Micro Focus Security ArcSight Investigate

Software Version: 3.1.0

Deployment Guide

Document Release Date: April, 2020
Software Release Date: April, 2020

MICRO FOCUS

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

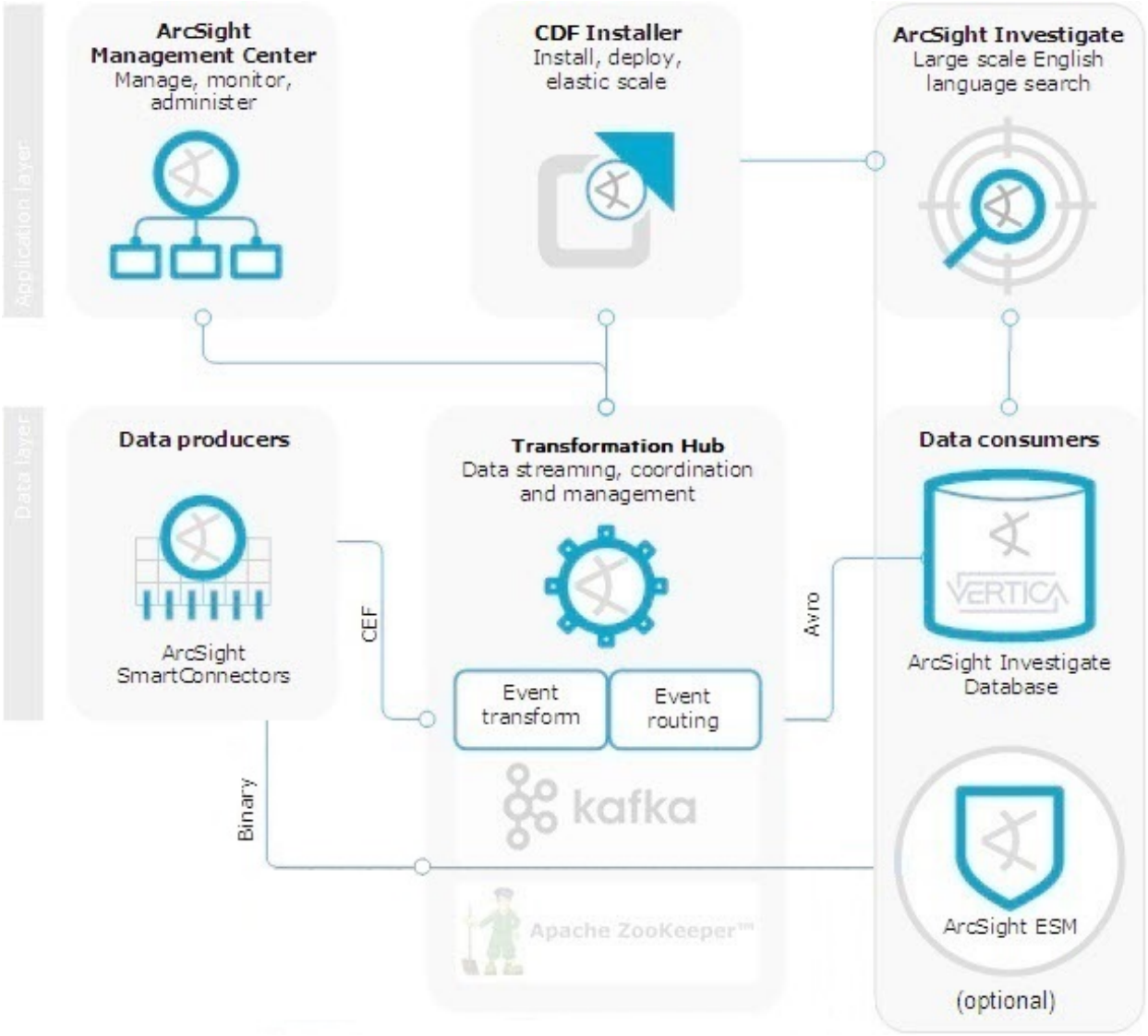
Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

Contents

Chapter 1: Overview

Arcsight Investigate

ArcSight Investigate 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. Investigate 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 and then create reports and visualizations based on the search results.



ArcSight Investigate Vertica database

- Investigate analytic database is powered by Vertica.
- Install the Vertica database separately.

Transformation Hub

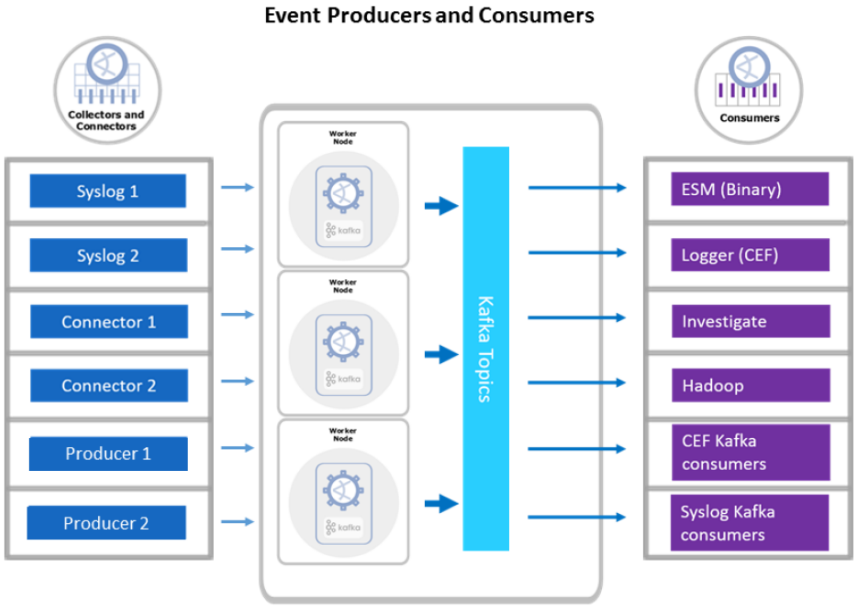
Transformation Hub is the high-performance message bus for ArcSight security, network, flows, application, and other events. It can queue, transform, and route security events to other ArcSight or third party software. This Kafka-based platform allows ArcSight components like Logger, ESM, and Investigate 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, analytics platforms, and other security technologies and the multiple systems within the Security Operations Center (SOC). Transformation Hub can seamlessly broker data from any source and to any destination. Its architecture is based on Apache Kafka and it supports native Hadoop Distributed File System (HDFS) capabilities, enabling both the ArcSight Logger and ArcSight Investigate technologies to push to HDFS for long-term, low-cost storage.

The latest releases of ArcSight Investigate 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 an analytics 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)

Identity Intelligence

Micro Focus Identity Intelligence provides interactive and reporting capabilities for identity governance data so you can evaluate requests and approval process activities, support audits of identity governance processes, and review the status of users and access rights. Identity Intelligence gathers data from Micro Focus Identity Manager and Micro Focus Identity Governance, then pushes it to the provided Transformation Hub for processing and Vertica for storage.

Logger

ArcSight Logger provides proven cost-effective and highly-scalable log data management and retention capabilities for the SIEM, expandable to hundreds of nodes and supporting parallel searches. Notable features of Logger include:

- Immutable storage
- High compression
- Archiving mechanism and management
- Transformation Hub integration
- Advanced reporting wizard
- Deployed as an appliance, software or cloud infrastructure
- Regulatory compliance packages

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: System Requirements

This chapter provides information about supported operating systems, browsers, and compatibility between ArcSight components.

Supported Operating Systems

ArcSight Investigate supports the following operating systems:

Version	Component	Operating system
Investigate 3.1.0	Investigate	CentOS/RHEL 7.7 and 8.1
	Vertica 9.2.1-6 database	CentOS/RHEL 7.6 and 7.7
Transformation Hub 3.2.0	Transformation Hub	CentOS/RHEL 7.7 and 8.1

Supported Browsers

You can use the following browsers with Investigate:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge

Investigate supports the browser version that is available at the time of the Investigate release.

NFS Server Requirements

- Use NFS4 server.
- If using NetApp NFS server, the 'NLM' service must be added on either or both NFS server and client to provide the file-locking capability.

Chapter 3: Deployment Planning and Preparation

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 Investigate.***

The complete process of deploying Investigate comprises the following high-level steps:

1. **Configure and Install the CDF Installer:** The CDF Installer installs the container management infrastructure. Containerized applications, such as Transformation Hub and Investigate, run in this environment. Depending on your environment, you may need to adjust the default installation parameter values.
2. **Configure and Deploy the Kubernetes Cluster:** Configure and deploy the Master and Worker Nodes, NFS storage, network connectivity, and other infrastructure requirements.
3. **Configure and Deploy Transformation Hub and Investigate:** Using the CDF Installer wizard, configure and deploy Transformation Hub and Investigate to run in the CDF-managed Kubernetes cluster.
4. **Manage Transformation Hub from the Management Center:** Configure the Management Center (ArcMC) to recognize and manage the Transformation Hub cluster.
5. **Integrate Transformation Hub with Other ArcSight Products:** Configure your SmartConnectors and Collectors as producers of events into Transformation Hub and add destinations, as well as configure event Consumers such as Logger and ESM.

Note: The deployment process will validate the infrastructure environment of both, Transformation Hub and Investigate before and after deployment.

Deployment Overview

Before you deploy Investigate, you must install and configure the Vertica database and CDF Installer, and then use ArcSight Installer to deploy Transformation Hub.

Note: Micro Focus recommends that you install these components in a test environment before you put them into production.

1. Obtain the CDF Installer.
2. Obtain the Investigate image and Vertica Installer.
3. Obtain the Transformation Hub images.

4. Configure the Vertica server and install the database.
5. Ensure that Transformation Hub and Investigate each have a dedicated server.

   If other applications run on the same servers as Transformation Hub and Investigate, you might experience performance problems.
6. Install the CDF Installer
7. Deploy both Transformation Hub 3.2 and Investigate 3.1.0.
8. Configure both Transformation Hub 3.2 and Investigate 3.1.0.

   Note: The installation process will validate the Transformation Hub infrastructure environment before performing the installation, as well as after the installation has completed.

For detailed instructions on the operation and management of Investigate and Transformation Hub after initial deployment, see the Investigate User's Guide and the Transformation Hub Administrator's Guide, available from the Micro Focus Community.

Gather Required Information

During the process described in CDF Planning Guide, you made configuration decisions about your environment, platforms, network, and storage. You will need this information handy now in order to complete the installation of CDF and Transformation Hub.

- **Master and Worker Node Info:** Ensure you have relevant configuration information of the Master and Worker Nodes, including:
  - Credentials for the root or `sudo` (non-root) user that will be used to run the deployment
  - IP Address and FQDN for every host system in the cluster
  - NFS Server IP Address and FQDN
  - Virtual IP (only required if Master Nodes are configured for high-availability)
- **License Keys:** Ensure you have all required Micro Focus License keys for the software being installed.
- **Security Mode:** Determine security settings (FIPS, TLS, and/or Client Authentication) for communication between ArcSight components.
- **Infrastructure:** Validate and, if necessary, remediate Transformation Hub infrastructure prerequisites.
  - Review, analyze and adjust your Transformation Hub infrastructure configuration properties to meet throughput expectations (for example, Events per Second processing rates).
  - Copy the CDF Deployment Disk Sizing Calculator spreadsheet (available from the Micro Focus support community) and edit its contents to determine your disk storage requirements and apply these during the pre-deployment configuration process.

- **Download Access:** Finally, ensure you have access to the Micro Focus software download location. You will download installation packages to the Initial Master Node.

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 appropriate 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 2.92 or later		443, 38080	• 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

Product	Preparations needed...	Ports	Supported security modes	More information
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

Leader Acknowledgement ("ACK") and TLS Enablement: In general, enabling leader ACKs and TLS results in significantly lower throughput rates, but greater fidelity in ensuring events are received by Subscribers. Micro Focus has seen results over 800% slower when both Leader ACK and TLS are enabled, versus when both were not active. For more information on Leader Acknowledgements and TLS enablement and their effects on processing throughput, refer to the Kafka documentation which explains these features.

Download Installation Packages

Now download the installation packages for both the CDF Installer and the 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 for Investigate 3.1.0 are:

- cdf-2020.02.00120-2.2.0.2.zip
- analytics-3.1.0.10.tar
- arcsight-installer-metadata-2.2.0.10.tar
- investigate-3.1.0.10.tar
- transformationhub-3.2.0.10.tar
- arcsight-vertica-installer_3.1.0-3.tar.gz
- cdf-upgrade-2019.08.00134-2.2.0.2.tar (Upgrade only)
- post-install-3.1.0.tar.gz (Upgrade only)

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.

Navigate to the version of Transformation Hub you wish to install and download the installation packages for the CDF Installer, the Transformation Hub, and all supporting scripts and wizards that help automate these installs 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 Transformation Hub 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.

Unarchive Installation Packages

Run the following commands to unarchive your installation packages.

```
unzip cdf-2020.02.00120-2.2.0.2.zip

tar -xvf transformationhub-3.2.0.xxx.tar

tar -xvf investigate-3.1.0.xxx.tar

tar -xvf analytics-3.1.0.xxx.tar
```

Resulting Directories

After the successful validation and decompression of the installation packages, the following directories and files will be located on your Initial Master Node and contain the installation materials:

```
/opt/arcsight/download /cdf-2020.02.00120-2.2.0.2

/opt/arcsight/download /transformationhub-3.2.0.xxx
```

```
/opt/arcsight/download /investigate-3.1.0.xxx

/opt/arcsight/download /analytics-3.1.0.xxx

/opt/arcsight/download /arcsight-installer-metadata-2.2.0.xxx.tar
```

Calculating volume storage usage for each Transformation Hub (TH) worker node

The **volume storage**, i.e. /opt, is where kubernetes and all its related product image and data reside.

The Deployment Size Calculator spreadsheet will be used to calculate the disk storage used for th-cef topic and th-arcsight-avro topic only.

The th-cef topic partition size will be applied to all other predefined topics, i.e. th-binary_esm, th-cef-other, and th-syslog,

Only th-cef and th-arcsight-avro topic partition number will be changed.

Determine CDF Hard Eviction Policy on Worker Node:

Container Deployment Foundation (CDF) uses a hard eviction policy for worker node. When a hard eviction policy threshold is met, Kubernetes stops all pods immediately.

The default CDF eviction policy is 15%, which means that 15% of the volume disk storage on the worker node can't be used.

Please determine your CDF eviction policy here. To modify the hard eviction policy please see"Additional Steps" on page 43

Total volume disk storage reference:

Total volume disk storage **=**

(CDF hard eviction policy) **+** (th-cef topic partition size ***** Partition number on each TH worker node) **+** (th-arcsight-avro topic partition size * Partition number on each TH worker node) **+**

(other topics size) **+** (Total th-cef and th-arcsight-avro topic partition overhead, i.e. 0.2%) **+**

(Total th-arcsight-avro topic partition overhead) **+** (Storage for upgrade , i.e.50GB) **+** (some buffer storage)

Chapter 4: Configuring the Vertica Server and Installing the Database

This chapter provides information about configuring the Vertica server and installing the database.

Note: Before you install Vertica, 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 Vertica Server

To configure the Vertica server details, please see the Vertica Hardware Guide, and the Vertica System Configuration Task Overview.

The procedure described in this section is a guideline for reference only.

The server configuration is based on an HPE ProLiant DL380 Gen9 server with 48 cores and 128 GB memory.

To avoid performance issues, the Vertica server should be a dedicated server.

Note: Vertica data should be backed-up routinely. For more information, please see "Backing Up and Restoring the Vertica Database" on page 63. Old Vertica data can be cleaned up, for more information, please see "Enabling the Data Retention Policy on the Vertica Cluster" on page 60.

To configure the Vertica server:

1. Provision the server with at least 2 GB of swap space, running on CentOS 7.6 and 7.7 or RHEL 7.6 and 7.7.

   Note: Vertica 9.2.1 supports ext3, ext4, NFS, and XFS file system. 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 processor 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 8192 /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.

Firewall Requirements

Vertica 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	Notes
22	TCP	sshd	Required by Administration Tools and the Management Console Cluster Installation wizard.
5433	TCP	Vertica	Vertica client (vsql, ODBC, JDBC, etc) port.
5434	TCP	Vertica	Intra- and inter-cluster communication.
5433	UDP	Vertica	Vertica spread monitoring.
5444	TCP	Vertica Management Console	MC-to-node and node-to-node (agent) communications port. See Changing MC or Agent Ports.
5450	TCP	Vertica 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.

Port	Protocol	Service	Notes
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/conf
   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 SSH Access

This section describes how to enable password-less SSH access from the node 1 server to all of the node servers in the cluster.

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

To enable password-less SSH access:

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

   ```
   ssh-keygen -q -t rsa
   ```
2. Copy the key from node 1 to all of the nodes, including node 1, 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 node 1 to the target node to verify that node 1 can successfully log in:

   ```
   ssh root@11.111.111.111
   ```

To Install Vertica

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

1. On the Vertica cluster node 1 server, create a folder for the Investigate Vertica database installer script:

   ```
   mkdir $vertica-install-DIR
   ```

   Note: `$vertica-install-DIR` should not be under `/root`.

2. Copy `arcsight-vertica-installer_3.1.0-3.tar.gz` to `$vertica-install-DIR.`
3. Extract the `.tar` file:

   ```
   cd $vertica-install-DIR
   tar xvfz arcsight-vertica-installer_3.1.0-3.tar.gz
   ```

4. Edit the `config/vertica_user.properties` file. The `hosts` and `license` properties are required.

Property	Description
`hosts`	A comma separated list of the Investigate Vertica 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.).
`license`	`$path/$license-file` Download the license file from the Software Licenses and Downloads portal, and then edit this parameter to point to the license file. **Note:** Without a valid license, an instant-on license will be applied to build a 3 node Vertica cluster only.
`db_retention_day`	Used for the data retention policy.

5. Install Vertica:

   ```
   ./vertica_installer install
   ```

   When prompted, create the database administrator user and the Investigate search user.

   Vertica now supports multiple users:

   • **Database administrator:** Credentials required to access the Vertica database host to perform database related operations, i.e. setup, configuration, and debugging.

   • **Search user:** Credentials required when configuring Vertica from the ArcSight Installer for Investigate search engine.

   • **Ingest user:** Should not be used or changed, this user is internally used for Vertica-scheduler, i.e.

ingestion.

For a list of options that you can specify when installing Vertica, see Vertica Installer Options.

Chapter 5: Installation and Deployment

Note: Before using Investigate 3.1.0 for the first time, users need to clean up their browser cookies. This applies for fresh-install, re-installation and upgrade.

Once the installation packages have been downloaded, validated, and uncompressed, you are ready to proceed with installation and deployment. In outline, the complete installation and deployment of Transformation Hub consists of these steps, which must be performed in order:

1. Configure and Deploy the CDF Installer
2. Configure and Deploy Kubernetes (k8s)
3. Upload Core Images to the Docker Registry
4. Configure and Deploy Transformation Hub and Investigate

Each of these steps is explained in detail in this chapter.

Configure and Install the CDF Installer

Once the installation packages have already 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 and install the CDF Installer:

1. Log in to the Initial Master Nodes where you downloaded and extracted the installation files as the root user. Installations will be initiated from the Initial Master Node.

2. 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/{unzipped_CDF_directory}
```

```
./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 your Admin password, 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 /opt/arcsight/download/cdf-2020.02.00120-2.2.0.2
```

```
./install -m /tmp/arcsight-installer-metadata-2.2.0.xxx.tar.gz --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,15.78.235.235" --nfs-server pueas-
vmenv-nfs.swinfra.net --nfs-folder /opt/nfs/volumes/itom/itom_vol --ha-
virtual-ip 216.3.128.12
```

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

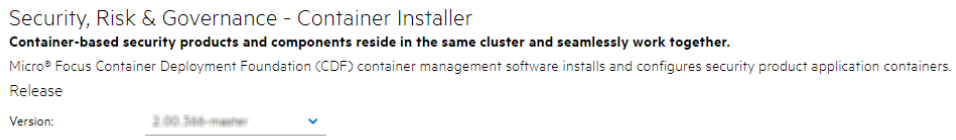
Note: For a description of valid CDF Installer command line arguments, see Installer CLI Commands.

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

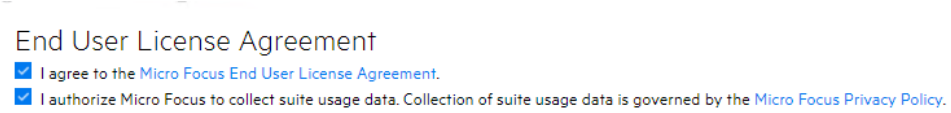
Configure and Deploy the Kubernetes Cluster

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

1. Browse to the 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. Then, click **Next**.

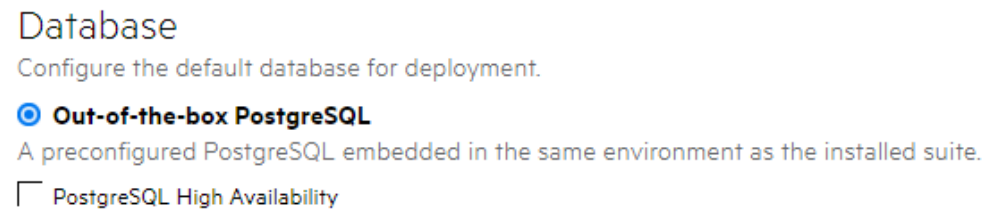


3. On the **End User License Agreement** page, review the EULA and select the ***'I agree...'*** checkbox. You may optionally choose to have suite utilization information passed to Micro Focus. Then, click **Next**.

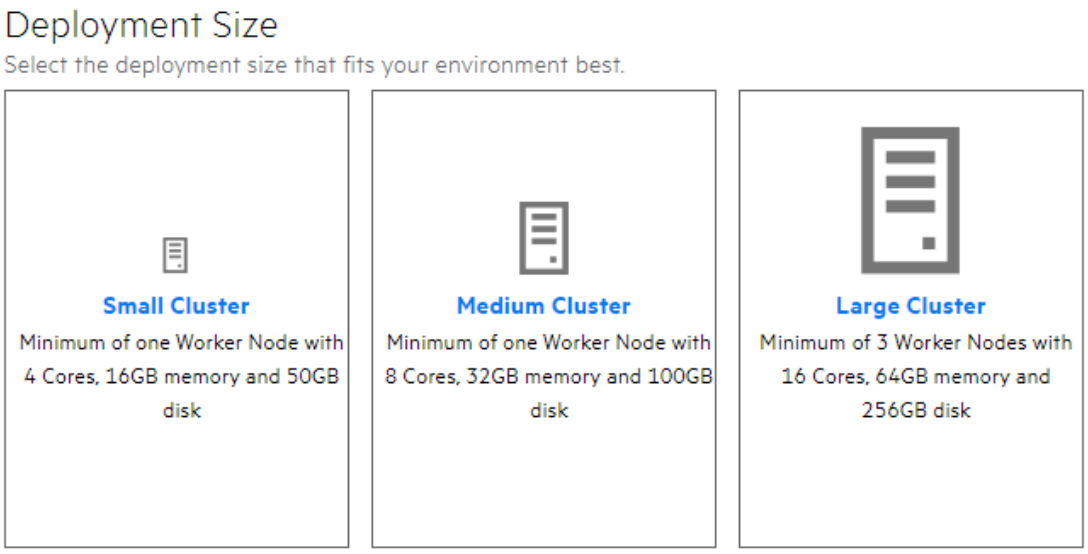


4. On the **Capabilities** page, choose the capabilities and/or products to be installed. To install Transformation Hub as a standalone install, select it. (Note that other products may require Transformation Hub or other capabilities as prerequisites. Such requirements will be noted in the pull-down text associated with the capability.) To show additional infomation associated with the product, click the **>** (greater than) arrow. 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)

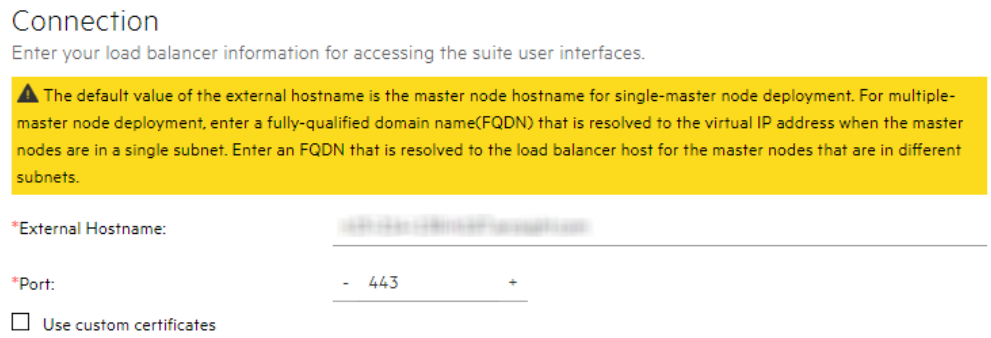
Note: The installation will not proceed until the minimal hardware requirements for the deployment are met.

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

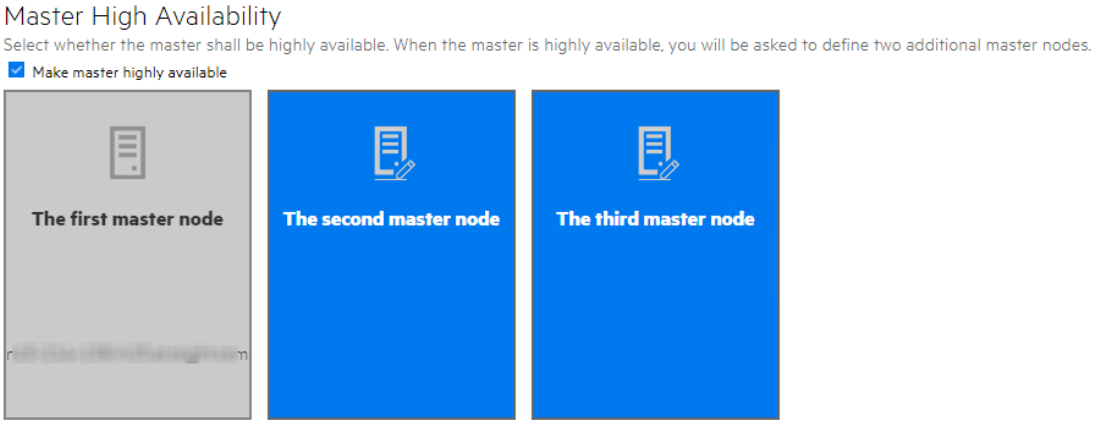
Select your appropriate deployment size, and then click **Next**.

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

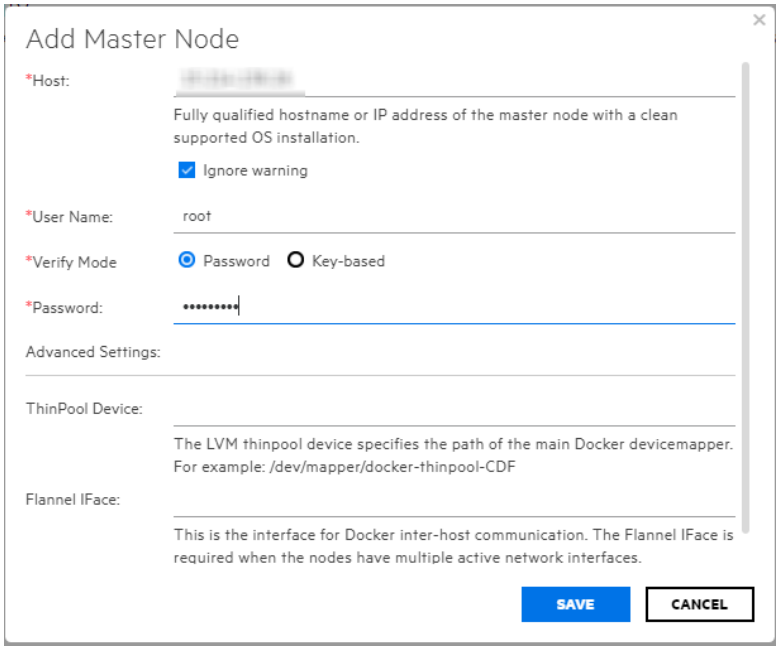
the Master Node hostname if the `--ha-virtual-ip` parameter was not specified during CDF installation. Confirm the VIP is correct and then click **Next**.



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**.



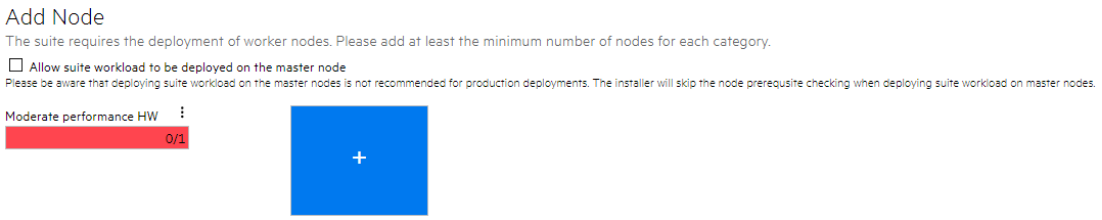
10. The installer prompts to add a number of Master Nodes depending on your selected deployment size. On the **Add Master Node** page, specify the details of your first Master Node and then click **Save**. Repeat for any additional Master Nodes.



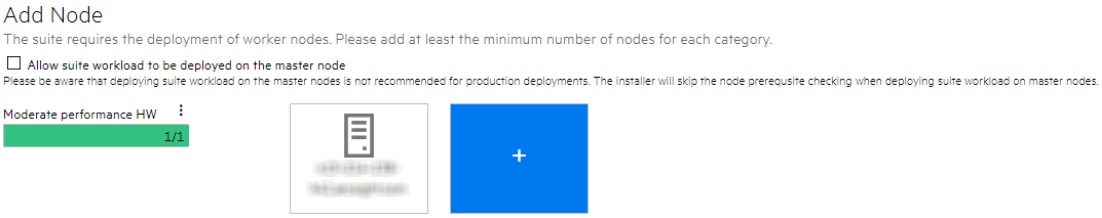
Master Node parameters include:

- **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.

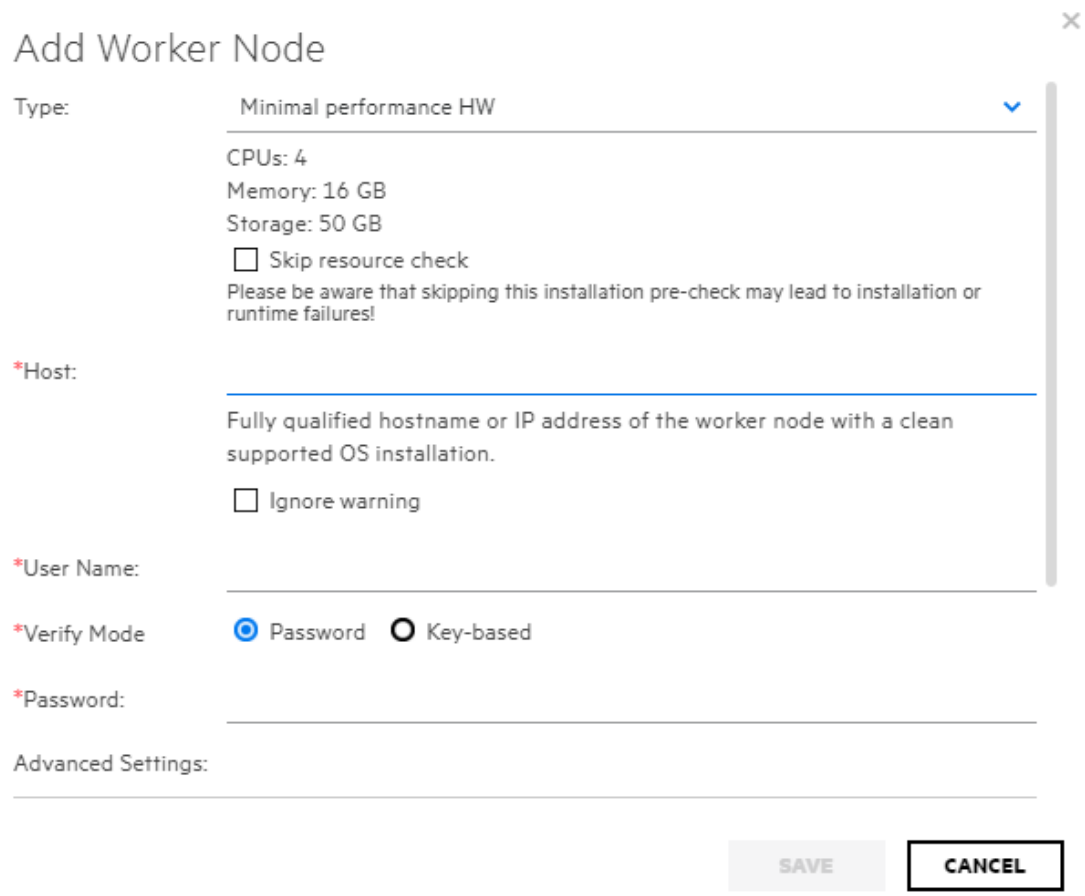


As you add Worker Nodes, each Node is then verified for system requirements. The node count progress bar on the **Add Node** page will progressively show the current number of verified Worker Nodes you have added. This progress will continue until the necessary count is met so the bar will turn from 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.

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

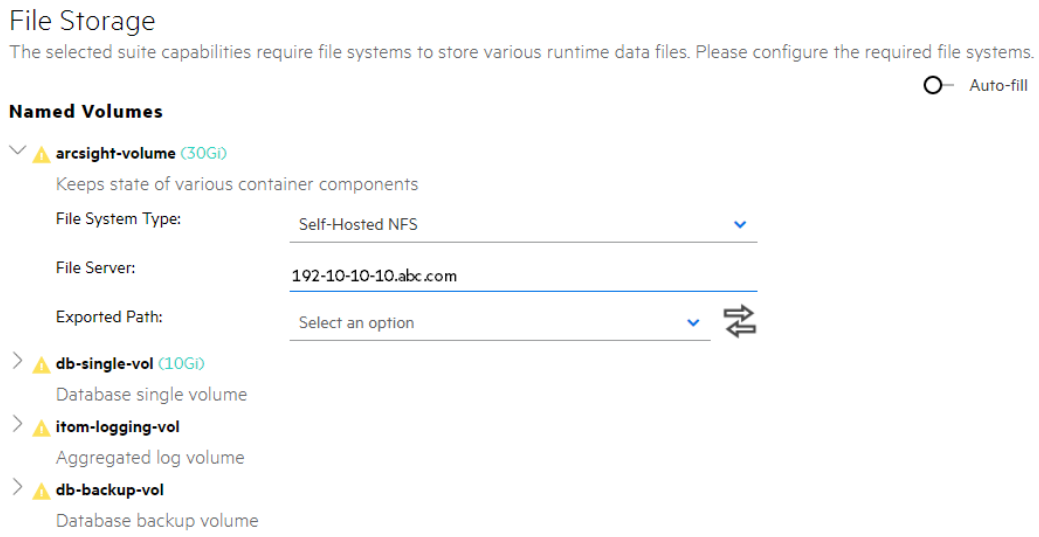
Once all the required Worker Nodes have been added, click **Next**.

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.

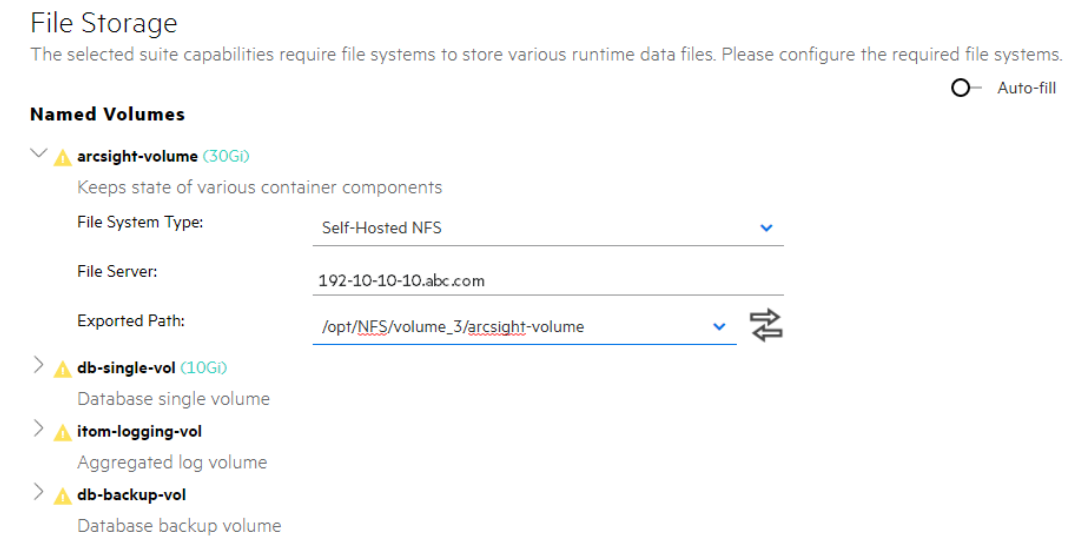


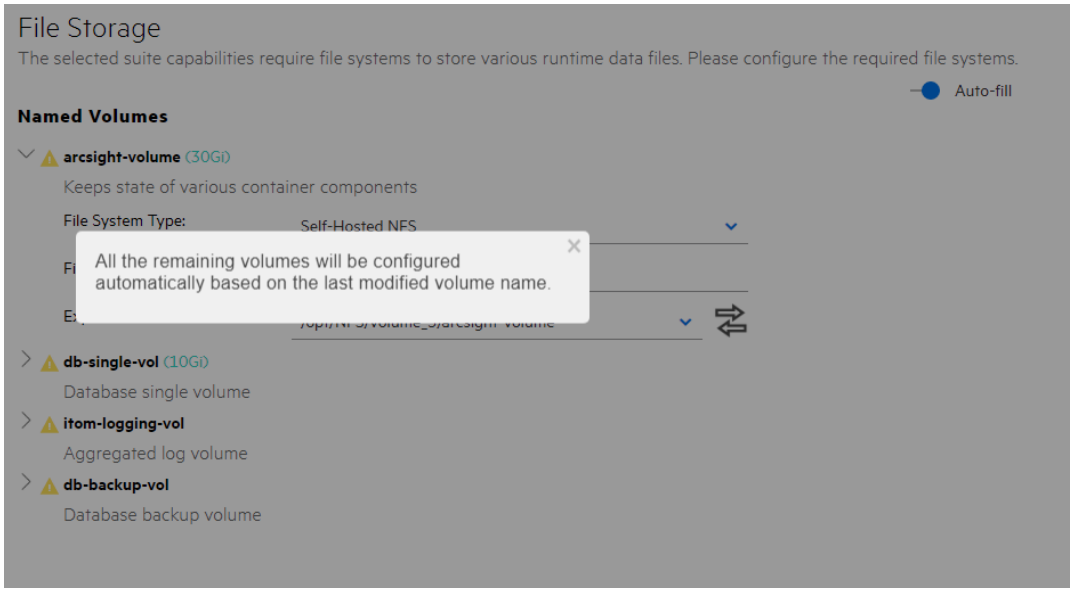
Note: If the NFS server is setup as described in the table below, the **Auto-fill** feature can be applied. Otherwise, each value would need to be filled out individually.

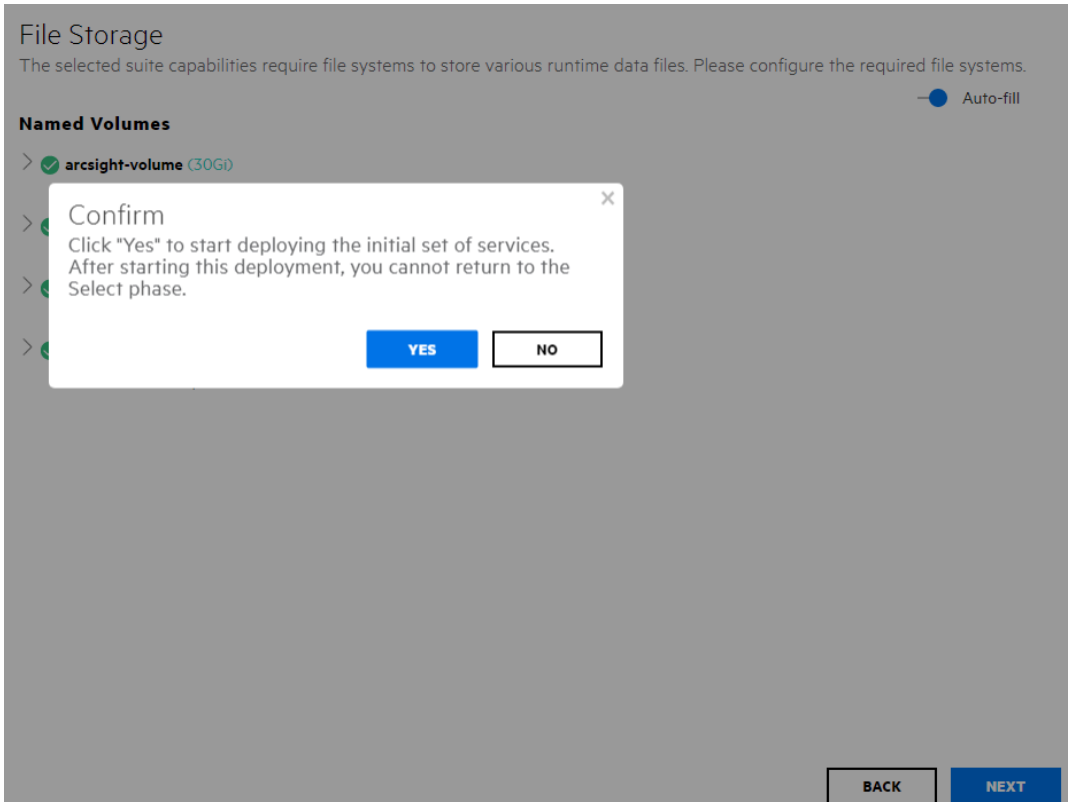
Note: A *Self-hosted NFS* refers to the external NFS that you prepared during the NFS server environment configuration, as outlined in the CDF Planning Guide. Always choose this value for **File System Type.**

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:



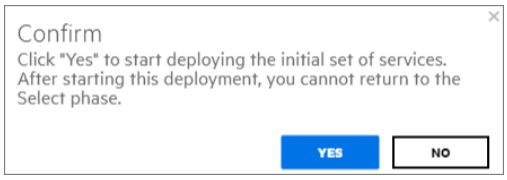




13. Click **Yes**.

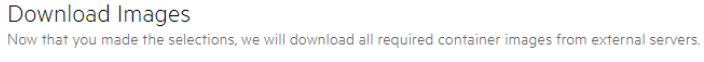
> **Warning:** After you click **Next**, the infrastructure implementation will be deployed. ***Please ensure that your infrastructure choices are adequate to your needs.*** An incorrect or insufficient configuration may require a reinstall of all capabilities.

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



Download Transformation Hub, Investigate and Core Images to the Local Docker Registry

By this point, the Transformation Hub, Investigate, and Analytics packages to be installed have already been downloaded from the Micro Focus software site, validated and uncompressed.



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:

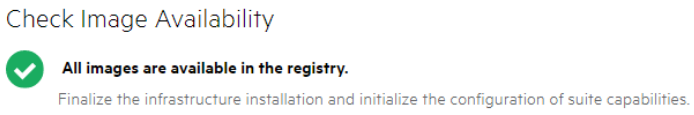
```
cd $k8s-home/scripts
./uploadimages.sh -u registry-admin -d $download_dir/transformationhub-3.2.0.xxx
./uploadimages.sh -u registry-admin -d $download_dir/analytics-3.1.0.xxx
./uploadimages.sh -u registry-admin -d $download_dir/investigate-3.1.0.xxx
```

Note: Prior running the image upload process by script, you will be prompted for the administrator password previously specified in the topic "Configure and Install the CDF Installer" on page 18.

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.



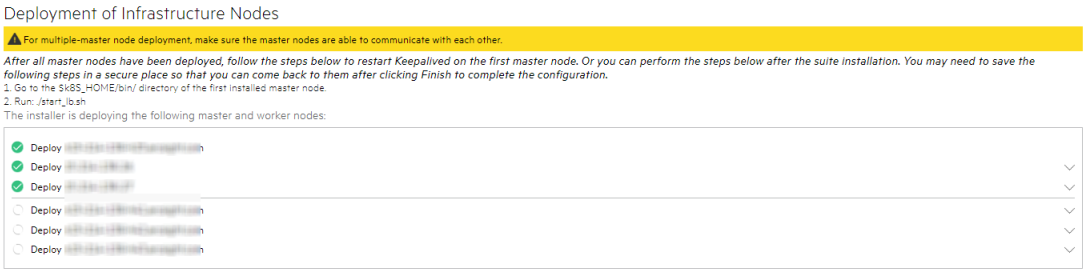
To verify that all images have been uploaded, return to the CDF Management Portal's Check Availability page and click **Check Image Availability Again**. All required component uploads are complete when the message displayed is: ***All images are available in the registry.***

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>
  ```

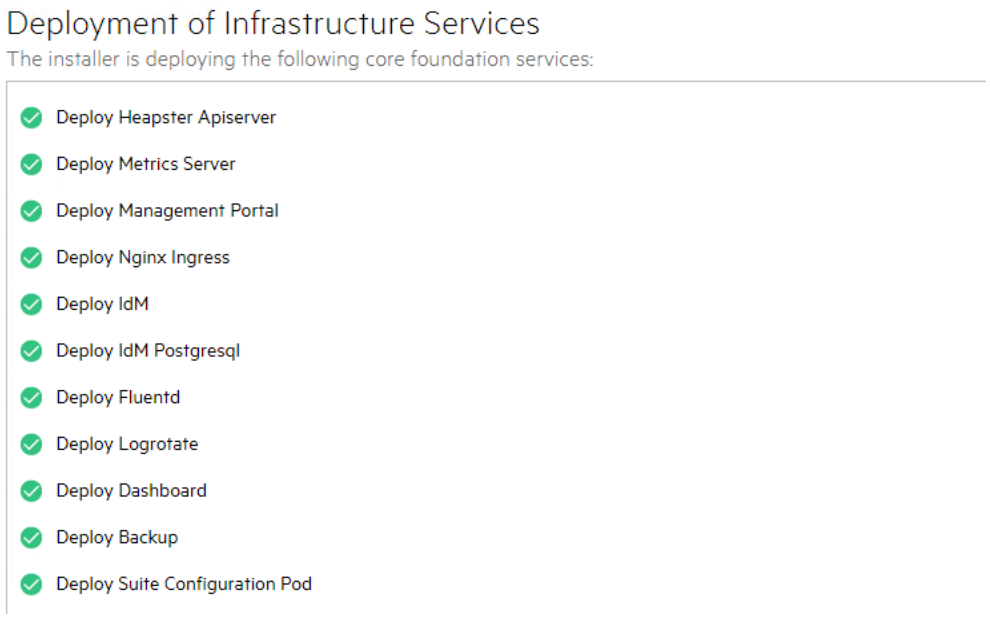
  - After installation has finished, the logs are copied to `$k8s-home/log/scripts/install`
- You can watch the status of deployment pods with the command:

  ```
  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:

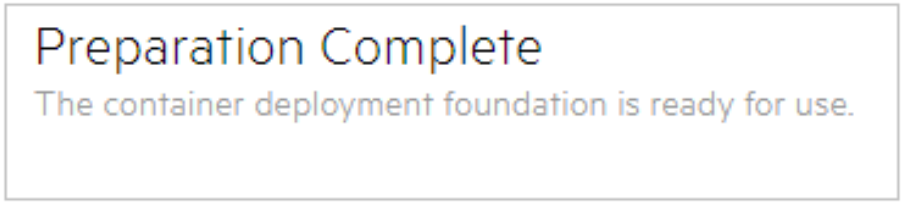
```
watch kubectl get pods --all-namespaces
```

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.

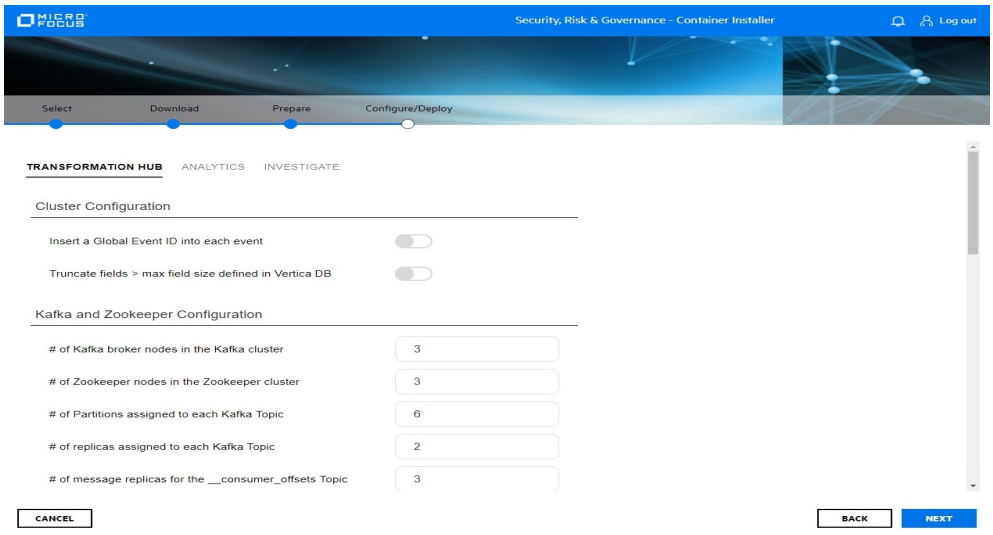


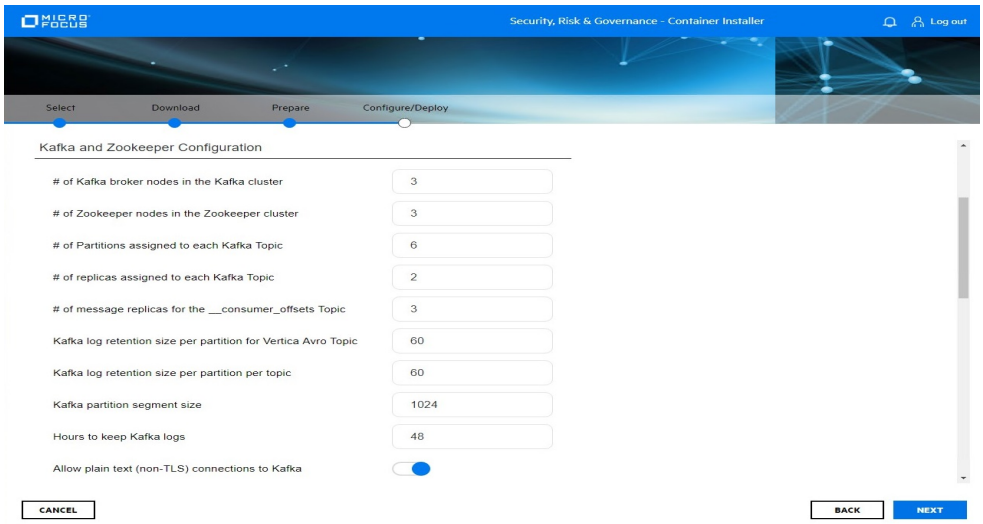
Click **Next** to configure the products and components of the deployment.

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.

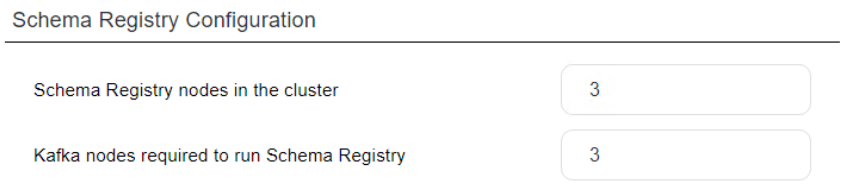
Input the following values into the Worker Nodes.

- # 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 Vertica.
# of partitions must be changed after deployment has been successfully completed. For more information, please see "Updating Topic Partition Number" on page 44

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 Vertica Avro Topic
  - Input the calculated th-arcsight-avro topic partition size
  - This value is exclusive for Vertica 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 Investigate

Investigate is now ready to be configured. Investigate 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.

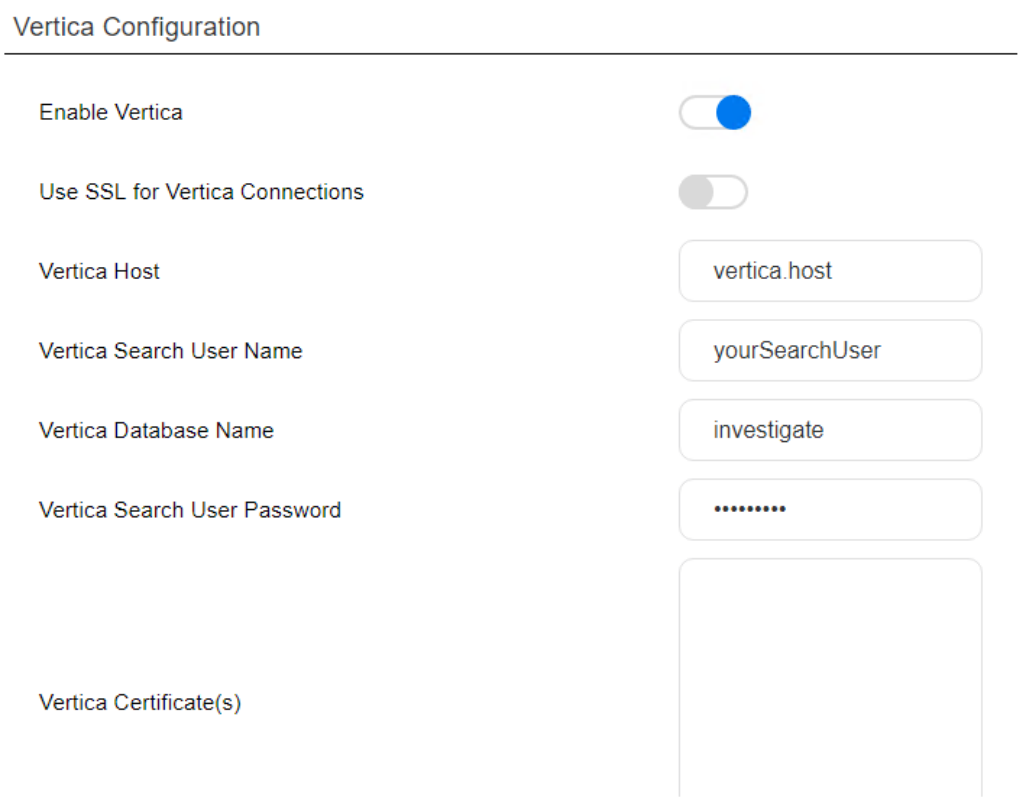
In order to Configure and Deploy investigate, perform the following procedures:

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

Setting Up Vertica Database Connection

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

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



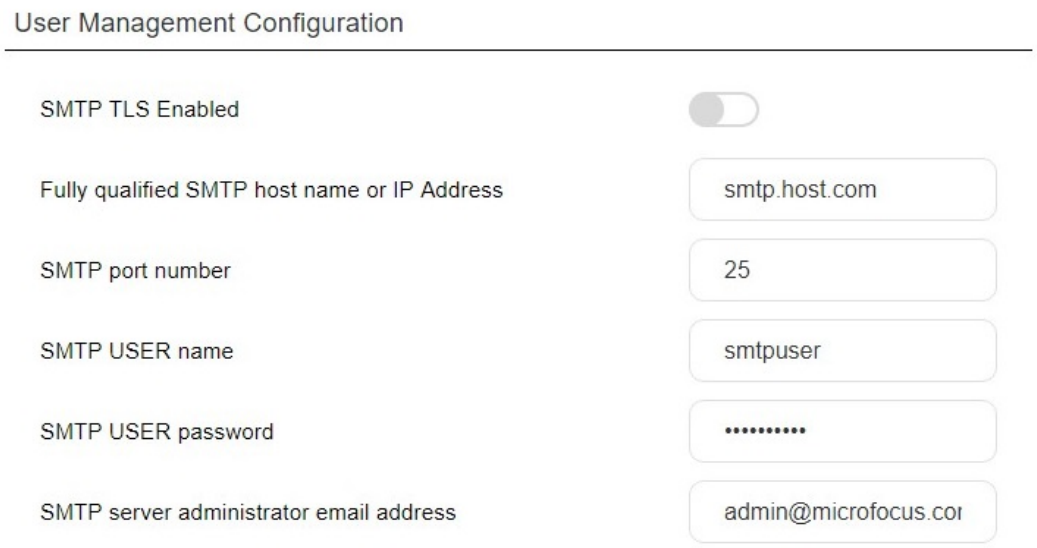
Under Vertica Configuration, provide the following information to update the Vertica connection parameters:

- **Vertica host name:** You can specify any Vertica node IP address, but only specify one address (**Use IP address only**).
- **Vertica search USER name:** The search user name that you defined when you installed Vertica.
- **Vertica database name:** Investigate.
- **Vertica search USER password:** The search user password that you created when you installed Vertica

Setup SMTP Server

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

In order to setup the SMTP Server, scroll down to **User Management Configuration**

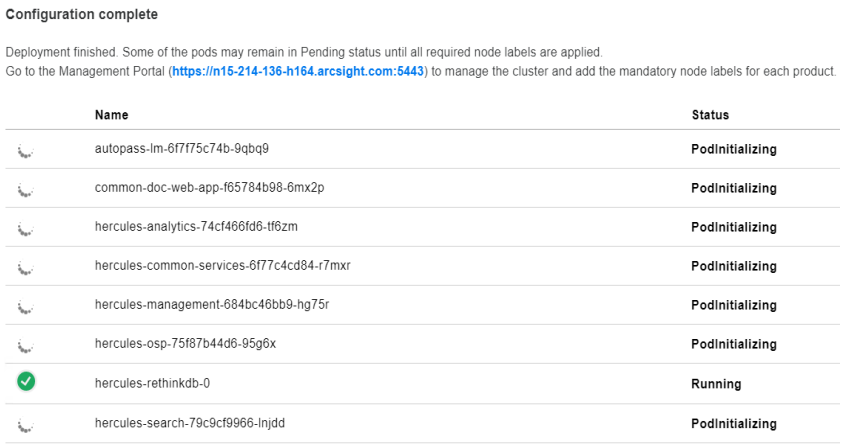


Input the following information, and click **SAVE:**

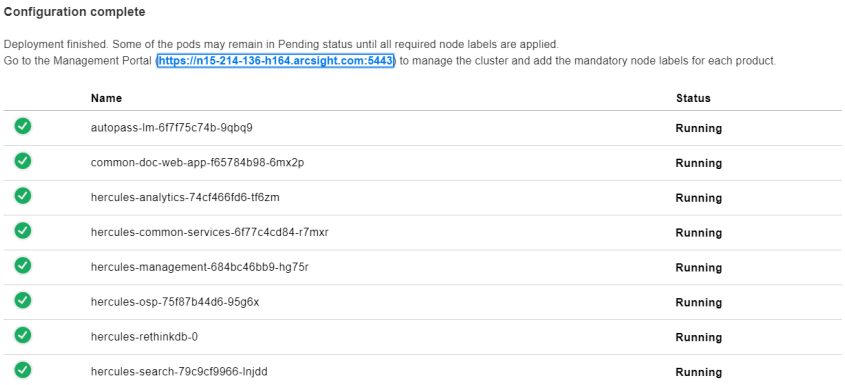
- SMTP TLS Enabled
- Fully qualified SMTP host name or IP Address
- SMTP port number
- SMTP USER name
- SMTP USER password
- SMTP server administrator email address
- User session timeout in seconds

Pre-deployment Configuration Completion

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

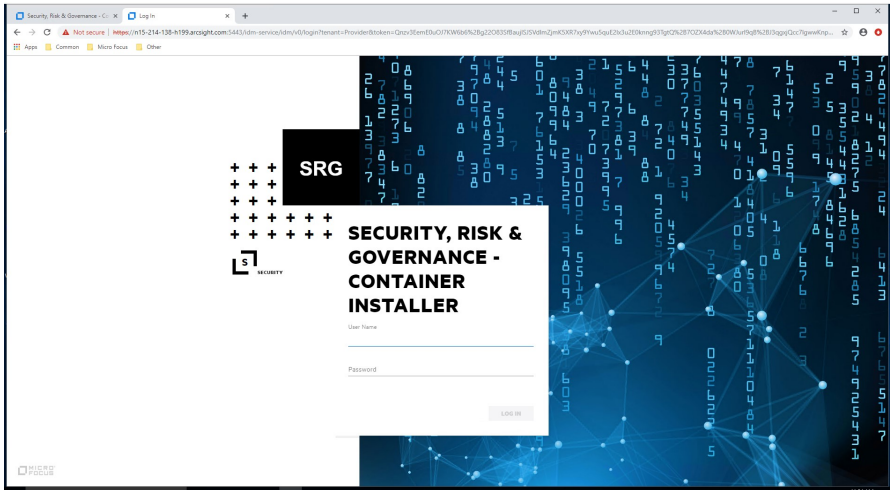


Pod status can be monitored on this page after the worker nodes have been labeled, and images have deployed.



To Continue Setup from Management Portal

Go to Management Portal by either clicking the Management Portal link displayed on the Configuration complete page, or browse to `https://<Master_FQDN>:5443` or `https://<Virtualhost_FQDN>:5443` if you deployed in multi-master mode.



Input the following information, and then click **LOG IN**

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

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:

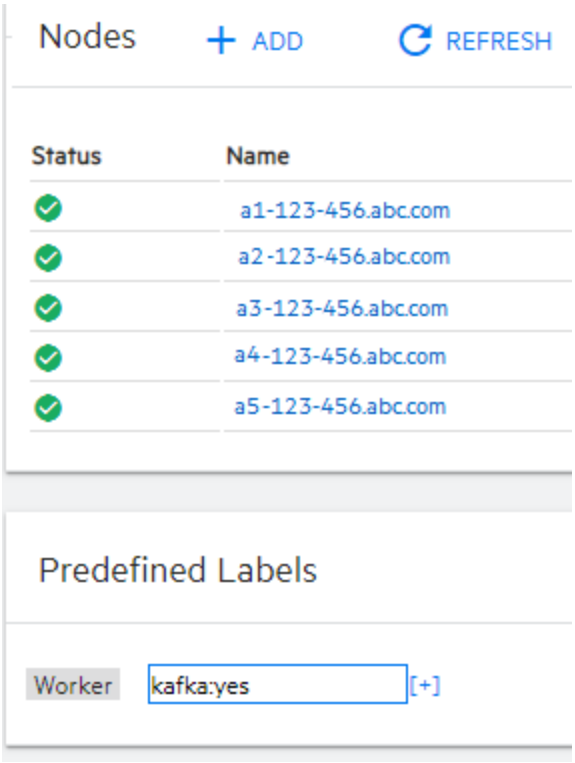
1. Login to Management Portal by clicking the link on the **Deployment status** (Configuration complete) page or browse to https://<ha-address>:5443, where:

- **Ha-address:** FQDN of the Virtual IP address provided during installation (--ha-virtual-ip) (or, for a single-master installation, the IP address of the master node).
- **User Name:** admin
- **Password:** Password

2. Go to **Administration -> 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.

```
analytics:yes
zk:yes
kafka:yes
th-processing:yes
th-platform:yes
```

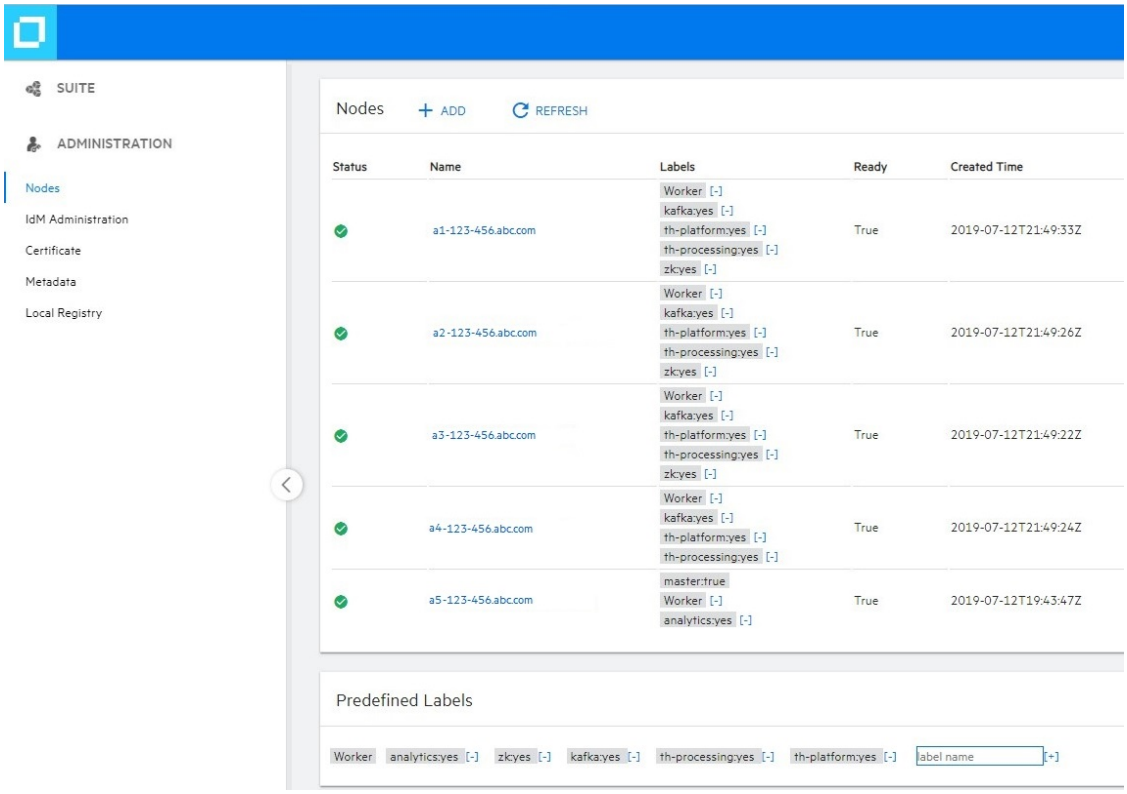


4. Drag and drop a new label from the **Predefined Labels** area to each of the Worker Nodes, based on your workload sharing configuration. This will apply the selected label to the Node.

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

For Kafka and ZooKeeper, 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.

For the Investigate node, drag the `analytics: yes` label to the Investigate node.



Once the Nodes have been properly labeled, the Transformation Hub services status will change from a ***Pending*** to a ***Running*** state. You can monitor the process by running the following command on the Initial Master Node:

```
kubectl get pods --all-namespaces -o wide
```

Check Deployment Status

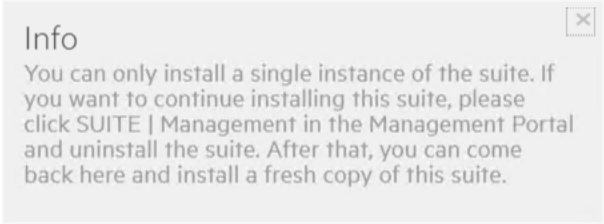
- Pods that have not been labeled will remain in a Pending state until labeled.
- For a pod that is not in Running state, you can find out more details on the pod by running the following command:

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

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

Note: If the following error is displayed when attempting to log in to the CDF Management Portal on port 3000, this typically means that the CDF installation process has completed, port 3000 is no longer required, and has been closed. Instead of port 3000, log in to the Management Portal on

port 5443.



Check Cluster Status

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

Note: You may need to wait 10 minutes or more for all pods to be in a *Running* or *Completed* state.

1. Log into the Initial Master Node.
2. Run the command:

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

Review the output to determine the status of all pods.

Post-Deployment Configuration

Depending on your architecture, after deployment, you may need to adjust some of the post-deployment configuration properties in order for Transformation Hub to function correctly.

If you plan to manage Transformation Hub with ArcMC, then you will need to adjust some settings in the post-deployment stage with your ArcMC details. Whether you need to adjust other properties during post-configuration will depend on the specifics of your implementation.

For a more detailed discussion of post-deployment configuration settings, see the Transformation Hub Administrator's Guide.

To configure post-deployment settings:

1. Browse to the Management Portal at `https://<master_FQDN>:5443` or : `https://<Virtualhost_FQDN>:5443` if you deployed in multi-master mode
   - *User Name:* admin
   - *Password:* Password provided during installation
2. Navigate to suite options: **Suite > Management** .
3. Click the **...** (Browse) icon to the right of the main window.

4. From the drop-down, click **Reconfigure**. The post-deployment settings page is displayed.
5. Select **TRANSFORMATION HUB**, and scroll down to **Stream Processors and Routers**
6. Under **Stream Processors and Routers**, input the appropriate value for **# of CEF-to-Avro Stream Processors instances to start.**

   Note: 15 was tested as the appropriate value for 120 partitions on a 3 node TH cluster.
7. For configuration management of Transformation Hub with ArcMC, see Configuring ArcMC to Manage Transformation Hub
8. Click **SAVE.**

Web services in the cluster will be restarted (in a rolling manner) across the cluster nodes.

Additional Steps

Updating CDF Hard Eviction Policy

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.

Updating Topic Partition Number

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

Perform the following steps to update the topic partition number from the master node 1:

1. Run the following commands :

- Find the server ($server), running th-kafka-0:

```
kubectl get pods --all-namespaces -o wide | grep th-kafka-0 | awk '{print $8}'
```

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

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

- Update th-arcsight-avro topic partition number:

```
kubectl exec -n $NAMESPACES th-kafka-0 -- /usr/bin/kafka-topics --zookeeper $server: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 $NAMESPACES th-kafka-0 -- /usr/bin/kafka-topics --zookeeper $server: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.`**

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.

Management Center: Configuring Transformation Hub

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.

Chapter 6: Complete Vertica Setup

Follow the steps below to complete the Vertica Setup.

1. Create the schema:

   ```
   ./vertica_installer create-schema
   ```

2. 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 "Configuring Vertica SSL " on page 50.

3. Create the Kafka scheduler:

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

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

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

4. Check the Vertica status:

   ```
   ./vertica_installer status
   ```

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

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

Vertica Installer Options

You can specify the following options when installing Vertica. To specify an option, type `./vertica_installer <Option_Name>`.

Option	Description
install	Installs the Vertica database
uninstall	Uninstalls the Vertica database and deletes data and users
create-schema	Creates the database schema for Investigate
delete-schema	Deletes the Investigate database schema

Option	Description
start-db	Starts the Vertica database with the `dba_password` specified in `vertica_credentials.properties`
stop-db	Stops the Vertica database
status	Prints the Vertica cluster status

Kafka Scheduler Options

You can specify the following options when installing the Kafka scheduler. To specify an option, type `./kafka_scheduler <Option_Name>`.

Option	Description
update	Updates the scheduler
start	Starts the scheduler and begins copying data from all registered Kafka brokers
stop	Stops the scheduler and ends copying data from all registered Kafka brokers
delete	Deletes all registered Kafka instances from the scheduler
status	Prints the following information and log status for a running or stopped scheduler: • Current Kafka cluster assigned to the scheduler • Name and Vertica host where the active scheduler is running • Name, Vertica host, and process ID of every running scheduler (active or backup)
events	Prints event copy progress for the scheduler
messages	Prints scheduler messages

Chapter 7: Setting FIPS on Vertica

In order to enable FIPS mode in Investigate 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
```

2. 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

Enabling FIPS in Nginx

No user action is required to enable FIPS for Nginx. The Nginx docker container is FIPS enabled by default. The FIPS enabled Nginx server will accept TLS 1.2 connections using FIPS compliant Cipher Suites.

Chapter 8: Configuring Vertica SSL

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

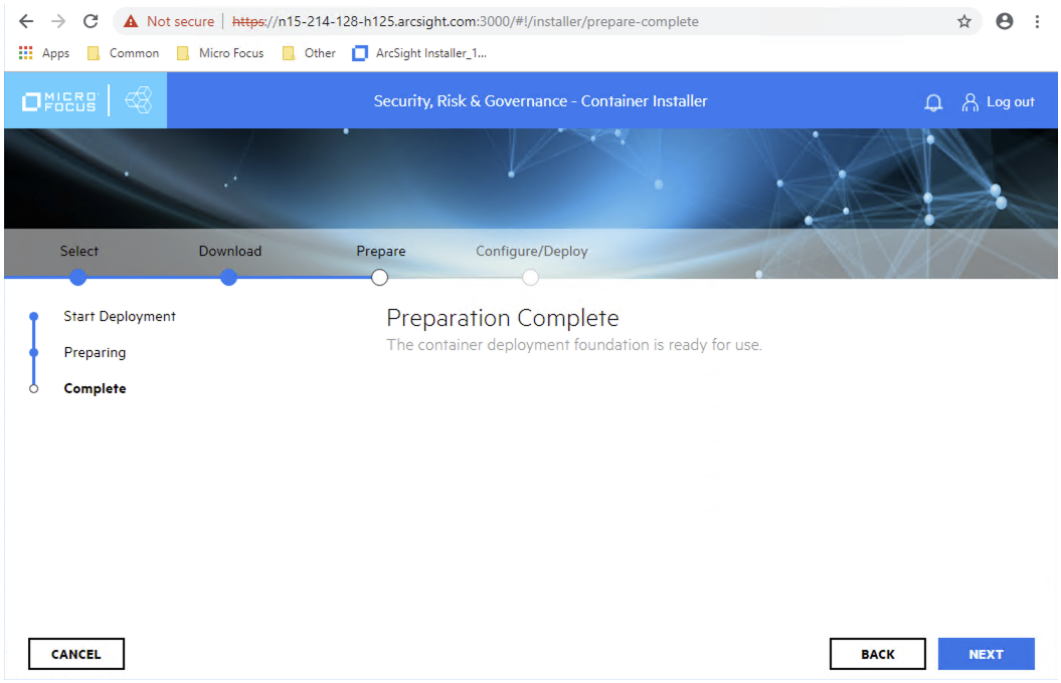
Change the key files permissions

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

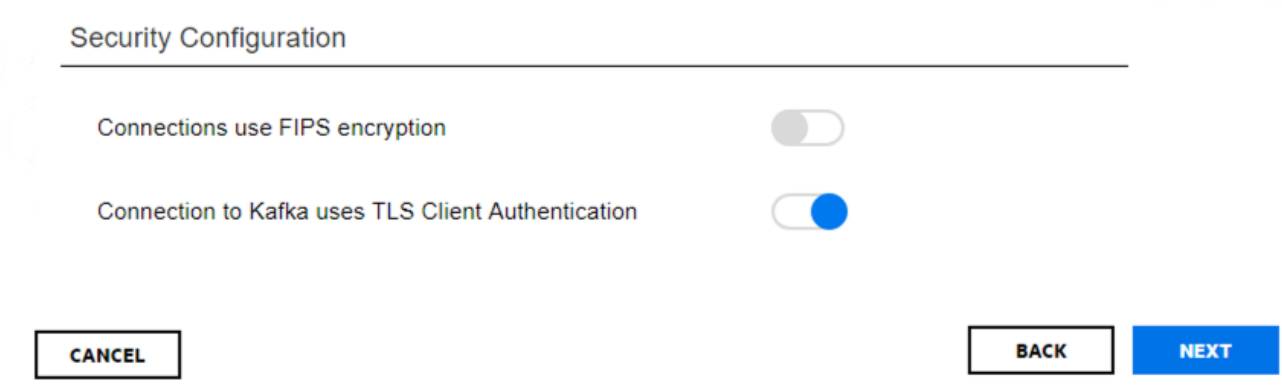
secret/nginx-default-secret configured

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 node 1 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
```

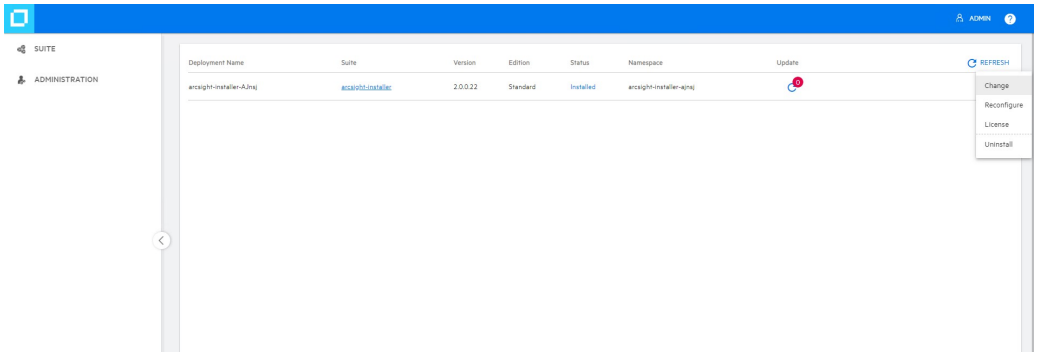
Creating Scheduler with SSL Enabled

To create Scheduler with SSL enabled, run the following command:

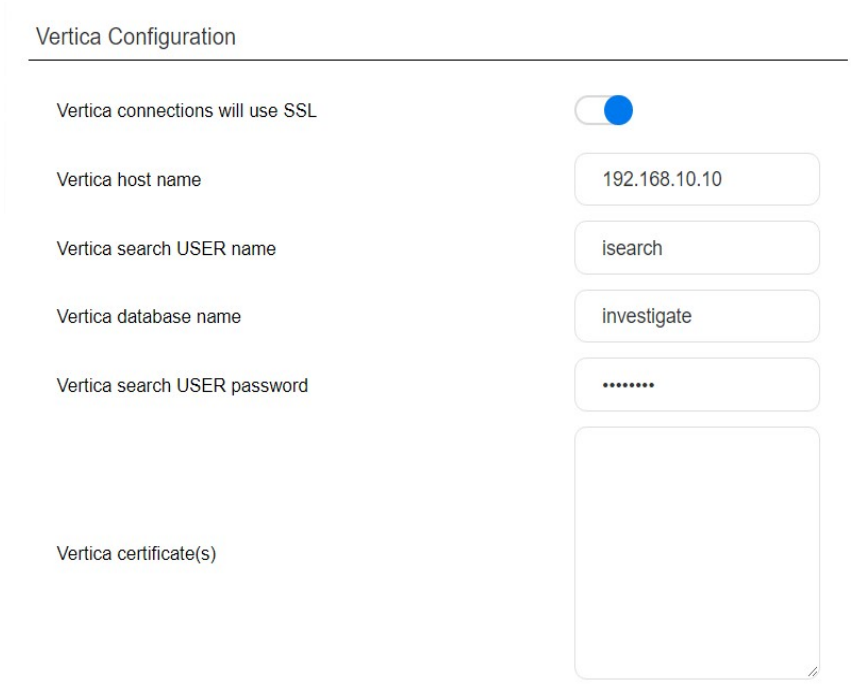
```
$vertica-install-DIR/kafka_scheduler create <WorkerNode1>:9093
```

Setting up Investigate with SSL Enabled

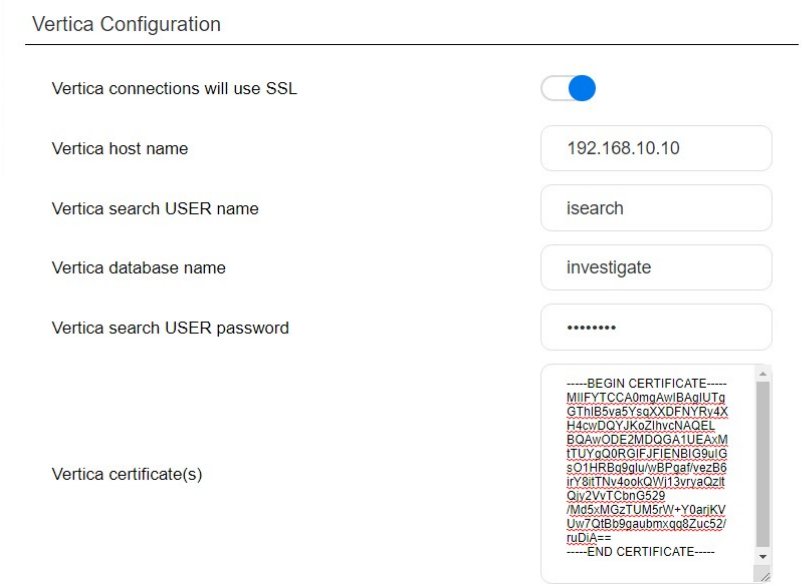
1. Browse to https://<virtual-server-FQDN>:5443, if it is a multiple master, or https://<master-FQDN>:5443, if it is a single master.
2. Navigate to suite options: **Suite > Management**
3. Click the **...** icon under **REFRESH** and Select **Reconfigure.** A new tab will be opened.



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

Chapter 9: Configuring ArcSight Investigate and Components

After you deploy Investigate, use the **Configuration** page of the ArcSight Installer to configure the product. After you change a product setting, Investigate restarts.

Creating the System Administrator

When you log in to Investigate for the first time, you must create the system administrator account. Investigate assigns the `system admin` role to this account.

To create the system administrator account:

1. If you deployed Investigate in single-master mode, open `https://<Master_FQDN>`.

   If you deployed Investigate in multi-master mode, open `https://<Virtualhost_FQDN>`.
2. On the welcome page, enter the name, email, and password information for the system administrator account and then click **Create System Admin.**
3. On the login page, enter the email and password for the system administrator account.

Updating the Vertica Database Connection

Use the ArcSight Installer to update the connection to the Vertica database. Each time you change the connection, the search engine container restarts.

Note: The Vertica database name was defined when you created the schema. You cannot change the name.

To configure the Vertica database connection:

1. Log in to the ArcSight Installer:

   `https://<Master_FQDN>:5443` or `https://<Virtualhost_FQDN>:5443` if you deployed in multi-master mode.
2. Navigate to suite options: **Suite > Management**
3. Click the 3 dots at the end of the selected investigate suite and Select **Reconfigure.**
4. Select **ANALYTICS**, and scroll down to **Vertica Configuration**
5. Under Vertica Configuration, provide the following information to update the Vertica connection parameters:
   - **Vertica host name:** You can specify any Vertica node IP address, but only specify one address.
   - **Vertica search USER name:** The search user name that you defined when you installed Vertica.
   - **Vertica database name:** The name is hard coded to Investigate. You should not change it.
   - **Vertica search USER password:** The search user password that you created when you installed Vertica

Updating the SMTP Server

Update access to your SMTP server to enable users that you create in Investigate to receive notification emails.

To update the SMTP server:

1. Log in to the ArcSight Installer:

   `https://<Master_FQDN>:5443` or `https://<Virtualhost_FQDN>:5443` if you deployed in multi-master mode.
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 **User Management Configuration**
5. Input the following information, and then click **SAVE**:
   - 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

Configuring Search Settings

You can configure the following properties in ArcSight Installer:

- Search query timeout

  Search queries might take a long time and impact performance. You can limit the amount of time that a search query runs. The default search query timeout is 60 minutes.

To configure session and search settings:

1. Log in to the ArcSight Installer:
   `https://<Master_FQDN>:5443` or `https://<Virtualhost_FQDN>:5443` if you deployed in multi-master mode.
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 under **Cluster Configuration**, input the appropriate value for **Search Query Timeout in minutes**.

Chapter 10: Enabling the Data Retention Policy on the Vertica Cluster

When Vertica storage approaches usage limits, storage needs to be cleaned up for new events. Data retention script purges old data to reclaim storage.

Note: Storage usage limits are defined by the User.

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).

The default data retention period is 90 days. 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: Vertica 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. Run the following command to check disk usage:

   ```
   cd $vertica-install-DIR
   ./vertica_installer status
   Check the disk_space_free_percent
   ```

2. Back up Vertica data.

   For more information, see "Backing Up the Vertica Database" on page 68.

3. Run the following commands:

   ```
   cd $vertica-install-DIR/config
   vi vertica_user.properties
   ```

   Uncomment `#db_retention_days=90`

4. Verify the number of days of data in the Vertica database:

   ```
   cd $vertica-install-DIR/script
   ./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.

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

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

To purge Vertica data:

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)
   ------------------------------------------------------------------------------------
   ```

Chapter 11: Backing Up and Restoring the Vertica Database

You should back up and restore the Vertica database before you upgrade Vertica or before you add or remove a Vertica 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 Vertica. 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 Vertica node should have its own backup host.
- Use one directory on each Vertica node to store successive backups.
- You can save backups to the local folder on the Vertica node 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 Vertica operation to estimate the required storage space for the Vertica 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 Vertica 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 Vertica node 1 to the backup hosts, as well as from the restored Vertica node 1.

To set up password-less SSH:

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

   `$db_admin` is the administrator for the Vertica cluster.
3. Ensure that `$db_admin` has write permission to the dedicated directory where you will store the backup.
4. Log in to Vertica node 1 as `root`.
5. Become the Vertica database administrator:

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

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

Preparing Backup Configuration File

Vertica includes sample configuration files that you can copy, edit, and deploy for your various ***vbr*** tasks. Vertica 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 - idbadmin

# cp /opt/vertica/share/vbr/example_configs/backup_restore_full_external.ini
vertica_backup.ini

# vi vertica_backup.ini
```

Note: You must save a copy of `vertica_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 vertica_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 = $your_db_admin
```

Backing Up the Vertica Database

The **$db_admin** user must perform the backup from the Vertica node 1 of the cluster.

Note: vbr Command Reference.

To back up the database:

1. Stop Vertica scheduler

   Login Vertica node 1 as **root**

   ```
   # cd $vertica-install-DIR
   # ./kafka_scheduler stop
   ```

2. Initialize backup location

   ```
   # su - $db_admin
   # vbr -t init --config-file vertica_backup.ini
   ```

   Initializing backup locations.

   Backup locations initialized.

3. Back up Vertica data:

   ```
   # vbr -t backup -c vertica_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 Vertica 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 vertica_backup.ini
```

Verifying the Integrity of the Backup

Use the `full-check` option to verify the integrity of the Vertica 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 vertica_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 vertica_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 vertica_backup.ini --archive 20190909_
010826

Enter vertica password:

Removing restore points: 20190909_010826

Remove complete!
```

Restoring Vertica Data

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

- You can only restore backups to the same version of Vertica 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 Vertica 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 Vertica nodes in the target cluster are running.
- All Vertica node names in the target cluster match the names from the backup.

Restoring the Vertica Database

The `$db_admin` user must restore from the Vertica node 1 of the cluster.

To set up password-less SSH:

1. Log in to the target Vertica node 1 as root.

2. Become the Vertica database administrator:

```
# su -l $db_admin
```

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

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

To restore the database:

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

   ```
   # cd $vertica-install-DIR
   # ./vertica_installer stop-db
   ```
3. Become the `$db_admin` user:

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

   ```
   # vbr --task restore --config-file vertica_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
# ./vertica_installer start-db
```

The output should be similar to the following:

```
Starting nodes:
v_investigate_node0001 (127.0.0.1)
Starting Vertica on all nodes. Please wait, databases with a large catalog
may take a while to initialize.
Node Status: v_investigate_node0001: (DOWN)
Node Status: v_investigate_node0001: (DOWN)
Node Status: v_investigate_node0001: (DOWN)
Node Status: v_investigate_node0001: (DOWN)
Node Status: v_investigate_node0001: (UP)
Database investigate started successfully
```

7. Start the Kafka scheduler:

```
# ./kafka_scheduler start
```

Chapter 12: Vertica upgrade

Changes in Investigate 3.0.0

- A new column, Persisted Time (PT), has been added to the events table in the Vertica database. PT records when each event is inserted into the events table.
- A new column, Normalized Event Time (NET) has been added to the events table in the Vertica database. This solves the ROS container issues during ingestion. When an event comes in with deviceReceiptTime outside of the default threshold which is 7 days in past and 1 day in future, NET will be the current Vertica system time, otherwise it will be the same as deviceReceiptTime. For most of the events coming in normal time range, NET will be same as DRT. Performance is improved by partitioning and searching on NET.
- Number of active partitions increased from four to ten to improve ingestion performance.
- Investigate searches are based on normalizedEventTime for better performance. Since NET will be the same as DRT for most events, this should not be too different than directly searching on DRT. For cases where DRT is different than NET, you can perform a search with an explicit DRT search term or by searching and then using the filter feature to refine the DRT range you want.

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.

Vertica upgrade steps

- On the Vertica cluster node 1 server, create a folder for the new Investigate Vertica database installer script:

```
mkdir $new-vertica-install-DIR
```

Note: `$new-vertica-install-DIR` should not be under /root.

- Copy arcsight-vertica-installer_3.1.0-3.tar.gz and arcsight-vertica-installer_3.1.0-3.tar.gz.md5 to `$new-vertica-install-DIR.`

md5sum arcsight-vertica-installer_3.1.0-3.tar.gz

The output check sum should match the number recorded in arcsight-vertica-installer_3.1.0-3.tar.gz.md5

- Untar arcsight-vertica-installer_3.1.0-3.tar.gz.

```
tar xvfz arcsight-vertica-installer_3.1.0-3.tar.gz
```

- Run the upgrade command in order

Note: The command execution can't be re-ran.

```
./investigate_upgrade
```

Usage:

Execute the following commands in this order

1. `./investigate_upgrade -c upgrade-investigate`
2. `./investigate_upgrade -c update-configuration`

Options:

-h, --help show this help message and exit

```
-c COMMAND, --command=COMMAND
```

[REQUIRED] specify upgrade command:

['upgrade-investigate', 'update-configuration',

'upgrade-vertica-rpm']

Run as an example: `./investigate_upgrade -c upgrade-investigate`

Upgrade related changes cannot be rolled back, do you want to continue with the upgrade (Y/N): y

Starting upgrade...

********** Start of Investigate Upgrade **********

Enter previous installed location (/opt/install-vertica):/opt/installer

Running Pre-Upgrade checks

Checking all Vertica nodes are UP

All Vertica nodes are UP

Replacing files in installed location

Upgrading script and config files.

Creating backup directory: /opt/installer/oldVersion

Backing up: /opt/installer/vertica_installer

Backing up: /opt/installer/resources

Backing up: /opt/installer/scripts

Backing up: /opt/installer/data

Backing up: /opt/installer/upgrade

Backing up: /opt/installer/lib

Backing up: /opt/installer/vertica.properties

Backing up: /opt/installer/kafka_scheduler

Backing up: /opt/installer/sched_ssl_setup

Backing up: /opt/installer/vertica_ssl_setup

Backing up: /opt/installer/vertica_upgrade.py

Backing up: /opt/installer/investigate_upgrade

Backing up: /opt/installer/copyright.txt

Upgrading: /opt/installer/vertica_installer

Upgrading: /opt/installer/resources

Upgrading: /opt/installer/scripts

Upgrading: /opt/installer/data

Upgrading: /opt/installer/upgrade

Upgrading: /opt/installer/lib

Upgrading: /opt/installer/vertica.properties

Upgrading: /opt/installer/kafka_scheduler

Upgrading: /opt/installer/sched_ssl_setup

Upgrading: /opt/installer/vertica_ssl_setup

Upgrading: /opt/installer/vertica_upgrade.py

Upgrading: /opt/installer/investigate_upgrade

Upgrading: /opt/installer/copyright.txt

Upgrading: /opt/installer/vertica-upgrade.log

********* Start of Investigate Upgrade to 3.10.0 *********

Pre Upgrade check for 3.10.0

Current Investigate version is: 3.00.0

Investigate will be upgraded to 3.10.0

Create data quality table and create data quality crontab ...

data quality table has been created successfully.

********** Investigate Upgraded Complete. Version is 3.10.0 **********

Run as an example: `./investigate_upgrade -c update-configuration`

Upgrade related changes cannot be rolled back, do you want to continue with the upgrade (Y/N): y

Starting upgrade...

********** Start of Configuration Updade **********

Enter previous installed location (/opt/install-vertica):/opt/installer

Running Pre-Upgrade checks

Checking all Vertica nodes are UP

All Vertica nodes are UP

Grant general resource pool to search user

Restart Kafka scheduler,

```
cd $vertica-install-DIR
./kafka_scheduler start
SSL/TLS mode is disabled
```

Terminating all running scheduler processes for schema: [investigation_scheduler]

```
scheduler instance(s) deleted for 192.168.100.100
scheduler instance(s) added for 192.168.100.100
```

Note: If Investigate has not been upgraded, continue to upgrade Investigate. If Investigate has been upgraded, resume normal operations.

Chapter 13: Backing Up and Restoring Investigate Management and Search Datastores

Micro Focus recommends that you use a backup location that is not under the `/opt/arcsight` directory. Use a local folder on the system or a remote location.

This procedure uses the `/opt/investigate/backup` directory as an example.

To back up the data stores:

1. To prohibit database access, undeploy Investigate.

   For information about undeploying Investigate, see .
2. SSH to the Kubernetes cluster master node 1.
3. Run the following commands:

   ```
   # cd /opt/arcsight/volumes/investigate/
   # mkdir -p /opt/investigate/backup
   # cp –R * /opt/investigate/backup
   # diff -r -s /opt/investigate/backup/mgmt
   /opt/arcsight/volumes/investigate/mgmt
   # diff -r -s /opt/investigate/backup/search
   /opt/arcsight/volumes/investigate/search
   ```

   If you do not receive a message that states that the files are identical, repeat the commands.
4. Redeploy Investigate to resume operations.
5. Before you resume Investigate operations, ensure that the pods are in Running status:

   ```
   # kubectl get pods --all-namespaces | grep investigate
   ```

Restoring Investigate Management and Search Datastores

When restoring the Investigate management and search datastores, retain the original directory structure under `/opt/arcsight/volumes/investigate/`.

The management datastore will be restored to the `/opt/arcsight/volumes/investigate/mgmt/db/` directory. The search datastore will be restored to the `/opt/arcsight/volumes/investigate/search` directory.

To restore the datastores:

1. Ensure that you have a valid backup of the datastores.

   For more information, see Backing Up and Restoring Investigate Management and Search Datastores.

2. To prohibit access to the database, undeploy Investigate.

   For information about undeploying Investigate, see .

3. SSH to the Kubernetes master node, and then run the following commands:

   ```
   # cd /opt/investigate/backup
   ```

   ```
   # cp –R search/* /opt/arcsight/volumes/investigate/search
   ```

   Reply `yes` to overwrite files and folders.

   ```
   # cd /opt/arcsight/volumes/investigate/mgmt/db/
   ```

   ```
   # rm - rf h2.lock.db
   ```

   ```
   # cp /opt/investigate/backup/mgmt/db/h2.mv.db .
   ```

   Reply `yes` to overwrite files and folders.

   ```
   # diff -r -s /opt/arcsight/volumes/investigate/mgmt/db/h2.mv.db
   /opt/investigate/backup/mgmt/db/h2.mv.db
   ```

   ```
   # diff -r -s /opt/investigate/backup/search
   /opt/arcsight/volumes/investigate/search
   ```

   You should receive a message stating that all files are identical. If they are not identical, repeat the procedure.

4. Change the permission of the Investigate directory:

   ```
   # chown 1999:1999 -R /opt/arcsight/volumes/investigate/
   ```

5. Redeploy Investigate to resume operations.

6. Before you resume Investigate operations, ensure that the pods are in Running status:

   ```
   # kubectl get pods --all-namespaces | grep investigate
   ```

Chapter 14: Arcsight Suite Upgrade

The following topics are included in this chapter:

- Upgrade CDF and Upgrade Arcsight suites
- Upgrade CDF includes:
  - Upgrade CDF from 2019.05 to 2019.08
  - Upgrade CDF from 2019.08 to 2020.20
  - Both manual upgrade steps and auto-upgrade steps
- Upgrade Arcsight suites includes:
  - Upgrade Arcsight Investigate from 3.0.0 to 3.1.0
  - Upgrade Arcsight Transformation Hub from 3.1.0 to 3.2.0

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

Upgrading CDF 2019.05 to 2020.02

Investigate 3.1.0 is supported on CDF version 2020.02. As a result, users running an earlier version of CDF (version 2019.05) must upgrade to version 2020.02. The manual CDF upgrade process, which is run on each node in your environment, is described here.

Note: A properly-performed upgrade of CDF will not interrupt the flow of events from producers, through Transformation Hub, to the consumers, as long as the Transformation Hub environment includes more than 1 Kafka broker. No event data will be lost in this situation.

Upgrade is a lengthy process and should be run with a stable and reliable SSH connection. The complete process of upgrade includes an upgrade from CDF 2019.05 to CDF 2019.08, and then an upgrade from 2019.08 to 2020.02.

Note: The upgrade of a single-master environment to a multi-master (high availability/HA) environment is not supported by this process.

Prerequisites

- Docker and Kubernetes must be upgraded separately.
  - CDF upgrade from 2019.05 to 2019.08 does not include the upgrade of Docker or Kubernetes versions.

  - CDF upgrade from 2019.08 to 2020.02 does include the upgrade of Kubernetes from v1.13.5 to v1.15.5
  - CDF upgrade from 2019.08 to 2020.02 does include the upgrade of Docker from v18.09.2 to v19.03.5-3
- Verify that your environment meets the system requirements for a new cluster, as outlined in the *CDF Deployment Guide.*, including the following:
  - Linux OS version is RHEL/CentOS 7.5 or 7.6 and Kernel version is 7.4 v3.10.0-693.21.1.el7 (or above)
  - Make sure you have enough space on all cluster nodes. Default value for the pod eviction threshold is 85% of used space for the filesystem where the Kubernetes home directory is mounted (by default, `/opt/arcsight`). In addition, the cluster nodes should reserve 50 GB disk space for upgrades, preferably under a different location than the Kubernetes home directory.
- Verify that these two packages are installed on all nodes:

```
socat
container-selinux [version 2.74 or later]
```

Note: If these are not installed, then install each using the command:
`yum install <package-name>`.

Preparation

1. Ensure that you have the permission to reboot the cluster nodes. You may need to reboot the nodes during the upgrade.
2. To ensure that all nodes (master nodes and worker nodes) are in running status, run:
   `kubectl get nodes`
3. To ensure all core pods are running and all necessary checks are passed, run:
   `${K8S_HOME}/bin/kube-status.sh`
4. If you are using a non-root user to perform the manual upgrade, please verify that you have already configured your `sudo` permission.

Download the upgrade packages to each node

1. Download and copy the CDF 2019.08 and CDF 2020.02 upgrade packages to every node (master and worker) of the cluster into a download directory; for example `/tmp/upgrade-download`.
   Files:

   cdf-2020.02.00120-2.2.0.2.zip

   cdf-upgrade-2019.08.00134-2.2.0.2.zip

2. Create a `/tmp/upgrade-backup` directory with a minimum size of 30 GB on every node in your cluster. If you are a non-root user on the nodes inside the cluster, make sure you have permission to this directory with this command.

   ```
   mkdir /tmp/upgrade-backup
   ```

Manual Upgrade Process from CDF 2019.05 to 2019.08

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

1. Unzip the upgrade package on each node by running these commands:

   ```
   cd /tmp/upgrade-download
   unzip cdf-upgrade-2019.08.00134-2.2.0.2.zip
   ```

   Note: In the event that command execution fails for the below steps, please run them again.

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

   ```
   /tmp/upgrade-download/cdf-upgrade-2019.08.00134-2.2.0.2/upgrade.sh -i
   ```

3. On the initial master node, run the following commands to upgrade CDF components:

   ```
   /tmp/upgrade-download/cdf-upgrade-2019.08.00134-2.2.0.2/upgrade.sh -u
   ```

4. 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-upgrade-2019.08.00134-2.2.0.2/upgrade.sh -c
   ```

5. Verify the cluster status. First, check the CDF version on each node by running the command:

   ```
   cat ${K8S_HOME}/version.txt
   >> 2019.08.00134
   ```

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

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

7. Remove the 2019.08 upgrade directory from each node:

   ```
   rm -rf /tmp/upgrade-download/cdf-upgrade-2019.08.00134-2.2.0.2
   ```

8. On initial master node, run the following command to configure IDM pod affinity,

```
kubectl patch deployment idm -n core --patch '{ "spec": { "template": {
"spec": { "affinity": { "podAffinity": {
"preferedDuringSchedulingIgnoredDuringExecution": [ { "labelSelector": {
"matchExpressions": [ { "key": "app", "operator": "In", "values": [ "idm-app"
] } ] }, "topologyKey": "kubernetes.io/hostname" } ] } } } } } }'
```

9. Wait until IDM pods are up and running and use the following command:

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

Manual Upgrade Process from CDF 2019.08 to 2020.02

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

1. Unzip the upgrade package on each node by running these commands:

```
cd /tmp/upgrade-download
unzip cdf-2020.02.00120-2.2.0.2.zip
```

Note: In the event that command execution fails for the below steps, please run them again.

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

```
/tmp/upgrade-download/cdf-2020.02.00120-2.2.0.2/upgrade.sh -i
```

3. On the initial master node, run the following commands to upgrade CDF components:

```
/tmp/upgrade-download/cdf-2020.02.00120-2.2.0.2/upgrade.sh -u
```

4. Optionally, 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-2020.02.00120-2.2.0.2/upgrade.sh -c
```

5. Verify the cluster status. First, check the CDF version on each node by running the command:

```
cat ${K8S_HOME}/version.txt
>> 2020.02.00120
```

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

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

Automated Upgrade to CDF 2020.02

The automatic upgrade has 2 phases: the first for upgrade from CDF 2019.05 to CDF 2019.08, and the second for upgrade from CDF 2019.08 to 2020.02. The automated upgrade to CDF 2020.02 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)

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: Next, download the upgrade files for CDF 2018.08 and CDF 2020.02 to a download directory (referred to as `<download_directory>`) on the UM.

There are 4 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_201905_upgrade will be auto generated on every node. (approximate size 1.5 GB)
- A backup directory /tmp/CDF_201908_upgrade will be auto generated on every node. (approximate size 1.7 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.

Phase I: Auto-upgrade from CDF 2019.05 to CDF 2019.08

On the upgrade manager, run the following commands:

```
cd {download-directory}
```

```
unzip cdf-upgrade-2019.08.00134-2.2.0.2.zip
```

```
cd cdf-upgrade-2019.08.00134-2.2.0.2
```

```
./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
```

Note: In case of Automatic Upgrade failure please refer to "In Case of Automatic Upgrade Failure" on the next page.

Phase II: Auto-upgrade from CDF 2019.08 to CDF 2020.02

Proceed with the second phase of the automated upgrade, as follows:

1. Remove the 2019.08 upgrade directory:

   ```
   rm -rf {download-directory}/cdf-upgrade-2019.08.00134-2.2.0.2
   ```
2. Run a kubectl patch command to configure IDM pod affinity and wait until IDM pods are up and running with this command:

   ```
   kubectl patch deployment idm -n core --patch '{ "spec": { "template": {
   "spec": { "affinity": { "podAffinity": {
   "preferedDuringSchedulingIgnoredDuringExecution": [ { "labelSelector": {
   "matchExpressions": [ { "key": "app", "operator": "In", "values": [ "idm-app"
   ] } ] }, "topologyKey": "kubernetes.io/hostname" } ] } } } } } }'
   ```
3. Run the CDF 2020.02 auto-upgrade by executing these commands:

   ```
   cd {download-directory}
   unzip cdf-2020.02.00120-2.2.0.2.zip
   cd cdf-2020.02.00120-2.2.0.2
   ./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
```

Note: In case of Automatic Upgrade failure please refer to "In Case of Automatic Upgrade Failure" below.

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
```

In Case of Automatic Upgrade Failure

- If the automatic upgrade fails, run **`autoUpgrade.sh`** again as outlined above. The process may take several attempts to succeed.
- In some cases, the automatic upgrade may return an error message about the upgrade process still running and the existence of a `*.lock` file which prevents autoupgrade.sh to continue. This file is automatically deleted in a few minutes. Alternatively, you can manually delete this file. Once the file is deleted either automatically or manually, run **`autoUpgrade.sh`** again.
- If the automated upgrade process for Phase I is still unsuccessful, continue the process on the failed node using the procedure outlined in " Manual Upgrade Process from CDF 2019.05 to 2019.08" on page 82
- If the automated upgrade process for Phase II is still unsuccessful, continue the process on the failed node using the procedure outlined in "Manual Upgrade Process from CDF 2019.08 to 2020.02" on page 83.

Upgrading Arcsight Suite

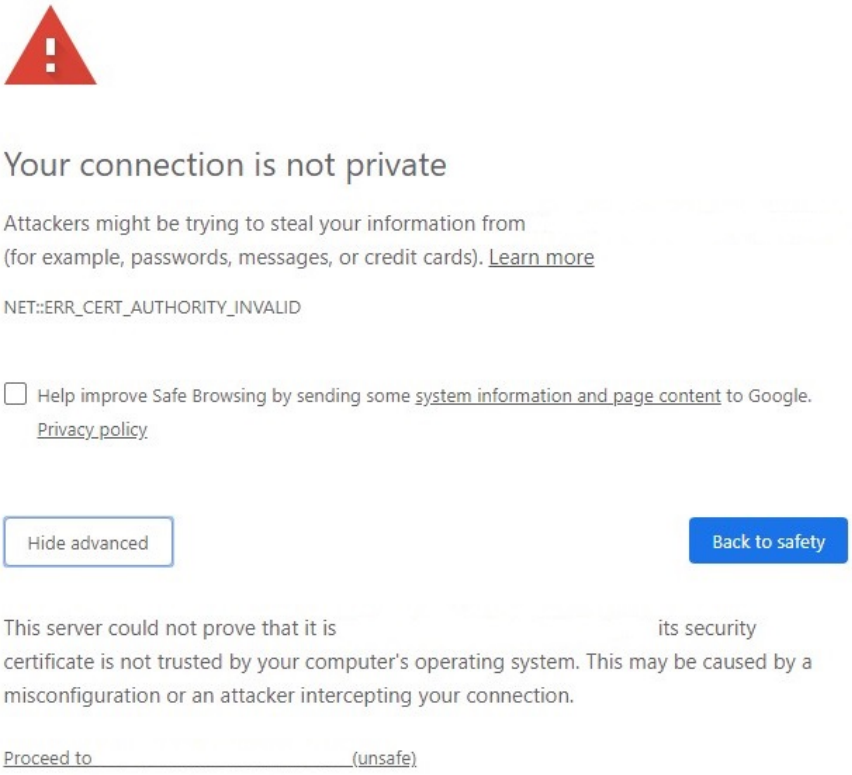
Before performing the upgrade:

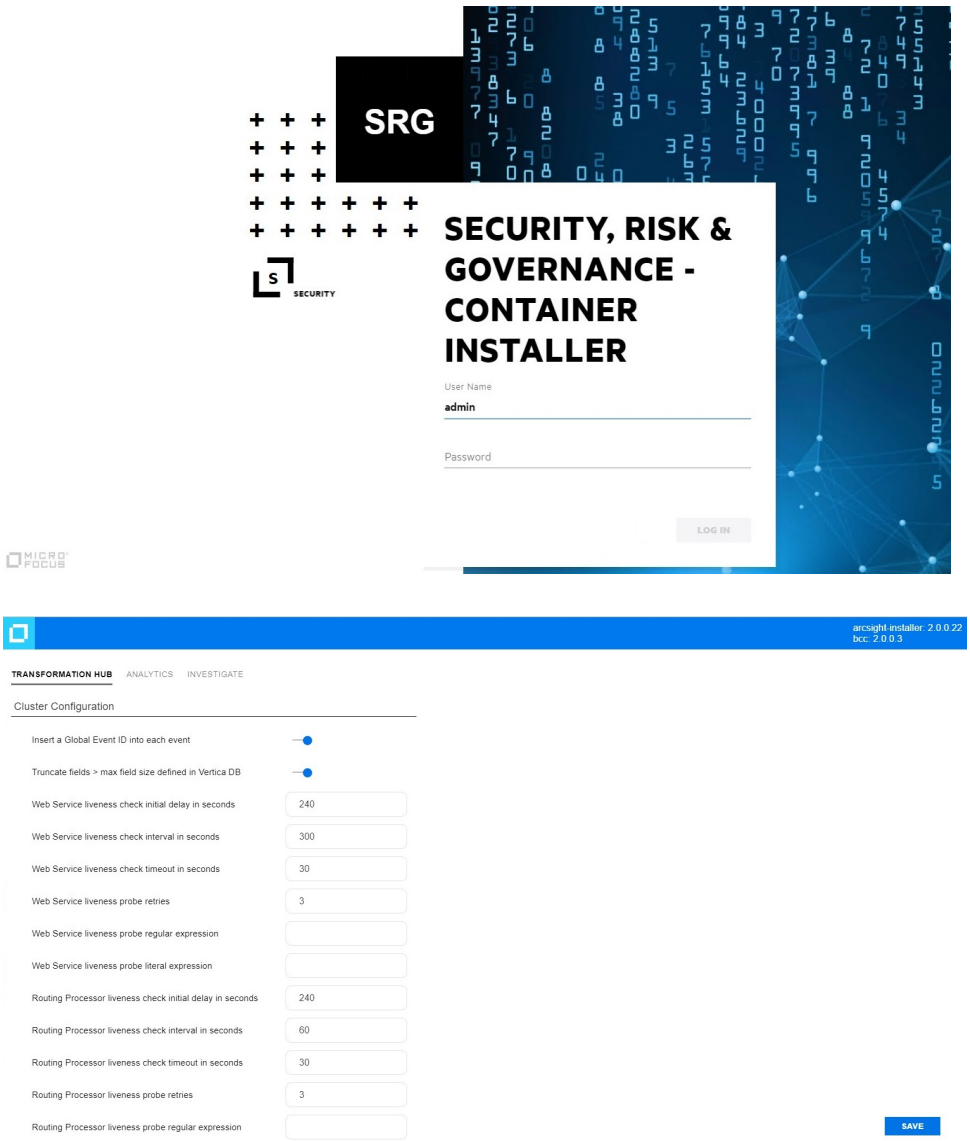
- Stop all operations
- Pause outliers scoring
- Backup Investigate Management and Search datastore

Note: Vertica and Investigate must be upgraded together. There is no specific upgrade order between Vertica and Investigate.

Suite Upgrade Steps

1. Accept Config Page Certificate
   - On the installed cluster make sure you have accessed configuration properties at least once to accept certificate. This step is important to avoid certificate error during upgrade.
     - Go to **Management Portal > Suite > Management > 3 dots > Reconfigure**
     - Accept the certificate





Note: If the web site takes you directly to the Post deployment UI (see image above) it means the page has been accessed before.

2. Download the upgrade bits - Metadata and Product offline images to the master node 1 directory. For example: `/tmp`.

`arcsight-installer-metadata-2.2.0.10.tar`

`analytics-3.1.0.10.tar`

`investigate-3.1.0.10.tar`

`post-install-3.1.0.tar.gz`

`transformationhub-3.2.0.10.tar`

Unpack offline images tar

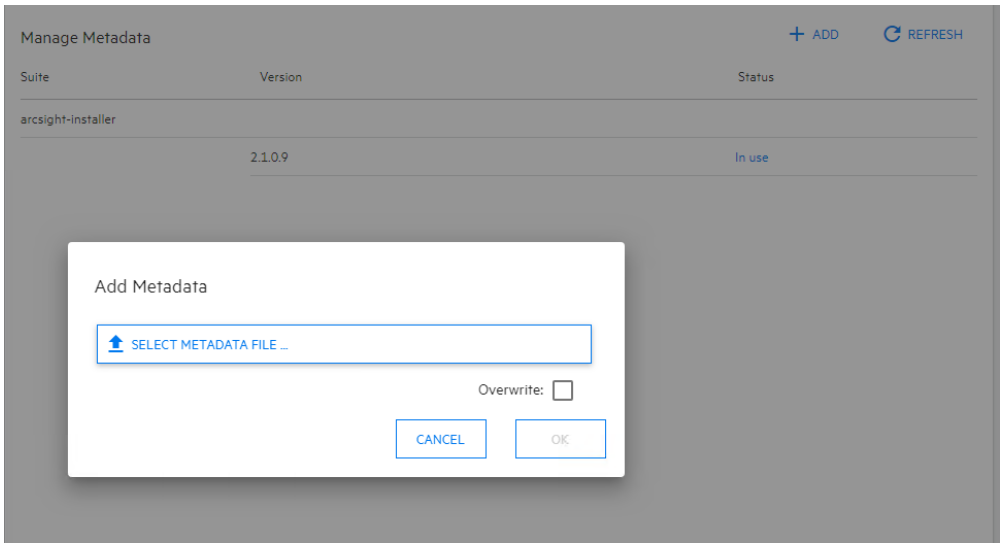
```
cd /tmp

tar -xvf analytics-3.1.0.10.tar

tar -xvf investigate-3.1.0.10.tar

tar -xvf transformationhub-3.2.0.10.tar

tar -xvf post-install-3.1.0.tar.gz
```

3. Add new metadata

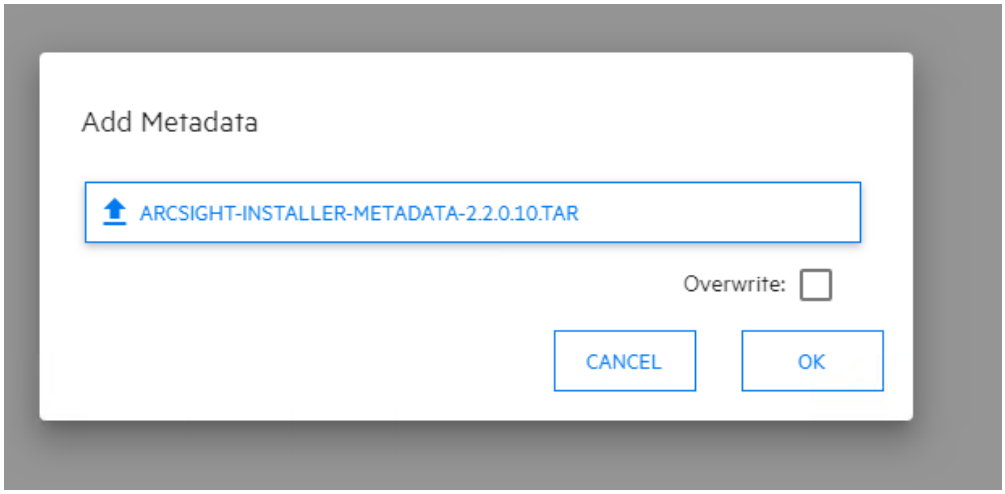
> **Note:** Make sure to copy the `arcsight-installer-metadata-2.2.0.10.tar` to your system before perform the process below.

From **Management portal** - add new metadata

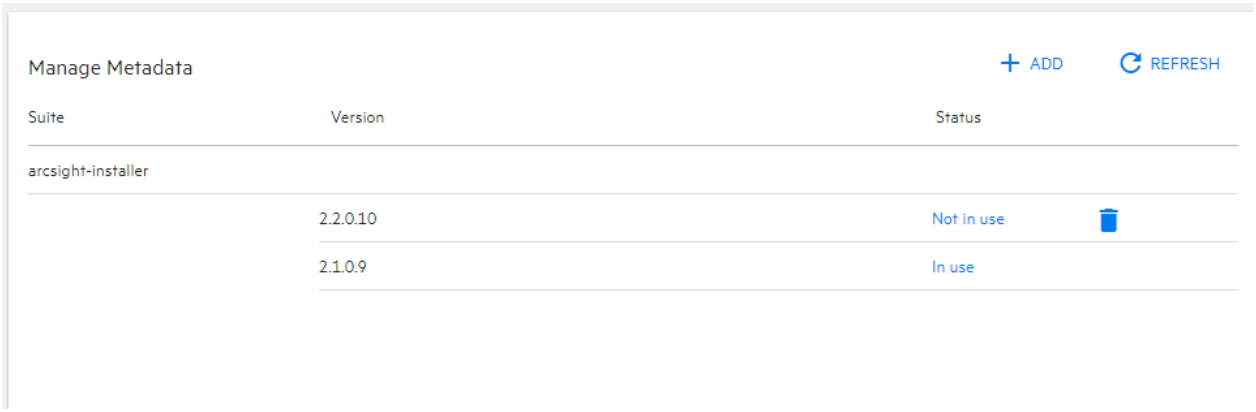
Go to **Administration > Metadata** and click the **+ Add** button



Select `arcsight-installer-metadata-2.2.0.10.tar` from your system

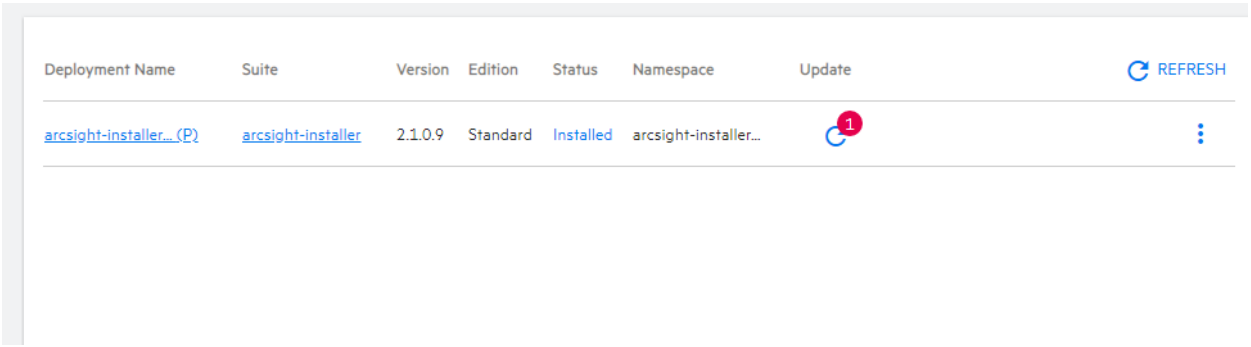


The new metadata will be added to the system.

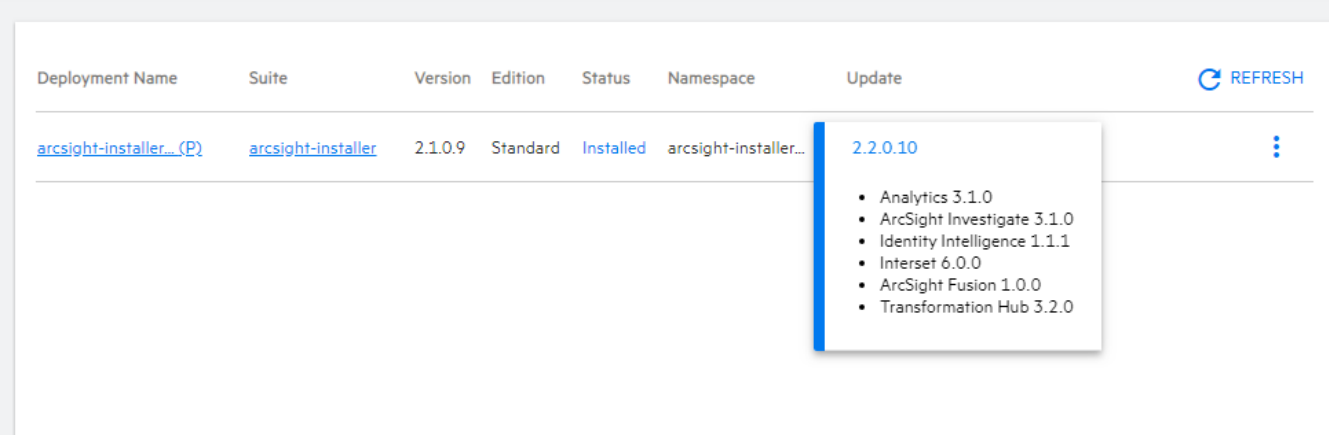


4. Start the upgrade process

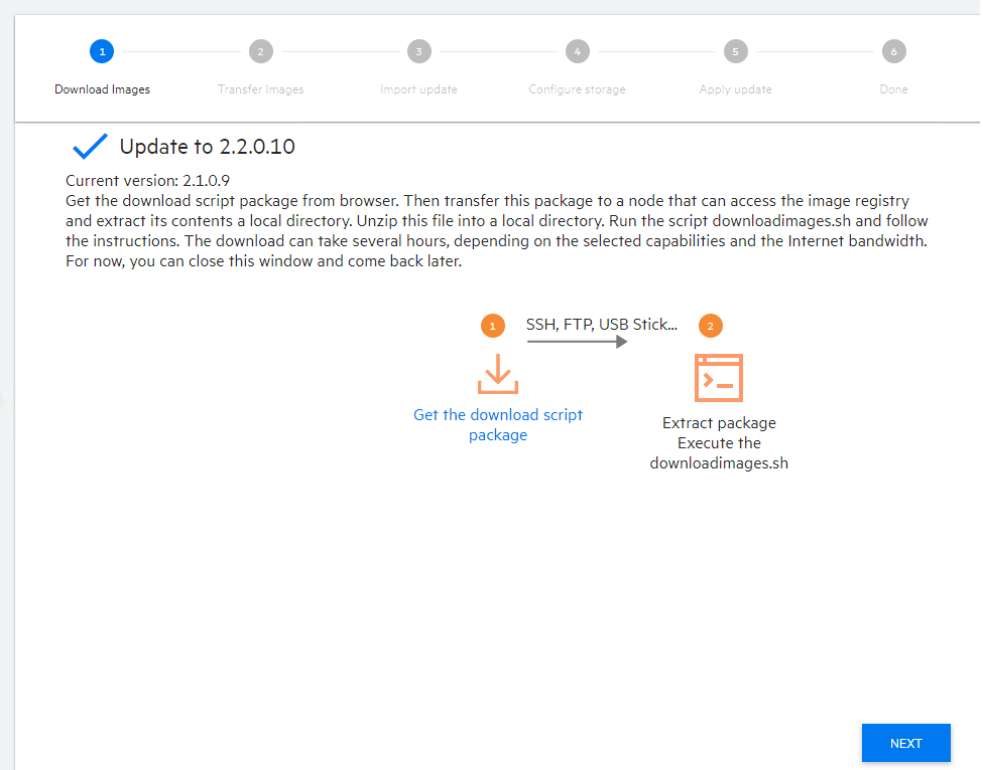
Go to **Suite > Management**. Notice the number **1** in the red circle on the Update column



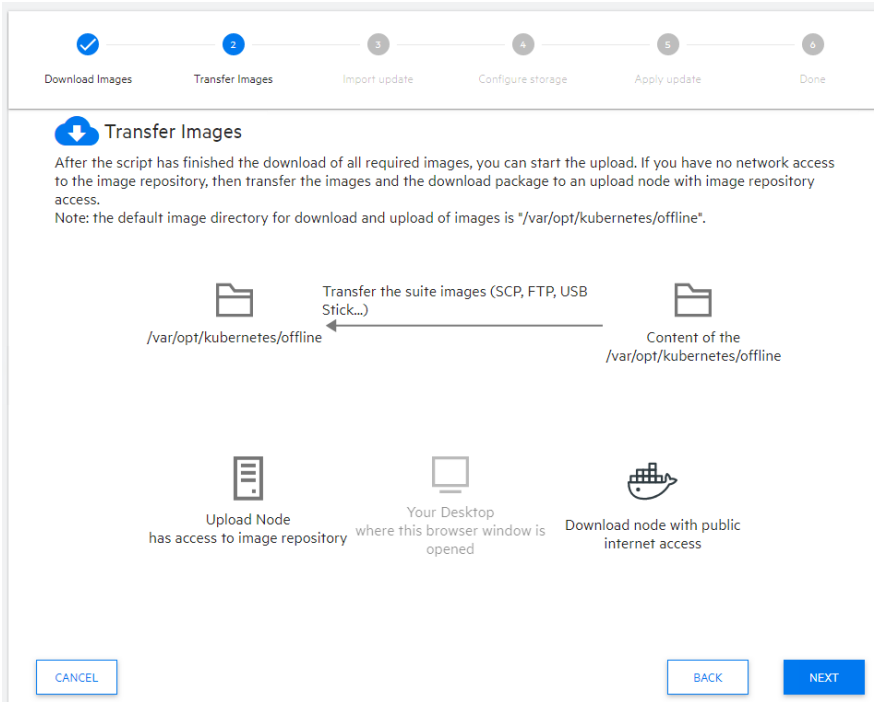
Click the red circle and choose your recently added metadata to initiate the upgrade



On the **Update to** page click **Next**

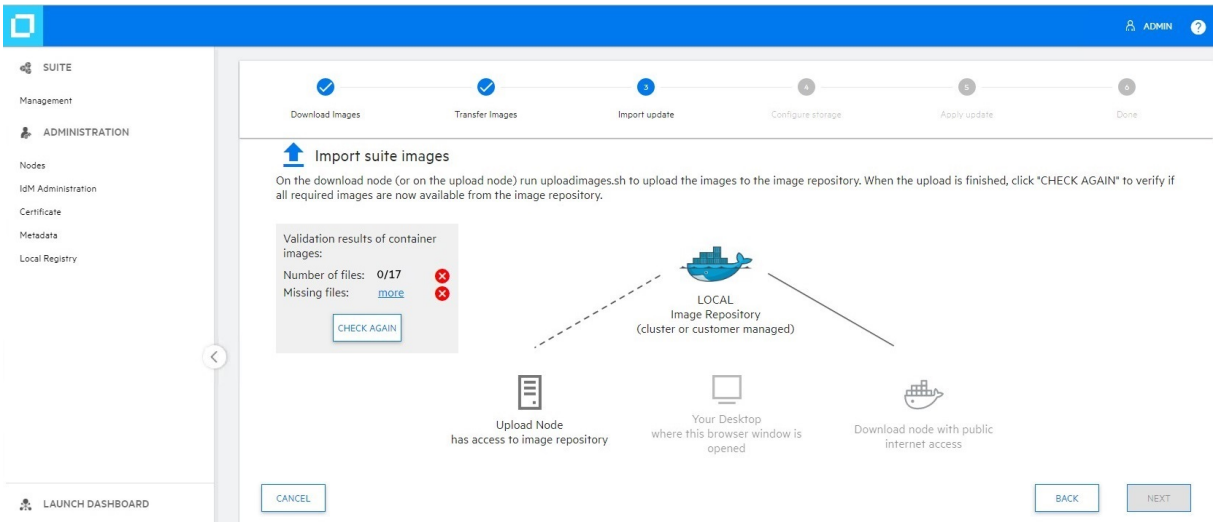


On the **Transfer images** page click **Next**



On Import suite images page click more to see what images are expected (3.x.0.4). On the next step we will upload them to docker registry.

Under **Management Portal> Import suite images** validation results of container images failed due to no images, as seen in the picture below.



5. Upload offline images from the master node 1

• Upload images to the local docker registry

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

Example: `cd /opt/arcsight/kubernetes/scripts`

```
./uploadimages.sh -u registry-admin -p {your_admin_password} -d
/path/to/extracted/product/folder
```

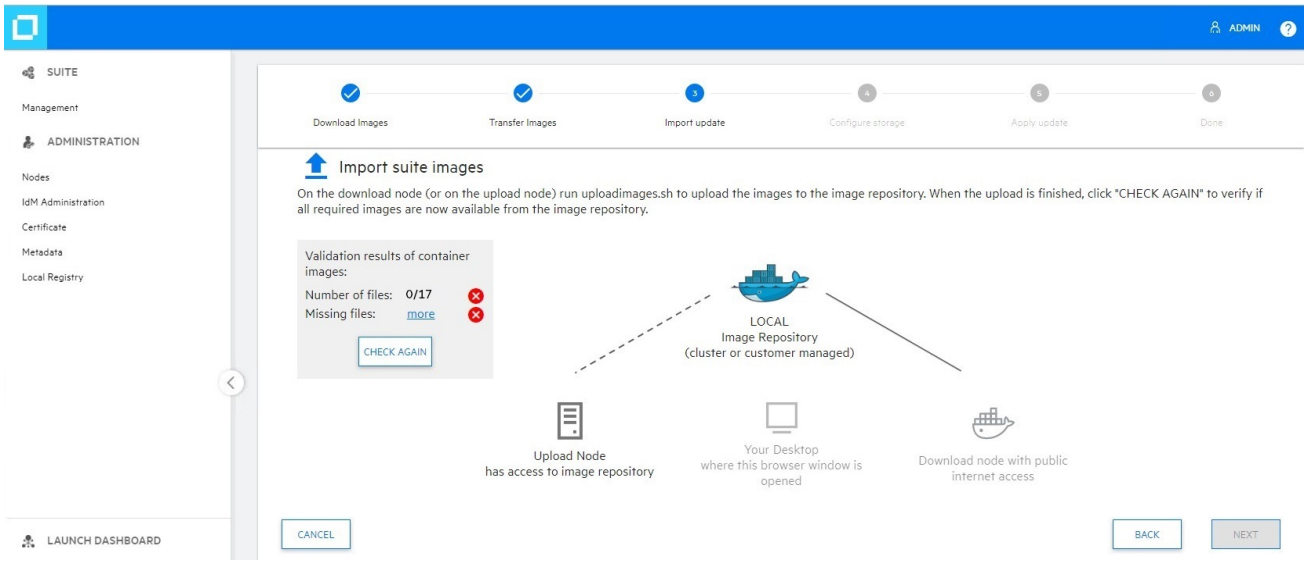
Example: `./uploadimages.sh -u registry-admin -p $password -d /tmp/transformationhub-3.2.0.10`

Example: `./uploadimages.sh -u registry-admin -p $password -d /tmp/investigate-3.1.0.10`

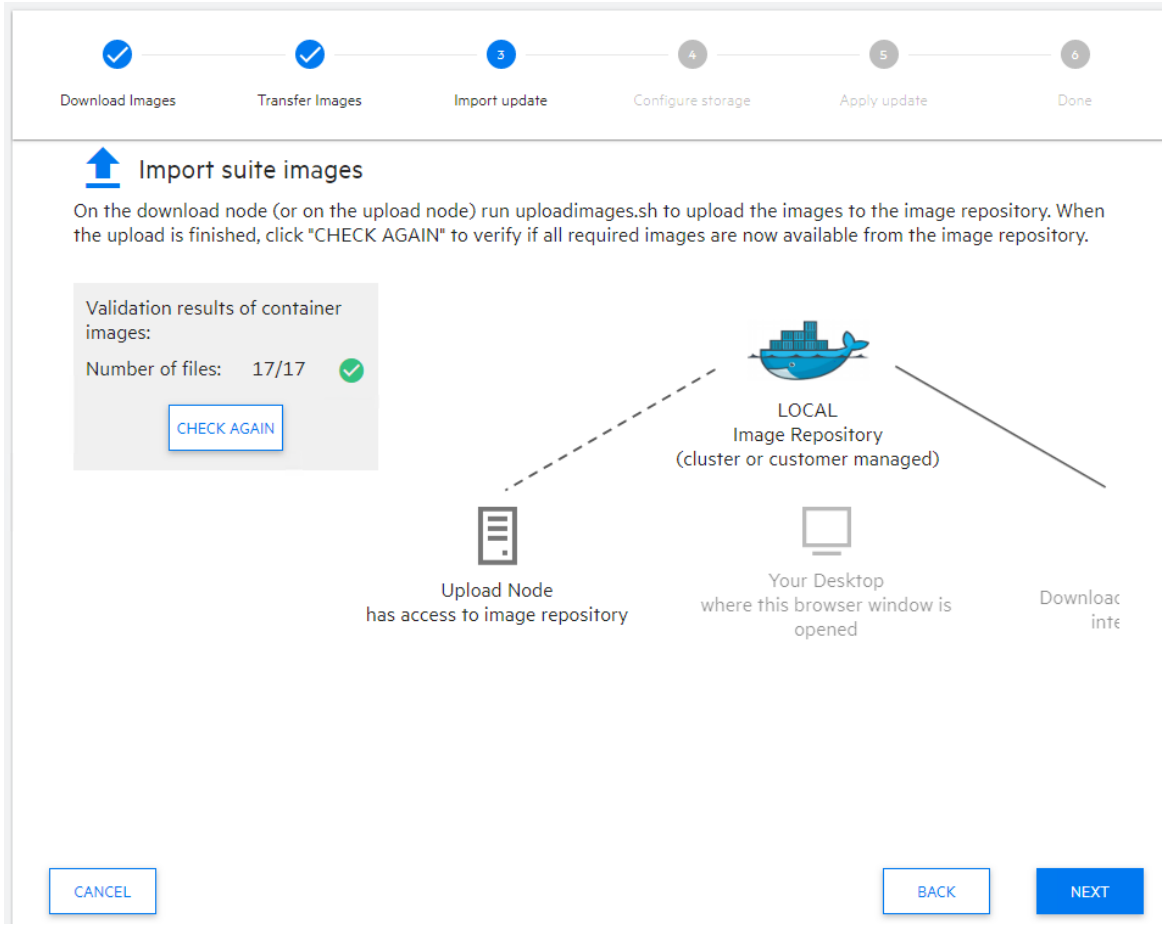
Example: `./uploadimages.sh -u registry-admin -p $password -d /tmp/analytics-3.1.0.10`

6. Finalize upgrade process

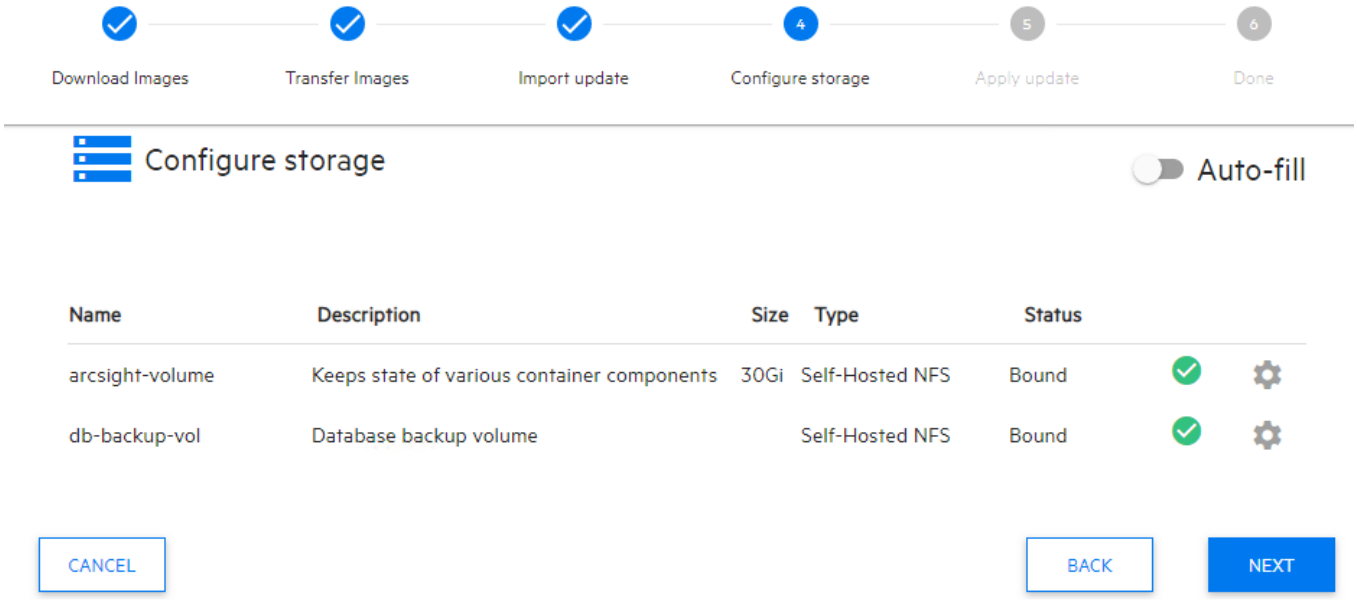
Go back to **Management Portal> Import suite images** page



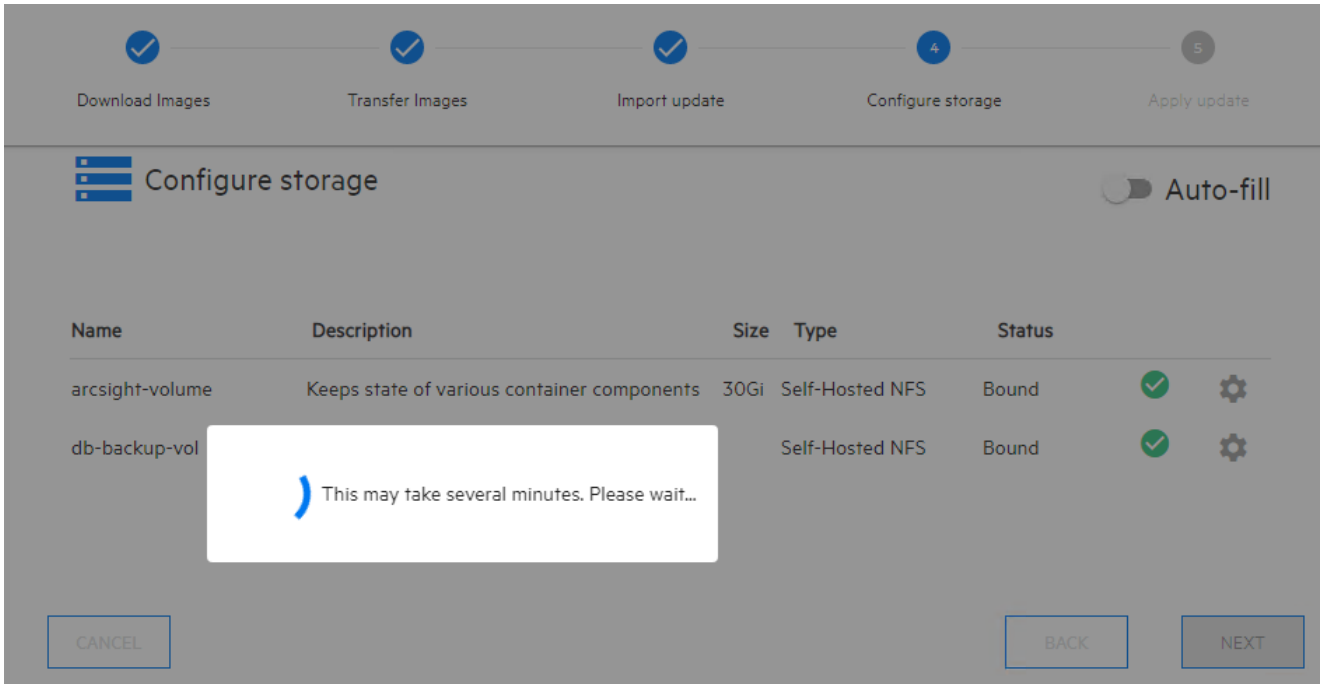
Click **CHECK AGAIN** button until you see that all the required images are available and the **Next** button is enabled.



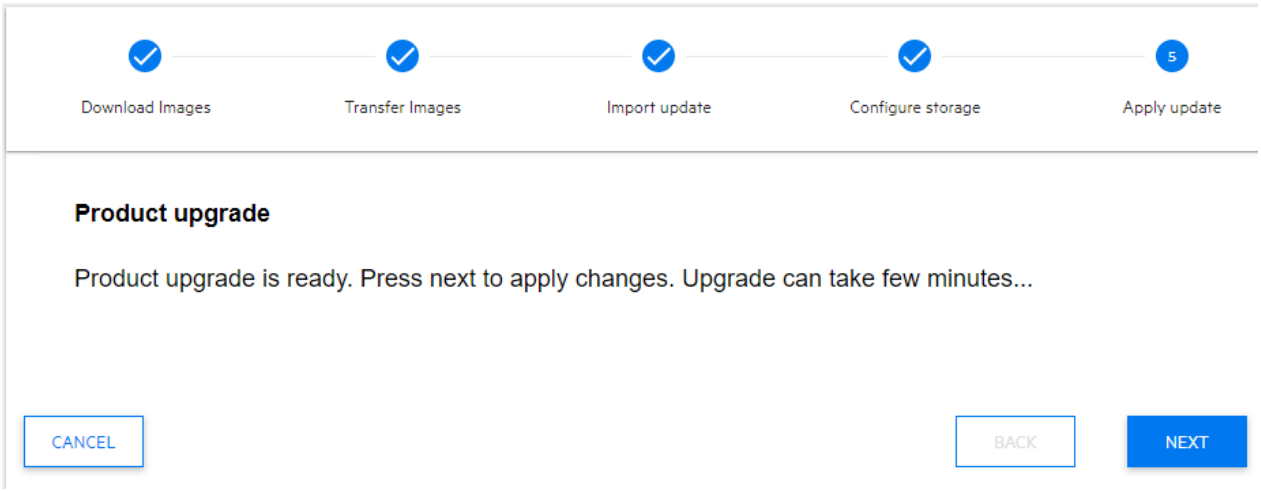
On Configure storage click **Next**. Wait until the next page shows up.

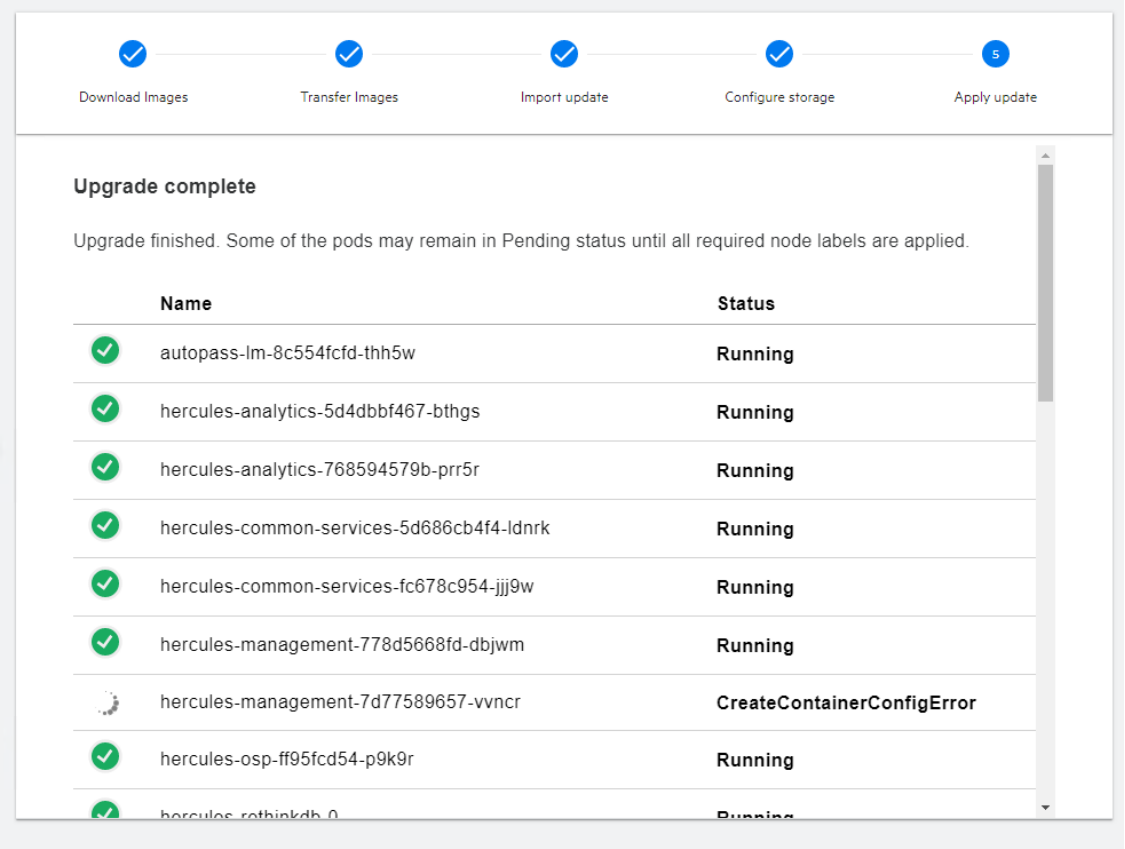


Upgrade config container is being deployed to the cluster.



On the Product upgrade page click **Next**. Now the process of upgrading TH and Investigate pods has started.





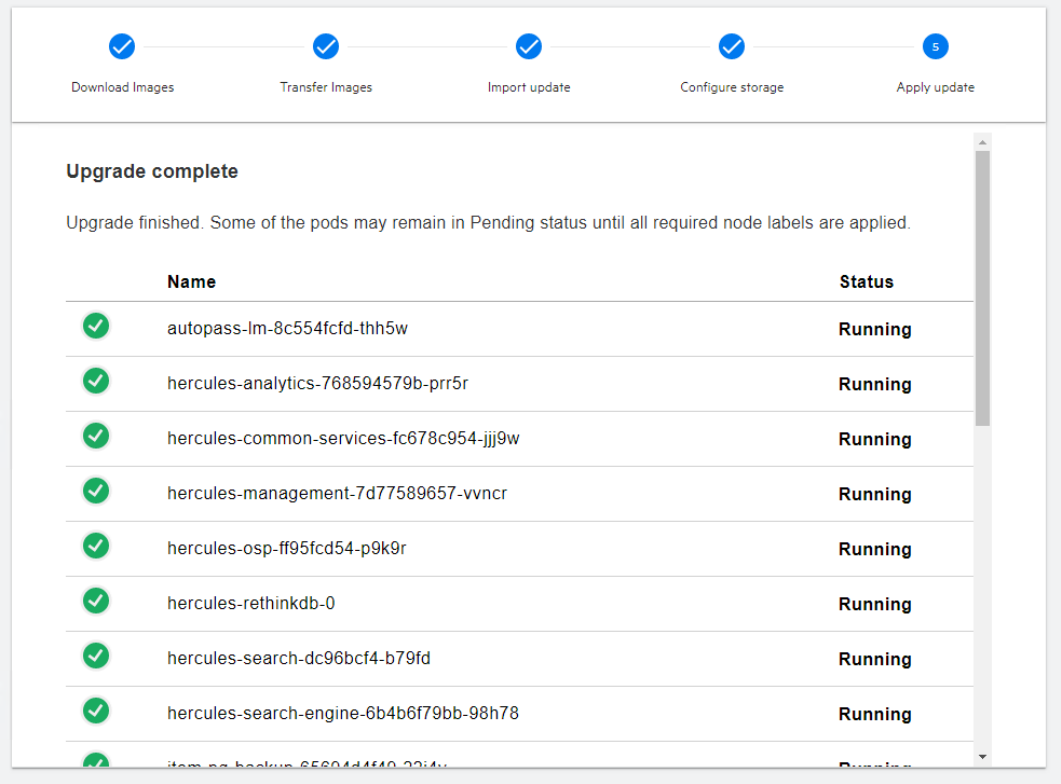
The error **CreateContainerConfigError** on the management pod is expected.

Perform the following steps on master node 1, where post-install-3.1.0.tar.gz was extracted, i.e. `/tmp`, to resolve the **CreateContainerConfigError** issue:

```
cd /tmp
./post-install-3.1.0.sh
```

```
./post-install-3.1.0.sh
Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply
Post installation successful.
```

Verify the management pod is in Running state:

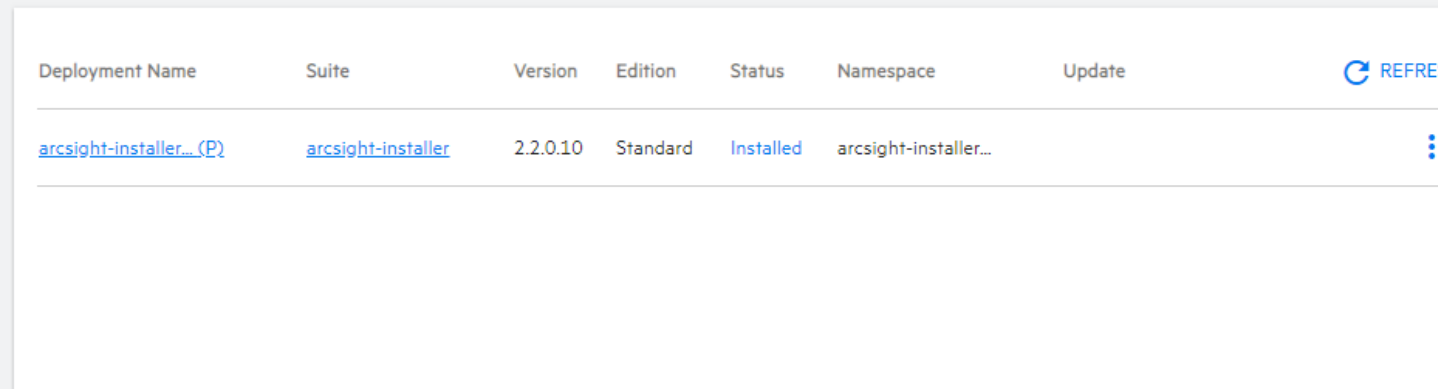


To monitor the management pod run the following command:

```
kubectl get pods –all-namespaces | grep hercules-management
```

The upgrade is now finished.

To see the new version of the suite go to **Suite > Management > Version column**



Restart and resume all operations.

Upgrade Returns `INTERNAL SERVER ERROR`

In some cases, after a successful upgrade of CDF, Transformation Hub, and Investigate, after attempting to reinstall Transformation Hub, the installer may display the error on the Configuration/Deployment page. If this error is encountered, follow this procedure to resolve the issue:

1. Run:

   ```
   Kubectl delete –n core $(kubectl get pods -n core -o name | grep itom-postgresql-default)
   ```

2. Wait for the pod to enter the Running state. Then run:

   ```
   Kubectl get pods –o wide- n core | grep itom-postgresql-default
   ```

3. On the **Configuration/Deployment** page, click **Deploy** again to deploy the product.

Chapter 15: 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 Investigate. 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 Investigate, 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.

This chapter includes the following sections:

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 Investigate.	
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.

Data Preservation

Topic data is preserved across restarts and reinstalls.

- When a Transformation Hub reinstall or redeployment is performed, all data that resides in Kafka topics is preserved. No data is lost. By default, events data is stored in a worker node: `/opt/arcsight/k8s-hostpath-volume/th/kafka.`
- When an Investigate reinstall or redeployment is performed, all data that resides in Kafka topics is preserved. No data is lost.
- When a consumer resumes data consumption from Kafka topics, the consumer restarts where it left off. No data is lost.
- If a Transformation Hub worker node is stopped, that node will be reported as unavailable to the cluster. All events data stored on the worker node will be preserved and events processing will resume as soon as the node is started again.
- If an Investigate worker node is stopped, that node will be reported as unavailable to the cluster. Investigate will not function until the node is started again.
- If a master node is stopped, that node will be reported as unavailable to the cluster. All other functionality, including events processing on the worker nodes and Investigate will continue.

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:

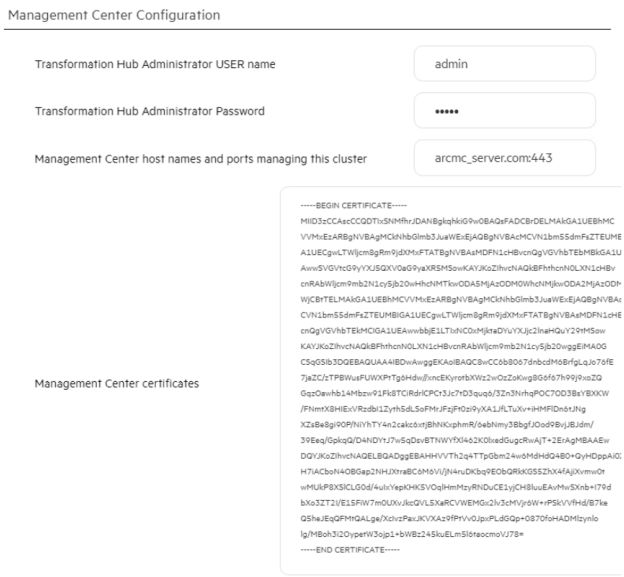
- Retrieve the ArcMC certificate from your ArcMC
- Configure the CDF cluster with ArcMC details
- Retrieve the CDF certificate
- Configure ArcMC

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

Follow these instructions to configure a security mode for SmartConnectors with Transformation Hub destinations. Transformation Hub and SmartConnectors with Transformation Hub destinations. 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
```

Chapter 16: Maintaining the Transformation Hub

Administration of the Transformation Hub cluster is performed from the Transformation Hub Kafka Manager Portal, available at `https:<Your high-availabiliy FQDN>:5443.`

Changing Transformation Hub Configuration Properties

To change Transformation Hub configuration properties:

1. In the Management Portal, select **Suite.**
2. Click **...** (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
3. Update configuration properties as needed.
4. Click **Save.**

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

Adding a Product (Capability)

To add a product (capability) to your cluster:

1. As explained under Upload Images, upload the offline images for the product you want to add.
2. Click **Suite.**.
3. Click **...** (Browse) on the far right and choose **Change**. A new screen will be opened in a separate tab.
4. On the next page, select a product you want to add, and click **Next**
5. On the **File Storage** page, fill in the NFS volume data if needed, and click **Next**.
6. Wait until the spinner disappears (This page will remain blank) and click **Next.**
7. Update configuration values if needed, and click **Next**.

After a short wait, the Configuration Complete page confirms the change to the cluster.

Removing a Product

To remove a product (capability) from your cluster:

1. Click **Suite.**.
2. Click **...** (Browse) on the far right and choose **Install**. A new screen will be opened in a separate tab.
3. On the next page, deselect the product you want to remove, and click **Next**
4. On the **File Storage** page click **Next**.
5. Update configuration values if needed, and click **Next**.

After a short wait, the **Configuration Complete** page confirms the change to the cluster.

Uninstalling ArcSight Suite (including Transformation Hub)

To (gracefully) uninstall the ArcSight Suite (including Transformation Hub):

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. Click **Suite > Management**.
4. Click on the far right button and choose **Uninstall**.

The pods are progressively shut down and then uninstalled.

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}: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. 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**.

Viewing 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.

To display the current CA for external communication:

Run the following command on the Initial Master Node:

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

To change the CA, run:

```
cdf-updateRE.sh write --re-key={New Intermediate Key Name}.pem --re-crt=
{New Intermediate Key Name}.pem --re-ca={New CA Cert Name}.pem}
```

Note: Changing the CA after Transformation Hub deployment will necessitate undeploying and then deploying the Transformation Hub capability. 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 Transformation Hub rollout. See the section on Deploying Transformation Hub for information on re-deploying the capability.

Chapter 17: Integrate Investigate Single Sign-On with any External SAML 2 Identity Provider

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

Note: Investigate 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.

1. Log in to the CDF server and navigate to the sso configuration folder:
   `/opt/arcsight/nfs/vol/arcsight/sso/default`
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`
   - `com.microfocus.sso.default.login.saml2.metadata-url = <IDP SAML metadata URL>`

   `IDP SAML metadata URL` is the address where the IDP supplies its metadata document. 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 Investigate Single Sign-on keystore for HTTPS to work properly.

   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>`

   Note for 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, the Single Sign-on provider that will interact with the trusted provider must be configured with the trusted provider's metadata.

This is done with a `<Metadata>` element found under the `<AccessSettings>` element under the `<TrustedIDP>` element or the `<TrustedSP>` element.

```
com.microfocus.sso.default.login.saml2.mapping-attr = email
```

The `email` attribute refers to the email attribute name from the SAML2 IDP.

3. Restart the pod.
   - Get the pod information:
   ```
   kubectl get pods --all-namespaces | grep osp
   ```
   - Restart the pod by deleting the current running pod:
   ```
   kubectl delete pod hercules-osp-xxxxxxxxxx-xxxxx -n arcsight-installer-xxxxx
   ```

4. Retrieve the Investigate Single Sign-On SAML service provider metadata from the Investigate server:

   ```
   https://EXTERNAL_ACCESS_HOST/osp/a/default/auth/saml2/spmetadata
   ```

   `EXTERNAL_ACCESS_HOST` is the hostname or IP address of the Investigate server.

5. Use the Investigate Single Sign-On SAML service provider metadata to configure your IDP. For detailed instructions, see the IDP software documentation.
6. To establish a trust relationship between Investigate 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.
7. Log in to the CDF server and navigate to the Investigate Single Sign-on default configuration folder.
8. Import the IDP certificate file to the Investigate Single Sign-On keystore:

   ```
   ./keytool -importcert -file FileName.cer -keystore
   /path/to/sso/default/config/sso.keystore -alias AliasName
   ```

   - `FileName` corresponds to the name of the certificate file you want to import.
   - `AliasName` is the new alias name to be assigned to the certificate in the Investigate Single Sign-On keystore.

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.

Chapter 18: Troubleshooting

The following can help to diagnose common Investigate issues.

Upgrade Returns INTERNAL SERVER ERROR

In some cases, after a successful upgrade of CDF, Transformation Hub, and Investigate, after attempting to reinstall Transformation Hub, the installer may display the error on the Configuration/Deployment page. If this error is encountered, follow this procedure to resolve the issue:

1. Run:

   ```
   Kubectl delete –n core $(kubectl get pods -n core -o name | grep itom-postgresql-default)
   ```

2. Wait for the pod to enter the Running state. Then run:

   ```
   Kubectl get pods –o wide- n core | grep itom-postgresql-default
   ```

3. On the **Configuration/Deployment** page, click **Deploy** again to deploy the product.

[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 Investigate 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.

Appendix A: 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.

Appendix B: Creating an Intermediate Key and Certificate

This appendix details the process for creating an intermediate key and certificate (cert) file. It contains the following sections:

Best Practice: Note that in order to import an intermediate (user) certificate, the Transformation Hub must be deployed, undeployed, and then re-deployed. We recommend that you perform this procedure when Transformation Hub is first installed to avoid downtime and data loss.

In outline, your initial Transformation Hub deployment would then consist of these steps:

1. Install CDF
2. Deploy Transformation Hub with default settings
3. Perform the operations described here to create the intermediate certificate (detailed below).
4. From the CDF UI, uninstall the Transformation Hub
5. After the Transformation Hub is uninstalled, redeploy the Transformation Hub.
6. Configure your pre-deployment parameters (such as Client Authentication or FIPS) as desired.

To obtain the contents of the RE certificate, use the following script:

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

However, the CA (certificate authority) private key is not available. Therefore, in order to create a signed certificate, you will need to create and use an intermediate key and CA.

Create a New CA Certificate

1. Make the directory and configure:

```
mkdir /root/ca

cd /root/ca

mkdir certs crl

newcerts private
```

```
chmod 700 private

touch index.txt

echo 1000 > serial
```

2. Open the configuration file in a text editor (`vi /root/ca/openssl.cnf`), and then add the following contents (values shown are examples; change parameter values to match yours):

```
# 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.pem

certificate       = $dir/certs/ca.cert.pem

# 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                     = Country
stateOrProvinceName             = State
localityName                    = Locality
0.organizationName              = EntCorp
organizationalUnitName          = OrgName
commonName                      = Common Name
emailAddress                    = Email Address
# Optionally, specify some defaults.
countryName_default             = GB
stateOrProvinceName_default     = England
localityName_default            =
0.organizationName_default      = Microfocus
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.pem 4096

chmod 400 private/ca.key.pem
```

4. Create the new CA cert:

```
openssl req -config openssl.cnf \  -key private/ca.key.pem \  -new -x509 -
days 365 -sha256 -extensions v3_ca \ -out certs/ca.cert.pem
```

5. Verify the root CA:

```
chmod 444 certs/ca.cert.pemopenssl x509 -noout -text -in certs/ca.cert.pem
```

Create a New Intermediate Key and Certificate

1. Make the directory and configure:

```
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
```

2. Open the configuration file in a text editor (`vi /root/ca/openssl.cnf`), and then add the following contents (values shown are examples; change parameter values to match yours). Make sure the directory is unique for each intermediate CA.

```
[ 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.pem
certificate       = $dir/certs/intermediate.cert.pem
# 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      = Micro Focus
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 Intermediate CA key:

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

4. Create the intermediate CA signing request (CSR):

```
chmod 400 intermediate/private/intermediate.key.pem
```

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

5. 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.cert.pem

   # Sign the certificate? [y/n]: y

   chmod 444 intermediate/certs/intermediate.cert.pem
```

6. Verify the intermediate CA:

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

7. Verify the intermediate certificate against the root certificate

```
   openssl verify -CAfile certs/ca.cert.pem \

   intermediate/certs/intermediate.cert.pem

   # intermediate.cert.pem: OK
```

8. Verify the intermediate CA against the root CA:

```
   openssl verify -CAfile certs/ca.cert.pem \

   intermediate/certs/intermediate.cert.pem

   # intermediate.cert.pem: OK
```

Update the Certificate Set on the Transformation Hub Cluster

In order to update the CA cert used by Transformation Hub, copy your intermediate key and intermediate cert along with the CA cert from the server where they were created to the Initial Master Node, and then do the following:

1. Run:

```
   ${k8s-home}/scripts/arcsight-cert-util.sh write --re-
   key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-
   ca=/tmp/ca.cert.pem
```

Note: the path to the `intermediate.key.pem, intermediate.cert.pem,` and `ca.cert.pem` can be any path desired

2. From the CDF UI, uninstall the Transformation Hub.
3. After the Transformation Hub is uninstalled, redeploy the Transformation Hub.

Appendix C: Fields Indexed by Default in Vertica

Investigate indexes a subset of event fields for use in free form text search. Free form text search can only be done for values in event fields that are indexed. Following is the list of event fields that are indexed by default in Vertica:

agentDnsDomain	deviceCustomString2Label	flexNumber2Label
agentHostName	deviceCustomString3	flexString1
agentTranslatedZoneURI	deviceCustomString3Label	flexString1Label
agentZoneURI	deviceCustomString4	flexString2
applicationProtocol	deviceCustomString4Label	flexString2Label
cryptoSignature	deviceCustomString5	message
destinationDnsDomain	deviceCustomString5Label	name
destinationGeoLocationInfo	deviceCustomString6	oldFileId
destinationHostName	deviceCustomString6Label	oldFileName
destinationNtDomain	deviceDnsDomain	oldFilePath
destinationProcessName	deviceDomain	oldFileType
destinationServiceName	deviceEventCategory	rawEvent
destinationTranslatedZoneURI	deviceExternalId	reason
destinationUserId	deviceFacility	requestClientApplication
destinationUserName	deviceHostName	requestContext
destinationUserPrivileges	deviceNtDomain	requestCookies
destinationZoneURI	devicePayloadId	requestUrl
deviceAction	deviceProcessName	requestUrlFileName
deviceAssetId	deviceProduct	requestUrlQuery
deviceCustomDate1Label	deviceSeverity	sourceDnsDomain
deviceCustomDate2Label	deviceTranslatedZoneURI	sourceGeoLocationInfo
deviceCustomFloatingPoint1Label	deviceVendor	sourceHostName
deviceCustomFloatingPoint2Label	deviceZoneURI	sourceNtDomain
deviceCustomFloatingPoint3Label	eventOutcome	sourceProcessName
deviceCustomFloatingPoint4Label	externalId	sourceServiceName
deviceCustomIPv6Address1Label	fileId	sourceTranslatedZoneURI

deviceCustomIPv6Address2Label	fileName	sourceUserId
deviceCustomIPv6Address3Label	filePath	sourceGeoCountryCode
deviceCustomIPv6Address4Label	fileType	sourceUserName
deviceCustomNumber1Label	flexDate1Label	sourceUserPrivileges
deviceCustomNumber2Label	categoryBehavior	sourceGeoPostalCode
deviceCustomNumber3Label	destinationGeoCountryCode	sourceGeoRegionCode
deviceCustomString1	flexNumber1Label	sourceZoneURI
deviceCustomString1Label	destinationGeoPostalCode	
deviceCustomString2	destinationGeoRegionCode	

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 Vertica installer before installing Vertica.

If users want to modify the event fields indexed after Vertica 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.

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 Deployment Guide (Investigate 3.1.0)

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!