



Micro Focus Transformation Hub

Software Version: 3.1.0

Deployment Guide

Document Release Date: November 21, 2019

Software Release Date: November 22, 2019

Legal Notices

Copyright Notice

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Except as specifically indicated otherwise, a valid license from Micro Focus is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

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

UNIX® is a registered trademark of The Open Group.

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://communitysoftwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

Contents

Transformation Hub Deployment Guide	6
Transformation Hub	7
Secure Open Data Platform (SODP)	8
Management Center (ArcMC)	8
SmartConnectors	8
ArcSight Investigate	9
Identity Intelligence	9
Glossary	9
Chapter 2: Deployment Planning and Preparation	14
Gather Required Information	14
Secure Communication Between Micro Focus Components	15
Download Installation Packages	16
Chapter 3: Installation and Deployment	19
Configure and Install the CDF Installer	20
Configure and Deploy the Kubernetes Cluster	22
Download Transformation Hub and Core Images to the Local Docker Registry	28
Uploading Images	29
Verify Prerequisite and Installation Images	29
Deploy Node Infrastructure and Services	30
Preparation Complete	31
Configure and Deploy Transformation Hub	32
Security Mode Configuration	33
Label Worker Nodes	34
Check Deployment Status	36
Check Cluster Status	37
Post-Deployment Configuration	37
Additional Steps	38
Management Center: Configuring Transformation Hub	38
Reminder: Install Your License Key	39
Chapter 4: Integrating Transformation Hub Into Your ArcSight Environment	40
Default Topics	40
Configuring ArcMC to Manage a Transformation Hub	42
Configuring Security Mode for Transformation Hub Destinations	44

Configuring a Transformation Hub Destination without Client Authentication in non-FIPS Mode	44
On the SmartConnector Server	44
Configure a Transformation Hub Destination with Client Authentication in FIPS Mode	46
Step 1: On the Connector Server	47
Step 2: On the Transformation Hub Server	49
Step 3: On the Connector Server	49
Step 4: On the Transformation Hub Server	49
Step 5: On the Connector Server	50
Step 6: On the Transformation Hub Server	52
Configure a Transformation Hub Destination with Client Authentication in Non-FIPS Mode	52
Step 1: On the Connector Server	52
Step 2: On the Transformation Hub Server	54
Step 3: On the Connector Server	55
Step 4: On the Transformation Hub Server	55
Step 5: On the Connector Server	55
Step 6: On the Transformation Hub Server	57
Configure a Transformation Hub Destination without Client Authentication in FIPS Mode	57
On the SmartConnector Server	58
Troubleshooting SmartConnector Integration	60
Configuring Logger as a Transformation Hub Consumer	61
Configuring ESM as a Consumer	63
Chapter 5: Maintaining the Transformation Hub	65
Changing Transformation Hub Configuration Properties	65
Adding a Product (Capability)	65
Removing a Product	66
Uninstalling ArcSight Suite (including Transformation Hub)	66
Resetting the Administrator Password	66
Viewing and Changing the Certificate Authority	67
Upgrading ArcSight Suite	67
Appendix A: CDF Installer Script install.sh Command Line Arguments	71
Appendix B: Creating an Intermediate Key and Certificate	74
Create a New CA Certificate (example)	74
Create a New Intermediate Key and Certificate	79
Update the Certificate Set on the Transformation Hub Cluster	84
Appendix C: Troubleshooting	86

Send Documentation Feedback 88

Transformation Hub Deployment Guide

The Transformation Hub Deployment Guide describes the process of deploying and configuring CDF, Kubernetes, and Transformation Hub. The instructions contained here presume you have already reviewed the CDF Planning Guide, which describes the process of planning your deployment and is a prerequisite to the actual deployment process.

For further information on Transformation Hub, consult the following documentation:

- Transformation Hub Administrator's Guide: This guide describes the operations and maintenance of Transformation Hub in detail.
- Transformation Hub Release Notes: This document includes important information about this release of Transformation Hub.

Transformation Hub

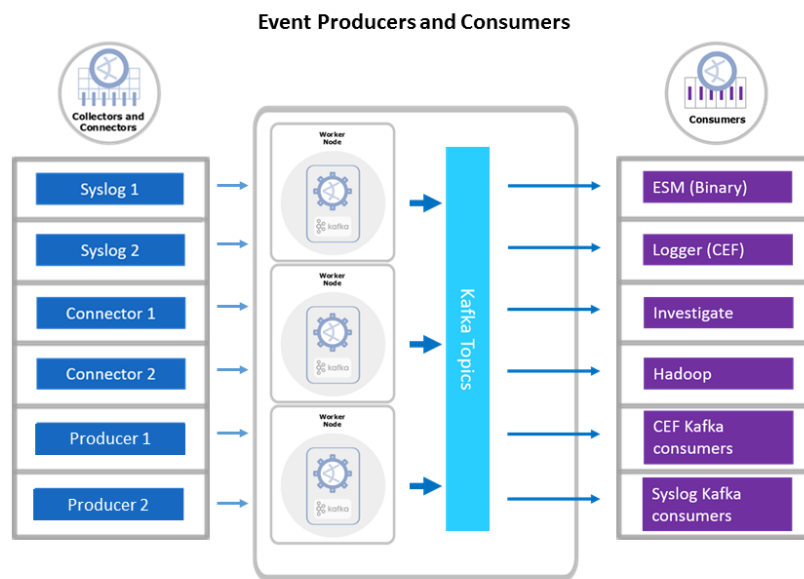
Transformation Hub is the high-performance message bus for ArcSight security, network, flows, application, and other events. It can queue, transform, and route security events to other ArcSight or third party software. This Kafka-based platform allows ArcSight components like Logger, ESM, and Investigate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

Transformation Hub ingests, enriches, normalizes, and then routes event data from data producers to connections between existing data lakes, analytics platforms, and other security technologies and the multiple systems within the Security Operations Center (SOC). Transformation Hub can seamlessly broker data from any source and to any destination. Its architecture is based on Apache Kafka and it supports native Hadoop Distributed File System (HDFS) capabilities, enabling both the ArcSight Logger and ArcSight 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.



Secure Open Data Platform (SODP)

The Secure Open Data Platform (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
- Management Center (ArcMC)
- Smart Connectors

Management Center (ArcMC)

ArcMC is a central administrative user interface for managing Secure ODP. This management console administers SODP infrastructure, including users, configurations, backups, updates, 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.

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.

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 you can view and search on them. You can use the English-like search language to generate results from which to create reports and visualizations.

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.

Glossary

A

Avro

Avro is a row-oriented remote procedure call and data serialization framework developed within Apache's Hadoop project. It uses JSON for defining data types and protocols, and serializes data in a compact binary format.

C

Cluster

A group of nodes, pods, or hosts.

Common Event Format (CEF)

CEF is an open log management standard that simplifies log management, letting third parties create their own device schemas that are compatible with a standard that is used industry-wide for normalizing security events. Technology companies and customers can use the standardized CEF format to facilitate data collection and aggregation, for later analysis by an enterprise management system.

Connectors in Transformation Hub (CTH)

A feature where SmartConnector technology operates directly in Transformation Hub to collect data. CTH enables the enriching, normalizing and sending of syslog data and routing it to Kafka topics.

Consumer

A consumer of Transformation Hub event data. Consumers may be Micro Focus products such as Logger or ESM, third-party products like Hadoop, or can be made by customers for their own use.

Container Deployment Foundation (CDF)

CDF is the container-based delivery and management model built on Docker containers managed by Kubernetes, which standardizes distribution, installation, upgrade, and operation of Micro Focus products and product suites.

D

Dedicated Master Node

A node dedicated to running the Kubernetes control plane functionality only.

Destination

In Micro Focus products, a forwarding location for event data. A Transformation Hub topic is one example of a destination.

Docker Container

A Docker container is a portable application package running on the Docker software development platform. Containers are portable among any system running the Linux operating system.

F

flannel

flannel (spelled with a lower-case f) is a virtual network that gives a subnet to each host for use with container runtimes. Platforms like Google's Kubernetes assume that each container (pod) has a unique, routable IP inside the cluster. The advantage of this model is that it reduces the complexity of doing port mapping.

Fully Qualified Domain Name (FQDN)

A fully qualified domain name (FQDN) is the complete domain name for a specific computer, or host, on the internet. The FQDN consists of two parts: the hostname and the domain name. For example, an FQDN for a hypothetical mail server might be mymail.example.com. The hostname is mymail, and the host is located within the domain example.com.

I

Initial Master Node

The Master Node that has been designated as the primary Master Node in the cluster. It is from this node that you will install the cluster infrastructure.

K

Kafka

An open-source messaging system that publishes messages for subscribers to consume on its scalable platform built to run on servers. It is commonly referred to as a message broker.

Kubernetes

Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery.

L

Labeling

Adding a Kubernetes label to a Master or Worker Node creates an affinity for the workload to the Master or Worker Node, enabling the node to run the specified workload on the labeled server.

Local Docker Registry

The Docker Registry location on the Master and Worker Nodes in the cluster. Application software is launched and managed from the Local Docker Registry.

M

Master Nodes

Master Nodes run the CDF Installer and process web services calls made to the cluster. A minimum of 1 Master Node is required for each cluster.

N

Network File System (NFS)

This is the location where the CDF Installer, Transformation Hub, and other components may store persistent data. A customer-provisioned NFS is required. This environment is referred to in this documentation as an "external" NFS. Although the CDF platform can host a CDF-provisioned NFS (Internal NFS), for high availability an External NFS service should be implemented.

Node

A processing location. In CDF containerized applications, nodes come in two types: master and worker.

P

Pod

Applications running in Kubernetes are defined as "pods", which group containerized components. CDF uses Docker Containers as these components. A pod consists of one or more containers that are guaranteed to be co-located on the host server and can share resources. Each pod in Kubernetes is assigned a unique IP address within the cluster, allowing applications to use ports without the risk of conflict.

Producer

A gatherer of event data, such as a SmartConnector or CTH. Typically data from a producer is forwarded to a destination such as a Transformation Hub topic.

R

Root Installation Folder

The root installation folder is the top-level directory that the Transformation Hub, CDF Installer, and all supporting product files will be installed into. The default setting is /opt/arcsight. It is referred to as RootFolder in this document, supporting scripts, and installation materials.

S

Shared Master and Worker Nodes

A configuration where both Master and Worker Node functionality resides on the same host. This is not a recommended architecture for high availability.

SmartConnector

SmartConnectors automate the process of collecting and managing logs from any device and in any format.

T

Thinpool

Using thin provisioning in Docker, you can manage a storage pool of free space, known as a thinpool, which can be allocated to an arbitrary number of devices when needed by applications.

Transformation Hub

Also TH. A Kafka-based messaging service that enriches and transforms security data from producers and routes this data to consumers.

Transformation Hub cluster

The Transformation Hub cluster consists of all Master and Worker Nodes in the TH environment.

V

Virtual IP (VIP)

To support high availability on a multi-master installation, a VIP is used as the single IP address or FQDN to connect to a dedicated Master infrastructure that contains 3 or more master Nodes. The Master Nodes manage Worker Nodes. The FQDN of the VIP can also be used to connect to the cluster's Master Nodes.

W

Worker Nodes

Worker nodes contain Kafka brokers which ingest, enrich and route events from event producers to event consumers. Worker nodes are automatically load-balanced by the TH infrastructure.

Z

ZooKeeper

Used in conjunction with Kafka, a centralized service used to maintain naming and configuration data and to provide flexible and robust synchronization within distributed systems.

Chapter 2: 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.

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

1. **Configure and Install the CDF Installer:** The CDF Installer installs the container management infrastructure. Containerized applications, like the Transformation Hub, 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:** Using the CDF Installer wizard, configure and deploy Transformation Hub to run in the CDF-managed Kubernetes cluster. Then, start the Transformation Hub services.
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 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 Transformation Hub after initial deployment, see the Transformation Hub Administration Guide, available from the [ArcSight support 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 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	ArcMC can be installed before or after Transformation Hub installation.	443, 38080	<ul style="list-style-type: none"> • TLS • FIPS • Client Authentication 	ArcMC Administrator's Guide
SmartConnectors and Collectors	<p>SmartConnectors and ArcMC onboard connectors can be installed and running prior to installing Transformation Hub, or installed after the Transformation Hub has been deployed.</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • TLS • FIPS (SC 7.6+ only) • Client Authentication 	SmartConnector User Guide, ArcMC Administrator's Guide
ArcSight ESM	<p>ESM can be installed and running prior to installing Transformation Hub.</p> <p>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.</p>	9093	<ul style="list-style-type: none"> • TLS • FIPS • Client Authentication 	ESM Administrator's Guide
ArcSight Logger	Logger can be installed and run prior to installing Transformation Hub.	9093	<ul style="list-style-type: none"> • 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. 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

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.

For a complete list of files and file versions to be downloaded, see the Transformation Hub 3.1.0 Release Notes.

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 and the Transformation Hub to a directory like **opt/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, and will have associated signature files (.md5 or .sig) that are used 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 that the downloaded files are authentic by comparing MD5 file signatures, perform the following steps as the root user on the Initial Master Node for downloaded files.

```
cd opt/download/  
  
md5sum <File Name>  
  
cat <File Name>.md5
```

Outputs from each set of compressed installation packages should match their corresponding MD5 signatures. If they do not match, contact Micro Focus Customer Support.

Unarchive Installation Packages

Run the following commands to unarchive your installation packages.

```
cd opt/download  
  
unzip cdf-2019.05.xxxx.zip  
  
tar -xvf transformationhub-3.1.0.xxxx.tar
```

Where **xxxx** is the build number of the file.

Note: Do not uncompress the **arcsight-installer-metadata** file.

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/download/cdf-2019.05.xxxx/`

`opt/download/transformationhub-3.1.0.xxx`

`opt/download/arcsight-installer-metadata.2.1.0.xxxx.tar.gz`

Chapter 3: Installation and Deployment

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](#)

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 one of the local Master Nodes where you downloaded and extracted the installation files as the root user. (In this document, the selected Master Node will be referred to as the Initial Master Node. 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."

```
cd /opt/download/<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> --tmp-folder <your_temp_folder>
```

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

Example:

```
cd /opt/arcsight/download/cdf-2019.05.xxxx
```

```
./install -m /tmp/arcsight-installer-metadata-2.1.0.xxx.tar --k8s-home /opt/arcsight/kubernetes --docker-http-proxy "http://web-proxy.example.com:8080" --docker-https-proxy "http://web-proxy.example.com:8080" --docker-no-proxy "localhost,127.0.0.1,my-vmenv-node1,my-vmenv-node1.example.com,example.com,216.3.128.12" --nfs-server pueas-vmenv-nfs.swinfra.net --nfs-folder /opt/nfs/volumes/itom/itom_vol --ha-virtual-ip 216.3.128.12 --tmp-folder /opt/tmp
```

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 parameters, 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 or IP}:3000**. Log in using **admin** userid and the password you specified during the platform installation. (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**.

Security, Risk & Governance - Container Installer

Container-based security products and components reside in the same cluster and seamlessly work together.

Micro® Focus Container Deployment Foundation (CDF) container management software installs and configures security product application containers.

Release

Version: 2.00.346-master

- 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**.

End User License Agreement

☒ I agree to the [Micro Focus End User License Agreement](#).

☒ I authorize Micro Focus to collect suite usage data. Collection of suite usage data is governed by the [Micro Focus Privacy Policy](#).

- 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 information associated with the product, click the **>** (greater than) arrow. Then, click **Next**.

Capabilities

Security, Risk & Governance - Container Installer comes in various editions with various capabilities.

Select your edition:

Standard

Select your capabilities:

- ☐ > Transformation Hub 3.1.0
- ☐ > Analytics 3.0.0. Prerequisite for ArcSight Investigate and Identity Intelligence
- ☐ > ArcSight Investigate 3.0.0
- ☐ > Identity Intelligence 1.1.0

- On the **Database** page, make sure the **PostgreSQL High Availability** box is *deselected*. This database is not used by capabilities in Open Data Platform (ODP).

Database

Configure the default database for deployment.

☒ **Out-of-the-box PostgreSQL**




A preconfigured PostgreSQL embedded in the same environment as the installed suite.

☐ PostgreSQL High Availability

6. Click **Next**.
7. On the **Deployment Size** page, choose a size for your deployment based on your planned implementation.

Deployment Size

Select the deployment size that fits your environment best.

 <p>Small Cluster</p> <p>Minimum of one Worker Node with 4 Cores, 16GB memory and 50GB disk</p>	 <p>Medium Cluster</p> <p>Minimum of one Worker Node with 8 Cores, 32GB memory and 100GB disk</p>	 <p>Large Cluster</p> <p>Minimum of 3 Worker Nodes with 16 Cores, 64GB memory and 256GB disk</p>
---	---	---

- **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. 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.

Add Master Node

*Host:
Fully qualified hostname or IP address of the master node with a clean supported OS installation.

☒ Ignore warning

*User Name:

*Verify Mode: ☒ Password ☐ Key-based

*Password:

Advanced Settings:

ThinPool Device:
The LVM thinpool device specifies the path of the main Docker devicemapper. For example: /dev/mapper/docker-thinpool-CDF

Flannel IFace:
This is the interface for Docker inter-host communication. The Flannel IFace is required when the nodes have multiple active network interfaces.

SAVE **CANCEL**

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:** Root or **sudo** user name.
- **Verify Mode:** Choose the verification mode as *Password* or *Key-based*, and then either enter your password or upload a private key file. If you choose Key-based, you must first enter a username 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-thinpool1**. 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.

- On the **Connection** page, an external hostname is automatically populated. This is resolved from the Virtual IP (VIP) specified earlier during the install of CDF (**--ha-virtual-ip parameter**), or the Master Node hostname if the **--ha-virtual-ip** parameter was not specified during CDF installation. Confirm the VIP is correct and then click **Next**.

Connection

Enter your load balancer information for accessing the suite user interfaces.

⚠ The default value of the external hostname is the master node hostname for single-master node deployment. For multiple-master node deployment, enter a fully-qualified domain name (FQDN) that is resolved to the virtual IP address when the master nodes are in a single subnet. Enter an FQDN that is resolved to the load balancer host for the master nodes that are in different subnets.

*External Hostname:

*Port:

☐ Use custom certificates

- 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**.

Master High Availability

Select whether the master shall be highly available. When the master is highly available, you will be asked to define two additional master nodes.

- ☒ Make master highly available

The diagram illustrates three master nodes in a Kubernetes cluster. Each node is represented by a vertical rectangle. The first node on the left is light gray and labeled "The first master node". The second and third nodes on the right are blue and labeled "The second master node" and "The third master node" respectively. Each node contains an icon at the top: a server rack for the first node and a document with a pencil for the other two. A faint watermark "not a CERN document" is visible at the bottom of the first node.

- 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.

Add Node

The suite requires the deployment of worker nodes. Please add at least the minimum number of nodes for each category.

☐ Allow suite workload to be deployed on the master node

Please be aware that deploying suite workload on the master nodes is not recommended for production deployments. The installer will skip the node prerequisite checking when deploying suite workload on master nodes.

Moderate performance HW 0/1

As you add Worker Nodes, each Node is then verified for system requirements. The node count progress bar on the **Add Node** page will progressively show the current number of verified Worker Nodes you have

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

Add Node

The suite requires the deployment of worker nodes. Please add at least the minimum number of nodes for each category.

☐ Allow suite workload to be deployed on the master node

Please be aware that deploying suite workload on the master nodes is not recommended for production deployments. The installer will skip the node prerequisite checking when deploying suite workload on master nodes.

Moderate performance HW 1/1



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.

Add Worker Node

Type:

Minimal performance HW

CPU: 4

Memory: 16 GB

Storage: 50 GB

☐ Skip resource check

Please be aware that skipping this installation pre-check may lead to installation or runtime failures!

*Host:

Fully qualified hostname or IP address of the worker node with a clean supported OS installation.

☐ Ignore warning

*User Name:

*Verify Mode

☒ Password
 ☐ Key-based

*Password:

Advanced Settings:

SAVE

CANCEL

Worker Node parameters include:

- **Type:** Default is based on the deployment size you selected earlier, and shows minimum system requirements in terms of CPU, memory, and storage.

- **Skip Resource Check:** If your Worker Node does not meet minimum requirements, select **Skip resource check** to bypass minimum node requirement verification. (The progress bar on the **Add Node** page will still show the total of added Worker Nodes in green, but reflects that the resources of one or more of these have not been verified for minimum requirements.)
- **Host:** FQDN (only) of node you are adding.

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

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

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.

File Storage

The selected suite capabilities require file systems to store various runtime data files. Please configure the required file systems.

> **arcsight-volume (30Gi)**
Keeps state of various container components

> **db-single-vol (10Gi)**
Database single volume

▼ **itom-logging-vol**
Aggregated log volume

File System Type: Self-Hosted NFS ▼

File Server: _____

Exported Path: _____ ▼

> **db-backup-vol**
Database backup volume

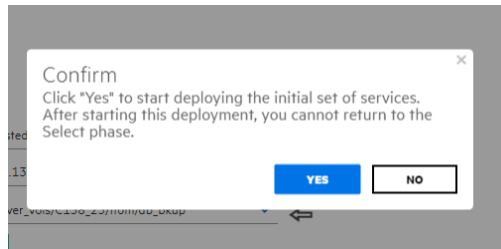
Note: A *Self-hosted NFS* refers to the external NFS that you prepared when you configured an NFS server environment, as outlined in the CDF Planning Guide. Always choose this value for **File System Type**.

CDF NFS Volume claim	Your NFS volume
arcsight-volume	<NFS_ROOT_FOLDER>/arcsight
itom-vol-claim	<NFS_ROOT_FOLDER>/itom_vol
db-single-vol	<NFS_ROOT_FOLDER>/itom/db
itom-logging-vol	<NFS_ROOT_FOLDER>/itom/logging
db-backup-vol	<NFS_ROOT_FOLDER>/itom/db_backup

13. Click **Next**.

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

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



Download Transformation Hub and Core Images to the Local Docker Registry

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

Download Images

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

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

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 CDF images to the Local Docker Registry:

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

```
./uploadimages.sh -u registry-admin -d /opt/download/cdf-core-images
```

3. Run the following command to upload the Transformation Hub images to the local Docker Registry:

```
./uploadimages.sh -u registry-admin -d /opt/download/transformationhub-3.1.0.xxx
```

Note: Prior to running the image upload process by script, you will be prompted for the administrator password previously specified in ["Configure and Install the CDF Installer" on page 20](#).

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.

Check Image Availability



All images are available in the registry.

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

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

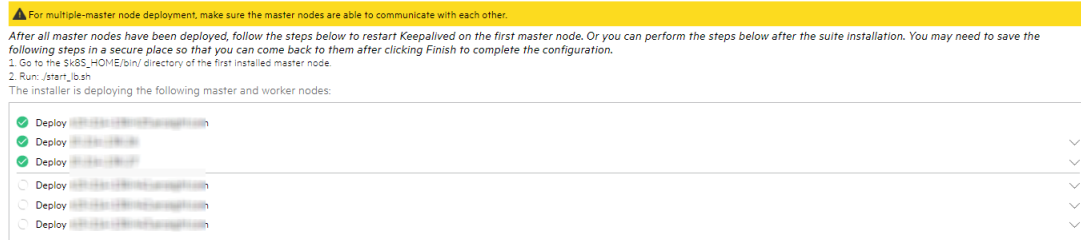
Once verified, click **Next**.

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.

Deployment of Infrastructure Nodes



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.

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

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

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

Deployment of Infrastructure Services

The installer is deploying the following core foundation services:

- ☐ Deploy Heapster Apiserver
- ☐ Deploy Metrics Server
- ☐ Deploy Management Portal
- ☐ Deploy Nginx Ingress
- ☐ Deploy IdM
- ☐ Deploy IdM Postgresql
- ☒ Deploy Suite Database
- ☐ Deploy Suite Configuration Pod

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

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

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

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 a 'Bad Gateway' error. Allow more time for the web UI to start before retrying your login attempt.

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.

Preparation Complete

The container deployment foundation is ready for use.

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 CDF Deployment Disk Sizing Calculator spreadsheet (available from the [Micro Focus software community](#)) for guidance on setting any of these properties. Hover over a value to see a detailed description associated with the configuration property.

The screenshot shows the 'Kafka and Zookeeper Configuration' page with the following settings:

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

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

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

Minimally, synchronize the following properties to the Worker Nodes.

- # of Worker Nodes in the Kafka cluster
- # of Worker Nodes running Zookeeper in the Kafka cluster
- # of Schema Registry nodes in the Kafka cluster
- # of Kafka nodes required to run Schema Registry
- Kafka replication factor (this must be set to '1' for a Single Worker deployment)

For example, if you chose a Single Worker installation, you would set the values of all of these properties to 1.

Log Properties: 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.

- Retention log size for each partition of a Vertica Topic
- Retention log size per Kafka Topic
- Hours to keep Kafka logs

ArcMC Properties: For managing your cluster with ArcMC, you can specify 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 the 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)?
<i>Allow Plaint Text Connections</i>	true	yes	yes
<i>Allow Plain Text Connections</i>	false	no	yes
<i>Client Authentication</i>	true	N/A	yes
<i>Client Authentication</i>	false	N/A	yes
<i>FIPS</i>	true	N/A	yes
<i>FIPS</i>	false	N/A	yes

- 9093 is the endpoint used for TLS, FIPS and Client Authentication. It is always enabled. Note that when Plain Text is disabled, the plain text port is still open (due to pending technical issues). However, the Kafka endpoint will not accept data.
- 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 3.1. 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.

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. For example, labeling a node with the label **kafka:yes** indicates that a Kafka instance will run on that node.

In this step, you will first define labels and then apply them to each node.

Immediately following Transformation Hub deployment, all pods (including **th-kafka**, **th-zookeeper**, **th-kafka-manager**, **th-web-service**, and **th-schemaregistry**) 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:

Note: Upon first login, you are prompted to change the admin password.

1. Login to Management Portal by clicking the link on the **Deployment status** (Configuration complete) page or browsing 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 that you changed to during the first login to port 5443.
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.

kafka:yes

th-processing:yes

th-platform:yes

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

The screenshot shows a user interface for managing nodes and labels. At the top, there is a 'Nodes' section with a '+ ADD' button and a 'REFRESH' button. Below this is a table with two columns: 'Status' and 'Name'. The table contains six rows, each with a green checkmark in the 'Status' column and a node name in the 'Name' column. Below the table is a 'Predefined Labels' section. It features a 'Worker' label and a text input field containing 'kafka:yes' with a '[+]' button next to it.

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

Predefined Labels

Worker [+]

4. Drag and drop each label from the **Predefined Labels** list to each of the Worker Nodes, based on your workload sharing configuration. This will apply the dragged label to the selected node.

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

For Kafka and ZooKeeper, make sure that the number of the nodes you have labeled correctly corresponds to the number of Worker Nodes in the Kafka cluster and the number of Worker Nodes running Zookeeper in the Kafka cluster properties from the pre-deployment configuration page. The default number is 3 for a Multiple Worker deployment.

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

Predefined Labels	
Worker	kafka:yes [-] zk:yes [-] th-platform:yes [-] th-processing:yes [-] <input type="text" value="label name"/> [+]

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

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

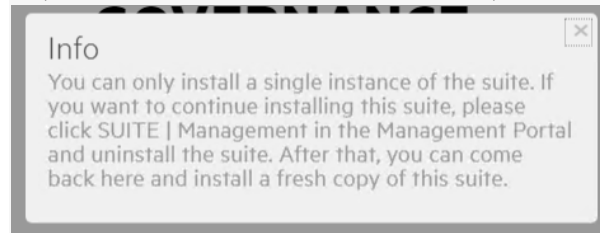
- Pods that have not been labeled will remain in a **Pending** state until labeled.
- For a pod that is not in the **Running** state, you can find out more details on the pod by running the following command:

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

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

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 CDF Management Portal.
2. Click **Suite**
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. Set values for the parameters as needed.
6. For configuring management of Transformation Hub with ArcMC, see [Configuring ArcMC Management of Transformation Hub](#)

7. In addition, under **Stream Processors and Routers**, adjust the '# of CEF-to-Avro Stream Processors' as per group as required.
8. Click **Save**.

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

Note: In order to enable ArcMC management, some configuration of ArcMC is also necessary. For more information, see [Configuring ArcMC Management of Transformation Hub](#)

Additional Steps

In some cases, after flannel pods have been running continuously for some time, the Kafka Manager pod (and others) may terminate abruptly.

To avoid this issue, you should take the following steps after you complete your deployment:

1. Backup existing flannel **yaml**:


```
cp ${K8S_HOME}/objectdefs/flannel.yaml ${K8S_HOME}/objectdefs/flannel.yaml.orig
```
2. Modify the flannel **yaml** file by doing one of the following:


```
vi ${K8S_HOME}/objectdefs/flannel.yaml
```

 file and change both the request and limits memory to 250Mi
 or run:


```
sed -i s/50Mi/250Mi/g ${K8S_HOME}/objectdefs/flannel.yaml
```
3. Delete the flannel file still in use:


```
kubectl delete -f ${K8S_HOME}/objectdefs/flannel.yaml
```
4. Create a new flannel **yaml** file:


```
kubectl create -f ${K8S_HOME}/objectdefs/flannel.yaml
```

Verify the change was made correctly. For each flannel pod run the following command and make sure the memory value is set to 250Mi:

```
kubectl get pod $f_pod -n kube-system -o yaml | grep -A6 resources|grep memory
```

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. For details on adding your Transformation Hub to ArcMC, see [here](#).

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.

Chapter 4: 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	40
• Configuring ArcMC to Manage a Transformation Hub	42
• Configuring Security Mode for Transformation Hub Destinations	44
• Troubleshooting SmartConnector Integration	60
• Configuring Logger as a Transformation Hub Consumer	61
• Configuring ESM as a Consumer	63

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 node 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. On a worker node, by default, events data is stored in: **/opt/arcsight/k8s-hostpath-volume/th/kafka**.
- When a consumer resumes data consumption from Kafka topics, the consumer re-starts where it last left off. No data is lost.
- If a 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 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, will continue.

Configuring ArcMC to Manage a Transformation Hub

ArcMC serves as the management UI for Transformation Hub. In order for ArcMC to manage a Transformation Hub, the Transformation Hub must be added as a managed host to ArcMC. This process will include these steps, 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 **Reconfigure**. A new screen will be opened in a separate tab.
3. Scroll down to the Management Center Configuration section. Then, enter values as described for the following:
 - **Username:** `admin`
 - 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.

Management Center Configuration

Transformation Hub Administrator USER name

Transformation Hub Administrator Password

Management Center host names and ports managing this cluster

Management Center certificates

```
-----BEGIN CERTIFICATE-----
MIID3zCCAAscCQDTIxSNMfhrJDANBgkqhkiG9w0BAQsFAADCBrDELMaKGA1UEBhMC
VVMxEzARBgNVBAgMCkNhbgGImb3JuaWEeEjAQBgNVBAcMCVN1bm55dmFsZTEUME
A1UECgwLTWljcm8gRm9jdXMxFTATBgNVBAzMDFN1cHBvcnQgVGVhbTEBMBkGA1U
AwwSVGVtcG9yYXJ5QXV0aG9yaXR5MSowKAYJKoZIhvcNAQkBFhthcnNOLXN1cHBv
cnRAbWljcm9mb2N1cy5jb20wHhcNMTkwODA5MjAzODM0WhcNMjkwODA2MjA2ODM0
WjCBTELMAKGA1UEBhMCVVMxEzARBgNVBAgMCkNhbgGImb3JuaWEeEjAQBgNVBAc
CVN1bm55dmFsZTEUMBGA1UECgwLTWljcm8gRm9jdXMxFTATBgNVBAzMDFN1cHE
cnQgVGVhbTEkMCIGA1UEAwwbbjE1LTlxNCOxMjktadYyYXJ5c2lnaHQuY29tMSow
KAYJKoZIhvcNAQkBFhthcnNOLXN1cHBvcnRAbWljcm9mb2N1cy5jb20wgwEiMA0G
CSqGSIb3DQEBBAQUAA4IBDwAwggEKAoIBAQC8wCC6b8067dnbcdM6Bf9LqJo76fE
7jaZC/zTPBWusFUWXPtTg6Hdw//xncEKyrotbXWz2wOzZoKwg8G6f67h99j9xoZQ
GqzOawhb14Mbzw91Fk8TCIRdrICPC13Jc7tD3quq6/3Zn3NrhqPOC7OD3BsYBXKW
/FNmtX8HIExVRzdbl1Zyth5dLSofMrJFzjFt0zi9yXA1JfLTuXv+iHMFIDn6tJNg
XZsBe8gi90P/NiYhTY4n2cakc6xtjBhNKcphmR/6ebNmy3BbgfJOod9BvjBJdm/
39Eeq/GpkqQ/D4NDYtJ7w5QdsvBTNNWYfXl462K0lxedGugcRwAjt+2ErAgMBAAEw
DQYJKoZIhvcNAQELBQADggEBAHhVVTTh2q4TTPGbm24w6MdHdQ480+QyHDppAiO:
H7IACboN4OBGap2NHJXtraBC6M6Vi/jN4ruDKbq9EObQRkkG55ZhX4fAjiXvmw0t
wMulkP8XSICLG0d/4ubYyepKHKSVOqIHmMzyRNDuCE1yjCH8luuEAvMwSXnb+i79d
bXo3ZT2I/E15FiW7m0UxvJkcQVLSXaRCVWEMGx2lv3cMVjr6W+rPSkVVFhd/B7ke
Q5heJEqQFMtQALge/XclvzPaixKVXAz9fPtVv0JpxPLdGQp+0870foHADMIzynlo
lg/MBoh3I2OypetW3ojp1+bWBz245kuELm5l6taocmoVJ78=
-----END CERTIFICATE-----
```

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

Retrieve the CDF certificate:

- On the initial master node of the cluster, run the following:
`${K8S_HOME}/scripts/cdf-updateRE.sh`
- Copy the contents of this certificate, from `--BEGIN cert` to `END cert--`, to the clipboard.

Configure ArcMC:

- Log in to the ArcMC.
- Click **Node Management > View All Nodes**.
- 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 a 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:

- Create a new file, **agent.security**, at **<install dir>/current/user/agent** (or at **<install dir>\current\user\agent** on Windows platforms).
- 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
```
- 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_CERT=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_CERT=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.bcfile -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_CERT=/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_CERT} -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_CERT} -alias INTCARoot -keystore ${STORES}/${TH}.truststore.bcfips -storepass ${STORE_PASSWD}
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%INTERMEDIATE_CA_CERT% -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 a 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_CERT=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_CERT=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 Client Authentication enabled. See the *Transformation Hub Deployment Guide* for details.

2.

```
export CA_CERT=/tmp/ca.cert.pem
export INTERMEDIATE_CA_CERT=/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% -alias INTCARoot -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_CERT} -alias
INTCARoot -keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_
PASSWD}
```

On Windows platforms:

```
jre\bin\keytool -importcert -file %STORES%\%INTERMEDIATE_CA_CERT% -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 a 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 Hub node(s).
Unable to test connection to Kafka server: [Failed to update metadata after 30000 ms.]	SmartConnector can resolve the short or full hostname of the Transformation Hub node(s) but can't communicate with them because of routing or network issues.
Unable to test connection to Kafka server: [Failed to update metadata after 40 ms.]	You have mistyped the topic name. (Note the lower value in ms than in other messages.)
Destination parameters did not pass the verification with error [; nested exception is: java.net.SocketException: Connection reset]. Do you still want to continue?	If using SSL/TLS, you did not 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 Tranformation 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 5: Maintaining the Transformation Hub

Administration of the Transformation Hub cluster is performed from the Transformation Hub Kafka Manager Portal, available at **https:<Your high-availability 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 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 uninstall the ArcSight Suite (including Transformation Hub):

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

The pods are shut down progressively and then uninstalled.

To gracefully uninstall, first stop all collectors and Connectors from sending events to Transformation Hub. Next, stop all consumers from receiving events after they have consumed all events from their topics. Then follow the procedure above.

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. In the left navigation page, click **IdM Administration**.
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. As a result, 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.

Upgrading ArcSight Suite

Follow these steps to upgrade your ArcSight suite to the latest version.

Prerequisite

Before proceeding, you should have accepted the browser certificate at least once. If not, you should take these steps now.

To ensure that you have accepted the certificate:

1. In the CDF management portal, select **Suite > Management**.
2. Click **... (Browse)** and then choose **Reconfigure**.
3. Accept the browser certificate.

Process Outline

The complete process of upgrade consists of these steps:

1. Download the upgrade files.
2. Add the new metadata file.
3. Initiate the upgrade process.
4. Upload the offline images to the Docker registry.
5. Finalize and complete the upgrade process.

These steps are explained in detail below.

Download the Upgrade Files

Download the metadata and product images files to a secure network location.

- **Metadata:** `arcsight-installer-metadata-<upgrade version>.<Build number>.tar`
 - **Example:** `arcsight-installer-metadata-2.1.0.9.tar`
- **Offline Images:** `<product>-<upgrade version>.<Build number>.tgz`
 - **Example:** `transformationhub-3.1.0.9.tgz`

After downloading, unpack the offline images tar:

```
tar -xvf <product>-<upgrade version>.<Build number>.tgz
```

Example:

```
tar -xvf transformationhub-3.1.0.9.tar
```

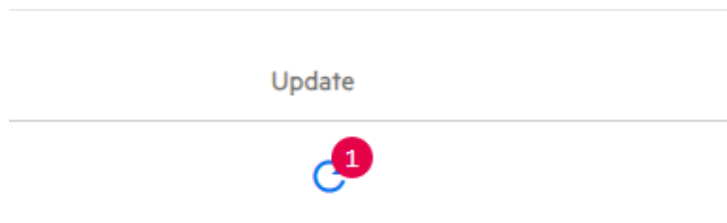
Add New Metadata

1. Copy the file **arcsight-installer-metadata** to the server from which you will be logging into the CDF UI.
2. Log into the CDF management portal from the host where you copied the file in Step 1.
3. Select **Administration > Metadata**.
4. Click **+ Add**.
5. Browse to and select your previously downloaded metadata file (**arcsight-installer-metadata-<upgrade version>.<Build number>.tar**).

Manage Metadata			+ ADD	REFRESH
Suite	Version	Status		
arcsight-installer	2.1.0.4	Not in use		
	2.0.0.22	In use		

Start the Upgrade Process

1. In the CDF management portal, select **Suite > Management**.
2. In the **Update** column, click the number 1 inside the red circle. Then, choose your recently-added metadata file version to initiate the upgrade.



3. On the **Update to** page, click **Next**.
4. On the **Transfer images page**, click **Next**.
5. On the **Import suite images** page, click **more** to verify the filenames of the expected images for the next step.(which will be of the format **3.1.<upgrade version>.<build number>**).

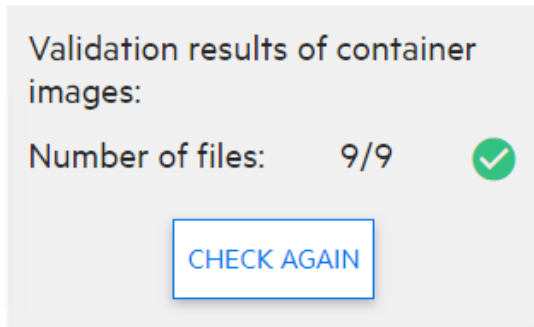
Upload Offline Images

1. `cd {K8S_HOME}/scripts`
Example: `cd /opt/arcsight/kubernetes/scripts`
2. `./uploadimages.sh -u registry-admin -p {your CDF admin password} -d <extracted product directory path>`
Example `./uploadimages.sh -u registry-admin -p Arcsight?123 -d opt/upgrade/transformationhub-3.1.0.9`

Note: Your CDF admin password is the password you were required to change when initially logging into the CDF UI.

Finalize and Complete

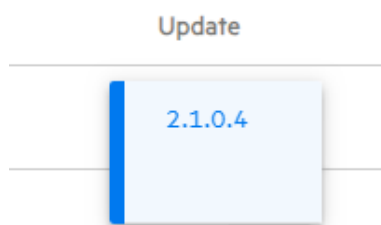
1. In the CDF management portal, return to the **Import suite images** page.
2. Click the **Check again** button to verify that all the required images are available and that the **Next** button is enabled. (There should be 9 of 9 images found and a green icon displayed.) Then click **Next**.



3. On the **Configure storage** click **Next**. The upgrade configuration container is being deployed to the cluster while the next page loads..
4. On the **Product upgrade** page click **Next**. Now the process of upgrading Transformation Hub pods has started. (You can monitor the process by running `kubect1 get pods -n {suite-namespace}`).

The upgrade is complete.

To verify upgrade success, browse to **Suite > Management** and check that the number in the Version column corresponds to your upgrade version.



Note: If you have successfully completed the upgrade image upload and the green icon is not displayed, wait 2 minutes and then click the **Check Again** button. If there are missing files, click **More** to determine the missing files and make sure all images are successfully deployed.

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	<p>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</p> <p>The VIP must be resolved (forward and reverse) to the VIP Fully Qualified Domain Name (FQDN)</p>
--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
<code>--keepalived-virtual-router-id</code>	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.
<code>--kube-dns-hosts</code>	Specifies the absolute path of the hosts file which used for host name resolution in a non-DNS environment. Note: Although this option is supported by the CDF Installer, its use is strongly discouraged to avoid using DNS resolution in production environments due to hostname resolution issues and nuances involved in their mitigations.
<code>--load-balancer-host</code>	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 <code>--load-balancer-host</code> or <code>--ha-virtual-ip</code> arguments.
<code>--master-api-ssl-port</code>	Specifies the https port for the Kubernetes (K8S) API server. The default is 8443.
<code>--nfs-folder</code>	Specifies the path to the ITOM core volume.
<code>--nfs-server</code>	Address of the NFS host.
<code>--pod-cidr-subnetlen</code>	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.
<code>--pod-cidr</code>	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 <code>--service-cidr</code> parameter below) in Kubernetes. The default is 172.16.0.0/16. For the default and allowable values see the CDF Planning Guide.
<code>--registry-orgname</code>	The organization inside the public Docker registry name where suite images are located. Not mandatory. Choose one of the following: <ul style="list-style-type: none"> Specify your own organization name (such as your company name). For example: <code>--registry-orgname=Mycompany</code>. Skip this parameter. A default internal registry will be created under the default name HPESWITOM.
<code>--runtime-home</code>	Specifies the absolute path for placing Kubernetes runtime data. By default, the runtime data directory is <code>\${K8S_HOME}/data</code> .
<code>--service-cidr</code>	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 <code>--pod-cidr</code> .

Argument	Description
<code>--skip-check-on-node-lost</code>	Option used to skip the time synchronization check if the node is lost. The default is true.
<code>--skip-warning</code>	Option used to skip the warnings in precheck when installing the Initial master Node. Set to true or false. The default is false.
<code>--system-group-id</code>	The group ID exposed on server; default is 1999.
<code>--system-user-id</code>	The user ID exposed on server; default is 1999.
<code>--thinpool-device</code>	Specifies the path to the Docker devicemapper, which must be in the /dev/mapper/ directory. For example: /dev/mapper/docker-thinpool
<code>--tmp-folder</code>	Specifies the absolute path of the temporary folder for placing temporary files. The default temporary folder is /tmp .
<code>-h, --help</code>	Displays a help message explaining proper parameter usage
<code>-m, --metadata</code>	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:

- [Create a New CA Certificate \(example\)](#) 74
- [Create a New Intermediate Key and Certificate](#) 79
- [Update the Certificate Set on the Transformation Hub Cluster](#)84

Best Practice: Note that in order to import an intermediate (user) certificate, the Transformation Hub must be deployed (and the user certificate imported), undeployed, and then re-deployed to pick up the certificate change. 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 (example)

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   = <your country code>
stateOrProvinceName_default = <your state or province>
localityName_default  =
0.organizationName_default = <your company name>
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         = <your country code>
stateOrProvinceName_default = <your state or province>
localityName_default        =
0.organizationName_default  = <your company name>
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: Troubleshooting

The following troubleshooting tips may be helpful in resolving issues with your Transformation Hub cluster.

Issue: Installation of Master Nodes Fails

During installation, installation of Master Nodes can fail with the error:

Unable to connect to the server: context deadline exceeded

If this occurs, make sure that your **no_proxy** and **NO_PROXY** variables include valid virtual IP addresses and hostnames for each of the master and worker nodes in the cluster, as well as the NFS server.

Issue: Installation Times Out

During installation, the process may time out with the error:

Configure and start ETCD database

If this occurs, make sure your **no_proxy** and **NO_PROXY** variables include correct Master Node information.

Issue: During **sudo** Installation, Worker Node Fails to Install

During the Add Node phase, if one or more of the worker nodes fails to install and the log shows the following error message:

[ERROR] : GET Url: https://itom-vault.core:8200/v1/*/PRIVATE_KEY_CONTENT_{hostname}_{sudo user}, ResponseStatusCode: 404**

You can take the following steps to rectify the issue:

1. Click **Cancel**. This takes you back to the version selection screen
2. Go through the installation screens again (all previous data is preserved).
3. On the **Add Node** screen, where you added the Worker Node data, remove the worker node which failed by clicking on the **Delete** icon.
4. Click **Add Node** and add the node again.
5. Click **Next** and proceed with the installation.

Issue: Cluster List Empty in Kafka Manager

If cluster list is empty in the Kafka Manager UI, delete the existing Kafka Manager pod and try the UI again after a new Kafka Manager pod is back to the Running state.

Issue: Worker Nodes Out of Disk Space and Pods Evicted

If the worker nodes run out of disk space, causing the pods on the node to go into Evicted status, try one of the following steps:

- Fix the disk space issue by adding an additional drive, or by removing unnecessary files.
- On the the node where the low disk space occurred, run the following command:

```
{install dir} /kubernetes/bin/kube-restart.sh
```

Refer to "Configuring Hard Eviction Thresholds for Worker Nodes" in the ***Transformation Hub Administrator's Guide*** for information on adjusting the eviction threshold.

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 (Transformation Hub 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@microsoft.com.

We appreciate your feedback!