ThreatConnect® Installation Guide: Containerized Deployment

Software Version 7.9.2

Technical Guide

June 26, 2025

10032-09 EN Rev. A

ThreatConnect, Inc.
3865 Wilson Blvd., Suite 550, Arlington, VA 22203

P: 1.800.965.2708 | F: 703.229.4489
www.ThreatConnect.com

Table of Contents

Overview

This guide describes how to install ThreatConnect®. As of ThreatConnect version 7.5, you will no longer be required to install Java®, Python®, OpenSearch®, and Redis® as part of the ThreatConnect installation process. Instead, all of this software, along with ThreatConnect, is now packaged together in a containerized solution using Docker® or Podman.

Important: The containerized deployment was tested on AlmaLinux OS™ and is the standard deployment method for all production and non-production systems. For instructions on installing ThreatConnect and having it run on an operating system (OS), see *ThreatConnect Installation Guide: Linux Operating System Legacy Deployment*.

Important: Instances running on Red Hat® Enterprise Linux® (RHEL) 8 must use Podman for containerized deployments. Rootless Podman is recommended.

Important: The `.env` file holds all passwords and configurations for the containerized deployment. Once the container is running, the `.env` file can be purged.

System Requirements

Hardware

ThreatConnect requires a server, virtual or physical, that meets the specifications listed in Tables 1–3.

Note: Multi-server installations are for advanced users only, who should consult with ThreatConnect as to the correct sizing that will meet their needs. See *ThreatConnect System Requirements* for additional information.

Table 1

	Memory Min (GB)[1],[2]	Min CPU Cores / vCPUs (2GHz)[3]	Estimated Storage (GB) [4],[5]
ThreatConnect Application	64	16	300
Containerized Redis	8	2	20
Containerized OpenSearch	32	12	60
Containerized Database	64	16	120

Important: The following guidelines apply to production deployments:

- The ThreatConnect Application and Redis can be deployed to the same server (virtual or physical).
- OpenSearch containers should be deployed to a dedicated server.
- Database containers should be deployed to a dedicated server.

[1]Allocated to ThreatConnect containers; the OS will need additional space.
[2]While Java virtual machines will be allocated memory, there is some allowance for additional memory available for Feed and Playbook Apps.
[3]While Java virtual machines will be allocated memory, there is some allowance for additional memory available for Feed and Playbook Apps.
[4]High IOPS, ideally SSDs, are preferred.
[5]ThreatConnect must be installed on an ext4 or XFS partition.

Table 2

	Highly Available Document Storage (usually network-mounted storage)
Document Storage	Equal to the desired capacity of documents stored

Table 3

	Memory Minimum (GB)	**Memory Recommended (GB)**
Swap Space	4	8

Note: As the number of users increases, or as the frequency or complexity of automated analysis increases, the need to increase system resources will likely occur.

Installation Steps

Important: Steps 1–4 apply to all Docker and Podman deployments.

Step 1: Download ThreatConnect Docker ZIP File

Note: You must complete this step on all hosts intended to run ThreatConnect or some component of ThreatConnect.

Download `threatconnect-docker-v<version number>.zip`, where `<version number>` is a placeholder value for the version number associated with the ThreatConnect version you are installing. For example, to download the ThreatConnect Docker ZIP file for ThreatConnect 7.9.2, run the following commands:

```
None
cd /opt
unzip threatconnect-docker-v7.9.2.zip
cd /opt/threatconnect-docker
```

Step 2: Update Environment Variables

Note: You must complete this step on all hosts that will run ThreatConnect or some component of ThreatConnect.

Copy `.env.sample` to your `.env` file, and then update each variable in your `.env` file with the appropriate value. For descriptions of the values that you must provide in your `.env` file, reference the comments in that file.

```
None
cp /opt/threatconnect-docker/.env.sample /opt/threatconnect-docker/.env
```

Step 3: Install ThreatConnect License

Note: You must complete this step on the hosts that will run messaging (**tc-mon**), applications (**tc-app**), and Playbooks (**tc-job**).

Place your ThreatConnect license XML file into `/opt/threatconnect-docker/config/license.xml`.

Step 4: Add Certificates

Note: You must complete this step on the hosts that will run messaging (**tc-mon**), applications (**tc-app**), and Playbooks (**tc-job**).

Add the two required certificates to the `certs` folder. These are the certificate authority–signed (CA-signed) certificate and private key:

```
None
/opt/threatconnect-docker/certs/fullchain.pem

/opt/threatconnect-docker/certs/privkey.pem
```

If you do not have a CA-signed certificate, follow these steps to generate self-signed certificates:

1. Create a certificate authority (replace the `<country>`, `<state>`, `<city>`, `<company>`, and `<department>` placeholder values):

```
None
mkdir certs && cd certs
openssl genrsa -out my-root-ca-key.pem 4096
openssl req -new -x509 -sha256 -key my-root-ca-key.pem \
      -subj "/C=<country>/ST=<state>/L=<city>/O=<company>/OU=<department>/CN=My
Root Authority" \
      -out my-root-ca.pem -days 3650
```

2. Create a private key:

```
None
openssl genrsa -out privkey.pem 4096
```

3. Create a certificate signing request (replace the `<country>`, `<state>`, `<city>`, `<company>`, `<department>`, and `<FQDN/IP of server>` placeholder values):

```
None
openssl req -new -sha256 -key privkey.pem \
       -subj
"/C=<country>/ST=<state>/L=<city>/O=<company>/OU=<department>/CN=<FQDN/IP of
server>" \
       -out <FQDN/IP of Server>.csr
```

4. Create a new file for alternate names in `alt-names.ext` (replace the `<FQDN/IP of server>` placeholder value):

```
None
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = <FQDN/IP of server>
IP.1 = <FQDN/IP of server>
```

5. Create a certificate signed by your CA (replace the `<FQDN/IP of server>` placeholder value):

```
None
openssl x509 -req -in <FQDN/IP of server>.csr -CA my-root-ca.pem \
       -CAkey my-root-ca-key.pem -CAcreateserial \
       -out fullchain.pem -days 398 -sha256 \
       -extfile alt-names.ext
```

6. Append the root CA certificate to `fullchain.pem`:

```
None
cat my-root-ca.pem >> fullchain.pem
```

7. Update `CUSTOM_CA_PEM_FILE` in your `.env` file as follows:

```
None
CUSTOM_CA_PEM_FILE=fullchain.pem
```

Step 5: Install Docker

Important: Steps 5 and 6 apply to Docker deployments only. For Podman deployments, skip to Step 7.

Note: You must complete this step on all hosts that will run ThreatConnect or some component of ThreatConnect.

Run the following commands to install Docker:

```
None
yum-config-manager --add-repo \
  https://download.docker.com/linux/centos/docker-ce.repo
yum install docker-ce docker-ce-cli containerd.io
systemctl start docker.service
systemctl enable docker.service
docker version
```

Step 6: Install Docker Compose

Important: Steps 5 and 6 apply to Docker deployments only. For Podman deployments, skip to Step 7.

Note: You must complete this step on all hosts that will run ThreatConnect or some component of ThreatConnect.

Run the following commands to install Docker Compose:

```
None
curl -SL
https://github.com/docker/compose/releases/download/v2.24.5/docker-compose-linu
x-x86_64 \
  -o /usr/local/bin/docker-compose
ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
chmod 755 /usr/local/bin/docker-compose
docker-compose version
```

Step 7: Install Podman

Important: Steps 7 and 8 apply to Podman deployments only. For Docker deployments, skip to Step 9.

Note: You must complete this step on all hosts that will run ThreatConnect or some component of ThreatConnect.

Run the following commands to install Podman:

```
None
dnf module install -y podman
export PODMAN_PATH=$(which podman)
ln -s $PODMAN_PATH /usr/bin/docker
```

Step 8: Install Podman Compose

Important: Steps 7 and 8 apply to Podman deployments only. For Docker deployments, skip to Step 9.

Note: You must complete this step on all hosts that will run ThreatConnect or some component of ThreatConnect.

Run the following commands to install Podman Compose:

```
None
dnf install python3.11 python3.11-pip
pip3.11 install podman-compose
ln -s /usr/local/bin/podman-compose /usr/bin/docker-compose
```

Step 9: Install AWS CLI

Note: You must complete this step on all hosts that will run ThreatConnect or some component of ThreatConnect.

Amazon Web Services® Command Line Interface (AWS CLI) is used to download Docker images directly from ThreatConnect's Elastic® Container Registry (ECR). Run the following commands to install AWS CLI:

```
None
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" \
       -o "awscliv2.zip" &&\
       unzip awscliv2.zip &&\
       ./aws/install
```

Step 10: Increase vm.max_map_count

Note: You must complete this step on the host that will run OpenSearch.

Run the following commands to increase `vm.max_map_count`:

```
None
echo 'vm.max_map_count=262144' >> /etc/sysctl.conf
echo 'net.ipv4.ip_unprivileged_port_start=25' >> /etc/sysctl.conf
sysctl -p
```

Step 11: Fix Shell Scripts

Note: You must complete this step on all hosts that will run ThreatConnect or some component of ThreatConnect.

Reformat and change permissions on shell scripts:

```
None
cd /opt/threatconnect-docker
sed -i 's/\r$//' load_schema.sh
chmod 755 load_schema.sh
sed -i 's/\r$//' docker-entrypoint.d/00_init.sh
chmod 755 docker-entrypoint.d/00_init.sh
sed -i 's/\r$//' docker-entrypoint.d/98_custom_ca.sh
chmod 755 docker-entrypoint.d/98_custom_ca.sh
sed -i 's/\r$//' docker-entrypoint.d/99_deploy.sh
chmod 755 docker-entrypoint.d/99_deploy.sh
sed -i 's/\r$//' create_db_user.sh
chmod 755 create_db_user.sh
sed -i 's/\r$//' set_opensearch_password.sh
```

```
chmod 755 set_opensearch_password.sh
sed -i 's/\r$//' tc-containers.sh
chmod 755 tc-containers.sh
sed -i 's/\r$//' docker-entrypoint.d/pythonwrapper-3.11
chmod 755 docker-entrypoint.d/pythonwrapper-3.11
sed -i 's/\r$//' docker-entrypoint.d/pythonwrapper-3.6
chmod 755 docker-entrypoint.d/pythonwrapper-3.6
```

Step 12: Create Users

Run the following commands to create **threatconnect** and **tc-job** user accounts:

```
None
adduser --uid 1000 threatconnect
adduser --uid 1001 tc-job
```

Step 13: Install Random-Number Generation Tools

Run the following commands to install random-number generation tools:

```
None
dnf install -y rng-tools
systemctl enable rngd
systemctl start rngd
```

Step 14: Configure Rootless Podman

Important: Steps 14 and 15 apply to rootless Podman deployments only. For Docker deployments and deployments running Podman as root, skip to Step 16.

Note: You must complete this step on all hosts that will run ThreatConnect or some component of ThreatConnect.

1. Execute `enable-linger` on user **threatconnect**:

```
None
loginctl enable-linger threatconnect
```

2. Change ownership of **threatconnect-docker** to **threatconnect**:

```
None
chown threatconnect:threatconnect -R /opt/threatconnect-docker
```

3. Log in as the **threatconnect** user:

```
None
su - threatconnect
```

Important: For rootless Podman deployments, perform the rest of the steps in this guide as non-root user **threatconnect**.

Step 15: Configure Podman Home Container

Important: Steps 14 and 15 apply to rootless Podman deployments only. For Docker deployments and deployments running Podman as root, skip to Step 16.

Run the following commands to configure the Podman home container:

```
None
cd /opt/threatconnect-docker
mkdir run
echo "export XDG_DATA_HOME=/opt/threatconnect-docker/run" >>
/home/threatconnect/.bashrc
source /home/threatconnect/.bashrc
```

Step 16: Log Into ThreatConnect's ECR

Configure AWS CLI using the credentials your ThreatConnect Customer Success Manager shared with you:

Important: If your system is located in a time zone other than U.S. East, you can replace `us-east-1` with a different AWS region before running these commands. The following AWS regions are supported at this time: `us-east-1`, `eu-central-1`, and `ap-southeast-2`.

```
None
/usr/local/bin/aws configure
Access Key ID:****
Secret Access Key:****
Region:us-east-1
```

```
None
docker login \
       -u AWS \
       -p $(/usr/local/bin/aws ecr get-login-password --region us-east-1) \
       373319941383.dkr.ecr.us-east-1.amazonaws.com
```

Step 17: Configure OpenSearch Data Folder

Note: This is an optional step that can be performed on the host that will run OpenSearch.

1. Create the OpenSearch mount folders:

```
None
mkdir /opt/threatconnect-docker/opensearch-data
mkdir /opt/threatconnect-docker/opensearch-logs
mkdir /opt/threatconnect-docker/opensearch-snapshots
```

2. Run the following additional command only if you are running rootless Podman:

```
None
docker unshare chown 1000:1000 -R /opt/threatconnect-docker/opensearch-*
```

3. Set variables in your `.env` file as follows:

```
None
OPENSEARCH_DATA=/opt/threatconnect-docker/opensearch-data
OPENSEARCH_LOGS=/opt/threatconnect-docker/opensearch-logs
OPENSEARCH_SNAPSHOTS=/opt/threatconnect-docker/opensearch-snapshots
```

Step 18: Configure Log Folders

Note: This is an optional step that can be performed on the hosts that will run messaging (**tc-mon**), applications (**tc-app**), and Playbooks (**tc-job**).

1. Create log folders for **tc-mon**, **tc-app**, and **tc-job**:

```
None
mkdir -p /opt/threatconnect-docker/logs/tc-mon
mkdir -p /opt/threatconnect-docker/logs/tc-app
mkdir -p /opt/threatconnect-docker/logs/tc-job
chown threatconnect:threatconnect -R /opt/threatconnect-docker/logs
```

2. Run the following additional command only if you are running rootless Podman:

```
None
docker unshare chown 1000:1000 -R /opt/threatconnect-docker/logs
```

3. Set the log file locations in your `.env` file as follows:

```
None
TC_MON_LOGS=/opt/threatconnect-docker/logs/tc-mon
TC_APP_LOGS=/opt/threatconnect-docker/logs/tc-app
TC_JOB_LOGS=/opt/threatconnect-docker/logs/tc-job
```

Step 19: Configure ThreatConnect Storage Data

Note: You must complete this step on the hosts that will run messaging (**tc-mon**), applications (**tc-app**), and Playbooks (**tc-job**).

1. If you intend to set `documentStorageType=LOCAL`, you must create a directory to which ThreatConnect can save documents. For example, if you want ThreatConnect to save documents to `/threatconnect-data/storage`, create that directory and make sure it is owned by the user your ThreatConnect container will use (`1000`):

```
None
mkdir /opt/threatconnect-docker/docstorage
chown 1000:1000 -R /opt/threatconnect-docker/docstorage
```

2. Run the following additional command only if you are running rootless Podman:

```
None
docker unshare chown 1000:1000 -R /opt/threatconnect-docker/docstorage
```

3. Then set `TC_DOC_STORAGE` in your `.env` file as follows:

```
None
TC_DOC_STORAGE=/opt/threatconnect-docker/docstorage
```

Step 20: Configure TC Exchange Data

Note: This is an optional step that can be performed on the hosts that will run messaging (**tc-mon**), applications (**tc-app**), and Playbooks (**tc-job**).

1. Create the TC Exchange™ directory and sub-directories. For example, if ThreatConnect is to use `/opt/threatconnect/exchange` as the TC Exchange directory, create the following subdirectories:

```
None
mkdir -p /threatconnect-data/exchange/jobs
mkdir /threatconnect-data/exchange/programs
```

2. Ensure the TC Exchange directory and its subdirectories are owned by the user your ThreatConnect container will use (`1000`):

```
None
chown 1000:1000 -R /threatconnect-data/exchange
```

3. Run the following additional command only if you are running rootless Podman:

```
None
docker unshare chown 1000:1000 -R /threatconnect-data/exchange
```

4. Set `TC_EXCHANGE` in your `.env` file as follows:

```
None
TC_EXCHANGE=/threatconnect-data/exchange
```

Step 21: Start ThreatConnect

1. Start each of the following services in the following order: OpenSearch → Postgres® → **tc-mon** → **tc-app** → **tc-job**. After starting each service, make sure to perform the following actions:
   - Run `docker-compose logs --tail=10 --follow` to verify the service starts up before moving on to the next.
   - Press **Ctrl+C** once the service is started.
2. After all services are started successfully, log into ThreatConnect:

```
None
admin/password1
```

If you encounter issues starting ThreatConnect, see the "Troubleshooting Notes" section for more information about known issues that may occur during this step.

Start OpenSearch

Note: You must complete this step on the host that will run OpenSearch.

1. Start OpenSearch:

```
None
docker-compose up -d opensearch
```

2. Set the OpenSearch `admin` password:

```
None
./set_opensearch_password.sh
```

Start Postgres

Note: You must complete this step on the host that will run Postgres.

1. Start Postgres:

```
None
docker-compose up -d postgres
```

2. Extract and load the database schema on the database server:

```
None
./load_schema.sh
```

Start tc-mon

Note: You must complete this step on the host that will run the ThreatConnect messaging server.

Run the following command to start **tc-mon**:

```
None
docker-compose up -d nginx redis tc-mon
```

Start tc-app

Note: You must complete this step on the host that will run the ThreatConnect application server.

Run the following command to start **tc-app**. Note that `nginx` is required only if you are on a host other than **tc-mon**.

```
None
docker-compose up -d nginx tc-app
```

Start tc-job

Note: You must complete this step on the host that will run the ThreatConnect Playbooks server.

Run the following command to start **tc-job**:

```
None
docker-compose up -d nginx tc-job
```

Step 22: Monitor ThreatConnect

Follow these steps to restart and monitor the ThreatConnect containers without an `.env` file in place:

1. Move your `.env` file to a secure location (e.g., a server where passwords are stored).
2. Docker Compose commands cannot be run without an `.env` file in place. Therefore, run the following command to check the status of the ThreatConnect containers. Note that the container names are in the first column.

```
None
docker ps --format "table {{.Names}}\t{{.Image}}\t{{.Status}}"
```

3. Restart the ThreatConnect containers by running the following commands one at a time, waiting for each container to start (about 60–90 seconds) before entering the next command:

```
None
docker restart tc-mon
docker restart tc-app
docker restart tc-job
```

4. If environment variables for TC_APP_LOGS, TC_JOB_LOGS, and TC_MON_LOGS are set, then run the following commands to tail monitor the ThreatConnect logs, respectively:

```
None
tail -f $TC_APP_LOGS/server.log $TC_APP_LOGS/tc.log
tail -f $TC_JOB_LOGS/server.log $TC_JOB_LOGS/tc.log
tail -f $TC_MON_LOGS/server.log $TC_MON_LOGS/tc.log
```

ThreatConnect logs can also be found in the following locations:

- Docker: `/var/lib/docker/volumes/`
- Podman: `$XDG_DATA_HOME/containers/storage/`

Step 23: Create Search Index

1. Log into ThreatConnect with a System Administrator account.
2. Hover over **Settings** on the top navigation bar and select **System Settings**.
3. Set the values of **searchAdminPassword** and **searchAdminUsername** to OPENSEARCH_PASSWORD and OPENSEARCH_USERNAME, respectively, in the `.env` file.
4. Bounce ThreatConnect by running the following commands one at a time, waiting for each container to restart (about 60–90 seconds) before entering the next command:

```
None
docker restart tc-mon
docker restart tc-app
docker restart tc-job
```

Step 24: Rotate Nginx Container Access Logs

1. Install **logrotate** using `yum`, `dnf`, or `apt-get`.
2. Run the following command:

```
None
touch /etc/logrotate.d/nginx
```

3. Add the following configuration to `/etc/logrotate.d/nginx` (replace the `<nginx_volume>`, `<number of days>`, `<bin_path>`, and `<path_to>` placeholder values):

```
None
/var/lib/docker/volumes/<nginx_volume>/_data/tc-access.log {
    daily
    rotate <number of days>
    dateext
    compress
    missingok
    notifempty
    create 640 root root
    postrotate
        <bin_path>/docker-compose -f <path_to>/docker-compose.yml exec nginx
nginx -s reload
    endscript
}
```

4. Run the following command:

```
None
logrotate -f /etc/logrotate.d/nginx
```

Appendix

Document Storage Network Share

If you intend to run ThreatConnect in a multi-server configuration (i.e., a configuration where applications, messaging, and Playbooks all run on different hosts), you must set up a network shared folder for document storage that can be shared by all three hosts.

This example uses Network File System (NFS) Utils to set up a network shared folder on the host that will run the ThreatConnect messaging server (**tc-mon**). On each host, there must be a user with `UID=1000`. If there is no such user, create one. In the following examples, `threatconnect` is the user with `UID=1000`.

1. Verify which user has `UID=1000`:

```
None
grep 1000 /etc/passwd
```

2. Set the ThreatConnect messaging host (replace `<tc-mon-host>` with the FQDN of the server that will run **tc-mon**):

```
None
yum install nfs-utils
echo "Domain = <tc-mon-host>" >> /etc/idmapd.conf
```

3. Run the following commands (replace the two IP addresses [`10.9.8.186` and `10.9.8.187`] with those of the servers that will run **tc-app** and **tc-job**):

```
None
echo "/threatconnect-data/storage
10.9.8.186(rw,sync,no_root_squash,no_subtree_check)" > /etc/exports
echo "/threatconnect-data/storage
10.9.8.187(rw,sync,no_root_squash,no_subtree_check)" > /etc/exports
```

4. Start the NFS and add a firewall rule:

```
None
systemctl start nfs-server
systemctl enable nfs-server
systemctl status nfs-server
firewall-cmd --add-service={nfs,nfs3,mountd,rpc-bind} --permanent
firewall-cmd --reload
```

5. Verify the NFS:

```
None
exportfs -v
```

6. Run the following commands on the ThreatConnect application host (replace the IP address [`10.9.8.185`] with that of the server that will run **tc-mon**):

```
None
yum install nfs-utils
showmount -e 10.9.8.185
mkdir -p /threatconnect-data/storage
mount 10.9.8.185:/threatconnect-data/storage /threatconnect-data/storage
ls -l /threatconnect-data/storage
```

7. Run the following commands on the ThreatConnect Playbooks host (replace the IP address [`10.9.8.185`] with that of the server that will run **tc-mon**):

```
None
yum install nfs-utils
showmount -e 10.9.8.185
mkdir -p /threatconnect-data/storage
mount 10.9.8.185:/threatconnect-data/storage /threatconnect-data/storage
ls -l /threatconnect-data/storage
```

Troubleshooting Notes

If you forget to put a value into the `.env` file or put a wrong value in the `.env` file and then started the containers, use one the following methods as a fix:

Method 1: (Recommended) Pause the installation process and remove one or more containers by running the following command:

```
None
docker rm -f tc-mon tc-app tc-job
```

The next time you start the containers, the corrected environment variable will get processed.

Method 2: Pause the installation process and remove all containers by running the following command:

```
None
docker-compose down
```

If you receive the following error the first time you execute `docker-compose up -d`, you must add more IP address space to Docker:

```
None
could not find an available, non-overlapping IPv4 address pool among the
defaults to assign to the network
```

To add more IP address space to Docker, add an IP address block that applies to your environment to `/etc/docker/daemon.json`:

```
None
{
    ...
    "default-address-pools":[
        {"base":"172.20.0.0/16","size":24},
        {"base":"172.21.0.0/16","size":24}
    ]
```

```
}
```

If you experience difficulties connecting to OpenSearch, try connecting to it with curl as follows:

```
None
source .env
curl -k -s -u $OPENSEARCH_USERNAME:$OPENSEARCH_PASSWORD https://localhost:9200/
```

If your attempts to connect to OpenSearch return an error saying that access is unauthorized, try running the following command to reset the password:

Note: You may need to update the value of the **searchAdminPassword** system setting on the **System Settings** screen once ThreatConnect is running. Note that you will need to restart ThreatConnect after changing the **searchAdminPassword** system setting in order for the change to take effect.

```
None
./set_opensearch_password.sh
```

Enabling SAML

Follow these steps to enable the Security Assertion Markup Language™ (SAML) configuration on ThreatConnect:

1. In the `.env` file associated with the containerized deployment of ThreatConnect, update each variable in the "SAML Settings" section with the appropriate value. For descriptions of the values that you must provide in the `.env` file, reference the comments in the "SAML Settings" section of that file.

2. Add the following `.pem` files to the `certs` folder:

    - `<path>/threatconnect-docker/certs/saml_privkey.pem`
    - `<path>/threatconnect-docker/certs/saml_fullchain.pem`
    - `<path>/threatconnect-docker/certs/saml_host.pem`

Note: The `saml_fullchain.pem` and `saml_privkey.pem` files can have the same content as the `fullchain.pem` and `privkey.pem` files. The `saml_host.pem` file must contain the Identity Provider (IDP) certificate.