Micro Focus Security
ArcSight Event Broker

Software Version: 2.21

Deployment Guide

Document Release Date: September 25, 2018
Software Release Date: July 2018

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notice

Trademark Notices

Support

Contact Information

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

Revision History

Date	Description
July 20, 2018	Initial release of this document.
September 25, 2018	Revised document with multiple improvements, sections on labeling nodes, editing properties file, installing on XFS file system.

Contents

Chapter 1: Event Broker overview

This document describes how to deploy ArcSight Event Broker using the ArcSight Installer.

About ArcSight Event Broker

ArcSight Event Broker centralizes event processing and enables event routing, which enables you to scale your ArcSight environment, and opens ArcSight event data to third-party solutions. Event Broker enables you to take advantage of scalable, highly available, multi-broker clusters for publishing and subscribing to event data. Event Broker integrates with ADP ArcSight SmartConnectors and Collectors, Logger, ESM, and Investigate; can be managed and monitored by ArcSight Management Center (ArcMC); and is foundational for using ArcSight ADP products.

- After you install and configure Event Broker, you can use ADP ArcSight SmartConnectors and Collectors to publish security events or raw syslog data, and subscribe to that security event data with ADP Logger, ArcSight ESM, Apache Hadoop, or your own custom consumer.
- Event Broker supports both Common Event Format (CEF) 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. Note, however, that although it supports IPv6 event content, Event Broker does not support installation on IPv6 only systems.
- Event Broker manages the distribution of events in topics to which consumers can subscribe. There are three default topics you can configure as your destinations:
  - *eb-cef:* accepts CEF event data.
  - *eb-esm:* accepts binary security events, which is the format consumed by ArcSight ESM.
  - *eb-con-syslog:* if you are using the Connector in Event Broker (CEB) feature, you send syslog data to this topic using a Collector.

  In addition, you can create new custom topics to which your SmartConnectors can connect.

Event Broker features and functionality are explained in detail in the Event Broker Administrator's Guide, available from the ArcSight documentation community on Protect724.

Deployment process outline

The complete process of deploying Event Broker includes these steps:

1. **Deployment Preparation.** Prepare and provision your network and dedicated hosts.
2. **Install the ArcSight Installer.** The ArcSight Installer installs the platform (which includes Docker and Kubernetes) on top of which the container-based Event Broker and Investigate applications run.
3. **Deploy Event Broker.** You can deploy Event Broker using online installation, where you download the product images from the online Docker Hub, or offline deployment, where you download the product images from the ArcSight software entitlement site.
4. **Licensing:** Install your permanent license, to ensure continuity of functionality and event flow.

Each of these steps is explained in the following sections.

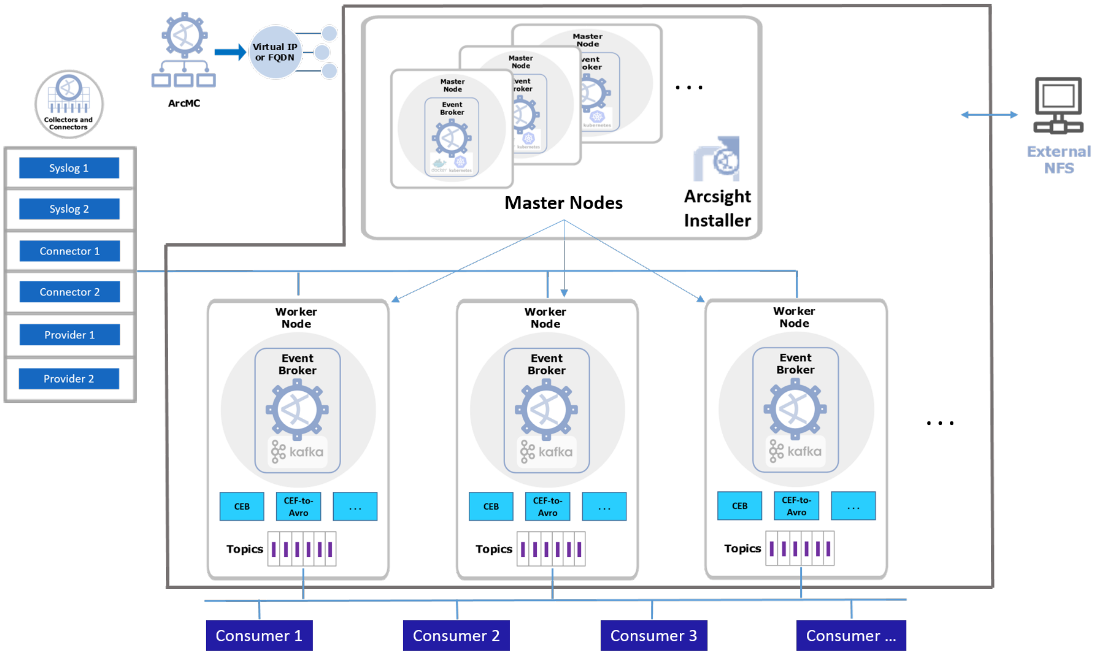
For detailed instructions on the operation and management of Event Broker after initial deployment, see the Event Broker Administration Guide, available from the ArcSight support community at Protect724.

Deployment topology planning

You can choose from multiple options when planning your deployment topology. Each option has benefits and disadvantages.

- **Multi-master:** This topology requires 3 master nodes and a highly-available external NFS server.This enables the cluster to continue function if one master node fails.
- **Single master:** This topology requires 1 master node and an internal or external NFS server. It does not provide high availability or failover of the master node. (This is not a supported production configuration, except for production installations that were in production before April 2018.)

Multi-master deployment topology



In a multi-master cluster deployment, there are three master nodes and three or more worker nodes in the cluster.

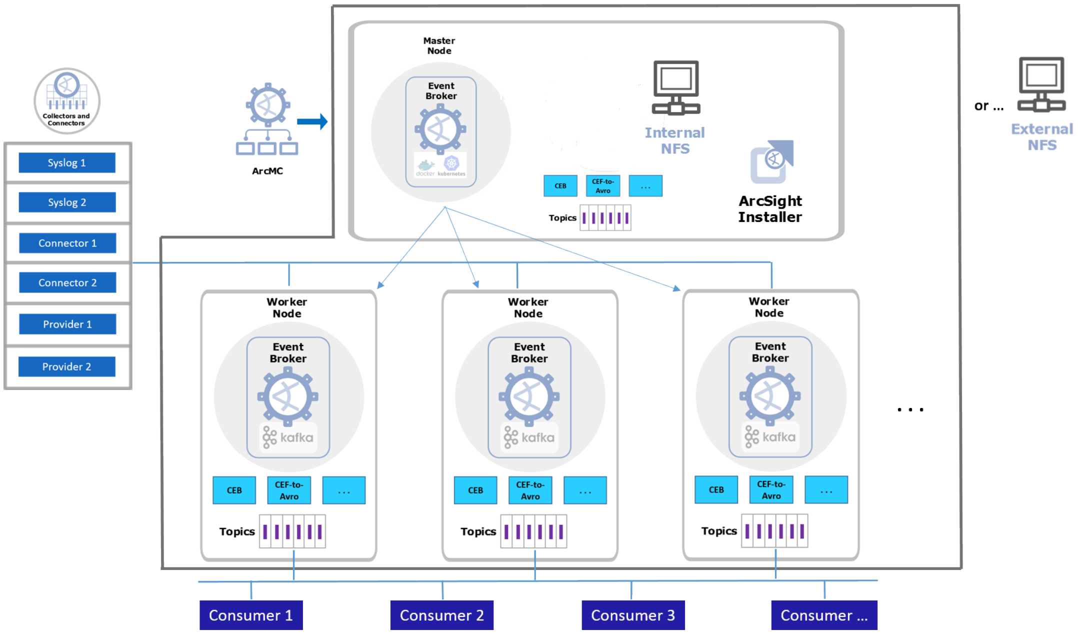
- **Master nodes:** Three master nodes that host ArcSight Installer, Event Broker Web Service, and the Kubernetes control plane containers. These nodes collectively provide high-availability and failover if one master node goes down. Application clients connect to the services using a Virtual IP address or FQDN. The request is directed to one of the master nodes. The FQDN is used when specifying the cluster host name in any configurations. ArcMC connects to the master nodes using the Virtual IP address or FQDN that one of the master nodes listens on.
- **Worker nodes:** Three or more systems that run Kafka and ZooKeeper pods. Connectors and Collectors connect directly to the worker nodes to send events.

Your infrastructure needs to include an external NFS server (hereafter referred to as "external NFS server") to provide a persistent store for the Event Broker cluster. Note that the Kafka topics are persisted locally on the Kafka nodes, not on the NFS server. The NFS server is recommended to be deployed in High Availability configuration to avoid making it a single point of failure.

A virtual IP address (VIP) in the same subnet as the Event Broker cluster must be allocated for use by the cluster. This VIP must have an associated FQDN, with correct forward and reverse DNS lookups.

This deployment topology provides high availability or failover of the master nodes. One master node can go down, and the cluster will continue to function. Two of the three master nodes must continue to run for a fully functional cluster. For production environments, it is strongly recommended that you choose the multi-master deployment topology.

Single master deployment topology



In a single master cluster deployment, there are 4 or more nodes in the cluster.

- **Master node**: One dedicated master node that runs ArcSight Installer and Event Broker Web Service container.
- **Worker nodes:** Three or more systems that run all other Event Broker pods including Kafka and ZooKeeper.
- SmartConnector and Collectors connect directly to the worker nodes to send events.
- ArcMC connects to the master node.
- The cluster uses either an internal or external NFS server to store persisted data for ArcSight Installer, Event Broker, and Investigate (if installed). Note that the Kafka topics are persisted locally on the Kafka nodes, not on the NFS server.
- This deployment topology does not provide high availability or failover of the master node or NFS service.

Note: Single-master node deployment configuration is supported for production deployments which were in production before April 2018, using EB 2.11. Support for those deployments will be discontinued in the near future. If you have such a deployment, ArcSight recommends to start planning your migration to a multi-master deployment as soon as possible.

Supported and unsupported deployment layouts

Recommended deployment layout

Dedicated masters and external NFS

Plan for 3 separate, dedicated master nodes; an external NFS server with 200 GB storage or more deployed in HA configuration; and 3 or more separate dedicated worker role nodes (where the Kafka topics reside).

This layout is robust and supports the best availability of event flows. The dedicated master nodes use less resources than do the worker nodes.

Dedicated master nodes

Dedicated master nodes provide the following benefits:

- Improved resilience with decoupling master nodes from workers.
- Reduced risks during maintenance and future upgrades.
- Simplified backup and less intrusive recovery.
- Independent sizing of master and worker nodes.

External NFS

Deploying an external NFS server eliminates single point of failure vulnerability when a master node with an internal NFS server fails.

Deployment layouts not recommended

Because of the mission-critical nature of Event Broker and the need for robustness in centralized data delivery, ArcSight does not recommend the following deployment layouts for production:

- Kubernetes master and Kafka worker sharing the same host
- Internal NFS server

Deployment layouts not supported for production environments

The following deployment layouts are not supported for production environments.

- Single-master cluster installation

  > **Note:** Any existing support for this configuration will be discontinued in the near future; please plan accordingly.

- OpenStack
- An IPv6 network

Grandfathering Event Broker 2.1x production environments

A production deployment of Event Broker 2.1x can be upgraded to version 2.21 but only in the same single-master deployment layout.

Chapter 2: Prepare systems for deployment

This section discusses the following preparatory steps to take before deployment.

System requirements

The installation of the ArcSight Installer platform has the following system requirements.

- **Disk Space:** Required disk space for each master node and worker node is as follows:

Partition	Master Node	Worker Node	Notes
`/opt`	200 GB	At least 150 GB **Note:**The actual storage needed for Kafka exceed this requirement by significantly more and is dependent on the event rate, event size and the event retention policy.	ArcSight Installer must be installed under the `/opt` directory.
`/var`	10 GB	10 GB	
`/root`	N/A	10 GB	

- **Operating System:** Supported operating systems are discussed in the ADP Support Matrix, available from Protect724.
- **Node Sizing:** Information on Event Broker node sizing requirements can be found in the Event Broker Administrator's Guide.
- The file system type must be ext4.
- All commands in the installer process must run as a user with root privileges.
- The following packages need to be installed using `rpm/yum` on the master and all workers:

```
yum install -y unzip nfs-utils libseccomp libtool-ltdl httpd-tools
conntrack-tools
```

- Java 8 JRE (version 1.8, update 131 or later) must be present on the master node. Update or install the package using `yum` or `rpm` to ensure the minimum version is installed.

Set up the Event Broker nodes (master and worker)

Provision your master and worker nodes.

Set up the systems in the cluster

Master Nodes: Depending on your deployment use case (production, testing), select one of the supported deployment topologies (for example, for production, dedicated masters, and workers)

Make sure that each system meets the minimum system requirements for a master node. The recommended configuration for a production Event Broker is 3 dedicated master nodes.

Worker Nodes: set up the systems that will function as the worker nodes. Make sure that each system meets the minimum system requirements for a worker node.

Enable password-less SSH on all systems in the cluster

> **Note:** SSH access needs to be enabled only to perform a few operations (such as installation, upgrade, or adding a new cluster node). If required by your security policy, you can disable password-less SSH access after these operations are complete.

To enable the master nodes to communicate with worker nodes over SSH during installation, upgrade, or adding a node, generate a key pair on any master node and copy the public key to every other master and worker node.

To generate the key pair, run the following command on the master node where you will execute the ArcSight Installer installation script:

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

On the same master node, run the following command to copy the generated public key to each of the other master and worker nodes in the cluster:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@<node_fqdn_or_ip>
```

Prepare logical volumes on cluster nodes (optional)

Follow the steps below on each master and worker node in the cluster to ensure that you have enough logical volumes for the ArcSight Installer platform installation. You can choose any volume group name, logical volume name and disk location address for your installation, according to your system's architecture.

Note: This is an optional step.

Prepare a physical host for the Kubernetes cluster nodes, which meets the system requirements.

1. Create a volume group by running the following command:

```
vgcreate [volume group name] [logical volume name]
```

For example: `vgcreate core-platform /dev/sdb`

2. Create a logical volume for the ArcSight Installer installation by running the following command:

```
lvcreate -l 100%FREE -n [logical volume name] [volume group name]
```

For example, to utilize 100% of the volume group:

```
lvcreate -l 100%FREE -n mylv core-platform
```

3. Activate the volume group by running the following command:

```
vgchange -ay [volume group name]
```

For example:

```
vgchange -ay core-platform
```

4. Format the logical volume with the ext4 file system by running the following command:

```
mkfs.ext4 [logical volume path]
```

For example: `mkfs.ext4 /dev/core-platform/mylv`

Note: If using the XFS file system, use the command `mkfs.xfs [logical volume path]` instead.

5. Mount the volumes under the folder in which you will install the ArcSight Installer by running the following command:

```
mount [logical volume path] [platform installation folder]
```

For example:

```
# mount /dev/core-platform/mylv /opt/arcsight/kubernetes
```

Plan the cluster's security configuration

Before installing Event Broker, determine which security mode you want for communication between ArcSight components. The security mode of consumers and producers connected to Event Broker must be the same as that set for Event Broker. Set up the other ArcSight components with the security mode you intend to use before connecting them to the Event Broker. Changing the security mode after Event Broker has been deployed will require system down time. If you do need to change the security mode after Event Broker deployment, see the *Event Broker Administrator's Guide*.

Product	**Preparations needed**	**Open ports**	**Supported security modes**	**Where to find more information**
ArcMC	Install ArcMC before Event Broker installation.	38080	• TLS • FIPS • ClientAuth	*ArcMC Administrator's Guide*
ArcSight SmartConnectors and Collectors	ArcSight SmartConnectors and ArcMC onboard connectors can be installed and running prior to installing Event Broker. FIPS mode setup is not supported between SmartConnector version 7.5 and Event Broker. TLS and ClientAuth are the only security modes supported between SmartConnector version 7.5 and Event Broker. FIPS mode is supported between Connectors version 7.6 and above and Event Broker.	9093	• TLS • FIPS • ClientAuth	*SmartConnector User Guide* *ArcMC Administrator's Guide*
ArcSight ESM	ArcSight ESM can be installed and running prior to installing Event Broker. Note that changing ESM from FIPS to TLS mode requires a redeployment of ESM; see the ESM documentation for more information.	9093	• TLS • FIPS • ClientAuth	*ESM Installation Guide* *ESM Administrator's Guide*
ArcSight Logger	ArcSight Logger can be installed and running prior to installing Event Broker.	9093	• TLS • FIPS • ClientAuth	*Logger Administrator's Guide*

Prepare a Virtual IP for High Availability (multi-master only)

A virtual IP address (VIP) is an IP address shared by all master nodes in a cluster deployed in a multi-master configuration. The VIP is used for redundancy by providing failover in a multi-master configuration. If a master node goes down, another master node takes over the VIP and responds to requests sent to the VIP.

Ask your network administrator to configure a VIP as follows:

- Allocate a free IP address in the *same subnet* as the cluster node.
- Create a fully qualified domain name (FQDN), and bind the IP address to the FQDN.

You will use the IP address as the HA VIP and the FQDN as your Kubernetes cluster host name.

In a multi-master cluster deployment, you will launch the applications (for example, ArcSight Installer or Investigate) using the Virtual IP, rather than the master node IP. The port remains the same (port 5443 for ArcSight Installer and default port 80 for Investigate).

Configure the network settings

To prepare your network for the deployment process, take the following steps.

- Ensure host name resolution through DNS across all nodes in the cluster, including correct forward and reverse DNS lookups. Host name resolution must not be performed through `/etc/hosts.`

> **Note:** Hostnames must comply with DNS-1123; that is, consist of lower case alphanumeric characters, '-' or '.', and must start and end with an alphanumeric character. (For example, `ebmaster1.example.com`, regex used for validation is '[a-z0-9]([-a-z0-9]*[a-z0-9])?(\.[a-z0-9]([-a-z0-9]*[a-z0-9])?)*')

- Ensure that all master and worker nodes are configured with an FQDN (fully-qualified domain name), and are in the *same subnet*.

> **Note:** Event Broker does not support installation on IPv6-only networks. However, it does support ingestion of event data that contains both IPv4 and IPv6 address.

- Event Broker uses the host system FQDN for Kafka `advertised.host.name`. If FQDN resolves in the 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.
- If you are performing a containers download, enable Internet access in order to download the product container images.

Configure firewall settings

Make sure that the `firewalld.service` is enabled and running on all nodes before running the `arcsight-installer-master.sh` and `arcsight-installer-add-node.sh` scripts. The following firewall ports will be opened during the installation process.

Used by	**Port**
ArcSight Installer	5443
Kubernetes	2379, 2380, 3000, 4001, 4194, 5000, 8080, 8088, 8200, 8285, 8443, 10248-10252, 10255

NFS	111, 2049, 20048, 37189 Note: NFS ports are used only in clusters that are configured to use an internal NFS server, which is not recommended for production.
Event Broker	2181, 9092, 9093, 38080, 39000, 39093, 32181
CEB	39001-39050
CAdvisor	4194

Avoiding Possible Conflicts with Network Ranges

The ArcSight Installer configures firewall settings during setup (assuming `firewalld.service` is up and running) on both Kubernetes master and worker nodes.

The Installer will use the following network ranges by default:

- 172.16.0.0/16: Subnetwork of 65,536 addresses for operation of Kubernetes pods with containers running in them. Each cluster node uses a /24 subnetwork from that range, with each pod having its own IP address; this allows for a maximum of 254 pods per cluster node.
- 172.30.78.0/24: Subnetwork of 256 addresses for operation of Kubernetes services, including internal Kubernetes DNS service, located on pod 172.30.78.78.

For best results, make sure your network is conflict-free with the /16 and /24 ranges of addresses. If those are occupied or inaccessible due to network configuration, make sure to utilize another range by making corresponding changes to `POD_CIDR` and `SERVICE_CIDR` parameters in the `arcsight-installer-master.sh` script before executing it.

Configure an NFS Server

The ArcSight Installer platform and ArcSight products require an NFS server (v3 or v4) to operate. You can choose to use an external NFS server or use the default internal NFS server.

External NFS server

In the case of a multi-master deployment (3 master nodes), NFS should be run on an external server which is highly available. This configuration is recommended for production environments.

The following setup procedure assumes:

- 3 master nodes with IP addresses: 10.1.2.11 - 13
- 3 worker nodes with IP addresses: 10.1.2.21 - 23
- An NFS share root directory of (by default)`/opt/arcsight/nfs/volumes`. To specify a different directory, create the directory, and then set the --`NFS_FOLDER_ROOT` parameter to the value of the NFS folder root when the installer script is run.

To set up the external NFS share:

1. Log in to the external NFS server as root or as a sudo user.
2. Run the command
   `rpm -qa|grep rpcbind`
   to make sure that the `rpcbind` package is installed on the host.
   If the package is not already installed, run the following command to install it:
   `yum install rpcbind`
3. Run the following command to install the NFS server:
   `yum install -y nfs-utils`
4. Run the following commands to enable the `rpcbind and nfs-server` services:
   ```
   systemctl enable rpcbind
   systemctl start rpcbind
   systemctl enable nfs-server
   systemctl start nfs-server
   ```
5. On the NFS server, run the following commands to create NFS share directories for Event Broker and Investigate. If you do not plan to deploy Investigate on this cluster, do not run the commands specific to Investigate.
   ```
   mkdir -p <NFS_server folder root path>/nfs/volumes/itom/core
   mkdir -p /<NFS_server folder root path>/nfs/volumes/itom/db
   mkdir -p <NFS_server folder root path>/nfs/volumes/eventbroker
   mkdir -p <NFS_server folder root path>/nfs/volumes/investigate
   ```
6. Add a group (GID=1999) and user (UID=1999):
   ```
   groupadd -g 1999 eventbroker
   useradd -g 1999 -u 1999 eventbroker
   ```
   Note: Ignore this step if the GID and UID already exist.
7. Set the owner/group for the NFS share directories:
   `chown -R 1999:1999 <NFS_server folder root path>`
8. In the `/etc/exports` file, add the following 4 lines (with no line breaks):
   ```
   /opt/arcsight/nfs/volumes/itom/core 10.1.2.11
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.12
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.13
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.21
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.22
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.23
   (rw,sync,anonuid=1999,anongid=1999,all_squash)

   /opt/arcsight/nfs/volumes/itom/db 10.1.2.11
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.12
   ```

```
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.13
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.21
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.22
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.23
(rw,sync,anonuid=1999,anongid=1999,all_squash)

/opt/arcsight/nfs/volumes/eventbroker 10.1.2.11
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.12
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.13
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.21
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.22
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.23
(rw,sync,anonuid=1999,anongid=1999,all_squash)

/opt/arcsight/nfs/volumes/investigate 10.1.2.11
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.12
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.13
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.21
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.22
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.23
(rw,sync,anonuid=1999,anongid=1999,all_squash)
```

9. Execute the following commands:
   ```
   systemctl restart rpcbind
   systemctl restart nfs-server
   exportfs -ra
   ```
10. Run the following, and verify all IP addresses and directories are correct, before proceeding to installation.
    ```
    exportfs
    ```

IMPORTANT: During ArcSight Installer installation, the following arguments must be passed to the `arcsight-installer-master.sh script:`

```
--NFS_SERVER=<nfs_server_IP address_or_hostname>
--NFS_FOLDER_ROOT=<root_nfs_folder>
```

where:

- `NFS_SERVER` is the IP address or hostname of the external NFS server
- `NFS_FOLDER_ROOT` is the root directory for the NFS shares used by Event Broker, such as `/opt/arcsight/nfs/volumes` in the example above.

Internal NFS server

The ArcSight Installer can create a default internal NFS server on the master node for shared use by the ArcSight Installer and products. To configure this during the installation, you will use the argument `--NFS_SERVER=internal` when you install the ArcSight Installer.

Note: Use of the default internal NFS server is only recommended for *non-production* deployment environments. For a production environment with multiple master nodes, use an external NFS server.

Some security hardening is performed by the installer, but it is strongly recommended that you take additional hardening actions, such as adding firewall rules restricting access only from the master and worker subnet for the following services:

- NFS Server (on port 2049/tcp and 2049/udp)
- rpcbind (on port 111/tcp)
- rpc.mountd (on port 20048/tcp)

Configure proxy settings

If Internet connectivity is needed (such as for an online installation, to connect to Docker Hub), and if your organization requires it, you may have to specify a proxy server for HTTP and HTTPS connections.

On each master and worker node in the cluster, set the proxy settings by editing the `~/.bashrc` file.

```
# vi ~/.bashrc

export http_proxy=<address of proxy server>

export HTTP_PROXY=<address of proxy server>

export https_proxy=<address of proxy server>

export HTTPS_PROXY=<address of proxy server>
```

If you have the `http_proxy` or `https_proxy set`, then no_proxy and NO_PROXY must also be set and contain at least the following values:

```
export no_proxy=localhost,127.0.0.1,<all cluster node IP addresses>,<all cluster node FQDNs>,<HA virtual IP Address>,<FQDN for the HA Virtual IP address>,<NFS server IP>,<NFS server FQDN>

export NO_PROXY=localhost,127.0.0.1,<all cluster node IP addresses>,<all cluster node FQDNs>,<HA virtual IP Address>,<FQDN for the HA Virtual IP address>,<NFS server IP>,<NFS server FQDN>
```

Note: If installing a multi master cluster, the HA virtual IP address must be included in the `no_proxy` list, in both the no_proxy and NO_PROXY environment variables.

Increase the per-user process limits

Perform the following steps on every master and worker node.

1. Open (or create, if necessary) the file `/etc/security/limits.d/20-nproc.conf`.
2. Add the lines below to the file, including the leading asterisks.

   ```
   * soft nproc 10240
   * hard nproc 10240
   * soft nofile 65536
   * hard nofile 65536
   * soft core unlimited
   * hard core unlimited
   ```
3. Reboot all master and worker nodes. Nodes can be rebooted in any order.
4. Verify that all nodes are up and running by running the following command.

   ```
   ulimit -a
   ```

Configure Network Time Protocol

A network time server must be available. `chrony` is a versatile implementation of the Network Time Protocol (NTP) and keeps the system clock of each master and worker node in the cluster in sync. `chrony` is installed by default on some versions of RHEL and CentOS.

Verify your `chrony` configuration by using the command:

```
chronyc tracking
```

If `chrony` is not installed on any of the cluster nodes, install it on each of the master and worker nodes with the following procedure.

1. `yum install chrony`
2. Update `/etc/chrony.conf` with your time server information.
3. Start `chronyd` and enable the `chrony` daemon:

   ```
   systemctl start chronyd
   systemctl enable chronyd
   ```
4. Verify that chrony is operating correctly:

   ```
   chronyc tracking
   ```

Activate your Docker Hub account (online deployment only)

Docker Hub is cloud-based registry service which can store manually pushed images, including ArcSight Event Broker product images.

To complete an online installation, you must use or create a valid company Docker ID (hub.docker.com) to grant you instant and secure cloud access to Event Broker software. If you do not have an existing Docker account, please follow the steps below to register a Docker ID.

> **Note:** If you plan to perform an offline deployment, you will download images from the ArcSight software entitlement site, and this step is not necessary.

1. Go to https://hub.docker.com
2. Create a Docker ID, enter your company email address, and create a password.
3. Click **Sign Up.**
4. Click the **Confirm Your Email** link in the email you received from Docker to confirm your Docker ID account.
5. Go to https://hub.docker.com to verify that you can log into Docker Hub.
6. After login, click your Docker ID on the top right of the page. Click **Settings** and take a screenshot to include your Docker ID and the linked email address.

Please email your corresponding regional contact below with your registered Docker ID as well as the screenshot for us to enable your Docker Account. Based on your region, contact your licensing team to enable your Docker ID as follows:

- For the Americas region, contact dockersupport.ams@microfocus.com
- For the APJ region, contact dockersupport.apj@microfocus.com
- For the EMEA region, contact dockersupport.emea@microfocus.com

Chapter 3: Install the ArcSight Installer and Deploy Event Broker

Once your planning is complete and prerequisites are met, you are ready to install the ArcSight Installer and then deploy Event Broker. The following topics are discussed in this section.

Install the ArcSight Installer

The ArcSight Installer enables the deployment of ArcSight products, such as Event Broker.

Note for RHEL 7.5 or CentOS 7.5 Clusters (Only): If your cluster is running RHEL or CentOS version 7.5, then prior to beginning this process, and on *each master and worker node*, edit the file `/etc/system-release` and change 7.5 to 7.4. After the installation is complete, you can revert this file on each node to its proper value.

1. Log in to the one of the local master nodes as the root user. In this procedure, this system will be referred to as the *initial master node*.
2. If you are not logged in as the roor user, `sudo` to root.
3. Create the `/opt/downloads` directory, and change to that directory.
4. Download the installation file from the ArcSight software entitlements site. Then, verify the digital signature of the downloaded file.

   Verifying the Download: Micro Focus provides a digital public key to enable you to verify that signed software you download from the software entitlement site is from Micro Focus and has not been manipulated in any way by a third party. For information and instructions, contact Micro Focus technical support.

5. Unzip the installation file in the `/opt/downloads` directory.
6. Change to the newly-created `arcsight-installer-1.50.9` directory.

> **Note:** If installing on an XFS file system, refer to Installing on an XFS file system before proceeding with step 7.

7. Run the installation pre-checks to verify the cluster setup is correct using one of the commands given here. Depending on the cluster topology, the actual command you execute may be different from the ones shown. Note that a different installation directory than the default (`/opt/arcsight`) can be specified with the option `--INSTALL_DIR=/opt/my/install/dir>`.

   For a *multi-master installation with an external NFS server*, install the initial master node with this command:

   ```
   ./arcsight-installer-master.sh --MASTER_NODES="<initial_master_ipv4><one_
   space><master2_ipv4><one_space><master3_ipv4>" --HA_VIRTUAL_IP=<virtual_
   ip> --EXTERNAL_ACCESS_HOST=<fqdn_of_virtual_ip> --NFS_SERVER=<nfs_ip_addr>
   --NFS_FOLDER_ROOT=<root_nfs_folder> --PRECHECK
   ```

   > **Note:** The order of the master nodes passed to the `MASTER_NODES` argument must reflect the order in which the masters will be installed; that is, the first master in that list must be the first master installed and so on.

   For a *single-master cluster with an internal NFS configuration*, run the following command.

   ```
   ./arcsight-installer-master.sh --NFS_SERVER=internal --PRECHECK
   ```

   For a *single-master cluster with an external NFS configuration*, run the following command.

   ```
   ./arcsight-installer-master.sh --NFS_SERVER=<nfs_ip_addr> --NFS_FOLDER_
   ROOT=<root_nfs_folder> --PRECHECK
   ```

   > **Note:** Values for each of these arguments are discussed in the Deployment Planning chapter. For a complete list of installer arguments, see Installer Command Line Arguments.

8. Once all the pre-checks are satisfied, start the installation. Use the same command as in Step 7, but omit the `--PRECHECK` option.

   > **Note:** During the installation of the initial master node, you will be prompted for an admin password. Your password may be any string, but may not start with the @-character.

9. After the installation on the initial master node completes, from the initial master node, install any additional master and worker nodes using the following example command. (You will need to execute this script for each master and worker host in the cluster.)

   > **Note:** This script performs the following operations:
   > If the <IPv4_node_address> value is listed in the `--MASTER_NODES` argument passed to the `arcsight-installer-master.sh` script, the script installs a master node.
   > If the `<IPv4_node_address>` value is not in that list, the script installs a worker node.

   ```
   source /etc/profile
   cd ${K8S_HOME}/scripts
   ./arcsight-installer-add-node.sh <IPv4_master_or_worker_node_address>
   ```

Note: If `arcsight-installer-add-node.sh` fails during installation of a new node, re-run the `arcsight-installer-add-node.sh` script for that same node a second time. If it fails again, the failed node must be removed, before the script can be re-run again.

Now run the command `kubectl get nodes`. If, after 5-10 minutes, the command shows the failed node in any other state than Ready, remove it using the command `kubectl delete node <Failed_Node_IP>`. You can then rerun the `arcsight-installer-add-node.sh` script.

Log in to the ArcSight Installer Web Application

1. Browse to the ArcSight Installer:
    - Multi-master cluster: `https://<Virtual IP or FQDN>:5443`
    - Single-master cluster: `https://<Master Node IP or FQDN>:5443`
2. Log in as the user admin, using the password that you created during the installation.
3. In ArcSight Installer, on the **Node Management** page, make sure that all nodes are listed and have the status Ready.

Installation on an XFS File System

The ArcSight Installer can be installed on an XFS system. Note that all pre-requisites for installation remain the same as described for the default file system (which is ext4).

For instructions on configuring XFS, contact ArcSight Support.

After configuring XFS, you can now proceed with the installation process as described here.

Load Images to the Local Docker Registry (Online Method)

Follow these steps downloading the product images from Docker.com, and then upload them to the local docker registry. Ensure you have activated your Docker account before proceeding.

1. Connect to the initial master node.
2. Run the following command to download images locally:
    ```
    cd ${K8S_HOME}/scripts
    ./downloadimages.sh --suite eventbroker --registry docker --org
    arcsightsecurity
    ```
3. Enter your Docker credentials. (See "Activate your Docker Hub account (online deployment only)" on page 20 for details on how to create your Docker Hub account, if you have not already done so.)
4. Enter the product version that you want to download.
5. Once the images have been downloaded, run the following command to upload them to the local

Docker registry. This step can take up to approximately 5 minutes to complete.

```
./uploadimages.sh --suite eventbroker
```

Load Images to the Local Docker registry (Offline method)

Follow these steps to download the product images from the ArcSight software entitlement site, and then upload them to the local docker registry.

> **Verifying the Download:** Micro Focus provides a digital public key to enable you to verify that signed software you download from the software entitlement site is from Micro Focus and has not been manipulated in any way by a third party. For information and instructions, contact Micro Focus technical support.

1. Log on to the initial master node.
2. Change to the `/opt/downloads` directory.
3. Download the Event Broker tar file `arcsight-eventbroker-2.21.9.tar` from the Micro Focus software entitlements site.
4. Verify the download (as described above) and unpack the tar file.

   ```
   tar -xvf arcsight-eventbroker-2.21.9.tar
   ```

   The `/opt/downloads/eventbroker` sub-folder will contain the product images. Verify that the folder contains a set of files.
5. Upload the images to the local registry as follows:

   ```
   cd ${K8S_HOME}/scripts
   ./uploadimages.sh --suite eventbroker --dir /opt/downloads/eventbroker
   ```

Perform pre-deployment configuration

The following steps should be performed before you deploy Event Broker containers.

Editing `arcsight-installer.properties`

You may need to change the default configuration for certain product capabilities before deploying Event Broker containers (such as FIPS, Client Authentication, and the data retention policy). To change the default configuration:

1. Log in to the initial master node.
2. Change directory to `${K8S_HOME}/../installer/`
3. Open the file `arcsight-installer.properties` in a text editor, and edit the values as needed. To change optional property values, remove the comment operator (#) and then make the desired change.

4. Save the file.
5. **Important:** Run the `update-arcsight-installer-properties.sh` script, located in the same directory, for your changes to take effect.

See "The arcsight-installer.properties file " on page 39 for a detailed list of configurable properties.

Label nodes in the cluster

You will need to label each worker node to identify which node the ZooKeeper and Kafka pods will run on when deployed in the cluster. The following steps must be performed from one of the master nodes.

The number of the Kafka and ZooKeeper nodes that are labeled must be greater than or equal to the values defined for `eb-kafka-count` and `eb-zookeeper-count` properties in the file `${K8S_HOME}/../installer/arcsight-installer.properties` . The default value for both properties is 3.

- The number of ZooKeeper nodes must be odd and at least 3 to support high availability and failover.
- The number of Kafka nodes can be even or odd, and must be at least 3. The number of Kafka nodes in the cluster will depend on the cluster design and how it optimizes data throughput, fault-tolerance, and other factors.

1. To label a worker node for Kafka, run:
   ```
   kubectl label --overwrite node {worker_node_ip} kafka=yes
   ```
   Note: You can also label the node from ArcSight Installer web application UI.
2. To label a worker node for ZooKeeper, run:
   ```
   kubectl label --overwrite node {worker_node_ip} zk=yes
   ```
   If the command returns a refused or timed-out connection, temporarily remove your proxy settings using `unset http_proxy` before repeating the `kubectl` command.
3. Check that nodes are labeled correctly with the following command:
   ```
   kubectl get nodes -L=kafka,zk
   ```
4. Launch the ArcSight Installer web application, then navigate to the **Node Management** page. Make sure that all nodes are listed, have the correct labels, and have the status Ready.

Deploy Event Broker

Once you have uploaded images to the local Docker registry, you are ready to deploy ArcSight Event Broker.

1. In the ArcSight Installer web application, browse to the **Deployment** page. The list of products should be displayed with status **-**
2. In the ArcSight Event Broker row, click **DEPLOY**.
3. In the version dialog, select 2.21 and then click **DEPLOY**. A popup message will display indicating that deployment has been started.

4. Once the product deployment is finished, the status will be changed to Deployed. Please give the process some time to complete. (This can take a few minutes.)
5. To check the pod status, enter:
   ```
   kubectl get pods --all-namespaces
   ```

> **Note:** Each of the pods comprising the Event Broker will proceed through the following stages: Init:0/x, PodInitializing, and ContainerCreating. In some cases, you might also see the following transitory states: Error and CrashLoopBackOff. This is expected behavior.

Once all pods have status Running, the product is ready to configure and use.

The **Undeploy** button will remove a product and all its containers from Kubernetes. This process will take a similar amount of time as deploying images, from 2-5 minutes.

Perform Post-Deployment Configuration

Configuration

To configure ArcSight Event Broker, go to **Configuration > ArcSight Event Broker.** Change the required configuration parameters (including adding an ArcMC, managing routing, and transforming stream processors, as needed) and click **Save**.

After changing a product setting, one or more pods will be restarted in the cluster. Depending on which pods need to restart, there may be a brief interruption to some running applications.

> **Note for RHEL 7.5 or CentOS 7.5 Platforms (Only):** If your cluster is running RHEL or CentOS version 7.5, after the installation is complete, on *each master and worker node*, edit the file `/etc/system-release` and change 7.4 back to 7.5.

Configuring Event Broker for management by ArcMC

ArcSight Management Center (ArcMC) is ArcSight's central management console for management of ArcSight products. Your Event Broker can be easily managed from ArcMC as well.

Connectivity between Event Broker and ArcMC is configured in ArcMC when you add Event Broker as a managed host. For details on adding your Event Broker to ArcMC as a managed host, see "Adding a host" in the *ArcSight Management Center Administrator's Guide*.

Install your license before the evaluation period ends

Event Broker ships with a 90-day instant-on evaluation license, which will enable functionality for 90 days after installation. In order for Event Broker to continue working past the initial evaluation period, you will need to apply the ADP ArcMC license to Event Broker.

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

For details on how to apply a new license file to Event Broker, see the Licensing chapter of the *Event Broker Administrator's Guide*.

Chapter 4: Uninstall Event Broker and ArcSight Installer

To remove a single node from the cluster

1. Open an ssh connection to the node you want to remove.
2. Run the following script: `${K8S_HOME}/uninstall.sh`
3. Enter *Y* and *N* at the prompts.
4. Reboot each node after the script has finished executing.

To uninstall Event Broker and ArcSight Installer completely

When uninstalling Event Broker and ArcSight Installer from a cluster, the uninstall script should be run on worker nodes, first, before it is run on master nodes. This procedure will remove application components, but will not delete data stored on cluster nodes.

1. Stop all Collectors and Connectors from sending events to Event Broker and all consumers from reading events as well.
2. Connect to each node in the cluster, and then run the following command:
   ```
   ${K8S_HOME}/uninstall.sh
   ```
3. Enter *Y* and *N* for the prompts.
4. Reboot each node after the script has finished executing.

To clean up existing data stored on cluster nodes (optional)

After uninstalling Event Broker and ArcSight Installer, product data (including the event data stored in Kafka topics) is still kept in multiple folders on each system. You can delete this product data if it is not planned for retention or future recovery.

1. Get the path to the installation directory as follows
   ```
   cd ${K8S_HOME}/..
   pwd
   ```
2. One at a time, connect to each node in the cluster, all masters and workers, and then run the following command on each one:
   ```
   rm -rf <path to install directory> /opt/kubernetes
   ```
3. If you set up an external NFS server, manually delete the shared folders on the external NFS server.

Chapter 5: Upgrade to Event Broker 2.21

This procedure will upgrade the ArcSight Installer from version 1.40 to 1.50, and then upgrade Event Broker to version 2.21 using the Upgrade step in the upgraded ArcSight Installer.

> **Note:** Upgrading from a single master to a multi-master installation, or changing an internal NFS to an external NFS, are **not** supported by this process.

Download and Verify Files

From the Micro Focus software entitlements site, download the following files to the initial master node's `/opt` directory, and verify the digital signature of each one:

- `arcsight-installer-1.50.9.zip`
- `arcsight-installer-upgrade-1.40_to_1.50-7.zip`

> **Verifying the Download:** Micro Focus provides a digital public key to enable you to verify that signed software you download from the software entitlement site is from Micro Focus and has not been manipulated in any way by a third party. For information and instructions, contact Micro Focus technical support.

After verification, unzip the file (and only the file) `arcsight-installer-upgrade-1.40_to_1.50-7.zip`.
Do not unzip `arcsight-installer-1.50.9.zip.`

Upgrade the Operating System to RHEL or CentOS 7.5 for Each Node

ArcSight Installer 1.50 requires RHEL or CentOS 7.5 to avoid a known kernel panic issue.

To upgrade the OS, execute the following steps on each node in the cluster, **one at a time, beginning with each of the master nodes and then each of the worker nodes.**

1. SSH to *one* selected node and stop it gracefully.
```
kubectl drain <node-ip> --force --delete-local-data --ignore-daemonsets
kube-stop.sh
```
2. Check the node status to make sure only the selected node has been stopped:
```
# kubectl get nodes
```

```
NAME            STATUS                          AGE       VERSION

192.168.138.25   Ready                           7h         v1.6.1
```

```
192.168.138.26   Ready                          5h        v1.6.1

192.168.138.27   Ready                          6h        v1.6.1

192.168.138.41   Ready                          6h        v1.6.1

192.168.138.42   NotReady,SchedulingDisabled   6h        v1.6.1

192.168.138.43   Ready                          6h        v1.6.1
```

3. Configure proxies if necessary to enable access to `yum` repositories.
4. Run the update operation:
   ```
   yum update -y
   ```
5. Reboot the node.
   ```
   sync; sync
   reboot
   ```
6. After rebooting, allow Kubernetes to deploy pods to the cluster node. Run:
   ```
   kubectl uncordon <node-ip>
   ```
7. After the OS has been upgraded on the node, run the following command from the master node to verify that all nodes are in Ready status:
   ```
   # kubetctl get nodes
   ```

```
NAME             STATUS    AGE       VERSION

192.168.135.174   Ready     1d        v1.6.1

192.168.135.175   Ready     1d        v1.6.1

192.168.137.177   Ready     1d        v1.6.1

192.168.137.178   Ready     1d        v1.6.1

192.168.137.179   Ready     1d        v1.6.1

192.168.137.180   Ready     1d        v1.6.1
```

8. Modify the `/etc/system-release` file to change 7.5 to 7.4.
   ```
   # vi /etc/system-release
   ```

For example, on CentOS:

```
# cat /etc/system-release
CentOS Linux release 7.4.1708 (Core)
```

9. Verify that all pods are in the running state by running:
   ```
   watch "kubectl get pods --all-namespaces -o wide"
   ```

Note: After upgrading the operating system on the node, the process can take 5-15 minutes for all nodes to return to the Running status. Give the process time to complete.

10. Once all pods on the node are running, repeat Steps 1-9 on the *next* node. (Remember to perform all of these operations *one node at a time*.)

Troubleshooting the OS Upgrade

For the master nodes:

- If, after upgrading the OS, rebooting and uncordoning a node, you find that one of the other master nodes is in NotReady state and not changing, you will need to run the following command on the node which is NotReady:
  `systemctl restart kubelet.`
- Verify that node comes fully back to the Ready state by running:
  `kubectl get nodes`
- In some cases, after updating the OS and uncordoning a node, an `eb-kafka` pod may not be running correctly, and this will cause the schema registry to go into a crashLoopBackoff mode. To fix this, run the following command to restart the Kafka pod:
  `Kubectl delete pod eb-kafka-{x}`
  where x is 0, 1, 2 (or greater if 3 or more Kafka nodes are present).

Next Steps

1. On each master and worker node in the cluster, run the following two commands to install the required packages:
   ```
   yum install conntrack-tools
   yum install httpd-tools
   ```
2. If using an external NFS server, create, then export a new share (`itom/db`). The following setup pocedure assumes:
   - 3 master nodes with IP addresses: 10.1.2.11 - 13
   - 3 worker nodes with IP addresses: 10.1.2.21 - 23
   - An NFS share root directory of `/opt/arcsight/nfs/volumes`.

   Then run the following commands on the external NFS server:

   ```
   mkdir -p /opt/arcsight/nfs/volumes/itom/db
   chown -R 1999:1999 /opt/arcsight/nfs/volumes/itom/db
   ```

   In the `/etc/exports` file, add the following line:

   ```
   /opt/arcsight/nfs/volumes/itom/db 10.1.2.11
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.12
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.13
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.21
   (rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.22
   ```

```
(rw,sync,anonuid=1999,anongid=1999,all_squash) 10.1.2.23
(rw,sync,anonuid=1999,anongid=1999,all_squash)
```

3. `exportfs -ra`
4. On each master and worker node in the cluster, create a backup folder on a disk partition with 30 GB or more of free disk space. This folder will be referred to as `<backup_folder>`. For example:
   `mkdir /opt/BKUP`

Run Platform Upgrade Pre-Checks

1. Change to the new installer-upgrade directory (named for the upgrade file you unzipped previously.) For example,:
   `cd /opt/arcsight-installer-upgrade-1.40_to_1.50-7`.
2. Run the platform upgrade pre-checks:
   ```
   ./upgrade.sh -i {path to installer}/arcsight-installer-1.50.9-master.zip
   -d <backup_folder> --precheck
   ```

Upgrade the ArcSight Installer from v1.40 to v1.50

Once the operating system has been upgraded and all pre-checks complete successfully, proceed with the actual platform upgrade. This will upgrade the infrastructure components (such as Docker and Kubernetes) on all the cluster nodes, but not the applications deployed on top of the platform (that is, Event Broker and Investigate).

> **Note:** The platform upgrade may take upwards of 3 hours on a 6 node cluster (3 dedicated masters and 3 dedicated workers). During the duration of the platform upgrade, there should be minimal disruption to the event flow.

1. Invoke the platform upgrade:
   ```
   ./upgrade.sh -i {path to installer}/arcsight-installer-1.50.9-master.zip
   -d <backup_folder>
   ```
2. After the platform upgrade completes successfully, manually copy the following two files on each master node:
   ```
   cp /<backup_folder>/arcsight-installer-1.50.9/arcsight-
   installer.properties ${K8S_HOME}/../installer/
   cp /<backup_folder>/arcsight-installer-1.50.9/update-arcsight-installer-
   properties.sh ${K8S_HOME}/../installer/
   ```
3. Edit `${K8S_HOME}/../installer/arcsight-installer.properties`, and uncomment and modify the following lines as necessary:
   ```
   ## Number of Schema Registry nodes
   # eb-schema-registry-count=3
   ```

```
## Minimum number of Kafka nodes required to run Schema Registry
# eb-schema-registry-min-kafka-count=3
```

4. On each master and worker node in the cluster, edit `/etc/system-release`. Change the OS version back to 7.5.

Upgrade Event Broker

During the upgrade process, the Event Broker will be down and non-operational. Therefore, plan accordingly for the downtime before performing the upgrade.

1. Depending on whether you are performing an online or offline image upload, select one of the following sections for the steps to download product images and upload them to the local Docker registry.
    - "Load Images to the Local Docker registry (Offline method)" on page 25
    - "Load Images to the Local Docker Registry (Online Method)" on page 24
2. Launch the ArcSight Installer web application, and click **Upgrade** for Event Broker. Event Broker is upgraded to version 2.21.

Upgrade Troubleshooting

The following troubleshooting scenarios and resolutions may be helpful when performing the upgrade.

Scenario # 1: Upgrade completes, but reports that one or more master or worker nodes was not upgraded

First, check for failed master nodes. Display the content of the file `/opt/arcsight-installer-upgrade-1.40_to_1.50-7/failedMasters` to get a list of failed master nodes. For each failed master node:

1. Uninstall the node (Note that the node may already be uninstalled).
2. Make sure the node is running CentOS/RHEL 7.5. If not, update the operating system.
3. Reboot the failed node.
4. Add the failed node back to the cluster by running the following command from the master node on which you invoked the upgrade script:
    ```
    $K8S_HOME/scripts/arcsight-installer-add-node.sh <failed master node
    IP address>
    ```

Next, check for failed worker nodes. Display the content of the file `/opt/arcsight-installer-upgrade-1.40_to_1.50-7/failedWorkers` to get a list of failed worker nodes. For each failed worker node:

1. Uninstall the failed node (note: the node may already be uninstalled):
2. Make sure the node is running CentOS/RHEL 7.5. If not, update the operating system.

3. Reboot the failed node.
4. Add the failed node back into the cluster by running the following command from the master node on which you invoked the upgrade script:
   ```
   $K8S_HOME/scripts/arcsight-installer-add-node.sh <failed worker node IP
   address>
   ```
5. Add back the original labels to the newly added worker node (your labels may vary in practice):
   ```
   kubectl label node <failed worker node IP address> <zk=yes> <kafka=yes>
   ```

Scenario #2: Upgrade fails in step #5 with error "Failed to generate parameter file"

This failure is typically transient. Wait a few minutes, then retry the upgrade.

Other Potential Issues

In other upgrade failure scenarios, check the upgrade logs:

- On master node from which `upgrade.sh` was initiated:
  - Log(s) for the upgrade of the platform across the cluster (12 steps)
    - Located here: `/opt/arcsight-installer-upgrade-1.40_to_1.50-7/upgrade-YYYYMMddhhmmss.log`
- One log file on each master node for phase 1 of the platform upgrade:
  - Located here: `<backup_folder>/upgrade-YYYYMMddhhmmss.log`
- One log file on each master/worker node for the phase 2 of the platform upgrade:
  - Located here: `/tmp/install-YYYYMMddhhmmss.log`
    - Except on master node from which `upgrade.sh` was initiated where it is:
      - Under `<backup_folder>/upgrade-YYYYMMddhhmmss.log`

If the cause of the upgrade failure can be determined from errors in the log files, address the cause first. Wait a few minutes, then re-try the upgrade. If the upgrade fails again, and the cause of the failure can't be determined, contact ArcSight Support.

Appendix A: Troubleshooting

This section includes material to help you troubleshoot problems or issues that may occur during the installation. Consult the Event Broker Administrator's Guide for additional detailed troubleshooting information.

Event Broker deployment fails with "Can not config persistent volume" error.

This is usually due to the NFS server not running. EB deployment requires that either NFS v3 or v4 is running properly before EB deployment. A quick test for NFS service is to do the following:

1. Log in to the `suite-installer-xxx` pod.
2. Run

   ```
   mount -t nfs /opt/arcsight/volumes/eventbroker /tmp
   ```

The command should be successful.

Why do I see `Failed to upload .. suite features ...` failures when running `uploadimages.sh` during the installation?

The following message is displayed:

```
 "The suite-installer container is not running. Please make sure your suite-
installer pod status is "RUNNING". Failed to upload the data of suite
features."
```

Check that you are running the `uploadimages.sh` script from the correct folder, which is the following folder::`${K8S_HOME}/scripts/`.

Appendix B: Installer command line arguments

The installer command line (`./arcsight-installer-master.sh`) can take the following arguments, described in alphabetical order.

Argument	Description	Use with Cluster Type
`--DOCKER_HTTP_PROXY`	Proxy settings for Docker. Specify if accessing the Docker hub or Docker registry requires a proxy. By default will be configured from http_proxy environment variable on your system.	Multi-master and Single master
`--DOCKER_HTTPS_PROXY`	Proxy settings for Docker. Specify if accessing the Docker hub or Docker registry requires a proxy. By default will be configured from https_proxy environment variable on your system.	Multi-master and Single master
`--DOCKER_NO_PROXY`	Proxy settings for Docker. Specify if accessing the Docker hub or Docker registry requires a proxy. By default will be configured from no_proxy environment variable on your system.	Multi-master and Single master
`--EXTERNAL_ACCESS_HOST`	Defines a fully-qualified domain name of the Virtual IP address This is required for multi-master cluster deployments.	Multi-master
`-h|--help`	Show help.	N/A
`--HA_VIRTUAL_IP`	A Virtual IP (VIP) is an IP address that is shared by all master machines. The VIP is used for the connection redundancy by providing fail-over for one machine. When a master goes down, the other master takes over the VIP address and responds to requests sent to the VIP. Mandatory for multi-master cluster.	Multi-master
`--INSTALL_DIR`	Path to a directory where Installer is to be installed under /opt. If not specified the default path will be used - /opt/arcsight	Multi-master and Single master
`--LOG_MAX_FILE`	The max number of files used by the by Docker log rotation, default if not provided will be 5 files.	Multi-master and Single master
`--LOG_MAX_SIZE`	The max file size used by Docker log rotation, default if not provided will be 20 MB.	Multi-master and Single master

Argument	Description	Use with Cluster Type
`--MASTER_NODES`	The value is a list of the three master node IPv4 addresses, separated by a blank space. The entire list is enclosed in one set of double-quotes. Three master nodes are recommended for production environments. Npte that the order of the master nodes in the list must reflect the order in which the masters will be installed. For example, given a list of (10.0.1.13, 10.0.1.12, 10.0.1.11), then 10.0.1.13 must be the first master installed.	Multi-master
`--NFS_FOLDER_ROOT`	The root folder on the external NFS server, for example: /opt/arcsight/nfs/volumes The default 'internal' NFS location is '/opt/arcsight/volumes'	Multi-master and Single master
`--NFS_SERVER`	Persisted platform and product data will be stored in this location. The value is either the external NFS server IP, external NFS server hostname, or the text 'internal'. If the value is 'internal', a simple NFS server on master node will be installed. The internal option is supported only with single master cluster deployments. Multi-master deployments must use an external NFS server.	Multi-master and Single master
`--POD_CIDR`	Kubernetes pod IP range. Default is 172.16.0.0/16.	Multi-master and Single master
`--REGISTRY_ORGNAME`	The organization name in the local Docker registry where application images are located. The default value is 'arcsightsecurity'.	N/A This option does not need to be specified when running the script.
`--ROOTCA`	Specify the root CA certificate for generating server and client certificates.	Multi-master and Single master
`--ROOTCAKEY`	Specify the root CA key for generating server and client certificates.	Multi-master and Single master
`--SERVICE_CIDR`	Kubernetes service IP range. Default is 172.30.78.0/24. Must not overlap POD_CIDR range.	Multi-master and Single master

Appendix C: The arcsight-installer.properties file

The `arcsight-installer.properties` file controls several important settings for your Event Broker installation. You will need to adjust the default configuration setting if you are deploying in FIPS mode, or adding more worker nodes to the default configuration, etc. The settings are described here.

Refer to Editing arcsight-installer.properties for instructions on editing the file.

Setting	Notes
## All Event Broker components will use FIPS-certified encryption algorithms	
eb-init-fips=false	Enables FIPS if set to true. Not recommended to change after deployment.
## Event Broker Kafka will use TLS Client Authentication to verify client connections	
eb-init-client-auth=false	Enabled TLS-CA if set to true. Not recommended to change after deployment.
## Number of partitions for Event Broker default topics in Kafka	
eb-init-noOfTopicPartitions=6	Default value. Will only affect newly created topics. (Add new partitions to existing topics with the Event Broker Manager.)
## Replication factor for Event Broker default topics in Kafka	
eb-init-topicReplicationFactor=2	Default value. Will only affect newly created topics. (Must delete old topics to change replication factor.)
## Kafka log retention size	
eb-init-kafkaRetentionBytes=10737418240	Default value per partition per topic. Deletion will occur when Event Broker hits either the duration or retention bytes, whichever comes first.
## Kafka log retention size for the Vertica Avro topic. This is uncompressed and requires more space to hold events for the same duration.	

Setting	**Notes**
eb-init-kafkaRetentionBytesForVertica=10737418240	Default value per partition per topic. May require additional space than other topics because data is uncompressed. To ensure data retention is the same as other topics, this topic may need to be significantly larger than other topics, as large as a factor of 7 or more. Deletion will occur when Event Broker hits either the duration or retention bytes, whichever comes first.
## Kafka log retention duration	
eb-init-kafkaRetentionHours=672	Based on environment. Requires calculation on customer behalf. Applies to all topics, including those created through ArcMC. Deletion will occur when Event Broker hits either the duration or retention bytes, whichever comes first.
##The replication factor for the offsets topic	
eb-init-kafkaOffsetsTopicReplicationFactor=3	Defines the replication factor for the __consumer_offsets topic.
##The max time that Kafka waits to establish a connection to ZooKeeper	
eb-init-kafkaZKConnectionTimeoutMs=6000	Set the number of milliseconds that Kafka waits while attempting to establish a connection to ZooKeeper.
##ZooKeeper session timeout configuration for the Kafka broker	
eb-init-kafkaZKSessionTimeoutMs=6000	Set the number of milliseconds that ZooKeeper waits to receive a heartbeat from a broker or any consumer. If a heartbeat is not received in this period of time, the broker/consumer is assumed to be dead. A rebalance operation will be performed.
## Kafka inter-broker protocol version	
inter-broker-protocol-version=0.11.0.0	Only to be used during upgrades.
## The message format version the broker will use to append messages to the logs.	
log-message-format-version=0.11.0.0	Only to be used during upgrades.

Setting	Notes
## Number of Kafka and ZooKeeper nodes	
eb-kafka-count=3	Determines cluster size for Kafka. Must match number of worker nodes labeled as kafka=yes in Kubernetes. 1 node to 1 host.
eb-zookeeper-count=3	Determines ZooKeeper cluster size. Max of 7. Must match number of worker nodes labeled as zk=yes in Kubernetes. MUST be an odd number.
## Host path to store data persistently	
eb-kafka-path=/opt/arcsight/k8s-hostpath-volume/eb/kafka	Used if you have configured 'internal' NFS server. It will be created if it does not exist.
eb-zookeeper-path=/opt/arcsight/k8s-hostpath-volume/eb/zookeeper	Used if you have configured 'internal' NFS server. It will be created if it does not exist.
## ArcMC hostname	
eb-arcmc-hosts=localhost:443	
## The endpoint identification algorithm to validate the server hostname using the server certificate.	
ssl-endpoint-identification-algorithm=https	Hostname verification for Kafka to Kafka connections.
## The number of stream threads	
stream-num-threads=6	Do not change unless performance issue.
## truncate fields in C2av	
c2av-field-truncate=false	Used by the CEF to Avro transformation stream processor used in the Investigate data pipeline. Defines how to handle events when the value for an attribute is longer than the maximum size defined by the Investigate Vertica schema. If set to false, the event will be rejected. It is not loaded to the Investigate events table, in Vertica. Instead, an entry is loaded to the rejected_events table. If set to true, the value will be truncated so that it fits the field limit. The event will be loaded to the Investigate events table.

Setting	Notes
## c2av config params	Configuration properties for the CEF to Avro transformation stream processor used in the Investigate data pipeline.
c2av-heartbeat-interval-ms=1000	
c2av-max-poll-interval-ms=3600000	
c2av-max-poll-records=100	
c2av-session-timeout-ms=180000	
c2av-request-timeout-ms=305000	
## Log level for Event Broker components	Used to change the logging levels of each Event Broker component.
kafka-log-level=info	
zookeeper-log-level=info	
schema-log-level=info	
webservice-log-level=info	
c2av-stream-processor-log-level=info	
routing-processor-log-level=info	
## Host path directory for ArcMC certificates	Deprecated.
arcmc-certs-path=/opt/arcsight/k8s-hostpath-volume/eb/arcmccerts	In Event Broker 2.2x, this is now configured using ArcSight Installer, under the Event Broker Configuration link > ArcMC Monitoring tab.
## Host path directory for AutoPass license data file persistence	
eb-autopass-path=/opt/arcsight/k8s-hostpath-volume/eb/autopass/	Full path on the initial master, where the permanent license is stored. See the section 'Install your license before the evaluation period ends' or more information.
## Number of Schema Registry nodes	
eb-schema-registry-count=3	
## Minimum number of Kafka nodes required to run Schema Registry	
eb-schema-registry-min-kafka-count=3	

Appendix D: Command Reference

The following command reference includes useful commands for Event Broker and ArcSight Installer administration.

Event Broker Core	Description
Kubernetes	
Node Management	
kubectl get nodes -o wide	Displays name, status, age, OS image, Kernel Version, Kuberentes version of each node in the cluster Extended: “kubectl get nodes -L fqdn -o wide” Show OS-Image, Kernel version, FDQN “kubectl get nodes -L fqdn -o wide”
kubectl get nodes -L Master -L fqdn	List the Master Nodes and FQDN
kubectl describe node <node IP>	Shows detailed information about a node, labels, conditions, capacity, SysInfo, and Perf Metrics
kubectl delete node <ip address>	Preferred method is to run uninstall script in EB Worker shell
kubectl describe pod -n core | grep Master=https	Get VIP (Virtual IP address assigned to every node in the cluster)
Service Management	
kubectl get services -n <namespace>	Show service ports for pod internal ports. Extended: kubectl get services -n <namespace> -o wide
kubectl describe svc --all-namespaces	Outputs IP/ports of services
Pod Management	
kubectl describe pods/<podname> -n <namespace>	Shows detailed information about a pod
kubectl describe pods -n core	Describe pods in the namespace core. Without namespace, will default to Nginx-controller. Can also grep for pods
kubectl get pods -o wide	Defaults to “default” namespace. Extended: List all namespaces “docker get pods --all-namespaces -o wide” Get other namespaces use -n <namespace name>

kubectl delete pod <pod name> -n <namespace>	Deletes a pod, then a new pod is created. Namespaces are “core”, “arcsighteventbroker1” “arcsightinvestigate1”
kubectl logs <pod name> -n <namespace>	Print logs to screen. Use -f to follow. Pipe logs to a file “> filename.txt
kubectl get pods --all-namespaces -o=jsonpath='{range .items[*]}{"\n"}{.metadata.name}{":\t"}{range .spec.containers[*]}{.image}{", "}{end}{end}' |sort;	List all pods running in their containers
kubectl exec -n core kube-dns-2604315512-tz260 -- env	Get Kube-DNS pod environment information. Note: replace pod name from kubectl get pods -n core
kubectl logs -f kube-dns-2604315512-816g4 -n core kubedns	Tail logs of a pod in the core namespaces in the container kubedns. Container names kubedns, dnsmasq, sidecar
kubectl get pv	Shows persistent volumes for EB, Investigate, and ITOM storage allocation
Kubectl get pvc --all-namespaces	Shows volumes claims and capacity
kubectl get ns	Shows a list of name spaces
kubectl get ns <namespace> -o yaml	Shows the kind, metadata, and status
kubectl get cm --all-namespaces	Get config maps (persistent) of all namespaces. Extended: -o yaml to output to a yaml file
kubectl get rs --all-namespaces	Get replica sets of all namespaces
kubectl get ds -all-namespaces -o wide	Get desired states of all namespaces. Desired states are the amount of pods that should be in ready status
kubectl get all	Output all pods, services, replica sets, etc
kubectl cluster-info dump	Dump Kubernetes cluster info. Extended: kubectl cluster-info dump > cluster_dump.txt
ArcsightRootFolder/kubernetes/bin/kube-status.sh	Check the status of nodes, pods, services, on each node in the cluster
ArcsightRootFolder/kubernetes/bin/kube-stop.sh	Stop Kuberentes, Docker, Docker-Bootstrap (unmounts Docker filesystem before OS shutdown)
ArcsightRootFolder/kubernetes/bin/kube-start.sh	Start Kuberentes, Docker, Docker-Bootstrap (If you stopped using kube-stop.sh)
ArcsightRootFolder/kubernetes/bin/kube-restart.sh	Restart Kubernetes, Docker, and Docker-Bootstrap pods

ArcsightRootFolder/kubernetes/bin/kube-redeploy.sh	Redeploy (delete and create) core namespace pods (should be used if port 5443 EB Install cannot be accessed)
systemctl list-units -t service | grep kubelet	Shows Kublet service
systemctl status kublet	Shows Kublet service status Stop/Restart should NOT be used! Use kube-restart.sh above.

Docker/ Docker-Bootstrap	Description
export TERM=xterm	Set terminal when inside a Docker container
docker images	List images loaded into the Docker repository. “docker images -a | wc –l” List all images and count the number of images
docker logs -f <container id>	Follow the logs in the Docker container
docker logs -f --details <container id> > test.txt	Get detailed logs and output to a file
docker ps | <container name>	Gets the container ID, Docker repository path, container uptime, internal NAT IP’s
ps -ef | grep dockerd	Get DockerD path
docker top <container id>	Display the running processes of a container
docker network ls	Show list of networks IDs, names, drivers
docker network inspect <network ID>	#Get network ID from “docker network ls” Shows IP address management and IPs of all Docker containers
brctl show	Show bridge name, ID, STP status, and interfaces of docker0 and virbr0
docker cp <container id>:/etc/hosts /tmp	Copy files from a Docker container to the guest OS
docker exec -it <container ID> /bin/bash	Executes Bash interactive terminal shell to Docker container
docker -H unix:///var/run/docker-bootstrap.sock ps	Displays the container ID, image names, commands, creation date, and status of Flannel, Vault, and ETCd
docker -H unix:///var/run/docker-bootstrap.sock logs <container ID>	Get the logs of the Flannel, Vault, or ETCd containers
docker -H unix:///var/run/docker-bootstrap.sock volume ls	List the volume for the Docker-bootstrap
docker -H unix:///var/run/docker-bootstrap.sock volume inspect <volume name>	Shows the volume name and mount point
docker -H unix:///var/run/docker-bootstrap.sock stats	Shows real time stats of the docker-bootstrap container for CPU, memory usage/limit, network, storage IO, PIDs
docker -H unix:///var/run/docker-bootstrap.sock top <container ID>	Show TOP processes running for the specified container ID

docker -H unix:///var/run/docker-bootstrap.sock restart <container name>	Restarts the specified container. Valid container names are “kube_flannel”, “vault_container”, “etcd_container”
docker -H unix:///var/run/docker-bootstrap.sock info	Shows information about status, storage, metadata, OS, etc
docker version	Show the Docker server and client versions
cat ArcsightRootFolder/kubernetes/cfg/docker-bootstrap	DOCKER_OPTS=" -H unix:///var/run/docker-bootstrap.sock --exec-root=/var/run/docker-bootstrap -g ArcsightRootFolder/kubernetes/data/docker-bootstrap -p /var/run/docker-bootstrap.pid "
systemctl cat docker-bootstrap.service	Show SystemD configurations for Docker-Bootstrap
systemctl cat docker.service	Show SystemD configurations for Docker
systemd-delta | grep docker	/usr/lib/systemd/system/docker.service → /usr/lib/systemd/system/docker.service.d/http_proxy.conf /usr/lib/systemd/system/docker-bootstrap.service → /usr/lib/systemd/system/docker-bootstrap.service.d/http_proxy.conf #Get Docker proxy information

Nginx Load Balancer	Description
kubectl get cm nginx-load-balancer-conf -o yaml	Get config map of Nginx load balancer and TLS protocol version
kubectl cp default/nginx-ingress-controller-4w6c6:etc/nginx/nginx.conf /tmp/nginx.conf	Get Nginx configuration file. Shows ports mapped to kube-proxy
kubectl exec nginx-ingress-controller-<pod hostname> -- cat etc/nginx/nginx.conf	Cat the output of the nginx.conf file
Get Nginx container logs	
docker ps | grep nginx	Get Nginx Docker processes and container ID
docker logs <container id> -f	Get logs from nginx pod
kubectl logs -f nginx-ingress-controller-<pod id> -n default	Get logs from nginx ingress controller
kubectl exec nginx-ingress-controller-<hostname> -n default -- curl --insecure http://127.0.0.1:18080/internal_nginx_status	Get status of each Nginx ingress controller showing active connections, requests, etc

ETCd Commands	Description
ArcsightRootFolder/kubernetes/bin/etcdctl --endpoint https://16.40.174.50:4001 --ca-file=ArcsightRootFolder/kubernetes/ssl/ca.crt --cert-file=ArcsightRootFolder/kubernetes/ssl/server.crt --key-file=ArcsightRootFolder/kubernetes/ssl/server.key ls --recursive /suite-installer	Get output of every key under /suite-installer

ArcsightRootFolder/kubernetes/bin/etcdctl --extended --endpoint https://X.X.X.X:4001 --ca-file=ArcsightRootFolder/kubernetes/ssl/ca.crt --cert-file=ArcsightRootFolder/kubernetes/ssl/server.crt --key-file=ArcsightRootFolder/kubernetes/ssl/server.key get /coreos.com/network/config	Get ETCd subnet configured from Flanneld
ArcsightRootFolder/kubernetes/bin/etcdctl --endpoint https://X.X.X.X:4001 --ca-file=ArcsightRootFolder/kubernetes/ssl/ca.crt --cert-file=ArcsightRootFolder/kubernetes/ssl/server.crt --key-file=ArcsightRootFolder/kubernetes/ssl/server.key cluster-health	Check ETCd cluster member health
cd ArcsightRootFolder/kubernetes/data/etcd/data/member/snap strings db | grep member	Check if ETCd members status
docker -H unix:///var/run/docker-bootstrap.sock ps	Get container ID of ETCd
docker -H unix:///var/run/docker-bootstrap.sock exec -i -t <container ID> /bin/sh	

Event Broker Application Kafka Command	Description
kubectl get pods -n arcsighteventbroker1 -o wide	Get pod IPs for namespace arcsighteventbroker1 and pod hostnames
kubectl exec -n arcsighteventbroker1 eb-web-service-<pod hostname> -- /opt/eb/scripts/check_status.py -c kafka -a <kafka_pod_IP>:9092	Get status of Kafka “eb-kafka-0” Example: Connection to Kafka on 172.16.20.3:9092 was successful kafka 172.16.20.3:9092 is available
kubectl exec eb-zookeeper-0 -n arcsighteventbroker1 -- kafka-topics --zookeeper localhost:2181 --create --topic eb-test --partitions 21 --replication-factor 3	Use kubectl to create a topic called “eb-test” with X partitions with a replication factor of X
kubectl exec eb-zookeeper-0 -n arcsighteventbroker1 -- kafka-topics --zookeeper localhost:2181 --list | grep eb	Use kubectl to connect to eb-zookeeper and get list of topics
kubectl exec -n arcsighteventbroker1 eb-kafka-0 -- kafka-consumer-groups --bootstrap-server localhost:9092 --list	Get list of consumer groups
kubectl exec -n arcsighteventbroker1 eb-kafka-0 -- kafka-consumer-groups --bootstrap-server localhost:9092 --describe --group <consumer group name>	Get consumer group information like partition, offset, log end, log, consumer id and host
printenv | <service name>	Identify services and ports in the Kafka container
ls -la /usr/bin |grep kafka	List all scripts used to manage Kafka. Note: Must be connected to Kafka Docker container in Bash
kubectl exec eb-zookeeper-0 -n arcsighteventbroker1 -- kafka-topics --zookeeper localhost:2181 --list	Displays all configured topics in Kafka
kubectl exec eb-kafka-0 -n arcsighteventbroker1 -- kafka-topics --describe --zookeeper X.X.X.X:32181, X.X.X.X:32181, X.X.X.X:32181 --topic eb-cef	Displays partition count, replication factor, and configs for the specified topic. X=IP’s of each EB Masters/Workers

kubectl exec eb-zookeeper-0 -n arcsighteventbroker1 -- kafka-topics --zookeeper localhost:2181 --describe | more	Use kubectl to describe all topics, partitions, and replicas. Pipe to grep for “eb-esm” for example
kubectl exec eb-zookeeper-0 -n arcsighteventbroker1 -- kafka-topics --zookeeper localhost:2181 --delete --topic eb-test	Delete a topic in Kafka. DO NOT DELETE eb-cef, eb-other, eb-esm, or internal topics!!! This should only be used for user created topics that where created using ArcMC or with Kafka manager This command will be supported in later versions of ArcMC in the UI
kafka-configs --describe --zookeeper X.X.X.X:32181,X.X.X.X:32181,X.X.X.X:32181 --entity-type topics	Get key-value pairs of configuration for all topics
kubectl exec eb-zookeeper-0 -n arcsighteventbroker1 -- kafka-configs --describe --zookeeper localhost:2181 --entity-type topics --entity-name eb-cef	Get configuration for a single topic

ZooKeeper Troubleshooting Commands	Description
kubectl exec -n arcsighteventbroker1 eb-zookeeper-0 -it bash	Connect to ZooKeeper container using Bash
echo envi | nc localhost 2181	Must be connected via Bash to ZooKeeper Gets ZooKeeper, java, and OS versions
zookeeper-shell localhost:2181	-Connect to ZooKeeper shell -Type “quit” to exit shell
ls /brokers/ids	List the broker ID’s
get /cluster	List the details of the cluster
get /schema_registry	List the details of the schema registry
get /controller	Show the broker ID
ls /brokers/topics	Show the broker topics
ls /brokers/topics/eb-cef/partitions	Show the partitions of topic eb-cef
kubectl exec -i -t eb-zookeeper-0 -n arcsighteventbroker1 -- ls -la /var/lib/zookeeper/data	Get ZooKeeper data
kubectl exec -i -t eb-zookeeper-0 -n arcsighteventbroker1 -- ls -la /shelltools/	List shelltools
kubectl exec -i -t eb-zookeeper-0 -n arcsighteventbroker1 -- cat /shelltools/kafkaEnvar.sh	Get Kafka environment variables

Kafka Scheduler Vertica DB	Description
/root/install-vertica	Default install path

./kafka_scheduler status	Prints status of running Kafka scheduler with some information of imported/rejected messages count
./kafka_scheduler create <IPaddress:port>	Creates new scheduler for given Kafka with number of partitions provided (defaults to: 6) Comma separated
./kafka_scheduler start	Starts scheduler for all Kafka instances registered Will try to stop all first
./kafka_scheduler stop	Kills all running scheduler instances
./kafka_scheduler delete	Delete scheduler metadata for all Kafka instances registered. This will shut down all scheduler instances

EB Web Service REST API

Cluster Details	
curl --noproxy 127.0.0.1 -k -u "admin:atlas" https://127.0.0.1:38080/ cluster/version	Get cluster version
curl --noproxy 127.0.0.1 -k -u "admin:atlas" https://127.0.0.1:38080/	Query REST API from EB Web Service Options are: cluster/version cluster/broker/metric cluster/broker/FQDN/metric cluster/zookeeper cluster/topic monitoring/streamprocessor monitoring/streamprocessor/metrics monitoring/streamprocessor/<brokerid>/metrics routing/rule routing/rule/<id>/ routing/route routing/route/<id>/ service/registry service/registry/<id>/ service/mapping/ service/mapping/<id>/ /routing/schema/cef /routing/language

DNS Commands	Description
Guest OS DNS Settings	
/etc/hosts	Should only contain local address for IPv4 and IPv6 entries for the loopback. Should NOT contain guest OS DNS name
/etc/resolv.conf	Should contain “nameserver X.X.X.X” for each DNS server. Can also include “search your_domain.com” as helper
/etc/nsswitch.conf	Default for DNS is to check /etc/hosts files first, then check /etc/nsswitch.conf
Kubernetes DNS Pods	
Pods in Kubernetes DNS container	
Sidecar	a daemon that exports metrics and performs healthcheck on DNS systems

Kube-dns	a process watches the Kubernetes Master for changes in Services and Endpoints, and maintains in-memory lookup structures to serve DNS requests
Dnsmasq	container adds DNS caching to improve performance
kubectl exec -n core kube-dns-XXXXXX-XXXX -- ping X.X.X.X	Ping a guest OS interface from inside the KubeDNS to the outside guest OS or another Master/Worker
kubectl exec -n core kube-dns-XXXXXX-XXXX – cat /etc/resolv.conf	Confirm KubeDNS picked up nameserver entries in the guest OS /etc/resolv.conf file
kubectl exec -n arcsighteventbroker1 <pod name> -- ping X.X.X.X	Ping an IP from an Event Broker pod to outside guest OS. All Event Broker pods talk to the KubeDNS and their /etc/hosts file point to the KubeDNS 172.30.78.78 IP defined in the install script.
kubectl exec -n core <pod name> -- /bin/cat /etc/hosts	Check /etc/hosts file on a pod to confirm KubeDNS IP is set
kubectl exec -n core kube-dns-XXXXXX-XXXX -- nslookup X.X.X.X	Perform NSLOOKUP on Kube-DNS pod to resolve a remote EB IP to <fddn> or <fqdn to IP>
kubectl exec -n core kube-dns-<pod hostname> -c dnsmasq -- cat /etc/resolv.conf	Get output of /etc/resolv.conf from pod Kube-DNS in container DNSMASQ to confirm Kube-DNS has correct DNS nameserver configuration
kubectl exec -n core kube-dns-<pod hostname> -c dnsmasq – cat /etc/hosts	Get pod hostname or “kubectl get pods -n core | grep kube-dns”
for i in X.X.X.X; do ssh $i 'hostname -f;cat /etc/resolv.conf';done	Get full hostname, and output of /etc/resolv.conf Replace X.X.X.X with Master and Worker IPs
Troubleshooting DNS	DNS Failure (possible schema-registry pod crashing)
kubectl get pods -n arcsighteventbroker1 | grep schemaregistry kubectl logs eb-schemaregistry-<hostname> -n arcsighteventbroker1 | grep "bootstrap.servers ="	Step 1: Find the schema registry pod name by running the command Step 3: Find the configured bootstrap server Example output: bootstrap.servers = [yourdomain.com:9092] If the FQDN is not showing, correct DNS forward and reverse lookup

System Guest OS / Node / Server	Description
Pre-deployment Setup	
sestatus	Check stats of SElinux
ssh-keygen -t rsa	Generate public RSA key on initial Master
ssh-copy-id -i ~/.ssh/id_rsa.pub root@< node ip address>	Copy initial Master Node SH public key to all Masters and Workers
Internal/External NFS File Server Troubleshooting	

systemctl status rpcbind systemctl status nfs-server systemctl status nfs-lock systemctl status nfs-idmap	
showmount -e <nfs_server_ip>	Show mounted file systems from remote NFS server
cat /etc/exports	Show mounted file systems on NFS server
mount | grep nfs	Filter NFS directories of the OS file systems
“journalctl -f | grep mount” “journalctl -f | grep nfs”	Watch the logs for Docker containers being mounted
Networking	
route -v	Shows the Kernel IP routing table networks, interfaces, and gateways of the host and all other EB worker nodes
netstat	Show network protocol ports bound to processes
watch netstat -s	Watch netstat interface counters
netstat -anlp | grep <port number>	Map a port to a process running in a container Example: netstat -anlp | grep 5443
netstat -anlp |grep <service> | grep LISTEN	Identify which port a service is running on
lsof -i :<portnumber>	Map port number to the file system path
pstree -p <PID> -sa	Shows command, PID, user, node name
Proxy	
/etc/proxy.d/proxy.sh	Proxy.sh created before install of Event Broker; contains exports for PROXY and NO_PROXY
env | grep proxy	Confirm proxy settings were extracted
wget <some_domain.com>	Test proxy by connecting to an external website to confirm proxy settings are correct

Glossary

A

ArcSight Root Installation Folder

The ArcSight Root Installation Folder is the root folder that the Event Broker, ArcSight Installer and all supporting ArcSight product files will be installed into. For example: /opt/arcsight It is referred to as ArcSightRootFolder in this document.

C

CEF (Common Event Format)

CEF is an IT industry standard log format that transforms log file data into normalized, enriched and categorized log data.

D

Dedicated master infrastructure

Three or more master nodes route work to three or more worker nodes. A single Virtual IP (VIP) is used to load balance between three or more master nodes. Events are automatically load balanced between the worker nodes. There are no single points-of-failure and this configuration can be run in development, testing and production environments. It is required for a production environment. A minimum of six physical or VM environments are needed (three master, three worker).

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 (OS).

Docker Hub

Docker Hub is a cloud-based registry service that hosts the ArcSight Container code repositories. It links to Docker Cloud so you can deploy your EB images to your master and worker nodes. You must have a valid login to Docker Hub and authenticated credentials to access EB materials.

Docker Registry

The Docker Registry is a storage and content delivery system, holding named Docker images, available as different versions that can be downloaded to the EB Initial master node and the imbedded Containers can be configured and started.

E

Event Broker cluster

The Event Broker cluster consists of all master and worker nodes in the EB environment.

F

Fully Qualified Domain Name (FQDN)

The complete domain name for a specific host. The FQDN consists of two parts: the hostname and the domain name. For example, an FQDN for a hypothetical mail server might be mail.yourcompany.com.

L

Local Docker Registry

The Docker Registry location on the master and worker nodes in the Event Broker cluster. Event Broker software is launched and managed from the Local Docker Registry.

M

master node

A master node runs the ArcSight Installer and processes web services calls made to Event Broker. Master nodes connect to and are administered by the ArcSight Management Center. A minimum of 1 master node is required. For a Dedicated master infrastructure, a minimum of 3 master nodes must be available and running for a functioning cluster environment. (If one goes down, the cluster will still function, but the downed node must be brought back up as soon as possible.) In the EB cluster, the Initial master node is the master node that has been designated as the primary master node in the EB cluster. It is from this node that you will install the EB infrastructure.

S

Shared Infrastructure for master and worker nodes

Both master and worker nodes reside on the same operating systems. This is only supported in environments running less than 50K Events per Second (EPS). It is expected that this option will be discontinued in future releases. It is recommended to upgrade to Dedicated master infrastructure as soon as possible.

Single master Infrastructure

A single master node connects to 3 or more worker nodes. Events are automatically load balanced between the worker nodes. The master node is a single point of failure. A minimum of 4 physical or VM environments are needed (1 master, 3 worker). This configuration is recommended only for development or testing environments

V

Virtual IP (VIP)

To support high availability, a VIP is used as the single IP address to connect to a Dedicated master infrastructure that contains 3 or more master nodes. Load balancing between master nodes is built into the Event Broker processing. The FQDN of the VIP can also be used to connect to the cluster’s master nodes.

W

worker node

A worker node ingests, enriches and routes events from event Providers to event Subscribers. Worker nodes are automatically load balanced by the EB infrastructure.

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 (Event Broker 2.21)

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!