Micro Focus Security ArcSight Recon

Software Version: 1.0

Administrator's Guide

Document Release Date: July, 2020
Software Release Date: July, 2020

Legal Notices

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

https://www.microfocus.com

Copyright Notice

Trademark Notices

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:

ArcSight Product Documentation on the Micro Focus Security Community

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

Contents

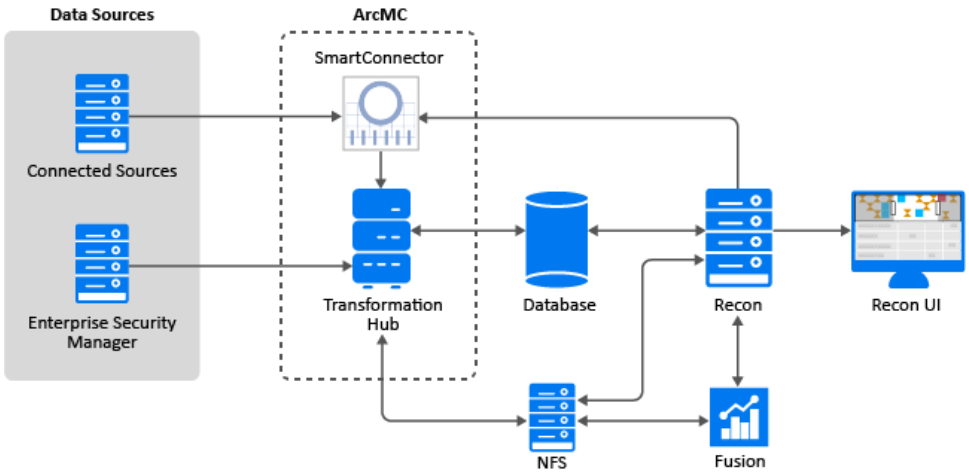
Chapter 1: Introduction

Micro Focus provides a platform that enables you to deploy a combination of security, user, and entity solutions into a single Container Deployment Foundation (CDF) environment. The platform's browser-based interface gives users fast access to the ArcSight suite of products that you have deployed. A common layer called Fusion provides the core services for this CDF environment, including the Dashboard, user management, and single sign-on configuration. The Dashboard enables users 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, such as:

- Performing deep-dive investigations with ArcSight Recon
- Real-time event monitoring and correlation with data from ArcSight Enterprise Security Manager (ESM)
- Analyzing end-user behavior with ArcSight Interset

Arcsight Recon Architecture

ArcSight Recon is a high-capacity data management and analysis engine that enables you to search, analyze, and visualize machine-generated data gathered from web sites, applications, sensors, and devices that comprise 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.



Understanding Recon Components

The following information describes the components incorporated into the Recon environment.

Database

Stores all collected events, provides event search and analysis capabilities.

Fusion

Provides user management, single sign-on, dashboard, high-capacity data management, search engine, and other core services that other capabilities in this suite integrate with to provide a unified solution experience.

Transformation Hub

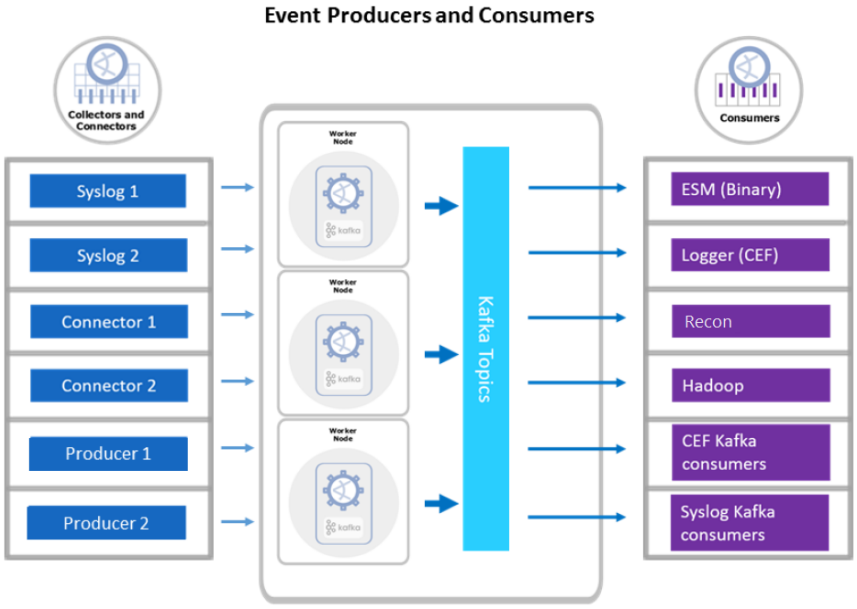
Transformation Hub is the high-performance message bus for ArcSight security, network, flows, application, and other events. It can queue, transform, and route security events to other ArcSight or third party software. This Kafka-based platform allows ArcSight components like Logger, ESM, and Recon to receive the event stream, while smoothing event spikes, and functioning as an extended cache.

Transformation Hub ingests, enriches, normalizes, and then routes Open Data Platform data from data producers to connections between existing data lakes, Fusion 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.

The latest releases of ArcSight Recon are integrated with the Transformation Hub for raw events, as well as integrated with ESM to receive alerts and start the investigation process.

ArcSight ESM receives binary event data for dashboarding and further correlation.

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 a Fusion streaming plug-in framework, supporting automated machine learning and artificial intelligence engines for data source onboarding, event enrichment, and entities and actors detection and attribution.



Security Open Data Platform (SODP)

SODP centralizes management, monitoring and configuration of the entire data-centric ecosystem using an open architecture. It is configured and monitored through the ArcSight Management Center (ArcMC) user interface. SODP comprises the following ArcSight products:

- Transformation Hub (TH)
- Management Center (ArcMC)
- Smart Connectors (SC)

SmartConnectors

SmartConnectors serve to collect, parse, normalize and categorize log data. Connectors are available for forwarding events between and from Micro Focus ArcSight systems like Transformation Hub and ESM, enabling the creation of multi-tier monitoring and logging architectures for large organizations and for Managed Service Providers.

The connector framework on which all SmartConnectors are built offers advanced features that ensures the reliability, completeness, and security of log collection, as well as optimization of network usage. Those features include: throttling, bandwidth management, caching, state persistence, filtering, encryption and event enrichment. The granular normalization of log data allows for the deterministic correlation that detects the latest threats including Advanced Persistent Threats and prepares data to be fed into machine learning models.

SmartConnector technology supports over 400 different device types, leveraging ArcSight's industry-standard Common Event Format (CEF) for both Micro Focus and certified device vendors. This partner ecosystem keeps growing not only with the number of supported devices but also with the level of native adoption of CEF from device vendors.

Management Center (ArcMC)

ArcMC is a central administrative user interface for managing SODP. This management console administers SODP infrastructure, including:

- User management
- Configuration management
- Backup, update and health monitoring to connectors and storage instances

ArcMC's Topology view shows administrators event flow through the entire environment, including a specific focus on monitoring endpoint device log delivery.

Chapter 2: Planning The Installation

This section provides the necessary information to plan your Recon installation.

Implementation Checklist

Use the following checklist to install and configure Recon. You should perform the tasks in the listed order.

Step	Task	See
1	Decide deployment type and configure server accordingly	"Deployment Options" below "Installation Options" on page 9 "Installing the Database" on page 13 "Installing Recon" on page 11
2	Ensure server components meet the specified requirements	Technical Requirements for ArcSight Recon
3	Decide the security mode	"Secure Communication Between Micro Focus Components" on the next page
4	Install Recon	"Installing Recon by Using Scripts" on page 11 "Installing Recon Manually" on page 12
5	Post Installation Configuration	"Post Installation Configuration" on page 40
6	Verify the installation	"Verifying the Installation" on page 42

Deployment Options

You can choose to deploy in a single-node or multi-node environment, depending on your anticipated workload and whether you need high availability.

Single-node Deployment

In a single-node deployment, you deploy all of the Recon components on a single node. This method of deployment is suitable only for small workloads and where you do not need high availability.

Multi-node Deployment

Multiple master nodes, a minimum of 3, will provide the high availability for the cluster management. Multiple worker nodes, a minimum of 3, will provide high availability of the worker node cluster, handle large workloads and, perform load balancing across worker nodes. Master nodes can only be added during the installation. Worker nodes can be added after the installation. Therefore, plan your deployment before you start the installation process.

Secure Communication Between Micro Focus Components

Determine which security mode you want for communication between infrastructure components. The security mode of connected producers and consumers must be the same across all components. Set up the other Micro Focus components with the security mode you intend to use before connecting them.

Note: The secure communication described here 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.

Changing the security mode after the deployment will require system downtime. If you do need to change the security mode after deployment, refer to the Administrator's Guide for the affected component.

The following table lists Micro Focus products, preparations needed for secure communication with components, ports and security modes, and documentation for more information on the product.

Note: Product documentation is available for download from the Micro Focus software community.

Product	Preparations needed...	Ports	Supported security modes	More information
Management Center (ArcMC) version 2.9.4 or later		443, 32080	• TLS • FIPS • Client Authentication	ArcMC Administrator's Guide
SmartConnectors and Collectors	SmartConnectors and ArcMC onboard connectors can be installed and running prior to installing Transformation Hub, or installed after the Transformation Hub has been deployed. • FIPS mode setup is not supported between SmartConnector v7.5 and Transformation Hub. Only TLS and Client Authentication are supported. • FIPS mode ***is*** supported between Connectors v7.6 and above and Transformation Hub.	9093	• TLS • FIPS (SC 7.6+ only) • Client Authentication	SmartConnector User Guide, ArcMC Administrator's Guide
ArcSight ESM	ESM can be installed and running prior to installing Transformation Hub. Note that changing ESM from FIPS to TLS mode (or vice versa) requires a redeployment of ESM. Refer to the ESM documentation for more information.	9093	• TLS • FIPS • Client Authentication	ESM Administrator's Guide
ArcSight Logger	Logger can be installed and run prior to installing Transformation Hub.	9093	• TLS • FIPS • Client Authentication	Logger Administrator's Guide

Download Installation Packages

Download the installation packages for both the CDF Installer, Recon, and Transformation Hub to your Initial Master Node from the Micro Focus Entitlement Portal. After download, validate the digital signature of each file, and then unarchive them.

The complete list of files required for download are:

```
arcsight-installer-metadata-2.3.0.29.tar
cdf-2020.05.00100-2.3.0.7.zip
fusion-1.1.0.29.tar
recon-1.0.0.29.tar
transformationhub-3.3.0.29.tar
```

To access the ArcSight software in the Micro Focus ArcSight Entitlement Portal, use your Micro Focus credentials which will be authenticated before allowing the download.

Download all the installation related files to the directory **`$download_dir`** of the initial master node. The recommended value for the **`download_dir`** is **`/opt/arcsight/download`**.

About the Micro Focus Entitlement Portal

The Micro Focus Entitlement Portal contains ArcSight installation and other product-related materials. This is the only location where you can download the full set of materials needed for Recon installation.

Some downloaded software will be in compressed format, and in addition it will have associated signature files (**`.sig`**) to ensure that the downloaded software is authentic.

Validating Downloaded File Signatures

Micro Focus provides a digital public key that is used to verify the software you downloaded from the Micro Focus software entitlement site is indeed from Micro Focus and has not been tampered with by a third party. Visit the Micro Focus Code Signing site for information and instructions on validating the downloaded software.

To verify the downloaded files are authentic compare each file with its corresponding file signatures (**`.sig`**).

If the set of compressed installation packages does not match their corresponding signatures (**`.sig`**), please contact Micro Focus Customer Support.

Installation Options

Recon provides scripts to easily create a single node server which runs the CDF installer, Transformation Hub, Recon, and the database. If the scripts are not suitable for your use case, the manual installation steps provide more options.

Installation for A Single-Node Deployment Using Scripts

Recon provides scripts that automatically take care of all the prerequisites, software installations, and post-installation configurations. The scripts are applicable for single-node deployments only where high availability is not needed.

The installation scripts expect your environment to be in a specific state. Before deciding to use the installation scripts, review the considerations for installation.

The scripts:

- Disable plain text communication between Transformation Hub (Kafka) and all the components outside the Kubernetes cluster, such as SmartConnector, database, and so on. Therefore, you must configure SSL between Transformation Hub (Kafka) and the components that are outside the Kubernetes cluster.
- Automatically configure SSL for database as database is installed as part of the script.
- Install Recon on the operating system with a default minimum installation.
- Install Recon only on a singled-homed network (a single-homed stub system is one that is connected with a single network link).
- Automatically tune the system for a small workload.
- Configure the database agent to use port 5438 instead of the default port 5444.
- Register a service with the operating system to automatically start the database Kafka scheduler to collect event data.
- Install the cluster with a single master node and single worker node running on the same system. You can add worker nodes after the installation to scale and enable worker high availability.
- If you use the scripts, you cannot configure high availability for the master node. If you want high availability for the master node, we recommend that you use the manual installation process.
- Disable the option to authorize Micro Focus to collect suite usage data.
- Create NFS shares on the system used by the containers in the cluster. They configure the firewall to disable remote access to this NFS server. If you plan to add nodes, remote access to the NFS server in the firewall must be enabled.

- Use the default path of:
  - Kubernetes home: /opt/arcsight/Kubernetes
  - NFS shared directories: /opt/NFS_Volume
  - Database installer directory: /opt/arcsight-database
  - Database directory: /opt/vertica

Manual Installation for Multiple-Node Deployment

Multiple node deployment can only be performed manually.

For information about installing Recon manually, see "Installing Recon Manually" on page 12

Chapter 3: Installing Recon

This section provides information about installing and configuring Recon

Installing Recon by Using Scripts

You can use the installation scripts in single-node deployments for end-to-end installation starting from configuring prerequisites to completing post-installation configurations.

- Prerequisites
- Understanding the Installation Scripts
- Using Scripts in Single Node Deployments

Prerequisites

Ensure the system requirements mentioned in ArcSight Recon 1.0 Technical Requirements are met.

Understanding the Installation Scripts

The installation scripts automatically take care of all the prerequisites, software installations, and post-installation configurations:

Script	Purpose
`./prepare-install-single-node-host.sh`	Installs all the necessary packages and configures the prerequisites.
`./install-single-node.sh`	Installs database, CDF, Transformation Hub, and Recon.
`./install-single-node-post.sh`	Performs post-installation configurations, such as labeling the nodes and scheduling Kafka.

Using the Scripts in Single Node Deployments

Note: Applies only when your deployment does not need high availability.

The installation scripts automatically take care of all the prerequisites, software installations, and post-installation configurations. For deployments with a small workload, the script sets the appropriate configuration settings for database. For medium and large workloads, you must manually adjust the configuration settings after the installation is complete.

To install Recon by using scripts:

1. Log in to the master node as root.
2. Download Recon script file, **`recon-installer-1.0.0.7.tar.gz.`** to **`/opt`**.
3. Change to the directory where you downloaded the Recon installer script file.

   ```
   cd /opt
   tar xfvz recon-installer-1.0.0.7.tar.gz
   cd recon-installer-1.0.0.7
   ```
4. Execute the scripts in the following order:
   a. **`./prepare-install-single-node-host.sh`**
   b. **`./install-single-node.sh`**
   c. **`./install-single-node-post.sh`**
5. (Conditional) If you want to use mutual SSL authentication between Transformation Hub and its clients, perform steps in the Enabling Client Authentication section.

Installing Recon Manually

This chapter provides information about installing Recon and the required software.

Deployment Steps

This section provides information about installing Recon and the required software.

Step	Task	See
1	Installing the Database	"Installing the Database" below
2	Preparing Your Environment for CDF	"Preparing Your Environment for CDF" on page 18
3	Installing the CDF Installer	"Installing the CDF Installer" on page 20
4	Configure and Deploy Transformation Hub	"Configure and Deploy Transformation Hub" on page 30
5	Configure and Deploy Recon	"Configure and Deploy Recon" on page 32
6	Complete Database Setup	"Complete Database Setup" on page 36

Installing the Database

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

Note: Before you install the database, make sure to estimate the storage needed for the incoming EPS (event per second) and event size, and also to evaluate the retention policy accordingly.

Configuring the Database Server

Ensure the system requirements mentioned in Recon 1.0 System Requirements are met.

The server configuration is based on the Hardware Requirements and Tuning Guidelines for Recon.

Note: The configuration settings for the server described in this section is based on the Hardware Requirements and Tuning Guidelines for Recon. If you're not using this type of hardware, adjusting the configuration settings may result in better performance.

To avoid performance issues with large workloads, the Database server should be a dedicated server.

Note: Database data should be backed-up routinely. For more information, please see "Backing Up and Restoring the Database" on page 73.

To configure the Database server:

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

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

2. Add the following parameters to `/etc/sysctl.conf`. You must reboot the server for the changes to take effect.

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`. You must reboot the server for the changes to take effect.

   Note: 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) Vertica 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.con`:

```
* 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"
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
```

For more information, see Firewall Considerations.

Port Availability

Database requires several ports to be open on the local network. It is not recommended to place a firewall between nodes (all nodes should be behind a firewall), but if you must use a firewall between nodes, ensure the following ports are available:

Port	Protocol	Service	Note
22	TCP	sshd	Required by Administration Tools and the Management Console Cluster Installation wizard.
5433	TCP	Database	Database client (vsql, ODBC, JDBC, etc) port.
5434	TCP	Database	Intra- and inter-cluster communication.
5433	UDP	Database	Database spread monitoring.
5438	TCP	Database Management Console	Used as Management Console-to-node and node-to-node (agent) communication port. This port replaced 5444 in the Single node installation.
5444	TCP	Database Management Console	MC-to-node and node-to-node (agent) communications port. See Changing MC or Agent Ports.
5450	TCP	Database Management Console	Port used to connect to MC from a web browser and allows communication from nodes to the MC application/web server.
4803	TCP	Spread	Client connections.
4803	UDP	Spread	Daemon to daemon connections.
4804	UDP	Spread	Daemon to daemon connections.
6543	UDP	Spread	Monitor to daemon connection.

7. Set SELinux to permissive mode:

   ```
   In /etc/selinux/config
   SELINUX=permissive
   ```

   For more information, see SELinux Configuration.

8. Configure the BIOS for maximum performance:

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

9. Reboot the system, and then use the `ulimit -a` command to verify that the limits were increased.

Enabling Password-less Communication

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

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

To configure password-less 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.

   The operation is successful when the system displays the following message:

   ```
   Number of key(s) added: 1
   ```
4. 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
   ```

Determine FIPS Configuration

Follow the steps in "Setting FIPS on Database Server " on page 124to enable or disable FIPS.

To Install Database

After you configured the Database server and enabled password-less SSH access, install the database.

1. On the Database cluster node1 server, create a folder for the Recon database installer script, for example `/opt/db-installer`.

   ```
   mkdir /opt/db-installer
   ```

   Note: `/opt/db-installer` should not be under `/root` and/or `/opt/vertica`.
2. From the "Download Installation Packages" on page 7 section, copy the database bits, `db-installer_3.2.0-4.tar.gz`, to `/opt/db-installer`
3. Extract the `.tar` file:

   ```
   cd /opt/db-installer
   tar xvfz db_installer_3.2.0-4.tar.gz
   ```
4. Edit the `config/db_user.properties` file. The `hosts` property is required.

Property	Description
`hosts`	A comma separated list of the Recon 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.).
`db_retention_day`	Used for the data retention policy.

5. Install Database:

   ```
   ./db_installer install
   ```

   When prompted, create the database administrator user, app admin user, and the Recon search user.

   Database now supports multiple users:

   - **Database administrator:** Credentials required to access the database host to perform database related operations, i.e. setup, configuration, and debugging.
   - **App admin user:** A regular user with granted permissions (db, schema, resource pool). Credentials required when configuring Database from the CDF Management Portal for Recon search engine.
   - **Search user:** A user designated for search operations. Credentials required when configuring Database from the CDF Management Portal for Recon search engine.
   - **Ingest user:** Should not be used or changed, this user is internally used for Database-scheduler, i.e. ingestion.

   For a list of options that you can specify when installing Database, see ./db_installer Options.

6. Database cluster status should be monitored constantly, for more information, please see "Monitoring the Database " on page 87
   - **Database nodes status:** Ensures all nodes are up
   - **Database nodes storage status:** Ensures storage is sufficient

Preparing Your Environment for CDF

The procedures in this section enable you to configure your environment for a successful installation of CDF.

Before proceeding with the installation process described in this document, it is assumed that you have already planned and provisioned your network, storage and the cluster of host systems based on requirements described in the CDF Planning Guide requirements. ***You must plan and configure set up a valid environment for deployment, as described in the CDF Planning Guide, before deploying Transformation Hub and Recon.***

Once the installation packages have been downloaded, validated and uncompressed in the download folder, you are ready to configure and install the CDF Installer.

Note: You can install the CDF Installer as a root user, or, optionally, as a `sudo` user. However, if you choose to install as a `sudo` user, you must first configure installation permissions from the root user. For more information on providing permissions for the `sudo` user, see Appendix B of the CDF Planning Guide.

To configure the Recon Server

1. Ensure the system requirements mentioned in Recon 1.0 System Requirements are met.
2. Provision the server with at least 10GB of free space in /root and /tmp.
3. Add packages:

```
yum install -y conntrack-tools httpd-tools libtool-ltdl java-1.8.0-openjdk
wget socat container-selinux
```

4. Turn on the firewall:

```
systemctl start firewalld

systemctl enable firewalld
```

5. Turn off swap:

```
swapoff -a

vi /etc/fstab

#comment out swap
```

6. Increase the process limit (does not apply for CentOS/RHEL 8.1), add the following to `/etc/security/limits.d/20-nproc.con`:

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

7. Change /etc/hosts

```
vi /etc/hosts

#comment out ::1
```

8. Setup bridge

```
echo "net.bridge.bridge-nf-call-ip6tables = 1" >> /etc/sysctl.conf

echo "net.bridge.bridge-nf-call-iptables = 1" >> /etc/sysctl.conf

echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf

echo 'modprobe br_netfilter && sysctl -p' >> /etc/rc.d/rc.local

chmod +x /etc/rc.d/rc.local
```

Note: If the net.ipv4.tcp_tw_recycle test fails, check the /etc/sysctl.conf file. Change net.ipv4.tcp_tw_recycle=1 to net.ipv4.tcp_tw_recycle=0; then run sysctl -p

9. Add the required proxy information into `/root/.bashrc`, on all nodes, for example:

```
export http_proxy=http://web-proxy.abc.com:8080

export https_proxy=http://web-proxy.abc.com:8080

export HTTP_PROXY=http://web-proxy.abc.com:8080

export HTTPS_PROXY=http://web-proxy.abc.com:8080

L1="localhost,127.0.0.1" #localhost

V1="192.168.1.254,S192-168-1-254.abc.com" #Virtual host, if setup 3 master node

M1="192.168.2.2,S192-168-2-2.arcsight.com" #Master node1

M2="192.168.2.3,S192-168-2-3.arcsight.com" #Master node2

M3="192.168.2.4,S192-168-2-4.arcsight.com" #Master node3

W1="192.168.2.5,S192-168-2-5.arcsight.com" #Worker node1

W2="192.168.2.6,S192-168-2-6.arcsight.com" #Worker node2

W3="192.168.2.7,S192-168-2-7.arcsight.com" #Worker node3

export no_proxy="${L1},${V1},${M1},${M2},${M3},${W1},${W2},${W3}"

export NO_PROXY="${L1},${V1},${M1},${M2},${M3},${W1},${W2},${W3}"
```

10. Reboot the server

Installing the CDF Installer

1. Login to the initial master node, which will be the master node1, as root.

```
mkdir $download_dir
```

2. Download files listed under "Download Installation Packages" on page 7 and copy to `$download_dir`
3. Change the directory to $download_dir and unzip the CDF Installer

```
cd $download_dir

unzip cdf-2020.05.00100-2.3.0.7.zip
```

4. Install the CDF Installer on the Initial Master Node with the following commands.

Note: For NFS parameter definitions, refer to the CDF Planning Guide section "Configure an NFS Server environment".

Note: If the NFS server directories setup match the details described in the following table, **Auto-fill** feature will work during the Kubernetes cluster configuration period.

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

```
cd  $download_dir/cdf-2020.05.00100-2.3.0.7
```

```
./install -m {path_to_a_metadata_file} --k8s-home {path_to_installation_
directory} --docker-http-proxy {your_docker_http_proxy_value} --docker-https-
proxy {your_docker_https_proxy_value} --docker-no-proxy {your_docker_no_
proxy_value} --nfs-server {your_nfs_server_FQDN or IP Address} --nfs-folder
{itom_volume_folder} --ha-virtual-ip {your_HA_ip}
```

You will be prompted for the corresponding password for the default admin user (admin), which will inherently meet your password strength requirements. Alternatively, users can include the optional `--password` parameter to supply the password in the installation command.

Example:

```
cd $download_dir/cdf-2020.05.00100-2.3.0.7
```

```
./install -m $download_dir/arcsight-installer-metadata-2.3.0.29.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.infra.net,infra.net,192.168.10.10" --nfs-server pueas-
vmenv-nfs.swinfra.net --nfs-folder /opt/nfs/volumes/itom/itom_vol --ha-
virtual-ip 192.168.1.2542
```

You may need to configure some additional parameters, depending on your organization's OS, network, and storage configurations.

Note: For a description of valid CDF Installer command line arguments, see Installer CLI Commands.

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

Deploying Recon

This section provides information about using the CDF Management Portal to deploy Recon.

- Configure and Deploy the Kubernetes Cluster
- Uploading Images
- Deploy Node Infrastructure and Services
- Preparation Complete
- "Configure and Deploy Transformation Hub" on page 30
- Configure and Deploy Recon
- Predeployment Configuration Completion
- Labeling Worker Nodes
- Updating Transformation Hub Information
- Complete Database Setup

Configure and Deploy the Kubernetes Cluster

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

1. Browse to the virtual IP (if you have a three master node cluster) at `https://{virtual_FQDN}:3000`, or to the Initial Master Node at `https://{master_FQDN}:3000.` 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.)
2. On the **Security Risk and Governance - Container Installer** page, choose the CDF base product metadata version 2.3.0.29. Then, click **Next**.
3. On the **End User License Agreement** page, review the EULA and select the ***'I agree... and I authorize...'*** checkbox. You may optionally choose to have suite utilization information passed to Micro Focus. Then, click **Next**.
4. On the **Capabilities** page, choose the capabilities and/or products to be installed. Select Transformation Hub, Fusion, and Arcsight Recon, then, click **Next**.
5. On the **Database** page, make sure the **PostgreSQL High Availability** box is *deselected*.
6. Select **Out-of-the-box PostgreSQL.**
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 with 4 cores, 16 GB memory, 50 GB disk)
- **Medium Cluster:** Minimum of 1 Worker Node deployed (each node with 8 cores, 32 GB memory, 100 GB disk)
- **Large Cluster:** Minimum of 3 Worker Nodes deployed (each node with 16 cores, 64 GB memory, 256 GB disk)

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

Additional Worker Nodes, with each running on their own host system, can be configured in subsequent steps.

Select your appropriate deployment size, and then click **Next**.

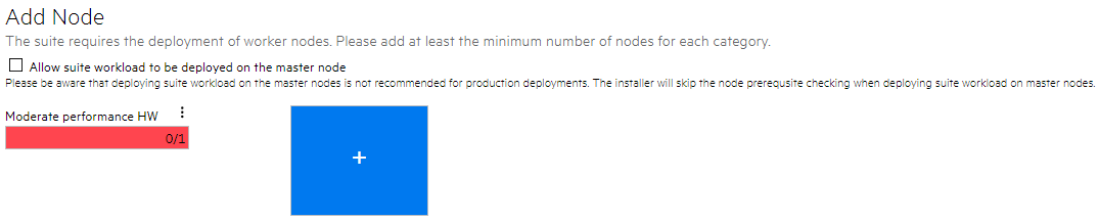
8. On the **Connection** page, an external hostname is automatically populated. This is resolved from the Virtual IP (VIP) specified earlier during the install of CDF (`--ha-virtual-ip parameter`), or the Master Node hostname if the `--ha-virtual-ip` parameter was not specified during CDF installation. Confirm the VIP is correct and then click **Next**.
9. 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**.
10. The installer prompts to add a number of 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 or IP address 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** deselected in order to view any warnings displayed. You may then evaluate whether to ignore or rectify any warnings, clear the warning dialog, and then click Save again with the box selected to avoid stopping.
- **User Name:** User credential for login to the Node.
- **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 and then upload a private key file when connecting the node with a private key file.
- **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, as described in the ***CDF Planning Guide***.

- **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.

11. 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.



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 added. This progress will continue until the necessary count is met so the bar will turn from red to green, meaning you have reached the minimum number of Worker Nodes, as shown selected in Step 7 above. You may add more Nodes than the minimum number.

Note: 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).

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 minumum requirements.)
- **Host:** FQDN (only) of Node you are adding.

Warning: 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 may wish to 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:** User credential to login to the Node.

- **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**.

12. On the **File Storage** page, configure your NFS volumes.

(For NFS parameter definitions, refer to the CDF Planning Guide 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**.

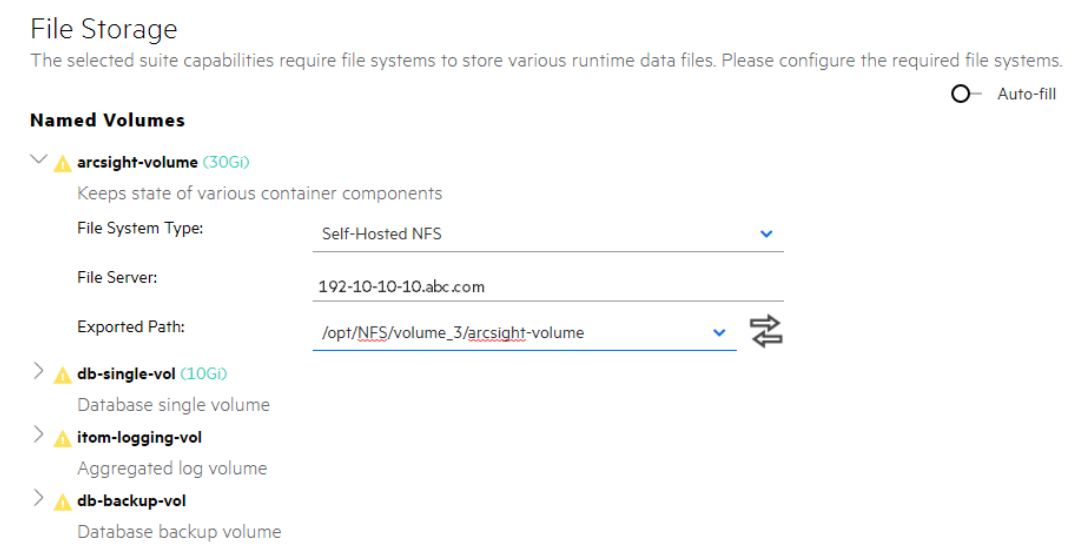
Note: All volumes must validate successfully to continue with the installation.

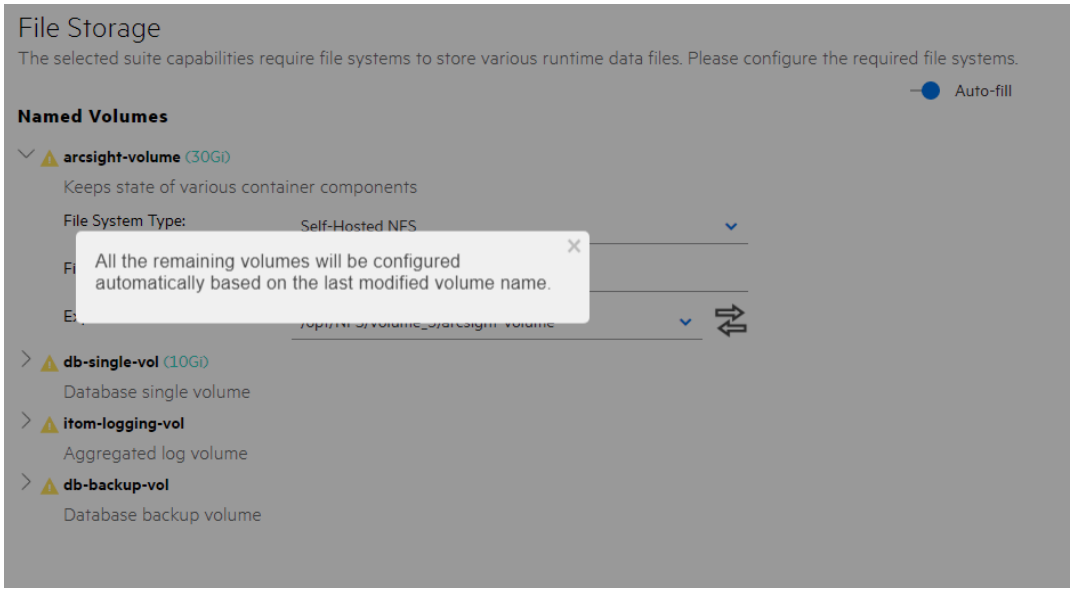
Note: If the NFS server is setup as described in the table below, the **Auto-fill** feature can be applied. Otherwise, each value would need to be filled out individually.

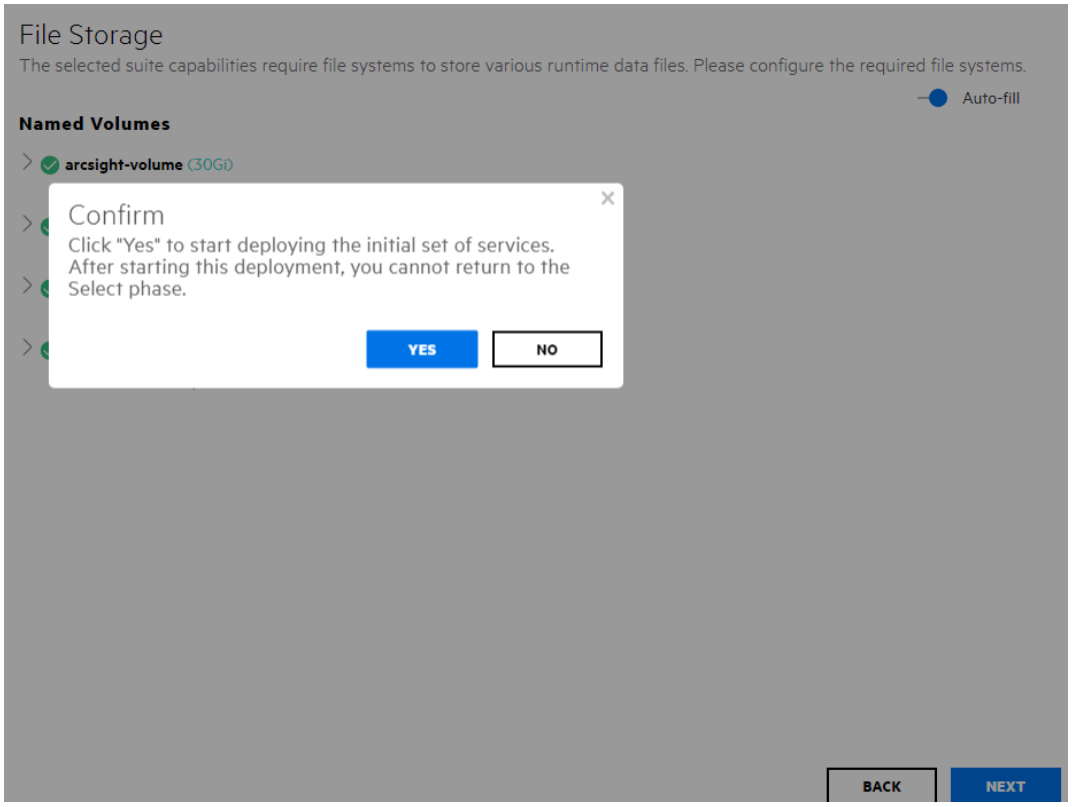
Note: A *Self-hosted NFS* refers to the external NFS that you prepared during the NFS server environment configuration, as outlined in the CDF Planning Guide. Always choose this value for **File System Type.**

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

The pictures below display the Autofill process:


File Storage
The selected suite capabilities require file systems to store various runtime data files. Please configure the required file systems.
Auto-fill
Named Volumes
arcsight-volume (30Gi)
Keeps state of various container components
File System Type:
Self-Hosted NFS
File Server:
192-10-10-10.abc.com
Exported Path:
/opt/NFS/volume_3/arcsight-volume
db-single-vol (10Gi)
Database single volume
itom-logging-vol
Aggregated log volume
db-backup-vol
Database backup volume


File Storage
The selected suite capabilities require file systems to store various runtime data files. Please configure the required file systems.
Auto-fill
Named Volumes
arcsight-volume (30Gi)
Keeps state of various container components
File System Type:
Self-Hosted NFS
All the remaining volumes will be configured automatically based on the last modified volume name.
db-single-vol (10Gi)
Database single volume
itom-logging-vol
Aggregated log volume
db-backup-vol
Database backup volume



13. Click **Yes**.

> **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.

14. On the **Confirm** dialog, click **Yes** to start deploying Master and Worker Nodes.

Download Transformation Hub, Recon and Fusion Images to the Local Docker Registry

From the "Download Installation Packages" on page 7 section, copy the images to `$download_dir`.

On the **Download Images** page, click **Next** to skip this step. No files require download at this point.

Uploading Images

The **Check Image Availability** page lists the images which have currently been loaded into the local Docker Registry from the originally-downloaded set of images. For a first install, it is expected that no images have already been loaded yet. You will upload the images at this step.

To upload the images to the local Docker Registry:

1. Log on to the Initial Master Node in a terminal session as the root or sudo user.
2. Run the following commands to upload the core images to the Local Docker Registry:

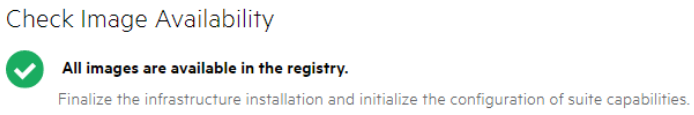
```
cd $k8s-home/scripts
./uploadimages.sh -u registry-admin -F $download_dir/transformationhub-3.3.0.29.tar
./uploadimages.sh -u registry-admin -F $download_dir/recon-1.0.0.29.tar
./uploadimages.sh -u registry-admin -F $download_dir/fusion-1.1.0.29.tar
```

Note: Prior running the image upload process by script, you will be prompted for the administrator password previously specified in the topic "Installing the CDF Installer" on page 20.

3. Wait until all images are uploaded successfully.
4. Go back to the Kubernetes configuration UI to continue.

Verify Prerequisite and Installation Images

The pre-deployment validation process will verify that all environment prerequisites have been met prior to installing the Transformation Hub.



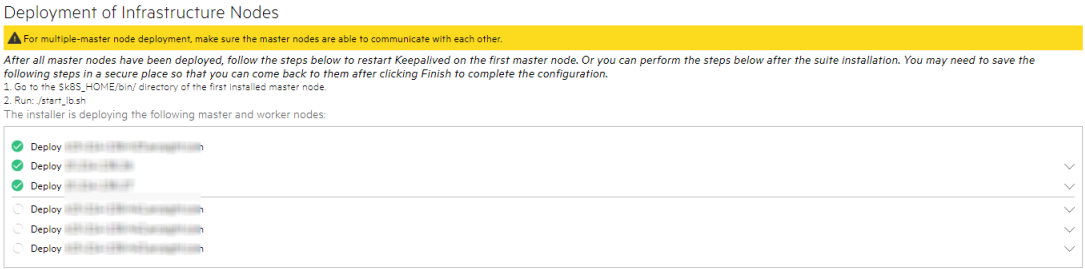
To verify that all images have been uploaded, return to the CDF Management Portal's Check Availability page and click **Check Image Availability Again**. All required component uploads are complete when the message displayed is: ***All images are available in the registry.***

Once verified, click **Next**.

Deploy 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.



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. Should any node show a red icon, then this process may have timed out. If this occurs, click the drop-down arrow to view the logs and rectify any issues. Then click the **Retry** icon to retry the deployment for that node.

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

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:

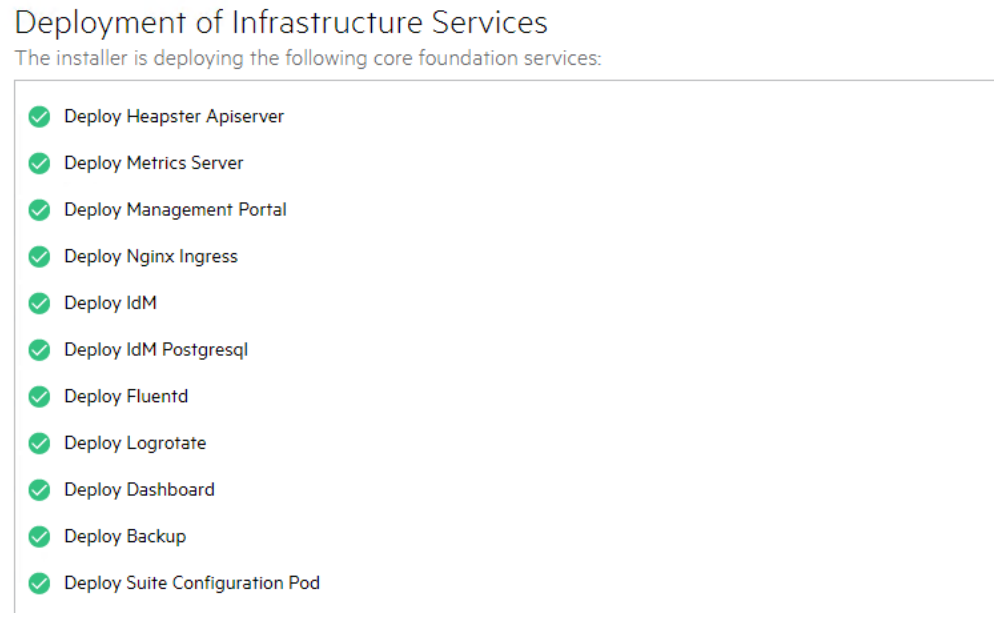
  ```
  tail - <logfilename>
  ```
  - After installation has finished, the logs are copied to `$k8s-home/log/scripts/install`
- You can watch the status of deployment pods with the command:

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

Note: 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 progress.



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.

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

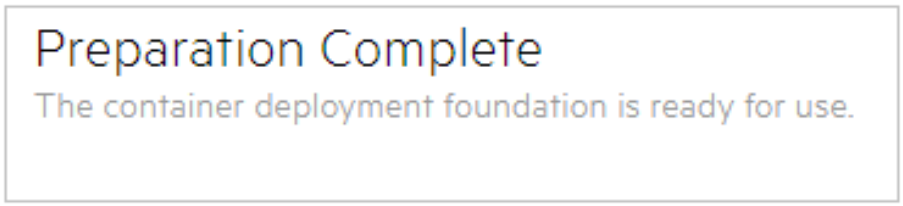
```
watch kubectl get pods --all-namespaces
```

Note: 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 'Bad Gateway' error. Allow more time for the Web UI to start before retrying your login.

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

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.



Click **Next** to configure the products and components of the deployment.

Configure and Deploy Transformation Hub

The Transformation Hub is now ready to be configured. The Transformation Hub Pre-Deployment Configuration page is displayed to configure the products and capabilities chosen at the start of the installation process.

The pre-deployment configuration page allows tuning of the initial installation properties. Click the **Transformation Hub** tab and modify the configuration properties as required, based on the size of your cluster and its throughput requirements. Refer to the Deployment Sizing Calculator spreadsheet for guidance on setting some of these properties. Hover over any value to see a detailed description associated with the configuration property.

Worker Node Properties: You must adjust several of these properties with the number of Worker Nodes installed earlier in this installation process.

Input the following values into the Worker Nodes.

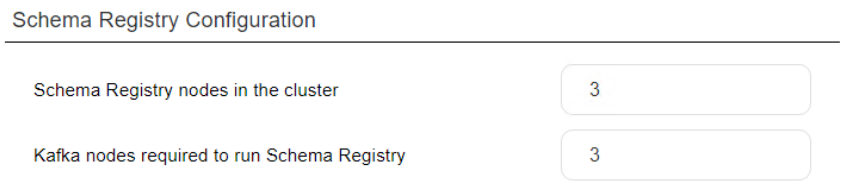
- # of Kafka broker nodes in the Kafka cluster
  - Input the number of worker nodes which will run kafka
  - The number is used to calculate topic partition size

- # of Zookeeper nodes in the Zookeeper cluster
  - Input the number of worker nodes which will run Zookeeper
- # of replicas assigned to each Kafka Topic
  - This must be set to 1 for a Single Worker deployment
- # of message replicas for the __consumer_offsets Topic
  - This must be set to 1 for a Single Worker deployment

Note: Do not change **# of partitions assigned to each kafka topic.**
# of partitions= 24* Number of database.
# of partitions must be changed after deployment has been successfully completed. For more information, please see "Post Installation Configuration" on page 40

Note: It is highly likely the following configuration properties should also be adjusted from their default values. Note that proper log sizes are critical. Should logs run out of space, messages (events) will be dropped and are not recoverable.

- Kafka log retention size per partition for database Avro Topic
  - Input the calculated th-arcsight-avro topic partition size
  - This value is exclusive for database Avro Topic.
- Kafka log retention size per partition per topic
  - Input the calculated th-def topic partition size
- Hours to keep Kafka logs
  - Input the hours used for calculating th-def topic partition size



- Schema Registry nodes in the cluster
  - Input the number of worker nodes which will run Schema Registry
  - This must be set to 1 for a Single Worker deployment
- Kafka nodes required to run Schema Registry
  - Input the number of kafka nodes which will run Schema Registry.
  - This must be set to 1 for a Single Worker deployment

ArcMC Properties: For managing your cluster with ArcMC, you can add your Management Center FQDN: {port}. Note that this can only be configured on the post-deployment configuration page.

After updating configuration property values, click **Next** to deploy Transformation Hub. After a few minutes, the CDF Management Portal URL will be displayed. Select this URL to finish Transformation Hub deployment.

Security Mode Configuration

Prior to deployment, you should choose and configure a security mode that Transformation Hub will use to connect.

By default, plain-text (or non-TLS) connections are permitted from external producers and consumers (such as connectors, ESM, and Logger), to maximize performance.

For higher security you can disable plain-text connections.

The following table shows the effect of each security mode configuration setting on communication over the given port.

Security Mode Configuration Setting	Value	Connect to 9092 (Plain Text)?	Connect to 9093 (TLS)?
Allow Plain Text Connections	true	yes	yes
Allow Plain Text Connections	false	no	yes
Client Authentication	true	N/A	yes
Client Authentication	false	N/A	yes
FIPS	true	N/A	yes
FIPS	false	N/A	yes

- 9093 is the endpoint used for TLS, and is always enabled.
- 9092 is the endpoint used for plain text, and is enabled by the Allow plain text connections configuration setting, which is new in Transformation Hub 3.2. This setting has no effect on the FIPS and Client Authentication settings.

Note: Configure these settings before deployment. Changing them after deployment will result in cluster downtime.

Configure and Deploy Recon

Recon is now ready to be configured. Recon Pre-Deployment Configuration page is displayed to configure the products and capabilities chosen at the start of the installation process.

The pre-deployment configuration page allows tuning of the initial installation properties.

In order to Configure and Deploy Recon, perform the following procedures:

1. Setup of Database Connection (Mandatory step)
2. Setup SMTP Server (Optional)

Setting Up Database Connection

Click the **FUSION** tab and modify the configuration properties as required.

In order to setup the set up database connection, scroll down to **Database Configuration**

Under Database Configuration, provide the following information to update the connection parameters:

- **Database host:** You can specify any database node IP address, but only specify one address (**Use IP address only**).
- **Database Application Admin User Name:** The application admin user name that you defined when you installed database.
- **Database Application Admin User password:** The application admin user password that you created when you installed database.
- **Search user name:** The search user name created when you installed the database.
- **Search user password:** The search user password created when you installed the database.

Click **SAVE.**

Configuration Completion

This page will be displayed, once pre-deployment has been successfully completed.

Pod status can be monitored on this page after the worker nodes have been labeled, and images have deployed.

To Continue Setup from Management Portal

1. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.
2. Input the following information, and then click **LOG IN**

- **User Name:** admin
- **Password:** Password provided during installation

3. Continue to "Label Worker Nodes" below section.

Label Worker Nodes

Labeling a node tells Kubernetes what types of workloads can run on a specific host system. Labeling is a means for identifying application processing and qualifying the application as a candidate to run on a specific host system.

Pods will remain in a ***Pending*** state awaiting the labeling process to be completed. Once labeling is completed, Kubernetes will immediately schedule and start the label-dependent containers on the labeled nodes. (Note that starting of services may take 15 minutes or more to complete.)

To label your worker nodes:

1. Login to Management Portal by clicking the link on the **Deployment status** (Configuration complete) page or browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.

- **User Name:** admin
- **Password:** Password

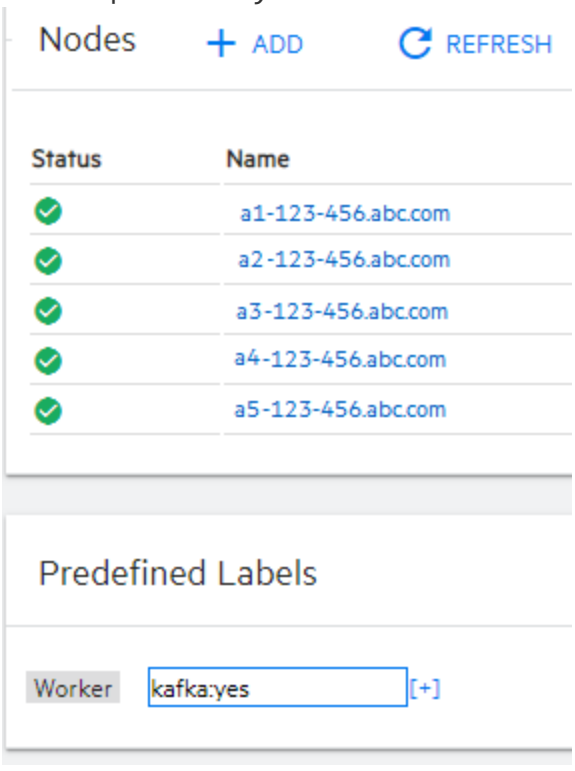
2. Go to **CLUSTER > Nodes.**
3. In **PredefinedLabels** enter the label `zk:yes` (case-sensitive) and then click the **+** icon. This will add the `zk:yes` label to the list of predefined labels you can use to label nodes. The label list will be shown to the left of the text box.
4. Repeat Step 3 for each of the following labels to add them to the list of predefined labels. Enter the text of the entire label, as shown here, including the `:yes` text. Labels are case-sensitive.

```
fusion:yes

kafka:yes

th-processing:yes

th-platform:yes
```



5. Drag and drop a new label from the **Predefined Labels** area to each of the Worker Nodes, based on your workload sharing configuration. This will apply the selected label to the Node.

Note: Only one worker node can be added for Recon. Recon and Transformation Hub should not reside on the same worker node.

Note: You must click **Refresh** to see any labels that you have already applied to Nodes.

For Kafka (kafka:yes) and ZooKeeper (zk:yes) labels, make sure that the number of the nodes you labeled correspond 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. Add the labels th-processing:yes and th-platform:yes to the same nodes as Kafka.

For the Recon node, drag the `fusion:yes` label to the Recon node. Label only one node for Recon. For large workloads, Recon and Transformation Hub should not reside on the same worker node for performance reasons.

Once the Nodes have been properly labeled, the Transformation Hub services status will change from *Pending* to *Running* state. To monitor the pods through the Kubernetes Dashboard go to **Cluster > Dashboard**.

Updating Transformation Hub Information

Configure CEF-to-Avro Stream Processor Number

To configure the CEF-to-Avro Stream Processor number, follow the steps in Configure CEF-to-Avro Stream Processor Number.

Note: 15 was tested as the appropriate value for 120 partitions on a 3 node TH cluster.

Update 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.

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

Note: 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.

- Run:

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

behind the line

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

add line

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

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

To verify, run: `systemctl status kubelet`

No error should be reported.

Update Transformation Hub Partition Number

Adjust the partition number for th-cef topic and th-arcsight-avro topic, from default (6) to the number you used to calculate the partition size.

Note: # of partitions assigned to each kafka topic = 24 * Number of database nodes.

To update the topic partition number, follow the steps in Updating Topic Partition Number.

Complete Database Setup

Follow the steps below to complete the Database Setup.

1. Login to the database node1 as root:

   ```
   cd /opt/db-installer
   ```

2. Create the schema:

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

3. In order to create the Kafka scheduler, run the below commands:
   - If SSL is disabled:

     ```
     ./sched_ssl_setup --disable-ssl
     ```

   - If SSL is enabled, see "Database SSL Root Certificate Support" on page 144.

4. Create the Kafka scheduler:

   ```
   ./kafka_scheduler create <Transformation_Hub_Node_1_IP>:9092
   ```

Note: Scheduler will obtain the Transformation Hub node information from kafka broker.

For a list of options that you can specify when installing the scheduler, see Kafka Scheduler Options.

5. Check the Database status:

```
./db_installer status
```

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

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

./db_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
delete-schema	Deletes the Recon 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

Kafka Scheduler Options

To specify an option, 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

Option Name	Description
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

Post Manual Installation Configurations

This section provides information about the post-installation configurations you must perform after installing Recon.

Updating Topic Partition Number

Perform the following steps to update the topic partition number from the master node1:

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

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

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

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

   - 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
     ```

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

   - 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
     ```

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

Configure CEF-to-Avro Stream Processor Number

Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.

1. Click **DEPLOYMENT**, and select **Deployments.**

2. Click the **Three Dots** (Browse) on the far right and choose **Reconfigure**, under **Transformation Hub > Stream Processors and Routers**perform the following actions:
   - Provide a value for the amount of c2av pods you need in **# of CEF-to-Avro Stream Processor instances to start**.
   - Click **SAVE**

Post Installation Configuration

This chapter provides information about the post-installation configurations you must perform after installing Recon.

Reminder: Install Your License Key

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.

For details on how to apply a your license key to Transformation Hub, see the Licensing chapter of the Transformation Hub Administrator's Guide.

IMPORTANT: To ensure continuity of functionality and event flow, make sure you apply your product license **before** the evaluation license has expired.

Setup SMTP Server

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 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
- SMTP server administrator email address
- User session timeout in seconds

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.

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

   ```
   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 installed Identity Intelligence by using scripts, remove all rich rules:

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

4. Reload the new firewall configuration:

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

5. Restart the nginx pod to apply the new firewall configuration:

   ```
   SUITE_NAMESPACE=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)

   kubectl delete pod --namespace=$SUITE_NAMESPACE -l app=nginx-ingress-lb
   ```

6. (Conditional) If you want to expose NFS shares to other hosts such as other master and worker node:
   a. Execute the command:

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

   b. Reload the new firewall configuration:

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

   c. Restart the nginx pod to apply the new firewall configuration:

      ```
      SUITE_NAMESPACE=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)

      kubectl delete pod --namespace=$SUITE_NAMESPACE -l app=nginx-ingress-lb
      ```

Configuring Management Center

The Management Center (ArcMC) is the centralized console for managing Micro Focus products.

Connectivity between Transformation Hub and ArcMC is configured in ArcMC when you add Transformation Hub as a managed host into ArcMC.

Configure CEF-to-Avro Stream Processor Number

The CEF-to-Avro Stream Processor number can also be reconfigured after the installation. To do this, follow the steps in "Configure CEF-to-Avro Stream Processor Number" on page 38.

Verifying the Installation

To determine whether the installation is successful, perform the following:

1. Login to recon at **`https://<virtual_FQDN>`**, or at **`https://<master_node1_FQDN>`** after successfully creating the default admin user.

For a more in-depth verification you can perform these additional steps:

1. Check that all pods are either in Running or Completed state:

   From the recon master node1 run: **`kubectl get pods --all-namespaces`**
2. Check whether the **`c2av`** pod number matches the number of pods input in Configure CEF-to-Avro Stream Processor Number from the recon master node1 run:

   `kubectl get pods --all-namespaces | grep c2av | wc -l`
3. From the Kafka manager:
   a. Verify whether the number of topics is 11. For more details, see "Monitoring Transformation Hub's Kafka" on page 68 step 3.
   b. Verify the th-cef topic and th-arcsight-avro topic partition numbers match the number defined in Updating Topic Partition Number.

Chapter 4: Configuring Data Collection

Recon gathers data (events) sent by Connectors through Transformation Hub.

Data Collection Configuration Checklist

To successfully configure data collection, complete the following checklist in the listed order:

Task Number	Task
1.	Install and configure the SmartConnector
2.	Verify whether the data collection configuration is successful

Installing and Configuring the SmartConnector

Recon uses the SmartConnector for collecting events of the data source, generated by Recon and sending the data to Transformation Hub for processing.

This chapter provides information about installing and configuring the SmartConnector in deployments where ArcMC is not being used. For information about installing and configuring the SmartConnector where ArcMC is in use, see the SmartConnector documentation.

Prerequisites

Complete the following prerequisites before you install the SmartConnector:

- Download the SmartConnector installation file to /opt in the Connector host.
- Install the following packages in the node where you plan to install the SmartConnector:

```
yum install libXext libXrender libXtst fontconfig
```

Installing the Smart Connector

To install the SmartConnector:

1. Log in to the Connector host as the root user.
2. Change to the directory (for example, `/opt`) where you downloaded the SmartConnector installation file.
3. Update permissions for the `<SmartConnector_installation>` file:

   Example:

   ```
   chmod 755 ArcSight-<version>-Connector-Linux64.bin
   ```
4. Install the SmartConnector:

   Example:

   ```
   ./ArcSight-<version>-Connector-Linux64.bin
   ```
5. Follow the Smart Connector configuration GUI to complete the Connector's installation.

Creating TrustStore for One-Way SSL with Transformation Hub

If you want to establish one-way SSL authentication between SmartConnector and Transformation Hub, you must first obtain the certificate from Transformation Hub, and then upload this certificate to a TrustStore in the computer where have installed SmartConnector for audit events collection:

1. Retrieve the CDF CA certificate and copy the certificate to the computer where you plan to install the SmartConnector.
2. Upload the certificate to the TrustStore file:

   ```
   /usr/lib/jvm/jre/bin/keytool -import -alias <alias name> -file <cert_file> -storetype JKS -keystore <trust_store_file>
   ```
   Example:

   ```
   /usr/lib/jvm/jre/bin/keytool -import -alias vlab2004 -file /tmp/ca.cer -storetype JKS -keystore VLAB2004.JKS
   ```
3. (Conditional) If you have multiple CA certificates, repeat Step 2 for each CA certificate in the certificate chain.
4. Continue with Configuring the SmartConnector.

Creating TrustStore and KeyStore for Mutual SSL with Transformation Hub

If you have enabled client authentication in Transformation Hub, you must configure mutual SSL authentication between SmartConnector and Transformation Hub.

To configure mutual SSL between Transformation Hub and SmartConnector, perform the following:

1. On the SmartConnector server:
   a. Change to the current directory,

      Linux:

      ```
      cd <install dir>/current
      ```
      Windows:

      ```
      cd <install dir>\current
      ```
   b. Set the environment variables for the static values used by keytool:

      Linux:

      ```
      export CURRENT=<full path to this "current" folder>
      export TH=<th hostname>_<th port>
      export STORES=${CURRENT}/user/agent/stores
      export STORE_PASSWD=<password>
      ```

```
export TH_HOST=<TH master host name>
export CA_CERT=ca.cert.pem
export CERT_CA_TMP=/opt/cert_ca_tmp
```

Windows:

```
set CURRENT=<full path to this "current" folder>
set TH=<th hostname>_<th port>
set STORES=%CURRENT%\user\agent\stores
set STORE_PASSWD=<password>
set TH_HOST=<TH master host name>
set CA_CERT=C:\Temp\ca.cert.pem
set CERT_CA_TMP=\opt\cert_ca_tmp
```

c. (Conditional) Create the stores directory if it does not exist:

Linux:

```
mkdir ${STORES}
```

Windows:

```
mkdir %STORES%
```

d. From a command prompt, change to the installation directory of the keytool utility. The default installation directory is:

Linux:

```
/usr/lib/jvm/jre/bin
```

Windows:

```
c:\usr\lib\jvm\jre\bin
```

e. Create the key pair:

i. Execute the command:

Linux:

```
./keytool -genkeypair -alias ${TH} -keystore
${STORES}/${TH}.keystore.jks -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 375
```

Windows:

```
.\keytool -genkeypair -alias %TH% -keystore
%STORES%\%TH%.keystore.jks -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 375
```

> **Note**: For dname, the FQDN, OU, O, L, ST and C values must be appropriate for your company and location. For example, -dname "CN=ig.mf.com,OU=IG,O=MF,L=Sunnyvale,ST=CA,C=US"

ii. When prompted, enter the password. Note the password as you will need it in a later step.

> **Note**: Ensure that the password is same as the store password you specified in Step 1.b.

iii. When prompted for the key password, press Enter if you want the key password to be same as the keystore password. Save the password. You will need it again in a later step.

f. List the keystore entries and verify that you have minimum one private key:

Linux:

```
./keytool -list -keystore ${STORES}/${TH}.keystore.jks -storepass
${STORE_PASSWD}
```

Windows:

```
.\keytool -list -keystore %STORES%\%TH%.keystore.jks -storepass %STORE_
PASSWD%
```

g. Create a Certificate Signing Request (CSR):

Linux:

```
./keytool -certreq -alias ${TH} -keystore ${STORES}/${TH}.keystore.jks
-file ${STORES}/${TH}-cert-req -storepass ${STORE_PASSWD}
```

Windows:

```
.\keytool -certreq -alias %TH% -keystore%STORES%\%TH%.keystore.jks -
file %STORES%\%TH%-cert-req -storepass %STORE_PASSWD%
```

2. On the Transformation Hub Server:

a. Ensure that the CDF root CA certificate and root CA key used by Transformation Hub are available in /tmp directory with the following names:

```
/tmp/ca.key.pem
/tmp/ca.cert.pem
```

b. Set the environment variables for the static values used by keytool:

```
export CA_CERT_TH=/tmp/ca.cert.pem
export CA_KEY_TH=/tmp/ca.key.pem
export CERT_CA_TMP_TH=/opt/cert_ca_tmp
export TH=<Transformation Hub hostname>_<Transformation Hub port>
```

c. Create a temporary directory on the Transformation Hub master server:

```
mkdir $CERT_CA_TMP_TH
```

3. Copy the `${STORES}/${TH}-cert-req` file from a Linux based SmartConnector server or `%STORES%\%TH%-cert-req` file from a Windows based SmartConnector Server to the `CERT_CA_TMP_TH` directory in the Transformation Hub master server created in Step 2.c.
4. On the Transformation Hub server, create the signed certificate using the openssl utility:

```
/bin/openssl x509 -req -CA ${CA_CERT_TH} -CAkey ${CA_KEY_TH} -in ${CERT_
CA_TMP_TH}/${TH}-cert-req -out ${CERT_CA_TMP_TH}/${TH}-cert-signed -days
366 -CAcreateserial -sha256
```

5. On the SmartConnector server:
   a. Copy the `${TH_CERT_CA_TMP_TH}/${TH}-cert-signed` and `/tmp/ca.cert.pem` certificates from the Transformation Hub server to the `${STORES}` directory on the Linux based SmartConnector server or `%STORES%` directory on the Windows based SmartConnector server.
   b. Import the CDF root CA certificate to the truststore:
      i. Execute the command:

      Linux:

      ```
      ./keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot-
      keystore ${STORES}/${TH}.truststore.jks -storepass ${STORE_PASSWD}
      ```

      Windows:

      ```
      .\keytool -importcert -file %STORES%\%CA_CERT% -alias CARoot-
      keystore %STORES%\%TH%.truststore.jks -storepass %STORE_PASSWD%
      ```

      ii. When prompted, specify a password for the truststore. Note the password as you will need it again in a later step.
      iii. When you are asked to trust the certificate, enter Yes.
   c. Import the CDF root CA certificate to the keystore:
      i. Execute the command:

      Linux:

      ```
      ./keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -
      keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_PASSWD}
      ```

      Windows:

      ```
      .\keytool -importcert -file %STORES%\${CA_CERT} -alias CARoot -
      keystore %STORES%\%TH%.keystore.jks -storepass %STORE_PASSWD%
      ```

      ii. When you are asked to trust the certificate, enter Yes.
   d. Import the signed certificate to the keystore:

      Linux:

      ```
      ./keytool -importcert -file ${STORES}/${TH}-cert-signed -alias ${TH}-
      keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_PASSWD}
      ```

      Windows:

      ```
      \keytool -importcert -file %STORES%\%TH%-cert-signed -alias %TH%-
      keystore %STORES%\%TH%.keystore.jks -storepass %STORE_PASSWD%
      ```

   e. Note the keystore and truststore paths:

      Linux:

```
echo ${STORES}/${TH}.truststore.jks
echo ${STORES}/${TH}.keystore.jks
```

Windows:

```
echo %STORES%\%TH%.truststore.jks
echo %STORES%\%TH%.keystore.jks
```

f. 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.

Linux:

```
rm ${STORES}/${CA_CRT}
rm ${STORES}/ca.key.pem
rm ${STORES}/${TH}-cert-signed
rm ${STORES}/${TH}-cert-req
```

Windows:

```
del %STORES%\ca.cert.pem
del %STORES%\ca.key.pem
del %STORES%\%TH%-cert-signed
del %STORES%\%TH%-cert-req
```

6. On the Transformation Hub server, delete the /tmp folder where the CDF root CA certificate, and root CA key of Transformation Hub are available.

> 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.

7. Continue with Configuring the SmartConnector.

Configuring the Smart Connector

You must perform the following steps for both the instances of SmartConnector: SmartConnector for audit events collection and SmartConnector for event collection.

1. Change to the following directory:

```
<SmartConnector Installation Directory>/current/bin
```

2. Execute the following command:

```
./runagentsetup.sh
```

3. Read through the warning, then enter yes to continue.

4. Enter the corresponding number for ***Transformation Hub*** as the *destination type*.
5. Configure the **destination** parameters:
   a. For *Initial Host:Port(s)*, enter the *FQDN/IP* and port of all Kafka nodes.
      - For *non-SSL/TLS*:

        `<kafka_host_name>:9092`
      - For *SSL/TLS*:

        `<kafka_host_name>:9093`

        Note: Ensure that the FQDNs of Kafka nodes resolve successfully.

   b. Press Enter to accept the default content type.
   c. Press Enter to accept th-cef as the default Topic.
   d. Press Enter to accept the default ESM version.
   e. Press Enter to accept the default Acknowledgment mode as leader.
   f. (Conditional) If you want to configure one-way SSL authentication with Transformation Hub:

      Enter the number corresponding to true for Use SSL/TLS and provide the following information:

      For information on how to create a trustore, please see "Configure a Transformation Hub Destination with Client Authentication in Non-FIPS Mode" on page 106.

      i. **SSL/TLS Trust Store file**: Specify the full path to the truststore file that contains the CDF root CA certificate.
      ii. **SSL/TLS Trust Store password**: Specify the password used to access the truststore file that contains the CDF root CA certificate.

   g. (Conditional) If you want to configure mutual SSL authentication with Transformation Hub:

      Enter the number corresponding to true for Use SSL/TLS and provide the following information:

      i. **SSL/TLS Trust Store file**: Specify the full path to the truststore file that contains the CDF root CA certificate.
      ii. **SSL/TLS Trust Store password**: Specify the password used to access the truststore file that contains the CDF root CA certificate.
      iii. **Use SSL/TLS Authentication**: Select Yes if you want to Transformation Hub to authenticate SmartConnector.

      For information on how to create a keystore, please see "Configure a Transformation Hub Destination with Client Authentication in Non-FIPS Mode" on page 106.

      iv. **SSL/TLS Key Store file**: Specify the full path of the keystore file that contains SSL private key and certificate.

v. **SSL/TLS Key Store pass**: Specify the password to access the keystore file.

vi. **SSL/TLS Key password**: Specify the password to access the private key.

h. Enter yes to confirm the destination parameter values are correct.

i. Continue to complete the Connector configuration.

For more information, see the Configuring Connectors section in the SmartConnector User Guide.

Verifying the SmartConnector Configuration

To verify whether the SmartConnector is configured correctly, you can check the `/<SmartConnector Installation Directory>/current/logs/agentsetup.log` file.

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 such as ArcSight ESM and Interset.

- "Using the Widget SDK" below
- "Considerations for Updating the Widget Store" below

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, located by default in the `/NFS_root/arcsight/fusion/widget-store` directory.
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 other 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/NFS_Volume/arcsight-vol/product/widget-store`. For example, a widget available for multiple applications might be stored in the `/opt/NFS_Volume/arcsight-vol/fusion/widget-store` directory.

- 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.

Chapter 5: Upgrading Recon

This section provides information about upgrading Investigate 3.1.0 to Recon.

Upgrade Steps

Follow the steps listed below in order to ensure a successful upgrade.

Step	Task	See
1.	Remove Investigate and Analytics	"Remove Investigate and Analytics and Stop EPS to Avro Topic" below
2.	Monitor database EPS	"Monitor the Database EPS" on the next page
3.	Delete th-arcsight-avro topic record	"Delete th-arcsight-avro Topic Record" on page 56
4.	Database Upgrade	"Database upgrade" on page 56
5.	Upgrade Suites	"Arcsight Suite Upgrade" on page 58
6.	Reset scheduler owner and recreate scheduler	"Reset Scheduler Owner and Recreate Scheduler" on page 64
7.	Delete old outlier model	"Delete old Outlier Model" on page 64

Remove Investigate and Analytics and Stop EPS to Avro Topic

1. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.

a. Click **DEPLOYMENT**, and select **Deployments.**

b. Click the **Three Dots** (Browse) on the far right and choose **Change**. A new screen will be opened in a separate tab. **> Change**

c. Uncheck the boxes of **Arcsight Investigate** and **Analytics**, click **NEXT** until you return to the Deployment page again.

d. Click the **Three Dots** (Browse) on the far right and choose **Reconfigure**, under **Transformation Hub > Stream Processors and Routers**perform the following actions:

- Note down the value of **# of CEF-to-Avro Stream Processor instances to start**, and then change it to 0.
- Click **SAVE**

2. Use kafka Manager to check `th-arcsight-avro` topic. For details on how to Monitor through Kafka manager, please see "Monitoring Transformation Hub's Kafka" on page 68.
3. Wait for the value of **Produce Message/Sec** to become 0.

Monitor the Database EPS

1. From the database cluster node1, `cd` to `<Vertica installer directory>` and run:

   `./kafka_scheduler events`
2. Check the output of **Event Copy Status for (th-arcsight-avro)** topic.
3. Wait until the **end_reason** field displays `END_OF_STREAM`

Delete th-arcsight-avro Topic Record

You need to delete the topic record for the th-arcsight-avro topic, since the event schema has changed for the current release.

From the Transformation Hub/Investigate cluster master node1 run:

Note: You can copy/paste the commands below to execute them.

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

server=`kubectl get pods --all-namespaces -o wide | grep kafka-0 \

| awk '{print $8}'`

#Confirm if Transformation Hub port 9092 or 9093 is available
#if port 9092 is available then
broker="$server:9092"

#else if port 9093 is available then
broker="$server:9093"

topic="th-arcsight-avro"

echo '{"partitions": [ ' > /tmp/test.json

kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-run-class \
kafka.tools.GetOffsetShell --broker-list "$broker" --topic "$topic" \
|sed -re 's/(.*):(.*):(.*)/{"topic": "\1", "partition": \2, \
"offset": \3},/' >> /tmp/test.json

sed -i '$ s/,$//' /tmp/test.json

echo '], "version":1 }' >> /tmp/test.json

kubectl cp /tmp/test.json $NS/th-kafka-0:/tmp/test.json kubectl exec \
-n $NS th-kafka-0 -- /usr/bin/kafka-delete-records --bootstrap-server \
"$broker" --offset-json-file /tmp/test.json

rm -rf /tmp/test.json
```

Database upgrade

Before performing the upgrade

- Stop all Investigate operations
- Stop scheduler

- Pause outliers scoring
- Backup the database

Note: The upgrade process is irreversible, make sure to backup the database.

Upgrade steps

1. Login to the database cluster node1 as root.
2. Create a folder for the new database installer script:

   ```
   mkdir /opt/3.2
   ```

3. From the "Download Installation Packages" on page 7 section, copy the database bits, `db-installer_3.2.0-4.tar.gz`, to `/opt/3.2.`
4. Access the directory:

   ```
   cd /opt/3.2
   ```

5. Untar `db-installer_3.2.0-4.tar.gz`:

   ```
   tar xvfz db-installer_3.2.0-4.tar.gz
   ```

6. Execute the following commands in order:

Note: This operation will take longer depending on the amount of events in the database. To speed-up the upgrade, if possible, reduce the event retention time period temporarily as described in the "Managing Recon" on page 65 section before performing this operation.

1. `./db_upgrade -c upgrade-utilities #provide username/password for newly added App Admin user`

…

```
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/installer

Please specify a username for [ App Admin ] user: iappadmin

Please specify a password for [ App Admin ] user :

Re-enter password:
```

…

```
********* Start of Database Upgrade to 3.2.0 *********
```

```
Pre Upgrade Check for DB Event_v1.0.0 Schema

DB will be upgraded to Event_v1.0.0 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 3.2.0
******************
```

2. `./db_upgrade -c upgrade-db-rpm`
   Follow the Post-upgrade instructions:

   -Optional: start firewall service

   -Run `/opt/installer/db_installer start-db`

   -Run `/opt/installer/kafka_scheduler delete`

   -Run `/opt/installer/kafka_scheduler create $server-list`

Arcsight Suite Upgrade

The following topics are included in this chapter:

- Upgrade CDF and Upgrade Arcsight suites
- Upgrade CDF includes:
  - Upgrade CDF from 2020.02 to 2020.05
  - Both manual upgrade steps and auto-upgrade steps
- Upgrade Arcsight suites includes:
  - Upgrade Investigate-3.1.0 to Recon-1.0.0
  - Upgrade Analytics-3.1.0 to Fusion-1.0.0
  - Upgrade Transformation Hub from 3.2.0 to 3.3.0

Note: The upgrade steps must be performed in the order displayed below.

Upgrading CDF 2020.02 to 2020.05

Manual Upgrade Process from CDF 2020.02 to 2020.05

Beginning with the master node1, upgrade your CDF infrastructure on every node of the cluster by running the following process ***on each node:***

1. Run: `mkdir /tmp/upgrade-download`
2. From the "Download Installation Packages" on page 7 section, copy the CDF bits, `cdf-2020.05.00100-2.3.0.7.zip` to `/tmp/upgrade-download.`
3. Unzip the upgrade package by running these commands:

   ```
   cd /tmp/upgrade-download
   unzip cdf-2020.05.00100-2.3.0.7.zip
   ```

4. Run the following commands on each node (follow this pattern: master1, master2, master3, to worker1, worker2, worker3, etc.):

   ```
   /tmp/upgrade-download/cdf-2020.05.00100-2.3.0.7/upgrade.sh -i
   ```

5. On the initial master node1, run the following commands to upgrade CDF components:

   ```
   /tmp/upgrade-download/cdf-2020.05.00100-2.3.0.7/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:

   ```
   /tmp/upgrade-download/cdf-2020.05.00100-2.3.0.7/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
   >> 2020.05.00100
   ```

8. Check the status of CDF on each node by running these commands:

   ```
   cd ${K8S_HOME}/bin
   ./kube-status.sh
   ```

Automated Upgrade to CDF 2020.05

The automated upgrade to CDF 2020.05 is run with a single command and requires no interaction until completion of each phase. Typically, each automated upgrade phase takes around 1 hour for a 3x3 cluster.

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 example

Configure Passwordless Communication: You must configure passwordless SSH communication between the UM and all the nodes in the cluster, as follows:

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>`

Download Upgrade File: Next, download the upgrade files for CDF 2020.05 to a download directory (referred to as `<download_directory>`) on the UM.

There are 3 directories involved in the auto-upgrade process:

- An auto-upgrade directory /tmp/autoUpgrade will be auto generated on the UM. It will store the upgrade process steps and logs.
- A backup directory /tmp/CDF_202002_upgrade will be auto generated on every node. (approximate size 1.5 GB)
- 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.

Note: 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.

Auto-upgrade from CDF 2020.02 to CDF 2020.05

Proceed with the automated upgrade, as follows:

1. Run: `mkdir /tmp/upgrade-download`
2. Download `cdf-2020.05.00100-2.3.0.7.zip` to `/tmp/upgrade-download`
3. Unzip the upgrade package by running these commands:

```
cd /tmp/upgrade-download
unzip cdf-2020.05.00100-2.3.0.7.zip
```

4. Run: `./autoUpgrade.sh -d /path/to/workinig_directory -n {any_cluster_node_adress_or_ip}`

Example:

```
./autoUpgrade.sh -d /tmp/upgrade -n pueas-ansi-node1.swinfra.net
```

Remove the auto-upgrade temporary directory from UM

The auto-upgrade temporary directory contains the upgrade steps and logs. If you want to upgrade another cluster from the same UM, remove that directory with this command:

```
rm -rf /tmp/autoUpgrade
```

Upgrading Arcsight Suite

1. Download the upgrade bits - Metadata and Product offline images listed in "Download Installation Packages" on page 7 to the master node1 directory. For example: `/tmp`.
2. Upload images to the local docker registry

   ```
   {K8S_HOME}/scripts/uploadimages.sh -u registry-admin -y -F fusion-1.1.0.29.tar
   ```

   ```
   {K8S_HOME}/scripts/uploadimages.sh -u registry-admin -y -F recon-1.0.0.29.tar
   ```

   ```
   {K8S_HOME}/scripts/uploadimages.sh -u registry-admin -y -F transformationhub-3.3.0.29.tar
   ```
3. Label fusion server

   Identify the server `$servr` running Fusion, run: `kubectl label --overwrite node $server fusion=yes`

4. Add new metadata

   > **Note:** Make sure to copy the `arcsight-installer-metadata-2.3.0.29.tar` to the system where your web browser is running before performing the process below.

   a. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.

   b. Click **DEPLOYMENT > Metadata** and click **+ Add**

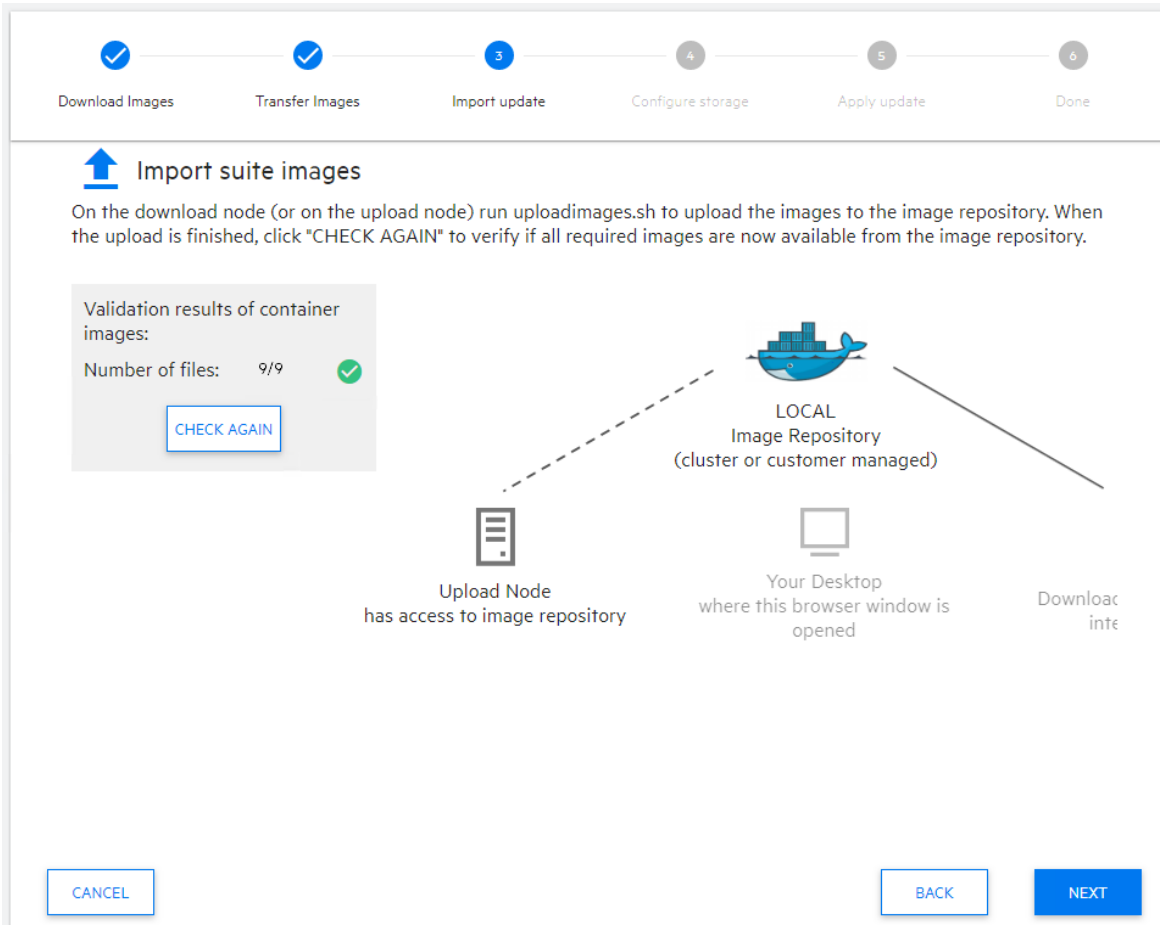
   c. Select `arcsight-installer-metadata-2.3.0.29.tar` from your system

   The new metadata will be added to the system.

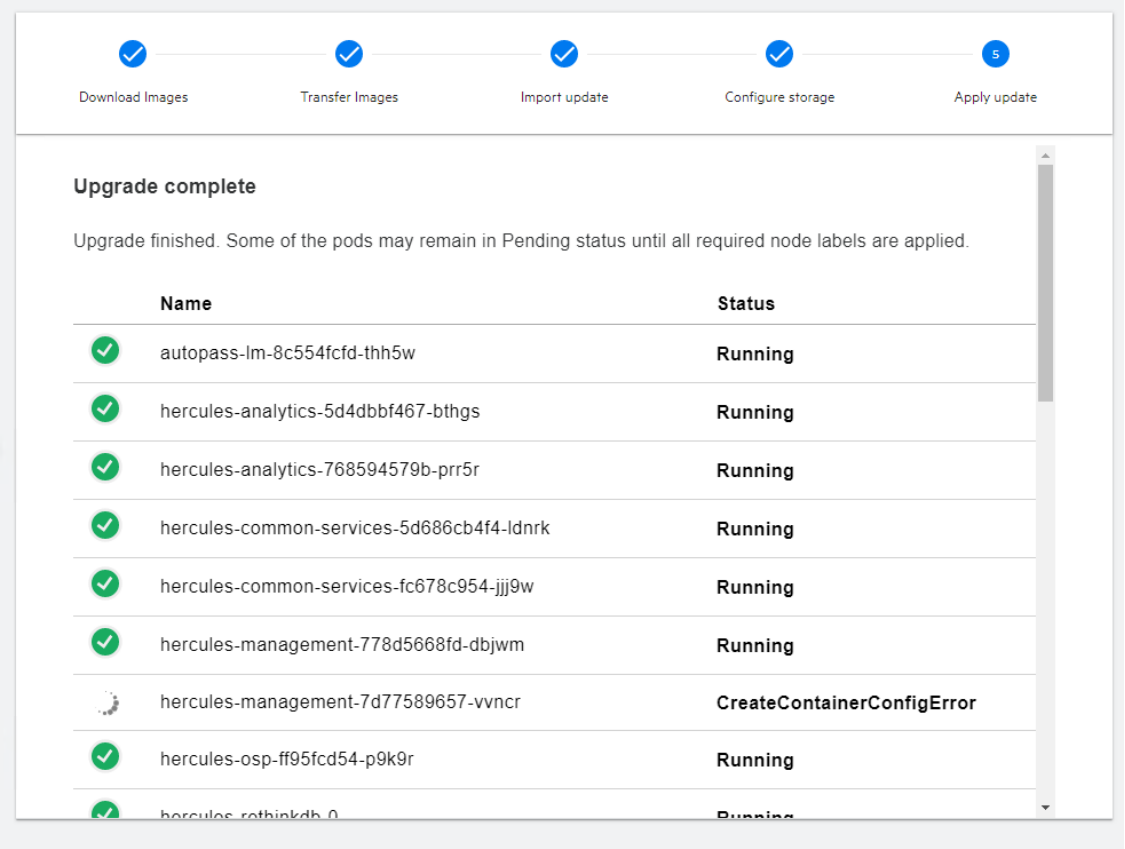
5. Start the upgrade process
   a. Go to **DEPLOYMENT > Deployments**. Notice the number **1** in the red circle on the Update column

   b. Click the red circle and choose your recently added metadata to initiate the upgrade
6. On the **Update to** page click **NEXT** until you reach the **Import suite images** page.

Ensure the validation results of container images is 9/9.



7. Click **NEXT** until you reach the **Upgrade Complete** page.



8. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443
   a. Click **DEPLOYMENT**, and select **Deployments**.
   b. Click the **Three Dots** (Browse) on the far right and choose **Change**. A new screen will be opened in a separate tab.
   c. Check the box of **Fusion** and **Arcsight Recon**, click **NEXT** until you reach the **Fusion** page.
   d. Under **Fusion > Database Configuration Page**, input the information for the following values and click **NEXT:**
      - Database Host
      - Database Application Admin User Name
      - Database Application Admin User Password
      - Search User Name
      - Search User Password
   e. You will be returned to the Configuration Complete page.
9. 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**

10. From the Kafka Manager, monitor EPS to `th-arcsight-avro` is increasing, i.e. not 0 anymore. For more information on how to set up the Kafka Manager, please see "Monitoring Transformation Hub's Kafka" on page 68.

11. To reload CDF images that were removed during the upgrade process due to a known issue, run the following command:

```
cd /tmp/upgrade-download/cdf-2020.05.00100-2.3.0.7/cdf/images

/opt/arcsight/kubernetes/scripts/uploadimages.sh \
-F cdf-master-images.tgz -u register-user -y

/opt/arcsight/kubernetes/scripts/uploadimages.sh \
-F cdf-common-images.tgz -u register-user -y

/opt/arcsight/kubernetes/scripts/uploadimages.sh \
-F cdf-phase2-images.tgz -u register-user -y
```

Note: The requested password is the admin USERID password.

Reset Scheduler Owner and Recreate Scheduler

From the database cluster node1:

1. `cd` to the database installer directory
2. Run the following commands:
   a. `./kafka_scheduler delete`
   b. `./sched_ssl_setup # with previously used options`
   c. `./kafka_scheduler create # with previously created options`

Delete old Outlier Model

1. Login to Recon UI
2. From the left navigation menu , select **Configuration > Outlier.**
3. Delete all existing Outlier Models.

Chapter 6: Managing Recon

This section provides information about managing Recon.

Monitoring Kubernetes Cluster

Log in to the CDF Management Portal, navigate to **Cluster > Dashboard** to access the Kubernetes Dashboard.

You can also use the following command/settings to monitor the Kubernetes cluster:

1. Check all pods status: `kubectl get pods --all-namespaces`
2. Check all pods status on each node: `kubectl get pods --all-namespaces -o wide`
3. Check all process status: `kubectl get svc --all-namespaces`
4. Check all pods status on each node: `kubectl get svc --all-namespaces -o wide`
5. Monitor pod status changes:

   ```
   watch 'kubectl get pods --all-namespaces | grep -v Running | grep -v Completed'
   ```

Pods Description

CDF Common Pods

The information given below describes the CDF Common Pods.

Namespace	Pod Prefix	Description
core	idm	User management and authentication to CDF Management Portal. **Note**: This is a separate user population than the Fusion user interface, which uses the hercules-management pod.

ArcSight Suite Common Pods

The information below describes the ArcSight Suite Pods in the namespace `arcsight-installer`.

Namespace	Pod Prefix	Description
arcsight-installer	autopass-lm	License key management service
arcsight-installer	itom-pg-backup	
arcsight-installer	nginx-ingress-controller	Proxy server, which end-users make web browser HTTPS port 443 connections to in order to access capabilities
arcsight-installer	suite-reconf-pod-arcsight-installer	Used for the Reconfiguration feature in the CDF Management Portal

Suite Feature Specific Pods

The pods given in this section are deployed only when the indicated Suite Feature is deployed.

Suite Feature	Pod Prefix	Description	Required Labels
Fusion	common-doc-web-app	User interface inline user guide user interface	fusion:yes
Fusion	dashboard-metadata-web-app	Dashboard metadata REST API	fusion:yes
Fusion	dashboard-web-app	Dashboard user interface	fusion:yes
Fusion	database-monitoring-web-app	Database monitoring REST API. When this pod starts up, it also installs the Health and Performance Monitoring out of the box dashboard as well as the widgets that are in that dashboard.	fusion:yes
Fusion	hercules-common-services	Core services, such as navigation menu capability management	fusion:yes

Suite Feature	Pod Prefix	Description	Required Labels
Fusion	hercules-management	User account and role management, authentication, for the Fusion user interface and the user interfaces that integrate with it. User and role data is stored within an embedded H2 database. **Note**: This is a separate user population than CDF Management Portal UI, which uses the **idm** pod.	fusion:yes
Fusion	hercules-osp	Single sign-on service, authentication	fusion:yes
Fusion	hercules-rethinkdb	A RethinkDB database that stores user configuration and preference information, such as a user's dashboards, favorites, etc.	fusion:yes
Fusion	hercules-search-engine	Provides APIs to access data in database	fusion:yes
Recon	hercules-analytics	Generates Outlier Analytics backend data. The Outlier user interface is served from hercules-search container	fusion:yes
Recon	hercules-search	Generates Search, lookup list, data quality dashboard and Outlier user interface	fusion:yes
ArcSight Layered Analytics	layered-analytics-widgets	When this pod starts up, it installs the **Entity Priority** out of the box dashboard and the Active List widget. This widgets connects to an ESM Manager server running outside of the Kubernetes cluster.	fusion:yes
ArcSight ESM Command Center	esm-widgets	When this pod starts up, it installs the **How is my SOC running?** out of the box dashboard and the ESM Case Management related widgets. The widgets connect to an ESM Manager server running outside of the Kubernetes cluster.	fusion:yes
ArcSight ESM Command Center	esm-acc-web-app	ESM Command Center user interface. This connects to an ESM Manager server running outside of the Kubernetes cluster.	fusion:yes
Interset	interset-widgets	When this pod starts up, it installs the Interset related widgets.	fusion:yes
Transformation Hub	th-cth	An instance of a Connectors IN Transformation Hub. Distributes the load coming from Collectors, by creating a consumer group based on the sourcetopic and the destination(s) topic name(s). There can be up to 50 CTH instances as of today.	th-processing:yes
Transformation Hub	th-c2av-processor	Converts CEF messages on topic th-cef to Avro on a topic th-arcsight-avro. The number of instances is based on the TH partition number and load. The default number of instances is 0.	th-processing:yes

Suite Feature	Pod Prefix	Description	Required Labels
Transformation Hub	th-c2av-processor-esm	Converts CEF messages on topic mf-event-cef-esm-filtered to Avro on a topic mf-event-avro-esmfiltered. There can be up to 10 instances of this particular type of pod. The default number of instances is 0.	th-processing:yes
Transformation Hub	th-kafka	Kafka Broker which is the core component of Kafka that publishers and consumers connect to in order to exchange messages over Kafka.	kafka:yes
Transformation Hub	th-kafka-manager	The Kafka Manager UI application to manage the Kafka Brokers.	th-platform:yes
Transformation Hub	th-routing-processor-group	Topics routing rules (a group of instances per source topic) that can be configured via ArcMC.	th-processing:yes
Transformation Hub	th-schemaregistry	Schema registry used for managing the schema of data in Avro format.	th-platform:yes
Transformation Hub	th-web-service	WebServices module of TH which is the API for ArcMC management to retrieve statistics, metrics, configuration current values and also serves as a way to push routing rules, new topics, configurations, etc	th-platform:yes
Transformation Hub	th-zookeeper	Confluent's Kafka's ZooKeeper pods.	zk:yes

Monitoring Transformation Hub's Kafka

1. Use **kafka manager** to monitor each topic:
   a. From a putty session, login to the Transformation Hub/Investigate cluster master node1:
      `kubectl get svc --all-namespaces | grep 9000 | awk '{print $4}'`
      The Kafka manager internal IP will be displayed, i.e. 172.17.17.184.
   b. Set up the tunnel to the Kafka manager from the putty session:
      i. Go to **Change Settings > Connection > SSH > Tunnels**
      ii. For **Source Port**, type in a value between 10001 and 19999.
      iii. For **Destination**, type the Kafka manager internal IP >:9000, i.e. 172.17.17.184:9000
      iv. Click **Add** then **Apply.**
   c. Start a browser session and navigate to: http://localhost:12345/
      i. Click **transformation-hub**
      ii. Under **Cluster Summary** click the number (default is 11) next to **Topics**

Recon License

This section explains the features, warnings and capacity of the Recon License, as well as the steps to install the 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 Recon 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 may 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.

Note: 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.

Installing Recon License

Follow the steps below to install the license.

1. Log in to the ArcSight Installer: `https://<Master_FQDN>:5443` or `https://<Virtualhost_FQDN>:5443` if you deployed in multi-master mode.
2. From the left menu click **Application > License.** This will display the license server tab.
3. Select **Install Licenses**, click **Choose File** to upload the corresponding XML file, and click **Next.**
4. Check the box next to the license you want to install and click **Install Licenses.**

The license has been added successfully.

Integrate Recon Single Sign-On with any External SAML 2 Identity Provider

This section provides the steps to integrate Recon Single Sign-on with any other external SAML 2.0 IDP software.

Note: Recon Single Sign-on 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.

Regarding the Trusted Provider Metadata:

The metadata document for a trusted SAML provider with which a Single Sign-on defined provider interacts must be obtained in a provider-specific manner. While not all providers do so, many supply their metadata documents via URL.

Once the trusted provider's metadata document (or the URL-accessible location of the document) is obtained, you must configure the Single Sign-on 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.

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.

Note: The IDP certificates need to be imported to the Recon Single Sign-on 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 osp
      ```

   c. Open a terminal in the currently running pod:

      ```
      kubectl exec -it hercules-osp-xxxxxxxxxx-xxxxx -n arcsight-installer-xxxxx -c hercules-osp –- 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:
   - Get the pod information:

     ```
     kubectl get pods --all-namespaces | grep osp
     ```

   - Delete the current running pod:

     ```
     kubectl delete pod hercules-osp-xxxxxxxxxx-xxxxx -n arcsight-installer-xxxxx
     ```

7. Retrieve the Recon Single Sign-On SAML service provider metadata from the Recon server:
   `https://EXTERNAL_ACCESS_HOST/osp/a/default/auth/saml2/spmetadata`
   `EXTERNAL_ACCESS_HOST` is the hostname of the Recon server.

8. Use the Recon Single Sign-On SAML service provider metadata to configure your IDP. For detailed instructions, see the IDP software documentation.
9. To establish a trust relationship between Recon Single Sign-On 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.

Single Sign-On Configuration

The fields below must be completed for the Single Sign-On Configuration. The values should not be null or empty.

- **Client ID:** Specifies the name to identify the SSO client to the OAuth server.
- **Client Secret:** Password for the SSO client.

Fresh Install

For Fresh install the default values for both **Client ID** and **Client Secret** will already be present. Users can change them in the configuration before proceeding and clicking save. Otherwise the default values will be used. Users will still be able to update the values and edit the configuration.

Upgrade

During the upgrade process the default values for both **Client ID** and **Client Secret** will be present in the configuration UI. Users will proceed with the default values. They should edit the configuration post-upgrade and change the default values.

Managing the Database

This section provides information about managing the database.

Backing Up and Restoring the Database

You should back up and restore the database before you upgrade it or before you add or remove a database node.

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 that 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 Vertica 9.1.0 and restore it to Vertica 9.2.1.
- Ingesting events into the database during backup might exclude the most recently ingested events

from the backup. To ensure that all events are backed up, stop ingestion before you start the backup.

- For optimal network performance, each database node should have its own backup host.
- Use one directory on each database node to store successive backups.
- You can save backups to the local folder on the database node, if there is enough space available, or to a remote server.
- You can perform backups on ext3, ext4, NFS and XFS file systems.

Preparing the Backup Host

Micro Focus recommends that each backup host have space for at least twice the database node footprint size. Consider your long-term backup storage needs.

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.

The database administrator must have password-less SSH access from database node1 to the backup hosts, as well as from the restored database node1.

To set up password-less SSH:

1. Log in to the backup server.
2. Create user `$dbadmin`.

   `$dbadmin` is the administrator for the database cluster.
3. Ensure that `$dbadmin` has write permission to the dedicated directory where you will store the backup.
4. Log in to database node1 as `root`.
5. Become the database administrator:

   ```
   # su -l $dbadmin
   ```
6. Setup password-less SSH for all backup servers:

   ```
   # 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. Database automatically installs these files at `/opt/vertica/share/vbr/example_configs.`

For more information, please 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
```

Note: You must save a copy of `db_backup.ini` for future tasks.

Note: 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 from the database node1 of the cluster.

Note: vbr Command Reference.

To back up the database:

1. Stop Kafka scheduler

   Login to database node1 as `root`

   ```
   # cd /opt/db-installer
   # ./kafka_scheduler stop
   ```

2. Initialize backup location

   ```
   # su - $dbadmin
   # vbr -t init --config-file db_backup.ini
   ```

   Initializing backup locations.

   Backup locations initialized.

3. Back up data:

   ```
   # vbr -t backup -c db_backup.ini
   Enter vertica password:
   Starting backup of database investigate.
   Participating nodes: v_investigate_node0001,v_investigate_node0002,v_
   investigate_node0003.
   Snapshotting database.
   Snapshot complete.
   Approximate bytes to copy: 270383427 of 270383427 total.
   [==================================================] 100%
   Copying backup metadata.
   Finalizing backup.
   Backup complete!
   ```

4. Verify that the backup files were written to the backup locations:

   ```
   # ssh 192.161.1.1 ls /opt/dbadmin/backups
   backup_manifest
   Objects
   Snapshots
   # ssh 192.161.1.2 ls /opt/dbadmin/backups
   backup_manifest
   Objects
   Snapshots
   # ssh 192.161.1.3 ls /opt/dbadmin/backups
   backup_manifest
   Objects
   Snapshots
   ```

Backing Up 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:

```
# 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:

```
# vbr --task full-check --config-file db_backup.ini

Enter vertica password:

Checking backup consistency.

List all snapshots in backup location:

Snapshot name and restore point: Vertica_backup_09_09_2019_20190909_010826,
nodes:['v_investigate_node0001', 'v_investigate_node0002', 'v_investigate_
node0003'].

Regenerating backup manifest for location rsync://
[192.168.10.11]:50000/opt/dbadmin/backups

Regenerating backup manifest for location rsync://
[192.168.10.12]:50000/opt/dbadmin/backups

Regenerating backup manifest for location rsync://
[192.168.10.13]:50000/opt/dbadmin/backups

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 Backups

This section describes how to view and delete backups.

To view available backups, run the following command:

```
# vbr --task listbackup --config-file db_backup.ini

Enter vertica password:

backup backup_type epoch objects include_patterns exclude_patterns nodes
(hosts) version file_system_type

Vertica_backup_09_09_2019_20190909_010826 full 6058

 v_investigate_node0001(192.168.10.11), v_investigate_node0002
(192.168.10.12), v_investigate_node0003(192.168.10.13) v9.2.1-6 [Linux]

The backup name includes the backup time-stamp.
```

Backup times-tamp can be found by using listbackup option, i.e. `20190909_010826` from `Vertica_backup_09_09_2019_20190909_010826.`

To delete a backup, run the following command:

```
# vbr --task remove --config-file db_backup.ini --archive 20190909_010826

Enter vertica password:

Removing restore points: 20190909_010826

Remove complete!
```

Preparing to Restore Database Data

Before you restore database data, ensure that your environment meets the following requirements:

- You can only restore backups to the same version of database from which you made the backup. For example, you cannot backup Vertica 9.1.0 and restore it to Vertica 9.2.1.
- You can restore backup to the original cluster where the backup was generated. However, all data ingested to the database after backup will be lost. If backup is restored to a new cluster, you must restore to a cluster that is identical to the cluster from which you made the backup (same or larger disk size). Ensure that the cluster meets the following requirements:
  - The target database is created and empty.
  - The target database name matches the backup database name.
  - The target database is stopped.
  - All database nodes in the target cluster are running.
  - All database node names in the target cluster match the names from the backup.

Restoring the Database

The `$dbadmin` user must restore from the database node1 of the cluster.

To set up password-less SSH:

1. Log in to the target database node1 as root.

2. Become the database administrator:

```
# su -l $dbadmin
```

3. Setup password-less SSH for all backup servers:

```
# ssh-copy-id -i ~/.ssh/id_rsa.pub $dbadmin@$back_up_server_ip
```

To restore the database:

1. Build a target database cluster that is identical to the original cluster.
2. Log in to the target database node1 and stop the database:

   ```
   # cd /opt/db-installer
   # ./db_installer stop-db
   ```
3. Become the `$dbadmin` user:

   ```
   # su -l $dbadmin
   ```
4. Copy `db_backup.ini` to `/home/$dbadmin`.
5. Restore the backup data:

   ```
   # vbr --task restore --config-file db_backup.ini
   ```
   The output should be similar to the following:

   ```
   Enter vertica password:
   Starting full restore of database Investigate.
   Participating nodes: v_investigate_node0001, v_investigate_node0002, v_
   investigate_node0003.
   Restoring from restore point: investigate_backup_20190909_010826
   Determining what data to restore from backup.
   [==================================================] 100%
   Approximate bytes to copy: 270383427 of 270383427 total.
   Syncing data from backup to cluster nodes.
   [==================================================] 100%
   Restoring catalog.
   Restore complete!
   ```

6. Start the database:

```
# exit
# ./db_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 Investigate started successfully
```

7. Start the Kafka scheduler:

```
# ./kafka_scheduler start
```

Configuring the Watchdog and Event Retention Time Policy on the Database

About Watchdog

A watchdog process automatically runs once a day to monitor cluster status and storage utilization.

When watchdog detects a cluster node is in DOWN state, it will try to restart the node.

When storage utilization reaches the defined threshold (default is 95%), watchdog will start to purge data until utilization is under threshold.

To modify the default threshold:

1. Login to database cluster node1 as root
2. Change the database installer directory:

```
cd /opt/db-installer
```

3. Change the storage threshold value:

```
vi db.properties
STORAGE_THRESHHOLD= <new value>
```

For better disk management you can also put in place a data retention policy alongside watchdog.

Data Retention Policy

The retention period can range from 1 to 366 days. The data retention policy is based on calendar days. Calendar day is based on event's Normalized Event Time (NET).

Time-based data retention is disabled by default. When you enable it, the default retention period is 90 days, but that can be modified at any time. If you run the data retention script on 6/30/2019 and the `db_retention_days property` is set to 90, then data older than 04/01/2019 will be deleted. You can purge data in real time or by using a scheduled cron job. Confirmation is needed when retention period is set to less than 30 days.

Note: Database data needs to be backed-up routinely. The backup policy is defined by the user. Always evaluate (-e option) retention policy before purging data.

To enable data retention:

1. Login to database cluster node1 as root
2. Change the database installer directory:

   ```
   cd /opt/db-installer
   ```
3. Check the cluster nodes disk usage

   ```
   ./db_installer status
   ```

   Check the `disk_space_free_percent` field to determine the retention day
4. Ensure your database is backed up.

   For more information, see "Backing Up the Database" on page 78.
5. Enable data retention policy:

   ```
   cd /opt/db-installer/config
   vi db_user.properties
   ```

   Uncomment `#db_retention_days=90`

6. Verify the number of days of data in the database:

   ```
   cd /opt/db-installer/scripts
   ./retention_policy_util.sh -t
   ```

   The result should be similar to the following:

   ```
   ------------------------------------------------------------------------
   Investigate has 100 day(s) with time-range: [2017-10-26 - 2018-02-06].
   ------------------------------------------------------------------------
   ```

   Note: There are more than 100 calendar days between 2017-10-26 and 2018-02-06. The results above show that there are only 100 event days, meaning that 100 days have incoming events. Certain calendar days did not have incoming events.

7. To change the default retention period, enter the following command:

   ```
   ./retention_policy_util.sh -u <Number_of_Days>
   ```

To enable automatic purging based on event retention time period:

1. To create the purge process, enter the following command:

   ```
   ./retention_policy_util.sh -s
   ```

   Note: A cron job is scheduled to purge data daily.

2. To verify the created cron job, enter the following command:

   ```
   ./retention_policy_util.sh -l
   ```

   Expected results:

   ```
   ------------------------------------------------------------------------------------

   Current retention value is set to: 90 day(s)
   ------------------------------------------------------------------------------------

   Current cronjob is running:
   (59 23 * * * /opt/installer/scripts/retention_policy_util.sh -p &>>
   /opt/installer/vertica-installer.log)
   ------------------------------------------------------------------------------------
   ```

3. To preview the purge results, enter the following command:

   ```
   ./retention_policy_util.sh -e
   ```

   The results should be similar to the following:

   ```
   ************************************************************************

   No data will be purged. This is only evaluation for your retention policy
   ************************************************************************
   ```

```
Will purge time range : [ 2017-10-26 - 2017-10-31 ].
Will purge day 1, (2017-10-26)
Will purge day 2, (2017-10-27)
Will purge day 3, (2017-10-28)
Will purge day 4, (2017-10-29)
Will purge day 5, (2017-10-31)
***** done *****
```

4. To purge data in real time, enter the following command:

   ```
   ./retention_policy_util.sh -p
   ```

5. To disable the purge cron job, enter the following command:

   ```
   ./retention_policy_util.sh -d
   ```

6. To verify the disabled cron job, enter the following command:

   ```
   ./retention_policy_util.sh -l
   ```

   Expected results:

   ```
   ------------------------------------------------------------------------------------

   Current retention value is set to: 90 day(s)
   ------------------------------------------------------------------------------------
   ```

Monitoring the Database

You can monitor the Database by using commands, or the out-of-the-box Health and Performance Monitoring dashboard included in Recon.

Watchdog

Database includes a watchdog, which monitors the database nodes, to automatically purge data when the disk usage exceeds storage threshold and to automatically restart the node when the database node goes down.

Database Status

Monitor the database status by using the following command:

```
/opt/db-installer/db_installer status
```

Scheduler Status

Monitor the scheduler's status by using the following command:

```
/opt/db-installer/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 Recon. 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.

Starting and Stopping Database

You can start or stop the Database by using the commands below.

Starting the Database

Start the database by using the following command:

```
/opt/db-installer/db_installer start-db
```

Stopping the Database

Stop the database by using the following command:

```
/opt/db-installer/db_installer stop-db
```

Backing Up and Restoring Recon Management and Search Datastores, and SSO

Backup and restore operation are performed on the NFS server when Recon needs to be re-installed to ensure the current state has been preserved.

Micro Focus recommends that you use a backup location that is not under the `<nfs_volume_path>.`

This procedure uses `/opt/recon/backup`, `/opt/sso/backup` directories as an example.

To back up the data stores:

1. After uninstalling Recon.
2. SSH to the NFS server.
3. Run the following commands:

   ```
   cd <arcsight_nfs_vol_path>/recon/
   ```

   Note: `<arcsight_nfs_vol_path>` is the nfs volume used for CDF installation, for example: `/opt/NFS_volume/arcsight-volume`

   ```
   mkdir -p /opt/recon/backup
   cp –R * /opt/recon/backup
   diff -rs /opt/recon/backup/mgmt <arcsight_nfs_vol_path>/recon/mgmt
   diff -rs /opt/recon/backup/search <arcsight_nfs_vol_path>/recon/search
   cd <arcsight_nfs_vol_path>
   mkdir -p /opt/sso/backup
   cp -r sso/* /opt/sso/backup
   ```

```
diff -rs /opt/sso/backup sso
```

If you do not receive a message that states that the files are identical, repeat the procedure.

4. Install Recon to resume operations.
5. Before you resume Recon operations, ensure that the pods are in Running status:

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

Restoring Recon Management and Search Datastores, and SSO

When restoring the Recon management and search datastores, and SSO, retain the original directory structure under `<arcsight_nfs_vol_path>/recon`, `<arcsight_nfs_vol_path>/sso.`

The management datastore will be restored to the `<arcsight_nfs_vol_path>/recon/mgmt/db` directory.

The search datastore will be restored to the `<arcsight_nfs_vol_path>/recon/search` directory.

The sso will be restored to the `<arcsight_nfs_vol_path>/sso` directory.

To restore the datastores:

1. Ensure that you have a valid backup of the datastores.
2. Restore the datastore before installing Recon.
3. SSH to the NFS server, and then run the following commands:

```
cd /opt/recon/backup
cp –R search/* <arcsight_nfs_vol_path>/recon/search
```

Reply `yes` to overwrite files and folders.

`<arcsight_nfs_vol_path>` is the nfs volume used for CDF installation, for example: /opt/NFS_volume/arcsight-volume

```
cd <arcsight_nfs_vol_path>/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 -rs <arcsight_nfs_vol_path>/recon/mgmt/db/h2.mv.db
/opt/recon/backup/mgmt/db/h2.mv.db
diff -rs <arcsight_nfs_vol_path>/recon/search /opt/recon/backup/search
cd /opt/sso/backup
cp –R * <arcsight_nfs_vol_path>/sso
```

Reply `yes` to overwrite files and folders.

```
diff -rs /opt/sso/backup sso
```

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 <arcsight_nfs_vol_path>/recon/
chown 1999:1999 -R <arcsight_nfs_vol_path>/sso/
```

Note: If your previous installation had SAML enabled you must re-run the steps under "Integrate Recon Single Sign-On with any External SAML 2 Identity Provider " on page 71.

5. Install Recon to resume operations.
6. Before you resume Recon operations, ensure that the pods are in Running status:

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

Adding Users and Groups

You can incorporate users either by manually adding them or by importing users and groups from ESM. To assign permissions to these users, you can create roles with specific sets of permissions and add users to those roles.

For more information about assigning permissions to users and roles, see the User's Guide for Fusion embedded in the product or posted with the documentation for the ArcSight Platform.

Changing Configuration Properties

To change configuration properties:

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. Update configuration properties as needed.
5. Click **Save.**

All services in the cluster affected by the configuration change will be restarted (in a rolling manner) across the cluster nodes.

Resetting the Administrator Password

You can change the administrator password on a CDF installation.

1. Browse to CDF Installer UI at `https://{master_FQDN or IP}:5443.` 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.)
2. Click **IDM Administration** in the left navigation pane.
3. In the main panel, click the large **SRG** button on the right.
4. In the left navigation bar, click **Users**.
5. In the list of users on the right, select *Admin* and click **Edit**.
6. In the bottom right, click **Remove Password.**
7. Click **Add Password.**
8. Enter a new admin password, and then click **Save**.

Displaying and Changing the Certificate Authority

The cluster maintains its own certificate authority (CA) to issue certificates for external communication. A self-signed CA is generated during installation by default. Pods of deployed products use the certificates generated by the CA on pod startup.

Displaying the current CA for external communication:

Run the following command on the Initial Master Node:

```
${k8s-home}/scripts/cdf-updateRE.sh read
```

Changing the CA:

Note: Changing the CA after Recon deployment will require undeploying and then redeploying Recon. This will result in a loss of configuration changes. It is highly recommended that if you need to perform this task, do so at the beginning of your Recon rollout.

1. Request 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. Request CSR from vault (you will need to export some 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 the vault to generate CSR:

```
/opt/arcsight/kubernetes/bin/vault write -tls-skip-verify -format=json \
-address=https://<VIP_or_external_access_host>:8200
RE/intermediate/generate/internal  \
common_name="none-MF CDF RE CA on <External accees host of FQDN single_
master>" \
| jq -r '.data.csr' > /tmp/pki_intermediate.csr
```

3. Use the csr file to sign it with your certificate authority and save it to `intermediate.cert.pem` An example with openssl:

```
openssl ca -keyfile your-rootca-sha256.key -cert your-rootca-sha256.crt \
 -extensions v3_ca -notext -md sha256 -in /tmp/pki_intermediate.csr -out
intermediate.cert.pem
```

4. Import the certificate back to the vault:

```
/opt/arcsight/kubernetes/bin/vault write -tls-skip-verify -format=json \
RE/intermediate/set-signed certificate=@intermediate.cert.pem
```

5. After confirmation of successful import you need to manually edit `RE_ca.crt` in your core and product namespace configmaps:

```
kubectl edit configmap -n core public-ca-certificates
kubectl edit configmap -n arcsight-installer-xxxx public-ca-certificates
```

6. Regenerate nginx certificate for your external access host:

```
/opt/arcsight/kubernetes/bin/vault write -tls-skip-verify -format=json \
RE/issue/coretech common_name=YOUR_EXTERNAL_ACCESS_HOST
```

7. Save the output results into nginx.CRT and nginx.KEY files accordingly and apply them:

```
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. Undeploy and then Redeploy Recon.

Note: Do not re-upload images during Recon redeployment.

Configuring Management Center

The Management Center (ArcMC) is the centralized console for managing Micro Focus products.

Connectivity between Transformation Hub and ArcMC is configured in ArcMC when you add Transformation Hub as a managed host into ArcMC.

Integrating Transformation Hub Into Your ArcSight Environment

Transformation Hub centralizes event processing and enables event routing, which helps you to scale your ArcSight environment and opens event data to ArcSight and third-party solutions. Transformation Hub takes advantage of scalable and highly-available clusters for publishing and subscribing to event data. 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.

Transformation Hub supports both Common Event Format (CEF) versions, 0.1 and 1.0.

- CEF 0.1 is the legacy ArcSight CEF version that supports IPv4 addresses available with SmartConnector version 7.4 and earlier.
- CEF 1.0, available with SmartConnector version 7.5 and later and Collectors version 7.8 and later, supports IPv4 and IPv6 addresses.

Transformation Hub third-party integration and other product features are explained in detail in the Transformation Hub Administrator's Guide, available from the ArcSight support community.

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
th-cef	CEF event data.	Can be configured as SmartConnector or Connector in Transformation Hub (CTH) destination.
th-binary_esm	Binary security events, which is the format consumed by ArcSight ESM.	Can be configured as a SmartConnector destination.
th-syslog	The Connector in Transformation Hub (CTH) feature sends raw syslog data to this topic using a Collector.	Can be configured as Collector destination.
th-cef-other	CEF event data destined for a non-ArcSight subscriber.	
th-arcsight-avro-sp_metrics	For ArcSight product use only. Stream processor operational metrics data.	
th-arcsight-avro	For ArcSight product use only. Event data in Avro format for use by ArcSight Recon.	
th-arcsight-json-datastore	For ArcSight product use only. Event data in JSON format for use by ArcSight infrastructure management.	

In addition, using ArcSight Management Center, you can create new custom topics to which your SmartConnectors can connect and send events.

Configuring ArcMC to Manage Transformation Hub

ArcMC serves as the management UI for Transformation Hub. In order for ArcMC to manage Transformation Hub, Transformation Hub must be added as a managed host to ArcMC. This process will include these steps, explained below:

- "Retrieve the ArcMC certificate" below
- "Configure the CDF cluster" on the next page
- "Retrieve the CDF certificate" on the next page
- "Configure ArcMC" on page 97

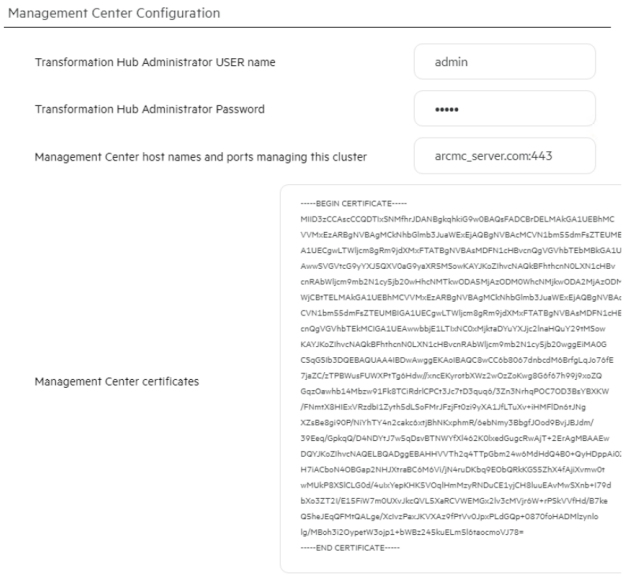
Retrieve the ArcMC certificate

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. Click **Suite**.
3. Click **...** (Browse) on the far right and choose and choose **Reconfigure**. A new screen will be opened in a separate tab.
4. Scroll down to the Management Center Configuration section. Then, enter values as described for the following:
   - **Username:** `admin`
   - 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.



5. Click **Save**. Web services pods in the cluster will be restarted

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.

Configure ArcMC

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:
   - **Hostname/IP:** IP address or hostname for the Virtual IP for an HA environment, or master node for a single- master node environment
   - **Type:** Select Transformation Hub Containerized (or, if using THNC, select *Non-containerized* instead)
   - **Port**: 38080
   - **Cluster Port:** 443
   - **Cluster Username:** admin
   - **Cluster Password:** <admin password created when logging into the CDF UI for the first time>
   - **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 Transformation Hub Destinations

To ensure secure communication among Recon, Transformation Hub, and the SmartConnectors, they all must use the same security mode. For additional Transformation Hub configuration, see the Transformation Hub ***Administrator's Guide*** and "Transformation Hub" in the ***Smart Connector User Guide*** on the Micro Focus Community.

Note: 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 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 using a command prompt window to enter Windows commands. Do not use Windows PowerShell.

Configuring a Transformation Hub Destination without Client Authentication in non-FIPS Mode

Follow these steps to configure an Transformation Hub destination from the SmartConnector 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 TH=<Transformation Hub hostname>_<Transformation Hub port>
   set STORES=%CURRENT%\user\agent\stores
   set STORE_PASSWD=changeit
   ```
4. Create the `user/agent/stores` directory if it does not already exist, for example:

   ```
   mkdir ${STORES}
   ```

   On Windows platforms:

   ```
   mkdir %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}
   ```
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, 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 %BC_OPTS% -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%
```

Configure a Transformation Hub Destination with Client Authentication in FIPS Mode

Follow these steps to configure a Transformation Hub (TH) destination from the SmartConnector with client authentication in FIPS mode.

Step 1: On the Connector 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`, at `<install dir>/current/user/agent` (or at `<install dir>\current\user\agent` on Windows platforms).
   b. Add the following content to the file and save:

   ```
   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=${CURRENT}/jre/lib/ext:${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 `user/agent/stores` directory if it does not already exist, for example:

   ```
   mkdir ${STORES}
   ```

   On Windows platforms:

   ```
   mkdir %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
   ```

   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'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. Place them in `/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
   ```

   Note: 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. See the *Transformation Hub Deployment Guide* for details.

2. `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>`

3. Create a temporary location on the Transformation Hub master server: `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}
   ```

   If successful, 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%
   ```

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%
   ```

   If successful, 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**.

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.

Configure a Transformation Hub Destination with Client Authentication in Non-FIPS Mode

Follow these steps to configure an Transformation Hub (TH) destination from the SmartConnector with client authentication, but in non-FIPS mode.

Step 1: On the Connector 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 `user/agent/stores` directory if it does not already exist, for example:

   ```
   mkdir ${STORES}
   ```

   On Windows platforms:

   ```
   mkdir %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%
```

Step 2: 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
```

Note: 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. See the *Transformation Hub Deployment Guide* for details.

2.
```
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 server:

   ```
   mkdir $CERT_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.

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 ${CERT_CA_TMP}/${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 -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%
```

If successful, 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%
```

If successful, 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
```

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.

Configure a Transformation Hub Destination without Client Authentication in FIPS Mode

Follow these steps to configure an Transformation Hub destination from the SmartConnector 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
   ${CURRENT}/lib/agent/fips/bc-fips-1.0.0.jar
   -J-Djava.security.egd=file:/dev/urandom"
   export TH=<Transformation Hub hostname>_<Transformation Hub port>
   export STORES=${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 `user/agent/stores` directory if it does not already exist, for example:

   ```
   mkdir ${STORES}
   ```

   On Windows platforms:

   ```
   mkdir %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
```

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
```

Troubleshooting SmartConnector Integration

The following troubleshooting tips may be useful in diagnosing SmartConnector integration issues.

Error Message	Issue
Unable to test connection to Kafka server: [Failed to construct kafka producer]	SmartConnector can't resolve the short or full hostname of the Transformation Hubnode(s).
Unable to test connection to Kafka server: [Failed to update metadata after 30000 ms.]	SmartConnector canresolve the short or full hostname of the Transformation Hubnode(s) but can't communicate with them because of routing or networkissues.
Unable to test connection to Kafka server: [Failed to update metadata after 40 ms.]	You have mistypedthe topic name. (Note the lower value in ms than in other messages.)
Destination parameters didnot pass the verification with error [; nestedexception is: java.net.SocketException: Connection reset]. Do you still want to continue?	If using SSL/TLS, you didnot configure the SSL/TLS parameters correctly.

Configuring Logger as a Transformation Hub Consumer

The procedure for configuring a Logger as a Transformation Hub producer will depend on whether the Logger will be using SSL/TLS.

To configure a Logger 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:** true
   - **Enable:** Checked

Troubleshooting

The following troubleshooting tips may be useful in diagnosing Logger integration issues.

Error Message	Issue
IP Address th1.example.com is not a valid address	Use UP 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 have mistyped the topic name.

Note: This process is explained in more detail in the Logger Administrator's Guide, available from the Micro Focus software community.

Configuring ESM as a Consumer

This procedure describes how to configure ESM as a Transformation Hub consumer with client authentication using a User (intermediate) certificate:

1. On Transformation Hub, run:

```
${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 ESM, run each of these commands one at a time on a ESM which has not be configured as a consumer. Use the password for the ESM.

```
/opt/arcsight/manager/config/client.properties
```

```
/opt/arcsight/manager//opt/arcsight/manager/bin/arcsight keytool -store clientkeys -storepasswd -storepass ""
```

```
/opt/arcsight/manager//opt/arcsight/manager/bin/arcsight keytool -store clientkeys -keypasswd -keypass "" -alias services-cn
```

```
/opt/arcsight/manager//opt/arcsight/manager/bin/arcsight changepassword -f config/client.properties -p ssl.keystore.password
```

3. Copy the intermediate certificate files to `/tmp` on the ESM.

```
/opt/arcsight/manager/bin/arcsight keytool -store clientcerts -importcert -file /tmp/ca.cert.pem -alias thcert
```

```
/opt/arcsight/manager/bin/arcsight keytool -store clientkeys -importcert -file /tmp/intermediate.cert.pem -alias thintcert
```

```
/opt/arcsight/manager/bin/arcsight keytool -store clientcerts -importcert -
file /tmp/intermediate.cert.pem -alias thintcert

/etc/init.d/arcsight_services stop manager

/opt/arcsight/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

/opt/arcsight/manager/bin/arcsight keytool -certreq -store clientkeys -alias
th -file thkey.csr
```

4. Copy the `.csr` file to the Transformation Hub initial master node.
5. On the Transformation Hub Initial Master Node, run:

```
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
```

6. Copy the signed certificate to `/tmp` on the ESM.
7. On the ESM, run:

```
/opt/arcsight/manager/bin/arcsight keytool -store clientkeys -alias th -
importcert -file /tmp/signedTHkey.crt -trustcacerts
```

8. Start the manager configuration:

```
/opt/arcsight/manager/bin/arcsight managersetup
```

9. Follow the wizard to add the Transformation hub to the ESM. On the dialog, under **"ESM can consume events from a Transformation Hub..."**, enter *Yes*, and enter then the following parameters. (This will put an entry in the Manager `cacerts` file, displayed as `ebcaroot`):

Host:Port(s): `th_broker1.com:9093,192.th_broker1.com:9093,th_broker1.com:9093`

> **Note:** You must use host names, not IP addresses. In addition, ESM does not support non-TLS port 9092.

Topic to read from: `th-binary_esm`

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
```

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 **Fusion > Log Configuration** and **Recon > Log Configuration** select the appropriate value to update the Log Levels.

Collecting Diagnostics Logs

Diagnostic log files help in investigating and troubleshooting issues. You can collect diagnostic logs from Operating System, CDF, Recon, and Transformation Hub.

Note: This script resides only on the all-in-one single node.

To collect the logs:

1. Log in to the all-in-one single node as root.
2. Change to the directory where Recon is installed:

   ```
   cd /opt/recon-installer-1.0.0.7
   ```
3. Execute the script to generate logs:

   ```
   ./support_utils.sh
   ```
4. Specify the password to encrypt the output file.

   The encrypted log file is stored in the location:

   ```
   /opt/support_util/<yyyymmddhhmmss>
   ```

   For example: `/opt/support_util/20200707043015/recon-installer-1.0.0.7-support-util-20200707043015.aes`
5. Decrypt the file as follows:

   ```
   dd if=<log_file_name> | openssl aes-256-cbc -md sha1 -d -k <Encrypt-Password> | tar zxf -
   ```

   For example:

   ```
   cd /opt/support_util/20200707043015
   ```

```
dd if=recon-installer-1.0.0.7-support-util-20200707043015.aes | openssl
aes-256-cbc -md sha1 -d -k <Encrypt-Password> | tar zxf -
```

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
th-cef	CEF event data.	Can be configured as SmartConnector or Connector in Transformation Hub (CTH) destination
th-binary_esm	Binary security events, which is the format consumed by ArcSight ESM.	Can be configured as a SmartConnector destination or as a Connector in Transformation Hub (CTH) destination.
th-syslog	The Connector in Transformation Hub (CTH) feature sends raw syslog data to this topic using a Collector.	Can be configured as Collector destination
th-cef-other	CEF event data destined for a non-ArcSight subscriber.	Non-ArcSight subscriber or as a Connector in Transformation Hub (CTH) destination
th-arcsight-avro-sp_metrics	For ArcSight product use only. Stream processor operational metrics data.	ArcSight product
th-arcsight-avro	For ArcSight product use only. Event data in Avro format for use by ArcSight Recon.	ArcSight product/Recon
th-arcsight-json-datastore	For ArcSight product use only. Event data in JSON format for use by ArcSight infrastructure management.	ArcSight product
mf-event-avro-esmfiltered	Event data in Avro format filtered for ESM	Should only be configured as the destination topic of the ESM event filtering
mf-event-cef-esmfiltered	CEF event data filtered for ESM	ArcSight product

In addition, using ArcSight Management Center, you can create new custom topics to which your SmartConnectors can connect and send events.

Starting and Stopping Kubernetes

You can start or stop the Kubernetes by using the commands below.

Starting the Kubernetes

Start the Kubernetes by using the following command:

```
/opt/arcsight/kubernetes/bin/kube-start.sh
```

Stopping the Kubernetes

Stop the Kubernetes by using the following command:

```
/opt/arcsight/kubernetes/bin/kube-stop.sh
```

Check the Kubernetes Status

Check the Kubernetes status by using the following command:

```
opt/arcsight/kubernetes/bin/kube-status.sh
```

Note: To stop the kubernetes cluster, please stop the worker nodes first and then the master nodes. To start the kubernetes cluster, please start the master nodes first and then the worker nodes.

Chapter 7: Appendices

This section provides additional information for managing Recon environment.

Troubleshooting

The following can help to diagnose common Recon issues.

[Vertica][VJDBC](5156) Error

2019-10-13 14:11:38.954 | ERROR | Caught SQLException during Leadership Lock Procedure. Rolling back txn. | java.sql.SQLTransactionRollbackException: [Vertica][VJDBC](5156) ERROR: Unavailable: initiator locks for query - Locking failure: Timed out X locking

After the scheduler is created, the *[Vertica][VJDBC](5156)* error will be displayed in the message and log file. This is normal and no action needs to be taken.

The scheduler uses Vertica transactions and locks to guarantee exclusive access to the scheduler's config schema. When you operate in HA mode and point multiple schedulers at the schema, they compete to acquire this lock. The scheduler that doesn't get it will receive this error.

If the Vertica cluster downtime exceeds the retention time for the Kafka cluster, the Vertica-stored Kafka offset might not be present in the Transformation Hub cluster. In this case, the scheduler will not be able to consume new data. This section describes how to resolve the issue.

You can confirm whether the scheduler is copying data by checking the status and examining the last copied offset in the microbatch status. If the offset number is not increasing, then the scheduler can no longer find the valid offset and must be reset.

To check the scheduler offsets, run the following command in the Vertica installation directory:

```
./kafka_scheduler events
```

…

```
Event Copy Status for (th-internal-avro) topic:
frame_start | partition | start_offset | end_offset | end_reason | copied bytes | copied messages
-------------------------+-----------+--------------+------------+-----------+--------------+-----------------
2018-06-09 16:57:40.599 | 1 | 6672721851 | 6672743683 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:40.599 | 2 | 6693800372 | 6693818421 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:40.599 | 0 | 6710608899 | 6710626273 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:40.599 | 4 | 6684909292 | 6684928573 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:40.599 | 5 | 6690363437 | 6690385300 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:40.599 | 3 | 6703797344 | 6703813421 | END_OF_STREAM | 0 | 0
```

```
2018-06-09 16:57:15.573 | 2 | 6693782400 | 6693800372 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:15.573 | 1 | 6672702552 | 6672721851 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:15.573 | 3 | 6703785764 | 6703797344 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:15.573 | 4 | 6684890676 | 6684909292 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:15.573 | 5 | 6690346763 | 6690363437 | END_OF_STREAM | 0 | 0
2018-06-09 16:57:15.573 | 0 | 6710597067 | 6710608899 | END_OF_STREAM | 0 | 0
```

If the scheduler is not consuming data, recreate the scheduler:

```
# ./kafka_scheduler delete
Are you sure that you want to DELETE scheduler metadata (y/n)?y
Terminating all running scheduler processes for schema: [investigation_
scheduler]
scheduler instance(s) deleted for 192.214.138.94
bash: /root/install-vertica/kafka_scheduler.log: No such file or directory
scheduler instance(s) deleted for 192.214.138.95
bash: /root/install-vertica/kafka_scheduler.log: No such file or directory
scheduler instance(s) deleted for 192.214.138.96
db cleanup: delete scheduler metadata
# ./kafka_scheduler create
192.214.137.72:9092,192.214.137.71:9092,192.214.136.7:9092
create scheduler under: investigation_scheduler
scheduler: create target topic
scheduler: create cluster for
192.214.137.72:9092,192.214.137.71:9092,192.214.136.7:9092
scheduler: create source topic for
192.214.137.72:9092,192.214.137.71:9092,192.214.136.7:9092
scheduler: create microbatch for
192.214.137.72:9092,192.214.137.71:9092,192.214.136.7:9092
scheduler instance(s) added for 192.214.138.94
scheduler instance(s) added for 192.214.138.95
scheduler instance(s) added for 192.214.138.96
```

rethinkdb Process Creation Failure (CrashLoopBackoff mode) during Recon installation.

The NetApp in use did not have the file-locking capability required for rethinkdb.

Users must switch to a NFS4 server which supports file-locking capability.

Database Backup Error: Accessing remote storage: failed accessing remote storage on <y.y.y.y>: @ERROR: Unknown module 'vbr' rsync error: error starting client-server protocol (code 5) at main.c(1506) [Receiver=3.0.7]

1. Check if rsync is running, if the service is running under a different port, please stop the rsync service and run the daemon.

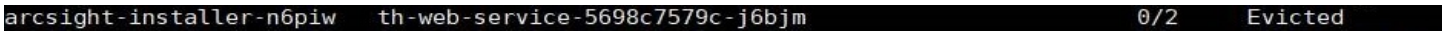
```
ps -aux| grep rsync
```

2. Run the rsync daemon with the specified port in the database backup.ini template

```
rsync --daemon --config=/tmp/vbr_rsyncd/vbr_rsyncd.conf --port=50000
```

3. Re-run the the backup task.

K8s Cluster Failed due to Pods Going to Evicted Status



1. Update the eviction policy.

2. Remove all the evicted pods (as shown in the image above):

```
kubectl delete pod -n arcsight-installer-n6piw th-web-service-5698c7579c-
j6bjm
```

3. Run `kube-restart.sh`

Setting FIPS on Database Server

In order to enable FIPS mode in Recon we have to set the OS in FIPS mode.

To enable FIPS in the OS

1. Run the below commands:

   ```
   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.

   ```
   mv -v /boot/initramfs-$(uname -r).img{,.bak}
   ```

```
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
```

b. To verify if FIPS has been enabled, run the following command:

```
sysctl crypto.fips_enabled
```

Expected Result: `crypto.fips_enabled = 1`

To disable FIPS

1. Run the below commands:

```
yum remove dracut-fips
dracut --force
grubby --update-kernel=$(grubby --default-kernel) --remove-args=fips=1
reboot
```

2. To verify if FIPS has been disabled, run the following command:

```
sysctl crypto.fips_enabled
```

Expected Result: crypto.fips_enabled = 0

Uninstalling ArcSight Suite

Caution: The uninstall process will remove all product suites.

To uninstall the ArcSight Suite

1. Stop all collectors and Connectors from sending events to Transformation Hub.
2. Stop all consumers from receiving events after they have consumed all events from their topics.
3. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.
4. Click **DEPLOYMENT**, and select **Deployments.**
5. Click the **Three Dots** (Browse) on the far right and choose **Uninstall**.

The pods are progressively shut down and then uninstalled.

Database SSL Chain Certificate Support

Creating Root Certificate

On a server:

Create a new ca key and cert

1. Create parameters for ca key

```
mkdir /root/ca

cd /root/ca

mkdir certs crl newcerts private

chmod 700 private

touch index.txt

echo 1000 > serial
```

2. Create the `/root/ca/openssl.cnf` file

`vi /root/ca/openssl.cnf` and add the following example contents:

```
# OpenSSL root CA configuration file.

# Copy to `/root/ca/openssl.cnf`.

[ ca ]

default_ca = CA_default

[ CA_default ]

# Directory and file locations.

dir               = /root/ca

certs             = $dir/certs

crl_dir           = $dir/crl

new_certs_dir     = $dir/newcerts

database          = $dir/index.txt

serial            = $dir/serial

RANDFILE          = $dir/private/.rand
```

```
# The root key and root certificate.
private_key       = $dir/private/ca.key
certificate       = $dir/certs/ca.crt
# For certificate revocation lists.
crlnumber         = $dir/crlnumber
crl               = $dir/crl/ca.crl.pem
crl_extensions    = crl_ext
default_crl_days  = 30
# SHA-1 is deprecated, so use SHA-2 instead.
default_md        = sha256
name_opt          = ca_default
cert_opt          = ca_default
default_days      = 375
preserve          = no
policy            = policy_strict
[ policy_strict ]
# The root CA should only sign intermediate certificates that match.
# See the POLICY FORMAT section of `man ca`.
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
[ policy_loose ]
# Allow the intermediate CA to sign a more diverse range of certificates.
# See the POLICY FORMAT section of the `ca` man page.
countryName             = optional
```

```
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
[ req ]
# Options for the `req` tool (`man req`).
default_bits        = 2048
distinguished_name  = req_distinguished_name
string_mask         = utf8only
# SHA-1 is deprecated, so use SHA-2 instead.
default_md          = sha256
# Extension to add when the -x509 option is used.
x509_extensions     = v3_ca
[ req_distinguished_name ]
countryName                     = US
stateOrProvinceName             = California
localityName                    = Sunnyvale
0.organizationName              = EntCorp
organizationalUnitName          = Arcsight
commonName                      = Common Name
emailAddress                    = Email Address
# Optionally, specify some defaults.
countryName_default             = GB
stateOrProvinceName_default     = England
localityName_default            =
0.organizationName_default      = abcd
```

```
organizationalUnitName_default  =
emailAddress_default            =
[ v3_ca ]
# Extensions for a typical CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
[ v3_intermediate_ca ]
# Extensions for a typical intermediate CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true, pathlen:0
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = client, email
nsComment = "OpenSSL Generated Client Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth, emailProtection
[ server_cert ]
# Extensions for server certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Generated Server Certificate"
```

```
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
[ crl_ext ]
# Extension for CRLs (`man x509v3_config`).
authorityKeyIdentifier=keyid:always
[ ocsp ]
# Extension for OCSP signing certificates (`man ocsp`).
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, digitalSignature
extendedKeyUsage = critical, OCSPSigning
```

3. Generate the new ca root key

```
cd /root/ca
openssl genrsa -out private/ca.key 4096
chmod 400 private/ca.key
```

4. Create the new ca cert

```
openssl req -config openssl.cnf \
-key private/ca.key \
-new -x509 -days 365 -sha256 -extensions v3_ca \
-out certs/ca.crt
...
If you enter '.', the field will be left blank.
-----
US [GB]:US
California [England]:California
```

```
Sunnyvale []:Sunnyvale

EntCorp [abcd]:

Arcsight []:Arcsight

Common Name []:root ca

Email Address []:admin@abcd.com
```

5. Verify the root ca

```
chmod 444 certs/ca.crt

openssl x509 -noout -text -in certs/ca.crt
```

Creating an Intermediate Certificate

1. Create parameters for intermediate key

```
mkdir /root/ca/intermediate/

cd /root/ca/intermediate

mkdir certs crl csr newcerts private

chmod 700 private

touch index.txt

echo 1000 > serial

echo 1000 > /root/ca/intermediate/crlnumber
```

   a. Create the /root/ca/intermediate/openssl.cnf file

```
vi /root/ca/intermediate/openssl.cnf  and add the following contents -
make sure the dir is unique for each intermediate cert created:

[ ca ]

default_ca = CA_default

[ CA_default ]

# Directory and file locations.

dir               = /root/ca/intermediate

certs             = $dir/certs

crl_dir           = $dir/crl

new_certs_dir     = $dir/newcerts
```

```
database          = $dir/index.txt
serial            = $dir/serial
RANDFILE          = $dir/private/.rand
# The root key and root certificate.
private_key       = $dir/private/intermediate.key
certificate       = $dir/certs/intermediate.crt
# For certificate revocation lists.
crlnumber         = $dir/crlnumber
crl               = $dir/crl/intermediate.crl.pem
crl_extensions    = crl_ext
default_crl_days  = 30
# SHA-1 is deprecated, so use SHA-2 instead.
default_md        = sha256
name_opt          = ca_default
cert_opt          = ca_default
default_days      = 375
preserve          = no
policy            = policy_loose
[ policy_strict ]
# The root CA should only sign intermediate certificates that match.
# See the POLICY FORMAT section of `man ca`.
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
[ policy_loose ]
```

```
# Allow the intermediate CA to sign a more diverse range of
certificates.

# See the POLICY FORMAT section of the `ca` man page.

countryName             = optional

stateOrProvinceName     = optional

localityName            = optional

organizationName        = optional

organizationalUnitName  = optional

commonName              = supplied

emailAddress            = optional

[ req ]

# Options for the `req` tool (`man req`).

default_bits        = 2048

distinguished_name  = req_distinguished_name

string_mask         = utf8only

# SHA-1 is deprecated, so use SHA-2 instead.

default_md          = sha256

# Extension to add when the -x509 option is used.

x509_extensions     = v3_ca

[ req_distinguished_name ]

# See <https://en.wikipedia.org/wiki/Certificate_signing_request>.

countryName                     = Country Name (2 letter code)

stateOrProvinceName             = State or Province Name

localityName                    = Locality Name

0.organizationName              = Organization Name

organizationalUnitName          = Organizational Unit Name

commonName                      = Common Name

emailAddress                    = Email Address
```

```
# Optionally, specify some defaults.
countryName_default             = GB
stateOrProvinceName_default     = England
localityName_default            =
0.organizationName_default      = abcd
organizationalUnitName_default  =
emailAddress_default            =
[ v3_ca ]
# Extensions for a typical CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
[ v3_intermediate_ca ]
# Extensions for a typical intermediate CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true, pathlen:0
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = client, email
nsComment = "OpenSSL Generated Client Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth, emailProtection
```

```
[ server_cert ]
# Extensions for server certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Generated Server Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
[ crl_ext ]
# Extension for CRLs (`man x509v3_config`).
authorityKeyIdentifier=keyid:always
[ ocsp ]
# Extension for OCSP signing certificates (`man ocsp`).
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, digitalSignature
extendedKeyUsage = critical, OCSPSigning
```

b. Generate the new Intermediate ca key

```
cd /root/ca
openssl genrsa -out intermediate/private/intermediate.key 4096
```

c. Create the intermediate ca certificate signing request (csr)

```
chmod 400 intermediate/private/intermediate.key
openssl req -config intermediate/openssl.cnf -new -sha256 \
-key intermediate/private/intermediate.key \
-out intermediate/csr/intermediate.csr.pem
...
```

```
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name [England]:California
Locality Name []:Sunnyvale
Organization Name [abcd]:
Organizational Unit Name []:Arcsight
Common Name []:intermediate ca
Email Address []:admin@abcd.com
```

d. Create the new Intermediate ca cert

```
cd /root/ca
openssl ca -config openssl.cnf -extensions v3_intermediate_ca \
-days 3650 -notext -md sha256 \
-in intermediate/csr/intermediate.csr.pem \
-out intermediate/certs/intermediate.crt
Sign the certificate? [y/n]: y
1 out of 1 certificate requests certified, commit? [y/n]y
chmod 444 intermediate/certs/intermediate.crt
```

e. Verify the Intermediate ca

```
openssl x509 -noout -text \
-in intermediate/certs/intermediate.crt
```

f. Verify the Intermediate cert against the root ca

```
openssl verify -CAfile certs/ca.crt \
intermediate/certs/intermediate.crt
# intermediate.crt: OK
```

Creating CA chains Certificate

```
cd /root/ca
```

```
cat certs/ca.crt intermediate/certs/intermediate.crt > chain.crt
```

Creating Database server Certificate

a. Create database key

```
openssl genrsa -out vertica.key 4096
```

b. Create database server certificate signing request

```
openssl req -new -key vertica.key -out vertica.csr  -subj
"/C=US/ST=California/L=Santa Clara/O=Micro
Focus/OU=Arcsight/CN=Vertica/emailAddress=admin@abcd.com" -nodes -sha256
```

c. Sign the certificate signing request

```
openssl x509 -req -in vertica.csr -CA
intermediate/certs/intermediate.crt -CAkey
intermediate/private/intermediate.key -CAcreateserial -extensions server
-days 3650 -outform PEM -out vertica.crt
```

d. Verify the scheduler client certificate

```
openssl verify -CAfile chain.crt vertica.crt

vertica.crt: OK
```

Creating scheduler client Certificate

1. Create client key

```
openssl genrsa -out scheduler.key 4096
```

2. Create client certificate signing request

```
openssl req -new -key scheduler.key -out scheduler.csr  -subj
"/C=US/ST=California/L=Santa Clara/O=Micro
Focus/OU=Arcsight/CN=Scheduler/emailAddress=admin@abcd.com" -nodes -sha256
```

3. Sign the certificate signing request

```
openssl x509 -req -in scheduler.csr -CA
intermediate/certs/intermediate.crt -CAkey
intermediate/private/intermediate.key -CAcreateserial -extensions client -
days 3650 -outform PEM -out scheduler.crt
```

4. Verify the scheduler client certificate

```
openssl verify -CAfile chain.crt scheduler.crt
```

```
scheduler.crt: OK
```

Installing self-signed CA during the TH installation

1. Install Transformation Hub

```
cdf-2020.02/install  --k8s-home /opt/arcsight/kubernetes -u admin ......
```

2. Access the CDF UI

```
https://n15-214-128-h125.arcsight.com:3000
```

After infrastructure services are deployed, wait for the **Preparation Complete** page to be displayed.

3. Installing intermediate certificate and key

```
scp previously generated intermediate.key, intermediate.crt, and ca.crt to
Master node1's /opt/cert.

On Master node1

mkdir /opt/cert

scp previously generated intermediate.key, intermediate.crt, and ca.crt
to /opt/cert

cd /opt/cert

/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh write --re-
key=/opt/cert/intermediate.key --re-crt=/opt/cert/intermediate.crt  --re-
ca=/opt/cert//ca.crt

...

Dry run to check the certificate/key files.

Success! Enabled the pki secrets engine at: RE_dryrun/

Success! Data written to: RE_dryrun/config/ca

Success! Disabled the secrets engine (if it existed) at: RE_dryrun/

Dry run succeeded.

Submitting the certificate/key files to platform. CA for external
communication will be replaced.

Success! Disabled the secrets engine (if it existed) at: RE/

Success! Enabled the pki secrets engine at: RE/

Success! Data written to: RE/config/ca
```

```
Success! Data written to: RE/roles/coretech

Success! Data written to: RE/config/urls

Warning: kubectl apply should be used on resource created by either
kubectl create --save-config or kubectl apply

secret/nginx-default-secret configured

configmap/public-ca-certificates patched

configmap/public-ca-certificates patched
```

4. Continue with the installation.
5. Under **Transformation Hub > Security Configuration page**, turn ON **Connection to kafka uses TLS Client Authentication.**

   Note: TLS Client Authentication and FIPS need to enabled at this time if the system is planning to use TSL client authentication and FIPS. Client Authentication in post-deployment can't be changed after this point.

6. Continue with the Transformation Hub/Recon suite deployment.
7. Enable SSL on Database cluster

```
On Vertica server node1,

mkdir /opt/cert

scp created chain.crt scheduler.crt scheduler.key vertica.crt vertica.key
intermediate.key to /opt/cert

chown -R $dbadmin:$dbadmin /opt/cert
```

8. Enable database server SSL

```
cd to /opt/db-installer

./db_ssl_setup --enable-ssl --vertica-cert-path /opt/cert/vertica.crt --
vertica-key-path /opt/cert/vertica.key --client-ca-path
/opt/cert/chain.crt

...

2020-07-15 14:27:11,422 DEBUG Installing Certs/Keys for SSL: Return code:
0, Out put: Parameters set successfully

2020-07-15 14:27:11,451 DEBUG Enabling EnableSSL flag: Return code: 0,
Output: A LTER DATABASE
```

```
WARNING 4324: Parameter EnableSSL will not take effect until database
restart

2020-07-15 14:27:11,451 INFO ENABLED SSL/TLS MODE FOR VERTICA

...

Starting Vertica on all nodes. Please wait, databases with a large catalog
may take a while to initialize.

...

Database investigate: Startup Succeeded. All Nodes are UP
```

9. Verify database SSL
   a. Login to dabase node1 server as $dbadmin

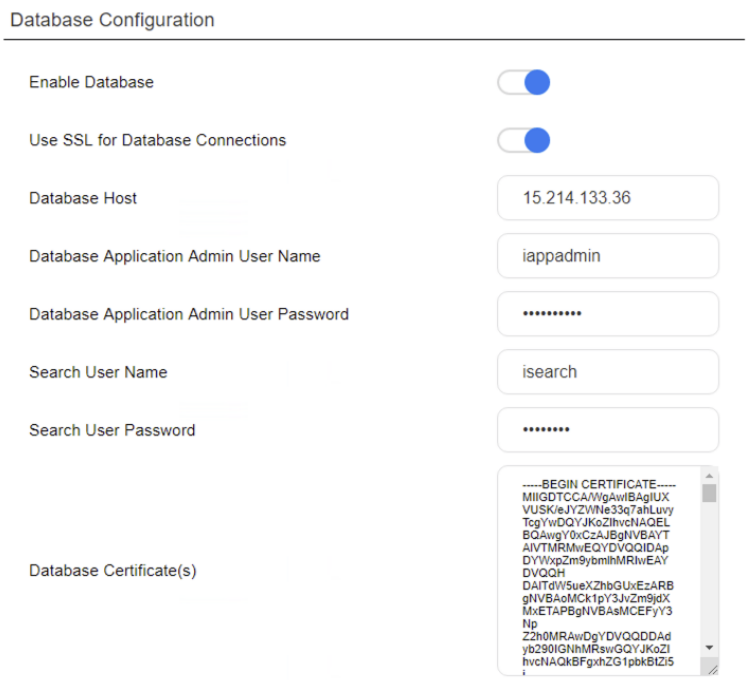
```
mkdir ~/.vsql
cp /opt/cert/scheduler.crt ~/.vsql/client.crt
cp /opt/cert/scheduler.key ~/.vsql/client.key
cp /opt/cert/chain.crt ~/.vsql/root.crt
chmod 400 ~/.vsql/client.key
```

   b. Check the database connection

```
vsql -m require
Password:
Expected result:
SSL connection (cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256, protocol:
TLSv1.2)
dbadmin=>select user,authentication_method, ssl_state from sessions
where session_id = current_session();
Expected result:
current_user | authentication_method | ssl_state
-------------+----------------------+-----------
dbadmin | Password | Mutual (1 row)
```

Configure SSL Connection for Database from Management Portal.

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. Go to **Fusion > Database Configuration >**
   a. Turn ON **Use SSL for Database Connections**
   b. Copy `/opt/cert/chain.crt` to the Database Certificate(s) field



Enabling SSL in scheduler

```
cd /opt/db-installer
```

```
./sched_ssl_setup --enable-ssl --sched-cert-path /opt/cert/scheduler.crt --
sched-key-path /opt/cert/scheduler.key --vertica-ca-path /opt/cert/chain.crt
--vertica-ca-key /opt/cert/intermediate.key --kafka-ca-path
/opt/cert/chain.crt
```

```
....
```

```
Entry for alias vertica_caroot successfully imported.
```

```
Import command completed: 1 entries successfully imported, 0 entries failed
or cancelled
```

```
...
```

```
2020-07-15 14:44:08,566 INFO Key pair imported successfully into
/opt/installer/wrk/ks.pkcs12
```

```
2020-07-15 14:44:10,040 DEBUG Import Key Pair: Return code: 0, Output:
Importing keystore /opt/installer/wrk/ks.pkcs12 to
/opt/installer/wrk/scheduler.keystore.bcfks...
```

```
Entry for alias scheduler_key successfully imported.
```

```
Import command completed: 1 entries successfully imported, 0 entries failed
or cancelled
```

```
2020-07-15 14:44:10,040 INFO Key pair imported successfully into
/opt/installer/wrk/scheduler.keystore.bcfks

2020-07-15 14:44:10,041 INFO Created file /opt/installer/wrk/vkconfig.cnf
successfully

...
```

Creating Scheduler with SSL Enabled

```
./kafka_scheduler create <TH_WorkerNode1>:9093
```

Database SSL Root Certificate Support

Certificate Creation:

Create a self-signed CA:

```
openssl req -newkey rsa:4096 -sha256 -keyform PEM -keyout ca.key -x509 \
-days 3650 -outform PEM -out ca.crt \
-subj "/C=US/ST=California/L=Santa Clara/O=Micro Focus/OU=Arcsight/\
CN=RootCA/emailAddress=admin@microfocus.com" -nodes
```

Generate the Certificate for Vertica

1. Create the server key:

```
openssl genrsa -out vertica.key 4096 -nodes -sha256
```

Generating RSA private key, 4096 bit long modulus

.....++

.....++

e is 65537 (0x10001)

2. Create Server certificate signing request:

```
openssl req -new -key vertica.key -out vertica.csr \
-subj "/C=US/ST=California/L=Santa Clara/O=Micro Focus/OU=Arcsight/\
CN=Vertica/emailAddress=admin@microfocus.com" -nodes -sha256
```

3. Sign the Certificate Signing Request with self-signed CA:

```
openssl x509 -req -in vertica.csr -CA ca.crt -CAkey ca.key \
-CAcreateserial -extensions server -days 3650 -outform PEM -sha256 \
-out vertica.crt
```

Signature ok

subject=/C=US/ST=California/L=Santa Clara/O=Micro Focus/OU=Arcsight/CN=FQDN/emailAddress=admin@microfocus.com

Getting CA Private Key

Create the Vertica Scheduler Client Certificate

1. Create the certificate key for the Vertica scheduler:

```
openssl genrsa -out scheduler.key 4096
```

Generating RSA private key, 4096 bit long modulus

.....++

.....++

e is 65537 (0x10001)

2. Create the Vertica scheduler client certificate signing request:

```
openssl req -new -key scheduler.key -out scheduler.csr \
-subj "/C=US/ST=California/L=Santa Clara/O=Micro Focus/OU=Arcsight/\
CN=Scheduler/emailAddress=admin@microfocus.com" -nodes -sha256
```

3. Sign the certificate signing request:

```
openssl x509 -req -in scheduler.csr -CA ca.crt -CAkey ca.key \
-CAcreateserial -extensions client -days 3650 -outform PEM -sha256 \
-out scheduler.crt
```

Signature ok

subject=/C=US/ST=California/L=Santa Clara/O=Micro Focus/OU=Arcsight/CN=scheduler/emailAddress=admin@arcsight.com

Getting CA Private Key

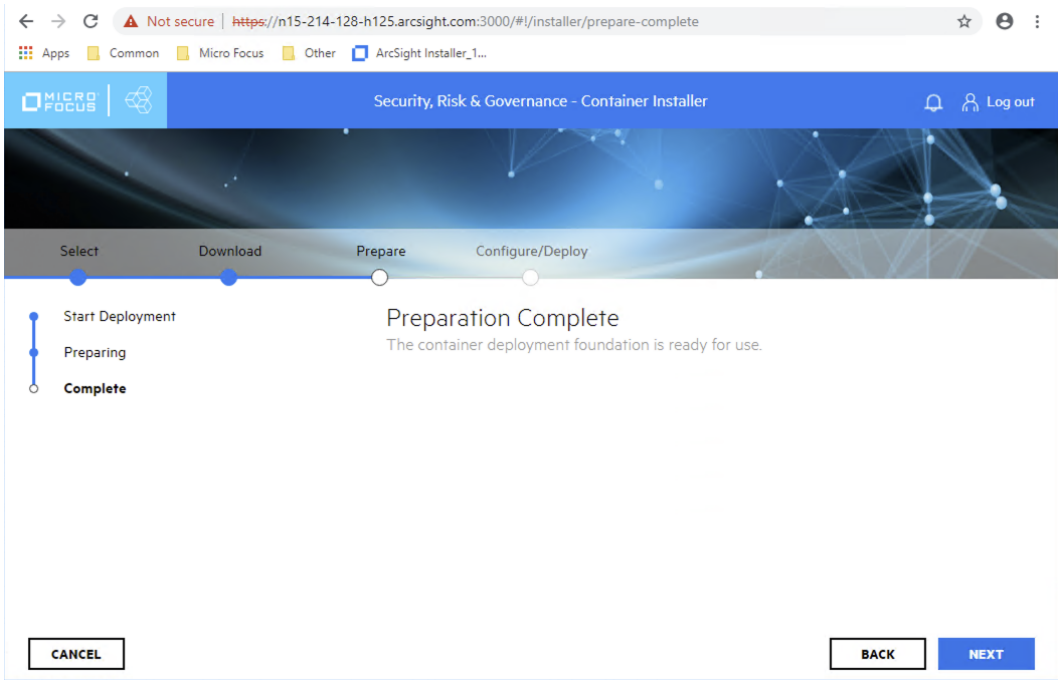
Change the key files permissions

Run the following command:

```
chmod 600 ca.key vertica.key scheduler.key
```

Installing Self-Signed CA during the Transformation Hub Installation

1. Install the Transformation Hub. For more information see the Transformation Hub Deployment guide available from the Micro Focus Community.
2. Access the CDF UI



3. After infrastructure services have been deployed, copy the generated ca.crt and ca.key to the Transformation Hub server /tmp directory and Install the self-signed CA

```
/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh write \
--re-key=/tmp/ca.key --re-crt=/tmp/ca.crt
```

Dry run to check the certificate/key files.

Success! Enabled the pki secrets engine at: RE_dryrun/

Success! Data written to: RE_dryrun/config/ca

Success! Disabled the secrets engine (if it existed) at: RE_dryrun/

Dry run succeeded.

Submitting the certificate/key files to platform. CA for external communication will be replaced.

Success! Disabled the secrets engine (if it existed) at: RE/

Success! Enabled the pki secrets engine at: RE/

Success! Data written to: RE/config/ca

Success! Data written to: RE/roles/coretech

Success! Data written to: RE/config/urls

Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply

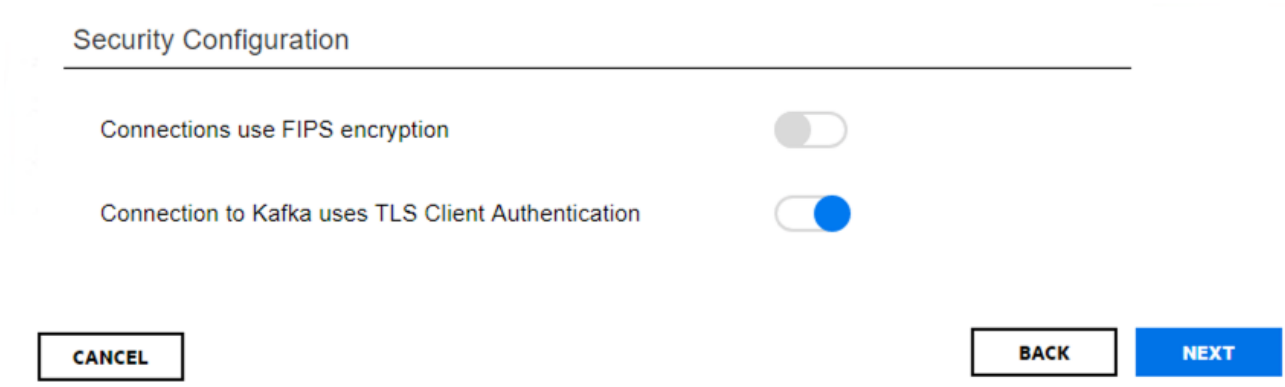
secret/nginx-default-secret configured

configmap/public-ca-certificates patched

configmap/public-ca-certificates patched

4. Proceed with the Transformation Hub installation and into the configuration page

> **Note:** TLS Client Authentication and FIPS need to be enabled at this time. Client Authentication and FIPS cannot be enabled or disabled in the Transformation Hub **Reconfigure** page.



Enabling Vertica SSL

1. Copy the following files to the Vertica server /tmp directory:
   - vertica.crt
   - vertica.key
   - schedule.crt
   - schedule.key
   - ca.crt
2. Change the certificate key file ownership:

```
chown <dbadmin user> vertica.key scheduler.key
```

3. Enable the Vertica server SSL

```
./vertica_ssl_setup --enable-ssl --vertica-cert-path /tmp/vertica.crt \
--vertica-key-path /tmp/vertica.key --client-ca-path /tmp/ca.crt
```

Verification:

4. Login to vertica server as dbadmin user

```
mkdir ~/.vsql
cp /tmp/scheduler.crt ~/.vsql/client.crt
cp /tmp/scheduler.key ~/.vsql/client.key
```

```
cp /tmp/ca.crt ~/.vsql/root.crt

chmod 600 ~/.vsql/client.key
```

5. Login to vertica cluster node1 as root user:

```
rm -rf /tmp/vertica.crt /tmp/vertica.key /tmp/issue_ca.crt /tmp/ca.crt
```

6. Check the Vertica connection:

```
vsql -m require
```

Password:

Expected result:

```
 SSL connection (cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256, protocol:
TLSv1.2)
```

Run the following command:

dbadmin=> `select user,authentication_method, ssl_state from sessions where session_id = current_session();`

Expected result:

```
current_user | authentication_method | ssl_state
-------------+----------------------+-----------

dbadmin | Password | Mutual

(1 row)
```

Enabling SSL in Scheduler

To enable SSL in scheduler, run the following command:

```
./sched_ssl_setup --enable-ssl --sched-cert-path /tmp/scheduler.crt \

--sched-key-path /tmp/scheduler.key --vertica-ca-path /tmp/ca.crt \

--kafka-ca-path /tmp/ca.crt
```

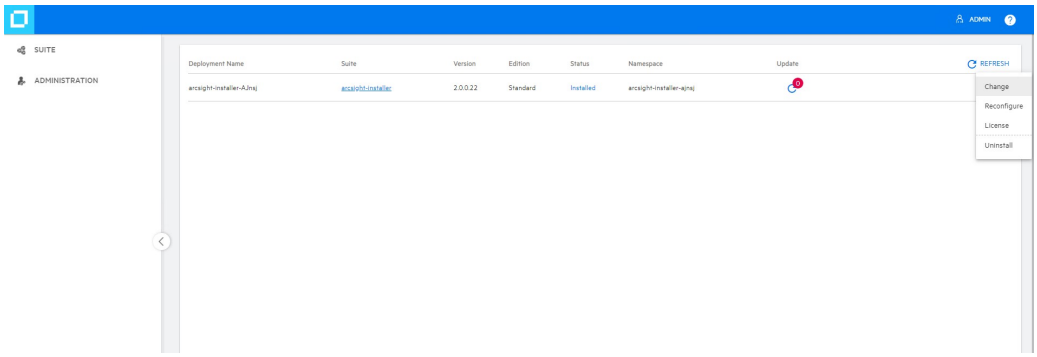
Creating Scheduler with SSL Enabled

To create Scheduler with SSL enabled, run the following command:

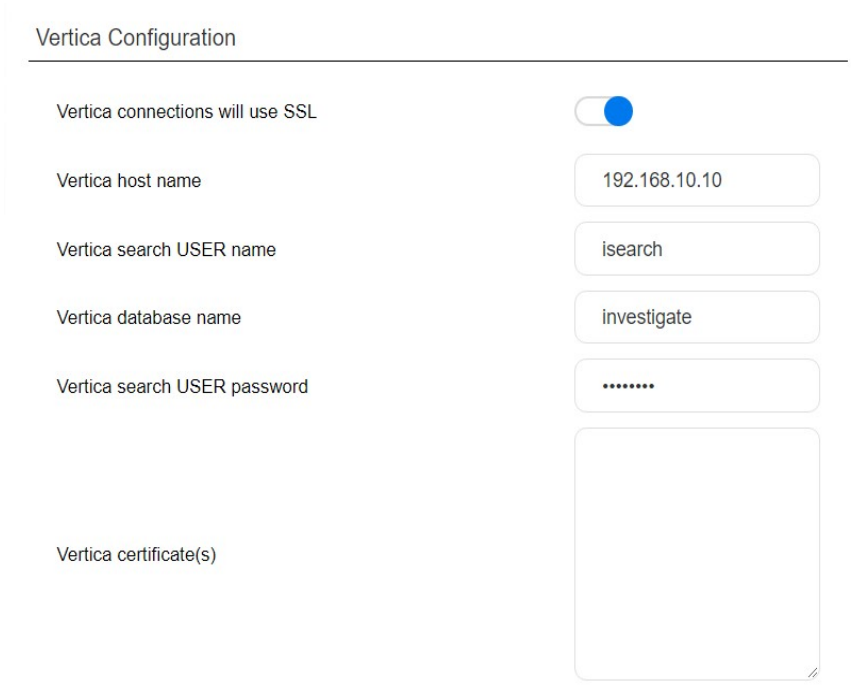
```
$vertica-install-DIR/kafka_scheduler create <WorkerNode1>:9093
```

Setting up Recon with SSL Enabled

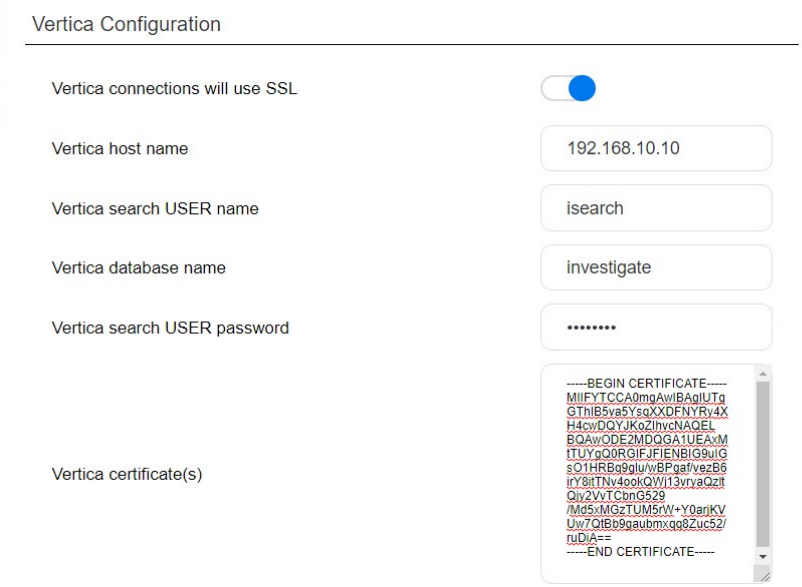
1. Browse to https://<virtual-server-FQDN>:5443, if it is a multiple master, or https://<master-FQDN>:5443, if it is a single master.
2. Navigate to suite options: **Suite > Management**
3. Click the **...** icon under **REFRESH** and Select **Reconfigure.** A new tab will be opened.



4. Select **ANALYTICS**, and scroll down to **Vertica Configuration**
5. Under **Vertica Configuration**, enable **Vertica connections will use SSL**



6. Copy the Vertica ca certificate into the **Vertica Certificate(s)** field, make sure not to include any blank spaces or missing line breaks to prevent a handshake authentication failure.



7. Click **SAVE**. This will restart the search engine pod for the SSL changes to take effect

Fields Indexed by Default in Database

Recon adds "rawEvent" field or a subset of event fields (see the table below) to a text index for use in free form text search. Free form text search can only be done for values in event fields that are indexed.

If the "rawEvent" field has a value, it will be tokenized and indexed. Otherwise, the following columns will be tokenized and indexed.

applicationProtocol	deviceDomain	message
categoryDeviceGroup	deviceEventCategory	name
categoryDeviceType	deviceEventClassId	oldFileId
categoryObject	deviceExternalId	oldFileName
categoryOutcome	deviceFacility	oldFilePath
categorySignificance	deviceHostName	oldFilePermission
categoryTechnique	deviceInboundInterface	oldFileType
destinationDnsDomain	deviceNtDomain	reason
destinationHostName	deviceOutboundInterface	requestClientApplication
destinationNtDomain	deviceProcessName	requestCookies
destinationProcessName	deviceProduct	requestMethod
destinationServiceName	deviceSeverity	requestUrl
destinationUserId	deviceVendor	sourceDnsDomain
destinationUserName	eventOutcome	sourceHostName
destinationUserPrivileges	externalId	sourceNtDomain
deviceAction	fileId	sourceProcessName
deviceCustomString1	fileName	sourceServiceName
deviceCustomString2	filePath	sourceUserId
deviceCustomString3	filePermission	sourceUserName
deviceCustomString4	fileType	sourceUserPrivileges
deviceCustomString5	categoryBehavior	transportProtocol
deviceCustomString6	flexString1	
deviceDnsDomain	flexString2	

If users need to index certain event fields that are not in the list above, they can work with support in editing the `superschema_vertica.sql` file in the installer before installing database.

If users want to modify the event fields indexed after database has been installed, and there are already events in the database, they will need to drop the text index and recreate it. This may take a while depending on how many events are in the system.

CDF Installer Script `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.
`--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.

Argument	Description
`--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 resolutuion in production environments due to hostname resolution issues and 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-hos`t 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 ITOM core volume.
`--nfs-server`	Address of the NFS host.
`--pod-cidr-subnetlen`	Specifies the size of the subnet allocated to each host for pod network addresses. For the default and the allowable values see the CDF Planning Guide.
`--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. For the default and allowable values see the CDF Planning Guide.
`--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.`

Argument	Description
`--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`
`--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.

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 (Recon 1.0)

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!