files/intermediate.key.pem -in /opt/thkey.csr -
out /opt/signedTHkey.crt -days 3650 -CAcreateserial -sha256
```

9. Copy the signed certificate to `/tmp` on the ESM host.
10. On the ESM host import the signed client certificate into the client keystore so it can be used to authenticate to Transformation Hub. Run these commands:

```
# /opt/arcsight/manager/bin/arcsight keytool -store clientkeys -alias th -
importcert -file /tmp/signedTHkey.crt -trustcacerts
```

11. Start the manager configuration wizard:

```
# /opt/arcsight/manager/bin/arcsight managersetup
```

> If on a host without X Window access, run the `managersetup` command with the -i parameter. Consult the ESM documentation for more information regarding the `managersetup` command.

9. Proceed through the wizard for adding the Transformation hub to the ESM, until the dialog is displayed that prompts for a connection to Transformation Hub. On the dialog, under **“ESM can consume events from a Transformation Hub…”**, enter *Yes*. Then enter then the following parameters. (This will put an entry in the Manager `cacerts` file, displayed as `ebcaroot`):

Host:Port(s): `th-broker-hostname1:9093,th-broker-hostname2:9093,th-broker-hostname3:9093`

> **Note:** You must use host names, not IP addresses. In addition, ESM does not support non-TLS port 9092.

Topics to read from: `th-binary_esm` and Avro topics.

Path to Transformation Hub root cert:{leave this empty}

8. On the ESM, restart the ESM Manager:

```
  # /etc/init.d/arcsight_services stop manager
```

```
  # /etc/init.d/arcsight_services start manager
```

Managing Transformation Hub

This section provides guidance for managing Transformation Hub functions and features within the deployment.

Maintaining an On-Premises Transformation Hub

This chapter contains the following sections:

Adding a New Worker Node to an On-Premises Cluster

You can add a new worker node to an existing on-premises Transformation Hub installation.

To add a new worker node:

1. Set up and provision the new node according to the guidelines and system requirements given here. Note the IP address of the new node for use in the following procedures.
2. Modify your NFS server settings to add the new node to the `/etc/exports` file.

> Refer to the NFS server section of the CDF Planning Guide for more information.

3. Run the following command to update the shared volumes:
   ```
   exportfs -ra
   ```
4. Log in to the CDF Management Portal (`https://<ha-address>:5443`).
5. Click **Cluster > Nodes.**
6. Click **+ Add.**
7. Enter values for the pop-up dialog. For host name, use the FQDN of the new node.
8. Click **ADD.**

Now log into the CDF Management Portal and add the appropriate labels to the new worker node.

Uninstalling a Master or Worker Node from an On-Premises Cluster

To uninstall an existing master or worker node from an on-premises cluster, open an SSH connection to the node and run the following commands.

```
  # cd $k8s-home
  # ./uninstall.sh
```

Then, reboot the node to complete node removal.

When removing the node from the cluster, make sure that the cluster will still have enough resources to host the product workload without the node you are removing . Also, make sure that you have sufficient nodes labeled with the product labels.

Effects on the Cluster

If a worker node is uninstalled, all events data will be stored on the node by default under `/opt/arcsight/k8s-hostpath-volume/th/kafka`.

If a master node is stopped or uninstalled, that node will be reported as unavailable to the cluster. All other functionality, including events processing on the worker nodes, will continue.

> ✓ From a multi-master cluster with 3 master nodes, you can safely remove only one master node. By removing one of three master nodes you will lose high availability, but the cluster will continue to function. If you remove two of three master nodes, the cluster might become unavailable, and you will then need to set up the cluster from scratch.

Removing a Crashed Worker Node

In case of a worker node failure, do the following:

1. Add a new worker node to replace the failed node before removing the crashed one.
2. Run the following command on one of the healthy nodes to delete the crashed node's IP address from the cluster:

```
    # kubectl delete node <crashed_node_ip_FQDN>
```

> This action needs to be performed manually by the cluster administrator, because there is no way for the cluster to distinguish permanent node failure from temporary network connectivity outage, restart or similar events.

When this command is run, the cluster re-schedules the stateful containers (Kafka, ZooKeeper, routing stream processors) to the remaining machines matching the container requirements (labels, resources).

Adding ZooKeeper Instances

Adding a ZooKeeper instance has the following prerequisites:

- You will need at least 2 available worker nodes already deployed that do not already have a Kafka broker or ZooKeeper instance deployed on them. (Only one ZooKeeper can be installed for each worker node, and there must be an odd number of ZooKeepers --1, 3, 5, 7, and so on. Therefore you need at least 2 additional worker nodes to keep the total number an odd one.)
- Any new node where a ZooKeeper instance is deployed should be labeled `zk:yes` (for on-premises installation) or `zk=yes` (for cloud installation).

- If you plan to deploy both Kafka brokers and ZooKeepers, it is recommended that you perform this procedure to add ZooKeeper instances *before* you have deployed your Kafka brokers.

To add new ZooKeeper instances:

1. Open the CDF Management Portal.
2. Click **... (Browse)** to the right.
3. From the drop-down, select **Reconfigure**. The post-deployment settings page is displayed.
4. Find the field *# of ZooKeeper nodes in the ZooKeeper cluster.*
5. From the field's drop-down, select the new number of ZooKeeper instances.
6. Click **Save**.
7. Verify the new ZooKeeper pods are up and in Running state by running the command:
   ```
   # kubectl get pods -n {arcsight_namespace_id} th-zookeeper-x
   ```

> Reducing the number of ZooKeeper instances is not currently supported.

Adding a Kafka Broker Instance for Consistency with the Zookeeper

Adding a new Kafka broker has the following prerequisites:

- You will need an available worker node already deployed that does not already have a Kafka broker or ZooKeeper instance deployed on it.
- The new node where a Kafka broker is deployed should be labeled `kafka:yes` (on-premises installation), or `kafka=yes` (for cloud installation).
- If you plan to deploy both Kafka Brokers and ZooKeepers , it is recommended that you perform this procedure to add Kafka brokers after you have deployed your ZooKeepers and they are up and running.

To add a Kafka broker:

1. Open the CDF Management Portal.
2. Click **... (Browse)** to the right.
3. From the drop-down, select **Reconfigure**. The post-deployment settings page is displayed.
4. Find the field *# of Kafka broker nodes in the ZooKeeper cluster.*
5. From the field's drop-down, select the new number of Kafka brokers.
6. Adjust any other related fields as needed. For example, if the topic replication factor is 1, consider increasing it.
7. Click **Save**.

8. Verify that the new Kafka brokers are up and in Running state by running the command:
   `kubectl get pods -n {arcsight_namespace_id} th-kafka-x`

Next, assign partitions to the new Kafka broker:

1. Connect to Transformation Hub Kafka Manager.

> Refer to Connecting to the Kafka Manager for more information.

2. In **Cluster > Transformation Hub > Topics,** click **Generate Partition Assignments.**
3. On the **Confirm Assignments** page, confirm partition assignments for the new broker and click **Generate Partition Assignments.**
4. On the main toolbar, click **Topic > List.**
5. Click **Run Partition Assignments.**
6. On the **Run Assignments** page, confirm partition assignments and click **Run Partition Assignments.**
7. The partition reassignment process begins. On the Reassign Partitions page, under **Status**, check for a date and time of the job completion to verify completion of the task.

> Reducing the number of Kafka brokers is not currently supported.

Managing Transformation Hub through ArcMC

After configuring ArcMC to manage your Transformation Hub, you can create topics and routing rules, monitor metrics, and receive notifications about Transformation Hub status through ArcSight Management Center (ArcMC).

Monitored Transformation Hub parameters include CPU usage, memory, event parsing errors, stream processing EPS, and stream processing lag.

To manage a Transformation Hub in ArcMC, add your Transformation Hub as a host to ArcMC. The procedure for adding Transformation Hub as a host is explained in detail in the *Micro Focus ArcSight Management Center Deployment Guide*, available from the Micro Focus support community.

> A single ArcMC can manage only a single Transformation Hub cluster, while a single Transformation Hub can be managed by up to 2 ArcMCs.

The *Micro Focus ArcSight Management Center Administrator's Guide* also explains in detail how to view the status of Transformation Hub consumers, as well as how to manage topics, routing rules, and monitored metrics.

Changing Transformation Hub Security Mode

You should decide on a security mode for Transformation Hub prior to deployment and setup. In general, the security mode of systems connected to Transformation Hub (consumers and producers) must be the same as the Transformation Hub security mode.

TLS is the default security mode. Optional modes include TLS with Client Authentication, as well as FIPS. A TLS performance impact is a known Kafka behavior. Exact details of the impact will depend on your specific configuration, but could reduce the event rate by half or more.

To protect against unknown clients sending events to Transformation Hub, or changing Avro topic schemas, enabling Client Authentication is recommended.

You can change the Transformation Hub security mode after deployment, but this will cause downtime for your Transformation Hub and associated systems, such as consumers and producers.

You will need to make sure all Transformation Hub-associated systems are re-configured as well. If the security mode change requires that Transformation Hub consumer or Transformation Hub producer restarts, the *producer or consumer must be disconnected from Transformation Hub first.* Consult the appropriate consumer or producer documentation for details.

> For an Azure cluster, please run all Azure Cloud Shell (command line) and `kubectl` commands from the authorized jump host. For an AWS cluster, all comments should be run from the bastion.

The process of changing security mode includes the following steps.

> *Undeploying Transformation Hub will remove all previous configuration settings.* Prior to proceeding further, you should make a note of your existing settings and then re-enter these on the pre-deployment configuration page during the re-deployment of the Transformation Hub.

1. Stop SmartConnectors from sending events. This will close connections. See the SmartConnector User Guide for information on stopping SmartConnectors from sending events.
2. Stop all consumers (Logger, ESM, Vertica Scheduler) from consuming from topics in Transformation Hub. (There is no need to clear out existing messages from the topics, and the consumers will continue from the last offset later.)
3. Log in to the CDF Management Portal.
4. Click **Administration**.
5. Click the **...** (Browse) icon to the right of the main window.

6. From the drop-down, click **Uninstall**. The post-deployment settings page is displayed.
7. Uninstall the Transformation Hub.
8. Follow the consumer and producer documentation to reconfigure those applications to align their security modes to be the same as Transformation Hub.
9. Redeploy the Transformation Hub with the appropriate security mode configured.
10. Reconnect the consumers and producers to the Transformation Hub.

Maintaining a Transformation Hub on Azure

Maintenance of a Transformation Hub on Azure is performed using the cluster jump host. You can do one of the following:

- RDP to log on to the jump host desktop and accessing the CDF portal on port 5443, or,
- Run `kubectl` commands from the jumphost CLI.

This chapter contains the following sections:

Enabling Access to Kafka Manager

Kafka Manager is the management tool used for maintenance, management, and monitoring of topics, partitions, consumers, and Kafka brokers.

To enable access to Kafka Manager on Azure:

1. Patch the load balancer by running the following command:

   ```
   # kubectl patch svc -n arcsight-installer-<th-namespace id> th-kafkamgr-
   svc -p '{"spec": { "type": "NodePort", "ports": [ { "nodePort": {local
   port number}, "port": {cluster port number}} ] } }'
   ```

For example:

```
# kubectl patch svc -n arcsight-installer-abcfn th-kafkamgr-svc -p '{"spec":
{ "type": "NodePort", "ports": [ { "nodePort": 32090, "port": 9000} ] } }'
```

2. In Azure, create a health probe and load balancing rule for port 32090. Follow the same procedure outlined for port 443 in Configuring the Load Balancer.
3. RDP to the jumphost associated with the cluster.
4. To forward the ports, run the following command on the jump host, and then log in using the jumphost password:

   ```
   # ssh -f -N localhost -L {any port}:{your load balancer hostname/ip}:32090
   ```

For example:

```
# ssh -f -N localhost -L 1234:cdf-dns.arcsight.private.com:32090
```

To open Kafka Manager:

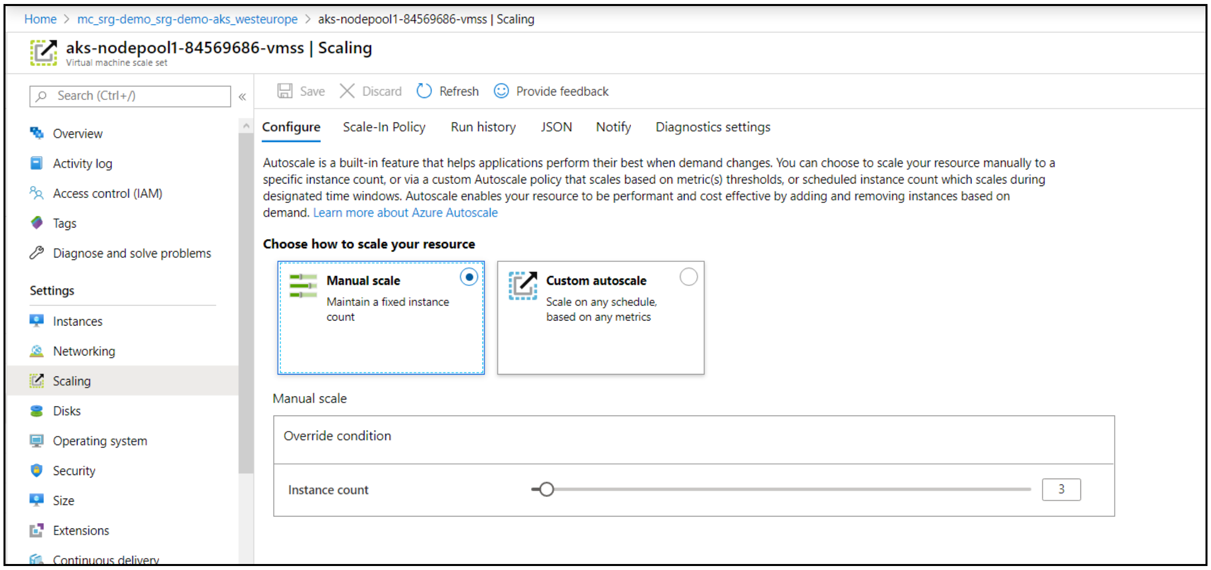
1. RDP to the jump host.
2. Open a browser on the jump host and log in to:
   `localhost:1234`

Scaling a Cluster Up

You can increase the number of nodes to an Azure cluster using either the Azure Portal or Azure Cloud Shell. After increasing the number of nodes, you must then label all new nodes.

To add nodes to a cluster using the Azure Portal:

1. In the Azure Portal, locate the Azure Kubernetes resource group. (The AKS resource group name is in the format MC_<your_resource_group>_<aks_name>_<location>.)
2. Open the virtual machine scale set. (The scale set name is in the format *aks-nodepool1-<NUMBER>-vmss.)*
3. Under **Settings**, click **Scaling**.
4. In **instance count**, increase the value to the desired number of nodes in the cluster.



Autoscale is a built-in feature that helps applications perform their best when demand changes. You can choose to scale your resource manually to a specific instance count, or via a custom Autoscale policy that scales based on metric(s) thresholds, or scheduled instance count which scales during designated time windows. Autoscale enables your resource to be performant and cost effective by adding and removing instances based on demand. Learn more about Azure Autoscale

Choose how to scale your resource

● **Manual scale** Maintain a fixed instance count

○ **Custom autoscale** Scale on any schedule, based on any metrics

Manual scale

Override condition

Instance count: 3

To add nodes to a cluster using the Azure Cloud Shell or jumphost CLI:

1. Get the AKS resource group and store it in an environment variable for later usage:
   ```
   # CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group <RESOURCE GROUP> -
   -name <AKS NAME> --query nodeResourceGroup -o tsv)
   ```

For example:

```
# CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group srg-demo --name
srg-demo-aks --query nodeResourceGroup -o tsv)
```

2. Get the AKS Virtual machine scale set by running the following command:
   ```
   # VMSS=$(az vmss list -g $CLUSTER_RESOURCE_GROUP | jq -r .[0].name)
   ```

3. Get the current number of nodes by running the following command:
   ```
   # az vmss list -g $CLUSTER_RESOURCE_GROUP | jq -r '.[0].sku.capacity'
   ```

4. Add instances by increasing the value of the `new-capacity` parameter to reflect the new number of instances. For example:
   ```
   # az vmss scale --resource-group $CLUSTER_RESOURCE_GROUP --name $VMSS --
   new-capacity 5
   ```

To label new nodes:

1. Get a list of all nodes by running the following command:
   ```
   # kubectl get nodes
   ```
2. Select one of the new nodes and label it by running the following command:
   ```
   # kubectl label node <virtual_machine_scale_set> role=loadbalancer
   Worker=label <<label1> <label2>... <labelN>
   ```

Parameters:

- `role=loadbalancer` and `Worker=label` are mandatory labels
- `label1` through `...labelN` represent any number of optional labels depending on node usage. For more information about labels, see Labeling Nodes.

> For example:
> ```
> kubectl label node aks- nodepool1- 84569686- vmss000003 role=loadbalancer
> Worker=label zk=yes kafka=yes th-platform=yes th-processing=yes
> ```

3. Repeat Step 2 for each of the other new nodes.
4. If any of the new nodes contain ZooKeeper and Kafka pods, new partitions must be assigned. Launch the Kafka Manager.
5. Click the **Topics** list.
6. Click **Generate New Partition Assignments.**
7. Assign the desired topics to the new nodes.
8. Click **Run Partition Assignments.**

Peering Virtual Networks

Peering enables services from different virtual networks to communicate with one another using private IP addresses. This section discusses how to peer two Azure virtual networks; for instance, peering an AKS virtual network with a SmartConnector or other ArcSight product.

You should consult the Azure documentation on peering virtual networks for precise commands. The procedure here is provided as an example.

In this peering example, the virtual network (vnet) demo-vnet, from the resource group srg-demo, will be peered with vnet qaprg-vnet from resource group qaprg. Peering will be performed using the Azure Cloud Shell.

To set up peering between these two example virtual networks using the Azure Cloud Shell:

1. Get the ID for virtual network demo-vnet from resource group srg-demo, and store it in variable `vNet1Id`:
   ```
   # vNet1Id=$(az network vnet show --resource-group srg-demo --name demo-
   vnet --query id --out tsv)
   ```
2. Get ID for virtual network qaprg-vnet from resource group qaprg, and store it in variable `vNet2Id`:
   ```
   # vNet2Id=$(az network vnet show --resource-group qaprg --name qaprg-vnet
   --query id --out tsv)
   ```
3. Establish peering for vnet demo-vnet from resource group srg-demo to remote virtual network (ID in `$vNet2Id`) with the following command:
   ```
   # az network vnet peering create --name demo-vnet-to-qaprg-vnet --
   resource-group srg-demo --vnet-name demo-vnet --remote-vnet $vNet2Id --
   allow-vnet-access
   ```

where `name` parameter is symbolic; you can choose a value for this as desired.

4. To establish conection, peering must also be established from qaprg-vnet to demo-vnet. Run the following command:
   ```
   # az network vnet peering create --name qaprg-vnet-to-demo-vnet --
   resource-group qaprg --vnet-name qaprg-vnet --remote-vnet $vNet1Id --
   allow-vnet-access
   ```
5. To verify the establishment of peering, run the following command:
   ```
   # az network vnet peering show --name demo-vnet-to-qaprg-vnet --resource-
   group srg-demo --vnet-name demo-vnet --query peeringState
   ```

> Change the name to the same name used in Step 4, and use your vnet and resource group.

6. If peering has been established successfully, then `Connected` will show as a result.

Configuring Health Probes and Load Balancing Rules for Product Integration

You must configure health probes and load-balancing rules for product integration. These ports include 9093 and, optionally, 9092.

Add a Health Probe for port 9093

1. On the Azure Portal, find your Azure kubernetes resource group (name format *MC_<you_resource_group>_<aks_name>_<location>*).
2. Open the AKS.
3. Find the entry for the Kubernetes load balancer, and then open it.
4. Click **Health probes.**
5. Click **+Add**to add a Kubernetes load balancer health probe. Then, assign these values:
   - **Name:** Assign a name to the probe.
   - **Protocol:** Select TCP
   - **Port:** Enter 9093.

Add Load Balancing Rule for Port 9093

1. Open the Kubernetes load balancer and click **Load balancing rules.**
2. Click **+ Add** to add a new Kubernetes load balancing rule. Then, assign these values:
   - **Name:** Assign a name to the rule.
   - **Port:** Enter 9093.
   - **Backend port:** 9093
   - **Health probe:** Select the port 9093 health probe you just created.
   - **Frontend IP Address:** Use the Public IP address you prepared earlier.

   ### Add a Health Probe for port 9092

   a. On the Azure Portal, find your Azure kubernetes resource group (name format *MC_<you_resource_group>_<aks_name>_<location>*).

   b. Open the AKS.

   c. Find the entry for the Kubernetes load balancer, and then open it.

   d. Click **Health probes.**

e. Click **+Add**to add a Kubernetes load balancer health probe. Then, assign these values:
   - **Name:** Assign a name to the probe.
   - **Protocol:** Select TCP
   - **Port:** Enter 9092.

Add Load Balancing Rule for Port 9092

a. Open the Kubernetes load balancer and click **Load balancing rules.**

b. Click **+ Add** to add a new Kubernetes load balancing rule. Then, assign these values:
   - **Name:** Assign a name to the rule.
   - **Port:** Enter 9092.
   - **Backend port:** 9092
   - **Health probe:** Select the port 9092 health probe you just created.
   - **Frontend IP Address:** Use the Public IP address you prepared earlier.

Removing a Product

To remove a product (capability) from your cluster:

1. Click **Deployment > Deployments.**
2. Click **... (Browse)** on the far right and choose **Install**. A new screen will be opened in a separate tab.
3. On the next page, deselect the product you want to remove, and click **Next**
4. On the **File Storage** page click **Next**.
5. Update configuration values if needed, and click **Next**.

After a short wait, the **Configuration Complete** page confirms the change to the cluster.

Uninstalling CDF on Azure

There are several options for uninstallation.

- You can uninstall any installed products.
- You can uninstall CDF and any installed products. Perform this installation if you plan to re-use the cluster and re-install CDF later.
- You can uninstall (and destroy) all resources created during platform setup. Only perform this option when the cluster is no longer needed.

To uninstall CDF and installed products:

1. Log in to the CDF Management portal and uninstall the suite, following the procedure outlined in Uninstalling the Suite.
2. Log in to the jump host and become `root`.
3. Get all namespaces by running the command:
   `# kubectl get namespaces`

For example:

```
# kubectl get namespaces

NAME                       STATUS   AGE

arcsight-installer-blk62   Active   41m

core                       Active   48m

default                    Active   84m

kube-public                Active   84m

kube-system                Active   84m
```

4. Delete product namespaces and the core namespace by running the command:
   `# kubectl delete namespace <namespace name>`

For example:

```
# kubectl delete namespace arcsight-installer-blk62

namespace "arcsight-installer-blk62" deleted

# kubectl delete namespace core

namespace "core" deleted
```

> Your own product namespace will have the name format `arcsight-installer-XXXXX`.

5. Wait for the selected namespaces to be deleted before continuing.
6. Get all PV (persistent volumes) by running the command::
   `# kubectl get pv`

For example:

```
# kubectl get pv

NAME                                        CAPACITY   ACCESS MODES   RECLAIM
POLICY   STATUS
```

```
arcsight-installer-blk62-arcsight-volume   30Gi       RWX            Retain
        Released
```

```
arcsight-installer-blk62-db-backup-vol     1Mi        RWX            Retain
        Released
```

```
db-single                                  10Gi       RWX            Retain
        Released
```

```
itom-logging                               1Mi        RWX            Retain
        Released
```

```
itom-vol                                   5Gi        RWX            Retain
        Released
```

7. Delete all PVs by running the following command for each PV:
   `# kubectl delete pv <PV_name>`

```
# kubectl delete pv arcsight-installer-blk62-arcsight-volume
```

```
persistentvolume "arcsight-installer-blk62-arcsight-volume" deleted
```

```
# kubectl delete pv arcsight-installer-blk62-db-backup-vol
```

```
persistentvolume "arcsight-installer-blk62-db-backup-vol" deleted
```

```
# kubectl delete pv db-single
```

```
persistentvolume "db-single" deleted
```

```
# kubectl delete pv itom-logging
```

```
persistentvolume "itom-logging" deleted
```

```
# kubectl delete pv itom-vol
```

```
persistentvolume "itom-vol" deleted
```

8. Clear the data from your NFS volumes by connecting twith SSH and clearing (but **not** deleting) all exported directories.

> If you deleted the SSH inbound rule, you will need to add it again to be able to SSH to your NFS.

CDF can now be installed again on the cluster.

Maintaining a Transformation Hub on AWS

Maintenance of a Transformation Hub on AWS is performed through the bastion host.

This chapter contains the following sections:

Enabling Kafka Manager Access

Kafka Manager is the management tool used for maintenance, management, and monitoring of topics, partitions, consumers, and Kafka brokers.

Enabling access to Kafka manager has the following steps:

1. Patch the Kafka node port service to allow access to the Kafka Manager from outside the Kubernetes cluster.
2. Create an AWS target group and a listener for the Kafka Manager node port.
3. Register the target group with all cluster nodes.
4. Create a listener on the Application Load Balancer (ALB).
5. Log in to the Kafka Manager.

Each of these steps, which are performed using the AWS CLI run on your bastion for the cluster, are explained in detail below.

To patch the Kafka node port service:

1. Run the following `kubectl` patch command on the bastion host:
   ```
   # kubectl patch svc -n arcsight-installer-{namespace id} th-kafkamgr-svc \
   -p '{"spec": { "type": "NodePort", "ports": [ { "nodePort":
   <LOCAL PORT NUMBER>, "port": <CLUSTER PORT NUMBER>} ] } }'
   ```

Example input and output:

```
# kubectl patch svc -n arcsight-installer-7ozvh   th-kafkamgr-svc  \
-p '{"spec": { "type": "NodePort", "ports": [ { "nodePort": 32090, "port":
9000 } ] } }'
```

```
service/th-kafkamgr-svc patched
```

```
[centos@ip-10-0-1-49 ~]$
```

To create a target group and listener for the node port (32090):

1. Get the values for the following:
   ```
   # PORT: in this example we use 32090
   # TARGET_GRP_NAME: example: th-port-32090-tg
   # ALB_ARN: <your ALB ARN>
   # VPC_ID: <your VPC ID>
   ```

   > Be sure to use HTTP protocol for the kafka manager.

2. Run the following command:
   ```
   # aws elbv2 create-target-group \
   --name $<TARGET_GRP_NAME> \
   --protocol HTTP \
   --port <PORT> \
   --vpc-id $<VPC_ID> \
   --health-check-protocol HTTP \
   --target-type instance
   ```

Example input and output:

```
# aws elbv2 create-target-group \
--name "th-port-32090-tg" \
--protocol HTTP \
--port 32090 \
--vpc-id vpc-0656f4932319ff12d\
--health-check-protocol HTTP \
--target-type instance
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-
2:115370848038:targetgroup/th-port-32090-tg/57bbea6f22c02aec",
            "TargetGroupName": "th-port-32090-tg",
            "Protocol": "HTTP",
            "Port": 32090,
            "VpcId": "vpc-0656f4832319ff11c",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": <
                "HttpCode": "200"
            },
```

```
            "TargetType": "instance"
        }
    ]
}
```

To register the target group with all cluster nodes:

1. Get the instance id's by running the following cmd from the bastion:
   ```
   # aws autoscaling describe-auto-scaling-instances \
    | jq -r '.AutoScalingInstances[] | select(.AutoScalingGroupName="< your
   cluster name>").InstanceId'
   ```

Example input and output:

```
# aws autoscaling describe-auto-scaling-instances   \
| jq -r '.AutoScalingInstances[] | select(.AutoScalingGroupName="test-
cluster").InstanceId'
i-0151613c774ef16e1
i-0a0c1f2ab38f0e7a6
i-0d5c41b6780f02e8a
```

2. Register the new target group with all nodes by running the following command:
   ```
   # register=$(aws elbv2 register-targets \
   --target-group-arn $target_grp_arn \
   --targets Id="$<INSTANCE_1>" \
   Id="$<INSTANCE_2>" \
   Id="$<INSTANCE_3>")
   ```

Example:

```
aws elbv2 register-targets \
--target-group-arn "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-
2:115370848038:targetgroup/th-port-32090-tg/57bbea6f22c02aec" \
--targets Id="i-0151613c774ef16e1" \
Id="i-0a0c1f2ab38f0e7a6" \
Id="i-0d5c41b6780f02e8a")
```

To create a listener on the ALB:

> Be sure to use HTTP protocol for the kafka manager.

1. Run the following command:
   ```
   # aws elbv2 create-listener \
   --load-balancer-arn <Your ALB ARN> \
   --protocol HTTP \
   --port <port> \
   --default-actions Type=forward,TargetGroupArn=$target_grp_arn
   ```

Example:

```
# aws elbv2 create-listener \
 --load-balancer-arn <ALB_ARN> \
 --protocol HTTP \
--port 32090 \
--default-actions Type=forward,TargetGroupArn=$target_grp_arn
```

To log in to the Kafka Manager:

1. On your bastion, open an instance of the Mozilla Firefox browser and browse to your ALB on port 32090, using the following URL:
   `http:// <ALB hostname>:32090`

Alternative method:

You can also use port forwarding to log in to the Kafka Manager as follows:

> Be sure to use HTTP protocol for the kafka manager.

1. At the bastion CLI, run the following command:
   ```
   # ssh -i {ssh key} \
   -L 32090:mycluster.arcsight-dev.com:32090 {bastion non-root
   user}@64.188.142.129
   ```

Example:

```
# ssh -i /home/myuser/MyDocuments/aws_keys/aws.pem \
-L 32090:mycluster.arcsight-dev.com:32090 centos@64.188.142.129
```

2. In a browser on your local system, browse to localhost:32090.

Refresh the ECR Credentials in Kubernetes

During the initial CDF installation, the credentials and the URL for the ECR (Elastic Container Repository) have been passed through the environment variables `ECR_USER_NAME, ECR_USER_PASSWORD` and `ECR_URL`. All these values are then stored inside Kubernetes as Docker secrets, and CDF will later use them during installing itself as well as the ArcSight Suite products.

The security policy on `AWS ECR` requires that the `ECR_USER_PASSWORD` is valid for next 12 hours after creation.

There are several scenarios where you might need to refresh the password used for accessing the Docker images stored in the ECR:

- The bootstrap of CDF was performed more than 12 hours before the installation on port 3000.

- You are adding new capability to the suite; images have been uploaded but CDF has not registered this.
- You are performing a suite upgrade.

In such cases, the stored user password needs to be replaced by a new freshly generated.

To refresh the credentials:

1. On the bastion, go to `aws-byok-installer/scripts` to run the following scripts.
2. Generate a new `ecr_credentials` snippet by running the command:
   ```
   # ./upload_images_to_ECR --get-ecr-credentials
   ```
3. Add the retrieved values to the environment:
   ```
   # source ecr_credentials
   ```
4. Run the following script to create a set of JSON files to be applied to the cluster:
   ```
   # ./generate_aws_secret
   ```

> The script has no output, except for the situation when one or more required environment variables (`ECR_USER_NAME`, `ECR_USER_PASSWORD` or `ECR_URL`) are empty.

As a result of the procedure, you will have several new files named `secret_xxxx.json` in the directory where you ran the `generate_aws_secret` script. Apply all of them to Kubernetes by running the following command:

```
# kubectl apply -f secret_xxxx.json
```

Now, in CDF, at the page where it notes missing images, you can let it check again. This time, all required images will be found as long as they have been correctly uploaded to the ECR.

Installing the Kubernetes Metrics Server

The Kubernetes Metrics Server is a service running inside the Kubernetes cluster, which collects various utilization and performance data later used by the ArcSight Management Center (ArcMC).

A detailed guide on how to install Kubernetes Metrics Server service is provided by AWS. It is expected that the version of Kubernetes Metrics Server will evolve as the time goes by.

Removing the Cluster

You can remove the cluster to make it possible to perform a fresh installation of CDF and the ArcSight Suite. As part of the removal, you will remove or clean some cluster-specific resources. However, some of your existing resources are reusable.

Reusable Resources

Many of the resources created during your CDF and ArcSight Suite installation are reusable, and do not need to be removed during the cluster removal. You might find it useful to keep such resources on hand for use with other product suites.

- Launch configuration is not dependent on the CDF installation and is not bound to a VPC. Other users performing installation in the same region could re-use an already existing launch configuration.
- Bastion instance is bound to the VPC created for CDF, and can be used only within this VPC. However, a bastion is a highly re-usable resource for installing and managing other clusters or product suites.
- The Route 53 record set, with its certificate, is not dependent on the installation.

Cluster Removal

As part of removing the cluster, you will perform the following tasks:

- Removal of the Auto Scaling group
- Removal of the EKS control plane
- Cleaning or deleting the EFS/NFS

Each of these procedures is explained below.

Removing the Auto Scaling Group

The AWS Auto Scaling group holds the worker nodes instances. Accordingly, in order to delete the worker nodes, you must delete the Auto Scaling group.

To delete the Auto Scaling group:

1. Run the following command:
   ```
   # aws autoscaling delete-auto-scaling-group --force-delete --auto-scaling-
   group-name <auto-scaling group name from AWS worksheet>
   ```
2. The command has no output, and in the background the deletion instances will start. Check the presence of the Auto-Scaling group by running the following command:
   ```
   # aws autoscaling describe-auto-scaling-groups \
   | jq -r '.AutoScalingGroups[] | select(.AutoScalingGroupName=="<auto-
   scaling group name>") | .AutoScalingGroupName'
   ```

> If the group name is returned, the Auto Scaling group and its instances are not fully deleted including its instances. The process can take around 5 minutes to complete.

3. Once the auto-scaling group and worker nodes are removed, you can check the pods by executing this command on the bastion:
   `# kubectl get pods -A -o wide`

All pods are shown in the Pending state, as they do not have a host to run on, but the Kubernetes control plane still has the cluster definition.

If desired, you can create another Auto Scaling group with a different launch configuration. All the pods will be deployed and started on the new worker nodes. Remember to add respective targets to Target Groups. Any new worker nodes will receive new instance IDs.

Removing the EKS Control Plane

The Kubernetes control plane holds the definitions of services, daemons, deployments, pods, and other resources, including the fully qualified identifier of Docker images in the registry. To clean the AWS infrastructre for a new installation, this control plane needs to be removed as well.

To remove the EKS control plane:

1. Run the following command:
   `# aws eks delete-cluster --name <cluster name from AWS worksheet>`

> The command output is very verbose. An example is given below.

2. Verify the cluster has been deleted by running the command:
   `# aws eks list-clusters | jq -r '.clusters[] | select(.=="<cluster name from AWS worksheet>")'`

   An empty output indicatesthat the cluster has been deleted.

Example output of a cluster in the process of being deleted:

```
{
   "cluster":{
      "name":"srgdemo-cluster",
      "arn":"arn:aws:eks:eu-central-1:115370848038:cluster/srgdemo-cluster",
      "createdAt":"2020-08-10T12:13:31.748000+02:00",
      "version":"1.17",
      "endpoint":"https://90842F339FC27B9BE1DD0554E508B914.gr7.eu-central-
1.eks.amazonaws.com",
      "roleArn":"arn:aws:iam::115370848038:role/ARST-EKS-Custom-Role",
      "resourcesVpcConfig":{
         "subnetIds":[
            "subnet-0fb2ebb5882c061f0",
            "subnet-0abd7cd806e04c7be",
            "subnet-0f0cac4ec6837abed"
         ],
         "securityGroupIds":[
            "sg-0ce3c569f73737b77"
         ],
         "clusterSecurityGroupId":"sg-0263ae0d4c33decc4",
         "vpcId":"vpc-0143197ca9bd9c117",
         "endpointPublicAccess":false,
         "endpointPrivateAccess":true,
         "publicAccessCidrs":[

         ]
      },
      "logging":{
         "clusterLogging":[
            {
               "types":[
                  "api",
                  "audit",
                  "authenticator",
                  "controllerManager",
                  "scheduler"
               ],
               "enabled":false
            }
         ]
      },
      "identity":{
         "oidc":{
            "issuer":"https://oidc.eks.eu-central-
1.amazonaws.com/id/90842F339FC27B9BE1DD0554E508B914"
         }
      },
      "status":"DELETING",
      "certificateAuthority":{

"data":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN5RENDQWJDZ0F3SUJBZ0lCQURBT
kJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJd0
1EZ3hNREV3TWpFd01sb1hEVE13TURnd09ERXdNakV3TWxvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVp
YSnVaWFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBSnZXCnUy
MDVNZFJkWUV1VkliQU1yeFJzKzFSMmtyRlhWUmpZd0ZXQUdIRUY2WmJ6V1F2L2Y0d052MmlxaFM0Q
0lJa2wKVTVvTmtaTzFBaU9USk9Ja1l3UFAwdjRGQkNyVFlvU3BldW1xelhqVFBHU2JFUVJ4OXFVM0
ttTkorUXlSZEhJeQpaTHV6b2tXbXJXSG1TVlRLNUxkZUppN3Z4enoweU1TNzczL01GK3FkcVNML2o
1dHJTNEt2cU5ObVRKMEVVY0hwCjdWNklENnFaSEVxZXdKQjl2cmhPdGFlc05TMFdhVWwwUFU4d3pW
aFVUWUlEMllFTU8rOXFsZEdVQVlWTmo3cVIKMUdXVVNVZVVIUWJqNEViMHg4VGhjcDNPYi9oZUNQW
WZ1Rno5MVRWUUR5enRxaDZtUDQvNXFZaW1QeklkaFh5LwpIdDltVmZ3M0tVemlzMURtNk9VQ0F3RU
FBYU1qTUNFd0RnWURWUjBQQVFIL0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0RRWUp
Lb1pJaHZjTkFRRUxCUUFEZ2dFQkFDU2pyZG5Xb0N1WTA4c3pqVU5BSHdnbnFtMDgKZlhydkxtVkxz
SHZiZHFSTmorUTJQMFQvVCtFZFRVWFg5SGNia1JwQU5QNTRkNzRQRmJGbzA0K0dmaTYrTHE5UAoyY
lBzZ2o3Mmo4WWx0V0twVHJiNFpKMnhyZXFsWnZ4MVFZNHpZWUhKdDdKZ1RRaU4xQ2JjaFZLR0V6K0
9nQ3ZTClZGMWE2OEJJajlUMFFDNXgzTTJncHdDa1JMOHArbXkzbkp0Z281Q0JHanhGU2ZHNnN3M0Z
MRXdlRHQyc2dOc1UKV2hpQWZGQmtPdUl2OENmMmlwMUZYQ2toWjJxTXdYanU4UzFFc3Z3bUcrSy9v
d3NiOUFLZG5TaVRQVXJSQWdGbwpsVjBrSGVaK1FpSG5wK0t3a1NpbkoyMVpXRUFMVG5GRjBCR3hYM
DhpU1cwM25Kcy9XemRFdTVFWWhUYz0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo="
      },
      "platformVersion":"eks.1",
      "tags":{
         "user":"user"
      }
   }
}
```

Cleaning or deleting NFS/EFS

Your NFS/EFS is a partially reusable resource. For the EFS you created, you have the following options:

- Leave the existing EFS folder structure intact. A new installation will use a parallel structure. No action needs to be taken.
- Delete and re-create the folder structure during a new installation. The procedure is discussed below.
- Delete the EFS instance completely. The procedure is discussed below.

To delete the folder structure:

1. Log on to the bastion host.
2. Unmount the EFS file system by running the following command:
   ```
   # sudo umount -f /mnt/efs
   ```
3. As a `sudo` user, open the file `/etc/fstab` in a text editor.
4. Locate the following line:
   ```
   # fs-5df66605.efs.eu-central-1.amazonaws.com:/ /mnt/efs nfs4
   nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,_netdev 0
   0
   ```
5. Uncomment the line and then save the file.

6. Mount EFS to the bastion by running the following command:
   ```
   sudo mount -a
   ```
7. Change your current working directory to the mount point, in our case `/mnt/efs`.
8. Delete the whole folder structure by running the following command:
   ```
   # sudo rm -Rf <parent folder from AWS worksheet>
   ```

To delete the EFS instance (not required for re-installing the CDF bootstrap):

1. Delete the mount targets by running the following command on each configured mount target:
   ```
   # aws efs delete-mount-target \
   --mount-target-id <mount target id from AWS worksheet>
   ```
2. Verify the deletion by running the following command:
   ```
   # aws efs describe-mount-targets \
   --file-system-id <filesystem Id from AWS worksheet>
   ```

Example output:

```
{
   "MountTargets":[

   ]
}
```

3. Delete the filesystem by running the command:
   ```
   # aws efs delete-file-system --file-system-id <filesystem Id from AWS
   worksheet>
   ```

Next Steps

At this point the filesystem has been deleted. As explained above, some reusable resources will remain.

- The Application Load Balancer, its listeners, and target groups are not dependent on installation. For a new installation, you will need to add new targets to all target groups.
- The VPC tag marking the EKS cluster has been removed, and the required tag `kubernetes.io/cluster/<cluster name>` has been removed as well. Remember to add it before new installation.

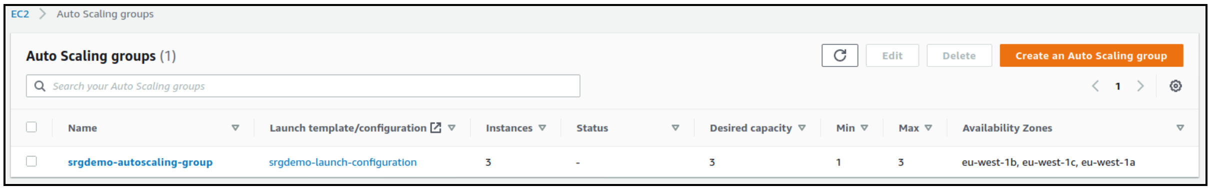
You can now perform a clean installation of a new cluster.

Scaling Up an AWS Cluster

The process of scaling up a cluster involves adding nodes, labeling them as required, and performing some additional configuration. In an AWS cluster, instances are managed by Auto Scaling groups, which include the current number of nodes, as well the upper and lower limits of the nodes for the cluster, and these values must be adjusted to accomodate new instances.

To scale up a cluster using the Web UI:

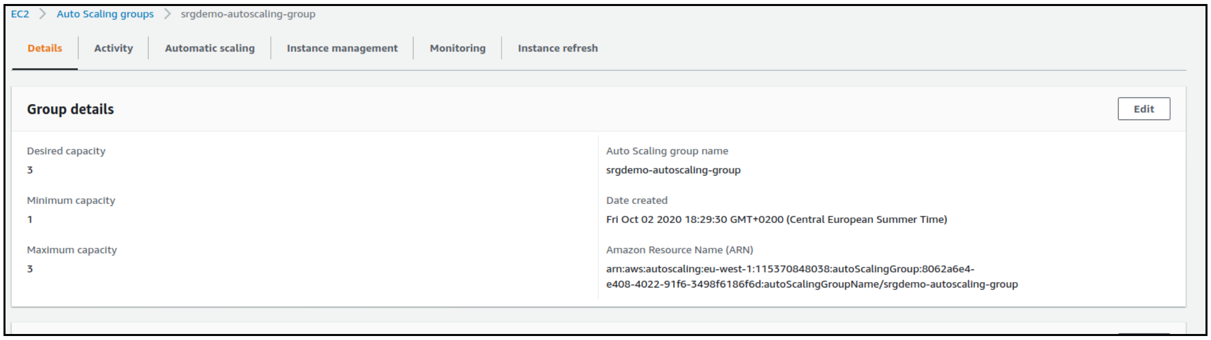
1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Auto Scaling,** click **Auto Scaling Groups**.
3. From the list of **Auto Scaling groups**, click the Auto Scaling group that you previously created.


EC2 > Auto Scaling groups
Search your Auto Scaling groups
Edit
Delete
Create an Auto Scaling group
1

Auto Scaling groups (1)

☐	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
☐	srgdemo-autoscaling-group	srgdemo-launch-configuration	3	-	3	1	3	eu-west-1b, eu-west-1c, eu-west-1a

4. On the **Details** tab, under **Group Details**, click **Edit**.



Group details

Desired capacity
3

Minimum capacity
1

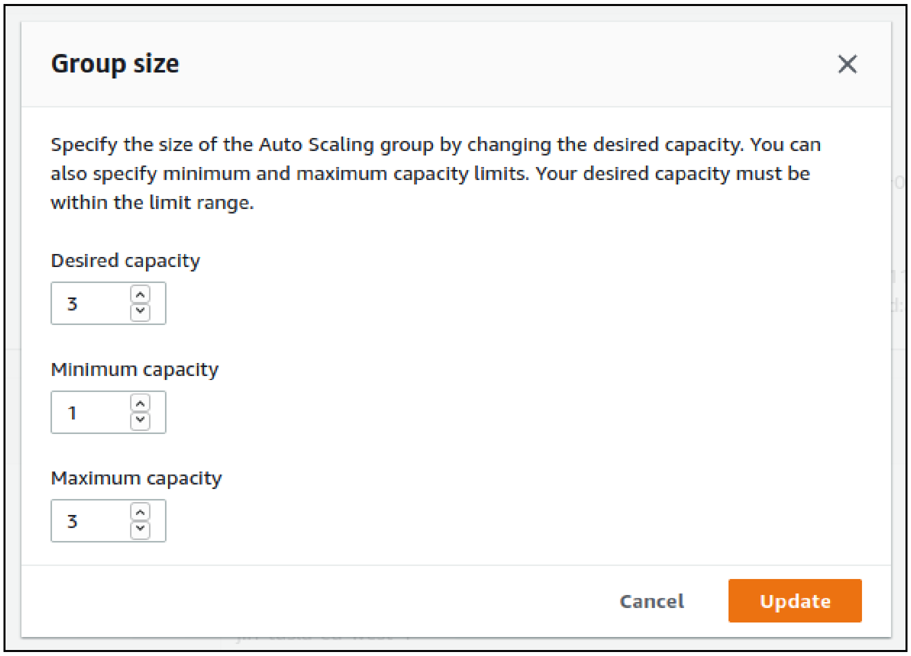
Maximum capacity
3

Auto Scaling group name
srgdemo-autoscaling-group

Date created
Fri Oct 02 2020 18:29:30 GMT+0200 (Central European Summer Time)

Amazon Resource Name (ARN)
arn:aws:autoscaling:eu-west-1:115370848038:autoScalingGroup:8062a6e4-e408-4022-91f6-3498f6186f6d:autoScalingGroupName/srgdemo-autoscaling-group

5. On the **Group Size** dialog, enter values for the cluster's desired, minimum, and maximum capacities.
6. Click **Update**.



Group size

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity: 3

Minimum capacity: 1

Maximum capacity: 3

7. You are returned to the Auto Scaling group details. The group status is shown as *Updating capacity.*

To scale an AWS cluster up using the CLI:

1. Verify that the new number of nodes will fit into the existing limits set on your Auto Scaling group. Run the following command:

```
# aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
<your auto-scaling group name> \
| jq -r '.AutoScalingGroups[0] | "MaxSize: " + (.MaxSize |
tostring),"DesiredSize: " + (.DesiredCapacity | tostring),"MinSize: " +
(.MinSize | tostring)'
```

Example input and output:

```
# aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
srgdemo-autoscaling-group \
| jq -r '.AutoScalingGroups[0] | "MaxSize:         " + (.MaxSize |
tostring),"DesiredCapacity: " + (.DesiredCapacity | tostring),"MinSize:
  " + (.MinSize | tostring)'
```

```
MaxSize:         3
```

```
DesiredCapacity: 3
```

```
MinSize:         1
```

In this example, both the `DesiredCapacity` as well as the `MaxSize` must be increased.

2. Increase the number of nodes in the Auto Scaling group by running this command (command has no output):
   ```
   # aws autoscaling update-auto-scaling-group \
   --auto-scaling-group-name <your auto-scaling group name> \
   --max-size <new maximum size of the cluster> \
   --desired-capacity <desired new number of nodes>
   ```

Example:

In this example, the number of active nodes in our example group `srgdemo-autoscaling-group` is raised to 5, and at the same time the group capacity is increased to 7. The command has no output.

```
# aws autoscaling update-auto-scaling-group \
--auto-scaling-group-name srgdemo-autoscaling-group \
--max-size 7 \
--desired-capacity 5
```

3. To verify the updated values, repeat the command shown in Step 1, above.
4. To verify creation of the new nodes, on the bastion host, run the following command until the desired number of nodes is shown in the Ready state:
   ```
   # kubetlc get nodes
   ```

Labeling New Nodes

You might need to label the new nodes after they are added and ready to use.

- You will not need to add new labels to new nodes if you have previously extended the set of labels for Launch configuration.
- Otherwise, label the new nodes now.

To verify labels on new nodes:

Run the following command:

```
# kubectl get nodes --show-labels
```

Extending Targets in Target Groups

Your newly created nodes are now labeled. Kubernetes might migrate pods used for installation, management and re-configuration to these new nodes as part of cluster operations. You will now need to perform some tasks for the new nodes that you have previously performed for the old nodes.

1. Retrieve instance IDs from auto-scaling group: Retrieve the IDs of all instances in your cluster. Identify the new instances by comparing with existing list of instance IDs. Note any new instance IDs to your AWS worksheet.

2. Repeat the steps in listed in Target Group 3000 - Get CDF Ingress service node port for 3000 and Add targets to target group 3000 to extend routing for installation portal.
3. Repeat the steps in chapter Configure access to CDF management - Get CDF Ingress service node port for 5443 and Add targets to target group 5443 to extend routing for management portal.
4. Repeat the step in chapter Configure access to re-configuration - Add targets to target group 443.

You might use the web UI or CLI to perform these tasks, as desired. Only apply these procedures to your new nodes.

Additional Steps

For Transformation Hub, see the following procedures:

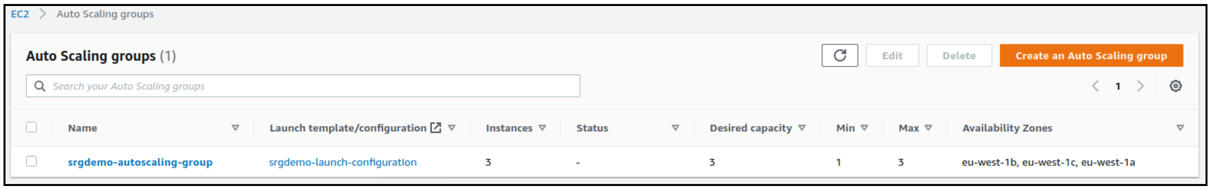
- Adding a Kafka Broker Instance
- Adding a ZooKeeper Instance

Scaling Down an AWS Cluster

In an AWS cluster, instances are managed by Auto Scaling groups, which include the current number of nodes, as well the upper and lower limits of the nodes for the cluster.

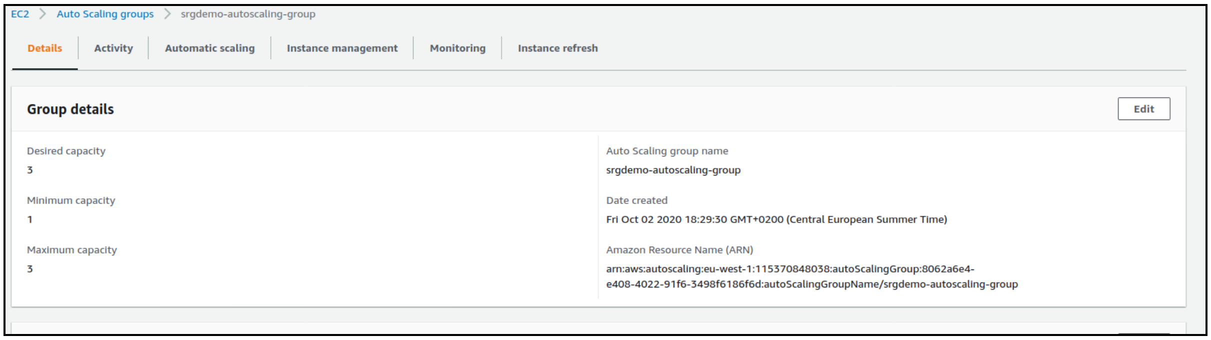
To scale down a cluster using the Web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Auto Scaling,** click **Auto Scaling Groups**.
3. From the list of **Auto Scaling groups**, click the Auto Scaling group that you previously created.


EC2 > Auto Scaling groups
Auto Scaling groups (1)
Edit
Delete
Create an Auto Scaling group
Search your Auto Scaling groups
1

□	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
□	srgdemo-autoscaling-group	srgdemo-launch-configuration	3	-	3	1	3	eu-west-1b, eu-west-1c, eu-west-1a

4. On the **Details** tab, under **Group Details**, click **Edit**.



Group details

Desired capacity
3

Minimum capacity
1

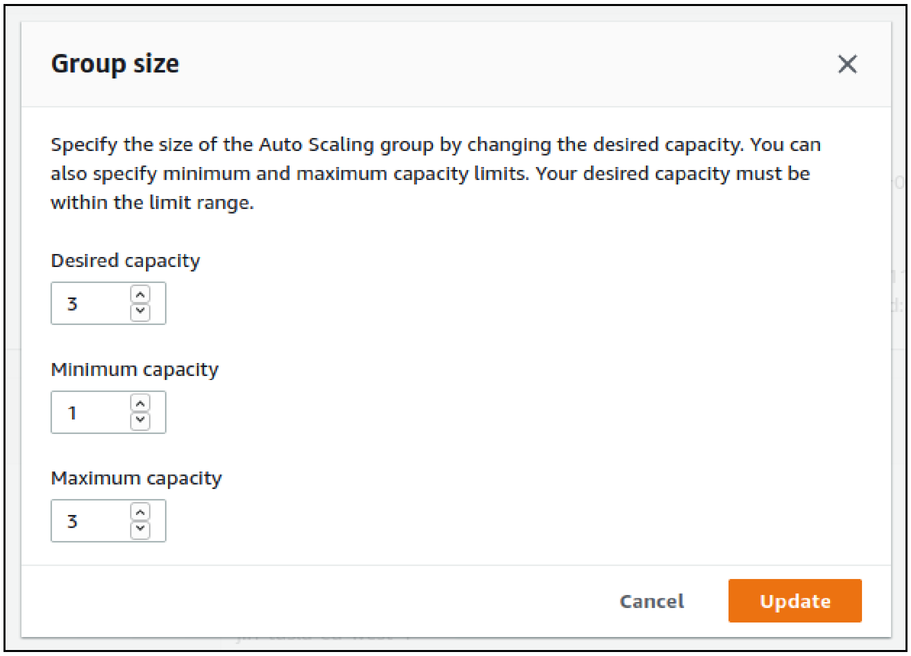
Maximum capacity
3

Auto Scaling group name
srgdemo-autoscaling-group

Date created
Fri Oct 02 2020 18:29:30 GMT+0200 (Central European Summer Time)

Amazon Resource Name (ARN)
arn:aws:autoscaling:eu-west-1:115370848038:autoScalingGroup:8062a6e4-e408-4022-91f6-3498f6186f6d:autoScalingGroupName/srgdemo-autoscaling-group

5. On the **Group Size** dialog, enter new values for the cluster's desired, minimum, and maximum capacities.



Group size

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity
3

Minimum capacity
1

Maximum capacity
3

6. Click **Update**.

7. You are returned to the Auto Scaling group details. The status is shown as *Updating capacity.*

To scale down the number of nodes in an AWS cluster using the CLI:

1. Verify the existing limits set on your Auto Scaling group. Run the following command:

```
# aws autoscaling describe-auto-scaling-groups \
--auto-scaling-group-names <your auto-scaling group name> \
| jq -r '.AutoScalingGroups[0] \
| "MaxSize: " + (.MaxSize | tostring),"DesiredSize: " + (.DesiredCapacity
| tostring),"MinSize: " + (.MinSize \
| tostring)'
```

Example input and output:

```
# aws autoscaling describe-auto-scaling-groups \
--auto-scaling-group-names srgdemo-autoscaling-group \
| jq -r '.AutoScalingGroups[0] \
| "MaxSize:         " + (.MaxSize \
| tostring),"DesiredCapacity: " + (.DesiredCapacity \
| tostring),"MinSize:         " + (.MinSize | tostring)'
```

```
MaxSize:         5
```

```
DesiredCapacity: 5
```

```
MinSize:         1
```

In this example, it would be possible to scale down to 3 nodes without any other configuration changes.

2. Decrease the number of nodes in the Auto Scaling group by running the command (command has no output):
   ```
   # aws autoscaling update-auto-scaling-group \
   --auto-scaling-group-name <your auto-scaling group name> \
   --desired-capacity <desired new number of nodes>
   ```
3. To verify the updated values, repeat the command shown in Step 1, above.

The AWS will then gracefully stop the removed nodes (2 in our example) and destroy their VMs.

Understanding the Transformation Hub Kafka Manager

The Transformation Hub Kafka Manager enables you to monitor and manage your clusters, topics, and partitions, including the following:

- Viewing and managing cluster states, including topics, consumers, offsets, broker nodes, replica distribution, and partition distribution.
- Creating and updating topics.
- Generating partitions and adding partitions to a topic.
- Reassigning partitions to other broker nodes, such as replacing a failed node with a new one.
- Reassigning partition leaders to their preferred broker node after a node temporarily leaves the cluster (for example, in case of a reboot).
- Managing JMX polling for broker-level and topic-level metrics.

Connecting to Kafka Manager for an On-Premises Cluster

Only users that can log into the Transformation Hub server can access the Transformation Hub Kafka Manager. These users can access the Transformation Hub Kafka Manager by using their local web browser directly from any of the Transformation Hub nodes or by using SSH forwarding from a local system.

You can connect to the Container Deployment Foundation Kafka Manager with the Chrome or Firefox browsers.

To access Kafka Manager:

1. On a Transformation Hub node, run the command to get the Kafka Manager service:

   ```
   kubectl get services --all-namespaces|grep th-kafkamgr-svc
   ```

2. Note `th-kafkamgr-svc` service and note its IP and port number.

To connect directly from a node:

1. Log into the Transformation Hub node.
2. In a terminal window, run the following command:

   ```
   kubectl -n <the arcsight-installer-* namespace> port-forward  <the th-kafka-manager-
   * pod name> 9000:9000
   ```

3. With a supported browser, connect to Transformation Hub Kafka Manager:

   ```
   http://localhost:9000
   ```

Once connected, the browser displays the **Clusters** page. For more information about this page, see Managing Clusters.

To connect from your local host:

1. From your local system, set up SSH forwarding and connect by using a command like the following:

   ```
   ssh -L <local port>:<Transformation Hub Kafka Manager Service IP>:<port>
   root@<TH master node address>
   ```

2. In a browser, connect by using the following URL:

   ```
   http://<localhost>:<local port>
   ```

Once connected, the browser displays the **Clusters** page. For more information about this page, see Managing Clusters.

Managing the Kafka Cluster

The **Clusters** page is the Transformation Hub Manager's home page. From here you can modify, disable or delete a cluster from view in the Transformation Hub Manager (the cluster itself is not deleted), or drill down into the cluster for more information.

Location: Clusters

Click the *Cluster Name* link. The Transformation Hub Manager displays the **Cluster Summary** page. For more information, see Viewing Information About a Cluster.

To edit the cluster:

1. Click **Modify**. The Transformation Hub Manager displays the **Update Cluster** page.
2. Update the appropriate fields, and click **Save**.

> Editing the cluster is an advanced operation, and normally the cluster should never be edited.

To disable the cluster:

Click **Disable**. Once a cluster has been disabled, a **Delete** button is displayed.

To delete the cluster:

After disabling the cluster, click **Delete**.

Viewing Information About a Cluster

On the **Summary** page, you can view the ZooKeeper processes in your cluster and drill down into its topics and broker nodes for more information.

Location: Clusters > *Cluster Name* > Summary

- "Viewing Information" below
- "Viewing or Editing the Topics" on the next page
- "Viewing or Editing the Broker Nodes" on the next page

Viewing Information

To view information about your cluster:

- If the cluster is not yet open, click **Cluster > List** in the navigation bar. Then click the *Cluster Name* link.
- If the cluster is already open, click **Clusters > *Cluster Name* > Summary**

Viewing or Editing the Topics

To view or edit the topics in your cluster:

Click the **Topics** hyperlink (number of topics) to show the topics in the cluster. For more information, see Managing Topics.

Viewing or Editing the Broker Nodes

To view or edit the broker nodes in your cluster:

Click the **Brokers** hyperlink (number of broker nodes) to show the broker nodes in the cluster. For more information, see Managing Brokers.

Stream Processor Groups

Transformation Hub implements two types of stream processors to process events.

- "Routing Stream Processors" below
- "Transforming Stream Processors" on the next page
- "Describing Routing" on the next page
- "Tuning Stream Processor Groups" on the next page
- "Best Practices for Routing Stream Processors" on page 516

Routing Stream Processors

Routing stream processors process event data and send it to destinations, based on Transformation Hub routing rules specified in ArcSight Management Center. There are two types of routing stream processors:

- CEF-to-CEF routing stream processing is supported in Transformation Hub 3.4.0 and all previous versions.
- New in Transformation Hub 3.4.0, Avro-to-Avro routing stream processing occurs between two `event-avro` topics. To use an Avro topic, it should be of the type `event-avro`. You can configure a topic with this type in two ways:
  - Create the topic with type `event-avro` using ArcMC 2.9.6 or later and Transformation Hub 3.4.
  - Change the type of an existing topic to `event-avro` using ArcMC 2.9.6 or later.

As a general guideline for routing stream processors, stream processor configurations and routes are refreshed every 60 seconds. Consider this factor when adding, deleting, or editing routing rules using ArcMC.

Transforming Stream Processors

As of ArcSight SmartConnector 8.1, the SmartConnector is capable of sending events to Transformation Hub in the Avro event format from which they can be consumed by Avro formatted event consumers, such as ESM and Database. Earlier versions of the SmartConnector were not capable of this and, as such, would send CEF formatted events to Transformation Hub that then needed to be transformed to Avro format in order to be consumed by Avro formatted event consumers. The following default CEF to Avro or C2AV transforming stream processors work to transform CEF data in the CEF source topic and route it to the dedicated Avro destination topic for use by Avro consumers.

1. The CEF-to-Avro stream processor transforms events from the th-cef topic to the th-arcsight-avro topic.
2. The CEF-to-Avro ESM Filtered Stream Processor transforms events from the mf-event-cef-esmfiltered topic to the mf-event-avro-esmfiltered topic. For more information about filtering events for ESM, see "Filtering Events for ESM" on page 524.

Describing Routing

Each stream processor includes six processing threads. All routes with the same source topic are processed by one *routing stream processor group*. You can scale a processor group independently as load increases by adding more routing processor instances to the group.

- The number of routing stream processor groups should match the number of source topics they are processing.
- Each routing stream processor group can contain multiple routing stream processors.
- You can configure up to 10 routing stream processor groups on Transformation Hub in the CDF Management Portal, allowing Transformation Hub to support up to 10 source topics.
- You configure routing in ArcMC.

Tuning Stream Processor Groups

The performance of stream processors is critical to Transformation Hub performance. In general, you can follow these guidelines for tuning stream processors and drive better performance.

- Since all routes which use the same source topic share the same routing stream processor group, adding more source topics can speed up processing.

- Increase the number of source topic partitions to handle high EPS throughput, depending on the CPU and memory resources of each worker node. For example, when the partition number is increased to 60, up to 10 routing (or C2AV) process instances can be used. Each stream processor uses 6 threads by default.
- Where possible, limit the number of routing rules per route.

Best Practices for Routing Stream Processors

The following best practices apply to management of routing stream processors.

- By default, Transformation Hub has 1 routing stream processor group. Accordingly, if you create 2 or more routes with different source topics, then make sure to enable more stream processor groups according to the number of source topics used in such routes (this applies to both type of routings: CEF-to-CEF or Avro-to-Avro).
- To enable and increase the number of instances of routing stream processor groups, in the CDF Management Portal, browse to the Reconfigure page. Identify the desired group number; and to enable it, just increase it from 0 to the desired value.
- To support high availability, routing stream processor groups can scale up and down partially. Once a group is enabled, you can increase or decrease the number of instances. However, it might never be reduced to 0, or the source topic mapped to that service group will no longer route until you increase the number of instances above 0.
- Always consider the available resources when enabling more routing stream processor groups.
- C2AV and routing stream processing in Transformation Hub are Kafka Streams applications. By default, Kafka Streams are using at-least-once processing guarantees in the presence of failure. This means that if the stream processing application fails, no data records are lost or will fail to be processed, but some data records maybe re-read and therefore reprocessed. Therefore, C2AV and routing stream processing is using an at-least-once processing guarantees configuration. In this case, when C2AV/Routing pods are killed abnormally and restarted, the user might see duplicated events.

Managing Brokers

On the **Brokers** page, you can see an overview of all of your Worker nodes and drill down into a node for more information.

> The term *Brokers* refers to nodes running Kafka services (that is, Kubernetes worker nodes, but not master nodes).

Location: Clusters > *Cluster Name* > Brokers

To view the broker nodes in your cluster:

Click **Brokers** in the navigation bar. The **Brokers** page opens.

To see more information about a specific broker:

Click the broker's *Id* link. The *Broker Name* ID opens. For more information, see "Managing Brokers" on the previous page

Viewing Broker Details

You can view detailed information about a broker from the *Broker Name* details page.

Location: Clusters > *Cluster Name* > Brokers > *Broker Name*

To view information on a specific broker:

1. Click **Brokers** in the navigation bar.
2. Click the *Broker Name* link. The *Topic Name* page opens.

The following data is displayed.

Summary

In the **Summary** section, you can see an overview of your broker, including the number of topics and partitions located on it.

Metrics

In the **Metrics** section, you can view information about the data flow.

Messages count

In the **Messages** section, you can view a message view chart.

Per Topic Detail

In the **Per Topic Detail** section, you can view topic replication and partition information and drill down to view more information about each topic.

To see more information about a specific topic:

Click the *Topic Name* link in the **Per Topic Details** section. See Viewing Topic Details

Managing Topics

On the **Topics** page, you can run or generate partition assignments, add a new partition, and drill down into individual topics for more information.

Location: Clusters > *Cluster Name* Topic > List

> **Note:** The following default topics are used internally by Transformation Hub and should not be deleted, modified, or used by external data producers or consumers.
>
> ```
> __consumer_offsets
> ```
>
> ```
>  _schemas
> ```
>
> ```
> th-arcsight-json-datastore
> ```
>
> ```
> th-arcsight-avro-sp_metrics
> ```
>
> ```
> th-syslog
> ```
>
> ```
> th-arcsight-avro
> ```
>
> ```
> mf-event-avro-esmfiltered
> ```
>
> ```
> mf-event-cef-esmfiltered
> ```
>
> ```
> th-cef
> ```

To manage the topics in your cluster:

Click **Topic > List** in the navigation bar.

To view information on a topic:

Click the *Topic Name* link. The *Topic Name* page displays the topic's summary, metrics, consumers, and partitions. See Viewing Topic Details.

To generate partition assignments:

1. Click **Generate Partition Assignments**.
2. Select the topics and broker nodes to reassign.
3. Click **Generate Partition Assignments**.

To assign partitions as generated:

1. Click **Run Partition Assignments**.
2. Select the topics to reassign.
3. Click **Run Partition Assignments**.

To add a partition:

1. From the Topics Summary page, click **Add Partition**.
2. Enter the new number of partitions.

3. Select the topics and broker nodes.
4. Click **Add Partitions**.

Default Topics

Transformation Hub manages the distribution of events to topics, to which consumers can subscribe and receive events from.

Transformation Hub includes the following default topics:

Topic Name	Event Type	Valid Destinations
mf-event-avro-esmfiltered	Filtered Avro events for consumption by ESM.	SmartConnector or Connector i Transformation Hub (CTH).
mf-event-cef-esmfiltered	Filtered CEF events for consumption by ESM.	SmartConnector or Connector i Transformation Hub (CTH).
th-arcsight-avro	For ArcSight product use only. Event data in Avro format.	Transformation Hub (CTH).
th-arcsight-avro-sp_metrics	For ArcSight product use only. Routing stream processor operational metrics data.	
th-arcsight-json-datastore	For ArcSight product use only. Event data in JSON format for use by ArcSight infrastructure management	
th-binary_esm	Binary security events, which is the formatconsumed by ArcSight ESM.	SmartConnector
th-cef	CEF event data.	SmartConnector or Connector i Transformation Hub (CTH).
th-cef-other	CEF event data destined for a non-ArcSight subscriber.	
th-syslog	The Connector in Transformation Hub (CTH) feature sends raw syslog data to this topic using a Collector.	Should only be configured as Collector or CTH destination.

In addition, using ArcSight Management Center, you can create new custom topics to which your SmartConnectors can connect and send events.

Topic Data Preservation

Topic data is preserved across Transformation Hub restarts, reinstalls, and upgrades.

- When a Transformation Hub reinstall is performed, all data in a Kafka topic is preserved. No data is lost.
- When the consumer resumes data collection from the topics, the consumer re-starts where it last left off.

No data is lost.

Creating Topics

> This method of creating topics does not permit you to specify topic type. As a result, it is strongly recommended that you use ArcMC to create new topics in Transformation Hub.

You can create a new topic on the **Create Topic** page.

Location: Clusters > *Cluster Name* Topics > Create Topics

To open the Add Topic page:

Click **Topic > Create** in the navigation bar.

To create a new topic:

1. Fill in values for the **Topic Name,** number of **Partitions**, and **Replication Factor** fields
2. Click **Create**.

For a discussion of field values, consult the Kafka documentation.

The number of custom topics you can create will be limited by Kafka, as well as performance and system resources needed to support the number of topics created.

Creating Routes for Topics

You can use ArcMC to view and create topics , as well as to create *routes*, which direct events into appropriate topics.

A *route* is a rule that directs Transformation Hub to duplicate events that meet certain criteria (filter) from a source topic to the route's destination topic. Rules are defined using event field names and expected values. Only CEF and Avro format events can be routed; binary security events in the `th-binary_esm` topic cannot be routed.

Using ArcMC, you can view, create, edit and delete routes based on CEF fields or Avro schema fields and event metadata. (You must create destination topics before you can route events to them.) Refer to the ArcMC Administrator’s Guide, available from the Micro Focus support community, for more information.

Viewing Topic Details

You can see details about a topic, including information about the summary, metrics, consumers, and partitions from the *Topic Name* details page.

Location: Clusters > *Cluster Name* Topics > *Topic Name*

To view information on a specific topic:

1. Click **Topic > List** in the navigation bar.
2. Click the *Topic Name* link. The *Topic Name* page opens.

The following data is displayed.

Topic Summary

In the **Topic Summary** section, you view information on the topic's replicas, partitions, and broker nodes.

Metrics

In the **Metrics** section, you can view information about the data flow.

Operations

In the **Operations** section, you can perform a variety of tasks on broker nodes.

To reassign partitions:

Click **Reassign Partitions**.

To update a topic's configuration:

1. Click **Update Config**.
2. Edit the configuration fields.
3. Click **Update Config**.

To specify partition assignments:

1. Click **Manual Partition Assignment**.
2. Select the desired assignments.
3. Click **Save Partition Assignment**.

Partitions by Broker

In the **Partitions by Broker** section, you can see topic partition information and drill down to see details for each broker.

To view details on a broker:

Click the **Broker** link. The **Topic Summary** page displays information on the topic's lag, partitions, and consumer offset.

In Transformation Hub Kafka Manager, users will see different offset values between CEF (Recon or Logger) topics and binary (ESM) topics. In CEF topics, the offset value can generally be associated with number of events that passed through the topic. Each message in a CEF topic is an individual event. However, that same association cannot be made for the ESM topic, as several events are batched into each message.

Consumers consuming from this topic

In the **Consumers consuming from this topic** section, you can drill down to see details on each consumer.

> New consumers can take some time to display properly. Give the process time to populate the view with the correct data.

To view details on a consumer:

Click the *Topic Name* link. The Topic Summary page displays information on the topic's lag, partitions, and consumer offset.

Partition Information

In the **Partition Information** section, you can view information about the topic's partitions and drill down for more information about each leader.

To view details on a leader:

Click the **Leader** link. The *Broker Name* ID page displays the broker's summary, metrics, message count, and topic details. See Viewing Broker Details.

Data Redundancy and Topic Replication

When configuring a Transformation Hub, you can specify the number of copies (replicas) of each topic which Transformation Hub should distribute.

Kafka brokers automatically distribute each event in a topic to the number of broker nodes indicated by the topic replication level specified during the Transformation Hub configuration. While replication does decrease throughput slightly, ArcSight recommends that you configure a replication factor of at least 2.

You need at least one node for each replica. For example, a topic replication level of 5 requires at least five nodes; one replica would be stored on each node. The following table illustrates how the replication factor provides redundancy in case of unavailable nodes.

Replication Factor	Number of brokers receiving the event	If one node becomes unavailable...
1	1	Data is lost
2 (or more)	Same as replication factor	• Copies of the event data are still present on other node. • Data is restored to an unavailable node when it becomes available again. • No data is lost unless all nodes become unavailable simultaneously.

When you add new consumers, you don't need to update your producers. The distribution and replication is handled for you. Refer to the Kafka documentation for more information.

Filtering Events for ESM

Transformation Hub is capable of filtering and routing from a source topic of type event-avro to a destination topic of type event-avro. This capability can be used to filter events from a source topic such as th-arcsight-avro to a destination topic which ESM can consume from, such as mf-event-avro-esmfiltered. Both of these are default topics described here.

1. Use ArcSight Smart Connectors or any producer that supports sending Avro formatted events to send the events directly to an event-avro topic. Smart Connectors by default will send Avro formatted events to the th-arcsight-avro topic.
2. Filter the events using Transformation Hub's Avro routing rules using ArcMC 2.96 or later. Create a routing rule with an event-avro topic as source topic (such as th-arcsight-avro) and an event-avro topic as destination topic (such as mf-event-avro-esmfiltered). For more information, please refer to the routing section in the ArcMC Administration Guide.

   > Earlier versions of Transformation Hub that did not yet support Avro routing rules required using a combination of CEF routing rules and CEF-to-Avro conversion. Using Avro routing rules is a more efficient way to filter Events for ESM, however, so it is now the recommended approach.

Managing Consumers

On the **Consumers** page, you can see a list of consumers, view their type, the topics they consume, and drill down into each consumer and topic for more information.

Location: Clusters > *Cluster Name* > Consumers

To view or edit the consumers in your cluster:

Click **Consumers** in the navigation bar.

To view more details on a specific consumer:

Click the *Consumer Name* link. The *Consumer Name* page displays details about the consumer. You can drill down further for more information, including Consumed Topic Information (such as Partitions Covered % and Total Lag).

To view more details on the topic it consumes:

Click the *Topic Name* link. The *Topic Name* page displays details about the topic. You can drill down further for more information including Consumer Lag, and Consumer Offset and LogSize data by Partition.

Producing Events with SmartConnectors

SmartConnectors can publish events to Transformation Hub topics. In order to publish events, you must configure your SmartConnectors to use the Transformation Hub destination. To send events to multiple topics, you can configure multiple concurrent destinations with the same Transformation Hub using different topics.

Once configured with a Transformation Hub destination, the SmartConnector sends events to Transformation Hub's Kafka cluster, which can then further distribute events to real-time analysis and data warehousing systems. Other applications, including Recon, ESM, Logger, and any third-party application that supports retrieving data from Kafka can receive them, for example, Apache Hadoop.

Transformation Hub balances incoming events between nodes, by distributing them evenly between the partitions in the configured topic.

Acknowledgments ("acks") ensure that Transformation Hub has received the event before the SmartConnector removes it from its local queue. You can disable acknowledgments, require acknowledgment only from the primary replica, or require every replica to acknowledge the event. (Acknowledgments do not indicate that consumers, such as Logger, have received the event data, only that Transformation Hub itself has.)

> Performance impact due to leader acks is a known Kafka behavior. Exact details of the impact will depend on your specific configuration, but could reduce the event rate by half or more.

Supported SmartConnector versions encode their own IP address as meta-data in the Kafka message for consumers that require that information such as Logger Device Groups.

- For information on supported SmartConnector versions, see the SODP Support Matrix.
- For more information about SmartConnectors and how to configure a Transformation Hub destination, refer to the SmartConnector User's Guide.

Micro Focus documentation is available for download from the Micro Focus support community.

Viewing Consumer Details

You can see a information about a consumer and drill down on the topics it consumes from the *Consumer Name* details page.

Location: Clusters > *Cluster Name* Consumer > *Consumer Name*

To view information on a consumer:

1. Click Clusters > *Cluster Name* Consumer.
2. Click the *Consumer Name*.

To view information on the consumed topic:

1. Click the *Topic Name*. The Consumed Topic Information page displays information about the topic. Click the topic name for more information including Consumer Lag and Consumer Offset and LogSize data by partition .

Consuming Events with ESM

ESM agents are the consumers for Transformation Hub's publish-subscribe messaging system. An ESM agent can connect to Transformation Hub and consume all events in binary or Avro format for the topics to which it is subscribed.

Additionally, ESM provides data monitors to monitor Transformation Hub health.

- For information on supported versions of ESM and SmartConnectors, see the *SODP Support Matrix.*

- For instructions on configuring a supported version of ESM as a consumer, see the *ESM Administrator's Guide*.

Consuming Events with Logger

To subscribe to Transformation Hub topics with Logger, you must configure a receiver on a supported Logger version to receive the Transformation Hub events. Logger's Transformation Hub receivers are consumers for Transformation Hub's publish-subscribe messaging system. They receive events in Common Event Format (CEF) from Transformation Hub topics. A Logger Transformation Hub receiver connects to Transformation Hub and consumes all events for the topics it subscribes to.

When configuring a Logger Transformation Hub receiver, specify the worker node FQDNs, topics to consume from, and consumer group name. You can configure multiple Loggers to consume from the same topic as a part of a consumer group.

For more information about Logger and how to configure a Transformation Hub receiver, refer to the *Logger Administrator's Guide,* available for download from the Micro Focus support community.

> Kafka consumers can take up to 24 hours for the broker nodes to balance the partitions among the consumers. Check the Transformation Hub Kafka Manager **Consumers** page to confirm all consumers are consuming from the topic.

Sending Transformation Hub Data to Logger

For a Logger to be able to consume Transformation Hub events, the Logger must have a Transformation Hub receiver configured with the Transformation Hub worker nodes, consumer group, and event topic list. SmartConnectors that send data to Transformation Hub must have a Transformation Hub destination.

A group of Loggers, called a pool, can be configured to receive and distribute events between themselves. This works similarly to the Logger pool created by using the Logger Smart Message Pool destination on SmartConnectors. The difference is that when the SmartConnectors have a Logger Smart Message Pool destination, the event load is balanced by each SmartConnector, but when the SmartConnectors have a Transformation Hub destination, the event load is balanced by the Loggers.

Additional Loggers can be added to the pool simply by configuring the same Transformation Hub worker nodes, consumer group, and event topic list in the new Logger's Transformation Hub receivers, without having to reconfigure either the existing Loggers or any SmartConnectors.

The events retrieved by the Logger pool are distributed among the Loggers in the pool. If one Logger is down, new events are rebalanced among existing Loggers. When a Logger is added or removed from the Consumer Group, the event load is distributed across the pool of Loggers.

To send events from a group of SmartConnectors to a pool of Loggers, configure their Transformation Hub destinations to send events to the topic from which the Logger pool is consuming.

To configure Logger to subscribe to event data from specific SmartConnectors, you can do either of the following:

- Configure all the SmartConnectors to publish events to the same topic. Configure the Logger's Transformation Hub receiver to subscribe to this event topic.
- Configure each SmartConnector to publish events to different topics and then configure the Transformation Hub receiver on the Logger to subscribe to multiple event topics.

> ✓ **Tip:** Loggers in the same Logger pool do not consume the same events, since they are in the same Consumer Group. In high availability situations, you need events to be stored on two different Loggers. To store the same events on two Loggers, configure the Loggers to have different Consumer Group names, but subscribe them to the same event topic.

The number of Loggers in a Logger pool is restricted by the number of event topic partitions. For example, if there are only five partitions configured, only five Loggers will receive the events. If you have more than five Loggers configured in the same Consumer Group, some Loggers will not normally receive events, but will be available as hot spares. When adding receivers, be sure to increase the number of event topic partitions. See Managing Topics for more information.

Sending Transformation Hub data to Logger (Overview):

1. Configure the SmartConnector:
   - Set up a SmartConnector to publish to a particular Transformation Hub topic. Connectors can only send to a single topic for each destination. Additional destinations need to be configured if each event needs to go to multiple topics. Note the number of partitions in the topic.
   - For more information about SmartConnectors and how to configure a Transformation Hub destination, refer to the CEF Destinations chapter of the SmartConnector User's Guide, available for download from the Micro Focus support community.
2. Configure Logger:
   - Create a Transformation Hub receiver on each Logger in the Logger pool.
   - Configure each receiver to subscribe to the topics to which the SmartConnectors are publishing data. To subscribe to multiple topics, indicate the topics by specifying them in

the Event Topic List parameter (a list of comma-separated values) while configuring the Transformation Hub receiver.

- Configure each receiver to be in the same Consumer Group.

Example Setup with Multiple Loggers in a Pool

You can set up your Logger pools to subscribe to events from a particular device type, such as "Firewall." To do this, you would:

1. In ArcMC, create a Kafka topic named *Firewall*.
2. Configure all the SmartConnectors that handle firewall events to publish these events to topic "Firewall."
3. Configure the Loggers in the Logger pool:
   - Create a Transformation Hub Receiver on each Logger in the pool.
   - Configure the receivers to subscribe to the event topic “Firewall,” and include them in the “Logger_Firewall” Consumer Group.

After the configuration is set up properly, the Logger pool will subscribe to device type *Firewall*.

> This example assumes that the Transformation Hub is being managed by an ArcSight Management Center for topic creation. Topics can also be managed through the Kafka Manager UI.

Consuming Events with Third-Party Applications

Transformation Hub is designed with support for third-party tools. You can create a standard Kafka consumer and configure it to subscribe to Transformation Hub topics. By doing this you can pull Transformation Hub events into your own data lake.

> Custom consumers must use Kafka client libraries of version 0.11 or later.

- All Transformation Hub nodes, consumers, and producers must be properly configured for forward and reverse DNS lookup, and be time-synchronized, using a time server such as NTP.
- Events are sent in standard CEF (CEF text) and binary (exclusively for ESM consumption). Any software application that can consume from Kafka and understand CEF text can process events.
- You can set up multiple consumer groups, and each group will get a copy of every event. Therefore you can have Logger and Apache Hadoop configured to consume from the same topic and each will get a copy of every event. This enables fanning out multiple copies of

events without reconfiguring SmartConnectors or using additional CPU or network resources for them.

Consuming Events with Apache Hadoop

Apache Hadoop is a software framework that enables the distributed processing of large data sets across clusters of computers. You can send events to Hadoop by using Apache Flume.

This section describes how to set up the Apache Flume agent to transfer Common Event Format (CEF) events from an Kafka cluster to Hadoop Distributed File System (HDFS).

It includes the following topics:

Architecture for Kafka to Hadoop Data Transfer

Apache Flume uses a source module to read a Kafka topic containing CEF events, and it then transfers the events using a memory channel, and persists them to HDFS using a sink module. The CEF files are stored on HDFS by time, in a year/month/day/hour directory structure.

Using Apache Flume to Transfer Events to Hadoop

One of the applications you could use to transfer Transformation Hub events into your data lake is Apache Flume. Flume is designed to push data from many sources to the various storage systems in the Hadoop ecosystem, such as HDFS and HBase. This section describes how to use Apache Flume as a data transfer channel to transfer events from Transformation Hub to Apache Hadoop or other storage systems.

Prerequisites

- Transformation Hub installed: Consult the *Micro Focus Transformation Hub Deployment Guide*.
- Flume installed: For information on how to install and configure Flume, refer to the Flume documentation.
- Storage system installed: Refer to your storage system documentation.

Procedure

Flume is controlled by an agent configuration file. You must configure Transformation Hub as the source agent, your storage system as the sink agent, and ZooKeeper as the channel agent in this file.

To configure Transformation Hub as the source:

Edit the agent configuration file to include the required properties, as in the table below. Configure other properties as needed for your environment.

Required Kafka Source Configuration

Property	Description
`type`	Set to `org.apache.flume.source.kafka.KafkaSource`.
`topic`	The Event Topic from which this source reads messages. Flume supports only one topic per source.

To configure the sink:

The required configuration varies. Refer to the Flume documentation for details on your storage system. The section Consuming Events with Apache Flume provides an example of how to configure Apache Hadoop as the sink.

Setting Up Flume to Connect with Hadoop

In the simplest deployment model, you need to deploy the Apache Flume agent on a Hadoop node server to pull events, and send them to Hadoop Distributed File System (HDFS).

Hadoop must be installed before you can connect it with Flume. If you do not already have your own Hadoop deployment, you can deploy Hadoop on a Red Hat Enterprise Linux 7.2 host. For more information, see Setting Up Hadoop.

For a detailed discussion of connecting Apache Flume with Hadoop, consult the Apache online documentation.

Sample Flume Configuration File

Before starting Apache Flume, create a configuration file based on the template below.

The configuration file should reside in `bin/flume/conf/`. This file is called *kafka.conf* in our example. You can name your own configuration file whatever is appropriate.

```
####################################################

#Sample Flume/Kafka configuration file

####################################################

#defines Kafka Source, Channel, and Destination aliases

tier1.sources = source1

tier1.channels = channel1

tier1.sinks = sink1
```

```
#Kafka source configuration
tier1.sources.source1.type = org.apache.flume.source.kafka.KafkaSource
tier1.sources.source1.kafka.bootstrap.servers= kafkaIP1:9092, kafkaIP2:9092,…
tier1.sources.source1.kafka.topics = th-cef
tier1.sources.source1.kafka.consumer.group.id = flume
tier1.sources.source1.channels = channel1
tier1.sources.source1.interceptors = i1
tier1.sources.source1.interceptors.i1.type = timestamp
tier1.sources.source1.kafka.consumer.timeout.ms = 150
tier1.sources.source1.kafka.consumer.batchsize = 100
#Kafka Channel configuration
tier1.channels.channel1.type = memory
tier1.channels.channel1.capacity = 10000
tier1.channels.channel1.transactionCapacity = 1000
#Kafka Sink (destination) configuration
tier1.sinks.sink1.type = hdfs
tier1.sinks.sink1.channel = channel1
tier1.sinks.sink1.hdfs.path = hdfs://localhost:9000/opt/\
hadoop/cefEvents/year=%y/month=%m/day=%d
tier1.sinks.sink1.hdfs.rollInterval = 360
tier1.sinks.sink1.hdfs.rollSize = 0
tier1.sinks.sink1.hdfs.rollCount = 0
tier1.sinks.sink1.hdfs.fileType = DataStream
tier1.sinks.sink1.hdfs.filePrefix = cefEvents
tier1.sinks.sink1.hdfs.fileSuffix = .cef
tier1.sinks.sink1.hdfs.batchSize = 100
tier1.sinks.sink1.hdfs.timeZone = UTC
```

Setting Up Hadoop

This is an overview of the steps necessary to install Apache Hadoop 2.7.2 and set up a one-node cluster. For more information, refer to the Hadoop documentation for your version.

To install Hadoop:

1. Be sure that your environment meets the operating system and Java prerequisites for Hadoop.
2. Add a user named 'hadoop'.
3. Download and unpack Hadoop.
4. Configure Hadoop for pseudo-distributed operation.
   - Set the environment variables.
   - Set up passphraseless SSH.
   - Optionally, set up Yarn. (You will not need Yarn if you want to use Hadoop only storage and not for processing.)
   - Edit the Hadoop configuration files to set up a core location, a Hadoop Distributed File System (HDFS) location, a replication value, a NameNode and a DataNode.
   - Format the Name node.
5. Start the Hadoop server using the tools provided.
6. Access Hadoop Services in a browser and login as the user "hadoop".
7. Execute the following commands to create the Hadoop cefEvents directory:
   ```
   hadoop fs -mkdir /opt
   hadoop fs -mkdir /opt/hadoop
   hadoop fs -mkdir /opt/hadoop/cefEvents
   ```
8. Execute the following commands to grant permissions for Apache Flume to write to this HDFS
   ```
   hadoop fs -chmod 777 -R /opt/hadoop
   hadoop fs -ls
   ```
9. Execute the following command to check Hadoop system status:
   ```
   hadoop dfsadmin -report
   ```
10. Execute the following command to view the files transferred by Flume to Hadoop.
    ```
    hadoop fs -ls -R /
    ```

Connectors in Transformation Hub (CTH)

To reduce the computational overhead and workload on a syslog SmartConnector infrastructure, you can make use of Connectors in Transformation Hub (CTH) instead.

CTH Functionality

Operationally, Micro Focus SmartConnectors hold two main responsibilities:

- **Collection**: A SmartConnector collects data from various sources.
- **Processing**: A SmartConnector processes the collected data into enriched security event data and posts them to a destination.

With CTH, the two functions of SmartConnector are handled in a slightly different manner. CTH takes advantage of the massive scalability of the robust Transformation Hub streaming architecture by moving the computationally intensive processing step directly to Transformation Hub.

- **Collection:** The collection step is performed by a dedicated Collector component, which gathers raw syslog data and publishes it to a dedicated syslog topic in Transformation Hub. As the name suggests, a Collector is a lightweight component responsible solely for collecting syslog data and passing it along to a dedicated CTH topic. A Collector is deployed on a VM or server using ArcMC.
- **Processing:** The CTH component reads the data from the Collector destination, and then parses, normalizes, enriches, and filters this data. It posts the data to a dedicated Transformation Hub topic for availability to any desired consumer. CTHs are deployed as Kubernetes pods within the CDF infrastructure.

CTH includes the majority of the functionality of ArcSight syslog connectors, except for data collection, which is handled by the lightweight Collector component instead. For more information about CTH configuration, consult the ArcSight Syslog Connector User Guide.

> In Transformation Hub 3.3.0 and later, CTHs support FIPS mode.

Advantages of CTH

CTH has the following advantages over traditional SmartConnector architecture.

- Hardware consolidation in the data collection layer where Collectors are deployed, due to the logical separation of collection and processing. A single data feed from a Collector can replace multiple SmartConnector feeds.
- Improved stability, easy horizontal scalability, and improved load balancing as data flows increase with time or fluctuate during operations.

- Ease of deployment, since CTHs are deployed with a single click in the ArcMC management console.
- Raw syslog data is now available in the CTH topic and can be shared with any desired consumer.

Limitations of CTH

- CTH presently supports the processing of syslog data only.
- Upgrades to CTH are performed by upgrading Transformation Hub, rather than by upgrading CTH itself.

Deploying and Managing CTH

Installation and management of CTH is performed on a managed Transformation Hub though the ArcMC management console. Consult the ArcMC Administrator’s Guide for instructions on how to deploy and manage CTH.

Destination Topics

Collectors should only be configured with the `th-syslog` topic as a destination (and no other destinations).

Valid routing topic destinations for CTH include the following:

- `th-cef`
- `th-binary_esm`
- `th-cef-other`

In addition, custom CTH source and destination topics might be configured on Transformation Hub. (Custom topics might only be created for CEF data.)

Collector/CTH Supported Security Modes

Collector destinations can support the following security modes:

- Plain text (no security mode selected)
- FIPS only
- TLS only

Collector security mode can be set during Instant Deployment in the ArcMC console. See the ArcMC Administrator's Guide for more information.

CTH source and destinations can support the following security modes:

- TLS + Client Authentication (default setting)
- FIPS + Client Authentication (automatically set when enabling FIPS mode in Transformation Hub.
- Plain text (no security mode selected)
- TLS only
- FIPS only

If desired, CTH's plain text, TLS-only, and FIPS-only modes can be set in ArcMC after deployment.

Configuring Consumers and Producers for Availability

Configure the Transformation Hub Kafka cluster endpoint to avoid single points of failure in both the producers sending data to Transformation Hub (such as SmartConnectors), and the consumers subscribing to data from the Transformation Hub (such as Logger and ESM).

For Producers

Configure the **Initial Host:Port(s)** parameter field in the Transformation Hub Destination to include all Kafka broker (worker) nodes as a comma-separated list.

Provide all Kafka broker (worker) nodes for a producer and a consumer configuration to avoid a single point of failure. For example, broker_hostname1:9093, broker_hostname2:9093, broker_hostname3:9093.

For more information about how Kafka handles this using bootstrap.servers, see here.

For Consumers

Configure the **Transformation Hub host(s) and port** parameter field in the Receiver to include all Kafka cluster nodes as a comma-separated list.

For more information about how Kafka handles this using bootstrap servers, see here.

Managing Preferred Replicas

You can update the replicas for each cluster on the **Preferred Replica Election** page.

Location: Clusters > *Cluster Name* > Preferred Replica Election

To open the Preferred Replica Election page:

Click **Preferred Replica Election** in the navigation bar.

To run the Preferred Replica Election for your topic:

Click **Run Preferred Replica Election**.

Managing Partitions

You can reassign partitions for your cluster on the **Reassign Partitions** page.

Location: Clusters > *Cluster Name* > Reassign Partitions

- "Opening Reassigned Partitions" below
- "Reassigning the Partitions" below
- "Configuring Topic Partitions Based on Number of Consumers" below

Opening Reassigned Partitions

To open the Reassign Partitions page, click **Reassign Partitions** in the navigation bar.

Reassigning the Partitions

To reassign the partitions for your topic, click **Reassign Partitions**.

Configuring Topic Partitions Based on Number of Consumers

You can scale the consumption rate for a consumer of a topic by adding more consumers to the consumer group. However, when adding new consumers to the consumer group, please consider the topic partition count of the topic you are consuming from. The following table shows the relationship between the number of consumers in a consumer group and data consumption from each partition.

Number of Consumers in Group is...	Consumption from Partitions
A single consumer	Consumer consumes from all partitions in the source topic.
Lower than partition count	Each consumer consumes from a subset of the topic partitions.
Equals partition count	Each consumer consumes from each of the topic partitions.
Exceeds partition count	Each consumer consumes from each of the topic partitions; additional consumers stay idle until new partitions are added to the source topic.

If you change the number of partitions in the source topic to match the consumer group size (same or a multiple) for a given consumer group consumption rate, or add additional consumers in the consumer to match the topic partition count, then the Transformation Hub will automatically re-balance the consumer groups.

Overriding Application Properties

Each Transformation Hub module (Kafka, Zookeeper, and so on) has many additional properties available, and there may be a need for system administrators to override the default values for some of these properties. This section covers how to override these property values.

Property values (for properties that support overrides) are set by injecting environment variables in the respective container's start-up environment. These variables are read from a user-supplied properties file, in a specific location on the Network File Server (NFS). To see the available properties for override, consult the respective module's published documentation.

In most cases, this feature is not required for normal operation of Transformation Hub, and most likely will be used at the direction of technical support. Not all properties support overrides; please check with technical support before making any changes to your configuration.

- For Kafka, ZooKeeper, and Schema Registry properties, consult the appropriate Confluent documentation.
- The properties for routing processor and stream processor modules are detailed below.

> **Note**: Legacy properties prefixed with `arcsight.eventbroker` will continue to function as they did in previous versions, but as explained below, newly added properties must be prefixed with `arcsight.th`. If two properties of the same name are set with different prefixes, the property with `arcsight.th` will supersede the other one.

- Configuring the Values
- Routing Processor and Stream Processor Properties
- Changing Value Examples
- Creating the NFS Shares
- Exporting the NFS Configuration

Configuring the Values

1. Create a file named `arcsight-env-override.properties` under the folder `<NFS_root_DIRECTORY>/transformationhub/config`.

> The `<NFS_root_DIRECTORY>` path is described in this guide as the external NFS root folder (usually `/opt/arcsight/nfs/volumes`). For more information, refer to the section NFS Directory Structure.

2. Add properties to the file. To each property, add the module prefix from the table below.

Module	Prefix
Kafka	`arcsight.th.kafka.`
Schema Registry	`arcsight.th.schema-registry.`
ZooKeeper	`arcsight.th.zookeeper.`
Routing Processor/Stream Processor	`arcsight.th.sp.`

3. Delete the pods for which properties were defined, or, alternatively, redeploy Transformation Hub.

4. To verify the changes, search the log file (after the container's status is back to Running) for matching properties.

Routing Processor and Stream Processor Properties

As explained above, prefix these properties with `arcsight.th.sp.` to create an override.

Property Name	Default Value	Description
`RETRIES`	2147483647	The number of retries for broker requests that return a retry-able error.
`RETRY_BACKOFF_MS`	100	The amount of time (milliseconds), before a request is retried. This applies if the retries parameter is configured to be greater than 0.
`RECEIVE_BUFFER_BYTES`	65536	The size of the TCP receive buffer to use when reading data. If the value is -1, the OS default will be used.
`MAX_PARTITION_FETCH_BYTES`	1048576	The maximum amount of data per-partition the server will return. Records are fetched in batches by the consumer.
`MAX_REQUEST_SIZE`	1048576	The maximum size of a request in bytes.
`BUFFER_MEMORY`	33554432	The total bytes of memory the producer can use to buffer records waiting to be sent to the server.
`BATCH_SIZE`	16384	the default batch size in bytes when batching multiple records sent to a partition
`LINGER_MS`	100	the producer will wait for up to the given delay to allow other records to be sent so that the sends can be batched together

Property Name	Default Value	Description
HEARTBEAT_INTERVAL_MS	1000	The expected time (milliseconds) between heartbeats to the consumer coordinator when using Kafka's group management facilities. Heartbeats are used to ensure that the consumer's session stays active and to facilitate rebalancing when new consumers join or leave the group.
MAX_POLL_INTERVAL_MS	3600000	The maximum delay (milliseconds) between invocations of poll() when using consumer group management
MAX_POLL_RECORDS	100	The maximum number of records returned in a single call to poll().
SESSION_TIMEOUT_MS	180000	The timeout (milliseconds) used to detect client failures when using Kafka's group management facility
REQUEST_TIMEOUT_MS	305000	The configuration controls the maximum amount of time (milliseconds) the client will wait for the response of a request.
CONNECTIONS_MAX_IDLE_MS	540000	The maximum amount of time (milliseconds) before idle connections are closed.
TH_NUM_THREADS	6	The number of threads to execute stream processing.

Changing Value Examples

To change the value of `ZOOKEEPER_MAX_CLIENT_CNXNS` to 65, in ZooKeeper, and to change the value of `SCHEMA_REGISTRY_KAFKASTORE_TIMEOUT_MS` in the Schema Registry, create a file, `<NFS Volume mount>/transformationhub/config/arcsight-env-override.properties`, and add the following lines:

```
arcsight.th.zookeeper.ZOOKEEPER_MAX_CLIENT_CNXNS=65
```

```
arcsight.th.schema-registry.SCHEMA_REGISTRY_KAFKASTORE_TIMEOUT_MS=20000
```

Example of verifying the change by searching the log:

```
# kubectl -n transformationhub1 logs th-zookeeper-0 | grep ZOOKEEPER_MAX_
CLIENT_CNXNS
Environment override script set: ZOOKEEPER_MAX_CLIENT_CNXNS=65
ZOOKEEPER_MAX_CLIENT_CNXNS=65
```

Creating the NFS Shares

To create the NFS directory structure:

1. Log in to your NFS server and create the following:
   - A GROUP named `arcsight`, with a GID of 1999
   - A USER named `arcsight` with a UID of 1999
   - An NFS `root` directory at `/opt/arcsight/nfs/volumes`

> If you have previously installed any version of CDF, you must remove all NFS shared directories from the NFS server before you proceed. To do this, run the following command for each directory: `rm -rf <path to shared directory>`

2. For each directory listed in the table below, run the following command to create each NFS shared directory:

```
# mkdir -p <path to shared directory>
```

For example: `mkdir -p /opt/arcsight/nfs/volumes/itom_vol`

Directory	Mount Point Example
`<NFS_root_DIRECTORY>/itom-vol`	`/opt/arcsight/nfs/volumes/itom-vol`
`<NFS_root_DIRECTORY>/db-single-vol`	`/opt/arcsight/nfs/volumes/db-single-vol`
`<NFS_root_DIRECTORY>/db-backup-vol`	`/opt/arcsight/nfs/volumes/db-backup-vol`
`<NFS_root_DIRECTORY>/itom-Logging-vol`	`/opt/arcsight/nfs/volumes/itom-Logging-vol`
`<NFS_root_DIRECTORY>/arcsight-volume`	`/opt/arcsight/nfs/volumes/arcsight-volume`

3. Set the ownership in this structure to UID 1999 and GID 1999. Change the directory to `/opt`, and then run the following command:

```
# chown -R 1999:1999 <NFS_root_DIRECTORY>/arcsight
```

> If you use a UID/GID different than 1999/1999, then provide it during the CDF installation in the install script arguments `--system-group-id` and `--system-user-id`. In addition, if you are using NetApp with NFSv4 configuration, consider applying stickybits to all <NFS_root_directory> shares with:
> `# chmod g+s`
> `#chmod w+s`

Exporting the NFS Configuration

For every NFS volume, run the following set of commands on the External NFS server based on the IP address. You will need to export the NFS configuration with appropriate IPs in order for the NFS mount to work properly. For every node in the cluster, you must update the configuration to grant the node access to the NFS volume shares. On the NFS server, edit the `etc/exports` file and add all the shared volumes to the file.

Here is a sample `etc/exports` file entry for IP address 192.168.1.0, for all of the volumes:

```
/opt/arcsight/nfs/volumes/arcsight 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)
```

```
/opt/arcsight/nfs/volumes/itom_vol 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)
```

```
/opt/arcsight/nfs/volumes/db 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)
```

```
/opt/arcsight/nfs/volumes/logging 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)
   /opt/arcsight/nfs/volumes/db_backup 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)
```

Save the `/etc/exports` file, and then run the following command:

```
    # exportfs -ra
```

Synchronize the time on the NFS server and the time on the other servers in the cluster, using the same time source used by the other nodes in the cluster.

If you add more NFS shared directories later, you must restart the NFS service.

Pushing JKS files from ArcMC

You can push JKS (Java Keystore) files to multiple managed SmartConnectors in ArcMC. First, you will upload the files to a file repository in ArcMC, and then push them out to their destination SmartConnectors. You must then configure and enable the Kafka destination on all SmartConnectors.

To upload the Java Keystore files:

1. Prepare the .jks files you want to push and store them in a secure network location.
2. In ArcMC, click **Administration > Repositories > New Repository.**
3. In **Name, Display Name**, and **Item Display Name**, enter KAFKA_JKS

4. Enter other required details as needed, and then click **Save**.
5. Click **Upload to Repository.**
6. Follow the prompts in the upload wizard and browse to the first .jks file. Make sure to choose the individual file option.
7. Upload as many files as needed by repeating the upload wizard.

To push the files to multiple SmartConnectors:

1. In ArcMC, browse to the file repository for the `.jks` files.
2. Click the **Upload** arrow.
3. Follow the prompts in the wizard and select your destination SmartConnectors.
4. The files are pushed to the managed SmartConnectors and stored in the designated SmartConnector folder.

To configure the Kafka destination on all SmartConnectors:

In ArcMC, click **Node Management > Connectors** tab.

1. Select the SmartConnectors to be configured.
2. Choose **Add a destination** and pick the Kafka destination type.
3. Add the destination details along with the `.jks` path and password, and save the changes.

Transformation Hub Liveness Probes

A *liveness probe* is a Kubernetes feature that can be configured to detect problematic pods. Once detected, Kubernetes will take action to restart a problematic pod. Liveness probes help ensure higher availability of pods as well as a more robust cluster environment. Consult the Kubernetes documentation for a more detailed explanation of liveness probes. Transformation Hub supports these liveness probe types:

- TCP/IP port-socket connection
- HTTP request
- Log scanning

Each container or pod supports the listed liveness probes, with their default parameter values shown.

Container/Pod	Probe	initialDelaySeconds	periodSeconds	timeoutSeconds	failureThreshold
Kafka	tcp socket :9092 and log scanning	240	60	30	3
Zookeeper	tcp socket :2181 and log scanning	240	60	30	3

Container/Pod	Probe	initialDelaySeconds	periodSeconds	timeoutSeconds	failureThreshold
Web Service	https GET :8080 and log scanning	240	300	30	3
Schema Registry	https GET :8081 config and log scanning	240	300	30	3
Kafka Manager	http GET :9000 and log scanning	240	600	30	3
Routing Processor	log scanning	240	60	30	3
C2AV (CEF-to-Avro) Processor	log scanning	240	60	30	3

Probe parameters are defined as follows:

Parameter	Definition
`initialDelaySeconds`	Number of seconds after the container has started before liveness probes are initiated. The first probe execution after startup is not until initialDelaySeconds + periodSeconds.
`periodSeconds`	How often to perform the probe.
`timeoutSeconds`	Number of seconds after which the probe times out.
`failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up and restarting the pod.

Managing Liveness Probes

To check if a pod has a liveness probe configured:

1. Run:
   ```
   kubectl -n <namespace> describe pod <podname>
   ```
2. Review the output. Look (or `grep`) for the line starting with the string `Liveness...`This will show some of the probe's configuration.

To check for probe failures:

1. Run:
   ```
   kubectl get pods --all-namespaces
   ```
2. If any pod shows 1 or more restarts, run:
   ```
   kubectl -n <namespace> describe pod <podname>
   ```
3. Review any list of events at the end of the output. Liveness probe failures will be shown here.

Configuring Liveness Probes

The default values for liveness probes can be overriden by changing the values of the appropriate properties on the Configuration page.

1. Log in to the CDF Management Portal.
2. Click **Administration**.
3. Click the **...** (Browse) icon to the right of the main window.
4. From the drop-down, click **Reconfigure**. The post-deployment settings page is displayed.
5. Browse the configuration properties list to find the desired property, and enter the new value.
5. Click **Save**.

Configuring Log Scanning Liveness Probes

Log scanning probes scan the application's output for a match to a configured pattern, such as a known error message. If the pattern is found, the pod is restarted.

In addition to the four parameters described in the table above, log scanning probes have two additional properties:

literal	A literal expression for matching against the application's log output.
regex	A regular expression for matching against the application's log output.

- The *literal* property specifies a literal (exact match) search string. If the value matches a portion of the log text, the liveness probe, on its next periodic check, will report a failure and restart the pod.
- The *regex* property is similar, except that a regular expression can be specified for the match. This regex must conform to Java regex rules. To specify a regex escape value within the regex, use 2 backslashes to escape it (\\).
- Multiple search patterns can be specified per property, separated by 4 vertical bars (`||||`). A match on any of the patterns will trigger the probe failure.
- There are no default values for these parameters. Log scanning is disabled in the default configuration.
- Matching across multiple rows is not supported. The match must occur on one log line.
- For example, to restart the CEF-to-Avro Routing Stream Processor pod when the value, `Setting stream threads to d` (where `d` could be any single digit), is found in the log, change the configuration property "CEF-to-Avro Routing Stream Processor liveness probes regular expression" to the following value .

```
   Setting stream threads to \\d
```

Verification

To verify that log scanning is configured as intended, review the pod's log and look for entries containing `InputStreamScanner`.

For example, to view the c2av-processor pod log, run:

```
  kubectl -n <namespace> logs th-c2av-processor-0 | more
```

For the previous property example, the corresponding log line would be:

```
  InputStreamScanner: Will scan for RegEx pattern [Setting stream threads to \d]
```

Managing Intelligence

This section provides guidance for managing Intelligence functions and features within the deployment.

Modifying Intelligence Analytics Configuration

Intelligence runs Analytics according to the Analytics configuration properties you set during deployment. However, you can modify any of the Analytics configurations, such as enabling Analytics to run on newly ingested data and scheduling when you need Analytics to run. You can also run Analytics on demand.

> **Important**: Reconfiguring Analytics properties causes Analytics to stop and restart all over again or might cause Analytics to fail, if it is already running. Therefore, ensure that Analytics is not running when you reconfigure any of these properties. Check the Analytics pod logs to see if Analytics is already running by executing the following commands on any of the nodes:
>
> ```
> export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
> kubectl -n $NS logs <interset-analytics-pod>
> ```

To modify Analytics configurations:

1. Launch the CDF Management Portal on port 5443.
2. Log in with the following credentials:

   User name: admin

Password: *<the password you provided during CDF installation>*

3. Click ⋮ and then click **Reconfigure**.
4. Click **Intelligence** and modify the desired properties.
5. Click **Save**.

Enabling Windowed Analytics

By default, Intelligence is configured to run Analytics in batch mode. When new data is ingested, Analytics is run on both the new and the existing data. Although this process is beneficial when you first deploy Intelligence (for testing and validation purposes), running Analytics on the entirety of your data on an ongoing basis unnecessarily uses system resources. Instead, you can enable Windowed Analytics.

When you enable Windowed Analytics, you configure Intelligence to run Analytics only on newly ingested data as determined by the date of the last Analytics run and the timestamp of the data. Intelligence identifies the data it has already analyzed, and then runs Analytics only on the new data. These results are then aggregated with the existing results to produce updated, current Analytics results for the entire data set.

Windowed Analytics has a positive impact on performance and stability because it allows the system to analyze and aggregate smaller, more consistently sized quantities of data than batch mode, particularly as the total amount of data in your system continues to grow.

> **Important**: After you have validated the initial data ingest and Analytics run for your Intelligence cluster, you might need to ingest and analyze historical data. In this scenario, you must continue to run Analytics in batch mode to ensure that all data is included.

To enable Windowed Analytics:

1. Launch the CDF Management Portal on port 5443.
2. Log in with the following credentials:

   User name: admin

   Password: *<the password you provided during CDF installation>*
3. Click ⋮ and then click **Reconfigure**.
4. Click Intelligence and disable **Batch Processing**.
5. Click **Save**.

> The first Windowed Analytics run performs a full batch run to establish the baseline for the system going forward. The second and subsequent runs occur as Windowed Analytics.

Running Analytics on Demand

Before you run Analytics on demand, do the following:

- Ensure that Analytics is not already running because running Analytics on demand can cause Analytics in progress to fail.
- If the previous Analytics execution failed, check whether the properties in the Intelligence tab are set correctly. If this does not solve the issue, contact Micro Focus Customer Support.

To run Analytics on demand:

1. Launch a terminal session and log in to the NFS node.
2. Navigate to the following directory:

   ```
   cd <NFSVolume>/interset/analytics
   ```

3. (Conditional) Delete the `blackhawk_down` file, if present. This is an error file and it is generated if the previous Analytics execution fails.

   ```
   rm blackhawk_down
   ```

4. When prompted whether you want to delete the file, enter yes.
5. Execute the following command to delete the latest `AnalyticsStarted.mk` and `AnalyticsCompleted` files:

   ```
   rm -rf AnalyticsStarted-0-<Today's_date>.mk AnalyticsCompleted-0-
   <Today's_date>.mk
   ```

6. When prompted whether you want to delete the files, enter yes.

   After 30 seconds of deletion of the files, Analytics is triggered automatically.

Reconfiguring Intelligence Properties

You can configure the Intelligence properties during deployment and reconfigure them at any point after deployment, as needed.

Important: Reconfiguring Intelligence properties causes Analytics to stop and restart all over again or might cause Analytics to fail, if it is already running. Therefore, ensure that Analytics is not running when you reconfigure any of these properties.

To reconfigure the Intelligence properties:

1. Launch the CDF Management Portal on port 5443.
2. Log in with the following credentials:

   User name: admin

   Password: *<the password you provided during CDF installation>*
3. Click ⋮, then click **Reconfigure**.
4. Click **Intelligence** and reconfigure the properties.
5. Click **Save**.

Changing Passwords for a Secure Environment

You can change the passwords for the components during deployment and also at any point after deployment, as needed.

1. Launch the CDF Management Portal on port 5443.
2. Log in with the following credentials:

   User name: admin

   Password: *<the password you provided during CDF installation>*
3. Click ⋮, then click **Reconfigure**.
4. Click **Intelligence** and modify the passwords.
5. Click **Save**.

Changing the Elasticsearch Node Data Path

Applies only if you have already deployed Intelligence in a cluster.

To change the Elasticsearch node data path, perform the following steps:

1. Launch a terminal session and as a root user, log in to a worker node labeled as **interset:yes**.
2. Execute the following commands to scale down the Elasticsearch master node and

Elasticsearch data nodes:

```
export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
kubectl -n $NS scale statefulset elasticsearch-master --replicas=0
kubectl -n $NS scale statefulset elasticsearch-data --replicas=0
```

3. (Conditional) To create an Elasticsearch data directory in the NFS server, log in to the server.
4. (Conditional)To create a new Elasticsearch data directory in a worker node labeled as **interset:yes**, log in to the node.
5. Execute the following commands to create a new directory:

   ```
   cd <path to create the new directory>
   mkdir <new directory in the path>
   ```

   > If you are creating a new directory in the NFS server, ensure that the directory is accessible or mounted on all the worker nodes labeled as **interset:yes** . The Elasticsearch data directory in the NFS server might impact the system performance.

6. Execute the following command to copy data from the existing directory to the new directory:
   - To copy the data to a worker node labeled as **interset:yes**:

     ```
     cp -rf <existing_directory_path> <new_directory_path>
     ```

     For example:

     ```
     cp -rf /opt/arcsight/k8s-hostpath-volume/interset
     /opt/arcsight/testpath/
     ```

     In this example, the existing directory path `/opt/arcsight/k8s-hostpath-volume/interset` and the new directory path is `/opt/arcsight/testpath/`.

   - To copy the data to the NFS server:

     ```
     scp -rf <existing_directory_path> root@<ip address or hostname of the
     NFS server>:<new_directory_path>
     ```

7. Execute the following command to change the permissions of the new directory:

   ```
   chown 1999:1999* <new_directory_path>
   ```

   For example:

```
chown 1999:1999* /opt/arcsight/testpath/
```

8. If you have created a new Elasticsearch directory in a worker node labeled as **interset:yes**, then repeat Steps 4 to 7 on all the worker nodes labeled as **interset:yes**.
9. Launch the CDF Management Portal on port 5443.
10. Log in with the following credentials:

    User name: admin

    Password: *<the password you provided during CDF installation>*
11. Click ⋮, then click **Reconfigure**.
12. Click Intelligence and provide the new value of the Elasticsearch directory path in the **Elasticsearch Node Data Path to persist data to** field.
13. Click **Save**.
14. Launch a terminal session and as a root user, log in to a worker node labeled as **interset:yes**.
15. Execute the following commands to scale up the Elasticsearch master node and Elasticsearch data nodes:

```
export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
kubectl -n $NS scale statefulset elasticsearch-master --replicas=1
kubectl -n $NS scale statefulset elasticsearch-data --replicas=<number_
of_replicas>
```

16. Execute the following curl command on any Kubernetes node and verify the status of the Elasticsearch cluster:

```
curl -k "https://<Elasticsearch_username:Elasticsearch_password>@<ip
address or hostname of the CDF>:31092/_cluster/health"
```

Enabling Elasticsearch to Start on Limited Hardware Sizing

If Elasticsearch is not able to start because of a lack of CPU resources, you can modify the **Elasticsearch Minimum Cores** field in the CDF Management Portal to enable Elasticsearch to start.

1. Launch the CDF Management Portal on port 5443.
2. Log in with the following credentials:

User name: admin

Password: *<the password you provided during CDF installation>*

3. Click ⋮ , then click **Reconfigure**.
4. Click **Intelligence**.
5. In the **Elasticsearch Configuration** section, modify the value of the **Elasticsearch Minimum Cores** field.

   For example, for a 0.5 CPU, you can specify the corresponding value in any of the following formats:

   - 500m
   - 0.5
6. Click **Save**.

Adding Support for New Devices

Intelligence supports the ingestion and analysis of data of the following data types:

- Access
- Active Directory
- VPN
- Web Proxy
- Repository

For the supported data types, Intelligence also provides support for new devices that provide data of relevance to the Intelligence analytics models. This section provides information on supporting new devices.

Checklist: Implementation

To add the support for new devices, perform the following tasks in the listed order.

	Task	See
□	(Conditional) If SmartConnectors are available for the new device, install and configure SmartConnectors for data collection.	SmartConnectors
□	(Conditional) If SmartConnectors are not available for the new device, install and configure FlexConnectors for data collection.	FlexConnectors

❑	(Conditional) If you have installed and configured FlexConnectors, perform data engineering.	Data Engineering
❑	(Conditional) If you have installed and configured FlexConnectors, perform event categorization.	Event Categorization
❑	Generate SQL Loader Scripts.	SQL Loader Scripts
❑	Update the Intelligence tables required for relations.	Intelligence Tables

SmartConnectors

SmartConnectors are applications that collect events from different devices, process them, and send them to the desired destinations. SmartConnectors are available for the following data types supported by Intelligence:

- Access
- Active Directory
- VPN
- Web Proxy

For more information about the SmartConnectors for the supported data types, see the Supported Data Sources and SmartConnectors/FlexConnectors section. If a new device needs to be supported for any of these data types for which there are corresponding SmartConnectors, then you can configure the SmartConnector for data collection. For more information, see SmartConnector User Guide, SmartConnector Configuration Guides, and Transformation Hub Administration Guide.

FlexConnectors

If there are no SmartConnectors for a new device of the supported data types, you can create FlexConnectors that can read and parse information from the devices and map that information to ArcSight's event schema. FlexConnectors are custom connectors you define to gather security events from log files, databases, and other software and devices. For the data of repository type, that is, GitHub Enterprise, Bitbucket Server, and Perforce, you can create FlexConnectors to collect the data. For every FlexConnector that you create, you need to create a corresponding configuration file. A configuration file is a text file containing properties (name, value pairs) that describe how the FlexConnector parses event data. For more information about FlexConnectors and the configuration files, see ArcSight FlexConnector Developer’s Guide.

Data Engineering

When a new device is supported and a FlexConnector is configured for it, you must identify data fields that are required, on which Intelligence must run analytics. Data engineering is the process of selecting the fields/columns that is required for Intelligence Analytics. This also entails cleansing the data and filtering it from unwanted information, such as noise.

Perform the following steps for data engineering:

1. Clean up data.
2. Filter data.
3. Normalize data. Perform the following as part of normalizing data:
   a. Ensure that the username is in lowercase.
   b. Set the depth value for filepath.
   c. Perform entity mapping.

For more details on data engineering, contact Micro Focus Customer Support.

Event Categorization

When a new device is supported and a FlexConnector is configured for it, you must perform event categorization. Event Categorization is the process of identifying the type and nature of events and categorizing them into groups. Categorizing events is helpful when customizing SQL Loader Scripts to filter specific types of events. For more information, see Event Categorization WhitePaper.

SQL Loader Scripts

To support a new device of the supported data types, you must update the corresponding loader scripts. For more information, contact Micro Focus Customer Support.

Intelligence Tables

The support of a new device necessitates updating the Intelligence schema tables so that Intelligence analytics can run on the data from the new device For more information, contact Micro Focus Customer Support.

Setting an Encoding Option for the URL

For better data security, Intelligence provides options to encode the Intelligence URL string. Based on your requirement, you can set the limit for the URL string length and then select a preferred URL encoding option.

The supported URL encoding options are:

- **plain**: Does not encode and/or compress the URL string.
- **base64**: Compresses the URL string with **zLib** and encodes the string to **base64**.
- **hash**: Stores the encoded **base64** URL string as **JSON** in **localstorage**. Then, it uses a **hash** of the **base64** encoded URL string as the key values.
- **limitLength**: The URL string uses the **plain** and then **base64** encoding options if either of the encoding options have lesser characters than **urlLimit**. The URL string uses the **hash** encoding option if both the **plain** and **base64** encoding options are above **urlLimit**.

  > **urlLimit** is an integer, which sets the maximum URL length (in characters) for encoding options before using **localstorage** . **urlLimit** is only available for the **limitLength** and **limitAndObscure** encoding options.

- **limitAndObscure**: The URL string uses the **base64** encoding option if it has lesser characters than **urlLimit**. The URL uses the **hash** encoding option if it has more characters than **urlLimit**.

> If you do not specify an encoding option, the default encoding option is set as plain.

To set an encoding option for the URL string:

1. Login to the Management portal as the administrator.

   `https://<virtual_FQDN>:5443`
2. Click **CLUSTER** > **Dashboard**. You will be redirected to the **Kubernetes Dashboard**.
3. Under **Namespace**, search and select the `arcsight-installer-xxxx` namespace.
4. Under **Config and Storage**, click **Config Maps**.
5. Click the filter icon, and search for `investigator-default-yaml`.
6. Click ⋮ and select **Edit**.
7. In the **YAML** tab, specify the preferred URL encoding option in **`urlEncoding`** and the preferred URL string length limit in **`urlLimit`**.
8. Click **Update**.
9. Restart the `interset-api` pods:

a. Launch a terminal session and log in to the master or worker node.

b. Execute the following command to retrieve the namespace:

```
export NS=$(kubectl get namespaces | grep arcsight|cut -d ' ' -f1)
```

c. Execute the following commands to restart the `interset-api` pods:

```
kubectl -n $NS scale deployment interset-api --replicas=0
kubectl -n $NS scale deployment interset-api --replicas=2
```

Appendix

This section provides additional information.

Troubleshooting

The following troubleshooting tips may be helpful in resolving issues with your Transformation Hub cluster.

Issue: Installation of master nodes fails

During installation, installation of Master Nodes can fail with the error:

```
     Unable to connect to the server: context deadline exceeded
```

If this occurs, make sure that your `no_proxy` and `NO_PROXY` variables include valid virtual IP addresses and hostnames for each of the master and worker nodes in the cluster, as well as the NFS server.

Issue: Installation times out

During installation, the process may time out with the error:

```
      Configure and start ETCD database
```

If this occurs, make sure your `no_proxy` and `NO_PROXY` variables include correct Master Node information.

Issue: During sudo installation, worker node fails to install

During the Add Node phase, if one or more of the worker nodes fails to install and the log shows the following error message:

```
     [ERROR] : GET Url: https://itom-vault.core:8200/v1/***/PRIVATE_KEY_
CONTENT_{hostname}_{sudo user}, ResponseStatusCode: 404
```

You can take the following steps to rectify the issue:

1. Click **Cancel**. This takes you back to the version selection screen
2. Go through the installation screens again (all previous data is preserved).

3. On the **Add Node** screen, where you added the Worker Node data, remove the worker node which failed by clicking on the **Delete** icon.
4. Click **Add Node** and add the node again.
5. Click **Next** and proceed with the installation.

Issue: Cluster list empty in Kafka Manager

If cluster list is empty in the Kafka Manager UI, delete the existing Kafka Manager pod and try the UI again after a new Kafka Manager pod is back to the Running state.

Issue: Worker nodes out of disk space and pods evicted

If the worker nodes run out of disk space, causing the pods on the node to go into Evicted status, try one of the following steps:

- Fix the disk space issue by adding an additional drive, or by removing unnecessary files.
- On the the node where the low disk space occurred, run the following command:

```
# {install dir} /kubernetes/bin/kube-restart.sh
```

Refer to "Configuring Hard Eviction Thresholds for Worker Nodes" in the *Transformation Hub Administrator's Guide* for information on adjusting the eviction threshold.

Issue: Kafka fails to start up; fails to acquire lock or corrupted index file found

Many scenarios can cause a failure for Kafka to start up and report either `Failed to acquire lock` or `Corrupted index file found.`

Workaround: To resolve this on the problematic Kafka node:.

1. Go to the directory:
```
cd /opt/arcsight/k8s-hostpath-volume/th/kafka/
```

2. Find the file `.lock`, and delete it.

3. Search for all index files:
```
find . -name "*.index" | xargs ls -altr
```

4. Delete all the corrupted index files

5. Restart the affected Kafka pod.

Issue: Slow network or slow VM response during upgrade causes delay or failure of web services operations

An intermittent issue has been observed with web service pod startup, during the upgrade to TH 3.3, that correlates with slow network and/or slow VM response. The pod startup gets blocked or delayed, leading to various issues, such as failing to create new topics and/or failing to register the new schema version.

One error seen in the web service log file is, `"Thread Thread[vert.x-eventloop-thread-0,5,main] has been blocked for 5715 ms, time limit is 2000"`. The workaround is to restart the web service pod.

Issue: ArcSight Database Fails to Restart

If the database fails to start, you can run a set of commands to recover the last known good set of data and restart the database. For example, the database might not restart after an unexpected shutdown. Please consult your database administrator for the commands to run.

Issue: Multiple Node Failures

Here are some considerations when handling node failures on 3 or more worker nodes.

• A Transformation Hub with 3 masters and 3 or more worker nodes should have at least 2 or more master and worker nodes running (quorum) to work properly in high availability.

• As a general rule in terms of data loss prevention, no more than `TOPIC_REPLICATION_FACTOR` minus 1 worker nodes can be down at any time.

Handling failures and stability if Worker nodes go down:

- Resume the stability of the cluster as follows:
  - Repair or replace any down worker nodes or replace with new ones
  - Delete any pods which are in “Terminating” state (this is the expected behavior for stateful pods in Kubernetes when nodes are down).
- Wait until the pod startup sequence is completed. The cluster should resume normal operation.
- Repair any issues on the lost nodes, the cluster should return to Running state

Issue: SmartConnector can’t resolve the short or full hostname of the Transformation Hub node(s)

Error Message: Unable to test connection to Kafka server: [Failed to construct kafka producer

Issue: SmartConnector can resolve the short or full hostname of the Transformation Hub node(s) but can’t communicate with them because of routing or network issues

Error Message: Unable to test connection to Kafka server: [Failed to update metadata after 30000 ms.

Issue: SmartConnector - You may have mistyped the topic name. Try re-entry

Error Message: Unable to test connection to Kafka server: [Failed to update metadata after 40 ms.

Issue: SmartConnector - If using SSL/TLS, you did not configure the SSL/TLS parameters correctly.

Error Message: Destination parameters did not pass the verification with error [; nested exception is: java.net.SocketException: Connection reset]. Do you still want to continue?

Retrieving CDF Root CA

You can retrieve the CDF root CA from a web browser or by using the command line.

Retrieving the CDF Root CA from a Browser

This procedure assumes you are using Google Chrome.

1. Specify the following URL in the browser:

   ```
   https://<master_node_FQDN>:5443
   ```

2. Click the icon next to the left of the URL, and then click **Certificate**.
3. Click **Certification Path**.
4. Double-click the CA certificate. A pop-up window displays.
   a. In the pop-up window, click **Details**, and then click **Copy to File...**.
   b. Click **Next**.
   c. Select **Base-64 encoded X.509 (.CER)** and click **Next**.
   d. Specify a file name (for example, ca.cer) and click **Next**.
   e. Click **Finish** and close the pop-up window.
5. (Conditional) If you have multiple CA certificates, repeat Step 4 for each CA certificate in the certificate chain.

Retrieving the CDF Root CA Using Command Line

1. Log in to the initial master node of the cluster.
2. Execute the following command to retrieve the CDF CA certificate:

   ```
   ${K8S_HOME}/scripts/cdf-updateRE.sh read > ca.cer
   ```

Intelligence Data Types and Schemas

This section provides detailed information about each data type for Intelligence and how it is used in Intelligence Analytics. For each data type, the following information is included:

- A description of the data type
- The supported SmartConnectors for that type
- The schema (with the mandatory columns identified)

Access

Access data sources: sh (Fileshare), rs (Resource)

The Access schema represents events collected from Identity and Access Management (IAM) solutions where users access resources such as servers or fileshares.

Examples of access events include:

- A user fails to access a network share object `VPM-CFDB01.data.int`
- A user attempts to access shared drive `Network Shares/HR/HR-Policies/`

Examples of IAM products include: Active Directory

The Intelligence Access data type best supports Windows Security Log (or Active Directory) event data.

The Microsoft Windows Security Log contains records of login/logout activity, as well as other security-related events specified in the system's Audit Policy. A System Administrator must enable the Windows Audit feature to allow events to be recorded in the Security Log.

Supported SmartConnectors

The following SmartConnectors are used for the collection and ingestion of Access events:

- SmartConnector for Microsoft Windows Event Log – Native Application and System Event Support
- SmartConnector for Microsoft Windows Event Log – Unified Application and System Event Support

Access Schema

The following table describes the Investigation.Events table columns for Access data.

Column Name	Data Type	Required (Y/N)	Description	Example
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT - 2020-06-22 15:22:00
destinatonUserName	Varchar	Y	The user involved in authentication.	john.legget
destinationHostName	Varchar	N	The server handling the authentication.	
filePath	Varchar	N	Path, project, or tag that the resource belongs to.	
fileType	Varchar	N	Type of collection that the resource belongs to, for example, shr	

Column Name	Data Type	Required (Y/N)	Description	Example
fileName	Varchar	N	File, ID, or Object that the resource is mapped to.	
externalId	Varchar	N	Usually a Windows event code (for example, **5140**, **4664**,and so on), but Analytics can be configured to accept other values, including **-1**.	4663
categoryOutcome	Varchar	N	An indicator of whether the authentication was successful. Usually either **success** or **failure** , however, Analytics can be configured to accept other values.	failure

Active Directory

Active Directory data sources: ad

The Active Directory schema represents events collected from Identity and Access Management (IAM) solutions that identify successful and failed log ins to authentication targets. These authentication targets include domain controllers/servers, resources, and file shares.

Examples of authentication events include:

- A user fails to log in to YOURDC.yourcompany.com
- A user attempts to access shared drive DEV_102_share

Examples of IAM products include:

- Active Directory

The Intelligence Authentication data type best supports Windows Security Log (or Active Directory) event data.

The Microsoft Windows Security Log contains records of login/logout activity, as well as other security-related events specified in the system's Audit Policy. A System Administrator must enable the Windows Audit feature to allow events to be recorded in the Security Log.

Supported SmartConnectors

The following SmartConnector is used for the collection and ingestion of Active Directory data:

- SmartConnector for Microsoft Active Directory Windows Event Log Native

Active Directory Schema

The following table describes the Investigation.Events table columns for Active Directory data.

Column Name	Data Type	Required (Y/N)	Description	Example
destinationUserName	Varchar	Y	The user involved in authentication. Primary entity for ad data source.	john.legget
categoryOutcome	Varchar	Y	The outcome of the event. One of success or failure.	success
destinationHostName	Varchar	Y	The target involved in the authentication. Typically the domain controller to which the user is authenticating. The secondary entity for an ad data source. The secondary entity type is always srv.	CONTROLLER3.interset.com
externalId	Varchar	Y	Usually a Windows event code (e.g., 4624, 4771, etc.), but Analytics can be configured to accept other values, including -1.	4624
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT -2020-06-22 15:22:00
destinationNTDomain	Varchar	N	The domain that contains the user that is affected by the event.	interset
categoryObject	Varchar	N	The type of the object.	/Host/Operating System
categoryBehavior	Varchar	N	The action or behavior associated with the event.	Authentication/Verify
deviceCustomString4	Varchar	N	The string that further explains why the user failed to authenticate. Usually a hexadecimal code, but can be any string.	0xc0000064
sourceGeoLocationInfo	Varchar	N	Combination of the latitude and longitude values separated by a comma.	45.1234, -74.4321

VPN

VPN data source: vpn

The VPN schema represents events collected from Identity and Access Management (IAM) solutions or from other VPN devices such as Pulse Secure that identify VPN events.

Examples of VPN events include:

- A Network Policy Server granted full access to a user
- A user failed to authenticate with a Network Policy Server

Examples of IAM products include:

- Active Directory

The Intelligence Authentication data type best supports Windows Security Log (or Active Directory) event data. It also supports login success and failure event data from the supported VPN devices.

The Microsoft Windows Security Log contains records of login/logout activity, as well as other securityrelated events specified in the system's Audit Policy. A System Administrator must enable the Windows Audit feature to allow events to be recorded in the Security Log.

Supported SmartConnectors

The following SmartConnectors are used for the collection and ingestion of VPN data:

- SmartConnector for Microsoft Network Policy Server File
- SmartConnector for Pulse Secure Pulse Connect Secure Syslog
- SmartConnector for Citrix NetScaler Syslog
- SmartConnector for Nortel Contivity Switch Syslog

VPN Schema

The following table describes the Investigation. Events table columns for VPN data.

Column Name	Type	Required (Y/N)	Description	Example
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT - 2020-06-22 15:22:00
sourceUserName	Varchar	Y	The user involved in authentication for Citrix NetScaler device. Primary entity for vpn data source.	john.legget
destinationUserName	Varchar	Y	The user involved in authentication. Primary entity for **vpn** data source.	john.legget
sourceAddressBin	Binary	N Exception: required for IPbased VPN models.	The IP address of the VPN user. Secondary entity	172.1.193.87
sourceGeoCountryCode	Varchar	N Exception: required for countrybased VPN models.	The country the user is authenticating from. Secondary entity	Canada
sourceGeoLatitude	Float	N	The latitude where the VPN connection is initiated.	45.1234
sourceGeoLongitude	Float	N	The longitude where the VPN connection is initiated.	-74.4321
externalId	Varchar	Y	Unique code assigned to a Network Policy Server events. Typically a Windows event code or -1. Analytics can be configured to accept other values.	6272
deviceEventClassId	Varchar	Y	Unique code assigned to a Pulse Secure or Citrix NetScaler event.	AUT24326
deviceAction	Varchar	Y	Unique code assigned to a Nortel event.	OK
categoryOutcome	Varchar	Y	The outcome of the event. One of success or failure. For Citrix NetScaler, the outcome is attempt.	success
categoryBehavior	Varchar	Y	The action or behavior associated with the event.	/Authentication/Verify

Column Name	Type	Required (Y/N)	Description	Example
categoryDeviceGroup	Varchar	Y	The type of events for the device. It is used for Pulse Secure, Citrix NetScaler, and Nortel events.	/VPN
categoryDeviceType	Varchar	Y	The events generated by a device type irrespective of the device group the events belong to. It is used for Citrix NetScaler and Nortel events.	VPN for Nortel Network-based IDS/IPC for Citrix NetScaler
deviceCustomString4	Varchar	N	The string that further explains why the user failed to authenticate. Usually a hexadecimal code, but can be any string. It is used for NPS events with externalId 6273.	18

Web Proxy

Web Proxy data source: pxy

Web Proxy data are raw events that capture network traffic, primarily Web surfing, from a collection of human users.

Examples

- A user accessed the Web site **https://yourcompany.com**
- A user received data from a web destination, **vap3iad3.lijit.com**

Examples of Web Proxy products include:

- Microsoft Internet Security and Acceleration Server (ISA)
- Squid
- Blue Coat Secure Web Gateway

Supported SmartConnectors

The following SmartConnectors are used for the collection and ingestion of Web Proxy data:

- SmartConnector for Microsoft Forefront Threat Management Gateway File
- SmartConnector for Squid Web Proxy Server File
- SmartConnector for Blue Coat Proxy SG Multiple Server File

Web Proxy Schema

The following table describes the Investigation. Events table columns for Web Proxy data.

Column Name	Data Type	Required (Y/N)	Description	Example
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT -2020-06- 22 15:22:00
requestMethod	Varchar	Y	The HTTP method of the request.	GET
deviceSeverity	Varchar	Y	The HTTP response status.	400
bytesIn	Integer	Y	Bytes returned to the client in the response.	410235
sourceUserName	Varchar	N	The name associated with the client making the request.	john.legget
destinationHostName	Varchar	N	The host name of the machine the client is trying to connect to.	a-0001.a-msedge.net
bytesOut	Integer	N	The number of bytes the client sent in its request.	690235
requestClientApplication	Varchar	N	The agent string of the Blue Coat devices.	Mozilla/5.0 (Windows NT 5.1; rv:8.0) Gecko/20100101 Firefox/8.0
deviceCustomString1	Varchar	N	The agent string of the Microsoft devices.	Windows Update Agent
deviceVendor	Varchar	N	The device vendor of the client.	Microsoft
deviceProduct	Varchar	N	The device product of the client.	ISA Server

Repository

Repository data source: rp

Repository data are raw events collected from a source control (repository) system.

Examples:

- A user fetched files from a directory **/project_files/linux/tools/**
- A user added files to a directory **/depot/project5/java_source/**

Information in this section pertain to the following repository systems and their versions:

Repository System	Version
GitHub Enterprise	2.21.0
Bitbucket Server	7.5.0
Perforce	2020.1

The repository systems store audit information in log files. The ArcSight FlexConnectors are installed and configured on the repository systems where they read the log files, filter the messages, tokenise them, and then populate them in the Investigation.Events table. For each of the repository systems and the specified versions, there is a corresponding configuration file (also referred to as a parser). The configuration file is a text file containing properties (name, value pairs) that describe how the FlexConnector parses event data.

The FlexConnector type that is used to process and parse the repository log files is the ArcSight FlexConnector Regex File.

Configuration Files

The configuration files provided in this section are designed only for the specified versions of the repository systems.

Configuration File for GitHub Enterprise 2.21.0

The configuration file that is used for GitHub Enterprise 2.21.0 is **git.sdkrfilereader.properties**.

```
text.qualifier="
comments.start.with=\#
trim.tokens=true
contains.empty.tokens=true

line.include.regex=(.+)"committer_date":"([^ ]+)(.+)"hostname":"([^,]+)"
(.+)"program":("upload-pack"|"run-hook-postreceive")(.+)"
real_ip":"([^,]+)"(.+)"repo_name":"([^,]+)"(.+)"user_login":"([^,]+)"(.+)
regex=(.+)"committer_date":"([^ ]+)(.+)"hostname":"([^,]+)"(.+)"program":"
([^,]+)"(.+)"real_ip":"([^,]+)"(.+)"repo_
name":"([^,]+)"(.+)"user_login":"([^,]+)"(.+)
token.count=13

token[0].name=CONSTANT1
token[0].type=String
token[1].name=EVENTTIME
```

```
token[1].type=Long
token[2].name=CONSTANT2
token[2].type=String
token[3].name=HOSTNAME
token[3].type=String
token[4].name=CONSTANT2
token[4].type=String
token[5].name=PROGRAM
token[5].type=String
token[6].name=CONSTANT3
token[6].type=String
token[7].name=REALIP
token[7].type=String
token[8].name=CONSTANT4
token[8].type=String
token[9].name=REPONAME
token[9].type=String
token[10].name=CONSTANT5
token[10].type=String
token[11].name=USERNAME
token[11].type=String
token[12].name=CONSTANT6
token[12].type=String

event.deviceVendor=__getVendor("GitHub")
event.deviceProduct=__stringConstant("GitGub Enterprise")
event.deviceVersion=__stringConstant("2.21.0")

event.deviceReceiptTime=__createLocalTimeStampFromSecondsSinceEpoch
(EVENTTIME)
event.destinationUserName=USERNAME
event.deviceCustomString1=__toLowerCase(REPONAME)
event.deviceCustomString1Label=__stringConstant("RepositoryName")
event.deviceAction=__ifThenElse(PROGRAM,"run-hook-post-receive","receive-
pack","upload-pack")
event.sourceAddress=__oneOfAddress(REALIP)
event.destinationHostName=__oneOfHostName(HOSTNAME)
event.name=__ifThenElse(PROGRAM,"run-hook-post-receive","receive-
pack","upload-pack")
event.bytesOut=__safeToInteger(__regexToken(CONSTANT5,".+uploaded_bytes.:
([^,]+)"))
#event.requestMethod=
#event.protocol=
#event.request=

event.categoryObject=__stringConstant("/Host/Resource")
event.categoryBehavior=__stringConstant("/Access")
event.categoryOutcome=__stringConstant("/Attempt")
event.categorySignificance=__stringConstant("/Informational")
event.categoryDeviceGroup=__stringConstant("Application")
event.categoryDeviceType=__stringConstant("Repository")
```

Configuration File for Bitbucket Server 7.5.0

The configuration file that is used for Bitbucket Server 7.5.0 is **bitbucket.sdkrfilereader.properties**.

```
text.qualifier="
comments.start.with=\#
trim.tokens=true
contains.empty.tokens=true

line.include.regex=(.+)\\|(.+)\\|(.+)\\|([^-]+)\\|(.+)\\|(.+git-upload-
pack.+|.+git-receive-pack.+)\\|(.+)\\|(.+)\\|
(.+)\\|(.+)\\|(.+)\\|(.+)\\|(.+)\\|(.*)
regex=(.+)\\|(.+)\\|(.+)\\|(.*)\\|(.+)\\|(.*)\\|(.+)\\|(.+)\\|(.+)\\|
(.+)\\|(.+)\\|(.+)\\|(.+)\\|(.*)

token.count=14

token[0].name=REALIP
token[0].type=String
token[1].name=PROTOCOL
token[1].type=String
token[2].name=REQUESTID
token[2].type=String
token[3].name=USERNAME
token[3].type=String
token[4].name=EVENTTIME
token[4].type=String
token[5].name=ACTION
token[5].type=String
token[6].name=REQUESTINFO
token[6].type=String
token[7].name=STATUS
token[7].type=String
token[8].name=BYTESREAD
token[8].type=String
token[9].name=BYTESWROTE
token[9].type=String
token[10].name=EXTRAINFO1
token[10].type=String
token[11].name=EXTRAINF02
token[11].type=String
token[12].name=EXTRAINF03
```

```
token[12].type=String
token[13].name=EXTRAINF04
token[13].type=String

event.deviceVendor=__getVendor("BitBucket")
event.deviceProduct=__stringConstant("BitBuket Server")
event.deviceVersion=__stringConstant("7.5.0")

event.deviceReceiptTime=__createOptionalTimeStampFromString
(EVENTTIME,"yyyy-MM-dd HH:mm:ss,sss")
event.destinationUserName=USERNAME
event.deviceCustomString1=__toLowerCase(__regexToken(__regexToken(__split
(ACTION," ",2),"(.*)\\.git(.+)"),".*\/(.+)"))
event.deviceCustomString2=__regexToken(__split(ACTION," ",2),"(\/.+)
(\/git-upload-pack|\/git-receive-pack)")
event.deviceCustomString2Label=__stringConstant("RepositoryName")
event.name=__regexToken(__split(ACTION," ",2),".+\/(.+)")
event.sourceAddress=__oneOfAddress(REALIP)
event.sourceHostName=__oneOfHostName(REALIP)
event.deviceAction=__regexToken(__split(ACTION," ",2),".+\/(.+)")
event.bytesIn=__safeToInteger(BYTESREAD)
event.bytesOut=__safeToInteger(BYTESWROTE)
event.requestMethod=__ifThenElse(__contains
(ACTION,"POST"),"true","POST","GET")
event.requestUrl=__split(ACTION," ",2)

event.categoryObject=__stringConstant("/Host/Resource")
event.categoryBehavior=__stringConstant("/Access")
event.categoryOutcome=__ifThenElse(STATUS,"200","/Success",__ifThenElse
(STATUS,"401","/Denied","/Attempt"))
event.categorySignificance=__stringConstant("/Informational")
event.categoryDeviceGroup=__stringConstant("Application")
event.categoryDeviceType=__stringConstant("Repository")
```

Configuration File for Perforce 2020.1

The configuration file that is used for Perforce 2020.1 is **perforce.sdkrfilereader.properties**.

```
text.qualifier="
comments.start.with=\#
trim.tokens=true
contains.empty.tokens=true

regex=(.+)\\s(.+)\\s(.+)\\s(.+)\\s(.+)\\s(.+)

token.count=6
```

```
token[0].name=EVENTDATE
token[0].type=String
token[1].name=EVENTTIME
token[1].type=String
token[2].name=USER
token[2].type=String
token[3].name=CLIENTIP
token[3].type=String
token[4].name=ACTION
token[4].type=String
token[5].name=RESOURCE
token[5].type=String

event.deviceVendor=__getVendor("Perforce")
event.deviceProduct=__stringConstant("Perforce")
event.deviceVersion=__stringConstant("2020.1")

event.deviceReceiptTime=__createOptionalTimeStampFromString(__concatenate
(EVENTDATE,EVENTTIME),"yyyy/MM/ddHH:mm:ss")
event.destinationUserName=USER

#########################################################################
#1.\/\/([^\/]+)\/([^\/]+)\/([^\/]+).*","/","//","")
# will return max of depth 4
# __regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//","")
# eg //csvr/A/B/C
# //csrv/main/null
# //csrv/null/null
# //csrv/A/master
#2.\/\/(.*)(?=\/main$|\/null$|\/rel$|\/master$)
#__regexToken(__regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?
([^\/]+)?\/?([^\/]+)?","/","//",""),"\/\/(.*)
(?=\/main$|\/null$|\/rel$|\/master$)")
#eg.returns all info nothign with main/null/rel/master
#3. remove version if any
#__regexToken(__ifGreaterOrEqual(__length(__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"\/\/(.*)
(?=\/main$|\/null$|\/rel$|\/master$)")),"1",__regexToken(__
regexTokenFindAndJoin
(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?([^\/]+)?","/","//",""),"\/\/(.*)
(?=\/main$|\/null$|\/rel$|\/master$)"),__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//","")),"(.*)[#\/][\\d.]+")
#eg.//crsv/A/12.3
# //crsv/A#1.2
##########################################################################
```

```
event.deviceCustomString1=__ifGreaterOrEqual(__length(__regexToken(__
ifGreaterOrEqual(__length(__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)
(?=\/main$|\/null$|\/rel$|\/master$)")),"1",__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)(?=\/main$|\/null$|\/rel$|\/master$)"),__
regexTokenFindAndJoin(RESOURCE,"\/\/
([^\/]+)?\/?([^\/]+)?\/?([^\/]+)?","/","//","")),"(.*)[#\/]
[\\d.]+")),"1",__regexToken(__ifGreaterOrEqual(__length(__
regexToken(__regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)
(?=\/main$|\/null$|\/rel$|\/master$)")),"1",__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)(?=\/main$|\/null$|\/rel$|\/master$)"),__
regexTokenFindAndJoin(RESOURCE,"\/\/
([^\/]+)?\/?([^\/]+)?\/?([^\/]+)?","/","//","")),"(.*)[#\/][\\d.]+"),__
ifGreaterOrEqual(__length(__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)
(?=\/main$|\/null$|\/rel$|\/master$)")),"1",__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)(?=\/main$|\/null$|\/rel$|\/master$)"),__
regexTokenFindAndJoin(RESOURCE,"\/\/
([^\/]+)?\/?([^\/]+)?\/?([^\/]+)?","/","//","")))
event.deviceCustomString2=RESOURCE
event.deviceAction=ACTION
event.sourceAddress=__oneOfAddress(CLIENTIP)
event.sourceHostName=__oneOfHostName(CLIENTIP)
event.name=ACTION

event.categoryObject=__stringConstant("Host/Resource")
event.categoryBehavior=__stringConstant("Access")
event.categoryOutcome=__stringConstant("/Attempt")
event.categorySignificance=__stringConstant("/Informational")
event.categoryDeviceGroup=__stringConstant("Application")
event.categoryDeviceType=__stringConstant("Repository")
```

You can also create or customize the configurations files for other versions of the repository systems.

FlexConnector Installation and Configuration

Ensure the following when you install and configure the FlexConnector:

- Select **ArcSight FlexConnector Regex** File as the **Connector Type**.
- When adding the parameters information, specify the following:
  - Select **Log Unparsed Events** as **False**.
  - Provide the absolute path and the repository log file name that the FlexConnector needs to read in the **Log File Name** field.

    For example:
    c:\temp\sample_data.log
  - For the **Configuration File** field, depending on the repository on which you are installing the FlexConnector, specify only **git**, **bitbucket**, or **perforce**.

    For example, for the GitHub Enterprise repository, you must specify only git. The suffix **.sdkrfilereader.properties** is appended automatically. The configuration file name now is **git.sdkrfilereader.properties**.
- When configuring the destination, select either **CEF File** or Transformation Hub as the destination.

Post-Installation Tasks

After you install and configure the FlexConnector and before you run the FlexConnector, copy the desired configuration (parser) files in the **ARCSIGHT_HOME\user\agent\ flexagent** location.

Repository Schema

The following table describes the Investigation.Events table columns for Repository data.

Column Name	Type	Required (Y/N)	Description	Example
deviceAction	Varchar	Y	The action performed on the device.	upload-pack
deviceCustomString1	Varchar	Y	The device involved in the event. Typically a file path. Can be any string identifying a repository. Secondary entity for the rp data source	dev3/rel/hydra
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT -2020-06-22 15:22:00
destinationUserName	Varchar	Y	The user involved in the event. Primary entity	john.legget
deviceVendor	Varchar	Y	The device vendor of the client.	GitHub

Column Name	Type	Required (Y/N)	Description	Example
deviceProduct	Varchar	N	The device product of the client.	GitHub Server
deviceVersion	Integer	N	The device version.	2.21.0
categoryObject	Varchar	N	The type of the object.	Host/Resource
categoryBehavior	Varchar	N	The action or behavior associated with the event.	/Access
categoryOutcome	Varchar	Y	The outcome of the event.	/Attempt
cateorySignificance	Varchar	N	The significance of the event.	/Informational
categoryDeviceGroup	Varchar	Y	The type of events for the device.	Application
categoryDeviceType	Varchar	N	The events generated by the device type irrespective of the device group the events belong to.	Repository
sourceAddressBin	Varchar	N	The IP address of the user involved in the event.	78.1.198.82
bytesOut/bytesIn	Integer	N	The size of data (in bytes) related to the action performed on the project.	2203

CDF Installer Script `install.sh` Command Line Arguments

On-Premises CDF install.sh Command Line Arguments

Argument	Description
`--auto-configure-firewall`	Flag to indicate whether to auto configure the firewall rules during node deployment. The allowable values are true or false. The default is true.
`--cluster-name`	Specifies the logical name of the cluster.
`--deployment-log-location`	Specifies the absolute path of the folder for placing the log files from deployments.
`--docker-http-proxy`	Proxy settings for Docker. Specify if accessing the Docker hub or Docker registry requires a proxy. By default, the value will be configured from the `http_proxy` environment variable on your system.
`--docker-https-proxy`	Proxy settings for Docker. Specify if accessing the Docker hub or Docker registry requires a proxy. By default, the value will be configured from `https_proxy` environment variable on your system

Argument	Description
`--docker-no-proxy`	Specifies the IPv4 addresses or FQDs that do not require proxy settings for Docker. By default, the value will be configured from the no_proxy environment variable on your system.
`--enable_fips`	This parameter enables suites to enable and disable FIPS. The expected values are true or false. The default is *false*.
`--fail-swap-on`	If 'swapping' is enabled, specifies whether to make the kubelet fail to start. Set to true or false. The default is *true*.
`--flannel-backend-type`	Specifies flannel backend type. Supported values are vxlan and host-gw. The default is host-gw.
`--ha-virtual-ip`	A Virtual IP (VIP) is an IP address that is shared by all master nodes. The VIP is used for the connection redundancy by providing failover for one machine. Should a master node fail, another master node takes over the VIP address and responds to requests sent to the VIP. Mandatory for a Multi-Master cluster; not applicable to a single-master cluster The VIP must be resolved (forward and reverse) to the VIP Fully Qualified Domain Name (FQDN)
`--k8s-home`	Specifies the absolute path of the directory for the installation binaries. By default, the Kubernetes installation directory is /opt/arcsight/kubernetes.
`--keepalived-nopreempt`	Specifies whether to enable nopreempt mode for KeepAlived. The allowable value of this parameter is true or false. The default is true and KeepAlived is started in nopreempt mode.
`--keepalived-virtual-router-id`	Specifies the virtual router ID for KEEPALIVED. This virtual router ID is unique for each cluster under the same network segment. All nodes in the same cluster should use the same value, between 0 and 255. The default is 51.
`--kube-dns-hosts`	Specifies the absolute path of the hosts file which used for host name resolution in a non-DNS environment. **Note:** Although this option is supported by the CDF Installer, its use is strongly discouraged to avoid using DNS resolution in production environments, due to hostname resolution issues and the nuances involved in their mitigations.
`--load-balancer-host`	IP address or host name of load balancer used for communication between the master nodes. For a multiple master node cluster, it is required to provide `–load-balancer-host` or `–ha-virtual-ip` arguments.
`--master-api-ssl-port`	Specifies the https port for the Kubernetes (K8S) API server. The default is 8443.
`--nfs-folder`	Specifies the path to the NFS core volume.

Argument	Description
`--nfs-server`	Address of the NFS host.
`--pod-cidr-subnetlen`	Specifies the size of the subnet allocated to each host for pod network addresses.
`--pod-cidr`	Specifies the private network address range for the Kubernetes pods. Default is 172.16.0.0/16. The minimum useful network prefix is /24. The maximum useful network prefix is /8. This must not overlap with any IP ranges assigned to services (see `--service-cidr` parameter below) in Kubernetes. The default is 172.16.0.0/16.
`--registry_orgname`	The organization inside the public Docker registry name where suite images are located. Not mandatory. Choose one of the following: • Specify your own organization name (such as your company name). For example: `--registry-orgname=Mycompany.` • Skip this parameter. A default internal registry will be created under the default name HPESWITOM.
`--runtime-home`	Specifies the absolute path for placing Kubernetes runtime data. By default, the runtime data directory is `${K8S_HOME}/data.`
`--service-cidr`	Kubernetes service IP range. Default is 172.30.78.0/24. Must not overlap the POD_CIDR range. Specifies the network address for the Kubernetes services. The minimum useful network prefix is /27 and the maximum network prefix is /12. If SERVICE_CIDR is not specified, then the default value is 172.17.17.0/24. This must not overlap with any IP ranges assigned to nodes for pods. See `--pod-cidr.`
`--skip-check-on-node-lost`	Option used to skip the time synchronization check if the node is lost. The default is true.
`--skip-warning`	Option used to skip the warnings in precheck when installing the Initial master Node. Set to true or false. The default is false.
`--system-group-id`	The group ID exposed on server; default is 1999.
`--system-user-id`	The user ID exposed on server; default is 1999.
`--thinpool-device`	Specifies the path to the Docker devicemapper, which must be in the `/dev/mapper/` directory. For example: `/dev/mapper/docker-thinpool`

Argument	Description
`--tmp-folder`	Specifies the absolute path of the temporary folder for placing temporary files. The default temporary folder is `/tmp.`
`-h, --help`	Displays a help message explaining proper parameter usage
`-m, --metadata`	Specifies the absolute path of the tar.gz suite metadata packages.

Azure CDF install Script Command Line Arguments (Optional)

Argument	Description
`-c, --config`	Absolute path of the configuration json file for silent installation.
`-d, --deployment-name, -n`	Deployment name for suite installation. (Note: -n is to be deprecated in future versions.)
`--backup-vol-size`	Specifies the volume size of pg-backup component. The size must be a plain integer or as a fixed-point integer and the unit must be one of E,P,T,G,M,K,Ei,Pi,Ti,Gi,Mi,Ki; example: 10Gi
`-fg, --feature-gates`	A set of key=value pairs that describe feature gates for alpha/experimental features. The allowable value of this parameter is mapStringBool. Comma-delimited list of strings, each entry format is NameOfFeature=true|false. Options are: • MultipleDeployment=true|false (Alpha - default=false) • Bosun=true|false (Alpha - default=false) • Prometheus=true|false (Alpha - default=false)
`--nfs-server`	Specifies the server for NFS, used to create persistent volume claim 'itom-vol-claim'
`--nfs-folder`	Specifies the folder for NFS, used to create persistent volume claim 'itom-vol-claim'.
`--loadbalancer-info`	Specifies the loadbalancer info. This parameter value formats such as: "KEY1=VALUE1;KEY2=VALUE2;...;KEYn=VALUEn" Example: For gcp: --loadbalancer-info "LOADBALANCERIP=x.x.x.x" For alicloud: --loadbalancer-info "LOADBALANCERID=xxx"
`--logging-vol-size`	Specifies the volume size of fluentd component. The size must be a plain integer or as a fixed-point integer and the unit must be one of E,P,T,G,M,K,Ei,Pi,Ti,Gi,Mi,Ki; example: 10Gi
`-P, password`	Specifies the password for suite administrator which will be created during installation. Wrap the password with single quotes. For example, 'Password@#$!123'.
`--registry-orgname`	Specifies the organization name(namespace) where the suite images are placed. The default name is 'hpeswitom'.
`--registry-ca`	Specifies the path of trusted CA root certificate (bas64 X.509 format) of external registry.
`--registry-password`	Specifies the password for registry.
`--registry-password-file`	Specifies the password file for registry.
`--skip-warning`	Option used to skip the warning(s) in precheck when install.

Argument	Description
`--tmp-folder`	Specifies the absolute path of the temporary folder for placing temporary files. The default temporary folder is '`/tmp`'.
`--db-user`	External suite database user name.
`--db-password`	External suite database password.
`--db-url`	External suite database connection URL.
`--db-crt`	External suite database connection certificate.
`--registry-url`	Specifies the registry for URL.
`--registry-username`	Specifies the username for registry.
`--external-access-host`	Specifies the external access host.
`--cloud-provider`	Specifies the cloud provider when installing CDF on a cloud server. The allowable value of this parameter is 'alicloud', 'gcp' (case- sensitive)

Reinstalling CDF

If you uninstalled CDF and plan to reinstall CDF and Intelligence in the same cluster, perform the following steps before reinstalling CDF and Intelligence:

1. Launch a terminal session and as a root user, log in to the node where NFS is present.
2. Delete the NFS directory:

   ```
   rm -rf /<nfs directory path>/nfs
   ```

3. Launch a terminal session and then log in to the node where the Kubernetes hostpath is present.
4. Navigate to the following directory:

   ```
   cd /opt/arcsight/
   ```

5. Delete the following directory:

   ```
   rm -rf k8s-hostpath-volume
   ```

6. Repeat Step 4 and Step 5 on all CDF worker nodes.
7. Launch a terminal session and then log in to a database node.
8. Navigate to the following directory:

   ```
   cd /[database_install_directory]/
   ```

9. Stop the Kafka Scheduler:

   ```
   ./kafka_scheduler stop
   ```

10. As a dbadmin user, do the following:
    a. Execute the following command and specify your dbadmin password:

       ```
       /opt/vertica/bin/vsql
       Password:<password>
       ```

    b. Execute the following command to delete the data in the investigation.events table:

       ```
       DELETE FROM investigation.events;
       ```

    c. Execute the following command to delete the UEBA schema:

       ```
       drop schema UEBA cascade;
       ```

11. Continue with reinstalling CDF and deploying Intelligence. Do one of the following:
    a. Deploy Intelligence manually. For more information, see Deploying Intelligence Manually.

    b. Deploy Intelligence by using the ArcSight Platform Installer.

Retrieving CDF Root CA

You can retrieve the CDF root CA from a web browser or by using the command line.

Retrieving the CDF Root CA from a Browser

This procedure assumes you are using Google Chrome.

1. Specify the following URL in the browser:

   ```
   https://<master_node_FQDN>:5443
   ```

2. Click the icon next to the left of the URL, and then click **Certificate**.

3. Click **Certification Path**.
4. Double-click the CA certificate. A pop-up window displays.
   a. In the pop-up window, click **Details**, and then click **Copy to File...**.
   b. Click **Next**.
   c. Select **Base-64 encoded X.509 (.CER)** and click **Next**.
   d. Specify a file name (for example, ca.cer) and click **Next**.
   e. Click **Finish** and close the pop-up window.
5. (Conditional) If you have multiple CA certificates, repeat Step 4 for each CA certificate in the certificate chain.

Retrieving the CDF Root CA Using Command Line

1. Log in to the initial master node of the cluster.
2. Execute the following command to retrieve the CDF CA certificate:

```
${K8S_HOME}/scripts/cdf-updateRE.sh read > ca.cer
```

Understanding the Database Installer Options

To specify an option:

Type `./db_installer <Option_Name>`.

Option Name	Description
install	Installs the database
uninstall	Uninstalls the database and deletes data and users
create-schema	Creates the database schema for Recon/Intelligence
delete-schema	Deletes the Recon/Intelligence database schema
start-db	Starts the database with the `dba_password` specified in `db_credentials.properties`
stop-db	Stops the database
status	Prints the database cluster status

Understanding Labels and Pods

During installation, you apply labels, which are associated with the deployed capabilities, to the Worker nodes in the Kubernetes cluster. The labels tell Kubernetes the types of workloads that can run on a specific host system. Based on the labels, Kubernetes then assigns pods to the nodes to provide functions, tasks, and services. Each pod belongs to a specific namespace in the CDF Management portal. On occasion, you might need to restart pods or reconfigure the environment by moving labels to different nodes, thus reassigning the workload of the pods.

> When using the CDF Management Portal, the label format is `<label name>:yes`. However, when using the kubectl command line the label format is `<label name>=yes`.

- "Adding Labels to Worker Nodes" below
  - "fusion:yes" on the next page
  - "interset:yes " on page 587
  - "interset-datanode:yes" on page 588
  - "interset-namenode:yes" on page 588
  - "interset-spark:yes" on page 589
  - "kafka:yes" on page 589
  - "th-platform:yes" on page 589
  - "th-processing:yes" on page 590
  - "zk:yes " on page 591
- "Understanding the Pods that Do Not Have Labels" on page 591

Adding Labels to Worker Nodes

Depending on the capabilities that you deploy, you must to assign certain a set of labels to the Worker Nodes. Eachof the following sections defines the pods and their associated capabilities that get installed per assigned label.

To avoid issues caused by conflicting label assignments, review the following considerations.

- **Labeling for the Intelligence capability**
  - The HDFS NameNode, which corresponds with the `interset-namenode:yes` label, should run on one worker node only. The worker node must match the hostname or IP address that you provided in the **HDFS NameNode** field in the **CDF Management Portal** > **Configure/Deploy** page > **Intelligence**.

  - Assign the label for Spark2, `interset-spark:yes`, to the same worker nodes where you placed the `interset-datanode:yes` label.
- For Transformation Hub's Kafka and ZooKeeper, make sure that the number of the nodes you have labeled corresponds to the number of worker nodes in the Kafka cluster and the number of worker nodes running Zookeeper in the Kafka cluster properties from the pre-deployment configuration page. The default number is 3 for a Multiple Worker deployment.
- Although ESM Command Center, Recon, Intelligence, and SOAR all require Fusion, you do not need to assign the label for Fusion to more than one worker node.

fusion:yes

The Fusion capability includes many of the core services needed for your deployed products, including the Dashboard and user management. With the exception of Transformation Hub, all deployed capabilities require Fusion. Add the `fusion:yes` label to the Worker Nodes where you want to run the associated pods. For high availability, add this label to multiple worker nodes.

Pod	Description	Namespace	Associated Capability
esm-acc-web-app	Manages the user interface for ESM Command Center. The interface connects to an ESM Manager server running outside the Kubernetes cluster.	arcsight-installer	ESM Command Center
esm-web-app	Manages how ESM Command Center links to main navigation of the Platform user interface.	arcsight-installer	ESM Command Center
esm-widgets	Manages the dashboards and widgets that are designed to incorporate data from ESM. The widgets connect to an ESM Manager server running outside of the Kubernetes cluster. For example, when you start this pod, it installs the provided *How is my SOC running?* dashboard.	arcsight-installer	ESM Command Center
fusion-common-doc-web-app	Provides the context-sensitive user guides for Fusion (the Platform), Recon, and Reporting.	arcsight-installer	Fusion
fusion-dashboard-metadata-web-app	Manages the REST API for the metadata of the Dashboard feature.	arcsight-installer	Fusion
fusion-dashboard-web-app	Manages the framework, including the user interface, for the Dashboard feature.	arcsight-installer	Fusion

Pod	Description	Namespace	Associated Capability
fusion-db-monitoring-web-app	Manages the REST API for the database monitoring function.	arcsight-installer	Fusion
fusion-db-search-engine	Provides APIS to access data in the ArcSight Database. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Fusion
fusion-metadata-rethinkdb	Manages the RethinkDB database, which stores information about a user's preferences and configurations.	arcsight-installer	Fusion
fusion-single-sign-on	Manages the SSO service that enables users to log in to any of the deployed capabilities and the consoles for ArcSight Intelligence, SOAR, and ESM Command Center.	arcsight-installer	Fusion
fusion-ui-services	Manages the framework, including the user interface, for the primary navigation functions in the user interface.	arcsight-installer	Fusion
fusion-user-management	Manages the framework, including the user interface, for the user management function.	arcsight-installer	Fusion
interset-widgets	Manages the widgets that are designed to incorporate data from ArcSight Intelligence. The widgets connect to an Interset server running outside of the Kubernetes cluster.	arcsight-installer	Intelligence
layered-analytics-widgets	Manages and installs the widgets that can incorporate data from multiple capabilities. For example, the provided *Entity Priority* widget connects to ESM Command Center and Intelligence servers outside the Kubernetes cluster to display entity data.	arcsight-installer	Layered Analytics
recon-analytics	Manages the backend of Outlier Analytics; the user interface for Outlier Analytics is managed by the recon-search-web-app pod.	arcsight-installer	Recon
recon-search-web-app	Manages the Search, Lookup lists, and Data Quality Dashboard functions, as well as the user interface for Outlier Analytics.	arcsight-installer	Recon
reporting-web-app	Manages the REST API and user interface for the Reporting feature. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Recon

Pod	Description	Namespace	Associated Capability
recon-search-and-storage-web-app	Manages the configuration of and sends events to storage groups.	arcsight-installer	Recon
soar-artemis	Manages the SOAR message broker.	arcsight-installer	SOAR
soar-web-app	Manages the user interface and services for the SOAR capability.	arcsight-installer	SOAR

interset:yes

Add the `interset:yes` label to Worker Nodes where you want to run the pods that manage functions and services for the ArcSight Intelligence capability. For high availability, add this label to multiple worker nodes.

Pod	Description	Namespace	Associated Capability
elasticsearch-data	Manages the Elasticsearch functions that store all raw events for Intereset Analytics and provide all data that drives the user interface.	arcsight-installer	Intelligence
elasticsearc-master	Manages theElasticsearch services.	arcsight-installer	Intelligence
h2	Stores user identities required to authenticate and authorize users.	arcsight-installer	Intelligence
interset-analytics	Determines the individual baselines , then discovers and ranks devisions from those baselines for theInterset Analytics feature. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Intelligence
interset-api	Manages the REST API that the Intelligence user interface uses to gather the Interset Analytics results. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Intelligence
interset-exports	Generates the PDF reports of organization risks and the users involved in risky behaviors.	arcsight-installer	Intelligence

Pod	Description	Namespace	Associated Capability
interset-logstash	Manages Logstash, which collects raw events from Transofrmation Hub and sends them to Elasticsearch for indexing.	arcsight-installer	Intelligence
interset-spark-config-file-server	Hosts a file server to provide configuraiton files for Spark2 to consume.	arcsight-installer	Intelligence
interset-ui	Manages the user interfacethat displays the Interset Analytics results and the raw data in the Interset dashboard	arcsight-installer	Intelligence

interset-datanode:yes

Add the `intelligence-datanode:yes` label to Worker Nodes where you want to run the pods that manage HDFS services for the ArcSight Intelligence capability.

> ⚠ Place this label on one worker node only. The worker node must match the hostname or IP address that you provided in the `HDFS NameNode` field in the **CDF Management Portal** > **Configure/Deploy** > **Intelligence**.

Pod	Description	Namespace	Associated Capability
hdfs-datanode	Manages how HDFS stores the results of Interset Analytics searches before transferring them to the ArcSight database. The HDFS DataNodes contain blocks of HDFS files.	arcsight-installer	Intelligence

interset-namenode:yes

Add the `interset-namenode:yes` label to a Worker Node for the HDFS NameNode.

Pod	Description	Namespace	Associated Capability
hdfs-namenode	Manages how the HDFS NameNode stores the location of all HDFS files distributed across the cluster.	arcsight-installer	Intelligence

interset-spark:yes

Add the `interset-spark:yes` label to Worker Nodes where you want to run the Analytics services for the ArcSight Intelligence capability. For high availability, add this label to multiple worker nodes. To reduce network traffic, add the label to the same worker nodes where you placed the `interset-datanode:yes` label.

Pod	Description	Namespace	Associated Capability
Spark2	Launches when users run the Interset Analytics feature. Spark2 generates multiple pods, changing the names of the pods according to the different phases of the analytics tasks.	arcsight-installer	Intelligence

kafka:yes

Add the `kafka:yes` label to Worker Nodes where you want to run the Kafka Broker functions and services for the Transformation Hub capability.

Ensure that you assign this label to the same quantity of nodes that you specified for the `# of Kafka broker nodes in the Kafka cluster` setting in the **CDF Management Portal** > **Configure/Deploy** > **Transformation Hub** > **Kafka and Zookeeper Configuration**. The default number is 3.

Pod	Description	Namespace	Associated Capability
th-kafka	Manages the Kafka Broker, to which publishers and consumers connect so they can exhange messages over Kafka. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Transformation Hub

th-platform:yes

Add the `th-platform:yes` label to Worker Nodes where you want to run the Kafka Manager, schema registry, and WebServices for the Transformation Hub capability. For high availability, add this label to multiple worker nodes.

Pod	Description	Namespace	Associated Capability
th-kafka-manager	Provides the user interface that allows the Kafka Manager to manage the Kafka Brokers.	arcsight-installer	Transformation Hub
th-schemaregistry	Provides the scheme registry that is used for managing the schema of data in Avro format. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Transformation Hub
th-web-service	Manages the WebServices module of Transformation Hub. WebServices provides the API that ArcMC uses to retrieve data. NOTE: This pod requires communication outside of the Kubernetes cluster to receive client requests from and initiate connections to ArcMC.	arcsight-installer	Transformation Hub

th-processing:yes

Add the `th-processing:yes` label to Worker Nodes where you want to run services that manage processing for the Transformation Hub capability. For high availability, add this label to multiple worker nodes.

Pod	Description	Namespace	Associated Capability
th-c2av-processor	Manages the instances that convert CEF messages on the topic th-cef to Avro on the topic th-arcsight-avro. The quantity of instances depends on the number of partition in the th-cef topic and load. The default is 0 instances.	arcsight-installer	Transformation Hub
th-cth	Manages up to 50 instances of connectors in Transformation Hub that distribute the load of data received from collectors by creating a consumer group that is based on the source top and destination and topic names.	arcsight-installer	Transformation Hub
th-c2av-processor-esm	Manages the instances that convert CEF messges on the topic mf-event-cef-esm-filtered to Avro on the topic mf-eent-avro-emsfiltered. The quantity of instances depends on the number of partition in the th-cef topic and load. The default is 0 instances.	arcsight-installer	Transformation Hub
th-routing-processor-group	Manages the routing rules for topics. Use ArcMC to configure the rules.	arcsight-installer	Transformation Hub

zk:yes

Add the `th-zookeeper:yes` label to Worker Nodes where you want to Kafka Zookeeper for the Transformation Hub capability.

> ⚠ Ensure that you assign this label to the same quantity of nodes that you specified for the `# of Zookeeper nodes in the Zookeper cluster` setting in the **CDF Management Portal** > **Configure/Deploy** > **Transformation Hub** > **Kafka and Zookeeper Configuration**. The default number is 3.

Pod	Description	Namespace	Associated Capability
th-zookeeper	Manages Kafka Zookeeper, which stores metadata about partitions and brokers.	arcsight-installer	Transformation Hub

Understanding the Pods that Do Not Have Labels

The Platform includes several pods that are not associated with a deployed capability and thus do not require a label. The installation process automatically creates these pods.

Pod	Description	Namespace
autopass-lm	Manages the Autopass service, which tracks license keys.	arcsight-installer
idm	Manages user authentication and authorization for the CDF Management Portal.	core
itom-pg-backup		arcsight-installer
nginx-ingress-controller	Provides the proxy web server that end-users need to connect to the deployed capabilities. By default, server uses HTTPS and port 443. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer
suite-reconf-pod-arcsight-intaller	Manages the Reconfiguration features in the CDF Management Portal.	arcsight-installer

Specifying Kafka Scheduler Options

Type `./kafka_scheduler <Option_Name>`.

Option Name	Description
update	Updates the scheduler
start	Starts the scheduler and begins copying data from all registered Kafka brokers
stop	Stops the scheduler and ends copying data from all registered Kafka brokers
delete	Deletes all registered Kafka instances from the scheduler
status	Prints the following information and log status for a running or stopped scheduler: • Current Kafka cluster assigned to the scheduler • Name and database host where the active scheduler is running • Name, database host, and process ID of every running scheduler (active or backup)
events	Prints event copy progress for the scheduler
messages	Prints scheduler messages

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Administrator's Guide for ArcSight Platform 20.11 (ArcSight Platform [[[Undefined variable _HPc_Basic_Variables._HP_Product_Version]]])

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to arcsight_doc@microfocus.com.

We appreciate your feedback!