Fortanix®

Administration Guide

FORTANIX DATA SECURITY MANAGER INSTALLATION – ON-PREM GUIDE

VERSION 3.3

Fortanix Inc. | 3910 Freedom Circle | Suite 104 | Santa Clara, CA 95052 | United States of America | ☎ +1 (650) 943-2484 |
✉ info@fortanix.com | 🌐 www.fortanix.com

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PURPOSE

Welcome to the Fortanix Data Security Manager (DSM) Administration guide. The purpose of this guide is to describe the Fortanix DSM installation steps.

1.2 INTENDED AUDIENCE

This setup guide is intended to be used by technical stakeholders of Fortanix DSM who will be responsible for planning, performing, or setting up the monitoring and alerting solution, such as the Systems Administrator, Chief Information Officer (CIO), Analysts, or Developers.

1.3 DEPLOYMENT

This setup is a two-phase process as follows:

1. **System Installation and Configuration**: In this phase, you install the operating system and configure the network on the servers.
2. **Fortanix DSM Installation and Configuration:** In this phase, you install the Fortanix DSM service and configure the Fortanix DSM cluster.

2.0 QUICK SETUP

FIGURE 1: FX2200 SERIES 2 FRONT VIEW

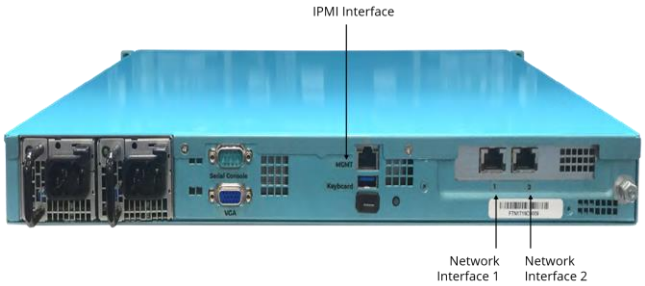

FIGURE 2: FX2200 BACK VIEW

To set up IPMI for FX2200 Series 2, refer to the IPMI Setup Guide.

2.1 LIST OF REQUIRED OPEN PORTS

For the list of open ports, refer to the Port Requirements.

3.0 PREREQUISITES

Before beginning the Fortanix DSM installation process, ensure that the following requirements are met:

1. You have at least three Fortanix FX2200 servers.
2. Network configuration (IP address, subnet mask, and gateway) has been assigned for each server.
3. If you are not planning on using an external load balancer, an additional IP address should be allocated for Fortanix DSM. This will serve as the cluster IP address to which Fortanix DSM clients will connect.
4. It is recommended assigning two Fully Qualified Domains (FQDNs) for the Fortanix DSM cluster. If the hostnames are assigned in your domain, for example, `your-domain.com`, then the preferred hostnames are `sdkms.your-domain.com`, and `sdkms-ui.your-domain.com`.
5. You should add DNS entries (A records) for the above two FQDNs, all mapped to the cluster IP address described above, to your DNS server.
6. All the ports mentioned in "List of required open ports" should be open between servers.
7. You should have the ability to issue or generate certificates for certificate requests (CSRs) generated by Fortanix DSM with the subject alternative name (SAN) containing the above stated hostnames.
8. You should be able to configure NTP on the servers. If the servers are not going to have access to public NTP servers, then they need to be able to connect to an NTP server on your network.
9. You should have received the software installation packages from Fortanix. The usual way of distribution is using a downloadable link from https://support.fortanix.com.
10. Ensure that none of the nodes in the cluster has the same hostname.

 **NOTE:** The hostnames **MUST** be in lower case.

11. If you are using a multi-node cluster, ensure that you update the hostname. Perform the following steps:
    1. Use the command `sudo hostnamectl set-hostname name` to update the hostname.
    2. Add the hostname in `/etc/hosts`.
    3. Verify that the hostname is correctly updated using the command `cat /etc/hostname`.
12. Ensure IPMI IPs are assigned to the servers if you want to remotely manage the servers. *For details of the IPMI setup, refer to the IPMI Guide.*
13. Get the default username/password for the FX2200 appliance. These will be distributed by email from Fortanix post shipment of the servers to the relevant contact person in your team.

4.0 SYSTEM INSTALLATION AND CONFIGURATION

4.1 REMOTE ADMINISTRATION BY IPMI

If your server has IPMI (Intelligent Platform Management Interface) setup, then you can remotely configure the server for the rest of the network configuration. *For details of the IPMI setup, refer to the IPMI guide.*

Once network configuration is completed, then direct SSH can be used for remote login. Perform the following steps for IPMI remote login:

1. Use IPMI web page accessible at the specified IPMI IP address through any browser. For example, http://192.168.1.25/#login
2. Use IPMI credentials provided by the Datacenter team which performed the IPMI setup. Go to Section **Remote Control** -> **Console Redirection**. This opens a Java console which provides the terminal view of the server. Now boot the server and login using the system administrator credentials provided by the Fortanix team. You can now follow the rest of the steps.

4.2 NETWORK CONFIGURATION

Configure a network interface with an IP address, subnet mask, and gateway, such that the servers are reachable from any intended client. You can do this using the console over IPMI if **IPMI** has been setup.

If you are using Fortanix supplied servers, note that these servers have two network interfaces. You can just use one network interface. If your network topology/deployment requires separating interfaces for external traffic and intra cluster traffic, then you may set up both the network interfaces.

For setting up the network, edit the `/etc/network/interfaces` file to specify IP address, gateway, netmask, and DNS server information. Refer to the following sample file to help you get started.

 **NOTE:**

- Replace `interface_name` with the name of the network interface on your server which are **`enp80s0f0`** and **`enp80s0f1`**.
- Replace `xxx.xxx.xxx.xxx` with appropriate values.
- After editing the file, save changes and reboot the server.

Example `/etc/network/interfaces` file:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto interface_name
iface interface_name inet static
  address xxx.xxx.xxx.xxx
```

```
gateway xxx.xxx.xxx.xxx
netmask xxx.xxx.xxx.xxx
dns-nameserver xxx.xxx.xxx.xxx
dns-nameserver xxx.xxx.xxx.xxx
```

To set the hostname on each node in the following file, remove `sdkms-server` and replace with the intended hostname:

```
sudo vi /etc/hostname
```

In the following file, add the `IP hostname` and/or `FQDN`.

```
sudo vi /etc/hosts
```

Reboot the system for changes to take effect.

4.3 LACP BONDING CONFIGURATION

The Fortanix DSM appliance can be configured to have LACP bonding.

Following is an example of network configuration that shows how to set it up:

1. Run the following command to install "**ifenslave**" package:

```
sudo apt-get install ifenslave
```

2. Example interfaces file location is `/etc/network/interfaces`.

```
# The ordering of these interfaces is important, see comment for
bond0 below.

auto enp80s0f0
iface enp80s0f0 inet manual
    bond-master bond0
    # This command is important, see comment for bond0 below.
```

```
    pre-up ifup bond0 &

# The ordering of these interfaces is important, see comment for
bond0 below.
auto enp80s0f1
iface enp80s0f1 inet manual
    bond-master bond0
    # This command is important, see comment for bond0 below.
    pre-up ifup bond0 &

# The ifenslave scripts are written for all interfaces being brought
up at once. However, `ifup` will bring interfaces up one-by-one.
Without doing anything extra, this will result in a deadlock or a
60-timeout, depending on the order of the master and slave
interfaces. We make sure to bring up the bond master simultaneously
with a slave, by running the ifup command for the master in the
background using `pre-up`. `ifup` implements locking and state
checking to make sure that a single device is only brought up once.
auto bond0
iface bond0 inet static
    bond-miimon 100
    bond-mode 802.3ad
    bond-slaves yes
    address ...
    gateway ...
    netmask ...
    dns-nameservers ...
```

5.0 FORTANIX DSM INSTALLATION AND CONFIGURATION

This section describes the tasks to install the Fortanix DSM software and configure the Fortanix DSM cluster.

5.1 INSTALL FORTANIX DSM

Perform the following steps to install the Fortanix DSM package on all the servers:

1. The latest installation file is available at https://support.fortanix.com/hc/en-us/sections/360001900792-Fortanix-Data-Security-Manager-Releases. **This needs a customer account, which Fortanix provides for the relevant contact person.**
2. Download and copy the Fortanix DSM installation file `sdkms_<version>_install.sh` to each server. The latest installation files are hosted on https://support.fortanix.com
3. Run the following command to install the package, replacing the package name with the filename of the package in your Fortanix DSM distribution.

```
sudo chmod +x sdkms_2.3.688-124_install.sh
sudo ./sdkms_2.3.688-124_install.sh
```

4. Run the following command to reboot the system if a new version of kernel was installed:

```
sudo reboot
```

5. Run the following command to reset on all servers:

```
sdkms-cluster reset --delete-data --reset-iptables
```

5.2 NTP CONFIGURATION

NTP is required to be able to run Fortanix DSM. If the servers are not going to have access to public NTP servers, you will need to update the NTP configuration file (`/etc/ntp.conf`) to specify NTP server(s) on your network. To enable local NTP when an external NTP server is not available, add the directive below to the `config.yaml` file (which is referenced in *Section 5.4*).

5.2.1 EXTERNAL NTP OPTIONS

1. Comment out servers for the NTP Pool Project and NTP server as a fallback section and add the following server IP addresses:

```
# pool 0.ubuntu.pool.ntp.org iburst
# pool 1.ubuntu.pool.ntp.org iburst
# pool 2.ubuntu.pool.ntp.org iburst
# pool 3.ubuntu.pool.ntp.org iburst
```

```
# Use Ubuntu's ntp server as a fallback.
# pool ntp.ubuntu.com

server 0.0.0.0
```

2. Run the following command to restart NTP:

```
sudo systemctl restart ntp
ntpq -p
```

5.2.2 INTERNAL NTP OPTIONS

You can run a local NTP server when an external NTP server is not available. This will run the NTP pod on each node and will keep the time synchronized within the cluster to one of the cluster nodes.

```
global:
  localntp: true
```

5.3 BEGIN FORTANIX DSM CONFIGURATION

This step needs to be run on **only one** of the servers.

5.3.1 SETUP DEPLOYMENT SPECIFIC CONFIGURATION FILE

Run the following command to copy the template file in the home directory and then edit it:

```
cp ~/config.yaml.example config.yaml
```

When deploying Fortanix DSM you have options with respect to the load balancer, remote attestation, internet subnet, and remote syslog logging. Based on the option you select, edit the `config.yaml` file as explained:

Load Balancer Options:

- Use the built-in load balancer – this is the default option.
    - Edit the `config.yaml` file to set the correct cluster IP address by adding the value of the parameter "`clusterIp`" under "`sdkms`" section.

```
sdkms:
  clusterIp: 10.0.0.1
keepalived:
  nwIface: enp80s0f0
```

- Use an external load balancer:
    - Edit the `config.yaml` file to add the following parameter under "`global`" section (consider the `yaml` file format and spacing).

```
sdkms:
  clusterIp: 1.2.3.4 #mandatory field, set to dummy for external LB
global:
  externalLoadBalancer: true
# keepalived:
#  nwIface: enp80s0f0
```

Remote Attestation Options:

- Remote attestation enabled.
    - This is the default option and no additional changes in `config.yaml` file is required.
    - This option requires that the appliances have internet access so they can connect to the Fortanix attestation proxy service.
    - Required URLs for remote attestation.
        - In this scenario, Fortanix DSM recommends setting up an **http proxy** to the internet. Make sure the following URLs are accessible through the proxy:
            - http://ps.sgx.trustedservices.intel.com:80, port 80
            - https://trustedservices.intel.com/content/CRL, port 443
            - http://trustedservices.intel.com/ocsp, port 80
            - http://whitelist.trustedservices.intel.com/SGX/LCWL/Linux/sgx_white_list_cert.bin, port 80
            - https://iasproxy.fortanix.com, port 443

- Edit the `config.yaml` file to add following parameters under "`global`" section (consider the `yaml` file format and spacing).

```
global:
  attestation:
    ias:
      url: ftx-proxy
      proxy: http://10.192.75.40:8080
sdkms:
  clusterIp: 10.0.0.1
keepalived:
  nwIface: enp80s0f0
```

- Remote attestation disabled.
  - Edit the `config.yaml` file to add the following parameters under "`global`" section (consider the `yaml` file format and spacing).

```
global:
  attestation: null
sdkms:
  clusterIp: 10.0.0.1
keepalived:
  nwIface: enp80s0f0
```

Internal Subnet Options:

- Kubernetes requires two internal networks for intra-cluster communication. By default, use `10.244.0.0/16` and `10.245.0.0/16` when unspecified.
- To configure custom subnet for Kubernetes internal network, set `serviceSubnet` and `podSubnet` under the "`global`" section (consider the `yaml` file format and spacing) as shown:

```
global:
  serviceSubnet: 172.20.0.0/16
```

```
  podSubnet: 172.21.0.0/16
```

Remote Syslog Logging Setting Options:

Fluentd supports forwarding logs to single syslog server and to multiple syslog servers.

- To configure forwarding the `'container'`, `'sdkms'`, and `'system'` logs from all the nodes in the cluster to one or more remote syslog servers, set the `host`, `port`, `protocol`, and `log_type` parameters for each type of log under "`advancedSyslog`" option of `fluentd` section of the `config.yaml` file (consider the `yaml` file format and spacing).

 **NOTE**:

- The "`remoteSyslog`" option, which allowed forwarding all logs to one remote Syslog server, will be deprecated in future releases of Fortanix DSM. Use "`advancedSyslog`" option to use the enhanced log forwarding functionality.

The valid values for,

- `host`: Any valid IP address
- `port`: Any valid port
- `protocol`: Either `tcp` or `udp`
- `log_type`: Either one of "`CONTAINER`", "`SYSTEM`", "`SDKMS`"

The following is an example for `advancedSyslog` if the forwarding is for 3 syslog servers:

```
fluentd:
  advancedSyslog:
   - host:
     port:
     protocol:
     log_type: "CONTAINER"
   - host:
     port:
     protocol:
     log_type: "SYSTEM"
   - host:
     port:
```

```
    protocol:
    log_type: "SDKMS"
```

Log Filtering Options:

There are three kinds of logs produced for every request:

- A **connection log**, indicating that a new connection has been made and the IP from where the connection is made.
- A **request log**, indicating that a request has been received, along with the IP from where that request is coming from and the associated HTTP Method and URL.
- A **response log**, indicating that a response has been returned, along with the associated status code.

When Log Filtering is enabled, it will cache request logs matching a shortlist of method-URL pairs. If the corresponding response is a success, the request and response logs are discarded. If the response is not a success, then both the request and response logs are printed.

The filtered combinations are:

- `GET /sys/v1/health`
- `HEAD /sys/v1/health`
- `HEAD /`

Since Kubernetes and most load balancers create a new connection for each request they make, this feature will attempt to profile individual IPs. If an IP is found to only make health check-related requests (that is, one of the three filter combinations listed above), then the connection logs from these IPs will also be filtered. If one of these IPs later makes a non-filtered request, its connections logs will no longer be filtered until it returns to making only filtered requests.

Log Filtering is enabled by updating the cluster-wide configuration. When enabled, the backend will no longer emit logs associated with successful health checks.

Depending on the cluster, this can amount to a substantial reduction in the volume of logs later forwarded over Syslog. The default is `false`.

```
filterConfig:
  useLogFilter: true | false
```

Hybrid Cluster Configuration:

With Release 3.25, Hybrid clustering of "Series- I and Series II ", "Azure and Series- I or Series- II ", and "AWS and Series- I or Series- II running Fortanix DSM in same (SGX/non-SGX) software mode" is possible. This allows nodes of different types to participate in a single cluster.

When using an internal DSM load balancer (that uses keepalived) with hybrid clusters consisting of Fortanix Series-I and Fortanix Series-II appliances, fetch the network interface names (Series-I = eno1 and Series-II = enp80s0f0f0) of the appliances and configure the `config.yaml` file as follows:

```
sdkms:
  clusterIp: 10.0.0.1
keepalived:
  nwIface: enp80s0f0,eno1
```

5.3.2 PROXY SUPPORT FOR OUTBOUND CONNECTIONS

You can add cluster-wide proxy configuration for outbound connections. This is defined in the `yaml` configuration files.

The Global proxy functionality is only available in SGX based deployments (FX2200 and Azure CC VMs). The configuration is defined in the `.global.proxy` entry. It holds two values:

- `url`: proxy value (mandatory).
- `exclude`: Comma separated list of hostnames and optionally port that should not use the proxy (optional).

```
global:
   proxy:
      url: <proxy_url>
      exclude: <host>,<host:port>
```

`exclude` Options:

- The hosts in `exclude` will be suffix matched. For example: `exclude: example.com` will match `example.com` and `subdomain.example.com` but will not match `example.com.net`
- Any leading '`.`' is removed. For example: `exclude: .example.com` is the same as `exclude: example.com`
- CIDR notation is accepted. For example: `111.222.0.0/16` will match all IP addresses in the format `111.222.xxx.xxx`
- The wildcards (`*`) or regex are NOT supported, and hostnames are not resolved to IP addresses.
- The `exclude` entry will automatically contain hostnames used internally.

5.3.3 HAPROXY SUPPORT

To enable HAProxy support, you must modify the `proxyHeaderConfig` field of the `config.yaml` file for the cluster and reapply it with the cluster tool. The field has three possible values:

- `disabled` is the default condition and is equivalent to the field missing from the `config.yaml` file.
- "`maybefrom=<cidr1>,<cidr2>,...`" where the CIDR ranges point to the server(s) initiating the proxy protocol enabled connection.
- "`requiredfrom=<cidr1>,<cidr2>,...`" where the CIDR ranges point to the server(s) initiating the proxy protocol enabled connection.

The minimum format is:

```
global:
  proxyHeaderConfig: disabled
  # OR
  proxyHeaderConfig: "requiredfrom=<cidr>,<cidr>,..."
  # OR
  proxyHeaderConfig: "maybefrom=<cidr>,<cidr>,..."
```

NOTE:

- It is important that there are no spaces within the quotes and that the entire value is wrapped in double quotes, as in the example above.
- The CIDRs should narrowly contain the addresses of the load balancers that will be injecting the Proxy Protocol headers and MUST exclude the cluster's pod subnet.

There are two options to securely handle transitions in accordance with the PROXY header spec. The sequence for enabling support is:

1. The `proxyHeaderConfig` field is `disabled`.
2. If you have an extant cluster that does not use proxy protocol, the process for enabling it is to first apply the `maybefrom` variant to the pods, with the given CIRDs pointing to the IPs of the load balancer. These should be as specific as possible. Once that is done, the proxy header is enabled on the load balancer serving the cluster.
3. Next, the config is changed to the `requiredfrom` variant with the same CIDRs. On fresh clusters, the header can start off as `requiredfrom`, if the load balancer already has proxy support enabled.

To disable it, the opposite steps are followed, that is:

1. First, the config is changed to `maybefrom`.
2. The load balancer's proxy support is disabled, and then the config is changed to `disabled`.

5.3.4 SEALING KEY POLICY

A Sealing key is used to wrap a cluster master key. Each node has its own unique Sealing key. Sealing key is derived and not saved anywhere in the system.

The Sealing Key policy defines what type of Sealing key should be used in the cluster.

Possible values are:

- **SGX** - This is the default value. With this policy SGX sealing key is used. SGX sealing key cannot be retrieved from outside SGX. This provides security guarantees for sealing key.
- **Personalization Key** - Personalization key is generated using Fortanix DSM Deterministic random bit generator (DRBG) and stored in tamper protected key storage module. In this policy type, the sealing is derived using SGX sealing key and

personalization key. This provides additional protection of sealing key. Personalization key is zeroized upon tamper, which in turn disallows deriving the sealing key.

- **Recovery Key** – In this policy type, the sealing key is derived using personalization key and a recovery key. Recovery key is automatically generated and can be retrieved by a sysadmin. In this policy, in addition to using personalization key based sealing key for wrapping cluster master key, recovery key is also used to wrap the cluster master key. This allows the sysadmin to recover a node using recovery key in case personalization key is zeroized. After the setup, a sysadmin must retrieve the recovery key and store securely.

Setup

When setting up a fresh cluster, add the following lines under the `global` section in `config.yaml` and provide one of the values mentioned above:

```
sealingKeyPolicy: VALUE
```

Example:

```
sealingKeyPolicy: personalizationKey
```

Node Recovery

If the personalization key or recovery key policy is used and if the personalization key is zeroized, then the node goes in "`NeedsReocver`y" state. In this state for the Fortanix DSM pod on this node to become functional, recovery needs to be performed. Recovery will allow the node to unwrap the corresponding cluster master key blob.

To perform recovery, run following script as root:

```
/opt/fortanix/sdkms/bin/node_recover.sh
```

Recovery can be done with or without a recovery key.

- If the recovery key is not available or if the sealing key policy value is "`personalizationKey`", then recovery requires that there be other working nodes in the cluster. In that case run the command as follows:

```
sudo /opt/fortanix/sdkms/bin/node_recover.sh --node-name NODE_HOST_NAME
```

Where,

`NODE_HOST_NAME` is node name where recovery is to be performed.

- If the recovery key is available and the sealing key policy is "`recoveryKey`", then recovery can be done even on a standalone node. In that case run the command as follows:

```
sudo /opt/fortanix/sdkms/bin/node_recover.sh --node-name NODE_HOST_NAME
--recovery-key RECOVERY_KEY_VALUE
```

Where,

- `NODE_HOST_NAME` is node name where recovery is to be performed.
- `RECOVERY_KEY_VALUE` is the value of recovery key.

Updating sealing key policy in the existing cluster

For existing clusters, the sealing key policy is by default "`sgx`". The procedure for changing the sealing key policy is as follows:

1. Update `config.yaml` to add one of following lines under the `global` section:

```
sealingKeyPolicy: personalizationKey
sealingKeyPolicy: recoveryKey
```

2. Apply the updated config by running following command:

```
sudo sdkms-cluster deploy --config ./config.yaml --stage DEPLOY
```

3. Check if the new value was applied by running the following command and check the value of "`sealingKeyPolicy`".

```
sudo KUBECONFIG=/etc/kubernetes/admin.conf kubectl get cm config-
values -oyaml
```

4. Run the following command:

```
sudo /opt/fortanix/sdkms/bin/update_sealing_key_policy.sh
```

5. The policy change will be effective on the next upgrade.

5.3.5 CREATE FORTANIX DSM CLUSTER

1. Run the following command to create a Fortanix DSM cluster:

```
sudo sdkms-cluster create --self=<server ip address/subnet mask>
--config ./config.yaml
```

2. Check the status of all pods to verify all pods are running correctly. Run the following command to list the pods and their status:

```
sudo KUBECONFIG=/etc/kubernetes/admin.conf kubectl get pods -o
wide
```

The Fortanix DSM related pods may continue to crash until the certificates are installed (next step), but verify that the pods related to Aesmd, Elasticsearch, and Cassandra are running.

If you want to set up a cluster that spans multiple sites or data centers, refer to the steps in the Data Center Labeling guide to label which nodes are in what datacenter.

5.3.6 CONFIGURE DATA CENTER LABELING

After all nodes have successfully joined the cluster, you must perform data center labeling. Data center labeling configuration in a multi-site deployment improves read resiliency by enabling requests to read data from the local data center and supports the Read-Only mode of operation when a global quorum is lost and a local quorum is available. Please refer to the Fortanix DSM Data Center Labeling guide (use the automated script to configure DC labeling). *Please refer to the Fortanix DSM Data Center Labeling guide (use the automated script to configure DC labeling).*

For more details, refer to Fortanix DSM Read-Only Mode of Operation.

5.4 CONFIGURE OTHER NODES

Now that the installation is complete, join all other nodes to the cluster by running the following join command.

NOTE: Do not perform the join operation on multiple nodes simultaneously. The join operation of a node should be started once the previous node has successfully joined the cluster.

- If your nodes are in the same subnet, run the following command:

```
sudo sdkms-cluster join --peer=<server ip address/subnet> --
token=<kubeadm-token>
```

- If your nodes are in the different subnets, run the following command:

```
sudo sdkms-cluster join --peer=<server ip address> --token=<kubeadm-
token> --self=<node IP address>
```

In the above specification,

1. Server IP Address corresponds to the node's IP address on which cluster was created. Specify it with the subnet.
2. The node IP address corresponds to the IP address of the joining node.

Example join commands for a joining node (10.198.0.65) to a created cluster on a node (10.198.0.66) on a /24 subnet:

```
sudo sdkms-cluster join --peer=10.198.0.66/24 --
token=525073.715ecf923e4ae1db
OR
sudo sdkms-cluster join --peer=10.198.0.66 --
token=525073.715ecf923e4ae1db --self=10.198.0.65
```

The token can also be retrieved by executing on the first host:

```
sudo kubeadm token list
```

5.5 INSTALL CERTIFICATES

WARNING: The process described in this section will generate the keys and certificate requests for TLS connectivity. Save the passphrase for the private key as it cannot be recovered later.

Fortanix DSM requires two SSL certificates for the services, one for the Main API service and another for Static asset service. The Certificate Signing Request (CSR) generation is supported for both these certificates which can be signed by your preferred CA provider.

1. Generate a Certificate Signing Request (CSR) to get an SSL certificate. The CSR contains information (for example: common name, organization, country) which the Certificate Authority (CA) will use to create your certificate. Generate CSRs by running the following command on the node where the cluster was created:

   ```
   sudo get_csrs
   ```

This script is going to generate Certificate Signing requests for both the Fortanix DSM cluster and UI. It also created the first system administrator(sysadmin) account that can log into Fortanix DSM. Domain names can be provided in two formats.

a. Simple, where Domain name and SANS can be provided. For example, www.fortanix.com.
b. Distinguished, where full DN string can be provided. for example: CN=www.fortanix.com O=Fortanix L=Mountain View ST=California.

There is also an option to add Subject Alternative Names. More than one can be added, each one on a new line.

The following will be then asked to enter through the script:

a. Domain name for the cluster (Main)
b. Domain name for the UI (Assets)
c. Cluster name
d. Sysadmin email address
e. Sysadmin password

NOTE: For authentication using mutual Transport Layer Security (TLS), the standard API key-based authentication interface will not work. To support mutual TLS authentication in Fortanix DSM, add an additional interface as SAN Name (apps.sdkms.dev.com) on the main certificate. *For more details, refer to the Mutual TLS guide*.

The CSRs are printed on the terminal named “Cluster Certificate Request” and “UI Certificate Request”. These certificates need to be signed by a Certificate Authority.

```
testuser@ali2:~$ sudo get_csrs
waiting for an sdkms pod with the backend running
There are 2 methods for entering domains name
    1. Simple, eg www.fortanix.com
    2. Distinguished, eg CN=www.fortanix.com O=Fortanix L=Mountain View ST=California
Which method would you like to use for the Main domain? :1
Choice = Simple
Please enter domain to be used for Main:dev-cluster.fortanix.com
Subject Alternate Names(SAN) for Main
Enter each name on a new line and enter an empty line to conclude:dev-apps.fortanix.com

Which method would you like to use for the Assets domain? :1
Choice = Simple
Please enter domain to be used for Assets:dev-cluster.fortanix.com
Subject Alternate Names(SAN) for Assets
Enter each name on a new line and enter an empty line to conclude:dev-apps.fortanix.com

Cluster name:testCluster
Sysadmin email address:john.doe@fortanix.com
Password:
Waiting to Join Cluster
Cluster Certificate Request:
-----BEGIN CERTIFICATE REQUEST-----
MIICtTCCAZ0CAQAwIzEhMB8GA1UEAwwYZGV2LWNsdXN0ZXIuZm9ydGFuaXguY29t
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAw55ccC6cCL9ZnnfOvWiH
Hso0d9t/XzjZAncXNfz/0f3u0RN5F9+Pg/VXbnFb3OAzmGPY0yDo7ugpNmTJJMD4
34d0tkGOanK892Gk1lgUaTs+C2+On60NTvdPd9GUv/VRara6Gun0NCtpK0eOw6Lv
TzU1kjjg3ccBOxNujFmmTApPdwGPVSHttEf+SoilqwZ2df4DLQfEprsHVw41UNWp
upuWFGHxacO2Ymb30P0rAC9eMWdxcrhuIlS2NEwh9UKOIgk8CLwjhNcSumKzWSkN
egyekEr1NCemF67FRazBno7NXPpjqXMpg/u8LFrhl4TR5ETEMKDyxvnOmprPeoGz
hwIDAQABoE0wSwYJKoZIhvcNAQkOMT4wPDA6BgNVHREEMzAxghVkZXYtYXBwcy5m
b3J0YW5peC5jb22CGGRldi1jbHVzdGVyLmZvcnRhbml4LmNvbTANBgkqhkiG9w0B
AQsFAAOCAQEAuBqcsc49XhhugEUQXgQwxz8zuz87pI3fSwaZRiec+Hq0mTNAqEai
Q5VD7QsG1f/VioK3gzEzjaPcXQF5N8tYKggCF//xCf8aiTh7sIvxk2gcWDGJ8bPa
wLdzqWzDZ6A4x4wEWtpDKJbXwP8pbaPHYPgzb91if3dRgt/LUmnG5ZrDCh4ueG7K
GGTjjnJUqdEg2TG+Fo2r7A937lUpx4UWG4RCgHbawERZZlDkc1GYpUq4GgjibYRy
5LNEuDIaih5M/yuktpyPuFnzbZ9EMcXEZo7HffLpgzCv7Gaj5FRMTdAauT+KjMhM
MzauDb5Yfw2IJ95Pjigbev+H/B08hPODzA==
-----END CERTIFICATE REQUEST-----
secret "sdkms-secrets" created
UI Certificate Request:
-----BEGIN CERTIFICATE REQUEST-----
MIICmzCCAYMCAQAwIzEhMB8GA1UEAwwYZGV2LWNsdXN0ZXIuZm9ydGFuaXguY29t
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0vpnALQZ7GWqQfEYMu8V
hNclwfqOeCAxQJeGV3rLKC4jUKa23NrPSybdYUtM86Tv+H479mgM1Rl01KyfAEXP
w4VUfRLUnrUMsyk6VjUy1KYG8gEFzB/LAOIAER2YOte3qZoSmHD7IAEIIfZOxjyt
9oeL+RDfk98xUvLbAS9xODXMr5sBWfCe6AmBVh9HHlsatLBlClF5TEI8Suxus1lF
xBJJIGURzxrGR187ONoyIuFaLwfDcWK6vzR9KZWN/04pb3+zzvTcZ2okn8K61yyI
oUxE1YriJQIFUrHjeEsBMk1CYHiY77ngLGC7xX55Vo1U/WSFQCVeExJd5LKSe7so
JQIDAQABoDMwMQYJKoZIhvcNAQkOMSQwIjAgBgNVHREEGTAXghVkZXYtYXBwcy5m
b3J0YW5peC5jb20wDQYJKoZIhvcNAQELBQADggEBAHZX+pyN4gxm8QXJWjon+z3Q
1CiwqOjarRsgtBRmvEmbpj7i9WktdFecGbefyfd0FnfnljvuL66ca+d2UCOiPRDX
Z3Dr3gZxbfBwabi0RNqsPVZdb8ZouASUOG4M+8Sp4osQ5Zsd4p66Me69Bsy59GGN
RKbKK/ffgWsj2UwZseU1Hp/hFLJwB/8GgCNTQFfIcvNQiFDoP3Lw7ptI4y8SOhs0
r7B7B3g6misJcS0nzMbYjxhWw+VgZOmnKQ8D+erIdnHv0DAmeusCITraCju2gvy7
QlttR3BZfvSSHFz8EWTTRZpvyEDx3QEXctXmmDUb3DIRlp2g8vhyMeakDgDrrVw=
-----END CERTIFICATE REQUEST-----
```

FIGURE 3: CERTIFICATE SIGNING REQUEST

A certificate request is created for both the CSRs, one for Main and one for Assets. You will need to sign both the CSRs from one of the CAs or locally sign it yourself.

NOTE: If `get_csrs` fails then view the logs at the location `/var/log/get_csrs.log`

2. Run the following command to install the signed certificates once you have them from CA:

```
sudo install_certs
```

This script asks the user to paste the signed certificate chain (leaf certificate first) for both Main (Cluster) and Assets (UI) domain.

```
root@azuresdkms-38release-setup1-vm1:/home/fortanix# install_certs
waiting for an sdkms pod with the backend running
Enter Signed Main certificates (enter an empty line to complete):-----BEGIN CERTIFICATE-----
MIIDGTCCAgGgAwIBAgICAVgwDQYJKoZIhvcNAQELBQAwHTEbMBkGA1UEAwwSVGVz
dCBEZXZDbHVzdGVyIENBMB4XDTE5MTIwMzA2NDA0NVoXDTIwMTIwMjA2NDA0NVow
NzE1MDMGA1UEAwwsYXp1cmVzZGttcy0zOHJlbGVhc2Utc2V0dXAyLnRlc3Rmb3J0
YW5peC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCaHPpiUepG
izm31HUZhUnmoRuVBvlufTlx97EXQPx6d8calftPzkRWJnt333yczMBImk/TXcZ4
EJ72Cj/TNHQsIoqb+PEQfmXuymmd0yKEU9jG3FwAHpaD7+fTEc9J8puTpUBkt7tL
5eQcyTak3pX3gkXEe1+4Aj0VI0VStXOfhYSIDxTVoGiJDQI5abUYY7R3uf/T+Xv/
XNEY4m25+skN19NNPngnqQnUWXJMH88qJz12mieKOFqbH83qxyJDSg0peEwW2Tpf
xkL/0oT5jE5pTezMOWF6Q91CIS59Wb5bSW+7M1tn99rXfkX1Ro5HAnE9UVTy4ke2
FXcN3735s/0ZAgMBAAGjSTBHMAwGA1UdEwEB/wQCMAAwNwYDVR0RBDAwLoIsYXp1
cmVzZGttcy0zOHJlbGVhc2Utc2V0dXAyLnRlc3Rmb3J0YW5peC5jb20wDQYJKoZI
hvcNAQELBQADggEBAJqHQDezDqX4uZEPqmHMKo7OLvPlWPLpC0KsYFrXsdg+ouhx
f0exvUDmDyQjn+rmhCcJt53l7U3B0mYVjV2weWDjXUprAlIvgQoNk1vohsKoEyH9
cwfW0DNgN9MZ8YDUgQ3WiVzbZCKR8kC/g4mvo5DwkJhQrRgepTcVilQrLREldVfr
Hl8IpTUFI8iuUOItBKb3ce3oIJ3Qdg706eIwOfR7nwJHcYKc4MrbGxKCaYHMCbVn
11CxsYQEFoKIcVRhgGoU73uyASAz3gLs01Af8bihouqFP6SnVccaMbaZPLMQW318
lSDWVQlzsRnHSeDMyWpijQc4pxGAelALQfcchdM=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIC0zCCAbugAwIBAgIJAOg7z1D75ZbVMA0GCSqGSIb3DQEBCwUAMB0xGzAZBgNV
BAMMElRlc3QgRGV2Q2x1c3RlciBDQTAeFw0xOTA2MTgyMDM5MDhaFw0yOTA2MTUy
MDM5MDhaMB0xGzAZBgNVBAMMElRlc3QgRGV2Q2x1c3RlciBDQTCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAMY+15YzjTKHe/hS5UpUth4Mu2Dq2TBZ0JW+
PqVuVpZ5r4VLhunF4wTn9j96btGkVhAx/K0O/hvdeNh+v5i1QPRZ6+t+DNvNj57g
lyYnEtq0U1uhH/YHdX8qPJnOk8+DDNW4t8H+bmuTTG/3L3RwGtcsHunMXjIKKrH9
BdMOMFzKtlgGV21OCd+pGvSGbxk/9ygbSypOQgcIYYld/6dPonS1x0zJxywCRMoc
iLuMb6j9cDW9c3CSbJWh9/VOXkcL0Fzeq1z3BLP75u8KoTX5nQdFWp/TcZLLyHLE
AXuI2Aq5AgDG0DCvo29DzXSHGVeeyO7Uc//u+gq4/HMovmdImMUCAwEAAaMWMBQw
EgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsFAAOCAQEAAcKwxemkjOv5
rvlBJsMmydDwCZBnZ8rMw7QXCpCTbD6IBcLTiuH/LuHVPnYXN0rgtcjHfmcpUjcP
tyBVQRR5iF4iZuajDvhEGn3ocy8SYUm/bnCc5hEMOtlkBxXX1Ua2COgQjvMGExAH
B+eV5+r+rpuop/xLF2klNeyB+QvQbi4FMDFRyT7L8z8uU3PKoXk+1HT/eS45K8+X
yc4vEOEmnx/u9ZdKa3BCMc2J/zpJ+eVgFNAaUj7oCjF5tLmztyJ+lLNXBzLw4MDs
hVqNBbcu29GFwJysUIqbuj/2sg7Rk27KNcPH6tT4KFxyaD9G9oboOI+RHr+oOVQm
MLWVORdRjw==
-----END CERTIFICATE-----
```

FIGURE 4: INSTALL CERTS IN MAIN

```
Enter Signed Asset certificates (enter an empty line to complete):-----BEGIN CERTIFICATE-----
MIIDGTCCAgGgAwIBAgICAVkwDQYJKoZIhvcNAQELBQAwHTEbMBkGA1UEAwwSVGVz
dCBEZXZDbHVzdGVyIENBMB4XDTE5MTIwMzA2NDA0OFoXDTIwMTIwMjA2NDA0OFow
NzE1MDMGA1UEAwwsYXp1cmVzZGttcy0zOHJlbGVhc2Utc2V0dXAyLnRlc3Rmb3J0
YW5peC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDH0zLZJRQr
SvGhxz2btH7G+a/VpFuspDhgKUgRXuGW+sN7GZ+y7JK6piINjOFXPJrBk7MD7w3J
wHmAclo7BD51a0pYtTdLfimFfjbCy4B+R1uz9qyTQM8F4VMgaGfEkz1Hhqx+A9Qt
pLzFo/kRBlylO7N/s2c+grYdQ/pcbffa8N2ksPnwg/jCaihEs0snPoUHJSmJGPjP
D0zs/RyRxSLAquJ7Fbwg8SGdnvd8Bzo4jnohTW1fv6BmIjBM6mhM8+0w/qlnUBQB
SsDf66WhE9oH477vfAesc4ae+atus5aT1GczMpHRNTerW7nBkK/5yQ1g+B/JFPfJ
40E4+O/BlQl7AgMBAAGjSTBHMAwGA1UdEwEB/wQCMAAwNwYDVR0RBDAwLoIsYXp1
cmVzZGttcy0zOHJlbGVhc2Utc2V0dXAyLnRlc3Rmb3J0YW5peC5jb20wDQYJKoZI
hvcNAQELBQADggEBAGXRfmvSIzMjQQr4okqunaDn0oqwMZ/Lh0+9nXDGvv/7cayc
fy/820W987mrxU7iTHHOM+bVl6xPy3A+niRL02/aiLHU/a1LhEMp8vyr5GvYo8Lw
4uwIzWl8ZVMdEKcHMV0Xacd8ZHgyoYyf6Tf9KVn9bWuTjOXcYJ9bJUTIHk9fYZrR
3Z75N7kx9unyr7qZWcZHMmphyztjN8+1PA1pKDhNnAsecuWZGWFvxkizYq28DAhl
JvI1Wr+092ZeMboemaLCTcpM95bZZlMd2X1ebQQO2BnJHR5Wpq8OfnpI41vBoFwU
brH/LJFH5a/zvK/dcT+dCKXKLPfOybGn4FBN1Wg=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIC0zCCAbugAwIBAgIJAOg7z1D75ZbVMA0GCSqGSIb3DQEBCwUAMB0xGzAZBgNV
BAMMElRlc3QgRGV2Q2x1c3RlciBDQTAeFw0xOTA2MTgyMDM5MDhaFw0yOTA2MTUy
MDM5MDhaMB0xGzAZBgNVBAMMElRlc3QgRGV2Q2x1c3RlciBDQTCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAMY+15YzjTKHe/hS5UpUth4Mu2Dq2TBZ0JW+
PqVuVpZ5r4VLhunF4wTn9j96btGkVhAx/K0O/hvdeNh+v5i1QPRZ6+t+DNvNj57g
lyYnEtq0U1uhH/YHdX8qPJnOk8+DDNW4t8H+bmuTTG/3L3RwGtcsHunMXjIKKrH9
BdMOMFzKtlgGV21OCd+pGvSGbxk/9ygbSypOQgcIYYld/6dPonS1x0zJxywCRMoc
iLuMb6j9cDW9c3CSbJWh9/VOXkcL0Fzeq1z3BLP75u8KoTX5nQdFWp/TcZLLyHLE
AXuI2Aq5AgDG0DCvo29DzXSHGVeeyO7Uc//u+gq4/HMovmdImMUCAwEAAaMWMBQw
EgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsFAAOCAQEAAcKwxemkjOv5
rvlBJsMmydDwCZBnZ8rMw7QXCpCTbD6IBcLTiuH/LuHVPnYXN0rgtcjHfmcpUjcP
tyBVQRR5iF4iZuajDvhEGn3ocy8SYUm/bnCc5hEMOtlkBxXX1Ua2COgQjvMGExAH
B+eV5+r+rpuop/xLF2klNeyB+QvQbi4FMDFRyT7L8z8uU3PKoXk+1HT/eS45K8+X
yc4vEOEmnx/u9ZdKa3BCMc2J/zpJ+eVgFNAaUj7oCjF5tLmztyJ+lLNXBzLw4MDs
hVqNBbcu29GFwJysUIqbuj/2sg7Rk27KNcPH6tT4KFxyaD9G9oboOI+RHr+oOVQm
MLWVORdRjw==
-----END CERTIFICATE-----

secret "sdkms-secrets" patched
deployment "sdkms" patched
deployment "sdkms-proxy" patched
Waiting for rollout to finish: 0 of 1 updated replicas are available...
```

FIGURE 5: INSTALL CERTS IN ASSETS

NOTE:

- A catch-all cert cannot be used, and a multi-domain cert for BOTH cluster and UI, cannot be used.
- Multi-domain certs can be used when a Subject Alternative Name (SAN) is provided, however, the SAN cannot match the DNS name used for the cluster.
- Typically, a Standard TLS certificate, without a Subject Alternative Name, is enough for most installations.
- To renew the cluster certificates on both Main (Cluster) and Assets (UI) use the `gets_csrs` and `install_certs` command that you used in *step 1* and *step 2* above for the cluster setup.

6.0 USING FORTANIX DSM

You can access the Fortanix DSM web interface using the hostname assigned to the Fortanix DSM cluster (for example `https://sdkms.<your-domain.com>`). *For detailed information on Fortanix DSM usage, refer to the https://support.fortanix.com/sdkms.*

7.0 ADD NODE TO AN EXISTING FORTANIX DSM CLUSTER

1. Run the following command to create the `kubeadm` token from one of the existing nodes in the cluster.

```
sudo kubeadm token create
```

2. Join the new node(s) with `sdkms-cluster join` command. Ensure the installer script has been executed on these nodes.
   - If your nodes are in the same subnet, run the following command:

```
sudo sdkms-cluster join --peer=<existing node IP>/<subnet-mask> --to
ken=<token>
```

   - If your nodes are in the different subnets, run the following command:

```
sudo sdkms-cluster join --peer=<server ip address> --token=<kubeadm-
token> --self=<node IP address>
```

8.0 REMOVE NODE FROM AN EXISTING FORTANIX DSM CLUSTER

1. Run the following command to identify the name of the node to be removed under the header "NAME":

```
sudo -E kubectl get nodes
```

2. Run the following command to remove the node from the cluster using the `<node name>`:

```
sudo sdkms-cluster remove --node <node name> --force
```

3. Run the following command to reset the node:

```
sdkms-cluster reset --delete-data --reset-iptables
```

TIP: There can be previous versions of UI pods which show up in the pending state.

Ignore them/ delete the older version from UI.

9.0 BEST PRACTICES

- Only issue accounts to trusted administrators.
- Utilize strong passwords.
- Monitor logs.
- Restrict physical access to the appliance to trusted administrators.
- Create two System Administrator accounts.

10.0 TROUBLESHOOTING AND SUPPORT

PROBLEM	RESOLUTION
The repair job fails.	The `cassandra-repair` cronjob must be restarted manually after fixing the failure.
Any of the scripts fail at any step.	They will print a detailed error message. Report the problem to Fortanix support (support@fortanix.com) and provide the script output.

Due to a known issue in Palo Alto Networks (PAN) firewalls, VXLAN traffic required by Fortanix DSM for intra-cluster communication could be mysteriously dropped without notification. After implementing the Fortanix DSM Port Requirements, connectivity between DSM nodes can be confirmed. Fortanix DSM creates an overlay virtual network for intra-cluster communication over UDP port 8472 as part of the standard deployment. On this virtual network, pinging or using CURL from one DSM node to another can also confirm connectivity. Unfortunately, when a PAN FW is in the network flow between DSM nodes, only a fraction of the packets sent over the VXLAN make it to the desired node due to this known issue. This lack of network consistency results in DSM nodes joining an existing cluster appearing to fail without much detail as to the cause. Cassandra pods may not be able to gossip with an expected peer, which is one clue that the cluster is not wholly joined. The joining node appears to have successfully joined the Kubernetes cluster. Still, the secure data synchronization expected to occur over this overlay VXLAN is never completed, leaving the cluster in a non-functional state.

Refer to the PAN Known Issues List, specifically issue PAN-160238, that explains this issue and the suggested workaround:

PAN Known Issues List

PAN-160238 - If you migrate traffic from a firewall running a PAN-OS version earlier than 9.0 to a firewall running PAN-OS 9.0 or later, you experience intermittent VXLAN packet drops if the TCI policy is not configured for inspecting VXLAN traffic flows.

Workaround:

On the new firewall, create an app override for VXLAN outer headers as described in What is an Application Override? and the video tutorial How to Configure an Application Override Policy on the Palo Alto Networks Firewall.

Turning on Tunnel Content Inspection (TCI) appears to resolve the issue in addition to the suggested workaround above from PAN to create an Application Override policy for the Fortanix DSM nodes. After making these recommended changes, VXLAN traffic should become more reliable and stop inadvertently dropping packets. The Fortanix DSM cluster should be able to communicate appropriately, and the Fortanix DSM node should be able to successfully join the existing cluster.

11.0 DOCUMENT INFORMATION

11.1 DOCUMENT LOCATION

The latest published version of this document is located at the URL:

https://support.fortanix.com/hc/en-us/articles/360020884152-Fortanix-Data-Security-Manager-Installation-Guide

11.2 DOCUMENT UPDATES

This document will typically be updated on a periodic review and update cycle.

For any urgent document updates, please send an email to: support@fortanix.com