Micro Focus
CDF On-Premises

Software Version: 2020.05

Planning Guide

Document Release Date: July 31, 2020
Software Release Date: July 31, 2020

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

The network information used in the examples in this document (including IP addresses and hostnames) is for illustration purposes only.

ArcSight products are highly flexible and function as you configure them. The accessibility, integrity, and confidentiality of your data is your responsibility. Implement a comprehensive security strategy and follow good security practices.

This document is confidential.

Restricted Rights Legend

Confidential computer software. Valid license from Micro Focus 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.

Copyright Notice

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

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

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

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

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

U.S. Government Rights Section 18.11 is in lieu of, and supersedes, any other FAR, DFARS, or other clause or provision that addresses government rights in computer software or technical data.

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/
Micro Focus Product Documentation	https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

Contents

Overview

The CDF Planning Guide will provide instructions on preparing your on-premises infrastructure environment for security products installed using Micro Focus’ Container Deployment Foundation (CDF) version 2020.05.

> **Note:** *This guide is required for on-premises installations of CDF only.* The CDF preparation requirements for cloud-based (Azure) installation are detailed in the Transformation Hub Deployment Guide, available for download from the Micro Focus community. If you are installing CDF on Azure, you do not need this document.

CDF enables customers to install pre-integrated application capabilities. The distribution unit for software delivery is the container, leveraging the speed and format of the containerized environment. By bundling an orchestration layer to bootstrap and manage the life-cycle of many suite-related containers, CDF supports standardized deployment, built-in upgrades and patching, seamless scaling, and rollbacks.

Several Micro Focus security products run on the CDF platform as a suite of applications. These applications include:

- Transformation Hub
- ArcSight Investigate
- Identity Intelligence
- Analytics (a prerequisite for ArcSight Recon and Identity Intelligence)

For more information about a product's compatibility with this version of the CDF installer (version 2020.05), consult the product's Release Notes, available from the Micro Focus support community.

The hardware recommendations described in this document are general guidelines that may be superseded or extended by requirements specific to each container-based application installed on CDF. You should refer to each container-based application's documentation for any additional requirements.

Chapter 1: Choosing a Deployment Infrastructure

All container-based products and capabilities run within the Container Deployment Foundation (CDF) infrastructure, which comprises Kubernetes and Docker container foundational management. After successfully preparing host system prerequisites as described in this guide, you will be ready to launch the CDF Installer. The CDF Installer deploys and manages upgrades and configurations of container-based security products.

The CDF Installer will validate minimum infrastructure requirements and then configure, install and start the services associated with the security products chosen during the installation process. You can install the security products as a `sudo` user, or optionally, as a `root` user. (For information on granting permissions for installing as a `sudo` user, see Appendix B.)

There are 2 primary deployment configurations, each of which depends on whether the deployment requires high availability or not. While it is recommended that all deployments be highly available, you may decide that development or testing environments don’t necessarily require redundancy and failover.

The number of host systems required will depend on the architecture chosen, types of which are described below.

Note: Appendix A includes a checklist for your use to track your progress implementing your preparation.

About Master Nodes

The Kubernetes master nodes control the cluster, manage its workload and direct communication across the system.

In order to ensure high availability of cluster services, 3 master nodes must be deployed. Deployment of multiple master nodes is strongly recommended for all environments, and required for highly available environments. When deployed in this manner, the cluster will survive a failure of one master node.

Should a single master node be deployed instead of the recommended 3 master nodes, failure of the single master node could cause the entire cluster to become unrecoverable, requiring a complete reinstall and reconfiguration.

Always run the cluster with three master nodes. If only two master nodes are used, and the primary master node is taken offline for maintenance or upgrade, there will only be a single master node available, creating a single point of failure. A failure of the available master node will result in the entire cluster failing, with consequences as described for the failure of a single master node deployment, above.

Adding master nodes after the cluster has been initially deployed is not supported. You must decide before deploying the cluster whether multiple master nodes will be initially deployed. Adding additional master nodes after deployment will require reinstalling the cluster, leading to downtime.

About Worker Nodes

Kubernetes worker nodes run the application components and perform the work in the cluster. A minimum of 3 dedicated worker nodes are recommended for all highly available deployment configurations. Worker nodes can be added or removed from the cluster as needed. Scaling the cluster to perform more work requires additional worker nodes, all of which are managed by the master nodes.

Use of Kubernetes and Docker

Kubernetes automates deployment, scaling, maintenance and management of containerized applications across a cluster of host systems.

Applications running in Kubernetes are defined as “pods”, which groups containerized components. Clusters use 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. Persistent services for a pod can be defined as a volume, such as a local disk directory or a network disk, and exposed by Kubernetes to the containers in the pod to use. A cluster relies upon an external Network File System (NFS) as its shared persistent storage.

Use of Kafka

Kafka is a messaging system to which producers publish messages for subscribers to consume on its scalable platform, built to run on servers. It is commonly referred to as a message broker.

This middleware is used to decouple data streams from processing, translate and enrich event data, and to buffer unsent messages. Kafka improves on traditional message brokers through advances in throughput, built-in partitioning, replication, latency and reliability.

Deployment Architectures

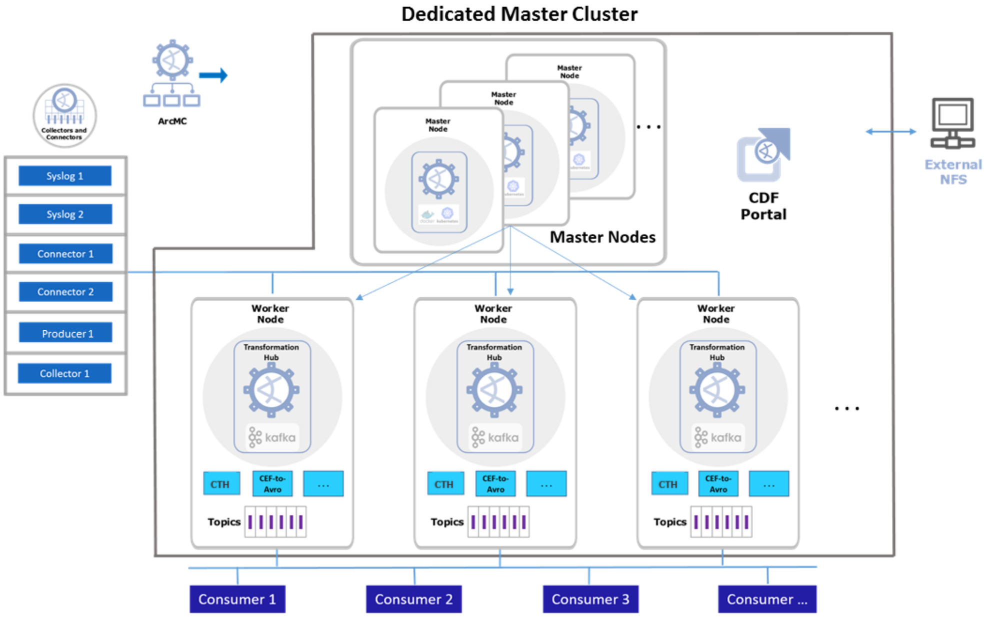
CDF installation supports the following deployment architectures, which are detailed in the following sections:

- Multiple Master and Multiple Worker Nodes
- Single Master and Multiple Worker Nodes
- Shared Master and Worker Node

Multiple Master and Multiple Worker Deployment

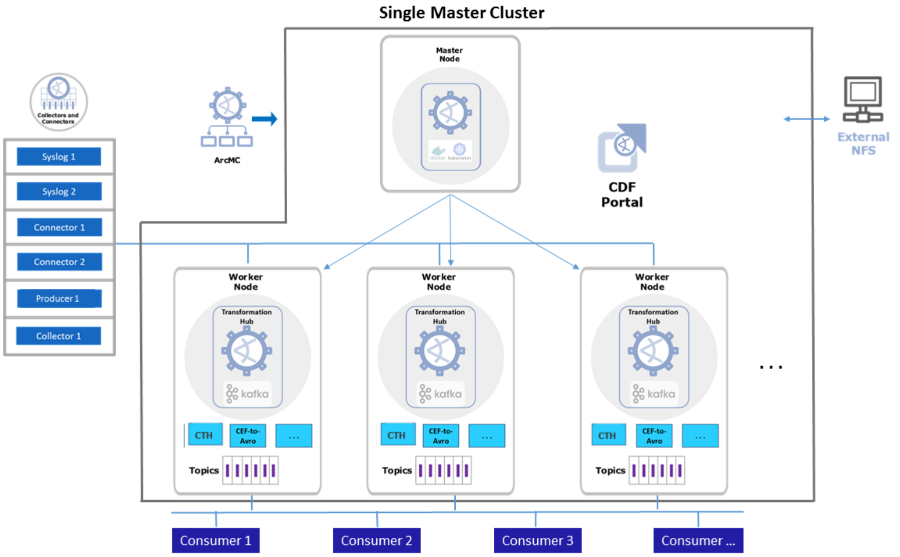
In this deployment, master and worker nodes are dedicated to a specific OS instance. This configuration can be run in development and testing, and it is the recommended configuration for highly available environments. Events are processed by the worker nodes, with failover to another worker node in the event of a Worker failure. There are no single points of failure.

A minimum of 6 physical or VM environments are needed (3 dedicated master nodes and 3 or more dedicated worker nodes), plus a customer-provisioned NFS server, referred to in this documentation as External NFS.



Single Master and Multiple Worker Node Deployment

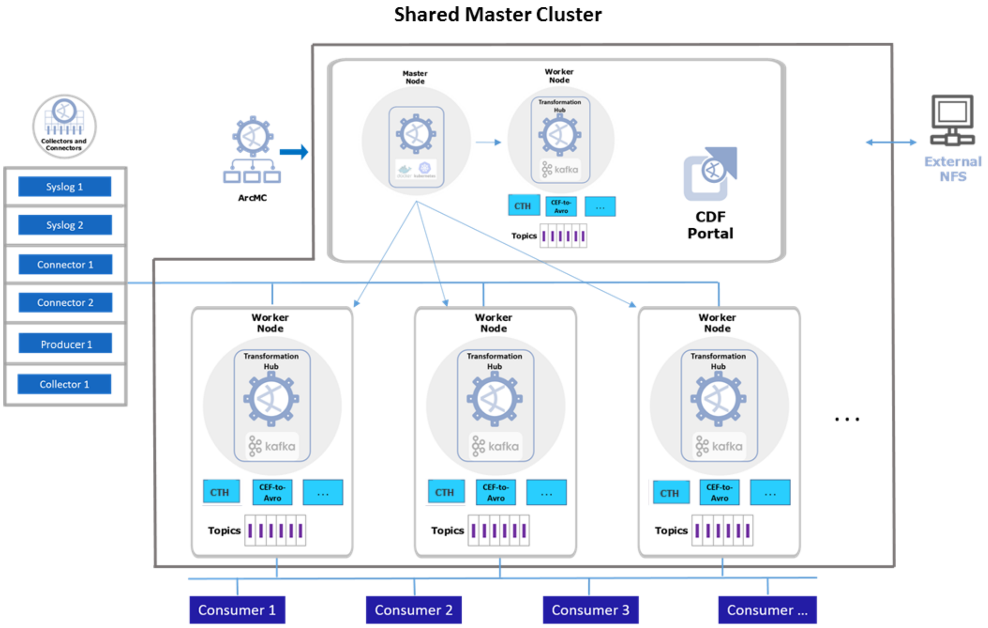
In this deployment, a single master node connects to 3 or more worker nodes. The Master and the worker nodes are dedicated to a specific OS instance. Events are processed by the worker nodes, with failover to another worker node in the event of a worker failure.



> **Note:** The single master node is a single point of failure, and as a result, this configuration is not recommended for high availability (HA) environments (see "About Master Nodes" on page 7).

Shared Master and Worker Node

In this configuration, the master node and one of the worker nodes are co-located on the same host, while supporting additional worker nodes on different hosts.



Note: The single master node is a single point of failure, and as a result, this configuration is not recommended for high availability (HA) environments.

Chapter 2: Prepare Infrastructure for Deployment

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

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

Note: Appendix A includes a checklist for your use to track your preparations.

This chapter contains the following sections:

Implementation Roles and Responsibilities

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

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

The following roles and responsibilities will be needed to properly configure the infrastructure environment.

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

Deployment Considerations and Best Practices

Before starting the installation process, there are several decisions to be made to plan and prepare your infrastructure. Listed below are the considerations on which you will need to decide, and an outline of steps you will follow during this planning and preparation process. Details are explained in later sections of this guide.

Consideration	Best Practices
Host Systems	• Provision cluster (master and worker node) host systems and operating environments, including OS, storage, network, and Virtual IP (VIP) if needed for high availability (HA). Note the IP addresses and FQDNs of these systems for use during product deployment. • The cluster may be installed using a `sudo` USER with sufficient privileges, or, alternatively, may be installed using the `root` USERID. For more information on granting permissions for installing as a `sudo` user, see Appendix B. • Systems must not only meet minimum requirements for CPU cores, memory and disk storage capacity, but also meet anticipated end-to-end events processing throughput requirements. • Master and worker nodes can be deployed on virtual machines. • Since most of the processing occurs on worker nodes, if possible, you should deploy worker nodes on physical servers. • All master nodes should use the same hardware configuration, and all worker nodes should use the same hardware configuration (which is likely to be different from that of the master nodes). • When using virtual environments, please ensure: ○ Resources are reserved and not shared. ○ The UUID and MAC addresses are static and do not change after a reboot or a VM move. Dynamic IP addresses will cause the Kubernetes cluster to fail. • All master and worker nodes must be installed in the same subnet. • Adding more worker nodes is typically more effective than installing bigger and faster hardware. Using more worker nodes also enables you to perform maintenance on your cluster nodes with minimal impact to your production environment. Adding more nodes also helps with predicting costs due to new hardware. • For high availability (HA) of master nodes on a multi-master installation, you must create a Virtual IP (VIP) which will be shared by all master nodes. Prior to installation, a VIP must not respond when pinged. • If a Master and Worker are sharing a node, then follow the higher-capacity worker node sizing guidelines. (Note that this configuration is not recommended for production Transformation Hub environments.

Consideration	Best Practices
Storage	• Available from the Micro Focus support community, the CDF Deployment Disk Size Calculator spreadsheet will enable you to determine your recommended disk storage requirements and other configuration settings based on throughput requirements. Download the spreadsheet to help determine your storage needs. • Create or use a preexisting external NFS storage environment with sufficient capacity for the throughput needed. Guidelines are provided below. • Determine the size and total throughput requirements of your environment using total EPS. For example, if there are 50K EPS inbound, and 100K EPS consumed, then the size would be 150K EPS. (Note: This does not apply to the Identity Intelligence (IDI) product, because IDI measures the number of identities and transactions per day.) • Data compression is performed on the producer side (for example, in a Smart Connector).
Network	• Although event data containing IPv6 content is supported, the cluster infrastructure is not supported on IPv6-only systems.
Security	• Determine a security mode (FIPS, TLS, Client Authentication) for communication between components. **Note:** Changing the security mode after installation may require downtime for uninstalling and re-installing the Transformation Hub.
Performance	• Kafka processing settings for Leader Acknowledgement (ACK) and TLS settings have a significant effect on throughput through the system. If ACK and TLS are both enabled, throughput performance may be degraded by a factor of 10 or more, requiring additional worker nodes to account for the processing overhead. • If CEF events are being transformed to Avro events and being stored in Vertica, consider the potential performance effects of the CEF-to-Avro data transformation, and allow a 20% increase in CPU utilization. This will generally only have a large impact with very high EPS (250K+) rates
Downloads and Licensing	• Ensure you have access to the Micro Focus software download location. You will download installation packages to the Initial Master Node in the cluster. • Ensure you have a valid Micro Focus license key for the software being installed.

Provision and Prepare the Master and Worker Nodes

Provision and then configure the master and worker node operating systems, and ensure that the images meet the operating standards for your enterprise.

- Plan for and request Red Hat or CentOS operating systems for your expected Development, Test and Production environment implementations. Master and worker nodes must meet minimum disk, CPU, memory, network and redundant node requirements based on the expected Events per Second (EPS).
- Deploy master and worker nodes on a minimal Red Hat or CentOS OS instance, plus the additional packages that are specified below.

Supported Operating Systems

Master node and worker node hosts must use the same operating system. Supported operating systems are detailed in the SODP Support Matrix, available from the Micro Focus support community.

Supported File Systems

- EXT4
- XFS (overlay2 recommended for production when adding nodes)
- EMC2 (Supported, but not certified)

Network Identification

• IPv4 (hostnames must also resolve to IPv4 addresses).

• Direct layer 2 connectivity between nodes is required, unless the flannel backend type is changed to vxlan during installation.

• **Static IP addresses and static hostnames:** each node must have a static IP address and static hostnames.

CDF Databases

• PostgreSQL from 9.4.x to 10.6.x (bundled with CDF as an internal database; no user preparation required).

Master Node and Worker Node Sizing Requirements

Sizing requirements for your master and worker nodes will depend on a number of factors.

To determine minimum master and worker node system requirements, download the CDF Planning Disk Sizing Calculator spreadsheet from the Micro Focus support community and compute your requirements.

Secure Communication Between Micro Focus Components

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

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.

When possible, you should set up the other Micro Focus components with the security mode you intend to use *before* connecting them to Transformation Hub.

- Changing to or from TLS or FIPS after the deployment will necessitate system downtime.
- Changing to or from client authentication cannot be performed at all after deployment.

If you do choose to change the security mode (TLS or FIPS) 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 where to find more information on the product. Micro Focus product documentation is available for download from the Micro Focus support community.)

Product	Preparations needed...	TCP Ports	Supported security modes
Management Center (ArcMC) version 2.9.4 or later	Install ArcMC before Transformation Hub installation. See also ArcMC Administrator's Guide.	443, 32080	• TLS • FIPS • Client Authentication
SmartConnectors and Collectors	SmartConnectors and ArcMC onboard connectors can be installed and running prior to installing Transformation Hub, or installed after the Transformation Hub has been deployed. See also *SmartConnector User Guide, ArcMC Administrator's Guide* • 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 later and Transformation Hub.	9092, 9093	• TLS • FIPS (SC 7.6+ only) • Client Authentication • Plain text

Product	Preparations needed...	TCP Ports	Supported security modes
ArcSight ESM	ESM can be installed and running prior to installing Transformation Hub. See also *ESM Administrator's Guide.* Note that changing ESM from FIPS to TLS mode (or from TLS to FIPS) requires a redeployment of ESM. Refer to the ESM documentation for more information.	9093	• TLS • FIPS • Client Authentication
ArcSight Logger	Logger can be installed and run prior to installing Transformation Hub. See also *Logger Administrator's Guide*	9092, 9093	• TLS • FIPS • Client Authentication • Plain text
ArcSight Recon	Install Recon after Transformation Hub installation Communication is between Transformation Hub and Recon database (aka Vertica).	9092, 9093	• Plain text • TLS • Client Authentication (TH-Scheduler) • FIPS (Database only)

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

Network File System (NFS) Requirements

The CDF Installer platform and some components require a customer-provisioned NFS server, referred to in this documentation as an *External NFS server.* External NFS server configuration guidelines can be found here.

Supported NFS Server Versions

- NFSv3
- NFSv4 or NFSv4.1

To determine the version of NFS you are running, on the NFS server, run the command:

```
# nfsstat -s
```

Supported NFS types

• Linux-based NFS

• NetApp

• HPE 3PAR File Persona

• Amazon EFS

Supported Browsers

- Google Chrome version 80 or later
- Mozilla Firefox versions 60, 60 ESR or later

Note: Browsers should not use a proxy to access CDF ports 5443 or 3000 applications, because this may result in inaccessible web pages.

Supported Screen Resolutions

- 1600x900
- 1280x1024
- 1920x1200
- Higher resolutions are also supported.

Supported Languages

The CDF Management Portal UI will inherit the local language from your browser. The following languages are supported.

- English (US + UK)
- French
- German
- Japanese
- Spanish

Note: Products installed using CDF may or may not support these same languages. Consult the product's release notes for details on its supported languages.

File System Requirements

The following table provides a reference of the directories that are required on each of the servers, and the space that is needed. Ensure you use the absolute path for the equivalent directory.

File system/device	Master Node Minimum*	Master Node Recommended*	Worker Node Minimum	Description
`$K8S_HOME`	8 GB	8 GB	N/A	This directory is used for the CDF installation. To specify a customized directory, run the install command with the following parameter: `--k8s-home`

`$K8S_HOME`
(same as `$RUNTIME_CDFDATA_HOME`)
	50 GB	200 GB
		Note: Make sure the total of used disk size plus 200 GB is no more than 80% of the whole system disk space.
			N/A	This directory is for the Kubernetes server, CDF Installer, and containers. The `$K8S_HOME` and `$RUNTIME_CDFDATA_HOME` are the same directory.
				To specify a customized file system, run the install commands with the following parameters:
				`--k8s-home`
				`--runtime-home`

File system/device	Master Node Minimum*	Master Node Recommended*	Worker Node Minimum	Description
`/var`	5 GB	20 GB	20 GB	This directory used is for the CDF build.
`/tmp`	10 GB	N/A	10 GB	This directory is for the CDF build. To specify an alternate `tmp` folder, during installation, run the install command with the following parameter: `--tmp-folder`
`/opt/arcsight/kubernetes/`	2 GB+suite image size	150 GB	50 GB	This directory includes the CDF images and all suite images. The `/offline/suite_images` subdirectory can be removed after uploading suite images.

Note: An asterisk (*) in the column header indicates that this value does not include NFS server space.

Set System Parameters (Network Bridging)

To set the `net.bridge` parameter on each master and worker node:

1. Log in to the node.
2. Run the following command:
   ```
   # echo -e "\nnet.bridge.bridge-nf-call-ip6tables=1\nnet.bridge.bridge-nf-call-iptables=1" >> /etc/sysctl.conf
   ```
3. Run the following commands:
   ```
   # modprobe br_netfilter && sysctl -p
   # echo -e '\nmodprobe br_netfilter && sysctl -p' >> /etc/rc.d/rc.local
   # chmod +x /etc/rc.d/rc.local
   ```
4. Open the `/etc/sysctl.conf` file in a text editor.
5. If installing on RHEL or CentOS earlier than version 8.1, change `net.ipv4.tcp_tw_recycle=1` to `net.ipv4.tcp_tw_recycle=0` if that line exists.
6. If installing on RHEL or CentOS 8.1 or later, remove or comment out this line, if it exists:
   ```
   net.ipv4.tcp_tw_recycle=
   ```
7. Save your changes and close the file.

8. Run this command to apply your updates to the node:
   ```
   # sysctl -p
   ```

Example Files

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

```
        net.bridge.bridge-nf-call-iptables=1

        net.bridge.bridge-nf-call-ip6tables=1

        net.ipv4.ip_forward=1

        net.ipv4.tcp_tw_recycle=0

        kernel.sem=50100 128256000 50100 2560
```

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

```
        net.bridge.bridge-nf-call-iptables=1

        net.bridge.bridge-nf-call-ip6tables=1

        net.ipv4.ip_forward=1

        kernel.sem=50100 128256000 50100 2560
```

Check MAC and Cipher Algorithms

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

- For MAC algorithms: hmac-sha1,hmac-sha2-256,hmac-sha2-512,hmac-sha1-96
- For Cipher algorithms: 3des-cbc,aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-ctr,arcfour128,arcfour256,blowfish-cbc

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

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

Check Password Authentication Settings

If you will use a user name and password authentication for adding cluster nodes during the installation, make sure the PasswordAuthentication parameter in the `/etc/ssh/sshd_config` file is set to "yes". There is no need to check the password authentication setting when you add the cluster nodes using a user name and key authentication.

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

1. Log on to the cluster node.
2. Open the `/etc/ssh/sshd_config` file.
3. Check if the parameter `PasswordAuthentication` is set to yes. If not, set the parameter to yes as below.

```
PasswordAuthentication yes
```

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

```
systemctl restart sshd.service
```

Ensure Required OS Packages Are Installed

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

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

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

To check for prior installation of any of these packages, setup the yum repository on your server and run this command:

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

This command returns an exit status code where:

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

To install a required package, run the following command:

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

Remove Libraries

You must remove any libraries that will prevent ingress from starting by running the following command, and confirm the removal when prompted:

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

System Clock

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

```
# chronyc tracking
```

Install chrony, start the chrony daemon, and verify operation with these commands:

```
# yum install chrony

# systemctl start chronyd

# systemctl enable chronyd

# chronyc tracking
```

Open Port Requirements

The following firewall ports will be opened during the installation process and must be available.

The default policy of INPUT chain must be set to "ACCEPT" to open the firewall or to add required iptables rules. If it is not set to "ACCEPT", contact your IT system administrator to change the policy.

To check the default policy of the INPUT chain, run the following command:

```
iptables -S | grep -- '-P INPUT'
```

To check whether a port is in use, run the following command:

```
netstat -antp | grep :<port_number_to_check>
```

Used by	TCP Ports	Notes
CDF Management	3000, 5443, 5444	Required for CDF setup and management.
Kubernetes	2379, 2380, 3000, 4001, 4194, 5000, 8080, 8088, 8200, 8201, 8285, 8443, 8472, 10250, 10251, 10252, 10256	Required by Kubernetes.
NFS	111, 2049, 20048	
Transformation Hub	2181, 9092, 9093, 32080, 32093, 32081	• Port 9092 is an insecure (plain-text) port. • Port 9093 is used by Kafka and is TLS-enabled. All customer data is secured by TLS.
Transformation Hub Kafka Manager	9999, 10000	The Transformation Hub Kafka Manager uses port 9999 and 10000 to monitor Kafka. These ports must be mutually reachable between all Transformation Hub nodes.
CTH (Connector in Transformation Hub)	32101-32150	ArcSight Management Center communicates with CTH on ports 32101-32150.

By default, ZooKeepers do not use TLS or FIPS to communicate with each other. Their communication is internal-only, and does not include customer data.

Firewall Settings

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

Enable masquerade setting in firewall

You must enable the masquerade settings only when the firewall is enabled. Run the following command on all master and worker nodes to check whether the masquerade setting is enabled:

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

If the returned value is `yes`, then the masquerade setting is enabled.

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

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

Proxy Settings

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

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

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

> **Note:** Incorrect configuration of proxy settings has proven to be a frequent installation troubleshooting problem. To verify that proxy settings are configured properly, on all master and worker nodes, run the following command and ensure the output corresponds to the recommendations.
>
> ```
> echo $http_proxy, $https_proxy, $no_proxy
> ```

If the firewall is turned off, the install process will generate a warning. To prevent getting this warning, the CDF Install parameter `--auto-configure-firewall` should be set to true.

Proxy Settings Example

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

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

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

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

Note: Optionally, in the above line, `swinfra` can be used to escape proxy for all hosts inside that domain.

DNS Configuration

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

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

All master and worker nodes must be configured with a Fully Qualified Domain Name (FQDN), and must be in the same subnet. Transformation Hub uses the host system FQDN as its Kafka `advertised.host.name`. If the FQDN resolves successfully in the Network Address Translation (NAT) environment, then Producers and Consumers will function correctly. If there are network-specific issues resolving FQDN through NAT, then DNS will need to be updated to resolve these issues.

Configuration Notes:

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

Test Forward and Reverse DNS Lookup

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

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

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

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

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

Procedure

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

```
# hostname
```

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

```
mastern
```

```
# hostname -s
```

```
mastern
```

```
# hostname -f
```

```
mastern.yourcompany.com
```

```
# hostname -d
```

```
yourcompany.com
```

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

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

```
 # nslookup mastern
```

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

```
# nslookup 192.168.0.1
```

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

Kubernetes Network Subnet Settings

The Kubernetes network subnet is controlled by the `--POD_CIDR` and `--SERVICE_CIDR` parameters to the CDF Installer.

The `--POD_CIDR` parameter specifies the network address range for Kubernetes pods. The address range specified in the `--POD_CIDR` parameter must not overlap with the IP range assigned for Kubernetes services (which is specified in the `--SERVICE_CIDR` parameter). The expected value is a Classless Inter-Domain Routing (CIDR) format IP address. CIDR notation comprises an IP address, a slash ('/') character, and a network prefix (a decimal number). The minimum useful network prefix is /24 and the maximum useful network prefix is /8. The default value is 172.16.0.0/16. For example:

```
POD_CIDR=172.16.0.0/16
```

The `CIDR_SUBNETLEN` parameter specifies the size of the subnet allocated to each host for Kubernetes pod network addresses. The default value is dependent on the value of the `POD_CIDR` parameter, as described in the following table.

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

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

Change the default `POD_CIDR` or `CIDR_SUBNETLEN` values only when your network configuration requires you to do so. You must also ensure that you have sufficient understanding of the flannel network fabric configuration requirements before you make any changes.

Configure the NFS Server Environment

NFS storage is used by all nodes in the cluster to maintain state information about the infrastructure and to store other pertinent data. In the case of a Dedicated Master deployment having a minimum of 3 master nodes, NFS must run on a highly-available external server.

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

NFS Prerequisites

1. Ensure the following ports are open on your external NFS server: 111, 2049, and 20048
2. Enable the required packages (`rpcbind` and `nfs-server`) by running the following commands on your NFS server

```
# systemctl enable rpcbind

# systemctl start rpcbind

# systemctl enable nfs-server

# systemctl start nfs-server
```

3. The following table lists the minimum required sizes for each of the NFS installation directories.

Directory	Minimum Size	Description
`<NFS_root_DIRECTORY>/itom_vol`	130 GB	This is the CDF NFS `root` folder, which contains the CDF database and files. The disk usage will grow gradually.
`<NFS_root_DIRECTORY>/db`	Start with 10 GB	This volume is only available when you did not choose PostgreSQL High Availability (HA) for CDF database setting. It is for CDF database. During the install you will not choose the Postgres database HA option.
`<NFS_root_DIRECTORY>/db_backup`	Start with 10 GB	This volume is used for backup and restore of the CDF Postgres database. Its sizing is dependent on the implementation’s processing requirements and data volumes.
`<NFS_root_DIRECTORY>/logging`	Start with 40 GB	This volume stores the log output files of CDF components. The required size depends on how long the log will be kept.
`<NFS_root_DIRECTORY>/arcsight`	10 GB	This volume stores the component installation packages.

NFS Directory Structure

To create the NFS directory structure:

1. Login to your NFS server and create the following:
   - A GROUP named `arcsight`, with a GID of 1999
   - A USER named `arcsight` with a UID of 1999
   - An NFS `root` directory at `/opt/arcsight/nfs/volumes`

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

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

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

For example: `mkdir -p /opt/arcsight/nfs/volumes/itom_vol`

Directory	Mount Point Example
`<NFS_root_DIRECTORY>/itom_vol`	`/opt/arcsight/nfs/volumes/itom_vol`
`NFS_root_DIRECTORY>/db`	`/opt/arcsight/nfs/volumes/db`
`<NFS_root_DIRECTORY>/db_backup`	`/opt/arcsight/nfs/volumes/db_backup`
`<NFS_root_DIRECTORY>/logging`	`/opt/arcsight/nfs/volumes/logging`
`<NFS_root_DIRECTORY>/arcsight`	`/opt/arcsight/nfs/volumes/arcsight`

3. The permission setting of each the parent directory and each sub-directory must be recursively set to 755. If it is not, run the following command to update the permissions:

```
# chmod -R 755 <path to shared directory>
```

For example:
`#chmod -R 755 /opt/arcsight/nfs`

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

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

Note: If you use a UID/GID different than 1999/1999, then provide it during the CDF installation in the install script arguments `--system-group-id and --system-user-`

id. In addition, if you are using NetApp with NFSv4 configuration, consider applying stickybits to all <NFS_root_directory> shares with:

```
# chmod g+s
#chmod w+s
```

Export the NFS Configuration

For every NFS volume, run the following set of commands on the External NFS server based on the IP address. You will need to export the NFS configuration with appropriate IPs in order for the NFS mount to work properly. For every node in the cluster, you must update the configuration to grant the node access to the NFS volume shares. On the NFS server, edit the `etc/exports` file and add all the shared volumes to the file.

Here is a sample `etc/exports` file entry for IP address 192.168.1.0, for all of the volumes:

```
/opt/arcsight/nfs/volumes/arcsight 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)

/opt/arcsight/nfs/volumes/itom_vol 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)

/opt/arcsight/nfs/volumes/db 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)

/opt/arcsight/nfs/volumes/logging 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)
   /opt/arcsight/nfs/volumes/db_backup 192.168.1.0/24
   (rw,sync,anonuid=1999,anongid=1999,all_squash)
```

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

```
   exportfs -ra
```

Synchronize the time on the NFS server and the time on the other servers in the cluster.

If you add more NFS shared directories later, you must restart the NFS service.

Testing NFS

1. Create a test directory under `/mnt`.
2. From the command prompt attempt to mount the `nfs`directory on your local system, to `/mnt/nfs`, using the sample commands below (for NFS v3 and v4).

- NFS v3 Test:

```
mount -t nfs 192.168.1.25:/opt/arcsight/nfs/volumes/arcsight /mnt/nfs
```

- NFS v4 Test:

```
mount -t nfs4 192.168.1.25:/opt/arcsight/nfs/volumes/arcsight /mnt/nfs
```

After creating all 5 required volumes, run the following commands on the NFS server:

```
# exportfs -ra
# systemctl restart rpcbind
# systemctl enable rpcbind
# systemctl restart nfs-server
# systemctl enable nfs-server
```

NFS Setup Using a Script

For non-high-availability, single-node shared master and worker node environments only, NFS may be configured with the script `setupNFS.sh,` which is located on the Initial Master Node in `/<ArcsightrootFolder>/kubernetes/scripts` folder.

To run the script:

1. Copy `setupNFS.sh` to the NFS server.
2. Do one of the following:

- If using the default UID/GID run :
  ```
  # sh setupNFS.sh /path_to_volumes/volumes/volume_name
  ```
- If using a non-default UID/GID run:
  ```
  # sh setupNFS.sh /path_to_volumes/volumes/volume_name true <uid> <gid>
  ```

3. Run the following command to restart the NFS service:
   ```
   # systemctl restart nfs
   ```

Disable Swap Space

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

To disable swap space:

1. Log on to the node.
2. Run the following command to disable the swap process:
   ```
   # swapoff -a
   ```
3. Open the `/etc/fstab` file in a supported editor, and then comment out the lines that display "swap" as the disk type. Then save the file.
   For example:
   ```
   #/dev/mapper/centos_shcentos72x64-swap swap
   ```

Create a CDF Installation Directory

For highly available environments, we recommend that you create a CDF installation directory, and then mount a logical volume to the installation directory. The CDF Installer will copy all installation packages to the `/tmp` directory on all master and worker nodes in the cluster.

Before proceeding, ensure there is sufficient disk space for this directory, or override the default directory using the `--tmp-folder` parameter in the CDF Installer command line. Perform these steps on each master and worker node:

1. Run the following command to create the CDF installation directory:
   ```
   # mkdir -p /opt/kubernetes
   ```
2. Add a new hard disk to the host server, and then restart the server.
3. Run the following command to check the disk partition and display the newly added disk:
   ```
   # fdisk –l
   ```
4. Run the following command to format the disk:
   ```
   fdisk <new disk device name>
   ```

For example:

```
# fdisk /dev/sdb
```

5. Enter `n` to create a new partition, and then enter the partition number, sector, and size.
6. Run the following command to create a physical volume:
   ```
   pvcreate <physical device name>
   ```

For example:

```
# pvcreate /dev/sdb
```

7. Run the following command to create a volume group. (Note: do not use a hyphen ‘-‘ in the volume group name, but use an underscore ‘_’ instead.)

   ```
       vgcreate <volume group name> <physical volume name>
   ```

For example, run the following command:

```
vgcreate coretech /dev/sdb
```

8. Run the following command to create a logical volume for the Platform installation. (Note: do not use a hyphen ‘-‘ in the volume group name, use an underscore ‘_’ instead.)
   ```
   # lvcreate -l 100%FREE -n <logical volume name> <volume group name>
   ```

For example, to use 100% of the volume group, run the following command:

```
# lvcreate -l 100%FREE -n coretech_lv coretech
```

9. Run the following command to activate the volume group:
   ```
   # vgchange -a y <volume group name>
   ```

For example:

```
# vgchange -a y coretech
```

10. Run the following command to format the file system:
    ```
    # mkfs -t xfs [logical volume path]
    ```

For example:

```
# mkfs -t xfs /dev/coretech/coretech_lv
```

11. Run the following command to mount the volumes under the directory in which you will install CDF:
    ```
    # mount <logical volume path> <CDF installation directory>
    ```

For example:

```
# mount /dev/coretech/coretech_lv /opt/Kubernetes
```

12. Get the UUID for the volumes by running the following command:
    ```
    blkid
    ```

For example:

```
# blkid
ssh root@192.168.20.51 blkid | grep coretech
ssh root@192.168.20.52 blkid | grep coretech
...
```

```
Master 01: BLKID: /dev/mapper/coretech-coretech_lv: UUID="7356ffbe-0474-
4e19-9547-0e51279db7a2e" TYPE="xfs"

FSTAB: UUID=7356ffbe-0474-4e19-9547-0e51279db7a2 /opt/kubernetes xfs
defaults 1 2

Master 02: BLKID: /dev/mapper/coretech-coretech_lv: UUID="2f10ccf2-049c-
4949-9963-309af66bb1f2" TYPE="xfs"

FSTAB: UUID=2f10ccf2-049c-4949-9963-309af66bb1f2 /opt/kubernetes xfs
defaults 1 2
...
```

13. Make the mount permanent. In the `etc/fstab` file, add a UUID line for each volume using the syntax as shown in the example output:
    ```
     UUID=<UUID>  /opt/kubernetes xfs defaults 1 2
    ```

14. Configure the `K8S_HOME` parameter in the `install.properties` file to use your installation path. The default value is `/opt/kubernetes`.

Next Steps

With your preparation complete, you are now ready to install the CDF Installer and then use it to deploy container-based applications. Such applications may include one or more of the following:

- Transformation Hub
- ArcSight Investigate
- Identity Intelligence

For deployment information, see the Micro Focus Deployment Guide corresponding to your product of choice.

Appendix A: CDF Planning Checklist

Refer to the checklist below to set up your hosts, network, storage, and other prerequisites for installation of CDF. Check off each task as it is verified and completed.

Prerequisite	Description	Completed
Meet system requirements	Memory, CPU, disk space and network connectivity for the expected EPS throughput rates. Download the CDF Planning Disk Sizing Calculator spreadsheet from the Micro Focus support community and compute your requirements.	
DNS connectivity	Ensure DNS connectivity between cluster nodes (master nodes and worker nodes).	
Validate cluster security configuration	Ensure that security protocols are enabled and configured properly for communication between all cluster nodes. The security mode of Producers and Consumers must be the same across the infrastructure. Options are TLS, FIPS and Client Authentication. Changing the security mode after the infrastructure has been deployed will require system down time.	
Create a sudo user	(Optional) Assign permissions to a `sudo` user if the install will use a non-`root` USER.	
Meet file system requirements	Ensure file systems have sufficient disk space.	
Set system parameters	Ensure that network bridging is installed.	
Check MAC and cipher algorithms	Ensure that MAC and cipher minimum requirements are met.	
Check password authentication	If using a USER and password authentication, ensure the PasswordAuthentication parameter is enabled.	
Ensure OS packages are installed	Ensure that all required packages are installed on master and worker nodes and the NFS server. Remove libraries that will cause conflicts.	
Ensure system clocks are in sync	Ensure that the system clock of each cluster master and worker node remains continuously in sync. A network time server must be available (for example, chrony).	
Disable swap space	Optional. For the best performance, disable disk swap space.	

Prerequisite	Description	Completed
Create CDF installation directory	(Optional) In highly available environments, create a CDF installation directory..	
Configure network settings	Ensure host name resolution through DNS across all nodes in the cluster. Infrastructure does not support being installed on IPv6-only networks.	
Configure Kubernetes network subnet	Configure the network subnet for the Kubernetes cluster.	
Configure firewall settings	Ensure that the firewalld.service is enabled on all master and worker nodes in the cluster.	
Configure proxy settings	Should you require internet access, ensure that your proxy and no-proxy settings are properly configured and tested.	
Configure NFS server settings	Ensure that the external NFS server is properly configured and available. NFS utilities must be installed.	
Validate Virtual IP address and FQDN	**Note:** Configuration of a virtual IP address (VIP) applies to multi-master installations only. A single-master installation does not require a VIP. Verify that the VIP address and FQDN shared by all master nodes in the cluster are accessible. The VIP provides high availability for master nodes. The Installation process will test and ping the VIP and try to resolve the FQDN, which is specified with the `--ha-virtual-ip` parameter in the CDF Installer. Should a master node fail, another master node takes over the VIP and responds to requests sent to the VIP.	

Appendix B: Enabling Installation Permissions for a `sudo` User

If you choose to install the Installer as a `sudo` user, the `root` user must grant non-`root` (`sudo`) users installation permission before they can perform the installation. Please make sure the provided user has permission to execute scripts under temporary directory `/tmp` on all master and worker nodes.

There are two distinct file edits that need to be performed: first on the Initial Master Node only, and then on all remaining master and worker nodes. These file edits are detailed below.

In addition, prior to CDF installation, the `CDF-updateRE.sh` script must be modified in order to install CDF as a `sudo` user.

Edit the `sudoers` File on the **Initial Master Node (only)**

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

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

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

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

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

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

- Replace the `<CDF_installation_package_directory>` with the directory where you unzipped the installation package. For example, `/tmp/cdf-2020.05.0xxx`.
- Replace `<K8S_HOME>` with the value defined from a command line. By default, `<K8S_HOME>` is `/opt/arcsight/kubernetes.`

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

```
%wheel ALL=(ALL) ALL

cdfuser ALL=NOPASSWD: CDFINSTALL

Defaults: <username>!requiretty

Defaults: root !requiretty
```

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

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

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

4. Save the file.

Edit the `sudoers` File on the **Remaining Master and Worker Nodes**

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

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

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

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

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

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

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

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

```
%wheel ALL=(ALL) ALL

cdfuser ALL=NOPASSWD: CDFINSTALL

Defaults: <username>!requiretty

Defaults: root !requiretty
```

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

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

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

4. Save the file.

Repeat the process for each remaining master and worker node.

Modify the `cdf-updateRE.sh` Script

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

The `cdf-updateRE.sh` script is used in installation and other utility operations in CDF and CDF-based products (such as Transformation Hub). In order to install CDF, the script must be modified before you install CDF, as follows:

1. In the location where you unzip the installer archive, modify the script `<unzipped CDF directory>/scripts/cdf-updateRE.sh` file in a text editor as follows:
   - Comment out the line containing the text `exit 1`
   - Add the following line inside the `if` block:
     `export K8S_HOME=<install directory>`

Example:

```
if [[ -z "${K8S_HOME}" ]]; then

echo "K8S_HOME not set. If running on fresh installation, please use new
shell session"

# exit 1

export K8S_HOME=/opt/arcsight/kubernetes

fi;
```

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

Installing Transformation Hub Using the sudo User

After completing the modifications to the `sudoers` files as described above, perform the following steps.

1. Log in to the Initial Master Node as the non-`root` `sudo` user to perform the installation.
2. Download the installer files to a directory where the non-`root` `sudo` user has write permissions.
3. Run the CDF installer using the `sudo` command (for more details, refer to the *Transformation Hub Deployment Guide*, available from the Micro Focus support community).

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 system, click the link above and an email window opens with the following information in the subject line:

Feedback on Planning Guide (CDF On-Premises 2020.05)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to arcsight_doc@microfocus.com.

We appreciate your feedback!

Glossary

A

ArcMC

The ArcSight central management console.

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)

CTH features enable 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.

Containerization

Application containerization is an OS-level virtualization method used to deploy and run distributed applications without launching an entire virtual machine (VM) for each app. Multiple isolated applications or services run on a single host and access the same OS kernel.

CTH

Connector in Transformation Hub (CTH). A feature where SmartConnector technology operates directly in Transformation Hub to collect data.

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 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.

kubectl

The Kubernetes command line management tool. For more information on kubectl, see https://kubernetes.io/docs/reference/kubectl/overview/

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 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 Nodes reside on the same hosts. 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

Transformation Hub

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

Z

ZooKeeper

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