Deployment Guide

FortiSOAR 7.6.6

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

March, 2026
FortiSOAR 7.6.6 Deployment Guide
00-400-000000-20210113

TABLE OF CONTENTS

Change Log

Date	Change Description
2026-04-23	Updated the Deploying FortiSOAR using offline repositories chapter.
2026-03-30	Initial release of 7.6.6

Introduction

Fortinet Security Orchestration Platform™ (FortiSOAR™) is a scalable, awareness-driven, and encrypted security management intelligence platform. FortiSOAR is a centralized hub for your security operations and dramatically improves the effectiveness and efficiency of your security operations teams, by providing automation and customizable mechanisms for prevention, detection, and response to cybersecurity threats.

The FortiGuard Distribution Network (FDN) has updated its CA certificates. FortiSOAR release 7.6.6. includes the updated certificates to ensure continued secure communication with FDN services and to prevent any disruption to license synchronization.

To support disk sizes larger than 2 TB, FortiSOAR OVAs starting with the 7.5.0 release come pre-configured with a GPT-based disk layout. Previously, FortiSOAR OVAs were shipped with an MBR-based disk layout, which limited disk management to a size of 2TB. If you already have a FortiSOAR instance and need a partition larger than 2 TB, we recommend creating a new FortiSOAR VM on release 7.5.0 or later and utilizing the Export and Import wizards to migrate your data from the old instance to the new one.

From release 7.6.1 onwards, support for an on-demand feature, i.e., ability to encrypt all FortiSOAR's 'data at-rest' is added. For more information, see the Encrypting FortiSOAR's Data at Rest chapter in the "Best Practices Guide."

For information on deploying FortiSOAR, see the Deploying FortiSOAR chapter.
For information on deploying FortiSOAR using offline repositories, see the Deploying FortiSOAR using offline repositories chapter.
For information on deploying FortiSOAR on a docker platform, see the Deploying FortiSOAR on a docker platform chapter.
For information deploying the FortiSOAR Docker on an EKS cluster, see the Deploying FortiSOAR Docker on an Amazon Elastic Kubernetes Cluster chapter.

FortiSOAR is also available as a hosted option on FortiCloud. For information of FortiSOAR on FortiCloud, see the FortiSOAR Cloud documentation.

Purpose

Use the deployment guide to deploy the FortiSOAR virtual appliance using VMware, the ESX/ESXi server and AWS.

This document provides you with all the procedures for setting up FortiSOAR in your environment, including deploying FortiSOAR, the initial configuration for FortiSOAR, and troubleshooting of FortiSOAR.

Prerequisites

Starting with release 7.6.5, the csadmin user's sudo privileges are restricted to only the commands required to work with FortiSOAR, instead of providing full 'root' access. This enhancement aligns with the principle of least privilege and reduces exposure to sensitive system files. Therefore, commands such as `yum`, `systemctl`, `csadm`, etc, must be prefixed with `sudo`, for example, `sudo csadm --help`.
To open or edit a file, prefix the command with '`sudo`' and specify the file's full path (`sudo vi <full path of file>`).
For example, `sudo vi /opt/cyops-auth/utilities/das.ini`.
Additionally note that for security reasons, '`root`' access is provided via the system console and is not available over SSH.

FortiSOAR does not support deployment in ***IPv6-only*** environments.

Before you deploy FortiSOAR, ensure you have done the following:

- Setup a system with minimal edition of either Rocky Linux version 9.3/9.4/9.5/9.6, or RHEL version 9.3/9.4/9.5/9.6, if you are installing FortiSOAR using the installation script. Release 7.6.6 has been tested with RHEL 9.6 and Rocky Linux 9.6.
  OR,
  If you are deploying FortiSOAR using the OVA, download the FortiSOAR Virtual Appliance and deploy it to VMware (via vSphere or vCenter) or AWS.
  NOTE: When installing FortiSOAR using the installation script, it is highly recommended to install FortiSOAR on a non-hardened operating system (OS). After the installation, the OS will undergo automatic hardening by FortiSOAR. Avoid any additional hardening of the OVA or consult with FortiSOAR support, to prevent issues in the FortiSOAR running instance. Installing FortiSOAR on a pre-hardened OS can lead to installation failure and issues with starting services, file permissions, etc. Any external monitoring agents must function correctly under FortiSOAR's current SELinux policy. If the required SELinux context is not properly configured, the customer must uninstall these agents before performing an FortiSOAR upgrade that includes an underlying OS version upgrade.
- Decide the Hostname and IP address.
- Know the DNS server IP address for the appliance.
- Disable the IPv6 protocol from your VM where you are deploying FortiSOAR if you are not using the IPv6 protocol. This is necessary because, starting with RHEL 9.0 or Rocky Linux 9.0, `ifcfg` files are deprecated. To disable IPv6, you should check the appropriate NIC config file and make changes in `/etc/NetworkManager/system-connections` (see the Using NetworkManager to disable IPv6 for a specific connection document). Starting with RHEL 9.0 or Rocky Linux 9.0, network configurations are stored at `/etc/NetworkManager/system-connections/` in a 'key file' format.
- Locale is set to `en_US.UTF-8`. FortiSOAR release 7.5.0 and later supports PostgreSQL 16. The postgresql-16 service will fail if the locale is not set to `en_US.UTF-8`, so make sure that the locale of your VM where you are deploying FortiSOAR is set to `en_US.UTF-8`. To install and apply the `en-US.UTF-8` locale on your system use the following commands:
  ```
  # sudo yum install glibc-langpack-en -y
  # sudo localectl set-locale en_US.UTF-8
  ```
- Company-specific SSL certificate, if you want to change the default certificate.
- Optionally configure an SMTP server and an NTP server. The SMTP server is used for outgoing notifications once the system is configured. The NTP server is used to synchronize the machine time after deployment.

> ⚠ Do not alter the `/etc/sudoers` file, as the users added by FortiSOAR in the `sudoers` file are necessary for FortiSOAR's functioning. Since certain commands are only accessible to these users, changing the `/etc/sudoers` file could interfere with FortiSOAR's functioning and cause issues such as services not starting or an inability to log into FortiSOAR's GUI.

Browser Compatibility

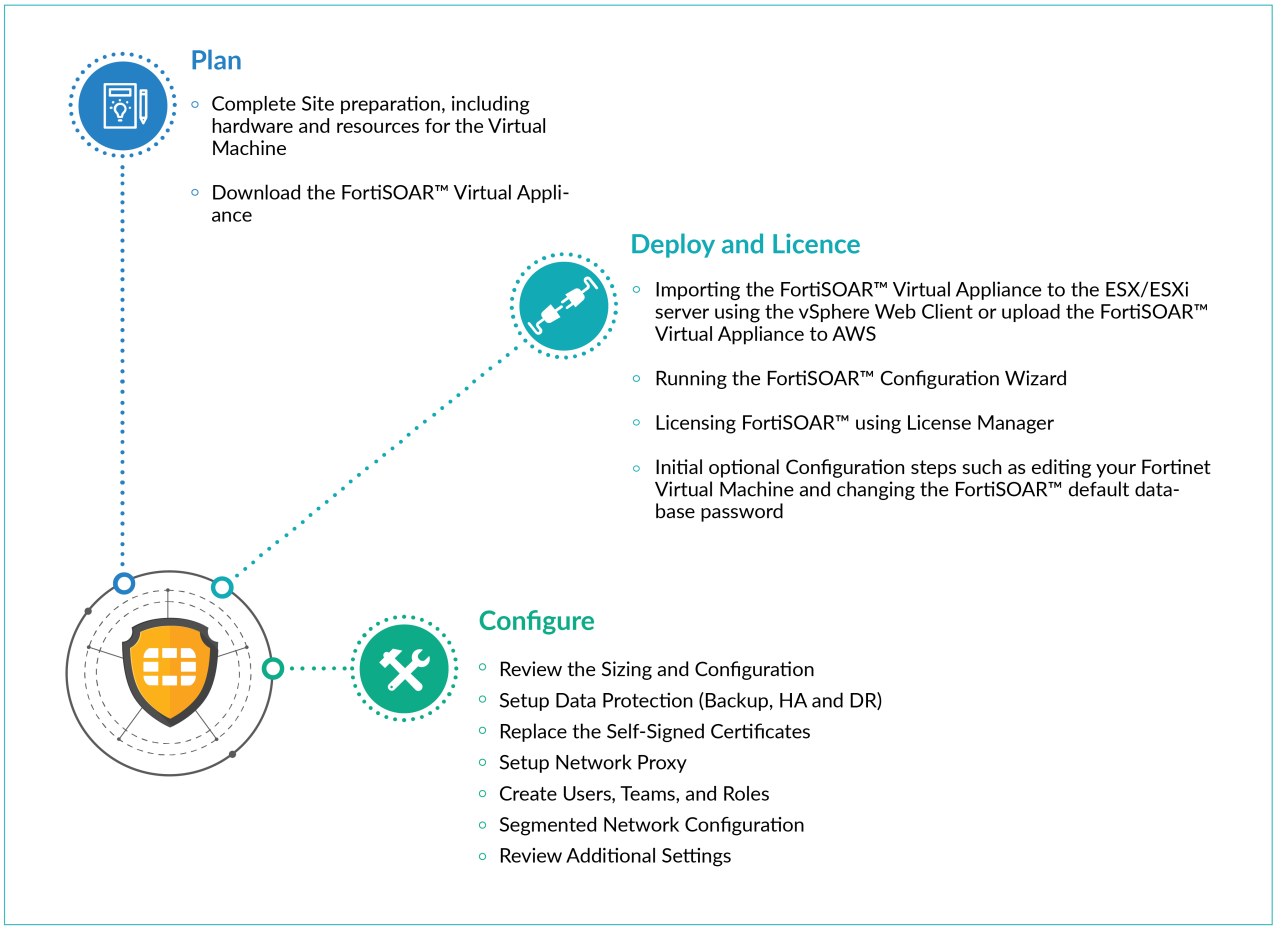
FortiSOAR 7.6.6 User Interface has been tested on the following browsers:

- Google Chrome version 146.0.7680.165
- Mozilla Firefox version 149.0
- Microsoft Edge version 146.0.3856.72
- Safari version 26.1 (20622.2.11.119.1)

Deploying FortiSOAR

This chapter covers the process of deploying FortiSOAR, and the initial configuration required for FortiSOAR. You can perform the initial configuration for FortiSOAR using the FortiSOAR Configuration Wizard.

The following image displays the high-level tasks for deploying FortiSOAR:



You can also install FortiSOAR 7.6.6 using the FortiSOAR installer (`.bin` file). For more information, see Installing FortiSOAR 7.6.6 using the FortiSOAR installer.

If you want to use the on-demand feature for encrypt all FortiSOAR's 'data at-rest', then see the Encrypting 's Data at Rest chapter in the "Best Practices Guide" before installing FortiSOAR.

You can also deploy FortiSOAR using the offline repositories for air-gapped environments. For more information, see the Deploying FortiSOAR using offline repositories chapter.

Unique Encryption Key per FortiSOAR Deployment

Starting with release 7.6.5, each FortiSOAR deployment generates a unique encryption key. This enhancement improves the security of stored credentials, database entries, and inter-service communication while maintaining backward compatibility. During the FortiSOAR Configuration Wizard, a new encryption key is automatically generated for each instance. All passwords saved thereafter are encrypted using 256-bit encryption. During upgrades, FortiSOAR generates a new unique encryption key (if one does not already exist) and uses it for all encryptions. All secrets encrypted with the old key are decrypted and re-encrypted using the new encryption key.

Notes:

- In High Availability (HA) environments, the same instance-specific key is shared across all HA nodes.
- In MSSP setups, both the Master node and each Tenant node maintain their own unique keys. Additionally, a dedicated key is generated for communication between the Master node and each Tenant node to ensure secure credential exchange. For details, see the Distributed Tenancy Support chapter in the "MSSP Guide."
- A unique, dedicated key is generated for each Agent to manage communication between the FSR Agent and the FortiSOAR node ensuring secure credential exchange and enabling encryption and decryption of data at rest. For more information, see Adding an FSR agent and Installing an FSR Agent topics.
- When using the Export and Import Wizards to transfer configurations, especially those that include credentials such as connector configurations, an export key is required. For details, see the Export and Import Wizards topic in the *Application Configuration and Customization* chapter of the "Administration Guide."

Planning

Recommended Resource Requirements

Virtual Machine (VM)

Recommended Specifications

- 12 available vCPUs
- 48 GB available RAM
- 1 TB available disk space: Recommended to have high-performance storage, preferably SSDs.
- 1 vNIC

When externalizing the FortiSOAR PostgreSQL database, ensure that the external database meets the recommended specifications for a FortiSOAR node. Customers are responsible for managing external databases, including performing regular backups and upgrading PostgreSQL to a version compatible with the installed FortiSOAR release. For more information, see the Externalization of your FortiSOAR PostgreSQL database topic in the *Setup & Advanced Configuration* chapter of the "Best Practices Guide."

Minimum Specifications

For optimal performance, use systems that meet or exceed the recommended specifications.

- 8 available vCPUs
- 32 GB available RAM
- 500 GB available disk space: Recommended to have high-performance storage, preferably SSDs.
- 1 vNIC

In case of multi-tenant configurations, contact FortiSOAR Support for sizing requirements.

The disk space that you require largely depends on your usage. audit, and workflow retention policies. So, for a more precise prediction of the database usage, you can check the current disk sizes using the following command:
`sudo csadm db --getsize`
The `sudo csadm db --getsize` command returns the sizes of the workflow logs, audit, and primary data. Based on the current usage, and your retention policies, you can extrapolate the usage for these databases. ElasticSearch disk usage would be the same as the primary database sizes. For some examples, see the 'Sizing Guide' available on https://docs.fortinet.com/product/fortisoar.

Supported Hypervisors

When installing FortiSOAR using the installation script, it is strongly recommended to use a non-hardened operating system (OS). After the installation, the OS will undergo automatic hardening by FortiSOAR. Avoid pre-hardening the OVA or performing any additional hardening steps unless directed by FortiSOAR Support, as this can cause installation failures, service startup issues, file permission errors, and other problems.
Any external monitoring agents must function correctly under FortiSOAR's current SELinux policy. If the required SELinux context is not properly configured, the customer must uninstall these agents before performing an FortiSOAR upgrade that includes an underlying OS version upgrade.

The following hypervisors are supported:

- AWS Cloud
- Fortinet-FortiCloud
- VMware ESXi versions 5.5, 6.0, 6.5, 7.0, and 8.0
- Redhat KVM
- Docker

For other virtualization or cloud environments (e.g., Google Cloud Platform, Microsoft Azure, Oracle Cloud Infrastructure [OCI], or OCI DRCC), you can install Rocky Linux 9.3/9.4/9.5/9.6 or Red Hat Enterprise Linux (RHEL) 9.3/9.4/9.5/9.6, and then install FortiSOAR using the CLI installer.
Note that release 7.6.6 has been tested with RHEL 9.6 and Rocky Linux 9.6.
For more information, see the Installing FortiSOAR 7.6.6 using the FortiSOAR installer topic.

VM Inbound Networking

Enable the following ports for the VM within your VM network:

- 22 Management (ssh)
- 443 User Interface (https)

VM Outbound Networking

For FortiSOAR to correctly interact with your network, you must provide access between the FortiSOAR VM and the third-party products and services configured within your network.

To accomplish this, enable the following ports for SSH, SMTP, and HTTPs access:

- 22 Management (ssh)
- 25 Email SMTP relay server. This port can be different based on your environment.
- 443 User Interface (https)

Depending on the type of connectors used in Playbooks, other ports might required to be opened.

Credentials

Credentials to access SSH management and the FortiSOAR User Interface are:

Username: csadmin

Password: changeme

The UI password of the 'csadmin' user for AWS is set to the "instance_id" of your instance.
To know the instance ID of your FortiSOAR AWS instance, you can SSH and run the `cloud-init query instance_id` command.

For the 'csadmin' user, the first FortiSOAR SSH login, mandates a password change, thereby enhancing the security of your csadmin account and preventing unauthorized parties from accessing the FortiSOAR administration account.

Requirements

It is highly recommended that Internet access is provided for a FortiSOAR upgrade, license deployment, and also for installing new out-of-the-box connectors.

Add the following entries in the allowlist of your Firewall or Proxy servers:

For installing or upgrading (migrating) to FortiSOAR 7.3.0 or later: `*.rockylinux.org`

For upgrading FortiSOAR, installing connectors, and accessing the widget library: https://repo.fortisoar.fortinet.com/

For connector dependencies: https://pypi.python.org

For ansible installation: https://files.pythonhosted.org

For synchronization of FortiSOAR license details: https://globalupdate.fortinet.net
Note: If SSL inspection is enabled on your proxy or firewall, ensure it is disabled for this URL to prevent connection issues.

Port Requirements

The following ports require to be locally free for various services, i.e., all these ports do not require to be open on the firewall for external access, they are used only inter-service communication within the appliance:

Service	Port Number
elasticsearch (Elasticsearch service uses this port for REST communication). Port needs to be opened in case of high availability (HA) environments.	9200
elasticsearch (Elasticsearch service uses this port for communication between nodes)	9300
rabbitmq	5671, 5672, 4369, 15672, and 25672
cyops-postman	7575
fsr-workflow	8888
postgresql-{version} Port needs to be opened in case of HA environments.	5432
cyops-auth	8443
cyops-integrations-agent	9595
cyops-tomcat (hosts cyops-gateway, cyops-notifier)	8080
cyops-api	443
Ports that need to be opened in case of HA environments	
MQ TCP traffic	5671
MQ Peer Discovery	4369
MQ inter-node	25672
Erlang distribution client ports	35672-35682 (range)
Ports that need to be opened for FortiSOAR Agent	
FSR Agent Communication Bridge for Manual Inputs or Approvals	8443

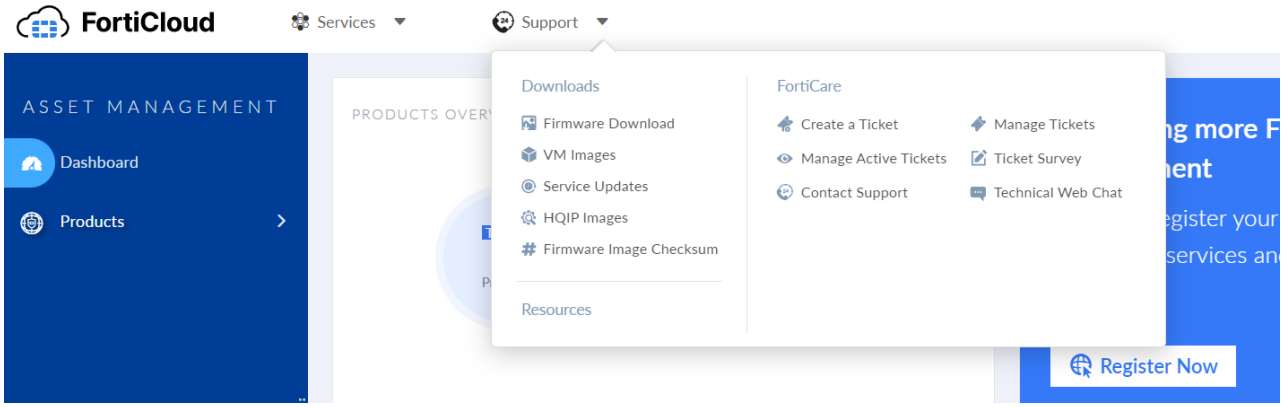
If you need to access ssh or start a terminal session from outside your network to troubleshoot or manage services or databases, then port "22/tcp open ssh" should be opened on the firewall for external access. Port 443/tcp open https must always be opened on the firewall for external access.

Downloading the FortiSOAR Virtual Appliance from the Support Portal

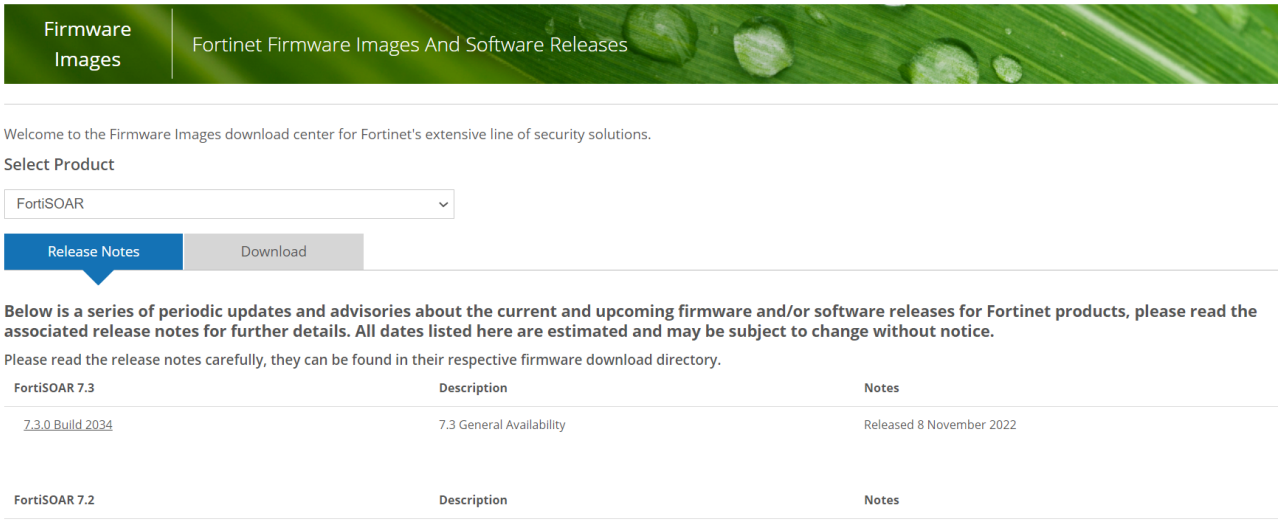
You can download the required FortiSOAR Virtual Appliance image from the support portal, which contains images for AWS, KVM-Supported QCOW2, VMware and Docker. You will need these images to be downloaded before you can begin deploying FortiSOAR using vSphere or vCenter, or KVM. In case of AWS, you can directly launch an instance using the images present on the AWS Marketplace images.

To download the FortiSOAR Virtual Appliance, do the following:

1. Log on to support.fortinet.com.
2. Click **Support** > **Firmware Download**.

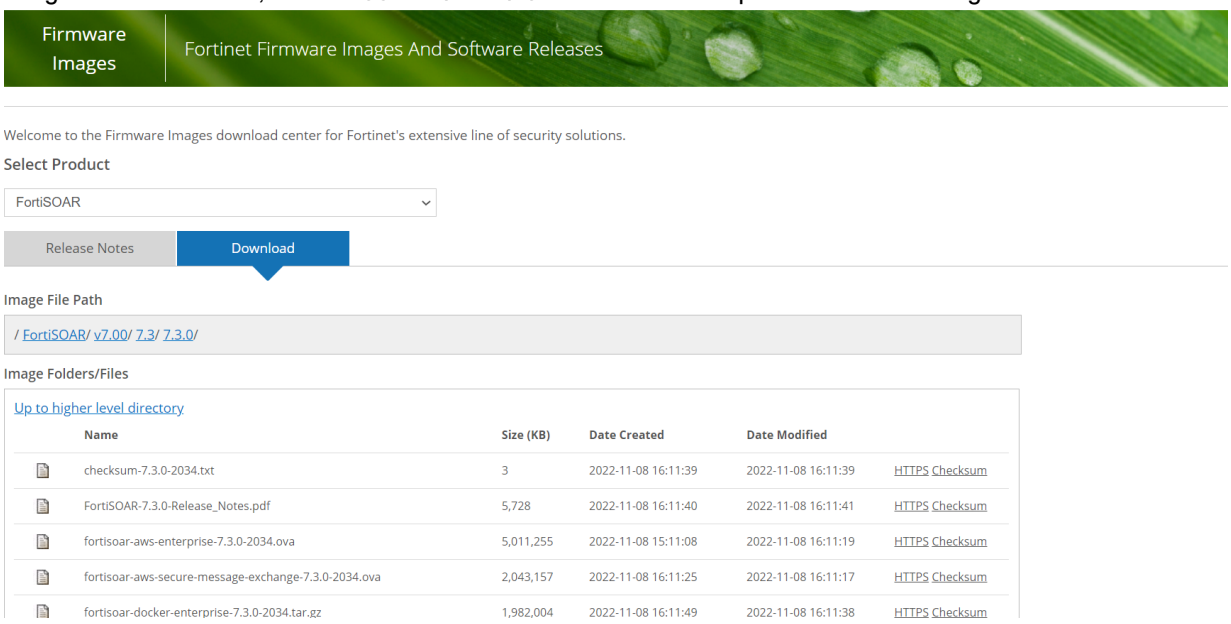


3. On the `Fortinet Firmware Images And Software Releases` page, from the **Select Product** drop-down list, select **FortiSOAR**.
   The page contains information about various released versions of FortiSOAR images, and contains two tabs: **Release Notes** and **Download**.



To view the Release Notes for a particular version, click the version and build number link, which opens the FortiSOAR Document Library, from where you can view or download the release notes for that version.

4. To download a firmware image, do the following:
   a. Click the **Download** tab.
   b. Navigate through the directory structure in the format, `<version number category>` > `<major version >` > `<minor version>`, to open the page containing the required images. For example, to download a firmware image for version 7.3.0, click **v7.00** > **7.3** > **7.3.0** and locate the required firmware's image:



   c. Download the firmware image by clicking the **HTTPS** link.
   An HTTPS connection is used to download the firmware image.
   d. Click the **Checksum** link for the image that you have downloaded.
   The image file name and checksum code are displayed in the `Get Checksum Code` dialog box.
   e. Confirm that the checksum of the downloaded image file matches the checksum provided on the download site.

Deploying the FortiSOAR Virtual Appliance

Download the required FortiSOAR Virtual Appliance image from the Support Portal, which contains images for AWS, KVM-Supported QCOW2 VMware, and Docker. You will need these images to be downloaded before you can begin deploying FortiSOAR using vSphere or vCenter, or KVM. In case of AWS, you can directly launch an instance using the images present on the AWS Marketplace images. For information on how to download images from the support portal, see the Downloading the FortiSOAR Virtual Appliance from the Support Portal section. Also, ensure that your VM is configured as per the specifications outlined in the Planning section.

Deploying the FortiSOAR Virtual Appliance using vSphere or vCenter

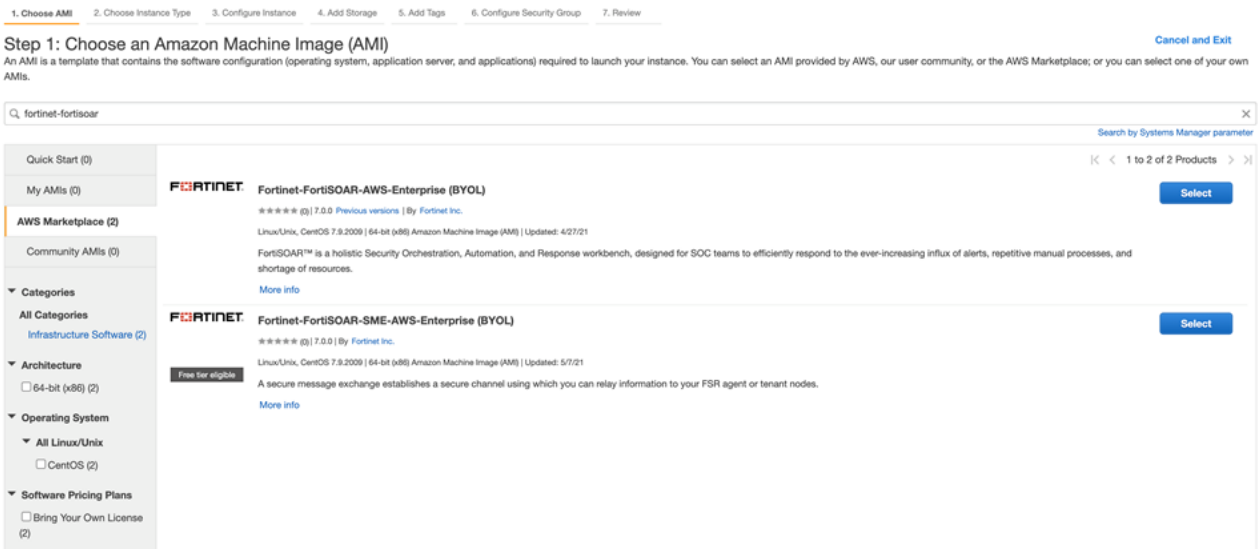
Download the VMware enterprise and secure message exchange images as outlined in the Downloading the FortiSOAR Virtual Appliance from the Support Portal section. After verifying that all system requirements are met, import and deploy the FortiSOAR Virtual Applianceto the ESX/ESXi server using vSphere or vCenter. See the VMware documentation for steps on how to deploy a Virtual Appliance.

> Once the FortiSOAR Virtual Appliance is imported and system boots up, its IP address will be displayed on the command prompt. You can share this IP address with users who need to configure FortiSOAR using the FortiSOAR Configuration Wizard.

Deploying the FortiSOAR Virtual Appliance using AWS

Once you have ensured that you have met all the specifications, perform the following steps to deploy the FortiSOAR Virtual Appliance on Amazon Web Services (AWS):

1. Log into your AWS account and from the Amazon EC2 console dashboard, choose **Launch Instance**, to launch the FortiSOAR instance.
2. On the `Choose an Amazon Machine Image (AMI)` page, enter `fortinet-fortisoar` in the search bar to find the latest version of the FortiSOAR Enterprise and SME (secure message exchange) AMIs in the AWS Marketplace. Choose the AMI and start configuring the instance. Following is a reference screenshot:



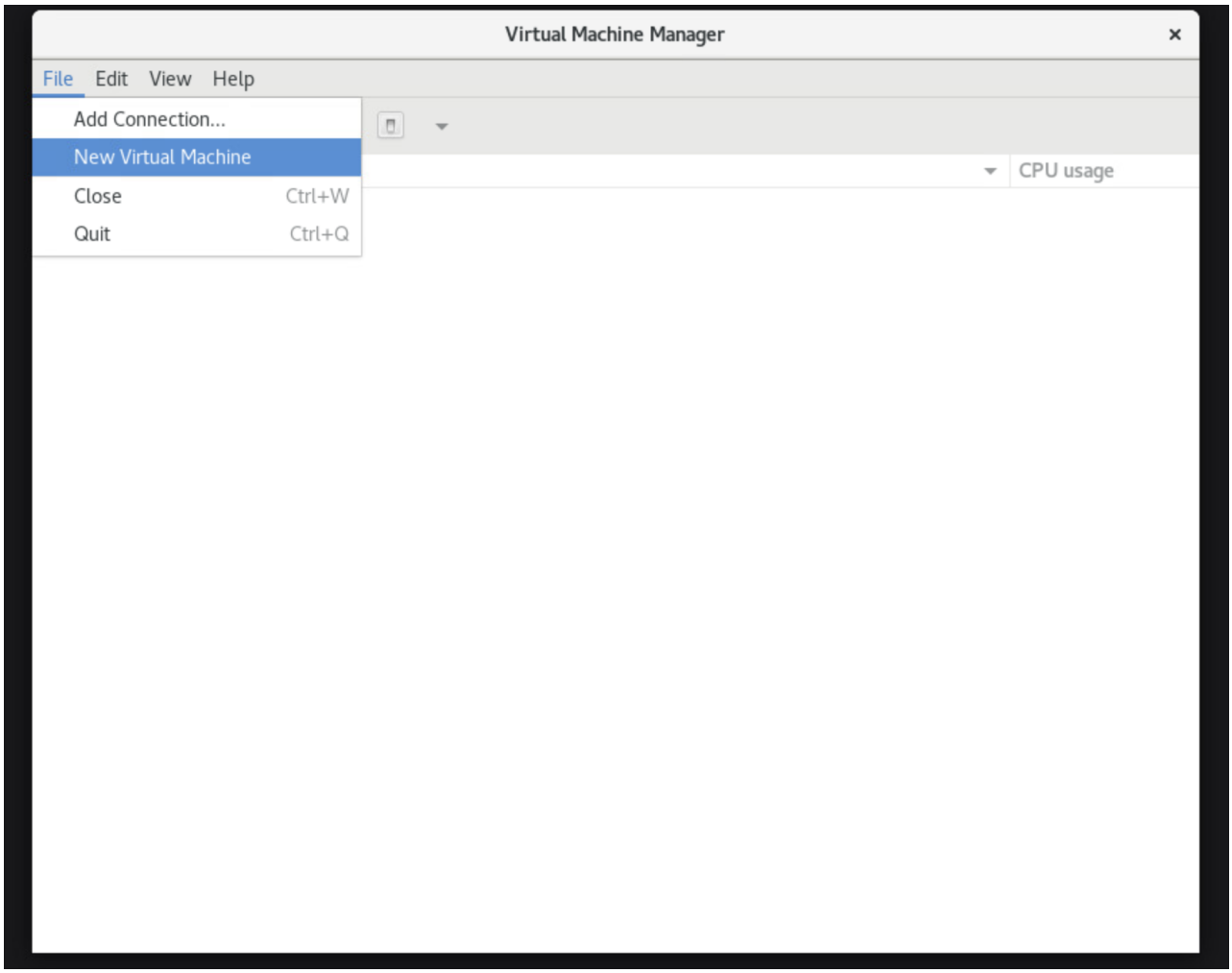
After you complete deploying your FortiSOAR and you connect the first time to your FortiSOAR VM, the EULA agreement page is displayed. You must accept the EULA to continue with your FortiSOAR configuration. If you do not accept the EULA, then the OS will halt, and you have to restart your FortiSOAR VM (power off-power on) and reconnect to the FortiSOAR VM and accept EULA to continue with your FortiSOAR configuration.

Deploying FortiSOAR using KVM

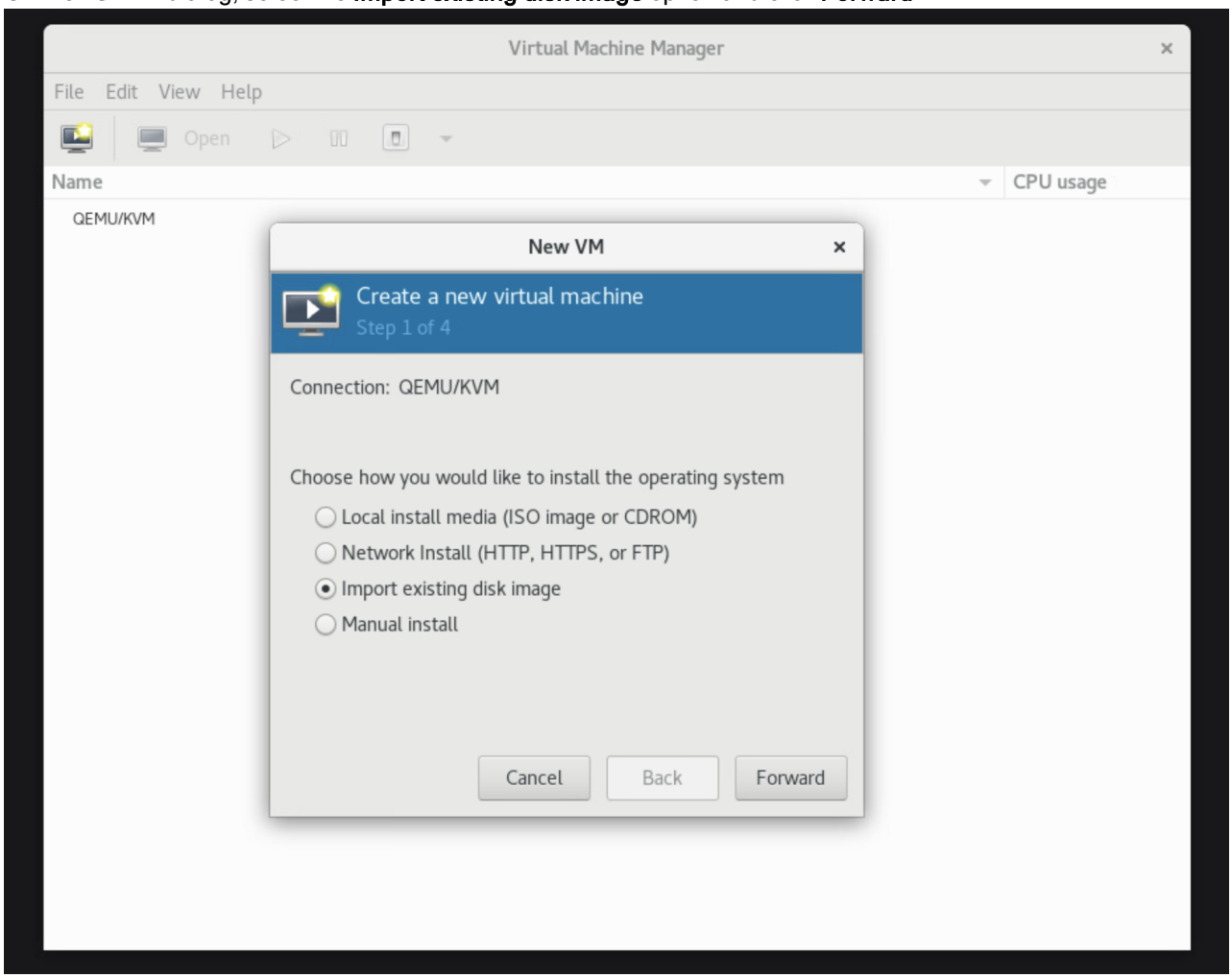
Once you have ensured that you have met all the specifications, perform the following steps to deploy the FortiSOAR QCOW2 on KVM:

1. Download the KVM-Supported QCOW2 images as outlined in the Downloading the FortiSOAR Virtual Appliance from the Support Portal section.
2. Copy the FortiSOAR QCOW2 image to the VM Image Datastore.
3. Open the Virtual Machine Manager application for KVM VM deployment.

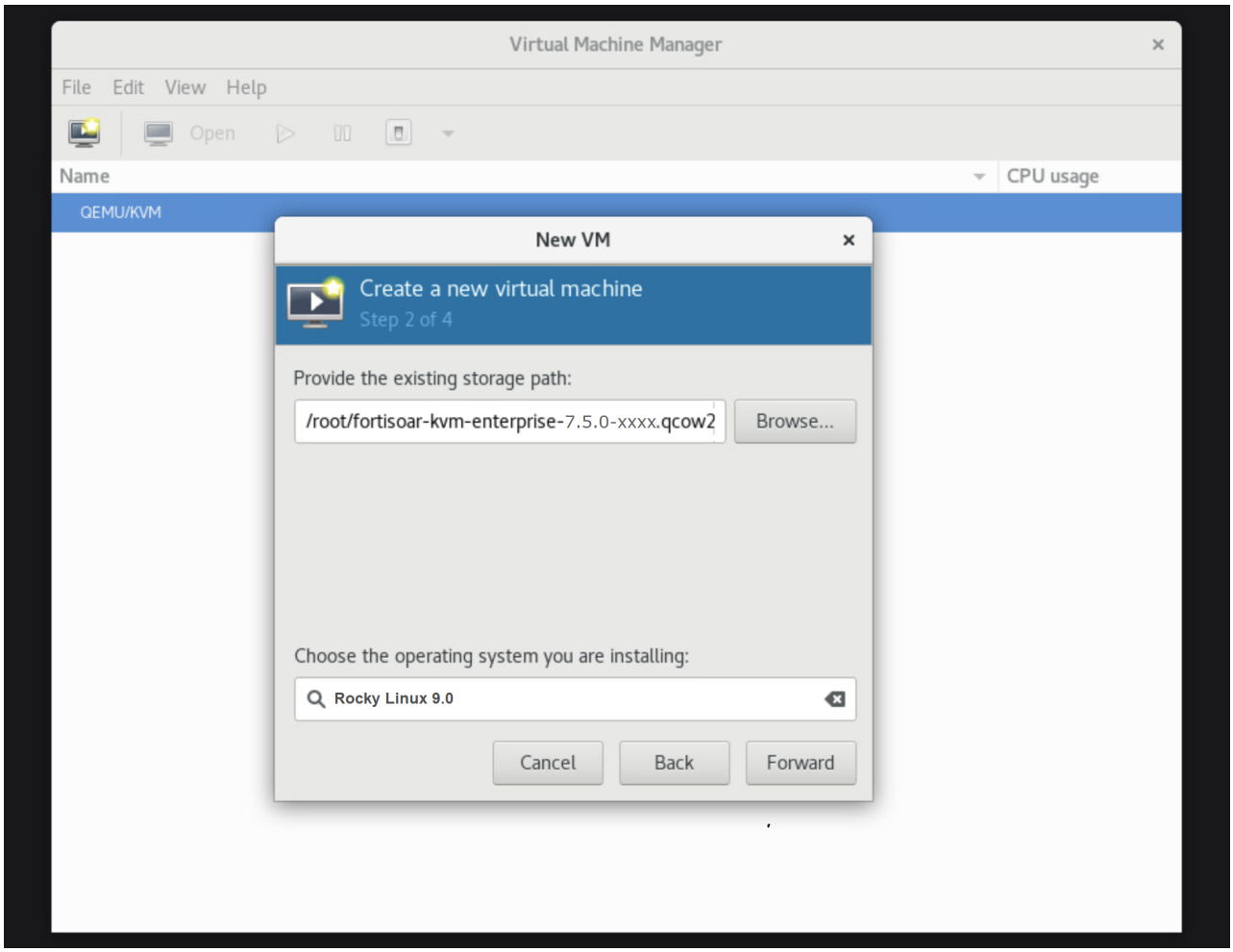
4. Click **File** > **New Virtual Machine**.



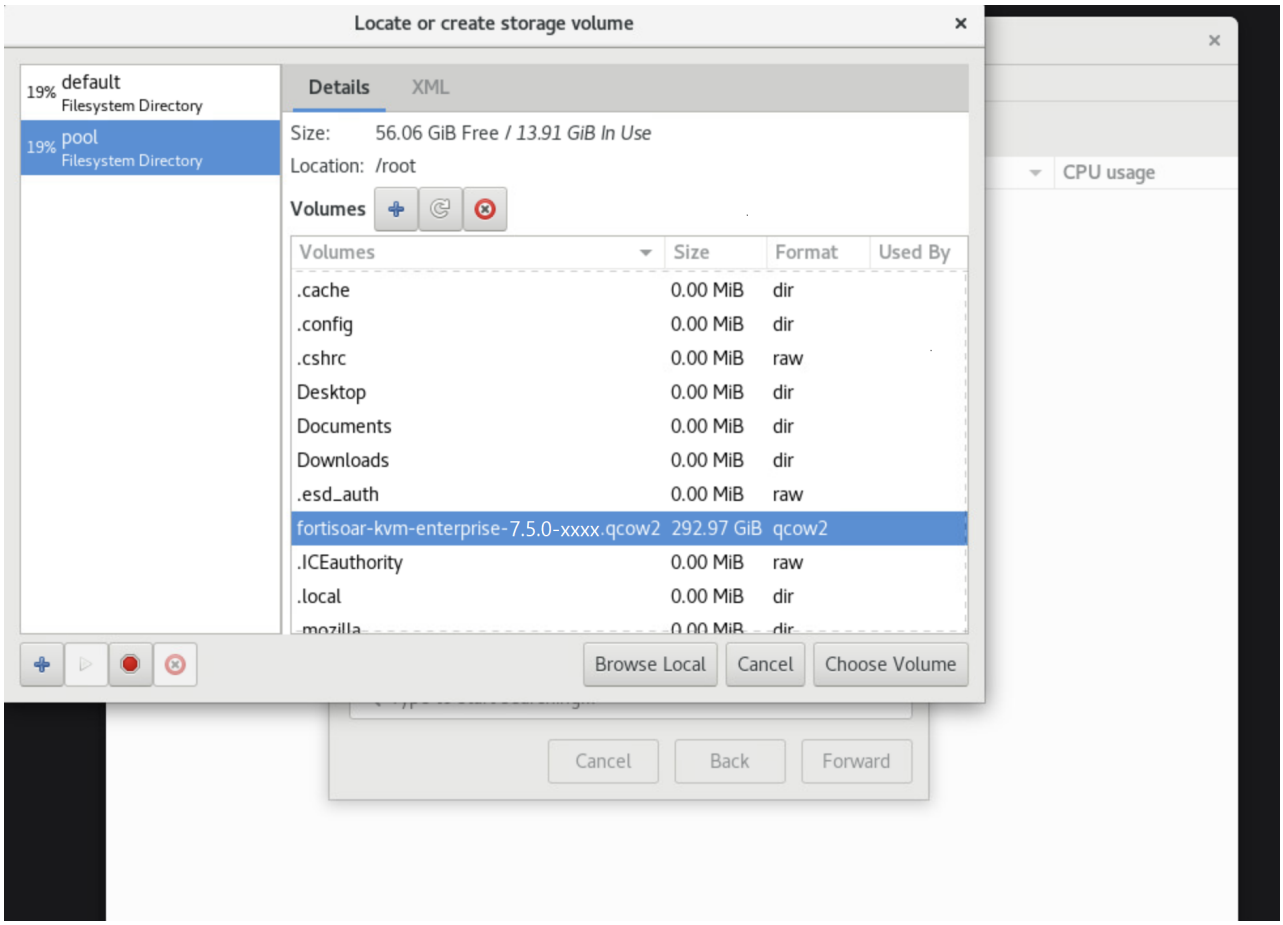
5. On the `New VM` dialog, select the **Import existing disk image** option and click **Forward**.



6. Click **Browse** and select the FortiSOAR image from the Image Datastore, select the **OS type** as **Linux** or **Rocky Linux** and **Version** as either **RHEL 9.3/9.4/9.5/9.6** or **Rocky Linux 9.3/9.4/9.5/9.6**, and click **Forward**.
Note: **Rocky Linux 9.3/9.4/9.5/9.6** will be listed in the **Choose the operating system** you are installing drop-down only if you have the newer versions of KVM. If Rocky Linux is not available, then you can choose the 'Linux' option (generic Linux) and continue with the process.

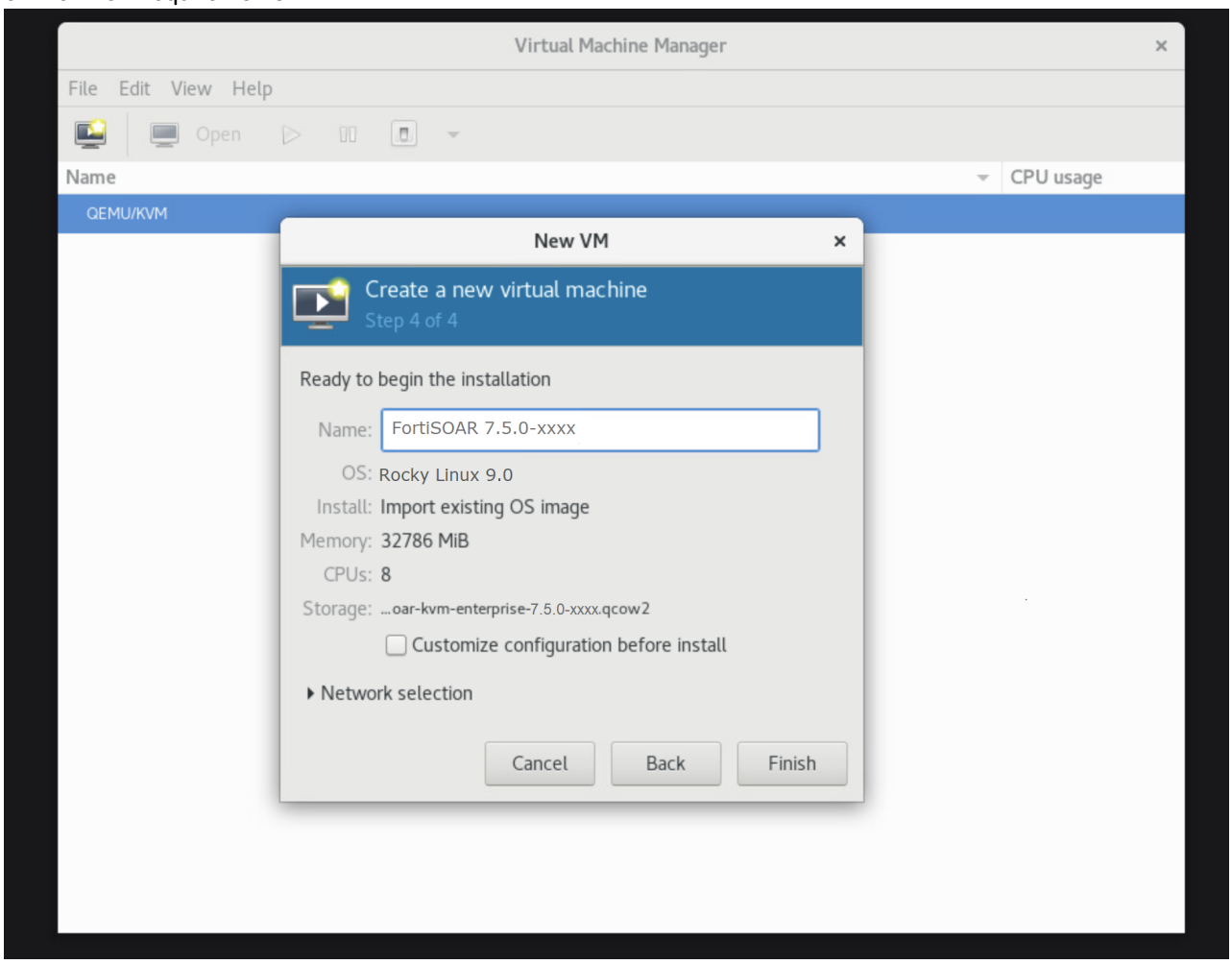


This displays the `Storage Volume` screen, where you can choose the KVM Qcowz image from the default storage volume, or you can add custom storage volume by clicking **+** on the bottom left corner to choose the KVM qcow2 image:

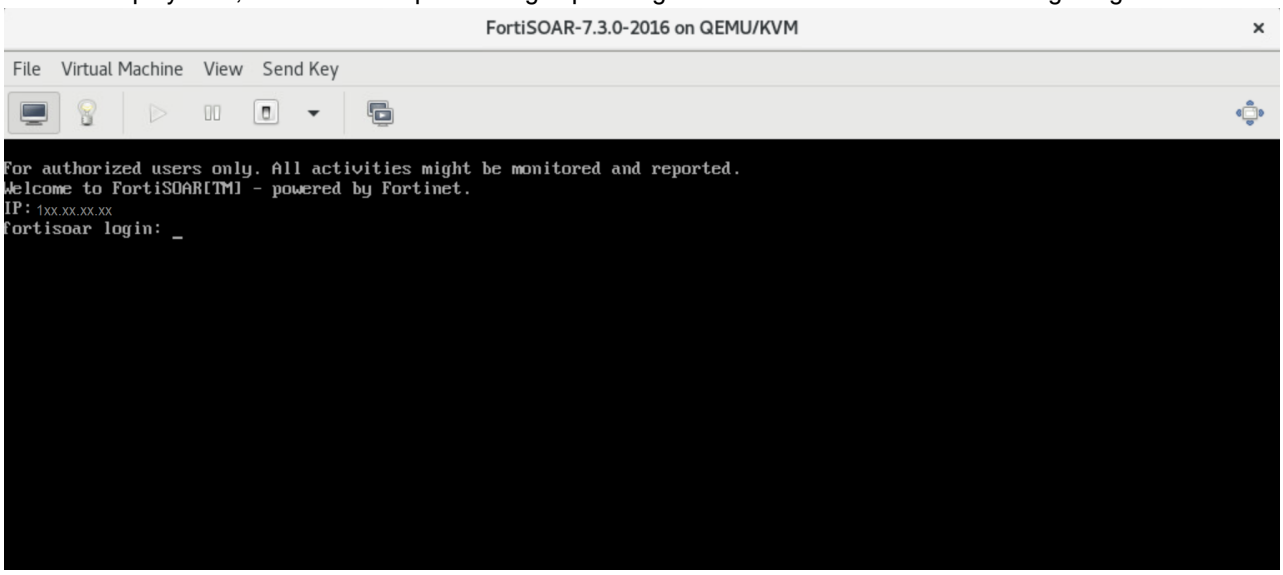


7. In the `Choose Memory and CPU settings` screen, enter `48000` (i.e. 48000 MB=48 GB) in the **Memory (RAM)** field and `12` in the **CPUs** field and click **Forward**.
8. Enter the name that you want to specify for the new virtual machine in the **Name** field, select the correct network adapter for this new virtual machine from the **Network Selection** field, and click **Finish** to complete the deployment.
Note: The network adapter type would change depending upon your KVM environment. As an example, in the following image the "Bridge adapter.." is selected. You must select the proper network adapter as per your

environment requirements.



9. Post VM deployment, the VM boots up and brings up the login console as shown in the following image:



CLI Deployment

If you want to deploy the FortiSOAR QCOW2 image on KVM using the CLI, then use the following command:
`sudo virt-install --memory 32768 --vcpus 8 --import --os-type=linux --os-variant rhel9.5 --name <virtual machine name> --disk </path/to/qcow2/image/.qcow2> --network <network type=kvm network name>`

Post deployment, perform the initial configuration steps using the FortiSOAR Configuration Wizard.

Installing FortiSOAR 7.6.6 using the FortiSOAR installer

This section provides the steps to install FortiSOAR 7.6.6 on a plain RHEL 9.3/9.4/9.5/9.6 or Rocky Linux 9.3/9.4/9.5/9.6 system that has minimal skew selected while installing the OS.

Prerequisites

Before you begin this procedure, ensure the following:

- Access to repo.fortisoar.fortinet.com. In case you have configured proxies, then ensure that you have added your proxy settings in the following files:
  - `/etc/environment`
  - `/etc/yum.conf`
  - `/etc/pip.conf`
    Following is an example of adding proxy settings to the above-mentioned files:
    - **`sudo vi /etc/environment`**
      ```
      product_yum_server=repo.fortisoar.fortinet.com
      http_proxy='http://username:password@10.1.1.0:8080'
      no_proxy='127.0.0.1','localhost'
      https_proxy='http://username:password@10.1.1.0:8080'
      ```
    - **`sudo vi /etc/yum.conf`**
      ```
      proxy=http://10.1.1.0:8080
      proxy_username=username
      proxy_password=password
      ```
      NOTE: From release 7.6.0 onwards, the proxy username and password can include special characters such as '@', '.', '$', etc.
- The FortiSOAR installer adds a 'csadmin' user with restricted access and who will not be part of "wheel" group. If you want the current user to have root access, ensure that user is added to "wheel" group. From Rocky Linux 9.x onwards, root by default is not allowed for ssh. FortiSOAR does not recommend enabling ssh access to root.

Procedure

The following procedure describes the steps that you require to run for installing FortiSOAR 7.6.6 Enterprise Edition and Secure Message Exchange:

1. Provision a Rocky Linux or RHEL version 9.3/9.4/9.5/9.6 system with the `wget` and `tmux` packages.
2. Ensure the VM has at least 500 GB of disk space. If large volumes of data are being ingested daily, it is recommended to have a disk space of 1 TB. A thin-provisioned disk is recommended.
   You can add three additional disks to your Virtual Machine (VM) and create separate Logical Volume Management (LVM) partitions for PostgreSQL, Elasticsearch, and FortiSOAR RPM data. For more information about multiple disk support, refer to the Multidisk Support article in the Fortinet Knowledge Base.
   NOTE: The database disk requires the most volume, so if you are provisioning the volume with multiple disks, ensure the data disk is adequately sized. FortiSOAR OVAs starting with version 7.5.0 use a GPT-based disk layout to support disks larger than 2 TB, compared to the MBR-based layout of earlier versions. If you already have a FortiSOAR instance and need a partition larger than 2 TB, we recommend creating a new FortiSOAR VM on release 7.5.0 or later and migrate your data using the Export and Import wizards from the old instance to the new one. This is required as FortiSOAR does not support a mix of MBR and GPT partitions.
   Refer to the table at the end of this procedure for recommended disk sizes.
3. To ensure that the upgrade is not affected if the session times out, run the `tmux` command.
   Note: Do not run `tmux` from the root shell if you have logged in as the 'csadmin' user.
4. To download the installer for FortiSOAR 7.6.6:
   ```
   wget https://repo.fortisoar.fortinet.com/7.6.6/install-fortisoar-7.6.6.bin
   ```
5. (Optional) FortiSOAR is installed with the latest version of SFSP by default. To install any other solution pack (SP), you need to export the variables for the solution pack name and version as follows:
   ```
   sudo export SP_NAME= <Name of the SP>
   sudo export SP_VERSION=<Desired version of the SP>
   ```
   For example, to install version 3.0.0 of the Continuous Delivery SP, you would export the variables as follows:
   ```
   sudo export SP_NAME= continuousDelivery
   sudo export SP_VERSION=3.0.0
   ```
6. To install FortiSOAR 7.6.6, run the following command:
   ```
   # sudo sh install-fortisoar-7.6.6.bin
   ```
   OR
   ```
   sudo chmod +x install-fortisoar-7.6.6.bin
   sudo ./install-fortisoar-7.6.6.bin
   ```
   To install FortiSOAR without SFSP, use the following command:
   ```
   # sudo sh install-fortisoar-7.6.6.bin --skip-sp
   ```
   After installation, it is recommended that you install SFSP on your FortiSOAR instance before you begin working with FortiSOAR. To install SFSP, open **Content Hub** > **Discover** and search for '`SOAR Framework`'. Click the **SOAR Framework** card, and then click **Install** on the solution pack popup.
   Once you run the `# sudo sh install-fortisoar-7.6.6.bin` command, the installer displays the following installation options:
   1. Enterprise
   2. Secure Message Exchange
   Choose **Enterprise** in this case and complete the installation.
   NOTE: If you are installing FortiSOAR in a closed or air-gapped environment, ignore connectivity messages such as "`The system does not have connectivity to https://globalupdate.fortinet.net. Connectivity....`" and proceed with the installation as there is no requirement to check the check connectivity to globalupdate.fortinet.com. For licensing information in closed environments, refer to the *FortiSOAR licensing using FortiManager* topic in the Licensing FortiSOAR chapter.
7. After installation, exit the terminal session and log back in with the default credentials:
   Username: `csadmin`
   Password: `changeme`

After entering the default credentials, you will immediately be prompted to change the default password. Only after you have changed the default password can you log into FortiSOAR.

8. Upon logging into FortiSOAR, accept the **EULA** to proceed to the FortiSOAR Configuration Wizard.
9. Retrieve your FortiSOAR Device UUID, which is created automatically during installation. This Device UUID is used to identify each unique FortiSOAR environment:
   `sudo csadm license --get-device-uuid`
   Use this Device UUID to obtain your FortiSOAR license as detailed in the Licensing FortiSOAR chapter.
10. Ensure connectivity to https://globalupdate.fortinet.net before deploying your FortiSOAR license to avoid deployment failure. This connectivity is necessary for fetching license entitlements and for product functionality post-upgrade.
    You can deploy your FortiSOAR license using the FortiSOAR UI or using the FortiSOAR Admin CLI. For more information on deploying the FortiSOAR license, see the Licensing FortiSOAR chapter.
11. Once licensed, log into the FortiSOAR UI with the default credentials:
    Username: `csadmin`
    Password: `changeme`
    After you enter the default credentials you will be prompted to change the password. Once you have specified the new password, you can log onto FortiSOAR.

Following is a table that contains the recommended disk sizes:

Mount Point	Recommended Size
/	30 GB
/boot	1.5 GB
/tmp	5 GB
/opt	20 GB
/var	10 GB
/var/lib/pgsql	1000 GB
/var/lib/elasticsearch	150 GB
/var/lib/rabbitmq	11.5 GB
/var/log	10 GB
/var/log/audit	1 GB
/var/log/cyops/coredump	49 GB
/home	10 GB

Installing the secure message exchange

A secure message exchange establishes a secure channel that is used to relay information to the agents or tenant nodes. To create a dedicated secure channel, you are required to add the reference of the installed and configured secure message exchange, when you add agent or tenant nodes to your environment. A default "self" secure message exchange configuration is available and can be enabled. However, for a production setup, we recommend that you configure a separate secure message exchange. For more information, see the Enabling the secure message exchange and Adding a secure message exchange sections.

Steps for installing the secure message exchange are as follows:

1. Provision a Rocky Linux or RHEL version 9.3/9.4/9.5/9.6 system with the `wget` and `tmux` packages.
2. Ensure your VM has disk space of minimum 50 GB, recommended is 100 GB. It is recommended to have a thin provisioned disk.
3. To ensure that the upgrade is not affected if the session times out, run the `tmux` command.
4. To download the installer for FortiSOAR 7.6.6 secure message exchange:
   ```
   wget https://repo.fortisoar.fortinet.com/7.6.6/install-fortisoar-7.6.6.bin
   ```
5. To install FortiSOAR 7.6.6 secure message exchange run the following command:
   ```
   # sudo sh install-fortisoar-7.6.6.bin
   ```
   OR
   ```
   sudo chmod +x install-fortisoar-7.6.6.bin
   sudo ./install-fortisoar-7.6.6.bin
   ```
   The installer will display the following installation options:
   1. Enterprise
   2. Secure Message Exchange
   Choose **Secure Message Exchange** in this case and complete the installation.

Once you have installed the secure message exchange, a FortiSOAR Secure Message Exchange Configuration Wizard similar to the FortiSOAR Configuration Wizard is automatically run on the first `ssh` login by the `csadmin` user and it performs the initial configuration steps that are required for the Secure Message Exchange. In the FortiSOAR Secure Message Exchange Configuration Wizard you require to provide certain inputs such as the hostname of the Secure Message Exchange VM, port numbers to be used for the API and TCP connections to the Secure Message Exchange, etc., which are required to complete the initial configuration steps for the Secure Message Exchange.

Installing FortiSOAR on RHEL using the FortiSOAR installer

This section describes the steps that you are required to follow to install FortiSOAR on a Red Hat Enterprise Linux (RHEL) system.

1. Register your RHEL instance:
   ```
   sudo subscription-manager register --username <username> --password <password>--auto-attach
   --force
   ```
2. Check if simple content access is enabled using the following command:
   ```
   sudo subscription-manager status
   ```

   Output if simple content access is enabled:
   ```
   +-------------------------------------------+
   System Status Details
   +-------------------------------------------+
   Overall Status: Disabled
   The Content Access Mode is set to Simple Content Access. This host has access to content,
   regardless of subscription status.
   System Purpose Status: Disabled
   ```

   Output if simple content access is disabled:
   ```
   +-------------------------------------------+
   System Status Details
   ```

```
+----------------------------------------+
Overall Status: Current
System Purpose Status: Matched
```

If the simple content access is enabled, you can skip to step 3.
If the simple content access is disabled, then enable the optional repositories:
`sudo subscription-manager repos --enable=rhel-9-for-x86_64-baseos-rpms --enable=rhel-9-for-x86_64-appstream-rpms --enable=codeready-builder-for-rhel-9-x86_64-rpms`

3. Install the required packages:
   `sudo yum install wget tmux vim -y`
4. Follow the steps outlined in the Installing FortiSOAR 7.6.6 using the FortiSOAR installer topic to install FortiSOAR on your RHEL instance.

FortiSOAR Configuration Wizard

A configuration wizard runs automatically on the first `ssh` login by the `csadmin` user and performs the initial configuration steps that are required for FortiSOAR.

> After deploying FortiSOAR 7.6.3 or later, users running `sudo` commands as the 'csadmin' user will be prompted for a password on all systems except AWS.
> After deploying FortiSOAR 7.6.5 or later, the `csadmin` user's `sudo` privileges are restricted to only the commands required to work with FortiSOAR, instead of providing full 'root' access. This enhancement aligns with the *principle of least privilege* and reduces exposure to sensitive system files. Therefore, commands such as `yum`, `systemctl`, `csadm`, etc, must be prefixed with `sudo`, for example, `sudo csadm --help`. To open or edit a file, prefix the command with 'sudo' and specify the file's full path (`sudo vi <full path of file>`)
> The configuration wizard allows you to reset the `root` user password as its first step. If the `root` password reset step is missed, the password must be reset later from the VM console. Refer to the Red Hat or Rocky Linux documentation for instructions on resetting the `root` password.
> **Note**: After login, the 'csadm' user is assigned limited sudo privileges. For security reasons, 'root' access is provided via the system console and is not available over SSH.

The wizard guides you through the configuration process with appropriate instructions so that you can efficiently perform the initial configuration required for FortiSOAR. To begin running the configuration wizard, you must accept the Fortinet End User License Agreement.

The wizard performs the following configuration steps:

1. Reset the 'root' user password. At this prompt, enter a new root password and then re-enter it to confirm. If both entries match, the password is successfully reset. If they do not match, or if you cancel the process, the prompt will reappear.
   Note: After login, the '`csadm`' user has limited `sudo` privileges. The **root** account can be accessed only from the system console, not via SSH.
2. Change hostname and refresh the MQ node name: (Optional) You can change the hostname for your FortiSOAR VM and refresh the node name of your MQ. Ensure that the hostname that you provide is resolvable. If the hostname gets resolved by a DNS server, ensure that you provide ***only the node name and not the complete FQDN***. The wizard checks if the hostname is valid or not; and throws an error in case of an invalid hostname. FortiSOAR optionally also asks for additional DNS servers.
   In an environment where DHCP is not enabled, i.e., in case of a static IP environment, the configuration wizard fails

since it cannot get the network. Therefore, the configuration wizard detects whether the FortiSOAR VM has an IP; if the wizard cannot detect an IP, a `Static IP` page is displayed where you can add the Static IP, Gateway, Netmask, DNS1 and DNS2, ensuring that the configuration wizard completes without failure. This also provides users with the flexibility of changing their environment from DHCP to Static without worrying that the configuration wizard will fail since it cannot get the network.
If the configuration wizard detects an IP, i.e. in case of a DHCP enabled system, the `DNS servers` input page is displayed.
Also, if you change the hostname, the configuration wizard automatically updates the `HOSTNAME` variable in `/etc/profile`.
NOTE: If you skip the 'Set Hostname' step in the FortiSOAR Configuration VM, the default hostname entry will be automatically added to the `/etc/hosts` file.

3. Get DNS: Gets the DNS for your FortiSOAR system
4. Update network configuration: This is an automatic process.
5. Set up intra-service authentication: This is an automatic process to generate new appliance keys unique to your instance for communication to the FortiSOAR services.
6. Generate certificates: This is an automatic process; you do not require to provide any inputs.
7. Generate Device UUID: This is an automatic process; you do not require to provide any inputs. The wizard also saves the Device UUID in the `/home/csadmin/device_uuid` file.
   This Device UUID is required for when you are generating your license for FortiSOAR. For more information, see the Licensing FortiSOAR chapter.
   Important: You get logged out after the FortiSOAR VM is configured, so that the changes can take effect. Therefore, you require to `ssh` again to the FortiSOAR VM.
8. Reset database passwords: This is an automatic process to reset database password to a new password unique to your instance.
9. Restart services: This is an automatic process to reset all FortiSOAR services.
10. Configure default single-node HA cluster: This is an automatic process that creates the default single-node HA cluster. This FortiSOAR server is created as a primary-active node.
11. Configure proxy: (Optional) You can configure a proxy server to serve all http/https requests from FortiSOAR. There is a single **Proxy** field used to configure both http/https proxies. To configure a proxy, you must specify the username and password, and the hostname and the port number of the proxy server. For example to configure aproxy, enter the proxy details in the following format: `http://user:password@[ip/fqdn]:port`. Once the proxy value is validated, the same is added for both http and https proxies.
    You can also configure a comma-separated list of hostnames that do not require to be routed through a proxy server. For example, `[ip1/fqdn1], [ip2/fqdn2]`.
    You can also set the proxy at a later time using the `sudo csadm network` command.
    NOTE: From release 7.6.0 onwards, the proxy username and password can include special characters such as '@', '.', '$', etc.
12. Install python libraries: This is an automatic process to install some python libraries required by FortiSOAR.
13. Install default widgets: This is an automatic process to install some default widgets as part of FortiSOAR.
14. Search index initialization: This is an automatic process.
15. Refresh SSH keys: This is an automatic process that refreshes the SSH key.
16. Generate default encryption keys: This is an automatic process that generates the default encryption keys.
17. Refresh data identifiers: This is an automatic process.
18. Post-configuration tasks: This is an automatic process.

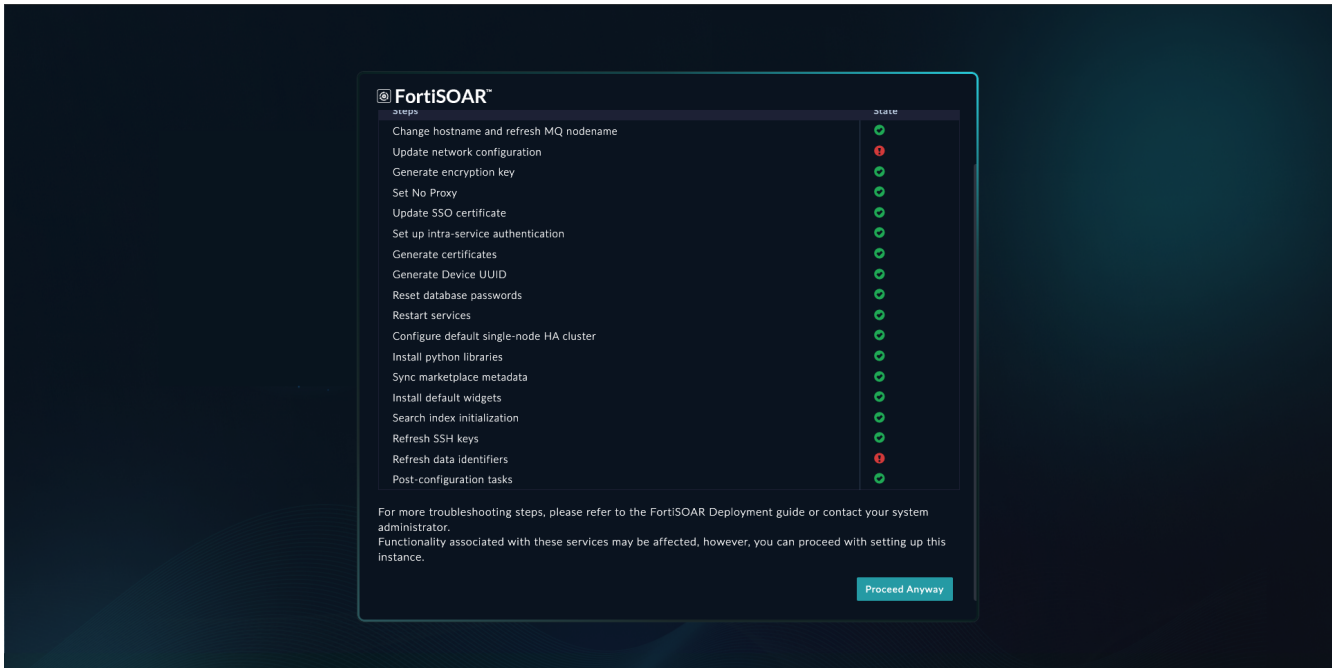
After the FortiSOAR Configuration Wizard is run, it displays the following:

- Device UUID
- Path where the Device UUID is saved
- Path of the Configuration Wizard log

If your FortiSOAR Configuration Wizard displays errors when it is being run, then you can troubleshoot the FortiSOAR Configuration Wizard errors by checking its logs, which are located at `/var/log/cyops/install/config_vm_<timestamp>`. For example, `sudo vi /var/log/cyops/install/config-vm-09_Nov_2018_05h_37m_36s.log`.

You should add CA signed certificates once configuration of FortiSOAR is completed. For details, see the `Updating the SSL certificates` topic in the Additional Configurations chapter.

If the FortiSOAR Configuration Wizard fails when provisioning your instance, then a failure screen detailing the status of each configuration step is presented on the FortiSOAR UI, making it simpler to identify the issue. Before using FortiSOAR, you must use the CLI to fix any issues with the failed steps as their functioning might be hampered. However, if you choose to access FortiSOAR without rectifying the failed steps, which can cause FortiSOAR functionality to deteriorate, a **Proceed Anyway** button is provided that lets you to use the product while acknowledging the configuration failure:



If the FortiSOAR UI is not accessible even after clicking **Proceed Anyway**, you can try the following steps to fix the issues:

- Use the `sudo csadm services --status` command to check the statuses of the services. Based on the output of this command, you can choose to restart the specific service that is not running; for example, if the 'postgresql-16' service is not running you can restart it using the `sudo systemctl restart postgresql-16` command. Alternatively, you can use the `sudo csadm services --restart` command to restart all the services. Use the `sudo csadm services --status` command once you have restarted any non-running services to verify their status.
  Re-run the configuration using the `sudo /opt/cyops/scripts/config-vm.sh` command if all the services are operational.
- Manually install ansible in the case of an ansible installation error using the following command:
  `sudo -u nginx /opt/cyops-workflow/.env/bin/pip install ansible==7.4.0 --extra-index-url https://repo.fortisoar.fortinet.com/prod/connectors/deps/simple/`

- If the failure screen keeps getting displayed on the FortiSOAR UI, even after you have attempted to resolve all the backend issues, then you can update the `fsr-boot.json` to update its state from '`failed`' to '`config_vm_failure_acknowledged`'.

Contact support if failures persist even after troubleshooting.

Creating a backup user for the FortiSOAR appliance to allow access to the CLI

It is highly recommended that you set up a backup user for the FortiSOAR appliance so that, in the event you forget the '`csadmin`' CLI password for CLI access and your `csadmin` user gets locked you can still access the CLI using the backup user's account.
Note: These steps are not applicable to FortiSOAR Cloud.
To create a backup user, follow these steps:

1. Login as 'root' from VM Console using root credentials.
2. Run the following command to create a backup user, `csadmin-bkp`:
   `sudo adduser csadmin-bkp`
3. Specify the password for the new backup user, `csadmin-bkp`:
   `sudo passwd csadmin-bkp`
   NOTE: Ensure that you note down this password.
4. Run the following command to add the new backup user to the wheel group:
   `sudo usermod -aG wheel csadmin-bkp`

Pointing the chronyd service to a valid ntp server

If the time on the FortiSOAR server is not correct, you might see issues such as ingestion workflows not pulling the latest data from an external source. It is highly recommended to keep the time in sync with an NTP server. Therefore, if you require to change the system time on your FortiSOAR instance, then perform this step immediately after running the FortiSOAR Configuration Wizard.

The `chronyd` service runs on your FortiSOAR instance, and it requires to be pointed to a valid ntp server. If the `/etc/chrony.conf` file contains entries to ntp server(s) that are not valid; then you might face **Invalid System Time** issues where you might not be able to log on to your FortiSOAR instance. Edit the file: `sudo vi /etc/chrony.conf` to add details of a valid ntp server(s). For a list of common NTP servers, go to https://www.ntppool.org/en/.

In case your FortiSOAR VM does not have access to the internet, then you must edit the file: `sudo vi /etc/chrony.conf` to add details of a valid ntp server within your data center.

Editing the VM configuration

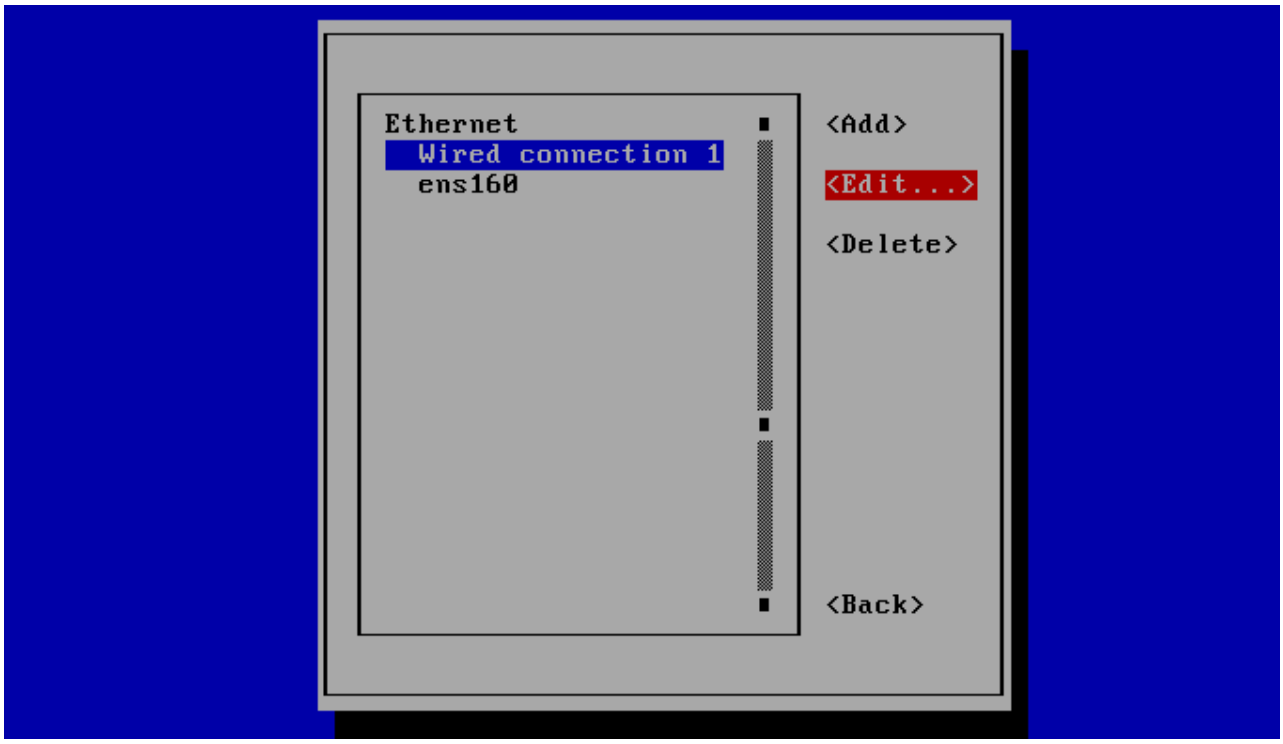
It is not necessary to perform the following steps, but they can quickly assist you to get access to the FortiSOAR VM:

1. Setting a static IP
2. Determining your DHCP IP Address

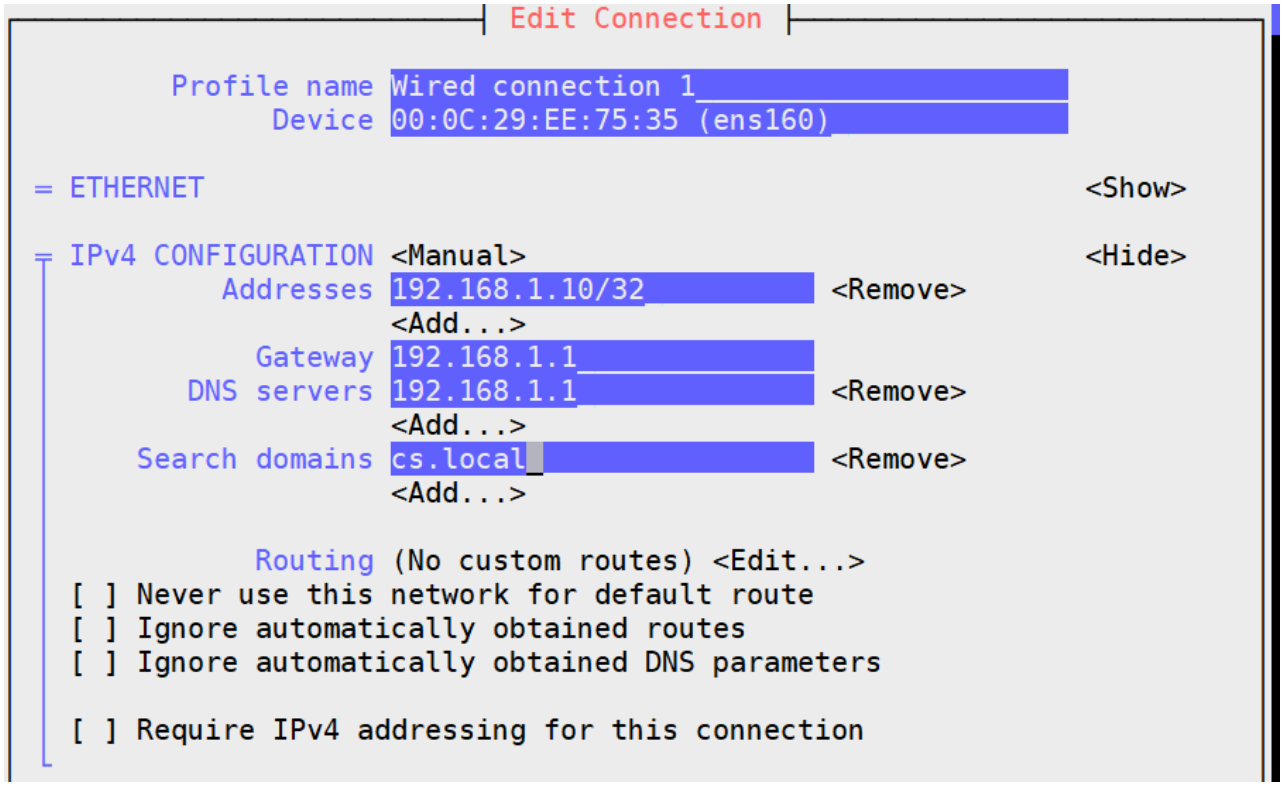
Setting a static IP

1. On the ESX console for your FortiSOAR VM, login to the VM as the `csadmin` user.
2. Type `sudo nmtui` and press `Enter`.
3. On the first screen, select the **Edit a connection** option and press `Enter`.

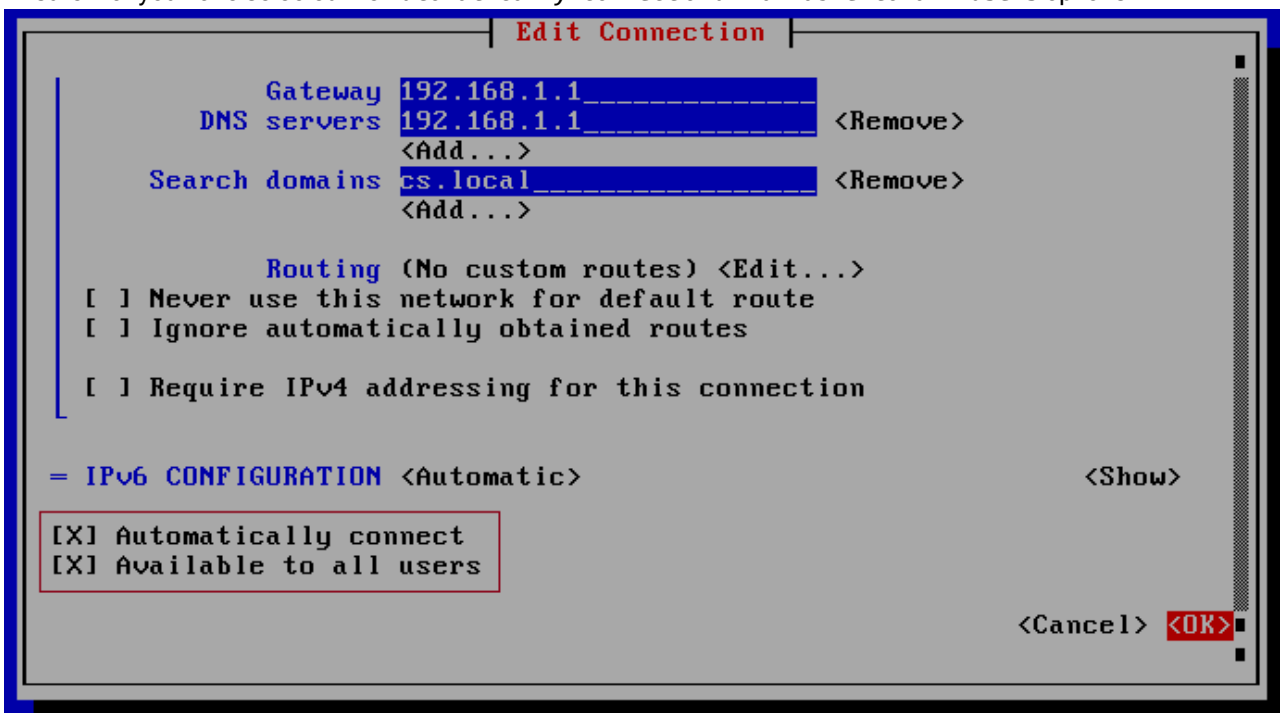
4. On the second screen, select the connection listed under Ethernet, which is **Wired connection 1** and select `<Edit...>`.



5. Use the arrow keys to select `<Show>` that appears to the right of the IPv4 Configuration option and press `Enter`.
6. Enter the required information for your network. You must enter all the information, such as IP Address, Gateway, DNS servers address, on this screen:



7. (Optional) If you want to configure IPv6, repeat steps 5 and 6 and then enter the required information, such as IPv6 Address, for your network.
8. Ensure that you have selected the `Automatically connect` and `Available to all users` options.

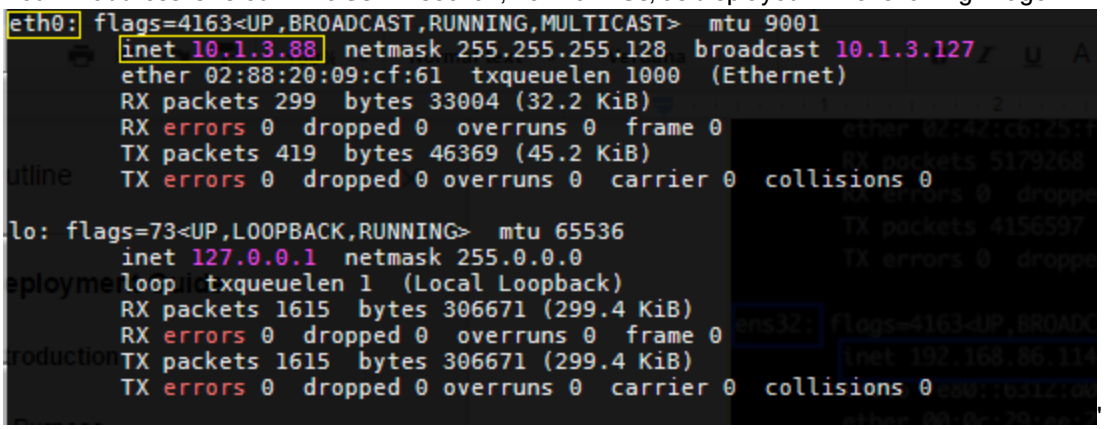


9. Select `<OK>` and press `Enter`.
10. Select `<Back>` and press `Enter`.
11. Select `<OK>` and press `Enter`.
12. Restart the network service using the `sudo systemctl restart network` command.

Once the network service restarts, you can use the assigned static IP.

Determining your DHCP IP address

1. On the ESX console for your FortiSOAR VM, login to the FortiSOAR VM.
2. Type `ifconfig | more` in the terminal and press `Enter`.
   Your IP address is listed in the `eth**` section, next to `inet`, as displayed in the following image:



Once you have completed configuring the hostname and IP address ensure that the default inbound ports mentioned in the 'VM Inbound Networking' section are open and accessible.

Now you must follow the licensing process required for FortiSOAR and then you can use this IP address to log on to the FortiSOAR UI and begin the configuring the system. See the Licensing FortiSOAR chapter for more information.

Deploying FSR Agents

FortiSOAR supports segmented networks, which facilitates investigation in a multi-segmented network by allowing secure remote execution of connector actions. If your requirement is to be able to remotely run connector actions, then you can use the "FSR Agent".

Automated ingestion, enrichment, or triage actions using a SOAR platform require network connectivity to various endpoints on which you want to run connector actions. These devices or endpoints, can at times, be in a different network segment than the one where the FortiSOAR node is deployed. To connect to such endpoints in segmented networks, FortiSOAR provides a lightweight component, called the "FSR Agent". A FSR Agent can be deployed in a network segment and configured to receive and execute connector actions from a FortiSOAR node using its secure message exchange. The FSR Agent only needs an outbound network connectivity to the secure message exchange server on its TCP port. It does not need a VPN setup or an inbound network connectivity.

You do not require any additional licensing for the FortiSOAR secure message exchange.

You can configure the following types of authentications to connect FSR agent and secure message exchange:

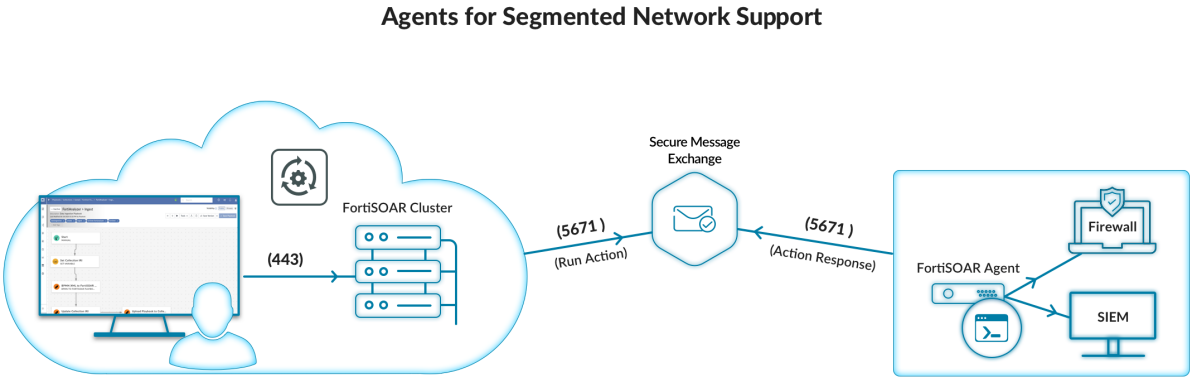
- **Basic Authentication**: Uses username and password to create connections between FSR agent and secure message exchange.
- **Basic Authentication with Peer Verification**: Uses username and password to create connections between FSR agent and secure message exchange, and also performs 'Certificate Verification'. This process will verify that the clients which are attempting to connect can be trusted by presenting a certificate that is signed by a CA and trusted by the server; thereby ensuring that only trusted clients can connect to the secure message exchange.
- **Client Certificate Authentication**: Presents a certificate to the server which is signed by a trusted CA. It is recommended that you create the certificate with the common name as the name of your agent or tenant. This provides enhanced security as this gives the facility to connect only to trusted clients.

To enable client certificate authentication, you can specify the authentication type as '**Certificate Auth**' while adding an FSR agent.

To enforce client certificate verification, you must provide a pair of exchange event listener client certificates and exchange event listener client key when you are adding a secure message exchange. Client verification ensures that whenever any client wants to connect to secure message exchange that client must present the client certificate to the secure message exchange for verification. You must also provide the pair of exchange event listener client certificates and exchange event listener client key, if you have enabled mutual TLS (mTLS). Use the `sudo csadm mq mtls` command to enable or disable mTLS. For more information on `csadm`, see the CLI Administration chapter in the "Administration Guide."

Starting with release 7.6.4, you can deploy an FSR Agent on a Docker platform, for details see the Deploying FSR Agent on a Docker Platform chapter.

Architecture



Recommended Resource Requirements for Virtual Machines (VM)

Recommended specifications for Secure Message Exchange

- 8 available vCPUs
- 16 GB available RAM
- 100 GB available disk space: Recommended to have high-performance storage, preferably SSDs.
- 1 vNIC

Recommended specifications for FSR agents

- 4 GB available RAM
- 2 available vCPUs
- 10 GB available disk space
- Rocky Linux 9.3/9.4/9.5/9.6 or Red Hat Enterprise Linux (RHEL) Server 9.3/9.4/9.5/9.6.
  NOTE: Release 7.6.6 has been tested with RHEL 9.5 and Rocky Linux 9.5.

Prerequisites for installing an FSR agent

- Ensure that the VM on which you want to install the FSR agent matches the recommended specifications, see Recommended specifications for FSR agents.
- Ensure that repo.fortisoar.fortinet.com is reachable or resolvable from the VM where you intend to install the FSR agent. Additionally, if you plan to create custom connectors using packages not available on the FortiSOAR repo server, ensure that pypi.org is reachable or resolvable from the VM where you intend to install the FSR agent.
- Ensure that the secure message exchange that you have specified when you have added the FSR agent is reachable or resolvable from the VM.
- The following prerequisites apply only to FSR Agents deployed on Virtual Machines not Docker platforms:
  - Ensure that the Rocky Linux or RHEL repository is enabled on the other system in the air-gapped environment for smooth installation your FSR agent. See Deploying FortiSOAR using offline repositories chapter for details on offline repo.
  - Ensure you review the `Installing an FSR agent using an offline repo, where the certificate on your offline repo is self-trusted topic in the where the certificate on your offline repo is self-trusted` in the Deploying FortiSOAR using offline repositories chapter, if you are installing an FSR agent using an offline repo.
  - Ensure that the following packages are installed on the instance where you are going to install the FSR agent:
    - **Python39-devel**: The FSR agent runtime needs "python39-devel". During FSR agent installation, the installer looks for an existing installation, and in the case, it is not installed, tries to install it using `sudo yum install`. If this package is not found, the FSR agent installation will fail.
      Note: If the connector has dependency on any other packages that are not pre-installed then such packages or applications will require to be separately installed on the FSR agent. For example, the 'nmap' connector requires the `nmap` application to be installed, therefore you would require to install this application separately on the agent.

Process of setting up an FSR agent

1. Add a Secure Message Exchange (SME).
   An SME establishes a secure channel to relay information to agents or tenant nodes. You can use an externally deployed SME, the default (embedded) SME, or both.
   If you want to use the local i.e., Default (Embedded) secure message exchange to connect to external FSR agents and run remote actions on various segments of your network or in case of a dedicated tenant, then you must enable the secure message exchange as described in the Enabling the secure message exchange section. In case of an external secure message exchange, then add the secure message exchange as described in the Adding a Secure Message Exchange section.
2. Add FSR agents to your FortiSOAR instance. See the Adding an FSR agent section.
3. Install FSR agents. See the Installing an FSR agent section.

Minimal permissions required

To configure and install FSR agents:

- `Create`, `Read`, and `Update` permissions on `Agents`.
- `Read` permissions on `Application` and `Secure Message Exchange`.

Enabling the secure message exchange

A secure message exchange establishes a secure channel using which you can relay information to your FSR agent or tenant nodes. A Default (Embedded) secure message exchange configuration is available on every FortiSOAR node that can be enabled as explained in this section.

To use the Default (Embedded) secure message exchange to connect to external FSR agents, you must enable the secure message exchange using the `sudo csadm secure-message-exchange enable` command. For more information on `csadm`, see the CLI Administration chapter in the "Administration Guide."

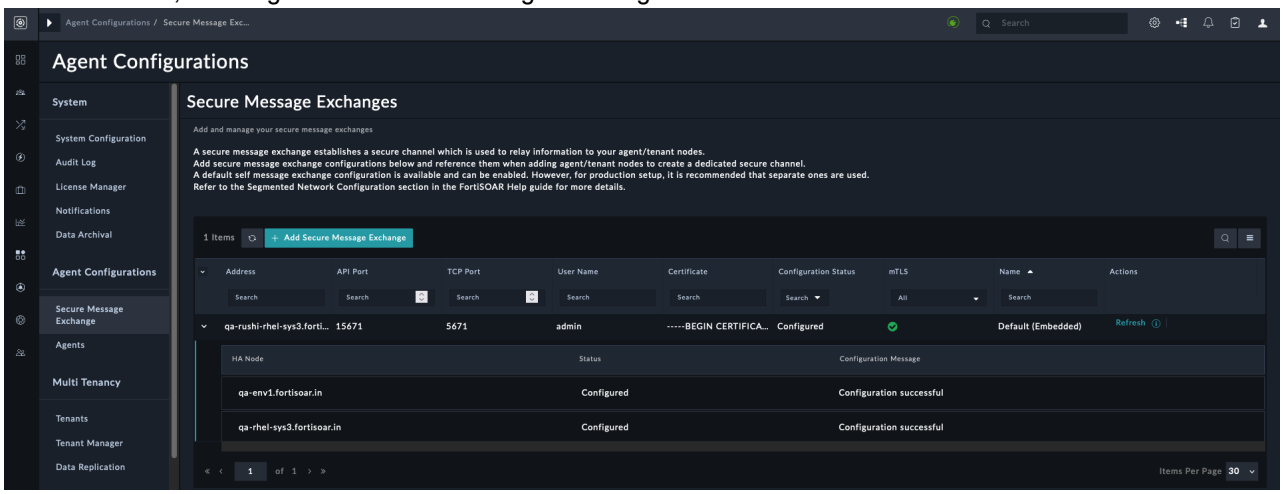
> For a production setup, it is recommended that you add and configure an external secure message exchange for handling scale and high availability.

Adding a Secure Message Exchange

Install a secure message exchange and then adding the reference of this secure message exchange in the tenant or FSR agent node(s) creates a dedicated secure channel of communication. You can have more than one secure message exchange in the configuration. You can distribute tenants across secure message exchanges based on the geographical locations, scale, or compliance policies of the respective customers.

To add a secure message exchange, do the following:

1. Log on to FortiSOAR as an 'administrator'.
   Note: Administrator role must have `Create`, `Update`, and `Delete` permissions on Secure Message Exchange.
2. Click the **Settings** ( ) icon to open the `System` page.
3. On the `System` page, you will see the `Agent Configurations` section. Click the **Secure Message Exchange** item in the left menu, to configure the secure message exchange.



4. Click **Add Secure Message Exchange** on the `Secure Message Exchanges` page.
   Important: To add a secure message exchange and configure tenants, you must have a role that has a minimum of `Create` and `Read` permissions on the **Secure Message Exchange** and **Tenants** modules. To update the details of the secure message exchange you additionally require `Update` permissions on the **Secure Message Exchange** and **Tenants** modules.
   To edit the configuration of an existing secure message exchange, click the secure message exchange row whose configuration you want to update. This displays the `Edit Secure Message Exchange` dialog. Update the configuration parameters, as required, in the dialog and click **Update**.

5. In the `Add New Secure Message Exchange` dialog, configure the following parameters:
   a. In the **Name** field, enter the name of the secure message exchange that you have configured to act as a secure channel of data replication between the FortiSOAR node and tenant or FSR agent nodes.
   b. In the **Address** field, enter the FQHN (Fully Qualified Host Name) of the secure message exchange.
   Important: Ensure that the FQHN matches the Certificate Name (CN) or the Subject Alternative Name (SAN) provided in the SSL certificate used to configure the secure message exchange.
   c. In the **Username** field, enter the username you will use to login to your secure message exchange as an administrator.
   By default, it is set as `admin`.
   d. In the **Password** field, enter the password you will use to login to your secure message exchange as an administrator.
   e. In the **Server Name Indication** field, enter the Server Name Indication (SNI) address for the Secure Message Exchange. You must specify the SNI address when the Secure Message Exchange is behind a reverse proxy or in a cluster behind a load balancer such as FortiADC.
   f. In the **API Port** field, enter the RabbitMQ Management port number that you had specified while configuring the secure message exchange, and ensure that the FortiSOAR node has outbound connectivity to the secure message exchange at this port.
   By default, it is set as `15671`.
   g. In the **TCP Port** field, enter the TCP port number that you had specified while configuring the secure message exchange, and ensure that the FortiSOAR node has outbound connectivity to the secure message exchange at this port.
   By default, it is set as `5671`.
   h. In the **CA Certificate** field, copy-paste the certificate text of the Certificate Authority (CA) that has signed the secure message exchange certificate in the `pem` format. You can also upload the certificate file. If it is a chain, then the complete chain must be provided.
   By default, the CA certificate for the FortiSOAR self-signed certificate is present at the following location: `/opt/cyops/configs/rabbitmq/ssl/cyopsca/cacert.pem`.
   Important: If in the future, your secure message exchange certificate expires, and you need to deploy a new certificate, then the new certificate must be copied back to the master node as well as the tenant's router entry.
   Client certificate can be opted from Certificate Authority in case CA signed certificates are deployed on secure message exchange or if there are no external CA signed certificates deployed. Client certificates can be generated using the following command on the Secure Message Exchange host:
   `sudo csadm mq client-certs generate --common-name MQ_CLIENT_CERT_COMMON_NAME [--target-dir MQ_CLIENT_CERT_TARGET_DIR]`
   Once the certificates are generated, the same can be used in the **Exchange Event Listener Client Cert** and **Exchange Event Listener Client Key** fields. For more information, see the CLI Administration chapter in the "Administration Guide."
   Note: The default self-signed certificates shipped with FortiSOAR are valid for two years from the inception of your FortiSOAR instance. For steps on regenerating these certificates, see the Regenerating Self-Signed certificates topic in the Monitoring & Optimization chapter of the "Administration Guide."
   i. (Optional) If you have enabled the mTLS, i.e., you require that clients that want to connect to secure message exchange must present their client certificate to the secure message exchange for verification, then you must also provide a pair of exchange event listener client certificates and exchange event listener client key, as follows:
      i. In the **Exchange Event Listener Client Cert** field, copy-paste the client certificate text or you can also upload the client certificate file.
      ii. In the **Exchange Event Listener Client Key** field, copy-paste the client key text or you can also upload the client key file.
   j. To save the configuration for the secure message exchange on the FortiSOAR node, click **Save**.

> After you have updated your secure message exchange configuration, the updated configurations might take some time to get reflected.

Clicking **Refresh** in the **Actions** column refreshes your secure message exchange (SME) configuration and clicking **Delete** deletes those SMEs that are currently not communicating with FSR Agents. Note that you must first configure another Secure Message Exchange server for communication with associated FSR Agents and then delete the old SME.

Adding an FSR agent

To add an FSR agent, do the following:

1. Log on to your base FortiSOAR node as an administrator and click the **Settings** icon to open the `System` page.
2. To add FSR agents, in the `Agent Configurations` section, click **Agents** in the left menu and click **Add**.
   To edit the configuration of an existing FSR agent, click the FSR agent whose configuration you want to update, which opens the FSR agent record in the detail view. Update the configuration parameters as required.
   If you no longer require an existing FSR agent, you can deactivate or deboard an FSR agent. To deactivate an agent, clear the **Enabled** checkbox in the FSR agent record.
   Deboarding an FSR agent is an irreversible operation which also deletes all data related to that FSR agent from the FortiSOAR node. For more information, see Deboarding FSR Agents.
3. On the `Agent` page, click **Add** to open the `Add New Agent` dialog and configure the following parameters for the FSR agent:
   a. In the **Name** field, enter the name of the FSR agent.
   b. From the **Auth Type** drop-down list, select the type of authentication you want to enforce for agents or clients to connect to the secure message exchange. You can choose between **Basic Auth** or **Certificate Auth**.
   c. From the **Secure Message Exchange** drop-down list, choose the secure message exchange that you have configured as the secure channel using which you can relay information to your FSR agent.
   Note: You must ensure that you have valid Secure Message Exchange configurations available before you configure the FSR Agent.
   d. (Optional) In the **Description** field, enter the description of the FSR agent.
   e. If you have selected **Certificate Auth** from the **Auth Type** drop-down list, then in the **Client Certificate** field, copy-paste the client certificate text of the Certificate Authority (CA) that has signed the secure message exchange certificate in the `pem` format. You can also upload the client certificate file.
   If you want to enforce client certificate verification with **Basic Auth** then also you must provide client certificate in this field, so that the secure message exchange will verify the certificate before allowing connection to any client.
   Note: If you are using CA signed certificates on secure message exchange, you must add these certificates to the truststore using the following command:
   `sudo csadm mq truststore add --ca-cert CA_CERT_PATH`
   For more information, see the CLI Administration chapter in the "Administration Guide."
   f. Similarly, in the **Client Key** field, copy-paste the client key text or you can also upload the client key file.
   Client certificate and key can be opted from Certificate Authority if CA signed certificates are deployed on secure message exchange or if there are no external CA signed certificates deployed. Client certificates can be generated using the following command:
   `sudo csadm mq client-certs generate --common-name MQ_CLIENT_CERT_COMMON_NAME [--target-dir MQ_CLIENT_CERT_TARGET_DIR]`
   Once the certificates are generated, the same can be used in the **Client Certificate** and **Client Key** fields. For more information, see the CLI Administration chapter in the "Administration Guide."

g. From the **Installer Type** drop-down list, select **Bash Script** to install the FSR Agent in a VM-based environment, or, select **Docker Compose** to install the FSR Agent on a Docker platform.

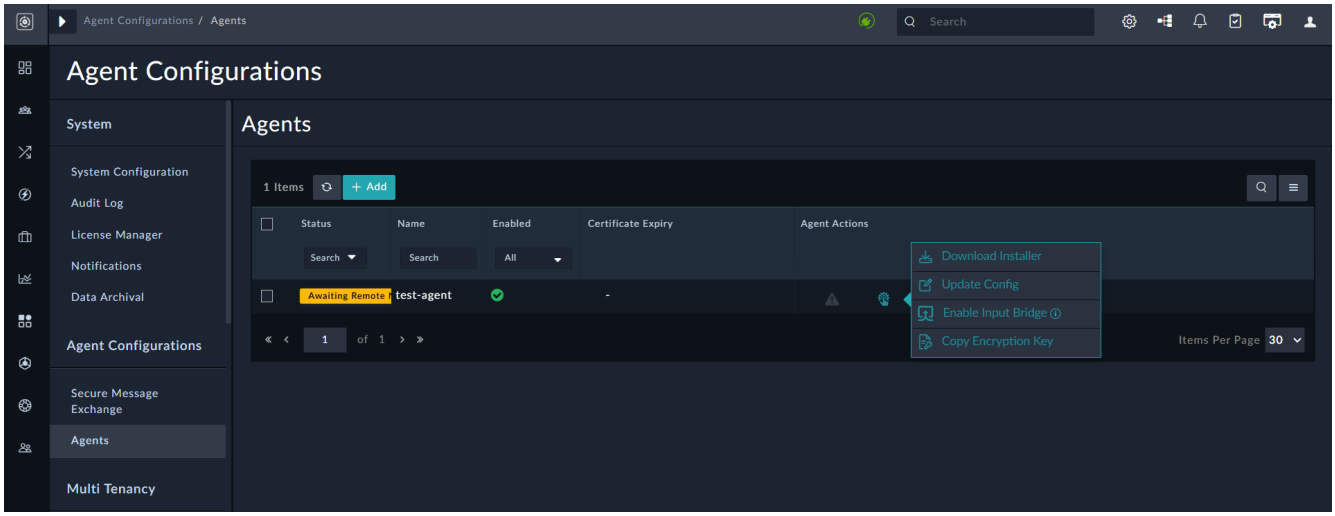
h. In the **Owners** section, the team of the user who is logged into the system is listed as pre-selected as the owner of the FSR agent. You can select additional teams that you want to add as owners of the FSR agent and click **Link Team**.
You can also delete the default team as the owner and replace it with another team. However, to add an agent, you must designate at least one team as the owner. If you remove all teams from the Owners section, the following warning message is displayed: '*As a best practice, ensure that you associate at least one of your teams with this tenant/agent else you will no longer able to access the tenant/agent settings or associated records after tenant/agent creation*
Teams help in governing RBAC for the FSR agent. While running connector actions using FSR agents, users will only see FSR agents that are associated with their teams.

i. To complete adding the FSR agent, click **Create**.

Once you have completed adding the FSR agent, you will see the status for the FSR agent. Following is the list of statuses that can be displayed:

- **Configuration In Progress**: Process of configuring the FSR agent has begun.
- **Awaiting Remote Node Connection**: Connection between the FortiSOAR node and secure message exchange is established and awaiting the connection to the FSR agent.
- **Remote Note Connected**: FSR Agent has been connected to the FortiSOAR node using secure message exchange.
- **Configuration Failed**: FSR Agent failed to be added on the secure message exchange.
- **Message Exchange Unreachable**: Secure message exchange is unreachable.
- **Remote Node Unreachable**: FSR Agent is unreachable from the FortiSOAR node.

If the connection between the FortiSOAR node and secure message exchange is established, then the `Status` column displays "Awaiting Remote Node Connection" along with a **Settings** icon in the `Agent Actions` column. Clicking the **Settings** icon, displays various options, including **Download Installer**. Click the **Download Installer** option to download the FSR agent installer and install the FSR agent as described in the following section.

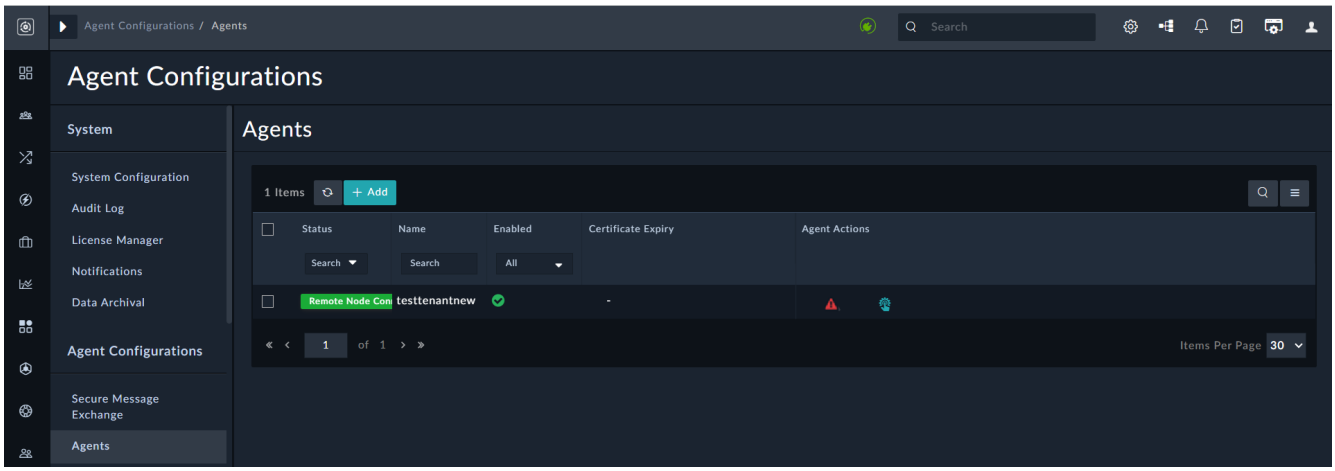


From release 7.6.5 onward, a **Copy Encryption Key** option is also available in the **Settings** icon that is part of the `Agent Actions` column. Clicking this option copies the unique encryption key generated for each FSR Agent to your clipboard. This key is used to manage communication between the FSR Agent and the FortiSOAR node to ensure secure credential exchange, and to encrypt and decrypt data in transit. Copy this key before running the agent installation

command (`sudo sh <agent-name>-install.bin`). During installation, you will be prompted to enter the copied key. For installation details, see the Installing an FSR Agent topic.

If the `Status` column displays statuses like "Message Exchange Unreachable" or "Remote Node Unreachable", the `Agent Actions` column will also display a **Retry** link that allows you to again perform the operation.

You might also see a **Warning** symbol in the `Agents Action` column as shown in the following image:



The **Warning** symbol indicates that the master cannot remotely execute or manage connector actions on the FSR agent, i.e., when remote connector management is disabled.

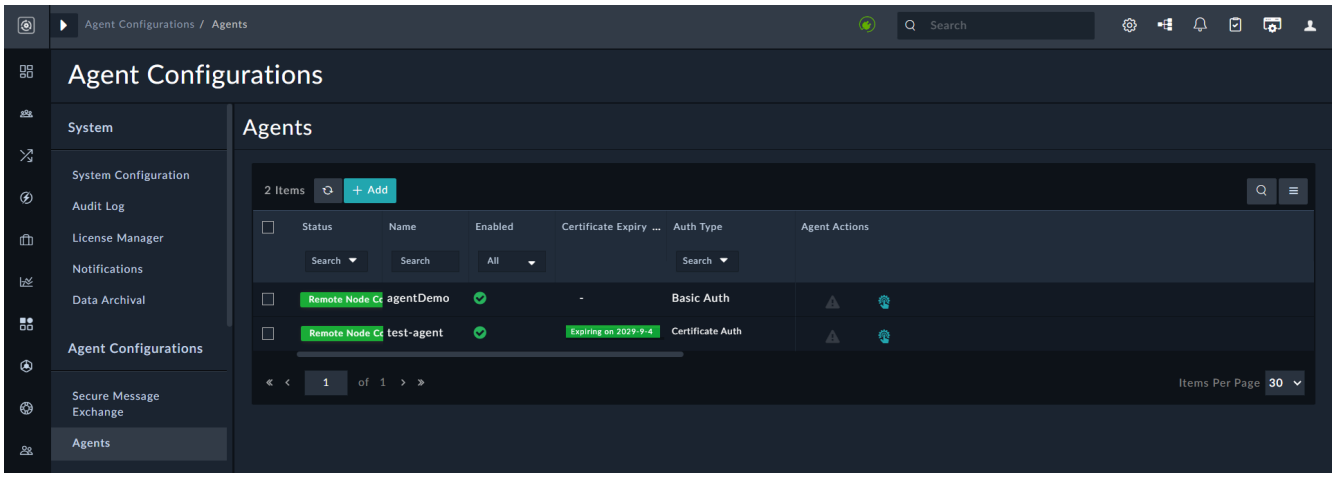
To enable or disable remote connector management on the FSR agent, you must have a minimum of `Upgrade` permission on the `Agent` module, and to disallow the master from remotely executing connector actions on an FSR agent, ensure that the FSR agent's version must be 6.4.4 and later.

Then, ssh to the FSR agent's VM and edit the `config.ini` file:
`sudo vi /opt/cyops-integrations/integrations/configs/config.ini`
Set the value of the `ENABLE_REMOTE_CONNECTOR_OPERATION` parameter as follows:

- To enable remote connector management, set the `ENABLE_REMOTE_CONNECTOR_OPERATION` parameter to `true`.
- To disable remote connector management, set the `ENABLE_REMOTE_CONNECTOR_OPERATION` parameter to `false`.

Once you have completed editing the `config.ini` file and setting the value of the `ENABLE_REMOTE_CONNECTOR_OPERATION` parameter, restart the `cyops-integrations-agent` service:
`sudo systemctl cyops-integrations-agent`
Restarting the `cyops-integrations-agent` service notifies these changes to the master node.

The agent grid will also contain an **Auth Type** column which displays whether the agent will use **Basic Auth** or **Certificate Auth** to connect to the secure message exchange. It also contains a **Certificate Expiry** column, which displays when the client certificate will expire in case of Certificate Authentication or Basic Authentication with Peer Verification. In the case of **Basic Auth**, if you provide certificates while adding agents, then the certificate expiry will be displayed; however, if you do not provide any certificates while adding agents, then a dash will be displayed in the Certificate Expiry column:



If you have enabled mTLS on secure message exchange, and you have added the secure message exchange client certificate and key after the FSR agent is added or if you have updated the secure message exchange client certificate and key after they have expired, then you require to first disable and again enable the agent to re-trigger the event listener and update agent status correctly.

Installing an FSR Agent

Before installing an FSR agent, ensure that you meet all the prerequisites. For more information, see Prerequisites for installing an FSR agent topic.

Before installing an FSR agent using an offline repository, ensure that you meet the prerequisites listed in *Installing an FSR agent using an offline repo* topic in the Deploying FortiSOAR using offline repositories chapter. After meeting the prerequisites, follow the steps in this procedure to complete the installation.
If you are using a self-trusted certificate on your offline repository, additional steps are required. For details, see *Installing an FSR agent using an offline repo, where the certificate on your offline repo is self-trusted* topic in the Deploying FortiSOAR using offline repositories chapter.

To install an FSR agent, follow these steps:

1. Click the **Download installer** link on the `Agents` page.
2. On the `Prepare and Download Agent Installer` dialog, select the connectors to include during installation from the **Include Specified Connectors to Install on Agent** drop-down list. FSR Agents use these connectors to remotely run actions to remotely execute actions on external networks. Options include.
   - **Do not install connectors by default**: Only the "Utilities" and "FSR Agent Communication Bridge" connectors are installed.
   - **Include pre-existing connectors on agent**: Bundles previously installed and configured connectors from an existing FSR Agent.
   - **Custom**: Manually select connectors from the **Select Connectors** list. Only connectors installed on the FortiSOAR (master) host can be selected.
   - **All connectors installed on Master**: Installs all connectors currently on the FortiSOAR (master) host.
3. Click **Download Installer**.

Once you click **Download Installer**, the installer file named as `<agent-name>-install.bin` is downloaded on your VM. Use '`scp`' or another secure transfer method to move the installer to the target FSR agent VM.

Install the FSR agent using the following commands:

1. `sudo sh <agent-name>-install.bin`
2. During installation, the agent installer prompts you to enter the unique encryption key generated for that agent.
   Note: You can copy the encryption key by clicking the **Settings** icon in the `Agent Actions` grid and then selecting the **Copy Encryption Key** option.
   After you enter the correct encryption key, the agent is installed on the FSR Agent VM.

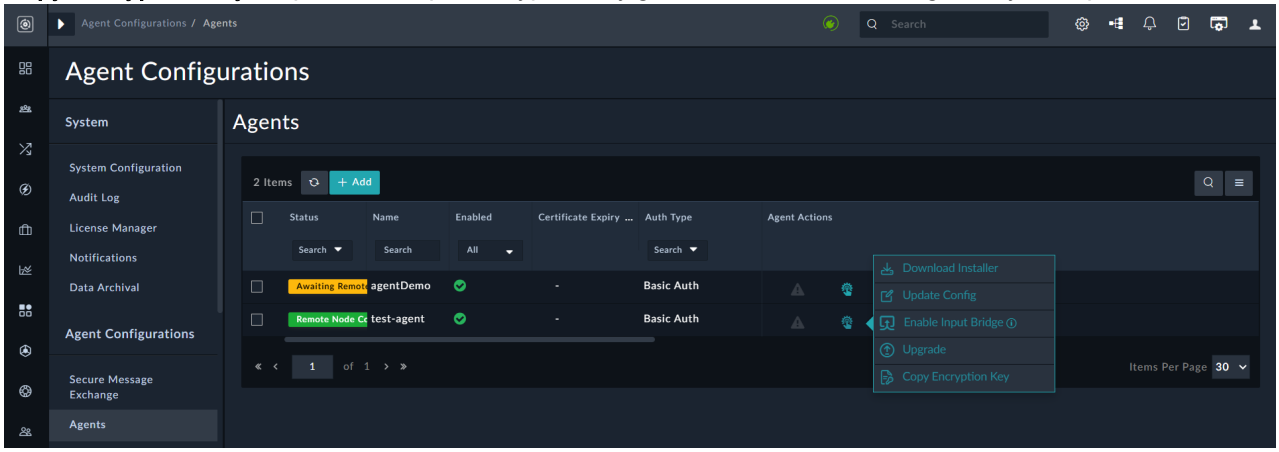
The installer installs the following for the FSR agent:

- PostgreSQL database.
- cyops-integrations-agent

Once the installation is complete, check the status of the `cyops-integrations-agent` service:
`sudo systemctl status cyops-integrations-agent.service`
Also, note that the password for PostgreSQL is the ID of the FSR Agent.

Once the FSR agent is installed you can perform actions by clicking the **Setting**s icon in the `Agent Actions` column:

- **Download Installer**: Install the FSR agent.
- **Update Config**: Update the configuration on the FSR Agent machine with the latest settings such as, changes in the hostname, SNI, address, etc.
- **Enable Input Bridge**: Configure the FSR Agent Communication Bridge. For details, see the FSR Agent Communication Bridge connector documentation.
- **Upgrade**: Automatically upgrade the FSR Agent.
  NOTE: Automatic upgrades are not supported for FSR Agents deployed on Docker platforms. For steps on upgrading an FSR Agent deployed on a Docker, see the Upgrading FSR Agent on Docker topic in the *Deploying an FSR Agent on a Docker Platform* chapter.
- **Copy Encryption Key**: Copies the unique encryption key generated for each FSR Agent to your clipboard.



Additionally, you can run connector actions on FSR agents, install and configure custom connectors, and perform administrative functions. For information on these functions and upgrading the FSR agent, see the FortiSOAR Agent Setup and Configuration chapter in the "Administration Guide".

Deboarding FSR Agents

To deboard existing FSR agents, you require to have `Read` and `Delete` permissions on the `Agents` module. Deboarding agents not only deletes the FSR agent but also removes the list of connectors installed and all the configurations of the connectors that have been installed on the specific FSR agent from the FortiSOAR node. Once you delete an FSR agent, you cannot retrieve any information related to that FSR agent, therefore you must be careful while performing this operation.

To deboard an FSR agent, log on to FortiSOAR as an administrator and click the **Settings** icon to open the `System` page. Click **Agents** in the left menu and on the `Agents` page, click **Delete**. FortiSOAR will display a warning dialog, click **Confirm** to deboard the FSR agent.

The steps for removing an FSR Agent deployed on a Docker platform differ from those outlined in this topic. For the correct procedure, see the Deboarding FSR Agent on Docker topic in the *Deploying an FSR Agent on a Docker Platform* chapter.

Moving an FSR agent to a new secure message exchange

If you have added a new secure message in your environment, and you want to move your FSR agent to the new secure message exchange, do the following on the FortiSOAR system:

1. Add the new secure message on your FortiSOAR system. For more information, see the Adding a Secure Message Exchange section.
2. Log on to your FortiSOAR node, master node in case of a distributed multi-tenant configuration, as an administrator and click the **Settings** icon to open the `System` page.
3. In the `Agent Configurations` section, click **Agents** in the left menu.
4. Click to open the FSR agent record that you want to move to the new secure message exchange.
5. In the FSR agent record, from the **Secure Message Exchange** drop-down list, select the secure message exchange on which you want to move the FSR agent.
6. Restart the `cyops-postman`, `uwsgi`, and `cyops-integrations-agent` services on the master node, using the following command:
   ```
   sudo systemctl restart cyops-postman uwsgi cyops-integrations-agent
   ```
   After you have completed updating the router information and restarting the services, you must download the FSR agent installer again and reinstall the FSR agent on the FSR agent VM.

Adding multiple disks and partitioning disks in your FortiSOAR VM

Multiple disks are supported for the FortiSOAR installation. Having multiple disks for the FortiSOAR installation has the advantage of being able to detach the disks that contain data and recover data, in the event of a FortiSOAR system crash.

To support partition sizes larger than 2 TB, FortiSOAR includes support for GPT disk partitioning in release 7.5.0. Previously FortiSOAR, only supported MBR disk partitioning, limiting partition sizes to 2 TB. If you already have a FortiSOAR instance and need a partition larger than 2 TB, we recommend creating a new FortiSOAR VM on release 7.5.0 or later and utilizing the Export and Import wizards to migrate your data from the old instance to the new one. This is required as FortiSOAR does not support a combination of MBR and GPT partitions.

For the procedure for recovering data see the Recovering Data topic. For the procedure for extending existing disks, see the Troubleshooting issues occurring in FortiSOAR due to insufficient space topic in the Troubleshooting FortiSOAR chapter.

You can add three more disks to your Virtual Machine (VM) and create separate Logical Volume Management (LVM) partitions for PostgreSQL and Elasticsearch data.

For example, you have added the following new disks:

- `/dev/sdb`: Recommended to have a thin provisioned disk for PostgreSQL data whose disk size is 500GB.
- `/dev/sdc`: Recommended to have a thin provisioned disk for Elasticsearch data whose disk size is 150GB.
- `/dev/sdd`: Recommended to have a thin provisioned disk for FortiSOAR™ RPM data whose disk size is 20GB.

Partitioning the disks

The steps mentioned in this topic are for a fresh installation of FortiSOAR that has been installed using the `.bin` file.

To partition the `/dev/sdb`, which is the disk for PostgreSQL data, run the following commands:

1. `sudo pvcreate /dev/sdb`
2. `sudo vgcreate vgdata /dev/sdb`
3. `sudo mkdir -p /var/lib/pgsql`
4. `sudo lvcreate -l 100%VG -n relations vgdata`
5. `sudo mkfs.xfs /dev/mapper/vgdata-relations`
6. `sudo mount /dev/mapper/vgdata-relations /var/lib/pgsql`
7. `sudo echo "/dev/mapper/vgdata-relations /var/lib/pgsql xfs defaults 0 0" >> /etc/fstab`

To partition the `/dev/sdc`, which is the disk for Elasticsearch data, run the following commands:

1. `sudo pvcreate /dev/sdc`
2. `sudo vgcreate vgdata /dev/sdc`
3. `sudo mkdir -p /var/lib/elasticsearch`
4. `sudo lvcreate -l 100%VG -n search vgsearch`
5. `sudo mkfs.xfs /dev/mapper/vgsearch-search`
6. `sudo mount /dev/mapper/vgsearch-search /var/lib/elasticsearch`
7. `sudo echo "/dev/mapper/vgsearch-search /var/lib/elasticsearch xfs defaults 0 0" >> /etc/fstab`

To partition the `/dev/sdd`, which is the disk for FortiSOAR RPM data, run the following commands:

1. `sudo pvcreate /dev/sdd`
2. `sudo vgcreate vgdata /dev/sdd`

3. `sudo mkdir -p /opt`
4. `sudo lvcreate -l 100%VG -n csapps vgapp`
5. `sudo mkfs.xfs /dev/mapper/vgapp-csapps`
6. `sudo mount /dev/mapper/vgapp-csapps /opt`
7. `sudo echo "/dev/mapper//vgapp-csapps /opt xfs defaults 0 0" >> /etc/fstab`

Recovering data

Commands for recovery of data must be prefixed with 'sudo'.

Following is the procedure for recovering data from the disks:

1. Deploy the recovery VM that has the same version of FortiSOAR installed (OVA or AMI) and power it **ON**.
2. Edit the `/etc/fstab` file:
   `sudo vi /etc/fstab`
   And comment out the lines that contain the word `vgdata` or `vgapp`.
3. Rename the `vgdata` and `vgapp` volume groups using the following command:
   `sudo vgrename vgdata old_vgdata`
   `sudo vgrename vgapp old_vgapp`
4. Stop all FortiSOAR™ services using the following command:
   `sudo csadm services --stop`
5. Run the `sudo umount /var/lib/pgsql/ && sudo umount /opt` command.
6. Deactivate the volume group using the following command:
   `sudo vgchange -a n old_vgdata old_vgapp`
7. Find out which disks contain the `vgdata` and `vgapp` volume groups using the '`pvs`' command.
   For example, if `vgdata` is on `/dev/sdb` and `vgapp` is on `/dev/sdd`, you require to skip these disks from `lvm` scanning. To skip the disks from `lvm` scanning add the 'skip' filter in the `/etc/lvm/lvm.conf` file as follows:
   a. Edit the `/etc/lvm/lvm.conf` file using the `sudo vi /etc/lvm/lvm.conf` command.
   b. In the "`devices {`" section in the `lvm.conf` file, add the following line:
      `filter = ["r|/dev/sdb|", "r|/dev/sdd|"]`
8. Stop the source VM and attach the existing PostgreSQL and RPM disks from the source VM to the recovery VM.
   The PostgreSQL disk will have the size of 150GB and the RPM disk will have the size of 10GB.
9. Run the `sudo vgs` command, which should display the `vgdata` and `vgapp` volume groups.
10. Edit the `/etc/fstab` file:
    `sudo vi /etc/fstab`
    And uncomment the lines that contain the word `vgdata` or `vgapp` that we had commented out in step 2.
11. Reboot your recovery VM.
12. Truncate the `envc` and `cascade` tables using the following command:
    `sudo psql -U cyberpgsql -d das -c "truncate envc cascade;"`
13. Update the cluster table using the following shell script. You can also create a temporary shell script using the following contents and run the same. For example,
    `sh temp_script_for_cluster_table_updation.sh:`

```
hardware_key=`sudo csadm license --get-hkey`
current_hostname=`hostname`
#First findout the number of nodes available in cluster table
number_of_nodes_in_cluster_table=`sudo psql -U cyberpgsql -d das -tAc "select COUNT(*) from
cluster;"`
if [ $number_of_nodes_in_cluster_table -eq 1 ]; then
```

```
    # Only single node is available in cluster, hence directly update the nodeid.
    sudo psql -U cyberpgsql -d das -c "UPDATE cluster SET nodeid='${hardware_key}';"
    sudo csadm ha set-node-name $current_hostname
elif [ $number_of_nodes_in_cluster_table -gt 1 ]; then
    # More than one node is available. Now update the nodeid where nodename in cluster
table matches with current hostname
    sudo psql -U cyberpgsql -d das -c "UPDATE cluster SET nodeid='${hardware_key}' where
nodename='${current_hostname}';"
else
    echo "Not able to update the cluster table"
fi
```

14. Change the *rabbitmq* password using the following commands:
```
sudo systemctl start rabbitmq-server
rabbitmq_password=`sudo grep "cyops.rabbitmq.password"
/opt/cyops/configs/rabbitmq/rabbitmq_users.conf | cut -d"=" -f2`
sudo rabbitmqctl change_password cyops $rabbitmq_password
```
15. Change the *elasticsearch* password using the following commands:
```
elasticsearch_password=`sudo csadm license --get-hkey`
printf $elasticsearch_password |sudo /usr/share/elasticsearch/bin/elasticsearch-keystore
add "bootstrap.password" -f
elasticsearch_password=`sudo /opt/cyops-auth/.env/bin/python
/opt/cyops/scripts/manage_passwords.py --encrypt $elasticsearch_password`
sudo /opt/cyops-auth/.env/bin/python /opt/cyops/scripts/confUtil.py -f
/opt/cyops/configs/database/db_config.yml -k "secret" -v $elasticsearch_password
```
16. Clear the API cache using the following commands. If any command fails, rerun that command:
```
sudo systemctl start php-fpm
sudo rm -rf /opt/cyops-api/var/cache/prod/
cd /opt/cyops-api
"sudo -u nginx php bin/console cache:clear --env=prod --no-interaction"
```
17. Refresh the keys using the following command:
```
sudo csadm certs --skip-hmac
```
18. Update the system using the following command:
```
cd /opt/cyops-api/
sudo -u nginx php bin/console cybersponse:system:update -la --env=prod --force
```
19. Change the hostname using the following command:
```
sudo csadm hostname --set <source-machine-hostname>
```
20. Reindex elasticsearch using the following command:
```
sudo -u nginx php /opt/cyops-api/bin/console cybersponse:elastic:create --env=prod
```
21. (Optional) If you do not remember the FortiSOAR UI password of your source instance and want to reset it to the default, which is 'changeme' for non-AWS instances and the 'instance_id' for AWS instances, run the following command:
```
sudo /opt/cyops-auth/.env/bin/python -c "import sys; sys.path.append(\"/opt/cyops-auth\");
import utilities.reset_user as reset_user; reset_user.start()"
```
22. Redeploy your FortiSOAR license.

For the embedded Secure Message Exchange, do the following:

1. Delete the existing embedded Secure Message Exchange using the following command:
```
/opt/cyops/scripts/api_caller.py --method DELETE --endpoint
https://localhost/api/3/routers/52c5cee8-5c28-4ed2-a886-ec8bf4dc5993
```
2. Enable the embedded Secure Message Exchange again using the following command:
```
sudo csadm secure-message-exchange enable
```

3. Reconfigure all your FSR agents. For information on FSR agents, see the FortiSOAR Agent Setup and Configuration chapter in the "Administration Guide".

Deploying FortiSOAR using offline repositories

This chapter describes the steps that you need to follow to deploy FortiSOAR using an offline repository. For information on upgrading FortiSOAR using an offline repository, see the "Upgrade Guide."

Prerequisites

- A virtual machine running one of the following (minimal installation):
  - RHEL 9.3/9.4/9.5/9.6
  - Rocky Linux 9.3/9.4/9.5/9.6
    NOTE: Release 7.6.6 has been tested with RHEL 9.6 and Rocky Linux 9.6.
- Minimum disk size: 500 GB at root (`/`) directory.
- Access to the FortiSOAR repository (repo.fortisoar.fortinet.com).
- Ensure that the `tmux` and `wget` packages are installed. If not, install them using:
  `sudo dnf install wget tmux -y`
- Ensure that the SSL certificates used for the offline repository are authorized by a Certificate Authority (CA). If you are using custom or internal CA certificates (not signed by a public Certificate Authority) for an offline server repository, you must manually add them to the truststore of both the offline repository server and each FortiSOAR instance as follows:
  a. Copy the certificate to the trusted anchors directory:
  `sudo cp <SSL_certificate>.crt /etc/pki/ca-trust/source/anchors/`
  b. Update the system truststore to recognize the new certificate:
  `sudo update-ca-trust extract`

Setting up the Offline Repository

FortiSOAR requires the offline repository system to run a supported OS version (RHEL or Rocky Linux 9.3-9.6). If you are using an existing repository, upgrade it to a supported OS version. This ensures compatibility with '`rsync`' and proper synchronization of modular metadata.

Complete the following steps before setting up the offline repository:

- Fortinet restricts repository access to registered source IP addresses. Contact Fortinet Support to register your public source IP address(es) by providing:
  - Public IP address(es)
  - Support entitlement details

  Wait for confirmation of IP registration before attempting repository synchronization.
- Ensure that the system hosting the offline repository:
  - Has outbound connectivity to the Fortinet repository.
  - Allows RSYNC over TCP port 873.
  - If your environment uses a proxy, configure `rsync` to use it:
    `sudo export RSYNC_PROXY=<proxy ip or fqdn>:<proxy port>`
- Ensure that the system hosting the offline repository uses a source IP address that is included in the Fortinet trusted host list. Check that you can successfully connect to and synchronize with the repository.

1. Execute the `tmux` command so your upgrade is not affected if your session times out:
   `# tmux`
2. Download `setup-fsr-offline-yum-repo.bin`:
   `wget --no-check-certificate https://repo.fortisoar.fortinet.com/7.6.6/setup-fsr-offline-yum-repo.bin`
3. Run the `setup-fsr-offline-yum-repo.bin` file as follows, where the `release_version` is FortiSOAR version that you want to synchronize:
   `# sudo sh /root/setup-fsr-offline-yum-repo.bin --release_version <release_version>`
   For example, to synchronize FortiSOAR version 7.6.6 use the following command:
   `# sudo sh /root/setup-fsr-offline-yum-repo.bin --release_version 7.6.6`
   Note: This script file creates a user whose ID and password are set to `yum`. This ID is used to assign ownership to the content in the '`/repos`' directory.
4. Check the default server certificate and server private key in the `/etc/httpd/conf.d/ssl.conf` file, and if required they should be replaced by editing the `ssl.conf` file:
   `sudo vi /etc/httpd/conf.d/ssl.conf`
   Replace the server certificate and private key as required:
   ```
   # Section Server Certificate
   SSLCertificateFile "/<path_to_cert>/<ssl_Certificate>.crt"
   # Section Server Private Key
   SSLCertificateKeyFile "/<path_to_cert>/<ssl_Certificate>.key"
   ```

   ```
   #   Server Certificate:
   # Point SSLCertificateFile at a PEM encoded certificate.  If
   # the certificate is encrypted, then you will be prompted for a
   # pass phrase.  Note that a kill -HUP will prompt again.  A new
   # certificate can be generated using the genkey(1) command.
   SSLCertificateFile /etc/pki/tls/certs/localhost.crt

   #   Server Private Key:
   #   If the key is not combined with the certificate, use this
   #   directive to point at the key file.  Keep in mind that if
   #   you've both a RSA and a DSA private key you can configure
   #   both in parallel (to also allow the use of DSA ciphers, etc.)
   SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
   ```

   After you have updated the certificates, restart the 'httpd' service:
   `# sudo systemctl restart httpd`

5. The `setup-fsr-offline-yum-repo.bin` script file synchronizes the repo. Therefore, if you want to resynchronize the repo, you must rerun the script. If you do not want to rerun the script manually, you can set up a cron job to perform this task. Use the following script to set up a cron job that will run daily at 00:00 hrs and synchronize the offline repo with the prod repo:
```
#!/bin/sh
#write out current crontab
sudo crontab -l > mycron
#sudo echo new cron into cron file
sudo echo "0 0 * * * sh /root/setup-fsr-offline-yum-repo.bin --release_version 7.6.6" >>
mycron
#sudo install new cron file
sudo crontab mycron
sudo rm mycron
```
Note: You can change the time of running the cron job as per your convenience.

Deploying FortiSOAR using the Offline Repository

1. Ensure that the offline repository host is accessible from the FortiSOAR appliance and run the `tmux` command so your upgrade is not affected if your session times out:
```
# tmux
```
Note: Do not run `tmux` from the root shell if you have logged in as the csadmin user
2. If you are using your private repository to install FortiSOAR, then use the following command to export the "custom_yum_url" variable before running the fresh install script:
```
sudo export custom_yum_url=<"custom_yum_url_name">
```
For example, `sudo export custom_yum_url="offline-repo.fortisoar.in"`
3. Download the installer for FortiSOAR 7.6.6 using the following command:
```
wget https://<offline repo>/7.6.6/install-fortisoar-7.6.6.bin
```
4. To install FortiSOAR 7.6.6, run the following command:
```
# sudo sh install-fortisoar-7.6.6.bin
```

   If you have not deployed an SSL certificate on your offline repository or if you have a self-signed certificate deployed, then run the following command on a plain Rocky Linux or RHEL system, to ignore the SSL check while installing FortiSOAR:
```
# sudo sh install-fortisoar-7.6.6.bin --ignore-ssl-check
```
5. Login to the FortiSOAR CLI and continue to configure FortiSOAR and add your FortiSOAR license. For more information, see the Deploying FortiSOAR chapter.
   Note: You can add CA-signed certificates in OS as a trusted certificate using the steps mentioned in the `Adding CA-signed certificates in Rocky Linux or RHEL as trusted certificates` topic in the Additional Configurations chapter.

Installing an FSR agent using an offline repo

Prerequisites

Before installing an FSR agent using an offline repository, delete the following files from the `/etc/yum.repos.d/` directory:

- `rocky-addons.repo`
- `rocky-devel.repo`
- `rocky-extras.repo`
- `rocky.repo`

Next, add the `fsr-os.repo` file to the same directory (`/etc/yum.repos.d/`). This file should point either to your offline repository or to the OS repository, depending on where the required packages are available.

Sample fsr-os.repo file

```
# Copyright start
# Copyright (C) 2008 - 2025 Fortinet Inc.
# All rights reserved.
# FORTINET CONFIDENTIAL & FORTINET PROPRIETARY SOURCE CODE
# Copyright end
#

[fsr-rockylinux-BaseOS]
name=FortiSOAR Rockylinux-$releasever BaseOS Repository
baseurl=https://$product_yum_server/$fsr_release_version/rockylinux/9/BaseOS/x86_64/os/
enabled=1
gpgcheck=1
gpgkey=https://$product_yum_server/$fsr_release_version/rockylinux/RPM-GPG-KEY-Rocky-9

[fsr-rockylinux-AppStream]
name=FortiSOAR Rockylinux-$releasever AppStream Repository
baseurl=https://$product_yum_server/$fsr_release_version/rockylinux/9/AppStream/x86_64/os/
enabled=1
gpgcheck=1
gpgkey=https://$product_yum_server/$fsr_release_version/rockylinux/RPM-GPG-KEY-Rocky-9

[fsr-rockylinux-extras]
name=FortiSOAR Rockylinux-$releasever Extras Repository
baseurl=https://$product_yum_server/$fsr_release_version/rockylinux/9/extras/x86_64/os/
enabled=1
gpgcheck=1
gpgkey=https://$product_yum_server/$fsr_release_version/rockylinux/RPM-GPG-KEY-Rocky-9

[fsr-rockylinux-CRB]
name=FortiSOAR Rockylinux-$releasever CRB Repository
baseurl=https://$product_yum_server/$fsr_release_version/rockylinux/9/CRB/x86_64/os/
enabled=1
gpgcheck=1
```

```
gpgkey=https://$product_yum_server/$fsr_release_version/rockylinux/RPM-GPG-KEY-Rocky-9

[fsr-epel-9]
name=FortiSOAR EPEL-$releasever Repository
baseurl=https://$product_yum_server/$fsr_release_version/epel/9/Everything/x86_64/
enabled=1
gpgcheck=1
gpgkey=https://$product_yum_server/$fsr_release_version/epel/RPM-GPG-KEY-EPEL-9
```

In the `fsr-os.repo` file:

- Replace `$product_yum_server` with the name of the server from which the packages should be downloaded.
- Replace `$fsr_release_version` with the version of FSR that you want to install.

Once you have made these updates, follow the steps mentioned in the *Installing a FSR agent* topic in the Deploying FortiSOAR chapter to complete the installation.

Installing an FSR agent using an offline repo, where the certificate on your offline repo is self-trusted

To install an FSR agent using an offline repo, where the certificate on your offline repo is self-trusted, you need to follow some steps in addition to the steps mentioned in the *Installing a FSR agent* topic in the Deploying FortiSOAR chapter. If you are installing an FSR agent using an offline repo, where the Certificate Authority is known to the FSR agent VM, you can follow the steps (no additional steps needed) mentioned in the *Installing a FSR agent* topic in the Deploying FortiSOAR chapter.

To install an FSR agent using an offline repo, where the certificate on your offline repo is self-trusted, do the following:

1. Before running the FSR Agent installer (`<agent-name>-install.bin`), edit the `/etc/yum.conf` file:
   `sudo vi /etc/yum.conf`
   And add `sslverify=false`, and then save the `yum.conf` file.
   Important: You also require to update the `<agent-name>-install.bin` file so that it points the FQDN of your own repo, using for example the following command:
   `sudo sed -n 's/product_yum_server="repo.fortisoar.fortinet.com"/product_yum_server="<offline_repo_hostname>"/1p' <agent-name>-install.bin`
   OR
   You can also modify the `/etc/hosts` file (`sudo vi /etc/yum.conf`) in your FSR Agent VM's so that it thinks that repo.fortisoar.fortinet.com resolves to your offline repo vm's IP address, using for example, the following command:
   `sudo echo "X.X.X.X repo.fortisoar.fortinet.com" >> /etc/hosts`
2. Install the FSR Agent using the FSR Agent installer as mentioned in the *Installing a FSR agent* topic in the Deploying FortiSOAR chapter.
3. Once the FSR Agent installer process is completed, edit the `sudo vi /opt/cyops-integrations/.env/pip.conf file` to add the `trusted-host` parameter and edit the `extra-index URL` to point to the offline repo, and then save the `pip.conf` file.
   Sample of the `pip.conf` file:
   ```
   [global]
   trusted-host = offline-repo.fortisoar.in
   extra-index-url= https://offline-repo.fortisoar.in/prod/connectors/deps/simple/
   ```
4. Run the following command to install connector dependencies:
   To find out the connector requirements (dependencies) and install the requirements:
   `for requirements in $(find /opt/cyops/configs/integrations/connectors/ -name`

```
requirements.txt); do sudo -u fortisoar /opt/cyops-integrations/.env/bin/pip install -r
$requirements; done
```

Troubleshooting

Peer Certificate issue not recognized error

If you have not deployed an SSL certificate deployed on your offline repo or you have a self-signed certificate deployed on your offline repo, then run the following command on plain Rocky Linux or RHEL system if you are installing version 7.2.0 or later, for example, for version 7.6.6, do the following:

```
# sudo sh install-fortisoar-7.6.6.bin --ignore-ssl-check
```

This command ignores the SSL check while installing FortiSOAR. However, you can get the following error while installing FortiSOAR on a plain Rocky Linux or RHEL system:

```
"[Errno 14] curl#60 - "Peer's Certificate issuer is not recognized."
```

Resolution

On the plain Rocky Linux or RHEL system, perform the following steps:

1. Edit the `/etc/yum.conf` file:
   `sudo vi /etc/yum.conf`
2. Add `sslverify=false` parameter.
3. Save the `yum.conf` file.
4. Restart the installation.

Deploying FortiSOAR on a Docker Platform

You can deploy FortiSOAR on Docker platforms such as VMware ESX or AWS. This allows you to easily provision FortiSOAR into your microservice's architecture and use it as cloud-native and DevOp-enabled.

You can also deploy the FortiSOAR Docker on Amazon Elastic Kubernetes (EKS) cluster, in the Amazon Web Services (AWS) Cloud. For more information, see the Deploying FortiSOAR Docker on Amazon Elastic Kubernetes cluster chapter. For information on upgrading your FortiSOAR Docker image, see the "Upgrade Guide."

The following topics introduce how to deploy the FortiSOAR image on Docker.

Planning

Prerequisites

- To deploy the FortiSOAR image on Docker, you must have already installed Docker in your environment. If not, refer to the Docker official website for Docker installation instructions: https://docs.docker.com/.
- To check whether Docker has been successfully installed, run `docker version`.
- For resource requirement specifications, review the Deploying FortiSOAR chapter.
- To deploy the FortiSOAR image on an Offline Docker, following are additional prerequisites that must be met:
  - Review the Deploying FortiSOAR using offline repositories chapter.
  - For licensing information about this type of deployment, review the "FortiSOAR licensing using FortiManager" topic in the Licensing FortiSOAR chapter.
  - Once you have completed deploying the FortiSOAR image on an offline docker, make sure to follow the additional steps outlined in the Post-deployment instructions for an offline Docker topic.

System Requirements

Supported Hypervisors

- Docker Engine CE 18.09.1 or higher versions, and the equivalent Docker Engine EE versions.

For best performance in hypervisor deployments, install FortiSOAR on a “bare metal” (Type 1) hypervisor. Hypervisors that are installed as applications on top of a general-purpose operating system (Windows, Mac OS X, or Linux) host have fewer computing resources available due to the host OS’s own overhead.

To ensure high performance, it is recommended to deploy FortiSOAR on machine types with a minimum of 8 vCPUs and a memory size larger than 32 GB.

Downloading the FortiSOAR Docker image

You can download the required FortiSOAR Docker image from the support portal.

To download the FortiSOAR Docker image, do the following:

1. Log on to support.fortinet.com.
2. Click **Support** > **Firmware Download**.
3. On the `Fortinet Firmware Images And Software Releases` page, from the **Select Product** drop-down list, select **FortiSOAR**.
   The page contains information about released versions of FortiSOAR images, and contains two tabs: **Release Notes** and **Downloads**.
   To view the Release Notes for a particular version, click the version and build number link, which opens the FortiSOAR Document Library, from where you can view or download the release notes for that particular version.
4. To download the Docker image, do the following:
   a. Click the **Download** tab.
   b. Navigate through the directory structure in the format, `<version number category>` > `<major version >` > `<minor version>`, to open the page containing the required image. For example, to download a Docker image for version 7.4.1, click **v7.00** > **7.6** > **7.6.6**, and locate the required Docker image.
   c. Download the Docker image by clicking the **HTTPS** link.
   An HTTPS connection is used to download the Docker image.
   d. Click the **Checksum** link for the image that you have downloaded.
   The image file name and checksum code are displayed in the `Get Checksum Code` dialog box.
   e. Confirm that the checksum of the downloaded image file matches the checksum provided on the download site.

Deploying the FortiSOAR Docker image

1. Download the FortiSOAR Docker installer from `https://repo.fortisoar.fortinet.com/<release_version>/install-fortisoar-docker-<release_version>.bin`
   For example, `https://repo.fortisoar.fortinet.com/7.6.6/install-fortisoar-docker-7.6.6.bin`
2. Load the downloaded Docker image using the following command:
   `docker load -i <image-path>`
3. Extract the default `fortisoar.env` file using the following command:
   `sudo ./install-fortisoar-docker-<release_version>.bin --export-default-env`
   For example, `sudo ./install-fortisoar-docker-7.6.6.bin --export-default-env`
   NOTE: This command exports the `fortisoar.env` file to the current directory.
4. Update the `fortisoar.env` file as per your environment. For more information, see Understanding the `fortisoar.env` file topic.
5. Once you have updated the `fortisoar.env` file, run the following command:
   `sudo ./install-fortisoar-docker-<release_version>.bin --env-file fortisoar.env`
   For example, `sudo ./install-fortisoar-docker-7.6.6.bin --env-file fortisoar.env`
   NOTE: The `fortisoar.env` file is an important configuration file. Therefore, it is recommended that you take a backup of this file for future reference.
6. To connect to FortiSOAR Docker using SSH, use the following CLI:
   `sudo docker exec -it <FSR container id or name> bash`

If the FortiSOAR Configuration Wizard fails when provisioning your instance, then a failure screen detailing the status of each configuration step is presented on the FortiSOAR UI, making it simpler to identify the issue. Before using FortiSOAR, you must use the CLI to fix any issues with the failed steps as their functioning might be hampered. However, if you choose to access FortiSOAR without rectifying the failed steps, which can cause FortiSOAR functionality to deteriorate, a **Proceed Anyway** button is provided that lets you to use the product while acknowledging the configuration failure. If the FortiSOAR UI is not accessible even after clicking **Proceed Anyway**, you can try to fix the issue using the steps mentioned in the Deploying FortiSOAR chapter.

Also note that the stop, start, or restart operations on the container leads to a boot time of approximately 10 minutes while it rebuilds.

Understanding the `fortisoar.env` file

The FortiSOAR Docker installer uses the `fortisoar.env` file for information for FortiSOAR container configuration. Use the FortiSOAR installer to export the default configuration using the following command:
`sudo ./install-fortisoar-docker-<release_version>.bin --export-default-env`
For example, `sudo ./install-fortisoar-docker-7.6.6.bin --export-default-env`
This command exports the `fortisoar.env` file to the current directory.

Sample `fortisoar.env` file:

```
# cat fortisoar.env
#
# Do not use space before or after of =
# You can retrieve the image id by executing the 'docker images' command
#
IMAGE_ID=1xxxxxxxxxx
PROJECT_NAME=fortisoar
HOSTNAME_DOCKER_HOST=docker-host.myorg.mydomain
HOSTNAME_CONTAINER=fsr-container.myorg.mydomain
PORT_UI=443
ENABLE_SME=true
PORT_SME=5671
EXTRA_PARAM="docker --publish docker_port/host_port:container_port"
# RAM in GB
RAM=32
CPUS=8
IP_REPO=10.1xx.2xx.1xx
HOSTNAME_REPO=fortisoar-offline.myorgdomain
IPV6=false
#
```

Configurable parameters of the `fortisoar.env` file:

- `IMAGE_ID`: The image ID of your FortiSOAR Docker image. You can find the image ID using `docker images`.
- `PROJECT_NAME`: The identifier for your FortiSOAR container resources. The FortiSOAR installer creates the container name as '`<PROJECT_NAME>_fortisoar_1`', and names all the required volumes as '`<PROJECT_NAME>_fortisoar_*`'.

- `HOSTNAME_DOCKER_HOST`: The DNS of the Docker host, which is added by default to the self-signed certificate SAN list.
- `HOSTNAME_CONTAINER`: The DNS of the Docker host, which is added by default to the self-signed certificate SAN list.
  NOTE: The value of this parameter is set as the default hostname of the Docker.
- `PORT_UI`: The host port of the Docker used to access the FortiSOAR UI. The traffic on this Docker host port is forwarded by the Docker to the container on port 443 (default). For example, if you want to change the default port (443) and set the `PORT_UI` as 5443, then you can access FortiSOAR at `https://<HOSTNAME_DOCKER_HOST>:5443/`.
- `ENABLE_SME`: By default, the FortiSOAR Docker image enables the embedded Secure Message Exchange (SME), and therefore the value of this parameter by default is set to `true`.
- `PORT_SME`: By default, the FortiSOAR Docker image enables the embedded SME. The `PORT_SME` is the host port of the Docker to access the TCP port of the embedded SME. The traffic on this Docker host port is forwarded by Docker to the container port 5671.
- `RAM`: The value of the RAM (in GB) of the FortiSOAR container.
- `CPUS`: The number of CPUs for the FortiSOAR container.
- `IP_REPO`: Only applicable if you are using offline repository for FortiSOAR. This parameter refers to IP address of the offline repository. The `/etc/hosts` file of the container contains the following entry:
  `<IP_REPO> repo.fortisoar.fortinet.com`
- `HOSTNAME_REPO`: Only applicable if you are using offline repository for FortiSOAR. This parameter refers to hostname of the offline repository. For offline repository, you must update the CA bundle/chain of the offline repository certificate in the container using the following steps:
  `# sudo docker cp <offline-repo-certificate-CA-bundle> <FortiSOAR-container-name>:/etc/pki/ca-trust/source/anchors/`
  `# sudo docker exec -ti <FortiSOAR-container-name> bash -c "update-ca-trust force-enable"`
  `# sudo docker exec -ti <FortiSOAR-container-name> bash -c "update-ca-trust extract"`
- `IPV6`: This parameter determines whether or not IPv6 should be enabled for the docker. Specify true to enable IPv6 after you have ensured that the docker runtime is able to assign IPv6 to the FortiSOAR container.

Post-deployment instructions for an offline Docker

After deploying the FortiSOAR image on an offline docker, make sure to do the following:

- Update the `pip.conf` file with the correct hostname of the offline repository.
- Remove any reference to the standard mirror in the pip install files.
- Install `NetworkManager` using the `sudo yum install NetworkManager` command to ensure Python libraries function properly within the Docker image.
- Ensure that the SSL certificates used for the offline repository are authorized by a Certificate Authority (CA). Export this CA certificate from the offline repository and then install that on the FortiSOAR Docker image.

Running the FortiSOAR Docker

Prerequisites

If your Docker runtime uses SELinux, ensure you enable the 'setsebool' parameter before starting the FortiSOAR Docker as follows:

```
sudo setsebool -P container_manage_cgroup 1
```

Mode of running the FortiSOAR Docker

The FortiSOAR Docker runs in the 'non-privileged' mode. The following default privileges are assigned to the FortiSOAR, which are also applied, by default, to your FortiSOAR instance:

- SYS_ADMIN: Required for bind mounting /tmp on /var/tmp and for various systemd services.
- SYS_RAWIO: Required for running 'dmidecode' and for various systemd services.
- SYS_TIME: Required for running 'ntpd'.
- SYS_PTRACE: Required for running 'systemd-journal'.

FortiSOAR High Availability Support on Dockers

FortiSOAR High Availability (HA) clusters are supported on multiple Docker hosts, requiring FortiSOAR HA nodes to be deployed on separate Docker host instances. Adding HA support provides improved uptime, scalability, load balancing, fault tolerance, and simplified management for your FortiSOAR Dockers.

FSR Agents are not supported on FortiSOAR HA clusters that are deployed on separate Docker host instances. This feature is also not supported on MSSP environments.

Steps to add another FortiSOAR node on a different Docker host and then create a FortiSOAR HA cluster are as follows:

1. Deploy the FortiSOAR Docker image to create another node on a separate host machine using the steps mentioned in the Deploying the FortiSOAR Docker image topic.
2. Edit the fortisoar.env file as follows (see the Understanding the fortisoar.env file topic for information on the fortisoar.env file)
   a. Update the HOSTNAME_DOCKER_HOST parameter to the DNS of the Docker host of your other node. This must not be set to localhost.
   b. Update the HOSTNAME_CONTAINER parameter to the DNS of the Docker host of your other node. This must not be set to localhost.
   c. (Optional) Change the ENABLE_SME parameter flag to "false" to disable the embedded Secure Message Exchange (SME). By default, the FortiSOAR Docker image enables SME, with the ENABLE_SME flag set to "true". In release 7.6.2 and later, you can retain this default setting.
   d. Update the EXTRA_PARAM parameter to publish the Docker ports for your other node:
   EXTRA_PARAM="---publish docker_port/host_port:container_port"
   For example:

```
EXTRA_PARAM="--publish 5671:5671 --publish 9200:9200 --publish 5432:5432 --publish
4369:4369 --publish 25672:25672 --publish 35672:35672 --publish 35673:35673 --publish
35674:35674 --publish 35675:35675 --publish 35676:35676 --publish 35677:35677 --publish
35678:35678 --publish 35679:35679 --publish 35680:35680 --publish 35681:35681 --publish
35682:35682 --add-host <other_node_hostname>:<other_node_ip>"
```

3. To configure the other node as a secondary node, do the following:
   a. SSH to your active primary node and run the `sudo csadm ha export-conf` command to export the configuration details of the active primary node to a configuration file named `ha.conf`.
   You must copy the `ha.conf` file from the active primary node to the node that you want to configure as a secondary node.
   b. On the active primary server, add the hostnames of the secondary nodes to the allowlist, using the following command:
   `# sudo csadm ha allowlist --nodes <other_node_hostnames>`
   You can add a comma-separated list of hostnames of the cluster nodes that you want to add to the allowlist after the `--nodes` argument.
   c. Ensure that all HA nodes are resolvable through DNS, and then SSH to the host that you want to configure as a secondary node and run the following command:
   `# sudo csadm ha join-cluster --status <active, passive> --role <primary, secondary> --conf <location of the ha.conf file>`
   For example,
   `# sudo csadm ha join-cluster --status passive --role secondary --conf tmp/ha.conf`
   This adds the node as a secondary node in the cluster.

Troubleshooting

FortiSOAR HA nodes deployed across various Docker hosts encounter post-takeover join cluster failures

After running the takeover operation, the join cluster operation on existing HA cluster nodes fails.

Resolution

FortiSOAR does not provide automated "join cluster" for other nodes in the HA cluster once "takeover" has been executed on a system. You must manually run the join cluster operation on the other nodes by exporting the config file from the new primary node.

SSO login is being redirected to the hostname of the FortiSOAR node instead of the load balancer hostname

After configuring FortiSOAR High Availability (HA) clusters on multiple Docker hosts with load balancers, you observe that the SSO login page is redirected to the hostname of the FortiSOAR node rather than the hostname of the load balancer.

Resolution

To redirect the SSO login page correctly to the load balancer' s hostname, add the following key to the `fortisoar.env` file:
`HOSTNAME_LOAD_BALANCER: Hostname of load balancer`

For information on the `fortisoar.env` file, see the Understanding the `fortisoar.env` file topic.

Frequently Asked Questions

How to clean up the FortiSOAR container?

To clean up the FortiSOAR container, run the following commands:

```
docker stop <container id>
docker rm <container id>
docker volume prune
```

What happens if users re-installs the FortiSOAR container without removing its volumes?

If users re-installs the FortiSOAR container without removing its volumes, then the FortiSOAR container is restored from its last saved state.

How to resolve the issue of Elasticsearch-based recommendations not working on a FortiSOAR instance on a Docker platform?

By default, Elasticsearch-based recommendations do not work on a FortiSOAR Docker instance due to size limitations. To know more about Elasticsearch-based recommendations, see the Recommendation Engine topic in the *Application Configuration and Customization* chapter of the "Administration Guide".

To use Elasticsearch-based recommendations, you must increase the memory allocated to Elasticsearch to 4 GB, using the following steps:

1. Update the value of the following parameters in the `fsr.options` file `sudo vi /etc/elasticsearch/jvm.options.d/fsr.options` file to 4 GB:
   `-Xms4g`
   `-Xmx4g`
2. Restart the Elasticsearch service using the following command:
   `sudo systemctl restart elasticsearch`
3. Reindex Elasticsearch data using the following command:
   `sudo -u nginx php /opt/cyops-api/bin/console app:elastic:create --sync=true`
   Now, you should be able to view Elasticsearch-based recommendations on your FortiSOAR Docker instance.

Deploying FortiSOAR Docker on an Amazon Elastic Kubernetes Cluster

You can deploy the FortiSOAR Docker on an Amazon Elastic Kubernetes (EKS) cluster, in the Amazon Web Services (AWS) Cloud. The product deployment solution facilitates an orchestrated deployment of the FortiSOAR components on EKS.

Required Terminology

The following table describes the important terms for FortiSOAR deployment on the EKS cluster.

Term	Description
Pod	A Pod is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers. For more information, see the Pods section in the Kubernetes documentation.
StatefulSet	StatefulSet is the workload API object used to manage stateful applications, and it represents a set of Pods with unique, persistent identities, and stable hostnames. For more information, see the StatefulSet section in the Kubernetes documentation and the Statefulset section.
Service	A Service enables network access to a set of Pods in Kubernetes. For more information, see the Service section in the Kubernetes documentation and the Load Balancer (Service) section.
Persistent Volume Claim	A PersistentVolumeClaim (PVC) is a request for storage by a user. For more information, see the Persistent Volumes section in the Kubernetes Documentation and the Persistent Volume Claim section.
Persistent Volume	A PersistentVolume (PV) is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using storage classes. For more information, see the Persistent Volumes section in the Kubernetes documentation and the Persistent Volume Claim section.
StorageClass	A StorageClass provides a way for administrators to describe the offered "classes" of storage. For more information, see the Storage Classes section in the Kubernetes documentation and the Storage Class section.
Namespace	Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called namespaces. For more information, see the Namespaces section in the Kubernetes documentation.

Preparing the environment for installing FortiSOAR on EKS

EKS-specific Requirements

- Use Kubernetes version 1.23 and later.
- The AWS default CNI is used during cluster creation.
- Create a node group with only one availability zone (The node group in EKS must not cross the availability zone), and that has at least one instance of 32GB RAM and 8 CPUs based on x86/64 or amd64 architecture. You can use t2.2xlarge.
- All the nodes in the node group must be running on the Linux operating system.
- While creating the node group you need an 'IAM' role with at least the following policies:
  - `AmazonEKSWorkerNodePolicy`
  - `AmazonEC2ContainerRegistryReadOnly`
  - `AmazonEKS_CNI_Policy`
- From Kubernetes version 1.24, docker support, 'dockershim', is removed. Ensure that your NodeGroup instance has the '`containerd`' runtime installed. For more information on docker removal from Kubernetes, see the dockershim deprecation document.
- Use an existing AWS Elastic Container Registry or create a new one, and ensure that the EKS has full access to pull images from the Elastic Container Registry.
- Deploy an AWS load balancer controller add-on in the cluster. This is required for automatically creating a network load balancer when the FortiSOAR service is created. For more information on installing the add-on, see the Installing the AWS Load Balancer Controller add-on document.
- Ensure that the EBS CSI driver and its requirements are fulfilled. Depending on your Kubernetes version, you might need to install the EBS CSI driver. For more information, see the EBS CSI guide.

Host-specific Requirements

You can install the following tools on your host while using Kubernetes to ease its administration:

- **AWS CLI**: For more information on installing the AWS CLI, see the Installing or updating the latest version of the AWS CLI document.
- **Kubectl CLI**: For more information on installing the Kubectl CLI, see the Installing kubectl document.
- **Docker**: Install and configure Docker to enable the push of the container images to the container registry. If you are using ECR, ensure that you are able to upload images to ECR correctly.

Apart from installing the tools, you also need to do the following:

- If an IAM role needs access to the EKS cluster, then run the following command from the system that already has access to the EKS cluster:
  `kubectl edit -n kube-system configmap/aws-auth`
  For more information on creating an IAM role, see the Enabling IAM user and role access to your cluster document.
- Log in to the AWS environment and run the following command on the AWS CLI to access the Kubernetes cluster:
  `aws eks --region <region_name> update-kubeconfig --name <cluster_name>`

- Ensure that you have free space of approximately 6GB at the location where the FortiSOAR Docker image will be downloaded and then uploaded to the Docker registry. Also, ensure that approximately 15GB of free space is available at the `/var/lib/docker` location so that the images can be loaded into the Docker cache, before being pushed to the container registry.

Recommendations for FortiSOAR deployment on EKS

- By default, FortiSOAR creates a storage class for EBS with type gp2. For better performance, you can override the default class of your EBS volume in the `yaml` file of your FortiSOAR storage class. For more information, see the Amazon EBS volume types document.
- Do not delete the Disk/EBS volumes linked with the PV used in the FortiSOAR deployment because this might result in data loss.
- Before creating the FortiSOAR deployment, you must configure the required DNS name for the load balancer. **Note**: Changing the DNS names after the deployment is not recommended.
- The FortiSOAR Docker image is supported only on x86_64/amd64 machine. Therefore, ensure that worker nodes on which the FortiSOAR Pods runs is x86_64/amd6
- Ensure that the FortiSOAR Pod has a resource request of 16GB RAM and 8 CPU core and a limit of 32GB RAM and 8 CPU core.

FortiSOAR EKS resource requirements

FortiSOAR EKS deployment uses the same Docker image that is designed for FMG/FAZ and enterprise Docker. For more information about deploying FortiSOAR on Docker platforms such as VMware ESX or AWS, see the Deploying FortiSOAR on a Docker Platform chapter.

Storage Class

A StorageClass provides a way for administrators to describe the offered "classes" of storage. Different classes might map to quality-of-service levels, backup policies, or arbitrary policies determined by the cluster administrators. Kubernetes is agnostic about what the classes represent. This concept is sometimes called "profiles" in other storage systems. For more information, see the Storage Classes documentation.

FortiSOAR on EKS defines the following storage class:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fortisoar-storageclass-ebs
provisioner: ebs.csi.aws.com
reclaimPolicy: Retain
allowVolumeExpansion: true
volumeBindingMode: WaitForFirstConsumer
parameters:
  type: gp2
  fsType: ext4
```

- reclaimPolicy 'Retain' specifies that even if PVC is deleted, the volume is retained. The default behavior is 'delete'.
- allowVolumeExpansion 'true' specifies that the volume should be allowed to expand when the volume is full. For example, if the `/var/log` volume is full, its associated PVC should be re-sized, so that we get expanded disk space inside the pod(container). For information on expanding the PVC, see the Resizing Persistent Volumes using Kubernetes article.

Persistent Volume Claim

All Docker volumes use PVCs, which in turn use the storage class mentioned in the Storage Class section.

The following PVC's are created:

```
/data                  (fortisoar-pvc-data 15GB)
/var/lib/rabbitmq      (fortisoar-pvc-rabbitmq 15GB)
/var/lib/elasticsearch (fortisoar-pvc-elasticsearch 40GB)
/var/lib/pgsql         (fortisoar-pvc-pgsql 150GB)
/var/log               (fortisoar-pvc-log 5GB)
/home/csadmin          (fortisoar-pvc-home-csadmin 20GB)
/opt/cyops/configs     (fortisoar-pvc-configs 15GB)
```

Load Balancer (Service)

AWS EKS offers ALB, NLB, and Classic load balancers. Their comparisons are listed at: https://aws.amazon.com/elasticloadbalancing/features/.

FortiSOAR has chosen the network load balancer (NLB) based on the following rationale:

- The Classic load balancer is an older version of the load balancer, and AWS itself does not recommend its usage.
- Long-lived TCP connections are supported by NLB, which are needed to connect FortiSOAR to FSR Agent. ALB does not support long-lived TCP connections.
- Websockets and preserving Source IP addresses are supported by NLB.
- FortiSOAR creates TCP listeners on NLB for port 443 and 5671. TCP listeners forwards traffic as is to the pod (container). The other way is to create a TLS listener on load balancer, which needs a TLS certificate deployed with the load balancer. Currently, TLS listeners on the AWS load balancer do not support mTLS. Therefore, if mTLS requires to be used, then use the simple TCP listener and implement the mTLS in application itself, which has already been done in FortiSOAR.

Statefulset

- FortiSOAR's statefulset creates only one pod. Scaling a statefulset, i.e., increasing or decreasing the number of replicas, is not supported.
- FortiSOAR Docker image is only supported on x86_64/amd64 machine. Therefore, you must ensure that the worker nodes on which the FortiSOAR Pods runs is x86_64/amd64.
- FortiSOAR Pod has resource request of 16GB RAM and 8 CPU core and limit of 32GB RAM and 8 CPU core.
- The FortiSOAR container in Kubernetes runs with 'privileged: true'. In future releases, we will enable the FortiSOAR container to run Kubernetes with limited capabilities.

Deploying the FortiSOAR Docker image on an Amazon Elastic Kubernetes (EKS) cluster

1. Download the FortiSOAR Docker image from FortiCare, i.e., https://support.fortinet.com.
2. Download the EKS deployment files (.zip) from `https://github.com/fortinet-fortisoar/fortisoar-deployment-templates`
3. Upload the downloaded FortiSOAR Docker image to your ECR or any other Docker repository that is accessible from within your Kubernetes cluster. For example:
   `# sudo docker push <account-id>.dkr.ecr.<region>.amazonaws.com/fortisoar/fortisoar:7.6.6`
4. Create the namespace using the following command:
   `#sudo kubectl apply -f fortisoar-namespace.yaml`
5. Create the storage class using the following command:
   `#sudo kubectl apply -f fortisoar-storageclass.yaml`
6. Create the service using the following command:
   `#sudo kubectl apply -f fortisoar-service.yaml`
7. Create the persistent volume claim using the following command:
   `#sudo kubectl apply -f fortisoar-pvc.yaml`
8. Note the name of the load balancer created by the service:
   `#sudo kubectl get svc -n fsr -o jsonpath="{.items[0].status.loadBalancer.ingress[0].hostname}"`
9. Update the load balancer hostname placeholder with the value returned in the previous step. For example:
   `#sudo sed -i 's#@PLACEHOLDER_HOSTNAME_LOAD_BALANCER@#k8s-fsr-fsr-b6fc340816-<account-id-here>.elb.<region>.amazonaws.com#g' fortisoar-statefulset.yaml`
10. Update the repository hostname placeholder.
    Note: Even if you do not have private repository set, then you need to run the following command with the FortiSOAR public repository hostname, i.e., `repo.fortisoar.fortinet.com`:
    `#sudo sed -i 's#@PLACEHOLDER_HOSTNAME_REPO@#repo.fortisoar.fortinet.com#g' fortisoar-statefulset.yaml`
11. Update the Docker image placeholder with the Docker image path. For example:
    `#sudo sed -i 's#@PLACEHOLDER_DOCKER_IMAGE@#<account-id-here>.dkr.ecr.<region>.amazonaws.com/fortisoar/fortisoar:7.6.6#g' fortisoar-statefulset.yaml`
12. Create the statefulset using the following command:
    `#sudo kubectl apply -f fortisoar-statefulset.yaml`
13. Access the FortiSOAR UI at `https://<load-balancer-dns-name>/`.
    Note: A spinner is displayed while provisioning is in progress. Provisioning takes about 10 minutes as it runs the FortiSOAR VM Config Wizard and configures the embedded SME in its first boot.
    If the FortiSOAR Configuration Wizard fails when provisioning your instance, then a failure screen detailing the status of each configuration step is presented on the FortiSOAR UI, making it simpler to identify the issue. Before using FortiSOAR, you must use the CLI to fix any issues with the failed steps as their functioning might be hampered. However, if you choose to access FortiSOAR without rectifying the failed steps, which can cause FortiSOAR functionality to deteriorate, a **Proceed Anyway** button is provided that lets you to use the product while acknowledging the configuration failure. If the FortiSOAR UI is not accessible even after clicking **Proceed Anyway**, you can try to fix the issue using the steps mentioned in the Deploying FortiSOAR chapter.

Uninstalling FortiSOAR from the EKS cluster

When you uninstall FortiSOAR from EKS, the FortiSOAR statefulset and data are deleted from the EKS cluster. This data is lost and cannot be recovered.

1. If your storage class is with the '`Retain`' policy, by default, FortiSOAR EKS files creates the storage class with the '`Retain`' policy, then you must write down the PVs that are associated with the FortiSOAR PVCs for deletion in the Kubernetes cluster level. Use the following command to get the PVs that are associated with the FortiSOAR PVCs:
   `#sudo kubectl get pvc -n fsr # note down all the volumes`
2. Delete all resources in the namespace where FortiSOAR is deployed. The following command considers 'fsr' as the namespace, and deletes 'fsr' statefulset, service, and namespace:
   `#sudo kubectl delete namespace fsr`
3. If your storage class is with the '`Retain`' policy, then you must delete the EBS volumes on the Amazon console. You can also use the following command on the AWS CLI to delete the EBS volumes:
   `#sudo aws ec2 delete-volume --volume-id <value>`

Limitations of FortiSOAR Docker on a EKS cluster

- Scaling a statefulset, i.e., increasing or decreasing the number of replicas, is not supported
- High availability is not supported.
- Kubernetes Liveness and Readiness probes are not available for FortiSOAR.

Troubleshooting Tips

Logs and Services

- To view the first boot or subsequent boot provisioning logs:
  `#sudo kubectl exec -ti fsr-0 -c fsr -n fsr -- bash`
  `#sudo vi /var/log/cyops/extension/boot.log`
- To view the status of FortiSOAR services:
  `#sudo kubectl exec -ti fsr-0 -c fsr -n fsr -- bash`
  `#sudo csadm services --status`

How to restart a FortiSOAR Pod

To restart a FortiSOAR Pod, do the following:

1. `#sudo kubectl scale statefulset fsr -n fsr --replicas=0`
2. Ensure that no Pod is visible in statefulset 'fortisoar'
   `# sudo kubectl get pods -n fsr`

3. `# sudo kubectl scale statefulset fsr -n fsr --replicas=1`
   Note: You must specify `--repliacas=1` only. FortiSOAR does not support scaling a statefulset. Hence, a value other than '1' can cause issues.

How to resolve the issue of Elasticsearch-based recommendations not working on a FortiSOAR Container deployed on an EKS Cluster?

By default, Elasticsearch-based recommendations do not work on a FortiSOAR container deployed on an EKS cluster due to size limitations. To know more about Elasticsearch-based recommendations, see the Recommendation Engine topic in the *Application Configuration and Customization* chapter of the "Administration Guide".

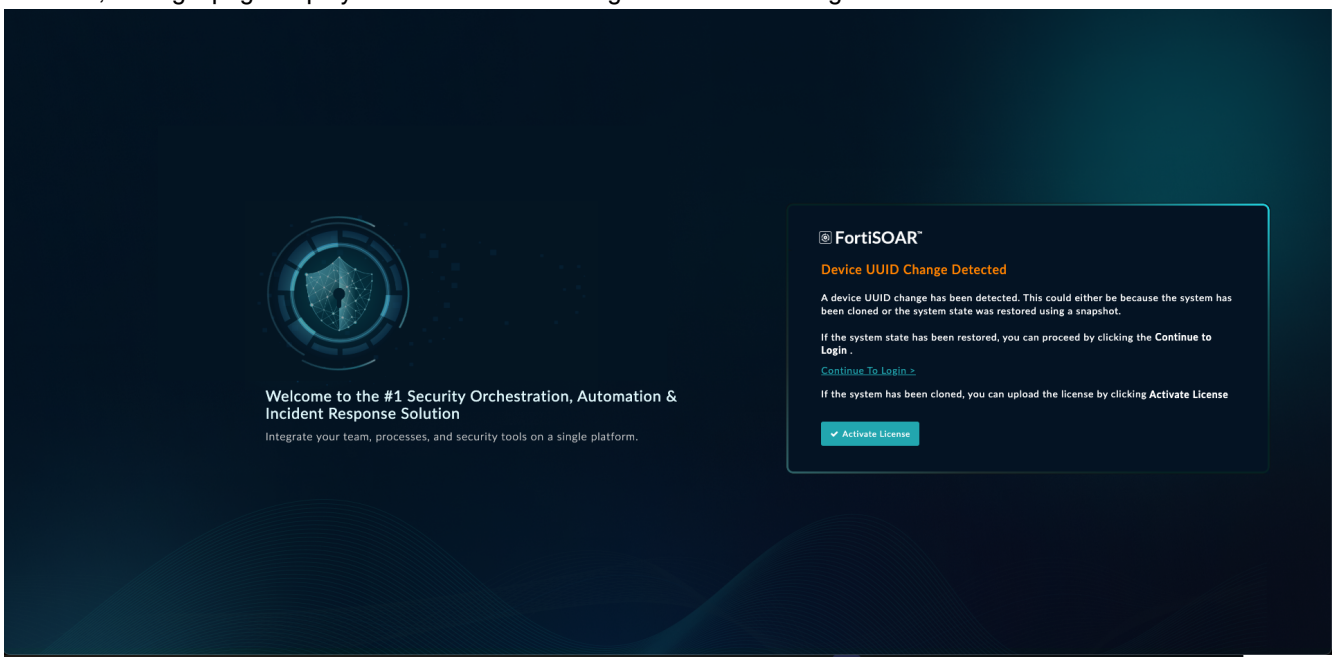
To use Elasticsearch-based recommendations, you must increase the memory allocated to Elasticsearch to 4 GB, using the following steps:

1. Update the value of the following parameters in the `fsr.options` file to 4 GB:
   ```
   sudo vi /etc/elasticsearch/jvm.options.d/fsr.options
   -Xms4g
   -Xmx4g
   ```
2. Restart the Elasticsearch service using the following command:
   ```
   sudo systemctl restart elasticsearch
   ```
3. Reindex Elasticsearch data using the following command:
   ```
   sudo -u nginx php /opt/cyops-api/bin/console app:elastic:create --sync=true
   ```
   Now, you should be able to view Elasticsearch-based recommendations on your FortiSOAR container deployed on your EKS Cluster.

The FortiSOAR login page displays a 'Device UUID Change Detected' message

If your EKS node group contains more than one node, and if the FortiSOAR pod gets scheduled on a different node than the previous node in the node group on which it was previously running, then when you try to log in to your FortiSOAR

instance, the login page displays a 'Device UUID Change Detected' message:



Resolution:

Click **Continue to Login** on the FortiSOAR login page. This screen continues to be visible till an FDN sync is performed after you have clicked **Continue to Login**. By default, the FDN synchronization occurs hourly.
If you want to get rid of this screen, then you can run the following two commands on the FortiSOAR CLI:

1. `sudo csadm license --refresh-device-uuid`
2. `sudo systemctl restart cyops-auth`

Deploying an FSR Agent on a Docker Platform

Starting with release 7.6.4, FSR Agents can be installed within a Docker container, enabling seamless integration and automation in containerized environments. This feature is especially useful for large or advanced organizations that prefer Docker-based deployments over traditional virtual machines.

> It is highly recommended to install only a single agent per Docker instance, unless the host is connected to multiple networks.

Prerequisites

- To deploy an FSR Agent on Docker, ensure that Docker is installed and running in your environment:
  To install Docker on RHEL, refer to https://docs.docker.com/engine/install/rhel/
  To verify the installation, run: `docker compose version`
- Before proceeding, review the Deploying FSR Agents topic of the *Deploying FortiSOAR* chapter for detailed information, including:
  - Overview of FSR Agents
  - **Specifications**: Recommended specifications for Secure Message Exchange and recommended specifications mentioned for FSR Agent. The docker host specifications must match the recommended specifications mentioned for FSR Agent.
  - **Prerequisites**: Fulfill all requirements in the Prerequisites for installing an FSR agent topic.
  - Minimal Permissions.

Process of setting up an FSR Agent

1. Enabling and Adding a Secure Message Exchange (SME)
2. Adding an FSR Agent
   Important: When adding the agent, select **Docker Compose** from the **Installer Type** drop-down menu.
3. Installing an FSR Agent on a Docker Platform

Installing an FSR Agent on a Docker Platform

Once the connection between the FortiSOAR docker host and SME is established, the `Status` column on the `Agents` page, displays "**Awaiting Remote Node Connection**" along with a **Download Installer** link. Next, you need to install the FSR Agent as follows:

1. Log on to your base FortiSOAR node as an administrator and click the **Settings** icon to open the `System` page.
2. On the `System` page, you will see the `Agent Configurations` section. Click the **Agent** item in the left menu, to open the `Agents` page.
3. Click the **Download installer** link on the `Agents` page.
4. On the `Prepare and Download Agent Installer` dialog, select the connectors to include during installation from the **Include Specified Connectors to Install on Agent** drop-down list. FSR Agents use these connectors to remotely run actions to remotely execute actions on external networks. Options include:
   - **Do not install connectors by default**
   - **Include pre-existing connectors on agent**
   - **Custom**
   - **All connectors installed on Master**
     For details on these options, see Installing an FSR Agent topic in the *Deploying FortiSOAR* chapter
5. Click **Download Installer**.
6. An installer file (`<agent-name>-install.bin`) is downloaded on your docker. Transfer it to the Docker host using `scp` or another method.
7. On the Docker host, run the following command to install the agent:
   ```
   sudo chmod +x <agent-name>-install.bin
   sudo ./<agent-name>-install.bin
   ```
   Once you enter the command, the installer performs Pre-Installation Checks, and once they are successful, the installer creates the required Docker volume and network and starts the container.

> Installation time depends on the number of connectors and available bandwidth. You can connect to the Docker container and monitor the installation logs using the following commands:
>
> 1. To get the Container ID of the Docker agent:
>   `docker ps`
> 2. Connect to the Docker Container:
>   `docker exec –it < Container ID> bash`
> 3. Monitor the installation logs at:
>   `/var/logs/cyops/<agent-boot-currenttimestamp>.log`.

Pre-Installation Checks

The installer performs the following checks before starting:

- Docker is installed and running.
- Sufficient CPU and memory are available.
- The Docker volume (`/var/lib/docker/volumes`) has enough disk space.
- Entries from `/etc/hosts` are added for domain name resolution based on the Docker host.

Post-Installation

Once the installation completes:

- The `Status` column on the `Agents` page shows "Remote Node Connected".
- Click **Enable Input Bridge** to configure the FSR Agent Communication Bridge. Before configuring the bridge, ensure you have added port mapping for the Agent in the Docker container. For steps, see the Adding Port Mapping for FortiSOAR Agent in Docker topic.
  For details, see the FSR Agent Communication Bridge connector documentation.
- Click **Update Config** to apply the latest settings, such as changes to the hostname, SNI, or address, to the Agent machine.

After successful deployment on Docker, you can

- Run connector actions on the FSR Agent.
- Install and configure connectors.
  For more information, see the FortiSOAR Agent Setup and Configuration chapter in the "Administration Guide".

Adding Port Mapping for FortiSOAR Agent in Docker

To find the name of the FortiSOAR Agent, its configuration file path, and other details, run:
`docker compose ls`

1. Edit the '`docker compose`' file:
   Open the '`docker compose`' file for the FortiSOAR Agent deployed on Docker. Under the `service` section, add the following `ports` mapping:
   ```
   ports:
    - "PORT:PORT"
   ```
   The first `PORT` is the port on the host machine. The second `PORT`, is the Port on which the Agent inside the Docker container listens for incoming connections. Value of both ports must be the same. For example,
   ```
   ports:
    - "8443:8443"
   ```
   Note: Ensure the `PORT` you choose is not already in use by another application on your host machine.
2. To stop the existing Agent container, run the following command:
   `docker compose --project-name <Agent Name> -f <path of config file> down`
3. To start the Agent container with the new port mapping, run the following command:
   `docker compose --project-name <Agent Name> -f <path of config file> up --detach`
4. Check if the port is correctly mapped to the container by running the following command:
   `docker ps | grep "<agent-name>"`
5. Open the port in the firewall (if necessary):
   a. Ensure the specified port is open in the firewall of the Docker host. To open the port, run the following command:
      `sudo firewall-cmd --add-port=<port>/tcp --permanent`
   b. Reload the firewall with the following command:
      `sudo firewall-cmd --reload`

Upgrading FSR Agent on Docker

To upgrade an FSR Agent on your Docker host, follow these steps:

1. Log in to your base FortiSOAR node as an administrator and click the **Settings** icon to open the `System` page.
2. On the `System` page, locate the `Agent Configurations` section. From the left menu, click **Agents** to open the `Agents` page.
3. On the `Agents` page, in the `Agent Actions` column, find the FSR Agent you want to upgrade. Click the **Upgrade** link to open the `Agent Upgrade` dialog.
4. In the `Agent Upgrade` dialog, click **Download Agent Installer** to download the upgrade file, which will be named `<agent-name>-install.bin`.
5. Copy the downloaded `<agent-name>-install.bin` file to the Docker host VM where the FSR Agent Docker container is running.
6. SSH to the Docker host VM and run the following commands:
   a. Run the following command to stop the existing FSR Agent Docker container:
   `docker compose --project-name <Agent Name> -f <path of config file> down`
   Note: To find the Agent Name, configuration file path, and other details, run:
   `docker compose ls`
   b. Run the `<agent-name>-install.bin` file that you downloaded and copied to the Docker host VM to complete the upgrade.

Deboarding FSR Agent on Docker

> ⚠ To deboard existing FSR agents, you must have `Read` and `Delete` Permissions on the `Agents` module. Deboarding an agent deletes the FSR agent, removes all installed connectors, and deletes their configurations from the base FortiSOAR node. This action is irreversible–once an FSR agent is deleted, its information cannot be recovered. Proceed with caution.

1. Log in to your Docker Host VM.
2. To view the details of the FSR Agents installed on Docker, run:
   `docker compose ls`
3. From the output of the `docker compose ls` command, note the *Agent Name* and *configuration file path* of the Agent you want to remove.
4. To stop and remove the Docker container for the agent, run:
   `docker compose --project-name <Agent Name> -f <path of config file> down`
   If you also want to remove associated Docker volumes, run:
   `docker compose --project-name <Agent Name> -f <path of config file> down –volumes`
5. To verify that the Docker container for the agent is removed, run:
   `docker ps -a`
6. Log in to your base FortiSOAR node as an administrator. Navigate to **Settings** > **Agents** to open the `Agents` page.
7. On the `Agents` page, select the agent you want to remove and click **Delete**.
8. FortiSOAR displays a warning dialog, click **Confirm** to deboard the FSR agent.

Licensing FortiSOAR

FortiSOAR integrates with FortiGuard Distribution Network (FDN) to retrieve updated contract details.

> ⚠ You need to be connected to FDN while you are deploying your license. If there is no connectivity to FDN, your FortiSOAR UI access will be blocked after the 96-hour grace period for license verification ends.If you encounter any errors during license deployment, see the Troubleshooting licensing issues section for tips in resolving the issue.

FortiSOAR enforces licensing and restricts the usage of FortiSOAR by specifying the following:

- The maximum number of active users in FortiSOAR at any point in time.
- The type and edition of the license.
- The expiration date of the license.

For a fresh install of FortiSOAR, see FortiSOAR licensing process. To retrieve your Device UUID, see Retrieving the FortiSOAR Device UUID.

FortiSOAR licensing process

1. You must have an account in FortiCare.
2. Contact FortiSOAR Support to obtain FortiSOAR product SKU. You will require to provide the following information to be able to get the license for FortiSOAR™:
   - The license type that you want for FortiSOAR. For information on the different license types, see License Manager Page.
   - The license edition that you want for FortiSOAR. For information on the different license editions, see License Manager Page.
   - The number of licensed users required for FortiSOAR.
     Once you complete purchasing FortiSOAR, you will be sent a service contract registration code to your registered email address.
     If a customer wants additional users, then the customer has to also register the contract for additional users. A separate registration code will be sent for the contract of additional users.
     Note: If you have opted for a "Perpetual" or "Evaluation" license, you should download the license file only after the additional user contract, if any, is registered. You can get a trial license for FortiSOAR through your FortiCare account; for details see Activating the FortiCare Trial License for FortiSOAR. You can also use FortiFlex licensing for FortiSOAR; for details see Activating FortiSOAR license using FortiFlex.
3. Login to your FortiCare account and click **Asset** > **Register/Activate** to register your FortiSOAR product. You can register your FortiSOAR product using the instructions provided in the FortiCare registration wizard.
   You will require to copy-paste the service contract registration code from your email to register FortiSOAR.
   Once you have verified the registration, click **Complete** to complete the registration.
4. Once you click **Complete** you are taken to the **Product Information** page. To generate the license file, click **Edit** on the `Product Information` page.
   On the `Edit Product Information` page, in the **UUID** field, enter the Device UUID of your FortiSOAR installation and click **Save**.
   Important: The license issued against one device UUID can later be used on another FortiSOAR virtual machine

with a difference device UUID, as well in case of disaster recovery (DR). However, the same license cannot be active simultaneously on more than one node.
To retrieve your Device UUID, see Retrieving the FortiSOAR Device UUID.
The license file is generated after you enter the Device UUID. You can now download and deploy the FortiSOAR license, using the steps mentioned in Deploying the FortiSOAR license.

If you are an existing customer, then your entitlements would have already been imported into FortiCare and you would have received an email with respect to your FortiCare account. Also, your FortiSOAR product would already have been registered. However, you do require to update your Device UUID.

To update your Device UUID, do the following:

1. Login to your FortiCare account and click **Asset** > **Manage/View Products** > **Basic View**.
2. Click the row that contains the FortiSOAR (FSR) product to view the `Product Information` page.
3. On the `Edit Product Information` page, in the **UUID** field, enter the Device UUID of your FortiSOAR installation and click **Save**.
   Important: The license issued against one device UUID can later be used on another FortiSOAR virtual machine with a difference device UUID, as well in case of disaster recovery (DR). However, the same license cannot be active simultaneously on more than one node.
   To retrieve your Device UUID, see Retrieving the FortiSOAR Device UUID.
   The license file is generated after you enter the Device UUID. You can now download and deploy the FortiSOAR license, using the steps mentioned in Deploying the FortiSOAR license.

FortiSOAR licensing using FortiManager

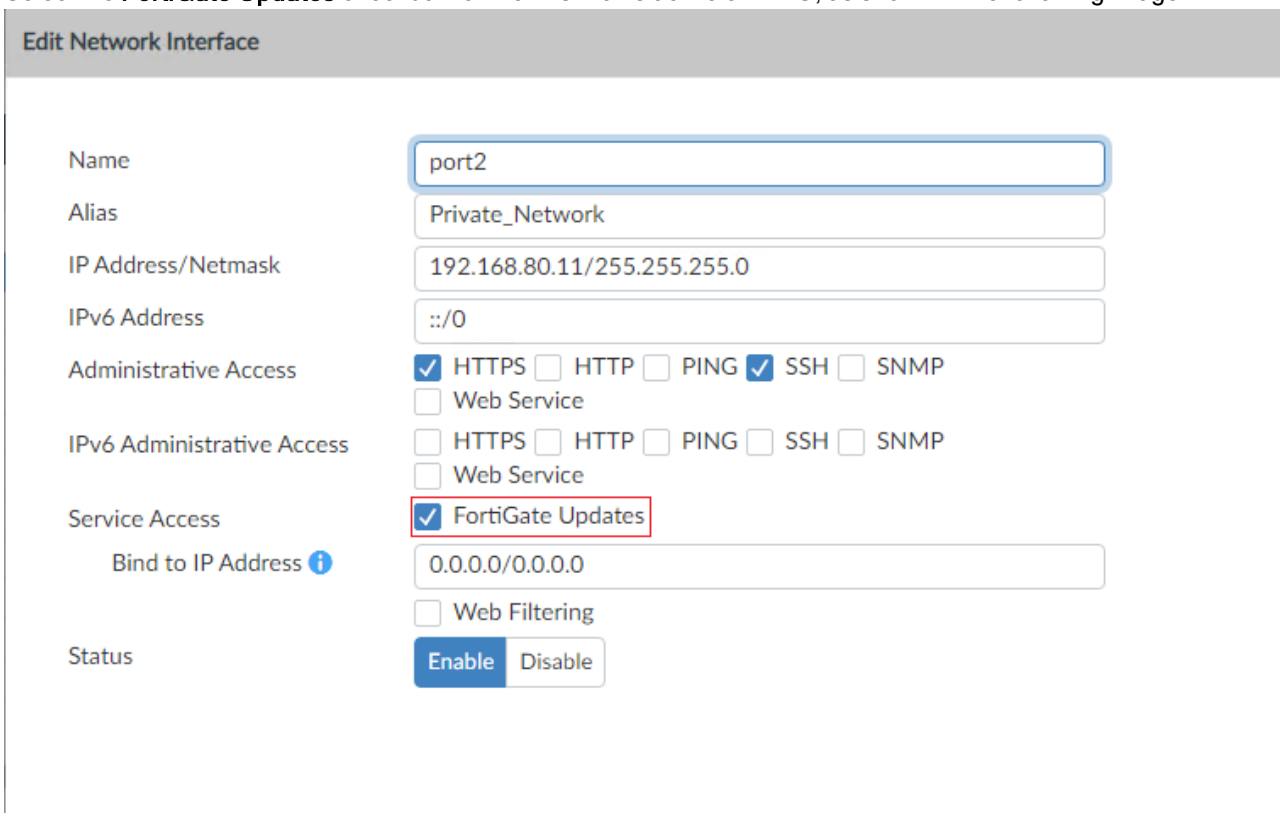
A closed or air-gapped environment is an environment where FortiSOAR does not have access to the internet and therefore cannot access the FDN servers. In such cases, FortiManager (FMG) can be used as an intermediary so that FMG provides license validation and FDN updates to FortiSOAR with limited or no internet connectivity. You can configure FMG for the following environments:

- Complete air-gapped environment where FMG also does not have connectivity to FortiGuard Distribution Servers (FDS) and manual synchronization is required for customer entitlements.
- FMG has network connectivity to FDS servers and can automatically synchronize customer entitlements.
  For more details on FMG and troubleshooting information, see the FortiManager documentation.

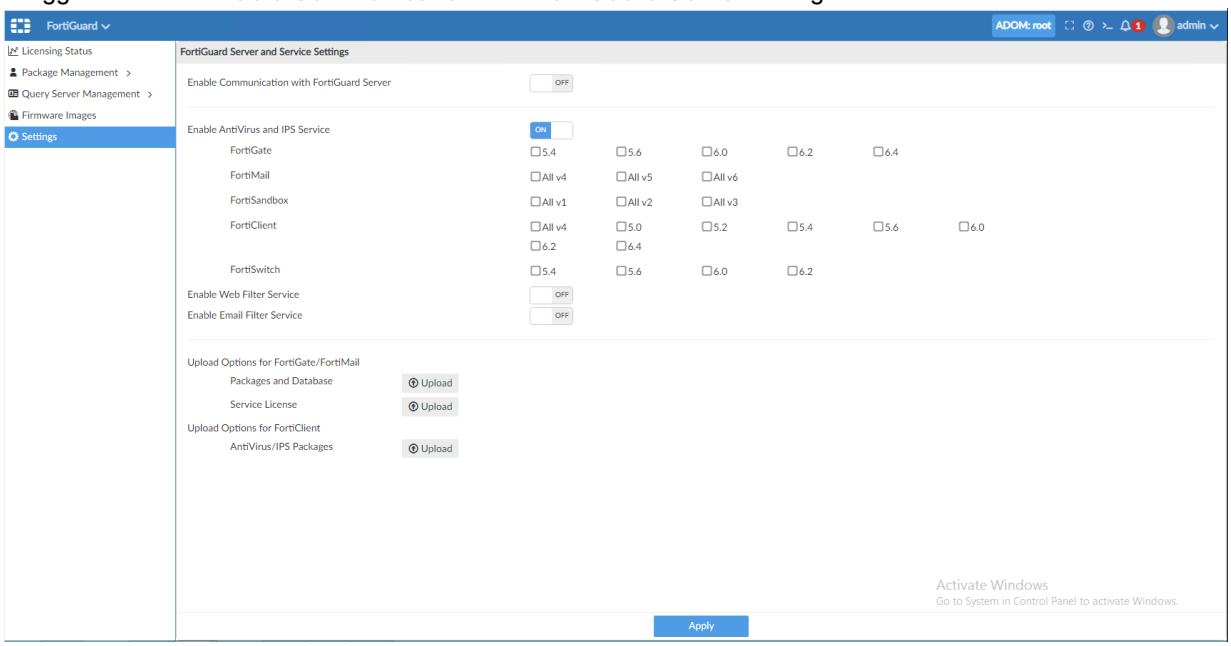
Process to deploy the FortiSOAR license when you are in a complete air-gapped environment

1. You must have an account in FortiManager (FMG).
2. Contact FortiSOAR Customer Support to obtain an entitlement file, which contains all the contract details.
3. Log onto FMG and navigate to FortiGuard.

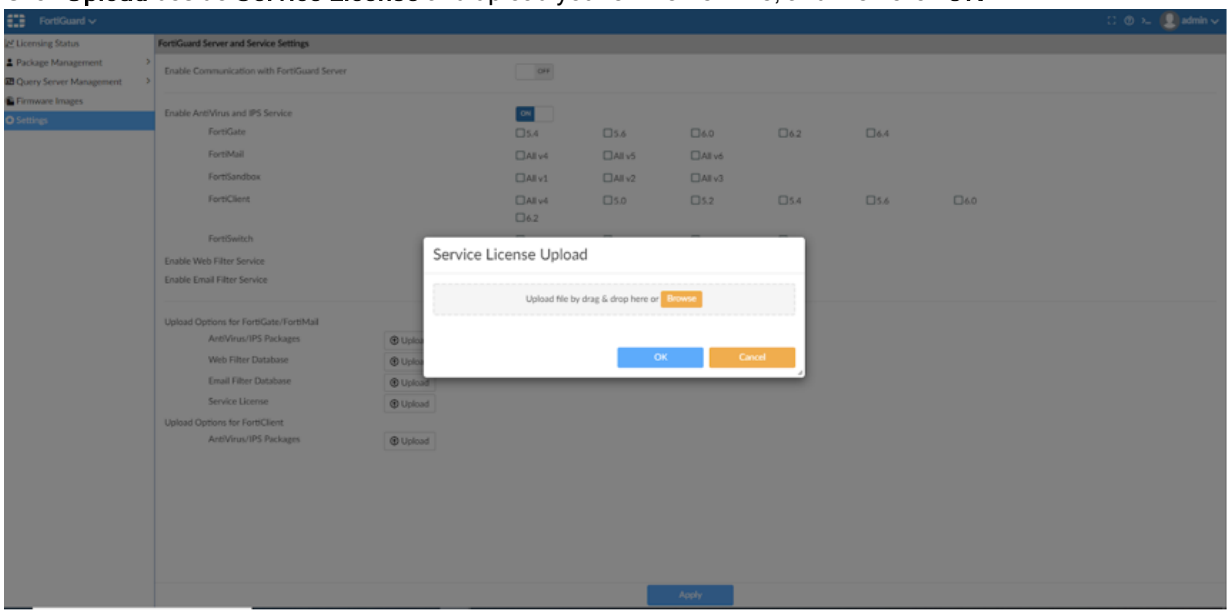
4. Select the **FortiGate Updates** checkbox for the NIC that is active on FMG, as shown in the following image:



5. On the left-menu, click **Settings**, and apply the following settings:
   a. "Toggle OFF" the **Enable Communication with FortiGuard Server** setting.



b. Click **Upload** beside **Service License** and upload your entitlement file, and then click **OK**.



c. Click **Apply** to apply the above settings.

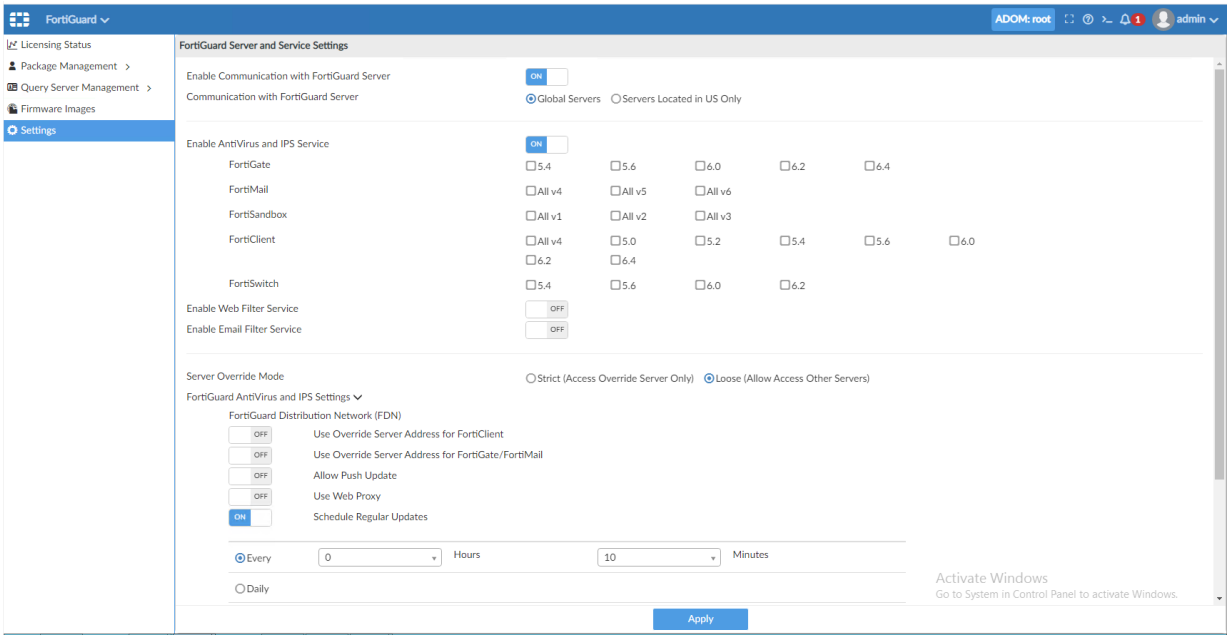
6. Ensure that FMG is reachable or resolvable from your FortiSOAR instance.
7. Modify your FortiSOAR config to connect to FMG by adding the following entry in the `das.ini` file:
   ```
   sudo vi /opt/cyops-auth/utilities/das.ini
   [FDN]
   host = https://<FMG Hostname>:8890
   ```
8. Restart the `cyops-auth` service:
   ```
   sudo csadm services --restart cyops-auth
   ```
9. Deploy your FortiSOAR license using the steps mentioned in Deploying the FortiSOAR license.

Process to deploy the FortiSOAR license when you are not in a complete air-gapped environment

You might choose to deploy the license using FMG even if you are not in an air-gapped environment. In such cases do the following:

1. You must have an account in FortiManager (FMG).
2. Contact FortiSOAR Customer Support to obtain an entitlement file, which contains all the contract details.
3. Log onto FMG and navigate to FortiGuard.
4. On the left-menu, click **Settings**, and apply the following settings:
   a. "Toggle ON" the **Enable Communication with FortiGuard Server** setting.
   b. For the **Communication with FortiGuard Server** settings, select **Global Servers**.
   c. For the **Server Override Mode** settings, select **Loose (Allow Access Other Servers)**.
   d. Expand "FortiGuard AntiVirus and IPS Setting", and "Turn ON" the **Schedule Regular Updates** setting. Once you turn on the Schedule Regular Updates settings, you need to define the frequency at which you want

to get the updates:



e. Click **Apply** to apply the above settings.

5. Ensure that FMG is reachable or resolvable from your FortiSOAR instance and ensure that FMG has access to the Internet.
6. Modify your FortiSOAR config to connect to FMG by adding the following entry in the `das.ini`:
   `sudo /opt/cyops-auth/utilities/das.ini` file:
   ```
   [FDN]
   host = https://<FMG Hostname>:8890
   ```
7. Restart the `cyops-auth` service:
   ```
   sudo csadm services --restart cyops-auth
   ```
8. Deploy your FortiSOAR license using the steps mentioned in Deploying the FortiSOAR license.
   Important: In case of a non-closed environment, license deployment from FortiSOAR does not work at the first attempt since FMG is unable to send contracts that are required for license deployment. Therefore, users need to retry deploying the license on the FortiSOAR environment. This happens only when FMG is not a part of the air-gapped environment.

Retrieving the FortiSOAR Device UUID

Your FortiSOAR installation generates a Device UUID for your installation. This key is used to identify each unique FortiSOAR environment.

When you provision a new instance, a configuration wizard runs automatically on the first `ssh` login by the `csadmin` user. This wizard automatically generates your Device UUID and saves the Device UUID in the `/home/csadmin/device_uuid` file from which you can retrieve your device UUID. For more information, see the *FortiSOAR Configuration Wizard* topic. However, if you require the device UUID in the future, you can use the FortiSOAR Admin CLI (`sudo csadm`) or from the see License Manager Page.

Users can retrieve the FortiSOAR Device UUID using the `sudo csadm license --get-device-uuid` command to print the Device UUID on the CLI. For more information on the FortiSOAR Admin CLI, see the CLI Administration chapter in the "Administration Guide."

Deploying the FortiSOAR license

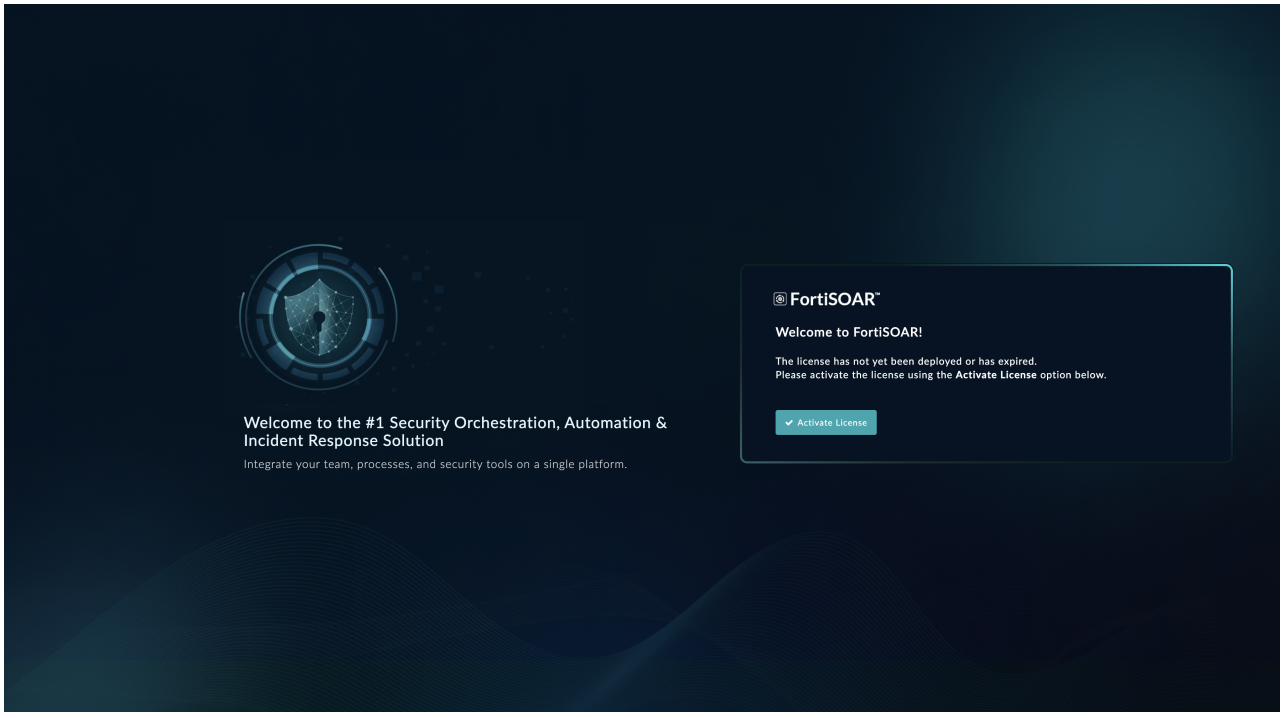
⚠ Before deploying your FortiSOAR license, ensure that the system can connect to https://globalupdate.fortinet.net. License deployment will fail without connectivity to this address, which is required to retrieve license entitlements and ensure proper product functionality after an upgrade.

Deploying the FortiSOAR license using the FortiSOAR UI

You can deploy your FortiSOAR license directly from the FortiSOAR UI without needing SSH access to your FortiSOAR machine. This process is especially useful when administrators do not have SSH access..

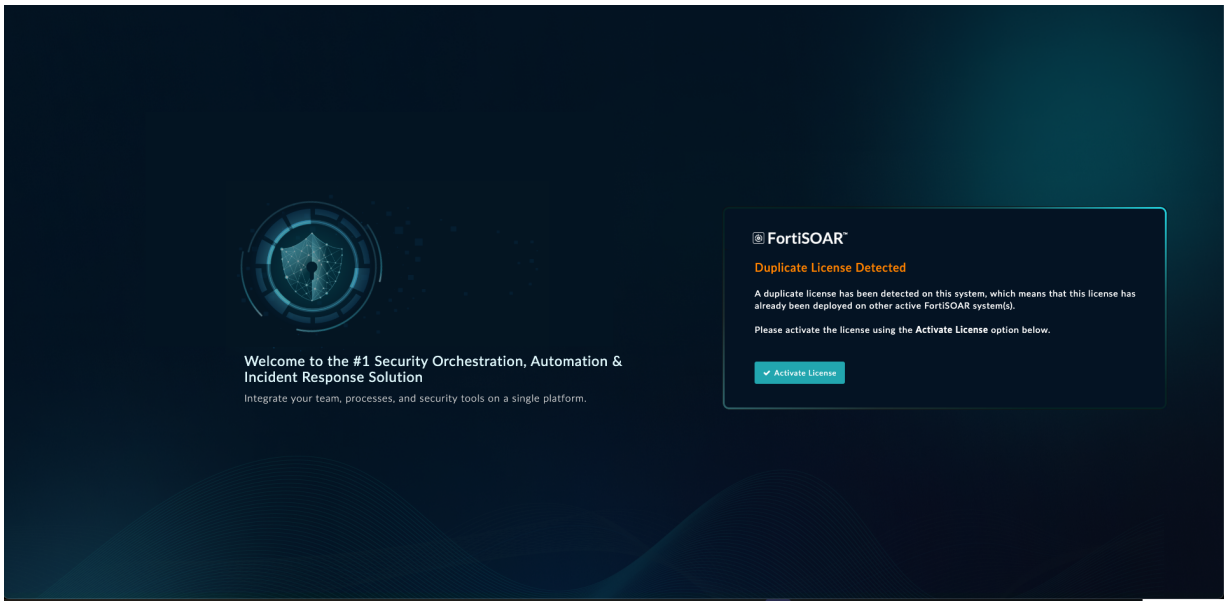
Follow the steps below to deploy the initial FortiSOAR license or upload a new one if your current license has expired:

1. In the browser go to `https://<YourFortisoarHostname>/login` to open the FortiSOAR login screen in the case of a fresh installation:

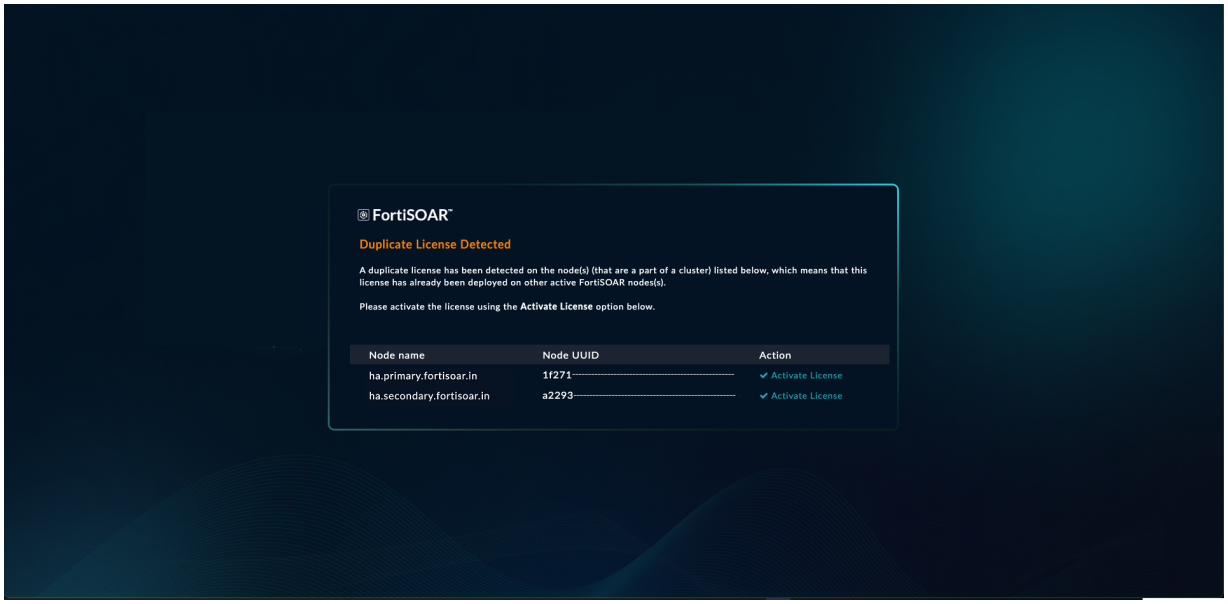


Note: On the `Activate License` screen, you may encounter one of the following scenarios:

- **Duplicate License Detection**: If FortiSOAR detects a duplicate license (i.e., the same license is already deployed on another active node), click **Activate License** to upload a new license on one of the nodes:

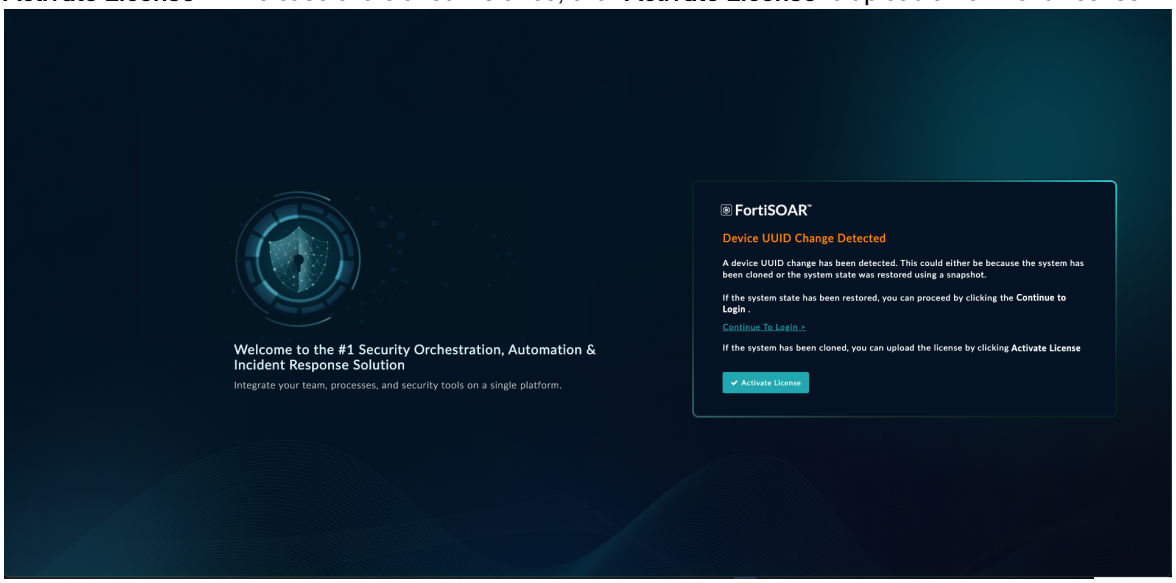


If FortiSOAR detects that a duplicate license has been deployed in an HA cluster, i.e., the same license has already been deployed on another active FortiSOAR node in the HA cluster, then you can click **Activate License** in the row of any of the nodes in the HA cluster as shown in the following screen to upload a new license on one of the two nodes:

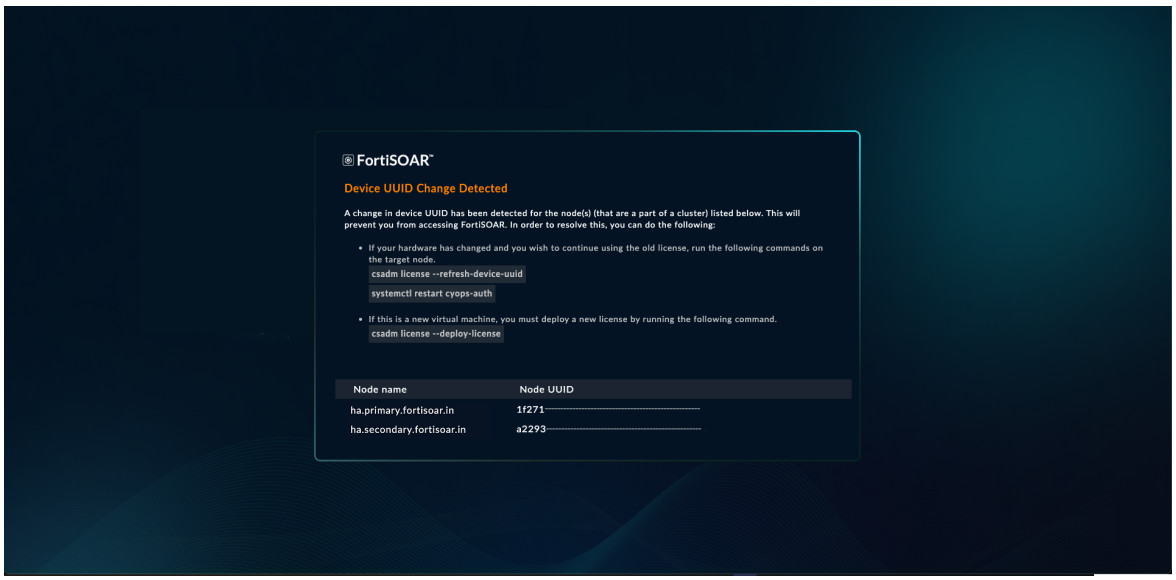


- **Device UUID Change**: If FortiSOAR detects a 'Device UUID change', generally due to restoring a snapshot of a FortiSOAR instance, or cloning of a FortiSOAR instance, you will be given the option to either:
  - **Continue to Login**: In case a snapshot is restored on the instance, you can continue to log in by clicking **Continue to Login**.

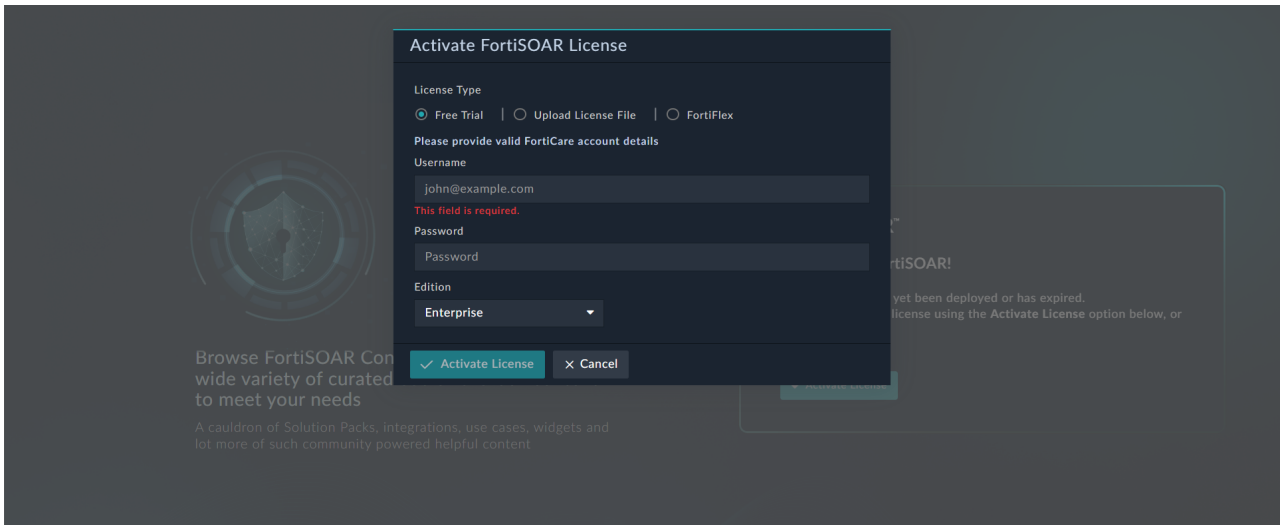
- **Activate License**: In the case of a cloned instance, click **Activate License** to upload a new valid license:



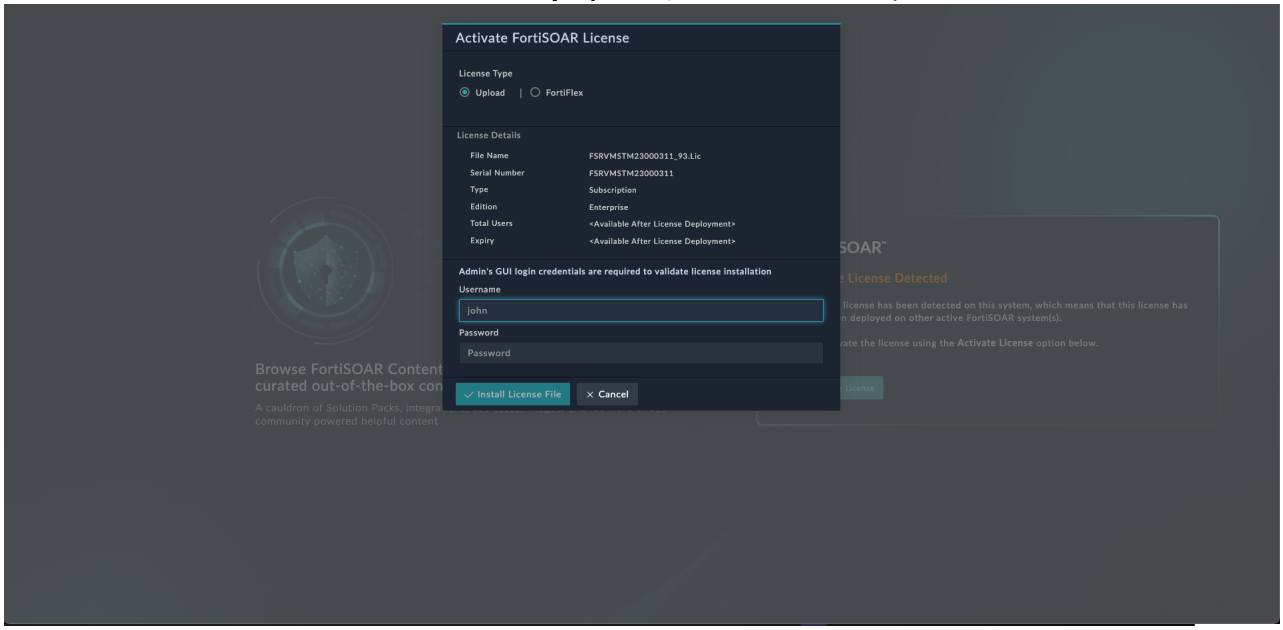
- **Device UUID Change in case of an HA cluster**: If FortiSOAR detects a Device UUID change in a high-availability (HA) cluster, the nodes with the UUID change will be listed. You can:
  - **Run the `sudo csadm license --refresh-device-uuid` command**: In the case of a hardware change, and if you want to continue using the old license, you can run the `sudo csadm license --refresh-device-uuid` command on the specific node of the HA cluster, and then continue to log in to the system.
  - **Run the `sudo csadm license --deploy-license` command**: In the case of new virtual machine, you can run the `sudo csadm license --deploy-license` to deploy the new valid license for the specific node of the HA cluster:



2. Click **Activate License** to display the following '`Activate FortiSOAR License`' dialog, if you are deploying the license for the first time:

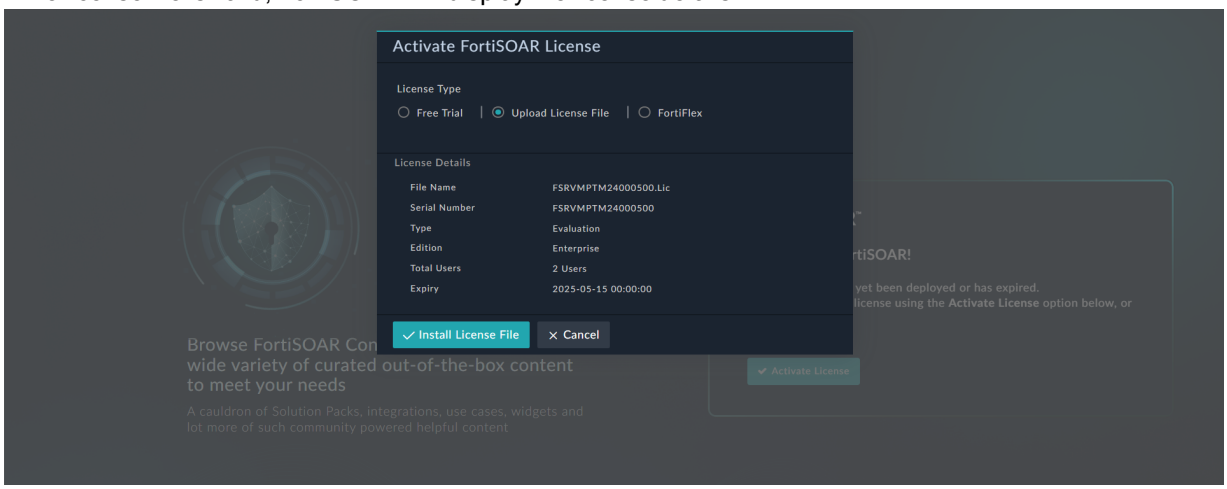


NOTE: In case you deploying a new license after the expiration of your FortiSOAR license, in the case of duplicate license detection, or while deploying a new license for a new virtual machine, you must also provide valid FortiSOAR administrator credentials with '**Security Update**' permissions, before you can install the license:



3. Choose the appropriate license type from the available **License Type** options:
   - **Free Trial**: Select this option to activate your FortiSOAR free trial. For details, see the Activating the FortiCare Trial License for FortiSOAR topic.
   - **Upload License File**: Select this option to upload your FortiSOAR license. This topic describes the steps for uploading a FortiSOAR license.
   - **FortiFlex**: Select this option to activate your license using FortiFlex. For details, see the Activating FortiSOAR license using FortiFlex topic.
4. Select **Upload License File** as the **License Type**, and then either drag and drop your license file or click the **Upload** icon to browse and select the license file.

- If the license file is invalid, an error message will appear, and the license will not be installed.
- If the license file is valid, FortiSOAR will display the license details:



5. Click **Install License File** to install your FortiSOAR license.
   Once the license is successfully installed, FortiSOAR displays a `License imported successfully` message and the EULA is displayed.
   Accept the EULA to complete the installation and proceed to log in to the FortiSOAR UI for system configuration.

Deploying the FortiSOAR license using the FortiSOAR Admin CLI

> Ensure that you have copied the FortiSOAR license file, using SCP or other methods, to your FortiSOAR VM. **Do not copy** the contents of the license file and paste it into a new file; this will cause license validation to fail.

You can deploy the FortiSOAR license using the `sudo csadm license --deploy-license <License File Path>` command. For example, `sudo csadm license --deploy-license temp/<Serial_No>.lic`. Your license file specifies the FortiSOAR edition: 'Enterprise', 'HA' or 'Multi-Tenant'. For details on editions, see the License Manager Page section, and for more information on `csadm`, see the CLI Administration chapter in the "Administration Guide."

The license path that you provide can either be relative to the current working directory or can be an absolute path. Once you have entered the license path, the `sudo csadm` checks the license file for validity and whether you have selected the appropriate license type (enabled or not enabled for multi-tenancy).

When you deploy a license on FortiSOAR the license entitlements are fetched from FDN.

NOTE: If you deploy a license that does not match the system UUID, you will receive a warning on the CLI during deployment. If you deploy the same license in multiple environments, it will be detected as a duplicate and you will be required to correct the license. Otherwise, your FortiSOAR UI will be blocked as follows:

- Once a license is identified as a duplicate, it is checked using FDN sync every hour to verify if the duplicate license issue has been resolved.
- After two such warnings, with a one-hour gap between them, the FortiSOAR UI will be blocked for users.
- The FortiSOAR UI will remain blocked until a new license is deployed or the system with the duplicate license is shut down.

The FortiSOAR Admin CLI displays a `Success` message, if your license file is deployed successfully, or an `Error` message that contains the reason for the failure.

Once your system is licensed, you can log on to the FortiSOAR UI and begin configuring the system.

Activating the FortiCare Trial License for FortiSOAR

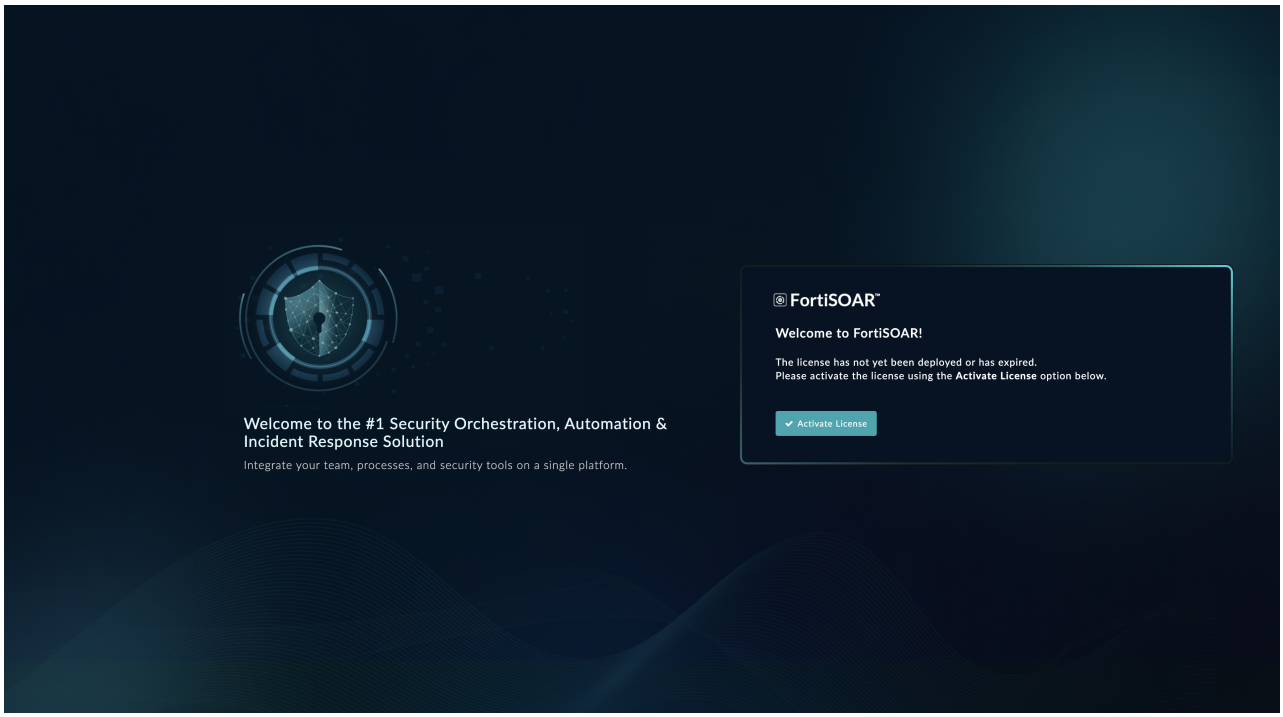
You can get a free trial license for FortiSOAR through your FortiCare account. This trial license has unlimited duration but comes with certain limitations, such as restrictions on the number of users and actions that can be performed in FortiSOAR per day. From release 7.6.0, the default trial license edition restricts FortiSOAR usage to 2 users for a maximum of 1000 actions per day, and users can choose the edition, either "Enterprise" or "Multi-Tenant" they want to deploy. Prior to release 7.6.0, the trial license was of type "Enterprise" and which restricted FortiSOAR usage to 3 users for a maximum of 200 actions a day. Due to this change in behavior, existing users should take note of the following:

- Users who had activated their trial license previously will retain their current action count of 200. If after upgrading you do not see the new action count of 1000, you should contact Fortinet Support to clear your existing trial license from your FortiSOAR system and then reactivate the trial license to get the updated action count of 1000.
- Users with an active trial license will remain on the "Enterprise" edition. If they want to change the edition to "Multi-Tenant (Manager)" they must contact Fortinet Support to decommission their current trial and re-register for a new trial license with the "Multi-Tenant (Manager)" edition.

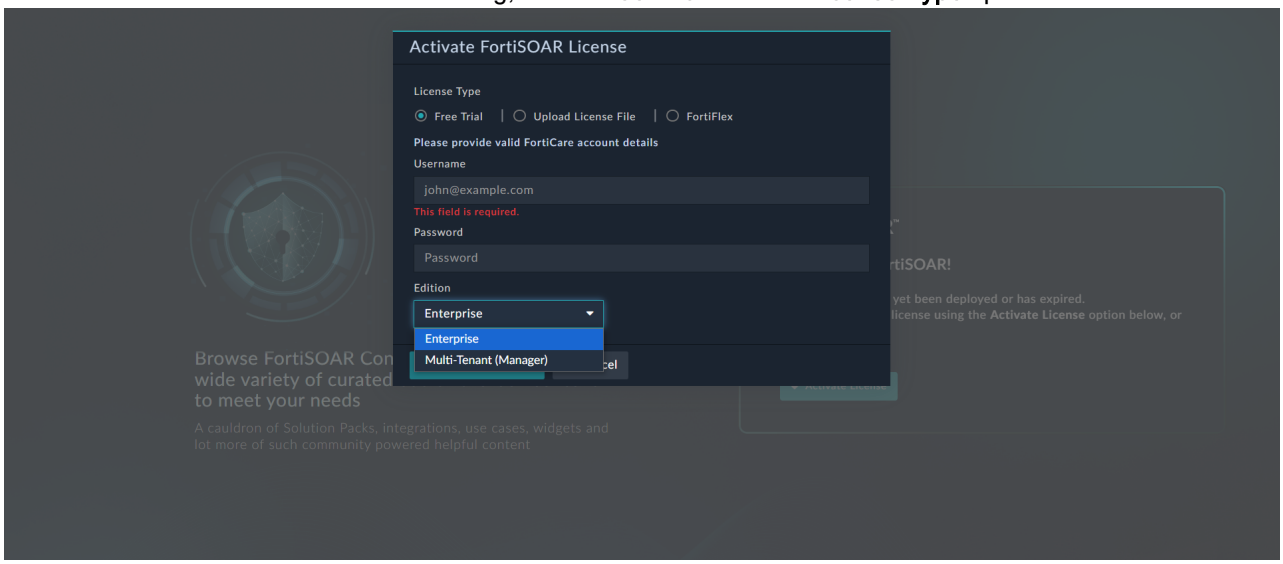
> Important steps such as "Create Records", "Update Records", "Connector Actions", "Set Variable", etc., are counted towards the maximum action count limit of 1000. However, steps used for data manipulation such as "Wait", "Approval", "Loops", "Reference a Playbook", etc. are not counted towards the action count restriction.

To activate the FortiCare trial license for FortiSOAR, do the following:

1. In the browser type `https://<YourFortisoarHostname>/login` to open your FortiSOAR UI. This will display the following screen:



2. Click **Activate License**.
3. In the `Activate FortiSOAR License` dialog, select **Free Trial** from the **License Type** options:



4. Enter your FortiCare username (email address) and password, and choose "Enterprise" or Multi-Tenant (Manager) as the edition you want to deploy on your instance. Selecting Multi-Tenant will designate the instance as the "manager" node.
   Once you have completed entering the details, click **Activate License**.
   If the email address and password provided are correct, then your FortiCare trial license for FortiSOAR is activated.

You can upgrade this trial license to a full production license at any time by purchasing a FortiSOAR license and updating it either the FortiSOAR CLI or UI.

Activating FortiSOAR license using FortiFlex

Starting with release 7.6.1, FortiSOAR integrates with FortiFlex, allowing customers to utilize FortiFlex's usage-based licensing. This system provides a simple, transparent, points-based approach, giving organizations the flexibility to optimize their cybersecurity services and spending. FortiFlex simplifies deployment decisions by enabling dynamic scaling without the need to pre-determine exact services.

You can manage entitlements, license seats, expirations, configurations, and generate licensing tokens through the FortiFlex portal. Additionally, you can monitor your point usage (FortiPoints) in the portal, assisting you in tracking costs and billings.

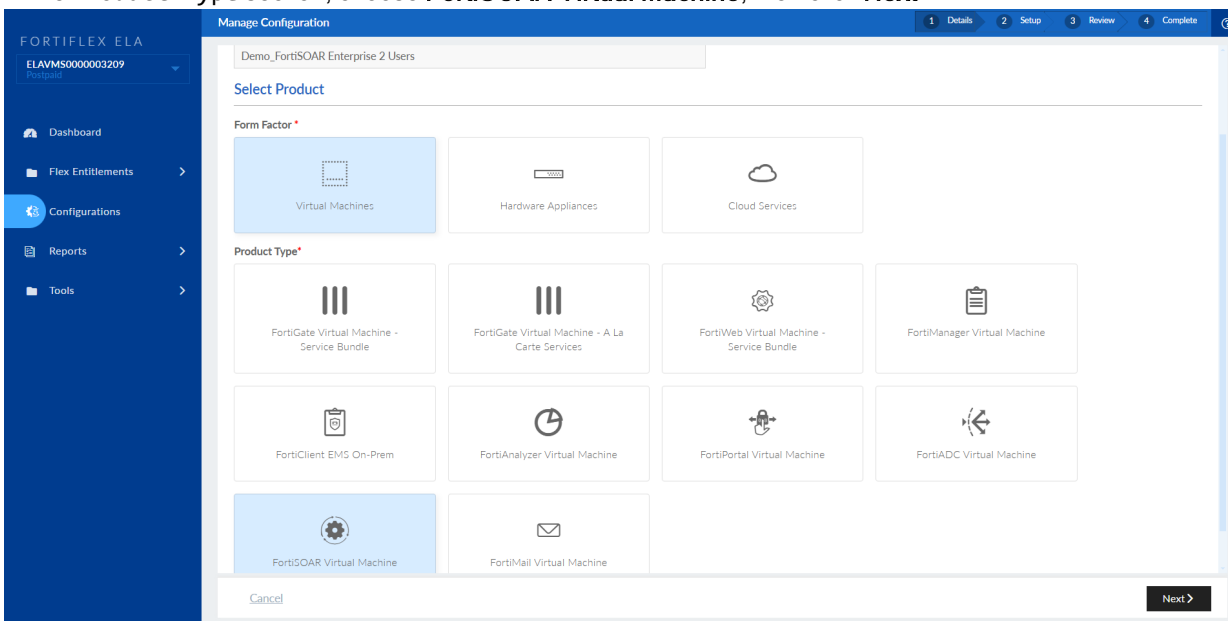
FortiFlex supports licensing for on-premises FortiSOAR Virtual Machines.

The following steps outline how to activate a FortiSOAR license using a FortiFlex token. For detailed instructions, refer to the corresponding topics:

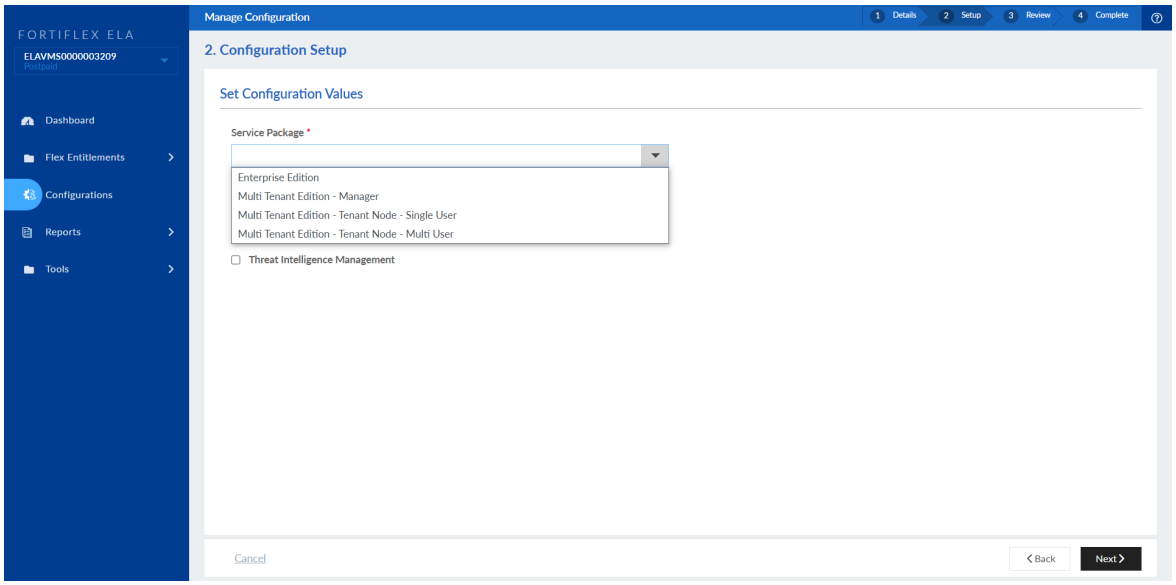
1. Log into your FortiCloud account and register a contract for the FortiFlex program. For FortiFlex ordering information, see the FortiFlex Ordering Guide.
2. Create a configuration for FortiSOAR in the FortiFlex portal.
3. Create an entitlement for FortiSOAR in the FortiFlex portal.
4. Activate your FortiSOAR license using the FortiFlex token.

Creating a FortiSOAR configuration in FortiFlex

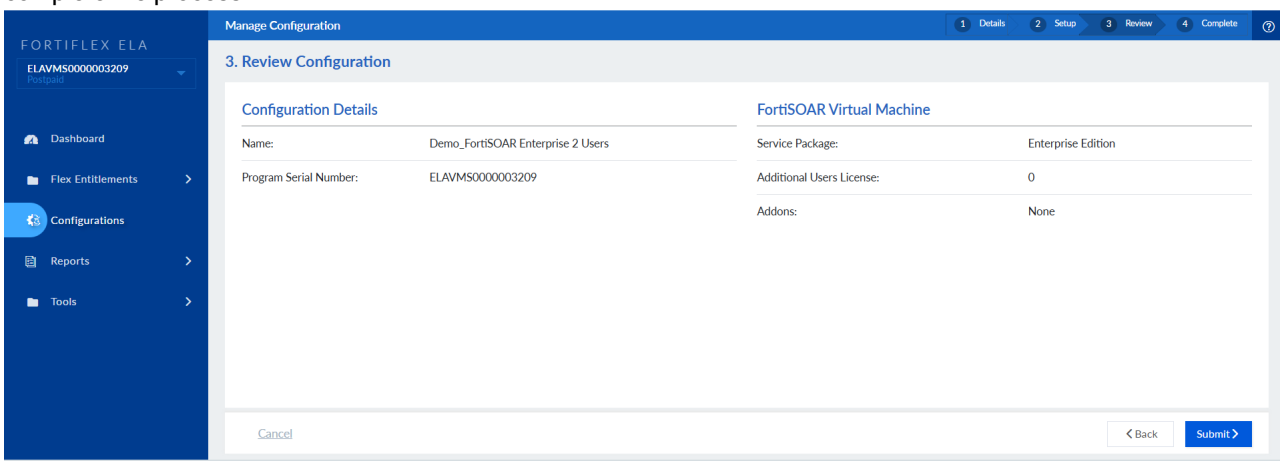
1. In the Support portal, navigate to **Services** > **FortiFlex** > **Configurations**.
2. On the `Configurations` page, click **New Configuration**.
3. On the `Configuration Details` page, provide the following details:
   a. In the **Name** field, enter a name for the configuration.
   b. In the `Form Factor` section, select **Virtual Machines**.
   c. In the `Product Type` section, choose **FortiSOAR Virtual Machine**, then click **Next**.



4. On the `Configuration Setup` page, choose the required service packages and licensing:
   a. From the **Service Packages** drop-down list, select one of the following options:
      - Enterprise Edition
      - Multi Tenant Edition - Manager
      - Multi Tenant Edition - Tenant Node - Single User
      - Multi Tenant Edition - Tenant Node - Multi User



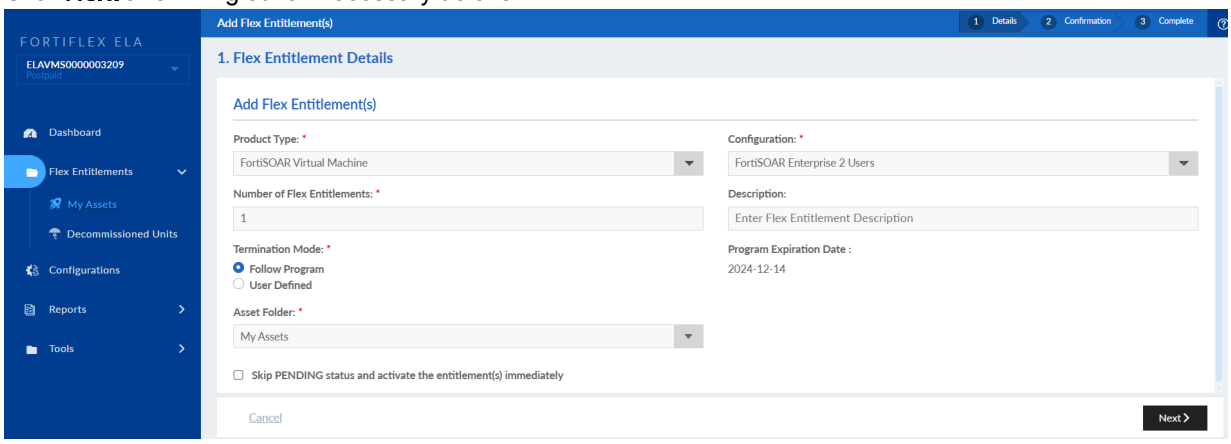
   b. In the **Additional User License** field, enter the number of additional users required beyond the default provided by the base license. By default, this is set to '0'.
   c. Click the **Threat Intelligence Management** checkbox to add the Threat Intelligence Management (TIM) subscription to your entitlement. This enables full access to TIM services.
   d. Click **Next** to proceed to the `Review Configuration` page.
5. On the `Review Configuration` page, verify the configuration details. If everything is correct, click **Submit** to complete the process.



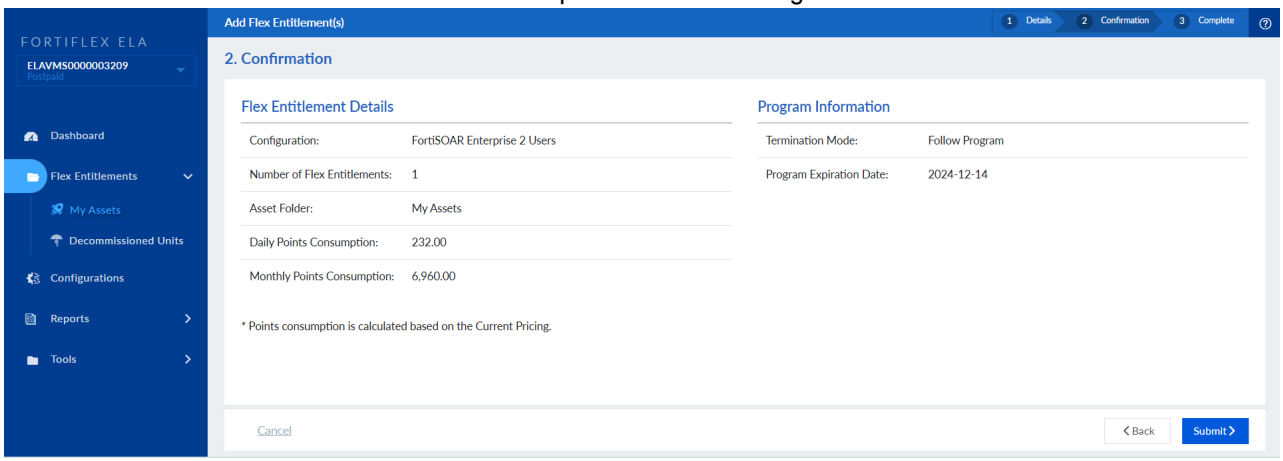
FortiFlex will create the configuration based on the provided details.

Creating an entitlement for the FortiSOAR configuration in FortiFlex

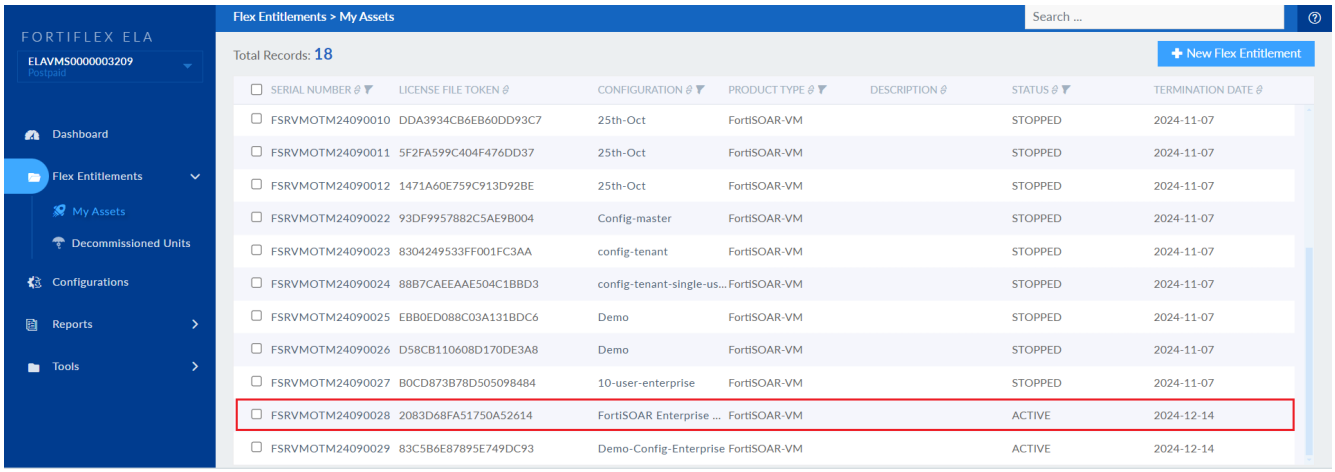
1. In the FortiFlex portal, go to **Flex Entitlements**.
2. Click **My Assets** > **New Flex Entitlement**.
3. On the `Flex Entitlement Details` page, provide the following details:
   a. In the `Product Type` section, select **FortiSOAR Virtual Machine**.
   b. From the **Configuration** drop-down list, select the configuration for which you are creating the entitlement.
   c. In the **Number of Flex Entitlements** field, enter the number of entitlements to be generated.
   d. In the **Description** field, provide a description for your FortiSOAR VM entitlement.
   e. For the **Termination** Mode, choose **Follow Program** or **User Defined**.
      - **Follow Program**: Sets the license expiry date to match the FortiFlex program expiry.
      - **User Defined**: Sets the expiry date based on your manually specified date.
        If selected, then in the **Expiration Date** field, set the license expiration date.
   f. From the **Asset Folder** drop-down list, select the folder in which you want to create the entitlement.
   g. Select the **Skip PENDING status and activate entitlement(s) immediately** checkbox if you want to set your entitlements to ACTIVE instead of PENDING status once the entitlements are created.
   NOTE: You will be billed once the entitlement is set to ACTIVE, which occurs when the license token is used.
   h. Click **Next** after filling out all necessary details.



4. On the `Configuration` page, review the Flex entitlement details, including daily and monthly point consumption. If correct, click **Submit**.
   FortiFlex will create the entitlements based on the provided details and generate a license serial number.



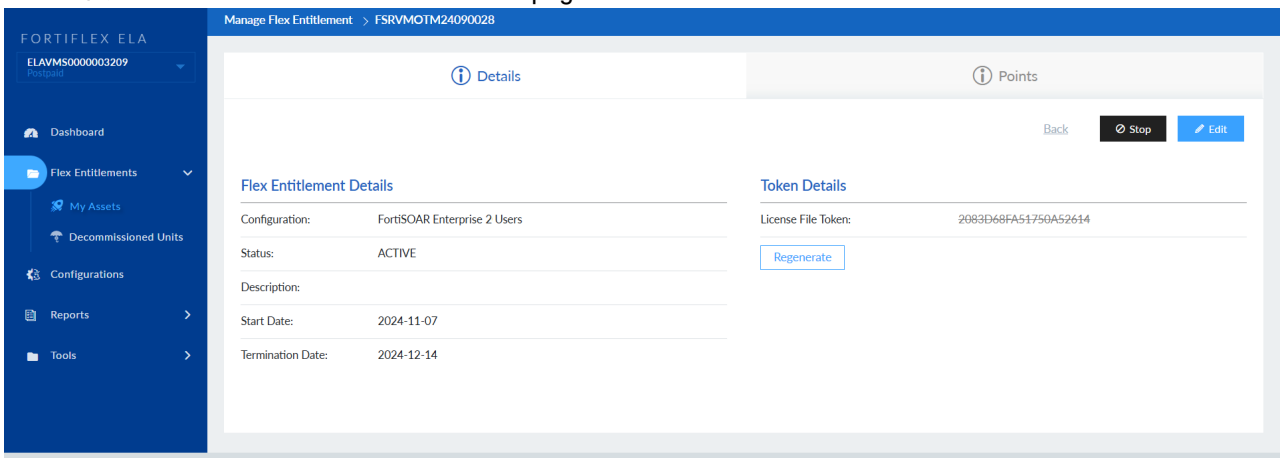
Click **My Assets** to view the license file token and check the entitlement status (ACTIVE or PENDING) associated with the serial number.



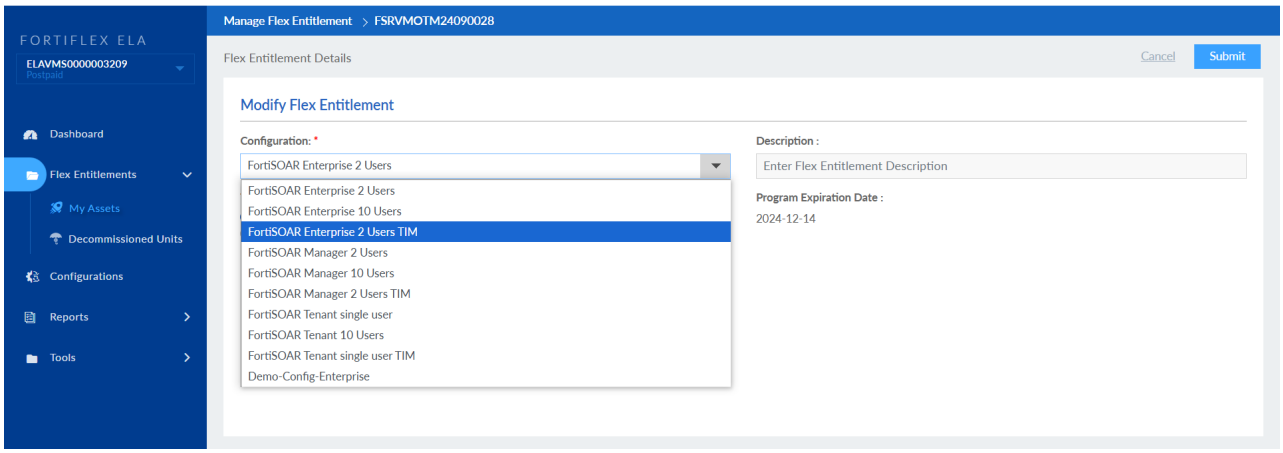
Modifying an entitlement for a FortiSOAR configuration in FortiFlex

FortiFlex allows you to easily modify the entitlement for your FortiSOAR configuration. For example, you can increase the validity of your license using the FortiFlex portal, if your license has expired.
To change a configuration, do the following:

1. Select the entitlement you want to update from the **Flex Entitlements** > **My Assets** listing page.
2. Click **Edit** on the `Flex Entitlement Details` page.



3. On the `Modify Flex Entitlement` page, select the desired configuration from the **Configuration** drop-down list and click **Submit**.
   For example, to change a 'FortiSOAR Enterprise 2 Users' configuration to 'FortiSOAR Enterprise 2 Users TIM', on the `Modify Flex Entitlement` page, select this option from the **Configuration** drop-down list. You can also optionally update the description and Termination Mode and click **Submit**.



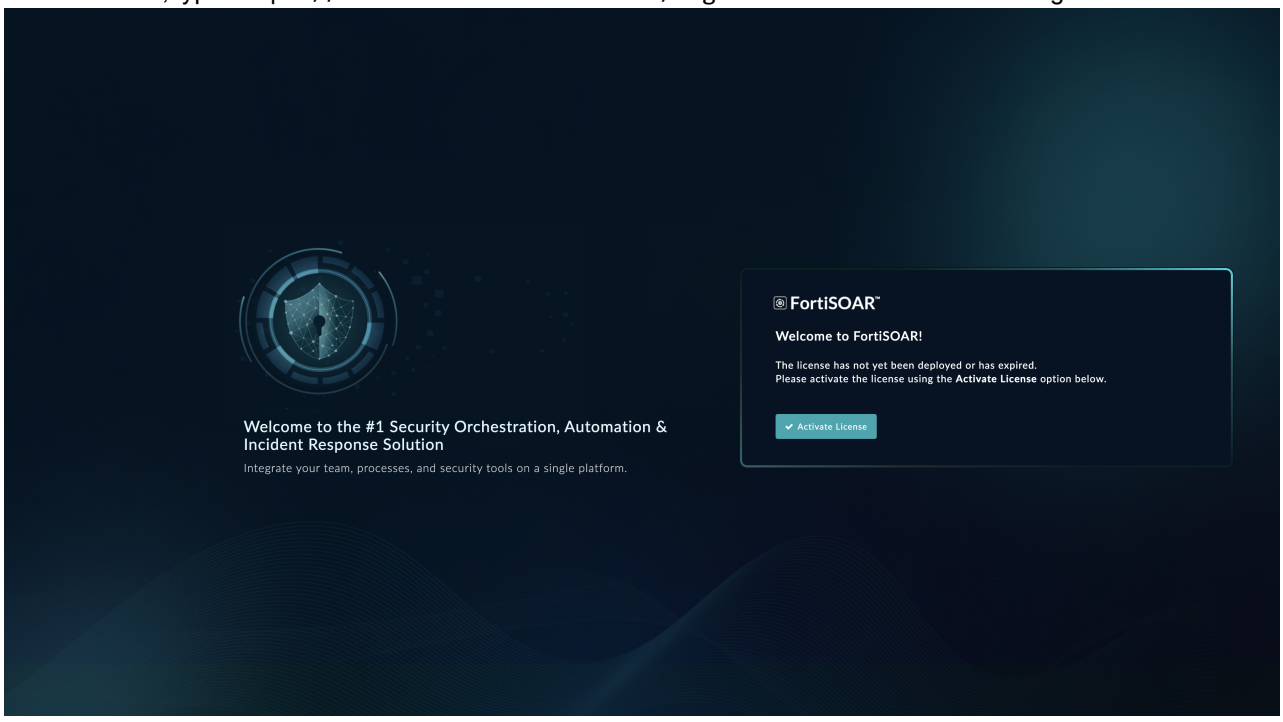
NOTE: Changes to the entitlement may take up to 2 hours to be reflected.

If your FortiSOAR license has been deleted from your instance, you will need to fetch the license again, which requires regenerating the license file token. To regenerate a license file token, click **Regenerate** on the `Flex Entitlement Details` page.

Activating your FortiSOAR license using the FortiFlex token

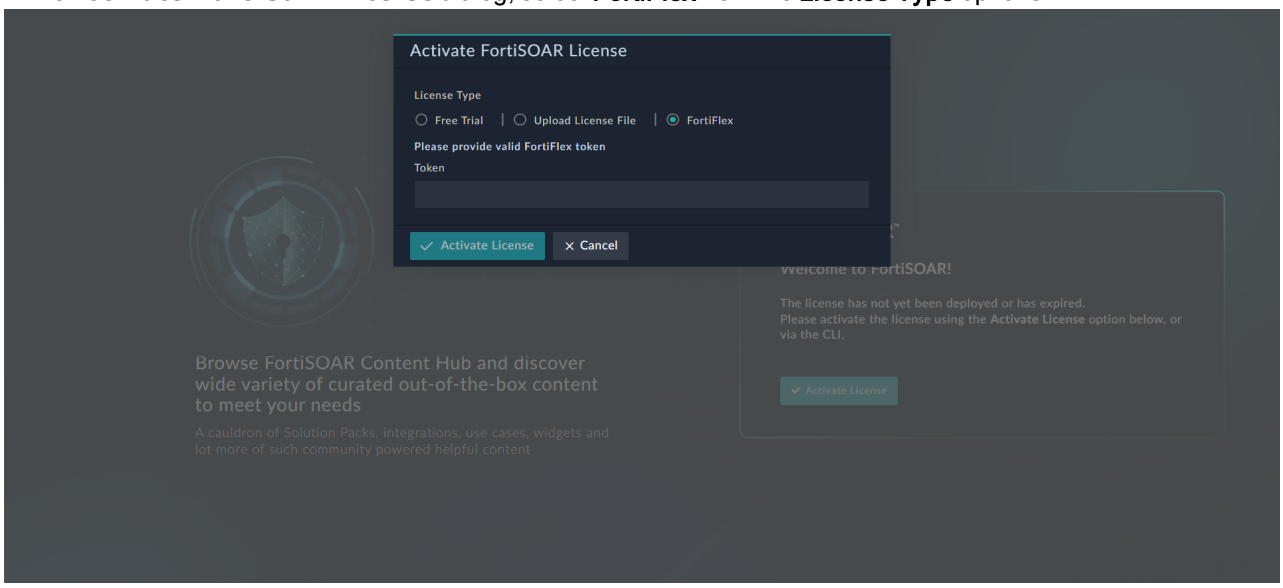
To activate your FortiSOAR license using the FortiFlex token, follow these steps:

1. In the browser, type `https://<YourFortisoarHostname>/login` to access the FortiSOAR login screen:



2. Click **Activate License**.

3. In the `Activate FortiSOAR License` dialog, select **FortiFlex** from the **License Type** options:



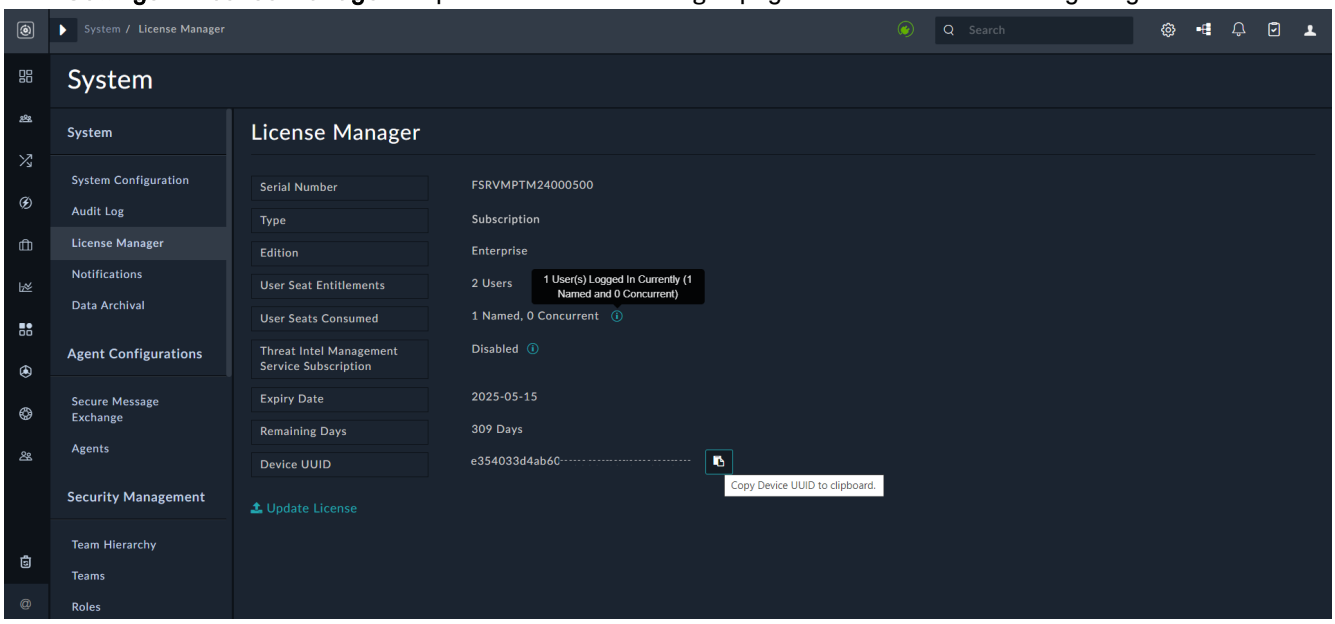
4. In the **Token** field, enter the FortiFlex token generated during entitlement creation in FortiFlex. Click **Activate License** after entering the details.

If the specified FortiFlex token specified is correct, your FortiSOAR license will be activated successfully, allowing you to log into FortiSOAR.

License Manager Page

FortiSOAR supports both 'Named' and 'Concurrent ' users. Concurrent user seats allow sharing a fixed number of user seats among an unlimited number of users, limited by the number of users accessing FortiSOAR simultaneously. This feature is beneficial for shift-oriented SOC environments, where a team may have a limited number of members working in a given shift, for example a 30-member team only has 10 members working in a given shift. In this scenario, administrators can create 10 concurrent users and efficiently manage user seats across all shifts. For more information, see the User Seat Support in FortiSOAR section.

Click **Settings > License Manager** to open the `License Manager` page as shown in the following image:



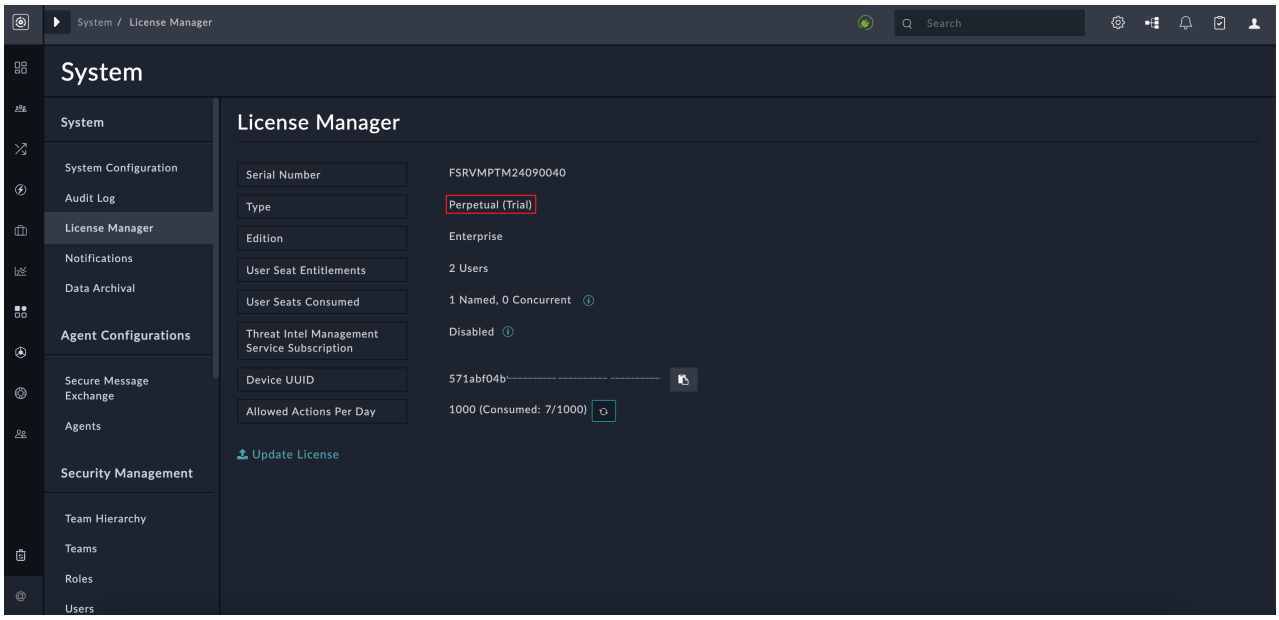
The `License Manager` page displays essential information such as the serial number, type and edition of the license issued, the total number of users FortiSOAR is licensed for, the number of users created on the system per access type, the number of users who are currently logged into FortiSOAR, the date when the FortiSOAR license will expire, the number of days till the expiry of the FortiSOAR license, and your Device UUID. You can click the **Copy Device UUID** button to copy your Device UUID.

If your license is about to expire, update it by clicking **Update License** and selecting the appropriate license type, either **Upload License File** or **FortiFlex**. If you choose **Upload License File**, you can either drag-and-drop your updated license file or click to browse and select the file from its location and then click **Open**. If you select **FortiFlex**, then in the **Token** field enter the FortiFlex token that was generated during entitlement creation. For details, see the Deploying the FortiSOAR license topic.
If the user count is reduced in updated license and exceeds the number of logged-in users, i.e., if the logged in users are more than the new count then the logged in users will get logged out at the time of session refresh one by one till the count becomes equal or less. Similarly, if the 'Named' user count exceeds the new license count, then only the '*Super Admin*' user will be able to log into system. For more information about named users, see the User Seat Support in FortiSOAR section. For more information about a '*Super Admin*' user, see Users topic in the *Security and Access Management* chapter of the "Administration Guide."

Serial Number: The serial number is a unique ID that is created by the FortiCare portal when you register your FortiSOAR product.

The FortiSOAR license can be of the following types:

- **Perpetual**: This type of license provides you with a license for an unlimited time for FortiSOAR.
- **Perpetual (Trial)**: This type of license provides you with a free trial license for FortiSOAR, allowing unlimited time usage but with restrictions on the number of users and actions that can be performed in a day. From release 7.6.0 onwards, his license limits FortiSOAR usage to 2 users for a maximum of 1000 actions per day, and you can choose the edition for this license to either "Enterprise" or "Multi-Tenant".



For more information on the trial license, see the Activating the FortiCare Trial license for FortiSOAR topic.

- **Subscription**: This type of license is a regular license that gives you subscription to FortiSOAR for a particular number of users and a specific timeframe.
  You can renew your subscription and change the number of users as per your requirements. FortiSOAR will synchronize with the FDN server and retrieve the latest subscription.
- **Subscription (Starter)**: Release 7.6.0 introduces this edition to offer small to medium-sized enterprises or teams within larger organizations a more accessible entry point into advanced security orchestration, automation, and response functionality. This edition supports scaling security operations efficiently in growing organizations, enabling them to handle an increased volume of alerts and incidents with a robust yet cost-effective toolset. This edition is priced lower than the full-fledged "Enterprise" edition, allowing growing organizations to experience the FortiGuard Outbreak and Threat Intel Management features fully. It also enables users to use FortiSOAR as a dev/staging instance without the need to pay for a full FortiSOAR edition. The Starter edition's license type is "Subscription" and edition is "Enterprise". By default, it limits FortiSOAR usage to 2 users for a maximum of '10000' actions per day.
- **Evaluation**: This type of license allows you to evaluate FortiSOAR. The evaluation license is shipped with a predefined user count and expiry date.

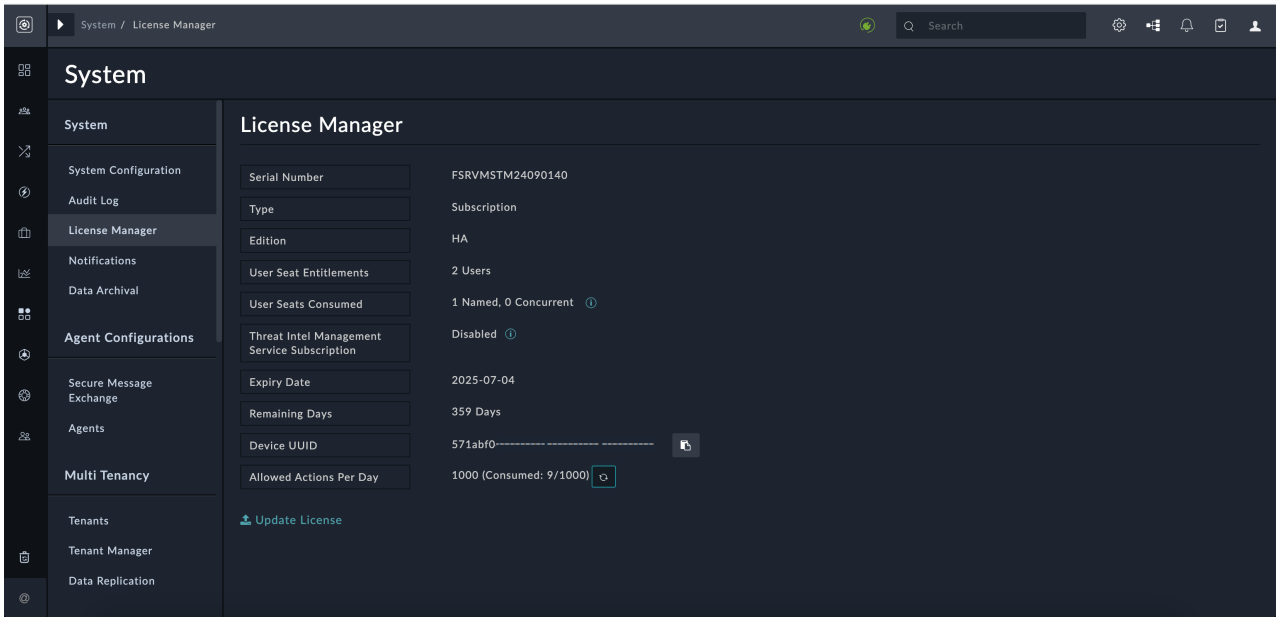
> Once the daily limit of action counts is reached based on the type of license such as Starter or Trial licenses, deployed on your FortiSOAR instance, appropriate messages such as "Daily limit for action count has been exceeded. You cannot execute any more actions." will be displayed on your FortiSOAR UI.

The FortiSOAR license can have the following editions and roles:

- **Enterprise**: This edition enables a regular "enterprise" production license.
- **Multi-Tenant**: This edition enables multi-tenancy; both shared and distributed multi-tenancy are supported. The edition along with the 'Role' determines whether the instance acts as a "manager" or "tenant" node of a Managed Security Services Provider (MSSP):
  - If the '**Role**' is set as "Manager" then the instance where this license is deployed serves as a “master” node in a distributed deployment.
    NOTE: Prior to release 7.6.0, this role was referred to as "`MT (Master)`".
  - If the '**Role**' is set as "Tenant (Single user locked)", then the instance where this license is deployed serves as a "dedicated tenant" to the MSSP server for syncing data and actions to and from the MSSP "master" server.
    NOTE: Prior to release 7.6.0, this role was referred to as "`MT_Tenant`".

- If the '**Role**' is set as "Tenant (Multiple user capable)", then the instance is enabled as a "Tenant (Multiple user capable)" deployment at an organization that has a distributed SOC, and which is a complete SOAR platform. At the same time, it can be configured as a "tenant" to the global SOC where the "Multi-Tenant (Manager)" license is deployed and sync data and actions from the Global SOC FortiSOAR server.
  NOTE: Prior to release 7.6.0, this role was referred to as "`MT_RegionalSOC`".
  For more information of what multi-tenancy is and what master nodes are, see the "Multi-tenancy support in FortiSOAR Guide."

- **HA**: Release 7.6.0 introduces this edition, to allow an instance to act as a High Availability (HA) node in a clustered setup.
  NOTE: Using this edition a FortiSOAR system can function as a standalone instance with 2 users and 1000 actions per day if it is not added as a cluster node:



Previously, to establish an HA setup, all nodes within the cluster needed to have identical licenses, which meant users required to acquire additional licenses for secondary/backup nodes of the same type and edition as the primary node, increasing the overall cost of the cluster setup. The HA edition is available at a lower cost than the current 'Enterprise' node license due to its specialized and restricted capabilities. This makes it cost-effective for creating an HA cluster, as setting up an n-node cluster requires n-1 HA licenses. For example, for a configuration with 1 active and 1 passive node (totaling 2 nodes), you will need 1 HA license. Similarly, for a setup with 2 active nodes and 1 passive node (totaling 3 nodes), you will need 2 HA licenses. These HA licenses should be applied to the non-primary nodes of the cluster.
The HA edition is available for both 'Perpetual' and 'Subscription' licenses and can be used in all environments supporting HA configurations. The 'HA' license should be applied to the node designated as the 'Secondary' node of the cluster, not the 'Primary' node. Additionally, all existing licenses installed on secondary nodes can be converted to the HA license, except for the Dedicated Tenant (Single User Locked), and vice versa.
IMPORTANT: HA License can only be used in conjunction with the FortiSOAR base license. Other licenses, such as TIM or additional users, do not need to be duplicated, and therefore are not applicable to the HA edition.
The following table illustrates the combination of license type of primary and secondary nodes that can be used for HA formation:

Primary Node		Secondary Node		HA Formation
Type	Edition	Type	Edition	
Perpetual	Enterprise	Perpetual	Enterprise	Allowed
Perpetual	Enterprise	Perpetual	HA	Allowed
Perpetual	Multi-Tenant (Manager)	Perpetual	Multi-Tenant (Manager)	Allowed
Perpetual	Multi-Tenant (Manager)	Perpetual	HA	Allowed
Perpetual	Multi-Tenant - Tenant (Multiple user capable)	Perpetual	Multi-Tenant - Tenant (Multiple user capable)	Allowed
Perpetual	Multi-Tenant - Tenant (Multiple user capable)	Perpetual	HA	Allowed
Perpetual	Multi-Tenant - Tenant (Single user locked)	Perpetual	Multi-Tenant - Tenant (Single user locked)	Not Allowed
Perpetual	Multi-Tenant - Tenant (Single user locked)	Perpetual	HA	Not Allowed
Perpetual (Trial)	Enterprise	Perpetual (Trial)	Enterprise	Not Allowed
Perpetual (Trial)	Multi-Tenant (Manager)	Perpetual (Trial)	Multi-Tenant (Manager)	Not Allowed
Subscription	Enterprise	Subscription	Enterprise	Allowed
Subscription	Enterprise	Subscription	HA	Allowed
Subscription	Multi-Tenant (Manager)	Subscription	Multi-Tenant (Manager)	Allowed
Subscription	Multi-Tenant (Manager)	Subscription	HA	Allowed
Subscription	Multi-Tenant - Tenant (Multiple user capable)	Subscription	Multi-Tenant - Tenant (Multiple user capable)	Allowed
Subscription	Multi-Tenant - Tenant (Multiple user capable)	Subscription	HA	Allowed
Subscription	Multi-Tenant - Tenant (Single user locked)	Subscription	Multi-Tenant - Tenant (Single user locked)	Not Allowed
Subscription	Multi-Tenant - Tenant (Single user locked)	Subscription	HA	Not Allowed
Subscription (Starter)	Enterprise	Subscription (Starter)	Enterprise	Allowed

Primary Node		Secondary Node		HA Formation
Type	Edition	Type	Edition	
Subscription (Starter)	Enterprise	Subscription	Enterprise	Allowed
Evaluation	Enterprise	Evaluation	Enterprise	Allowed
Evaluation	Multi-Tenant (Manager)	Evaluation	Multi-Tenant (Manager)	Allowed
Evaluation	Multi-Tenant - Tenant (Multiple user capable)	Evaluation	Multi-Tenant - Tenant (Multiple user capable)	Allowed
Evaluation	Multi-Tenant - Tenant (Single user locked)	Evaluation	Multi-Tenant - Tenant (Single user locked)	Not Allowed

Threat Intel Management Service Subscription displays whether unrestricted FortiGuard threat feeds and advanced Threat Intelligence Management features are enabled. For more information, see the Licensing option for Unrestricted FortiGuard Threat Feeds and Advanced Threat Intelligence Management topic.

User Seat Entitlements displays the number of user seats that you have purchased for FortiSOAR. You cannot create more named active users, in your FortiSOAR environment, than the value specified as in this field. For example, if the User Seat Entitlements field is set to five, then you can create a maximum of five named users, and an unlimited number of concurrent users; however, if all five named users are active, then no concurrent user will be able to log into FortiSOAR. Also, note that if a user is logging in from multiple places, then it is counted as a single user. For more information, see the User Seat Support in FortiSOAR section.

User Seats Consumed displays the number of active users, named and concurrent, who have consumed the FortiSOAR user seats. To view the number of users, named and concurrent, who are currently logged into FortiSOAR, you can hover over the tooltip.

Expiry Date displays the date at which your FortiSOAR license will expire and **Remaining Days** displays the number of days left for your license to expire.

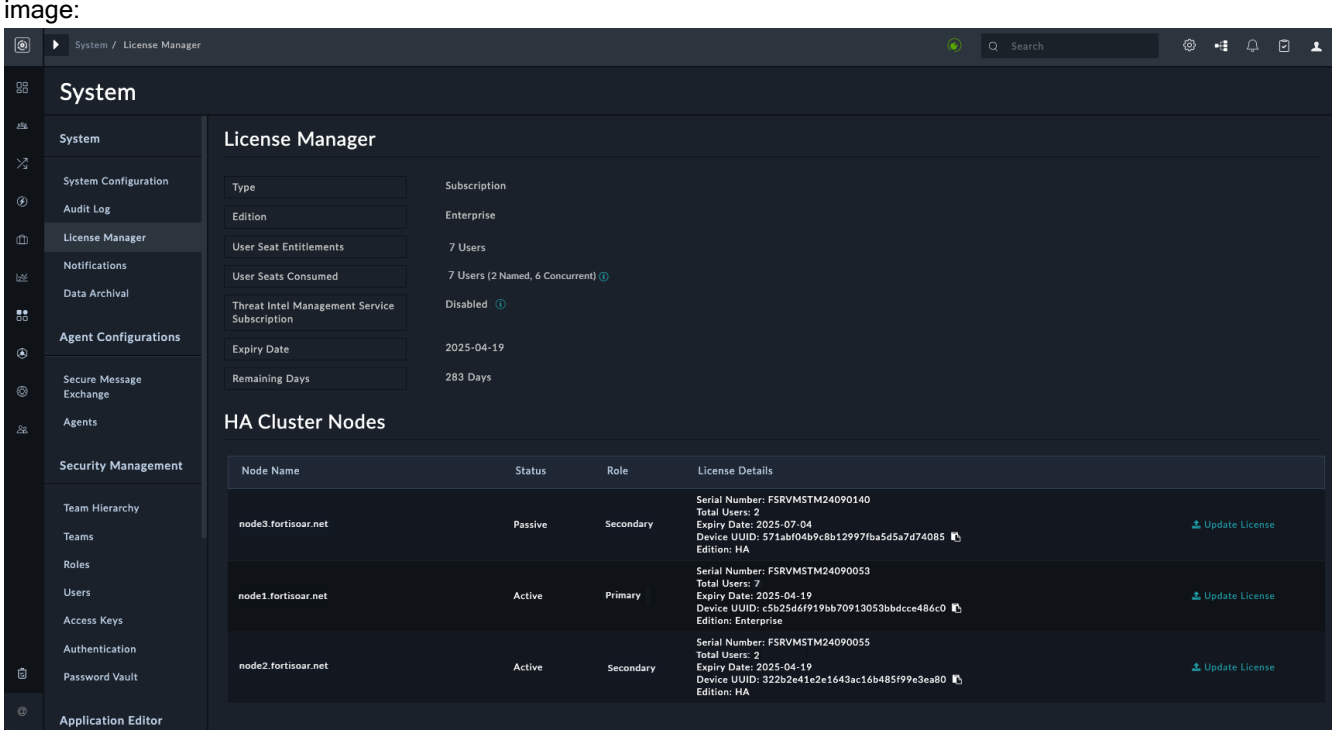
Allowed Actions Per Day displays both the total action count and the remaining number of FortiSOAR actions users can perform in a single day. A refresh button next to this field allows users to update the count without reloading the 'License Manager' page.

FortiSOAR does not mandate 'Additional Users' entitlement to be the same across all cluster nodes. User count entitlement is validated from the primary node. The secondary nodes can have the basic two-user entitlement. The HA cluster shares the user count details from primary node of the cluster. Hence, all 'Concurrent Users' count restrictions apply as per the primary node. If a node leaves the cluster, the restriction will apply as per its own original license.

In the case of an HA environment, you only need to buy one Threat Intelligence Management (TIM) subscription that can be used across your HA cluster. The primary node subscription is cascaded to the secondary nodes.

In case your FortiSOAR instance is part of a High Availability (HA) cluster, then the `License Manager` page also displays information about the nodes in the cluster, if you have added secondary node(s) as shown in the following

image:



As shown in the above image, the primary node is Node 1 and that node is licensed with 7 users, therefore the User Seat Entitlements count displays as 7 users. For more information on licensing of nodes in an HA cluster, see the High Availability Configuration and Maintenance chapter in the "Administration Guide."

You can update the license for each node by clicking **Update License** and either uploading the license file or entering the appropriate FortiFlex token for that node, as described in the Deploying the FortiSOAR License topic.

If you deploy a license that does not match the system UUID, you will receive a warning on the CLI during deployment. If you deploy the same license in multiple environments, it will be detected as a duplicate and you will be required to correct the license. Otherwise, your FortiSOAR UI will be blocked as follows:

- Once a license is identified as a duplicate, it is checked using FDN sync every hour to verify if the duplicate license issue has been resolved.
- After two such warnings, with a one-hour gap between them, the FortiSOAR UI will be blocked for users.
- The FortiSOAR UI will remain blocked until a new license is deployed or the system with the duplicate license is shut down.

User Seat Support in FortiSOAR

FortiSOAR supports 'Named' and 'Concurrent ' users for licensing. User access details are used to calculate the number of concurrent users that can simultaneously log onto FortiSOAR.

Named Users

'Named' users are users for whom a seat is permanently reserved, i.e., such a user can always log onto FortiSOAR except in case of a license violation.

Concurrent Users

The ability to designate a user seat as a 'concurrent user seat' allows system administrators to create a floating seat that can be shared by unlimited users (only limited by the user seat limit). A 'Named' user has a FortiSOAR seat permanently reserved, i.e., such a user can always log onto FortiSOAR except in case of a license violation. However, a concurrent user can log in only when there is a concurrent seat available. Note that if a user is logging in from multiple places, it is counted as a single user.

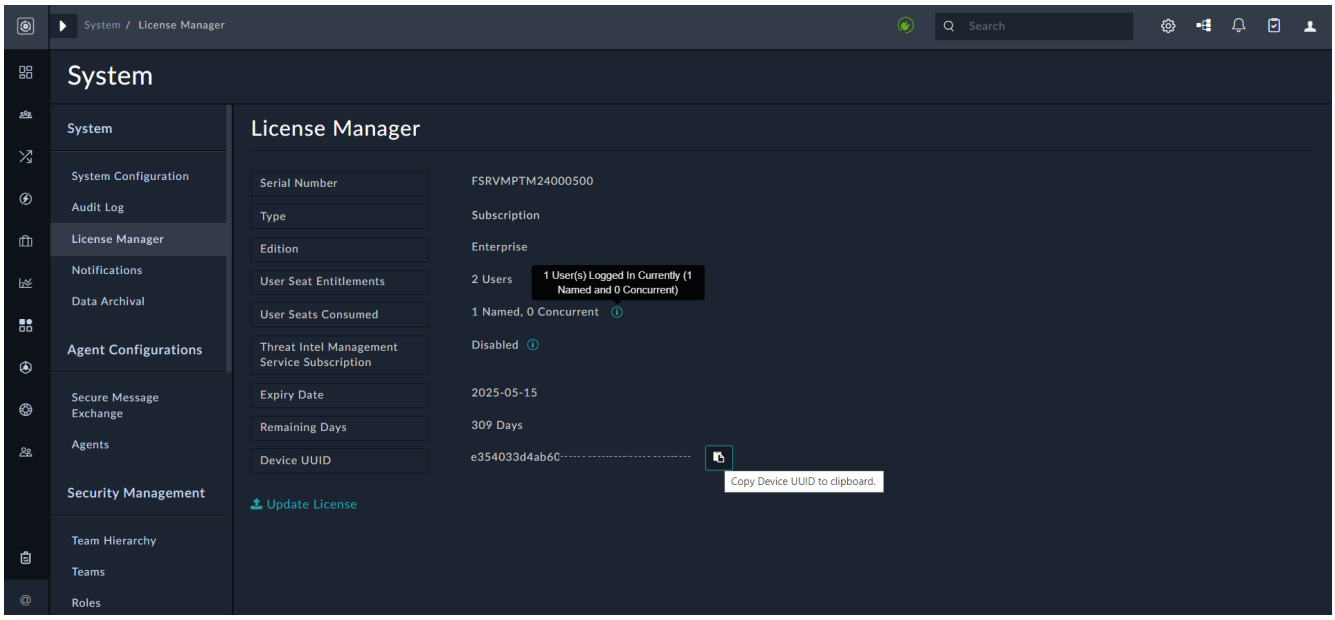
For example, if you have purchased a five-user license, then a maximum of 5 named active users can be present in the system at a given time. However, there is no limit to concurrent user creation, i.e., you can create as many concurrent users as you want. Therefore, if out of five user seats that you have purchased, you have created two Named users, then those users can log into FortiSOAR at any time, and the other three seats are reserved for Concurrent users, who can log into FortiSOAR when concurrent seats are available. However, if the you create five Named users, then only those users will be able to log into FortiSOAR and Concurrent users will not be able to log into the system.

Administrators, i.e., users with `Security and People Update` access, can selectively change users' access type, i.e., Concurrent users to Named users, and vice-versa, at any time, or they can also bulk change users access type from Named to Concurrent. For more information, see the Security and Access Management chapter in the "Administration Guide." They also have the privilege to forcefully log out selective 'Concurrent' users. When the administrators logs out a user from any instance, that user is notified before being logged out.

The default access type set for all SSO and MSSP users is 'Concurrent'. You can change the access type for the user later, if needed.

Updating your license using the FortiSOAR UI

You can update your license using your FortiSOAR UI. Click **Settings > License Manager** to open the `License Manager` page.



You can use the `License Manager` page to view your license details and to update your license. FortiSOAR displays a message about the expiration of your license 15 days prior to the date your license is going to expire. If you license type is **Evaluation**, then you must update your license within 15 days, if you want to keep using FortiSOAR. To update your license, click **Update License** and either upload the updated license file or enter the appropriate FortiFlex token, as described in the Deploying the FortiSOAR License topic. If your license type is **Subscription**, you must renew your subscription.

Licensing option for Unrestricted FortiGuard Threat Feeds and Advanced Threat Intelligence Management

FortiSOAR offers a licensing option, the Threat Intel Management Service Subscription, that provides full access to premium FortiGuard threat intelligence feeds. This subscription enables you to utilize the Threat Intelligence Management (TIM) Service without restrictions, including unlimited consumption of FortiGuard feeds.
The FortiGuard feed contains a comprehensive dataset of IP addresses, URLs, domains, and hashes, curated by our expert team. Where applicable, entries are tagged with relevant threat types and mapped to the Lockheed Martin Kill Chain phases, providing essential context to help users understand the nature of threats.

Verifying your License Status:
To check if this licensing option is enabled, navigate to the `License Manager` page within the FortiSOAR UI. Check the **Threat Intel Management Service Subscription** option, which will indicate whether it is **Enabled** or **Disabled**.

When you have the TIM Subscription enabled, the following advanced features become available:

- **Contextual Sighting Information**:
  Once Threat Intelligence Management is set up, including the configuration of threat intelligence feeds, integrations, feed processing rules, and confidence thresholds, FortiSOAR automatically matches each created indicator against the FortiGuard feed database (or any other configured feed database) and links matched feeds to

the extracted indicators. This contextual sighting is only available if the *TIM Service Subscription* is enabled. For configuration details, see the Threat Intel Management Solution Pack documentation.
Benefits of contextual sighting include:
  - Access to enriched contextual data even when standard enrichment sources lack information about suspicious targets.
  - A dashboard displaying the relevance of intelligence sources based on actual sightings in the user's environment.
- **Unlimited Feed Ingestion**:
With the subscription enabled, there is no daily limit on the volume of threat feed records ingested via the FortiSOAR Feeds API.
If the subscription is disabled, the ingestion limit is capped at 1,000 records per day. Attempts to ingest beyond this limit will result in a "Daily Feed Ingestion Limit reached" error, which resets daily at midnight (UTC).
For example, if you have 100 records remaining from the daily 1,000-record ingestion limit and submit 200 records as part of an ingestion feed, only the first 100 will be saved; the remaining 100 will be ignored.
- **Unlimited Threat Intel Searches**:
Licensed users can perform unlimited searches on FortiGuard Labs' global threat intelligence database for detailed insights on domains, IP addresses, and other indicators of compromise (IOCs).
Without a subscription, users are limited to a maximum of 100 searches per day. Each search can include a maximum of 10 indicators, regardless of subscription status.
- **Unlimited Feed Exports via FortiSOAR's TAXII API**
The subscription removes limits on exporting processed threat intelligence to SIEMs, firewalls, and other systems via the FortiSOAR TAXII API.
Without the subscription, the API response is limited to 100 records.

Summary of Feature Usage:

Feature	No TIM Service Subscription	TIM Service Subscription
Threat Intel Search (IOC)	100 searches/day, max 10 indicators per search	Unlimited, max 10 indicators per search
Contextual Sighting	Not supported	Supported
TAXII API Response	Limited to 100 records	Unlimited
Feed Ingestion Limit	1,000 records/day	Unlimited

For details on FortiSOAR's threat intel management capabilities, see the Threat Intelligence Capabilities topic in the *Use Advanced FortiSOAR Features* chapter of the "User Guide" and the Threat Intel Management Solution Pack documentation.

Troubleshooting licensing issues

FortiSOAR displays meaningful messages and troubleshooting tips during the license deployment process, and validates your FortiSOAR license, making it easier for you to debug licensing issues.

If your connection to FDN is via a proxy, you must update the proxy settings.

If any error occurs while deploying your license, following are some troubleshooting steps:

- If the license type is "Subscription", then the number of users and expiry date are not present inside the license. They require to be synced from FDN after the installation. The "License has expired issue after installation" issue occurs due to the following two reasons:
  - Sync with FDN failed
  - Sync was successful but we got wrong contract information.
    To verify the above-mentioned cases run the following command:
    `sudo java -jar <jar_path> <serial_no> <device_uuid> <globaupdate_url>`
    For example,
    `sudo java -jar /opt/cyops-auth/bin/fdnclient.jar <serial_no> <device_uuid> https://globalupdate.fortinet.net`
- If the license type is "Evaluation" or "Perpetual", then the number of users and expiry date are present inside the license. If a license deployment failure occurs for these types of licenses, then check the license information using the `sudo csadm license –show-details <lic_file>` command.
- After deploying the license if the system is yet not reachable, restart the `cyops-auth` service:
  `sudo csadm services --restart cyops-auth`
  Monitor the `fdn.log` and `das.log` files. If you continue to face issues, contact FortiSOAR support.

For additional guidance, see the Troubleshooting licensing issues in FortiSOAR article for tips on resolving common licensing problems.

Troubleshooting issues while deploying the FortiSOAR license in a proxy environment

You might get the following error, when you are deploying your FortiSOAR license in a proxy environment:

```
FSR-Auth-003: License Entitlement Sync Failed. Ensure that [https://globalupdate.fort](https://globalupdate.fort/) is accessible from your environment. If the issue still persists, contact support."
```

This issue might occur due to some proxies doing the SSL decryption, which means that these proxies can intercept the https connection by modifying the peer certificate and changing the issuer of the certificate to itself. This can cause the license deployment or synchronization to fail as the new issuer is not trusted.

To identify this issue, check the `PKIX path building failed` error message in the `fdn.log` file:
`# sudo vi /var/log/cyops/cyops-auth/fdn.log file`

Resolution

You can use the following two solutions to solve this issue.

Method 1: Do not use SSL decryption for globalupdate.fortinet.net.

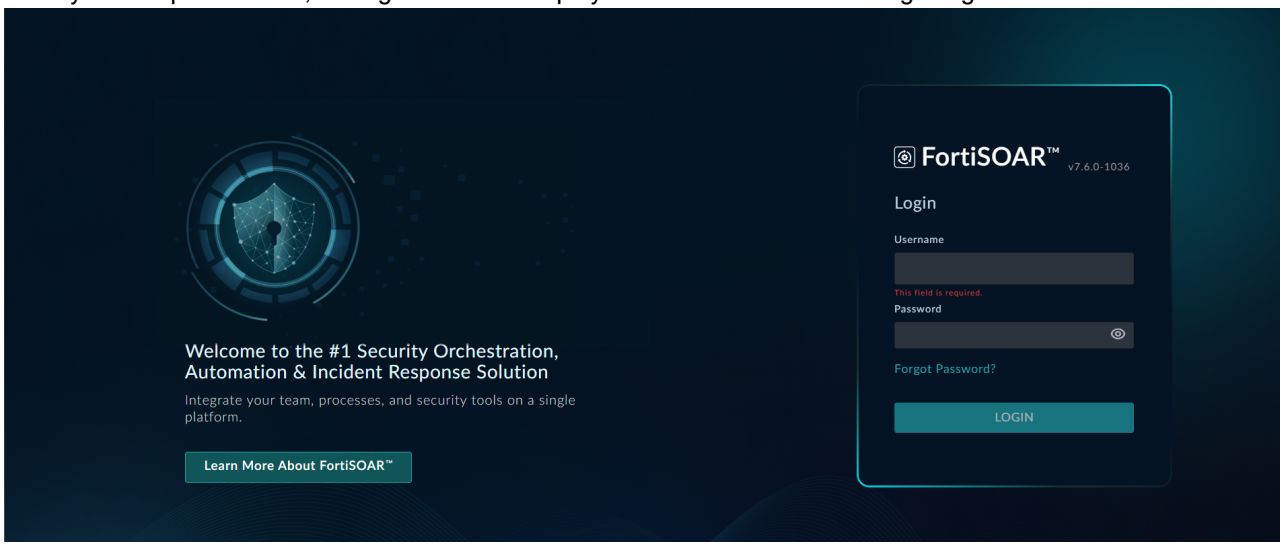
Method 2: Import the proxy issuer certificate into truststore using the following command:
`sudo keytool -import -alias proxy_issuer_cert -keystore /opt/cyops-auth/certs/fdn_server_truststore.p12 -file<cert_file> -storepass MXakK2bj6vAteC47 -noprompt`

Configuring FortiSOAR

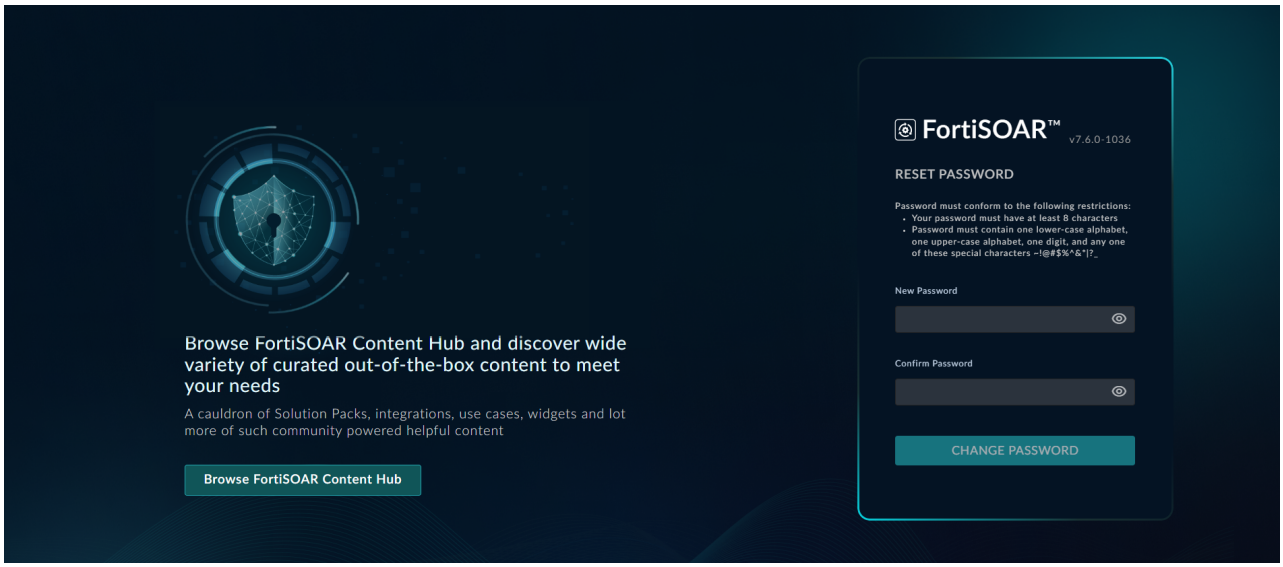
This chapter describes the initial configuration steps required for setting up your FortiSOAR system.

Logging on to FortiSOAR for the first time

1. In a browser, enter the IP address that you had identified using the steps mentioned in the *Determining your DHCP IP address* section as and press `Enter`.
   For example, https://{*Your_FortiSOAR_IP*}
   This will display the Fortinet End-User License Agreement (EULA). You must accept the EULA before you can log onto FortiSOAR.
   Once you accept the EULA; the login screen is displayed as shown in the following image:



2. Login using the following credentials:
   Username: csadmin
   Password: changeme
   The UI password of the 'csadmin' user for AWS is set to the "instance_id" of your instance. To know the instance ID of your FortiSOAR AWS instance, you can SSH and run the `sudo cloud-init query instance_id` command.
   If you are a 'csadmin' user, and you are logging into FortiSOAR for the first time, you will be mandated to change the *default password*. This enhances the security of your csadmin account and prevents unauthorized parties from accessing the administration account for FortiSOAR.
   New passwords that are set must contain at least 8 characters, one lower-case alphabet, one upper-case alphabet, one digit, and any one of the following special characters ~ ! @ # $ % ^ & * | ? _
   Once you enter the 'csadmin' username and default password the following screen is displayed, which prompts you to reset the default password:

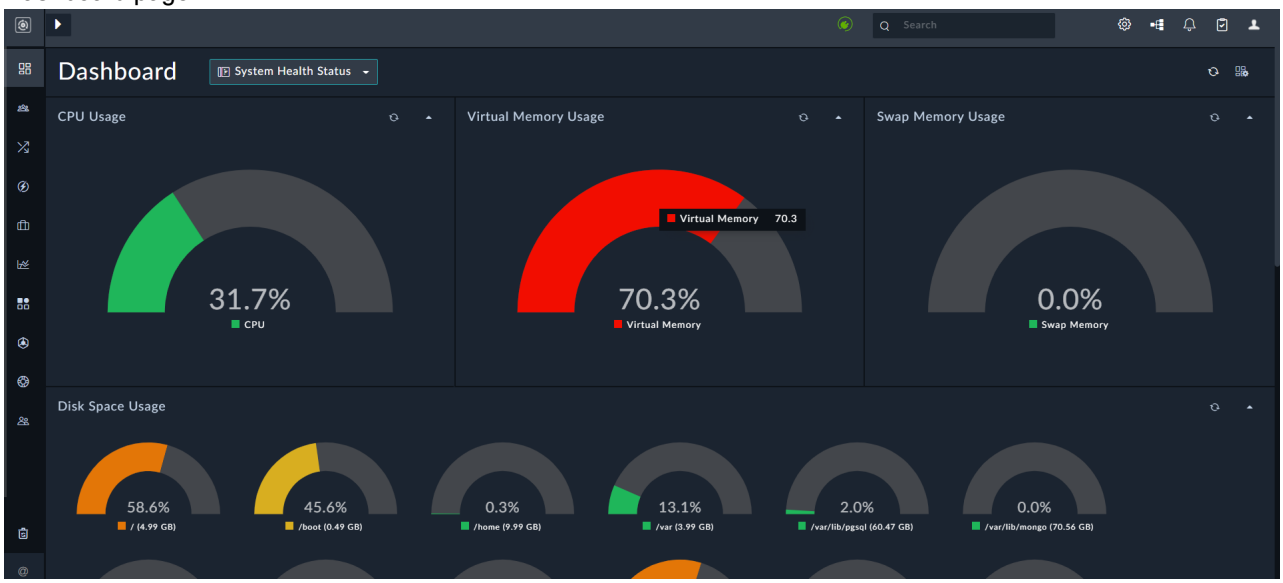


Ensure that you note down your csadmin password since if you forget your initial csadmin password, then you have to request FortiSOAR to reset this password. Also, when you are changing your csadmin password, you must ensure that you also update the email ID that is specified for csadmin, which by default is set to `soc@fortinet.com` (which is not a valid email ID). You can change the email ID by clicking the **User Profile** icon ( ) to open the `User Profile` page and change the email address in the **Email** field.

Once you set a valid email ID in the user profile, then you would be able to reset your password, whenever required, by clicking the **Forgot Password** link on the login page.

Important: It is also recommended that all new users should change their password when they first log on to FortiSOAR, irrespective of the complexity of the password assigned to the users.

After you have changed the default password, FortiSOAR logs you into the application and by default displays the Dashboard page:



Now you can begin configuring FortiSOAR for your network environment.

Configuring SMTP for FortiSOAR

The SMTP connector comes pre-configured with FortiSOAR and it is required to receive any system or email notifications, including requests for resetting passwords. The SMTP connector is part of a number of pre-installed connectors or built-ins that are included with FortiSOAR. By default, the SMTP connector is configured to use FortiSOAR appliance as an SMTP relay server. You must point it to a production SMTP server in your organization. For more information, see the SMTP Connector document.

> ⚠ When you configure the SMTP connector, ensure that you select the **Mark As Default Configuration** option for the configuration that will be used for sending system notifications.

It is highly recommended that you review the Additional configuration settings for FortiSOAR chapter to understand the configurations that you should make in your FortiSOAR system before you begin to use FortiSOAR.

Creating your first user and record

> The following steps provide a high-level view of how to get started with FortiSOAR. These steps are explained in detail in "Administrators Guide."

1. Successfully log into FortiSOAR.
2. Click the **Settings** () icon that is present in the upper right-hand corner near the **User Profile** icon.
   This displays the `System` page.
   Use the `Security Management` section to configure the following: Team Hierarchy, Teams, Roles, Users, Authentication, and Password Vault.
3. Add a new team in FortiSOAR.
   You can also use the default teams that are present in FortiSOAR.
4. Add a new role in FortiSOAR.
   You can also use default roles that are present in FortiSOAR.
   You provide user permissions on a module based on roles that you have assigned to that user.
   For example, if you want to provide a user with complete access to the Incident module, you must create a role that has `Create`, `Read`, `Update`, and `Delete` permissions on the Incident module and name it Incident Administrator. You must then assign that role to a user.
5. Add a new user and assign an appropriate role to the user.
   For example, create a user John A and assign John A the Incident Administrator role.
6. Create your first record.
   Log on to FortiSOAR as user John A, who has access to the `Incident` module. Click the **Add** button in the top bar of the `Incidents` module to open the `Create New Alert` form. Fill in the required details the `Create New Incident` form and click **Save** to create an incident.

Additional configuration settings for FortiSOAR

You can optionally perform the following additional configurations for FortiSOAR based on your requirements.

If you want to externalize your FortiSOAR databases, which are PostgreSQL and ElasticSearch, see the "Best Practices Guide." The *Externalization of your FortiSOAR PostgreSQL database* chapter covers the steps for externalizing your PostgreSQL databases, and the *ElasticSearch Configuration* chapter covers the steps for externalizing your ElasticSearch database.

If you face any issues while deploying or upgrading FortiSOAR, see the *Troubleshooting FortiSOAR* chapter. If you face deployment or upgrade failures due to insufficient space, or if you face issues while using FortiSOAR that might be caused due to insufficient space, like you are unable to log into FortiSOAR or FortiSOAR services stop working, then see the `Issues occurring in FortiSOAR due to insufficient space` section in the *Troubleshooting FortiSOAR* chapter.

The default self-signed certificates shipped with FortiSOAR are valid for two years from the inception of your FortiSOAR instance and must be regenerated once they expire. For steps on regenerating these certificates, see the *Regenerating Self-Signed certificates* topic in the Monitoring & Optimization chapter in the "Best Practices Guide."

Changing the hostname

The FortiSOAR Configuration Wizard is available only on the first `ssh` login. If at a later stage, you require to change the hostname of your FortiSOAR VM, then you can use the FortiSOAR Admin CLI (`sudo csadm`). For more information on `csadm`, see the CLI Administration chapter in the "Administration Guide."

To change the hostname, ensure that the hostname is resolvable and then do the following:

1. SSH to your FortiSOAR VM.
2. To change your hostname, type the following command:
   `# sudo csadm hostname --set [<hostname>]`
   This command changes your current hostname to the new hostname that you have specified, sets up the message broker, regenerates certificates, and restarts FortiSOAR services.

It is recommended that you set the hostname of your FortiSOAR VM, at the time of deployment only and **not after** the FortiSOAR instance is in active use. If any errors occur when you are running the hostname change command, see the *Troubleshooting FortiSOAR* chapter.

Note: After the hostname has been reset, when users execute playbooks with an external manual input link, it is observed that the link that is generated in the email contains the original FQDN (hostname) rather than the one that has been updated. Therefore, users who are required to provide the input, have to manually update the FQDN (hostname) in the manual input link present in the email.

Updating the SSL certificates

Use the following procedure to update Nginx certificates within the FortiSOAR Virtual Appliance when the FortiSOAR certificates expire. You can also use the following procedures to replace FortiSOAR self-signed certificates with your own signed certificates.

Note: Your SSL certificate file must be in the `.crt` and `.key` format. FortiSOAR does not support certificate formats such as `cer`, `p7b`, etc.

If your certificate is in another format such as, a CER certificate from Windows CA, then you need to create the .crt certificate from a .cer certificate, using the following command:

```
# sudo openssl x509 -inform DER -in ssl_certificate.cer -out ssl_certificate.crt
```

There are two methods that you can use to update your SSL certificates:

- Using the FortiSOAR Admin CLI (`sudo csadm`). For more information on `csadm`, see the CLI Administration chapter in the "Administration Guide."
- Manually: You can use this method in case you face some issues with `csadm`.

Method 1: Using `csadm`

1. SSH to your FortiSOAR VM.
2. To deploy your certificate, type the following command:
   `# sudo csadm certs --deploy`
   You must then specify the following at the prompt:
   The complete path of the private key file of your ssl certificate.
   The complete path to the crt file of your ssl certificate.

Method 2: Manually

1. SSH to your FortiSOAR VM and login.
2. Copy your certificates to `/etc/nginx/ssl/`.
   Note: When you deploy a custom certificate, you must ensure that the SAN name in the certificate should match the hostname (with or without a wildcard). If it is an IP address, it should be of type IPAddress in SAN name field.
3. Edit the `cyops-api.conf` file that is located in the `/etc/nginx/conf.d` directory to update the `ssl_certificate` and `ssl_certificate_key` as follows:
   ```
   sudo vi /etc/nginx/conf.d/cyops-api.conf
   ssl_certificate /etc/nginx/ssl/yourCert.crt;
   ssl_certificate_key /etc/nginx/ssl/yourCert.key;
   ```
   For selinux permissions, run the following command:
   `# sudo restorecon -v -R /etc/nginx/ssl`
4. Edit the `/etc/cyops/config.yml` file:
   `sudo vi /etc/cyops/config.yml`
   Update `crudhub_host` to the DNS name specified in SSL Certificate.
5. Restart the **nginx** service using the following commands:
   `# sudo systemctl restart nginx`
6. Clear your browser cache and re-login to FortiSOAR after updating the SSL Certificate.

Adding a custom CA (self-signed) certificate in Rocky Linux or RHEL as a trusted certificate

If you are using a self-signed certificate, then add your custom CA certificate in the OS and python trust store as a trusted certificate, when you are using an offline repository, or you have agents that use self-signed certificates to communicate with your FortiSOAR instance.

A CA certificate that is self-signed is not trusted by default in any OS. Due to this, tools like OpenSSL clients, curl, wget, etc raise issues. Sometimes, you can use tool flags to bypass certificate checks, for example, using `-k` in curl to bypass the certificate check

To solve this issue, you can add your custom CA (self-signed) certificate in the OS as a trusted certificate as follows:

1. Import your custom CA (self-signed) certificate (including all intermediate certificates) into the OS CA store using the process defined in https://access.redhat.com/solutions/6339061.
2. Add the custom CA (self-signed) certificate in the OS
3. Verify the custom CA (self-signed) certificate is added as trusted certificates

Adding the custom CA (self-signed) certificate in the OS

You require to add the custom CA (self-signed) certificate to the destination Rocky Linux or RHEL system(s).

1. Run the following command to get your server certificate and store it in a certificate chain:
   `echo | sudo openssl s_client -connect localhost:8444 -showcerts 2>/dev/null | sudo sed -n '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/p' > self_signed_certificate.crt`
   For example, `echo | sudo openssl s_client -connect qa-offlinerepo.fortisoar.in:443 -showcerts 2>/dev/null | sudo sed -n '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/p' > self_signed_certificate.crt`
2. Copy the contents of `self_signed_certificate.crt`.
3. Edit the `cacert.pem` file:
   `sudo vi /opt/cyops-auth/.env/lib64/python3.9/site-packages/certifi/cacert.pem`
   And append the copied contents of `self_signed_certificate.crt`, then save the `cacert.pem` file.
   IMPORTANT: This step is not required for fresh installations of FortiSOAR, but is necessary for upgraded FortiSOAR systems
4. Copy the generated certificate to the OS trust store:
   `sudo cp /home/csadmin/self_signed_certificate.crt /etc/pki/ca-trust/source/anchors/`
5. Execute the following commands:
   ```
   sudo update-ca-trust enable
   sudo update-ca-trust extract
   ```

Verifying that the custom CA (self-signed) certificate is added as a trusted certificate in Rocky Linux or RHEL

Run the following commands to verify if your custom CA certificate is added as a trusted certificate in Rocky Linux or RHEL:

- `curl -v https://<offline-repo-server>`
- `wget https://<offline repo>/7.6.6/install-fortisoar-7.6.6.bin`

If both the above commands work without any certificate warnings or errors, it means that your custom CA certificate is added as trusted certificate in Rocky Linux or RHEL.

Setting up monitoring for your FortiSOAR system

It is recommended that you set up the following as part of your initial deployment and configuration process to monitor various important parts of your FortiSOAR system such as disk space, audit logs, execution logs, etc.

Setting up system monitoring

You should set up system monitoring for FortiSOAR, both in case of a single node system and High Availability (HA) clusters on the `System Configuration` page. To know more about the setting up thresholds and enabling notifications to effectively monitor various FortiSOAR system resources such as CPU, Disk Space and Memory utilization, and the statuses of various FortiSOAR services, see the System Configuration and Management chapter in the "Administration Guide."

Setting up purging for audit and playbook logs

FortiSOAR persists each workflow step inputs, outputs and error details for providing granular details of each action run, which is very useful for subsequent analysis and debugging. However, the Playbook Execution History data is significantly large and generates large volumes of data, which might not be useful after some point of time. Therefore, it is recommended that the retention period for the playbook logs should not be more than a few weeks and it is very important that you schedule purging for these logs at regular intervals.

FortiSOAR also audits every login, logout, record create, update, delete of records and other important activity on the system. These logs also might be useful for only a few years.

One must, therefore, configure a purge schedule for both the playbook and audit logs as per the organization's retention policy. This would help keeping the database and disk usage for these logs constant over time.

You can schedule purging, on a global level, for both audit logs and executed playbook logs. Scheduling purging of audit and executed playbook logs ensures that the logs are periodically cleared. For the procedure for enabling and scheduling purging, see the System Configuration and Management chapter in the "Administration Guide."

For additional information about monitoring your FortiSOAR system, see the Monitoring & Optimization chapter in the "Best Practices Guide."

Configuring High Availability or Disaster Recovery options

You can configure FortiSOAR with either an externalized PostgreSQL database or an internal PostgreSQL database. For both cases you can configure Active-Active or Active-Passive high availability clusters. For more information, see the High Availability Configuration and Maintenance chapter in the "Administration Guide."

FortiSOAR provides backup scripts that are scheduled to run at pre-defined intervals and take full database backup on a shared or backed up drive. For more information on backing up and restoring FortiSOAR, see the Back up and Restore FortiSOAR topic in the *Resilience & Recovery* chapter of the "Administration Guide."

Starting and stopping FortiSOAR Services

You will need to stop and start the FortiSOAR Services in the following cases:

- Update/Upgrade your SSL certificates
- Post-update, if playbooks are not working as expected
- Post-reboot, if the FortiSOAR Platform is not working as expected

To stop and start all the FortiSOAR services, use the FortiSOAR Admin CLI (`sudo csadm`). For more information on `csadm`, see the CLI Administration chapter in the "Administration Guide." You can run the `sudo csadm` command on any FortiSOAR machine using any terminal. Any user who has `sudo` permissions can run the `csadm` command.

To view the status of all FortiSOAR services, type: `# sudo csadm services --status`

To restart FortiSOAR services, type: `# sudo csadm services --restart`

To start FortiSOAR services, type: `# sudo csadm services --start`

To stop FortiSOAR services, type: `# sudo csadm services --stop`

Changing the FortiSOAR default database passwords

After you complete the FortiSOAR deployment procedure, you can change the default database passwords using the FortiSOAR Admin CLI:
`# sudo csadm db --change-passwd`

The script will prompt you for the new passwords for the Postgres DB, and you must appropriately enter the password that you want to set for the Postgres DB.

After running this script and changing the passwords, this script makes FortiSOAR use the new passwords and stores the passwords in an encrypted format. For more information on `csadm`, see the CLI Administration chapter in the "Administration Guide."

Setting up a proxy server to service all requests from FortiSOAR

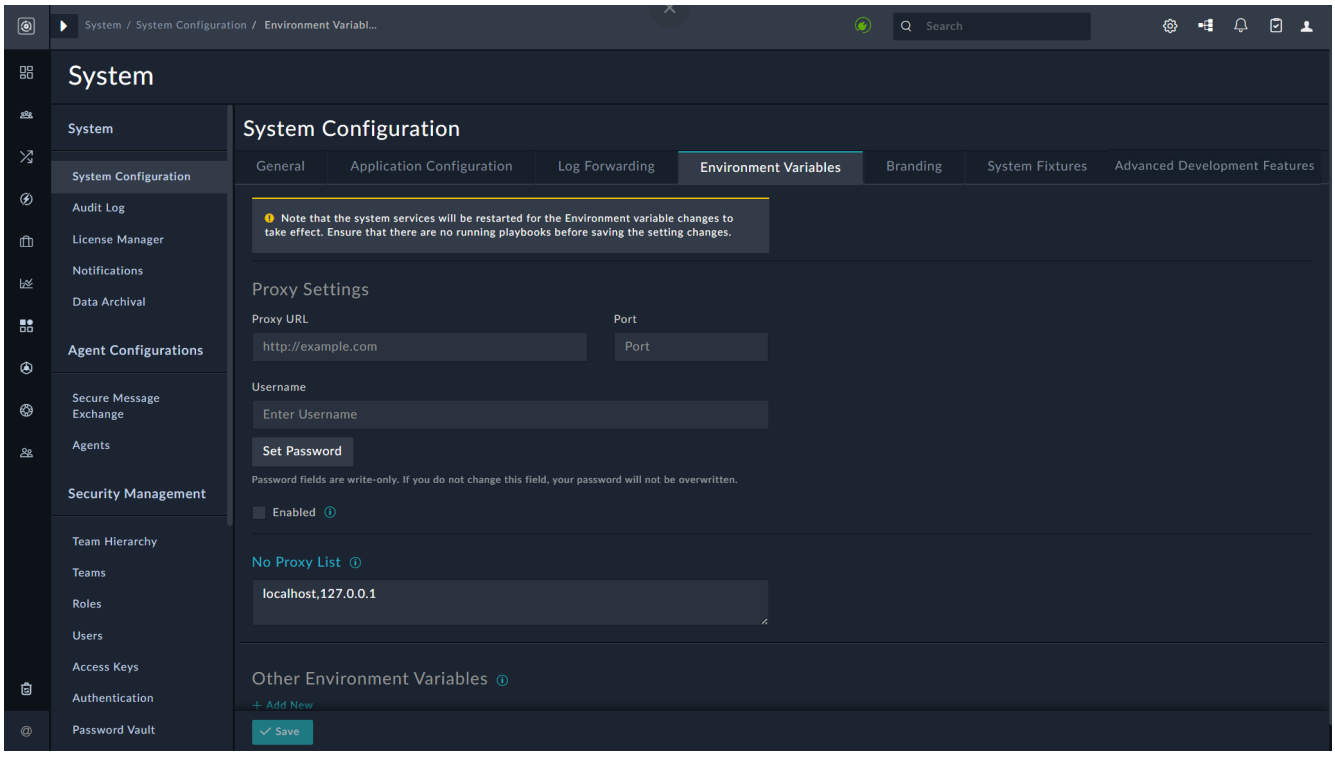
Your FortiSOAR instance would need access to the following endpoints on the public internet:

- For upgrading FortiSOAR, installing connectors, and accessing the widget library: https://repo.fortisoar.fortinet.com/
- For installing python dependencies for connectors: https://pypi.python.org
  Note: There is a parallel python repository also on repo.fortisoar.fortinet.com that can be used with some configuration if your organization does not approve pypi.
- For synchronization of FortiSOAR license details: https://globalupdate.fortinet.net
- For accessing any SaaS or API endpoint that you have configured, for example VirusTotal, and to which you require to be connected.

You must ensure that these endpoints are open from the organizations proxy. You can configure your proxy for the first time when you run the FortiSOAR Configuration Wizard. If you subsequently require to change the proxy, then you can use the `sudo csadm` CLI commands or use the UI as specified in the following procedure.

You can use the `# sudo csadm network set-https-proxy` command to set the proxy for both your system and the web services. For more information on `csadm`, see the CLI Administration chapter in the "Administration Guide."

Use the **Environment Variables** tab on the `System Configuration` page to configure proxy settings forFortiSOAR and to define any other environment variables.



External web pages that you open (for example, from a link included in the description field of an alert) or view (for example, using the iFrame Widget) in FortiSOAR goes through the configured proxy server if you have configured the proxy in the web browser's settings. If the proxy is not configured in the web browser's settings, then the external web pages are opened directly without using the configured proxy server.

Configuring Proxy Settings and environment variables

Use the following procedure to add proxy details and environment variables forFortiSOAR:

1. Log on to FortiSOAR as an administrator.
2. Click **Setting** to open the `System Configuration` page.
3. Click the **Environment Variables** tab.
4. To set up an HTTP or HTTPS proxy to serve all HTTP/HTTPS requests fromFortiSOAR, enter the following details in the `Proxy Settings` section on the `Environment Variables` page:
   a. In the **Proxy URL** field, enter the HTTP proxy server IP and in the Port field, optionally enter the HTTP proxy server port.
   Note: If you do not specify `HTTP` or `HTTPS` in the Proxy URL field, then by default `HTTP` is set.
   b. In the **Username** field, enter the username used to access the HTTP proxy server (if not applicable leave this field blank).
   c. Click **Set Password** to enter the password used to access the HTTP proxy server (if not applicable leave this field blank).
   d. Verify that the **Enabled** check box is selected to apply the proxy settings that you have specified. If you clear the **Enabled** check box, then the proxy settings that you have specified are saved but not applied.
5. (Optional) In the **No Proxy List** text box, enter a comma-separated list of addresses that do not require to be routed through a proxy server.
   For example, enter `http://example.com` in the **No Proxy List** text box.
   `localhost` and `127.0.0.1` are added by default to the no proxy list by the system.

   In the case of a FortiSOAR High Availability cluster environment, if a proxy is configured on a node in the cluster, then it is advisable to add other nodes in the 'no_proxy' list or ensure that the other nodes in the cluster are accessible to that node over the proxy.

6. (Optional) In the `Other Environment Variables` section, you can add environmental variables and setup proxies for other protocols, such as FTP (other than HTTP or HTTPS) in a key-value pair. Click the **+Add New** link and the **Key** and **Value** text boxes will be displayed. Enter the protocol for which you want to set up the proxy in the **Key** text box and its value in the **Value** box.
   For example, enter `FTP` in the **Key** field and `1.1.1.1` in the **Value** field.
7. Click **Save** to save your proxy server settings or the environment variables you have added.

The proxy values that you save and apply for the HTTP proxy server are also automatically saved and applied to the HTTPS proxy server

Backing up the data encryption keys

Encryption keys are used to encrypt data in FortiSOAR. When you install FortiSOAR for the first-time default encryption keys are added, which are unique per instance; therefore, you do not need to change the encryption keys.

Important: It is highly recommended that you back up the encryption keys, `.Defuse.key` and `PASSWORD_ENCRYPTION_KEY`. The `.Defuse.key` can be backed up from the `/opt/cyops/config/cyops-api` file. The `.Defuse.key` is a dot file, therefore you need to use `sudo ls -la /opt/cyops/configs/cyops-api/` to list/view the file and store the data encryption keys securely in a Password Manager or Vault. The `PASSWORD_ENCRYPTION_KEY` can be backed up from the `/opt/cyops/configs/keys/PASSWORD_ENCRYPTION_KEY` file.

> ⚠ Once you encrypt your production data in FortiSOAR using the encryption keys, you should not change those keys again; since if your encryption keys are changed, this might result in the loss of previously encrypted production data. If you do require to change the encryption keys, then contact FortiSOAR Support.

Configuring a reverse proxy (Apache proxy server)

If you have set up a reverse proxy, an Apache proxy server, in your environment, then configure this reverse proxy server so that the live sync functionality works, as follows:

Important: This procedure applies only to an Apache proxy server. You can enable any other reverse proxy using a similar pattern to support the web socket functionality.

Update the proxy configuration file on your proxy server as follows:

```
<VirtualHost *:80>

 #ServerName
 SSLProxyEngine on

 SSLProxyCheckPeerCN on
 SSLProxyCheckPeerName on

 /** Section required for enabling Websockets **/
 RewriteEngine On
 RewriteCond %{HTTP:Upgrade} =websocket [NC]
 RewriteRule /(.*)           wss://<FortiSOAR-URL>/$1 [P,L]
 /** End Section **/

 ProxyPass / https://<FortiSOAR-URL>/
 ProxyPassReverse / https://<FortiSOAR-URL>/

 RequestHeader set Host "<FortiSOAR-URL>"
```

```
 RequestHeader set Origin "https://<FortiSOAR-URL>"
</VirtualHost>
```

Troubleshooting FortiSOAR Issues

Troubleshooting issues occurring in FortiSOAR due to insufficient space

You can face deployment or upgrade failures due to insufficient space. If you have limited partition size for `/dev/mapper/vgapp-csapps`, then FortiSOAR upgrade might fail. Therefore, before you upgrade your FortiSOAR system, you must ensure that you increase the partition size to a minimum of 4 GB for `/dev/mapper/vgapp-csapps` to prevent potential loss of backups.

You might also experience any of the following symptoms when the disk space of the database on which FortiSOAR is running gets full:

- Users are unable to log into FortiSOAR.
- All FortiSOAR services might stop working, as they cannot write to their respective log files.
  For example, the PostgreSQL service fails to start when the PostgreSQL database disk is full.

Insufficient space in FortiSOAR can be caused due to a number of reasons, some of them are as follows:

- Increase in the number of log files in `/var/log/` and `/var/log/cyops`
- `/home` drive is full
- Increase in the data in the database
- PostgreSQL database disk is full

> You can fix the insufficient space issue using the `sudo csadm system disk expand-lv` command to extend a logical volume to occupy space that is available in its own volume group. Or, if a new disk is attached, then a single partition is created and the logical volume is expanded to occupy this partition according to the specified size in GB. It is important to note that the '`sudo csadm system disk`' subcommand supports only one volume group per disk.
> For additional information on this command, see the CLI Administration chapter in the "Administration Guide."

You can also use the following methods to fix this issue:

- Resolution 1: Extend the disk space by adding a new disk and then extending the logical volume in the new disk.
  NOTE: To support disk sizes larger than 2 TB, FortiSOAR OVAs starting with the 7.5.0 release come pre-configured with a GPT-based disk layout. Previously, FortiSOAR OVAs were shipped with an MBR-based disk layout, which limited disk management to a size of 2TB. If you already have a FortiSOAR instance and need a partition larger than 2 TB, we recommend creating a new FortiSOAR VM on release 7.5.0 or later and utilizing the Export and Import wizards to migrate your data from the old instance to the new one. This is required as FortiSOAR does not support a combination of MBR and GPT partitions.
- Resolution 2: Extend the logical volume by using the free space that is already available in the volume group.
- Resolution 3: Extend the logical volume on the existing disk without adding a new disk, if sufficient unallocated space is available on the existing disk.

- Resolution 4: Procedure to be followed when the Postgres service has failed which could be due to the PostgreSQL database disk is full.

Note: Perform the following procedures using the `sudo` command.

Resolution 1

Perform the following steps to extend your disk space by adding a new disk and then extending the logical volume (LVM) in the new disk.

1. Stop all FortiSOAR services using the following command:
   `# sudo csadm services --stop`
2. Add the new disk drive with the required size on the instance.
3. Run the following command to check the size of the newly added and unpartitioned disk.
   `# lsblk`
   This command displays the size of newly added and unpartitioned disk. In this example, `sde` is the newly added disk:
   ```
   NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   sda 8:0 0 8G 0 disk
   '-sda1 8:1 0 8G 0 part /
   sdb 8:16 0 100G 0 disk
   '-sdb1 8:17 0 100G 0 part
   sdc 8:32 0 300G 0 disk
   '-sdc1 8:33 0 300G 0 part
   '-vg_repo-lvol0 253:0 0 500G 0 lvm /repos
   sdd 8:48 0 200G 0 disk
   '-vg_repo-lvol0 253:0 0 500G 0 lvm /repos
   sde 8:64 0 200G 0 disk #This is the new attached partition
   sr0 11:0 1 1024M 0 rom
   ```
4. Use the `sudo csadm system disk expand-lv` command to extend a logical volume by the specified size. It is recommended that you use the `--validate` option to validate the inputs you have passed to the `sudo csadm system disk expand-lv` command. For example:
   `# sudo csadm system disk expand-lv --validate --logical-volume <LVM_name> --disk <disk_name> --size <disk_size>`
   The `--validate` option provides a summary of changes that will be made after running this command. This summary displays the current lvm size, the current free space on the disk, and the expected lvm size following the execution of the command, and also displays an appropriate error if the requested disk space for expansion is less than the free space that is available.
   The `--size` option is optional and must be specified in GBs; if not specified then the whole disk will be attached to that logical volume.
   For example, if you want to expand the logical volume, `pgsql`, by 10 GB, you can use the following command with the validate option:
   `# sudo csadm system disk expand-lv --validate --logical-volume pgsql --disk sda --size 10`
   If the summary displayed is acceptable, then run the command without the `--validate` option:
   `# sudo csadm system disk expand-lv --logical-volume pgsql --disk sda --size 10`
   Note: You must extend the disk size to less than or equal to the total size of the volume group.
5. Run the following command to check if the volume size is extended:
   `# sudo df -h /dev/mapper/<LVM_name>`
6. Start all FortiSOAR services using the following command:
   `# sudo csadm services --start`

Resolution 2

Perform the following steps to extend the logical volume by using the free space that is already available in the volume group:

1. Stop all FortiSOAR services using the following command:
   `# sudo csadm services --stop`
2. Use the `sudo csadm system disk expand-lv` command to extend a logical volume by the specified size. It is recommended that you use the `--validate` option to validate the inputs you have passed to the `csadm system disk expand-lv` command. For example:
   `# sudo csadm system disk expand-lv --validate --logical-volume <LVM_name> --use-vg --size <disk_size>`
   The `--validate` option provides a summary of changes that will be made after running this command. This summary displays the current lvm size, the current free space on the disk, and the expected lvm size following the execution of the command, and also displays an appropriate error if the requested disk space for expansion is less than the free space that is available.
   The `--size` option is optional and must be specified in GBs; if not specified then the all the free space in the volume group will be assigned to that logical volume.
   For example, if you want to expand the logical volume, `pgsql`, by 10 GB, you can use the following command with the validate option:
   `# sudo csadm system disk expand-lv --validate --logical-volume pgsql --use-vg --size 10`
   If the summary displayed is acceptable, then run the command without the `--validate` option:
   `# sudo csadm system disk expand-lv --logical-volume pgsql --use-vg --size 10`
   Note: You must extend the disk size to less than or equal to the total size of the volume group.
3. Run the following command to check if the volume size is extended:
   `# sudo df -h /dev/mapper/<LVM_name>`
4. Start all FortiSOAR services using the following command:
   `# sudo csadm services --start`

Resolution 3

Perform the following steps to extend the logical volume on the existing disk without adding a new disk, if sufficient unallocated space is available on the existing disk. Before proceeding further, it is recommended that you find out whether sufficient unallocated space is available on existing disk using the following command:
`# sudo parted /dev/<disk_name> print free`

For example, running the `# sudo parted /dev/sdb print free` command will display the following:

```
Model: VMware Virtual disk (scsi)
       Disk /dev/sdb: 215GB
       Sector size (logical/physical): 512B/512B
       Partition Table: msdos
       Disk Flags:
       Number  Start   End     Size    Type     File system  Flags

       32.3kB  1049kB  1016kB  Free Space
       1      1049kB  161GB   161GB   primary              lvm
       161GB   215GB   53.7GB  Free Space
```

The “Free Space” in above the output refers to unallocated space. If the mentioned size is sufficient, then perform the following steps:

1. Stop all FortiSOAR services using the following command:
   `# sudo csadm services --stop`

2. Use the `sudo csadm system disk expand-lv` command to extend a logical volume by the specified size. It is recommended that you use the `--validate` option to validate the inputs you have passed to the `sudo csadm system disk expand-lv` command. For example:
   `# sudo csadm system disk expand-lv --validate --logical-volume <LVM_name> --disk <disk_name> --size <disk_size>`
   The `--validate` option provides a summary of changes that will be made after running this command. This summary displays the current lvm size, the current free space on the disk, and the expected lvm size following the execution of the command, and also displays an appropriate error if the requested disk space for expansion is less than the free space that is available.
   The `--size` option is optional and must be specified in GBs; if not specified then the whole disk will be attached to that logical volume.
   For example, if you want to expand the logical volume, `pgsql`, by 10 GB, you can use the following command with the validate option:
   `# sudo csadm system disk expand-lv --validate --logical-volume pgsql --disk sda --size 10`
   If the summary displayed is acceptable, then run the command without the `--validate` option:
   `# sudo csadm system disk expand-lv --logical-volume pgsql --disk sda --size 10`
   Note: You must extend the disk size to less than or equal to the total size of the volume group.
3. Start all FortiSOAR services using the following command:
   `# sudo csadm services --start`

Resolution 4

Perform the following steps when you notice that your Postgres service has failed which could be due to the PostgreSQL database disk is full.

1. Run the following command and check the disk space in `/var/lib/pgsql/`:
   `# sudo df -h /var/lib/pgsql/`
   If there is no space or less than 1 GB space left in `/var/lib/pgsql/`, then the `psql` service will not start due to the space issue.
2. `# sudo cd /var/lib/pgsql/`
   `# sudo ls -lrth`
   You will observe that there is a file named `resv_space` that is taking up 1GB of space.
3. Stop all the services using the following command:
   `sudo csadm services --stop`
4. Move the `resv_space` file from `/var/lib/pgsql/` to `/home/csadmin` using the following command:
   `# sudo mv /var/lib/pgsql/resv_space /home/csadmin`
5. Check the disk space in `/var/lib/pgsql/` again:
   `# sudo df -h /var/lib/pgsql/`
   You will observe that 1GB of space has been freed.
6. Start all the services again using the following command:
   `sudo csadm services --start`
7. If this is a development or staging instance and you cannot extend the disk space, then the disk space can also be freed up by purging workflow logs. For information on purging workflow logs, see the System Configuration and Management chapter in the "Administration Guide."
8. Move the `resv_space` file from `/home/csadmin` back to `/var/lib/pgsql/`:
   `# sudo mv /home/csadmin/resv_space /var/lib/pgsql/`

Increasing the disk space for record storage in case of AWS AMI deployment

If you are deploying a fresh instance of FortiSOAR in AWS with AMI, and you require larger disk space for record storage, do the following:

1. Increase the size Elastic and PostgreSQL disks, for example `/dev/sdg` (Elastic) and `/dev/sdf` (postgresql).
2. Provision your FortiSOAR instance and complete running the FortiSOAR VM Configuration Wizard.
3. Start an SSH session and check the allocated disk space the following command:
   `sudo df -h`
4. If the newly increased disk space that is allocated is sufficient, then no changes are required, else you can use the `sudo csadm system` command to increase the partition size and allocate unused space. For more information on the `sudo csadm` command, see the CLI Administration chapter in the "Administration Guide."

Troubleshooting Deployment Issues

The FortiSOAR Virtual Appliance deployment on ESX is failing

Resolution:

1. Verify that FortiSOAR Virtual Appliance file that you have downloaded is not corrupted by running a `# sudo md5sum` command for the FortiSOAR Virtual Appliance.
2. Check that the ESX server has fulfilled all prerequisites specified for the VM. Refer to the *Planning* section for details.
3. If both points 1 and 2 are ok, contact VMWare support.

Cannot access the FortiSOAR portal

Resolution:

1. Check the ESX network to which FortiSOAR VM is connected.
2. Check if the IP address is assigned to your FortiSOAR VM, in the case of DHCP or static IP addresses.
   Refer to the *Editing the VM configuration* section for more information on Setting a static IP and Determining your DHCP IP address.

Cannot login to the FortiSOAR platform

Resolution:

1. Check if you are using the correct credentials that have been provided to you by FortiSOAR Customer Support.
2. SSH to the VM where you have deployed FortiSOAR to check the status of `cyops-auth` service:
   `sudo csadm services --status cyops-auth`
   The `cyops-auth` service must be running.

3. If both points 1 and 2 are ok, and the `cyops-auth` service is running, contact FortiSOAR support.

Getting a 502 error when you click on the Reports tab

Resolution:

1. SSH to the VM where you have installed FortiSOAR.
2. Log in using the SSH credentials.
3. Run the `$ sudo` command.
   Enter your FortiSOAR password.
4. Run the `$ sudo systemctl restart tomcat` command.
5. Run the `$ sudo systemctl restart nginx` command.

If the issue yet does not get resolved, contact FortiSOAR support.

Troubleshooting Upgrade Issues

For the procedure on how to upgrade to FortiSOAR 7.6.6, see the *Upgrading a FortiSOAR enterprise instance to 7.6.6* section in the "Upgrade Guide."

Post license renewal you cannot log into FortiSOAR

If you have requested for a license from FortiSOAR with lesser number of users than your existing users, you cannot log onto FortiSOAR post upgrade.

For example, when you had requested a license from FortiSOAR, you had requested for a license for 10 users; however, you have 15 users existing in your system, you will not be able to log onto FortiSOAR post-upgrade. So, it is very important for you to provide the correct number of users while requesting for a license from FortiSOAR.

Resolution:

Contact FortiSOAR Support to generate a new license for you with the correct number of users.

Failure to upgrade FortiSOAR

In case you face a failure while trying to upgrade FortiSOAR, then perform the following steps:

Resolution:

1. To gather logs and send them to FortiSOAR Support, do the following:
   a. SSH to machine and type the following command:
      `# sudo csadm log --collect`
   b. Specify the path where you want to collect the logs. By default, the logs are collected in the `/tmp/` folder.
      A file named `fortisoar-logs.tar.gz.gpg` gets generated in the path you have specified. Send this file to FortiSOAR Support.

2. Revert the snapshot of your system to the latest working state. You must take a snapshot of your system before you attempt to upgrade FortiSOAR on your system.
   Follow the steps mentioned in the documentation of your platform for taking a snapshot and reverting to the current snapshot.

Post-upgrade your playbooks fail to execute, and the playbooks are also not listed in the executed playbooks log

If you have not cleaned the workflow history prior to upgrading your FortiSOAR version, and if you have a large number of records in the workflow history (> 50000), then the overall upgrade time would increase, and this issue might occur. For the method to clean the workflow execution run history, see the "Auto-Cleanup of Workflow Execution History" topic in the *Playbook Execution and Debugging* chapter in the "Playbooks Guide."

Resolution

Perform the following steps:

1. Check the install logs to see if errors such as the following are present:
   `psycopg2.InternalError: missing chunk number 0 for toast value 1502832 in pg_toast_17046`
2. Connect to the postgres db and run the following command:
   `# sudo REINDEX table pg_toast.pg_toast_XXXXX;`
   (where XXXXX is the number from the error message present in step 1).
3. Run the following commands:
   ```
   # sudo cd /opt/cyops-workflow/sealab
   $ sudo -u nginx /opt/cyops-workflow/.env/bin/python3 manage.py migrate
   ```
   Note: If `manage.py` fails again for `pg_toast` value (error mentioned in step 1), then you must execute steps 2 and 3 again with the new `pg_toast` value mentioned in the error, till the `manage.py` executes successfully.

Login and logout events are not audited after you have upgraded your FortiSOAR version

After you have upgraded FortiSOAR on your system, you observe the following error in the `auditlog` log file located at `/var/log/cyops/cyops-gateway/auditlog.log` :

`ERROR c.c.a.service.RecordLogService.processRecordLogs - 500 Internal Server Error`, and you also do not see any `Login` and `Logout` events in **Audit Logs** on FortiSOAR UI, then perform the steps mentioned in the resolution.

Resolution

To resolve this issue and include login and login events in audit logs, run the following commands on your FortiSOAR VM:

```
yes | sudo cp /opt/cyops-workflow/sealab/.envdir/APPLIANCE_PUBLIC_KEY /etc/cyops/APPLIANCE_
PUBLIC_KEY
yes | sudo cp /opt/cyops-workflow/sealab/.envdir/APPLIANCE_PRIVATE_KEY /etc/cyops/APPLIANCE_
PRIVATE_KEY
sudo chmod -R 644 /etc/cyops/APPLIANCE_*_KEY
sudo chown tomcat:tomcat /etc/cyops/APPLIANCE_*_KEY
sudo systemctl restart tomcat
```

www.fortinet.com