ThreatConnect® Migration Guide: Containerized Deployment

Software Version 7.9

Technical Guide

April 23, 2025

10034-08 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 is intended for customers who want to upgrade ThreatConnect® and migrate it to a containerized deployment and, at the same time, switch from CentOS™ 7 to AlmaLinux OS™ 9 due to CentOS 7 reaching end of life (EOL). As of ThreatConnect 7.5, you will no longer be required to install Java®, Python®, OpenSearch®, and Redis® during the ThreatConnect upgrade process. Instead, all of this software, along with ThreatConnect, is now packaged together in a containerized solution using Docker® or Podman. The only thing that is not included in the containerized environment is the database, which will not be touched during the upgrade process.

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

Prerequisites

Credentials

- Amazon Web Services® (AWS) Access Key ID
- AWS Secret Access Key
- ThreatConnect `keystore.jks` password
- Old Postgres® database `tcuser` password
- OpenSearch `admin` password

Upgrade and Migration Steps

Step 1: Migrate Data

Important: Steps 1—5 apply to all Docker and Podman deployments.

1. Shut down ThreatConnect and its supporting services:

```
Unset
systemctl stop threatconnect redis opensearch
```

2. Create a folder on the new host that will serve as the repository for data copied over from the old ThreatConnect system.

Note: The name and location of this folder are arbitrary and can be anything you want. This guide uses `/threatconnect-data` as the folder name.

```
Unset
mkdir /threatconnect-data
```

3. Place the following items into the `/threatconnect-data` folder:
    - OpenSearch Data
        - The location of OpenSearch data can be found in `//<old OpenSearch host>/etc/opensearch/opensearch.yml` under property `path.data`.
        - Example command to archive data: `cd /opt && tar -czf /tmp/opensearch-data.tar.gz opensearch-data`
        - Copy `opensearch-data.tar.gz` to the host that will run OpenSearch.
        - Untar `opensearch-data.tar.gz` to `/threatconnect-data`.
        - Copy `//<old OpenSearch host>/etc/opensearch/opensearch-security/internal_users.yml` to the host that will run OpenSearch.
    - Postgres dump file
        - See the "Export Postgres Dump File" section for instructions on exporting a Postgres dump file.

    - The Postgres dump file needs to be copied only to the host intended to run Postgres.
- ThreatConnect Certificates
    - See the "Export Certificates" section for instructions on exporting certificates.
- TC Exchange™
    - The location of TC Exchange data can be found in `//<old ThreatConnect host>/path/to/threatconnect/exchange`.
    - Example command to archive data: `cd /opt/threatconnect && tar -czf /tmp/exchange.tar.gz exchange`
    - If migrating from a multi-server environment, TC Exchange data need to be copied only from and to the appropriate environment. For example, TC Exchange data from the messaging server need to be copied to the host that will be the new messaging server (**tc-mon**).
    - Copy `exchange.tar.gz` to the host(s) that will run ThreatConnect.
    - Untar `exchange.tar.gz` to `/threatconnect-data`.
- ThreatConnect Storage
    - The location of ThreatConnect storage data can be found in `//<old ThreatConnect host>/path/to/threatconnect/storage`.
    - Example command to archive data: `cd /opt/threatconnect && tar -czf /tmp/storage.tar.gz storage`
    - If you intend to run messaging (**tc-mon**), application (**tc-app**), and Playbooks (**tc-job**) containers on different hosts, a networked document storage folder that will be shared by all three hosts is required. See the "Document Storage Network Share" section for instructions on mounting a document storage network share.
    - Copy `storage.tar.gz` to the host that will run ThreatConnect or the document storage network share.
    - Untar `storage.tar.gz` to `/threatconnect-data`.
- ThreatConnect License
    - The license file must be copied to all hosts that will run messaging (**tc-mon**), applications (**tc-app**), and Playbooks (**tc-job**).

Step 2: 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 7.9.0, run the following commands:

```
Unset
cd /opt
unzip threatconnect-docker-v7.9.0.zip
cd /opt/threatconnect-docker
```

Step 3: 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.

```
Unset
cp /opt/threatconnect-docker/.env.sample /opt/threatconnect-docker/.env
```

Step 4: 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 5: Add Certificates

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

1. Create the two required certificate files in the `certs` folder. These are the certificate authority–signed (CA-signed) certificate and private key.

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

2. If using a custom CA, update `CUSTOM_CA_PEM_FILE` in your `.env` file as follows:

```
Unset
CUSTOM_CA_PEM_FILE=fullchain.pem
```

Step 6: Install Docker

Important: Steps 6 and 7 apply to Docker deployments only. For Podman deployments, skip to Step 8.

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:

```
Unset
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 7: Install Docker Compose

Important: Steps 6 and 7 apply to Docker deployments only. For Podman deployments, skip to Step 8.

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:

```
Unset
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 8: Install Podman

Important: Steps 8 and 9 apply to Podman deployments only. For Docker deployments, skip to Step 10.

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:

```
Unset
dnf module install -y podman
export PODMAN_PATH=$(which podman)
ln -s $PODMAN_PATH /usr/bin/docker
```

Step 9: Install Podman Compose

Important: Steps 8 and 9 apply to Podman deployments only. For Docker deployments, skip to Step 10.

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:

```
Unset
dnf install python3.11 python3.11-pip
pip3.11 install podman-compose
ln -s /usr/local/bin/podman-compose /usr/bin/docker-compose
```

Step 10: Install AWS CLI

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

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

```
Unset
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" \
  -o "awscliv2.zip" &&\
  unzip awscliv2.zip &&\
  ./aws/install
```

Step 11: 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`:

```
Unset
echo 'vm.max_map_count=262144' >> /etc/sysctl.conf
echo 'net.ipv4.ip_unprivileged_port_start=25' >> /etc/sysctl.conf
sysctl -p
```

Step 12: 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:

```
Unset
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 13: Create Users

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

```
Unset
adduser --uid 1000 threatconnect
adduser --uid 1001 tc-job
```

Step 14: Install Random-Number Generation Tools

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

```
Unset
dnf install -y rng-tools
systemctl enable rngd
systemctl start rngd
```

Step 15: Configure Rootless Podman

Important: Steps 15 and 16 apply to rootless Podman deployments only. For Docker deployments and deployments running Podman as root, skip to Step 17.

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**:

```
Unset
loginctl enable-linger threatconnect
```

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

```
Unset
chown threatconnect:threatconnect -R /opt/threatconnect-docker
```

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

```
Unset
su - threatconnect
```

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

Step 16: Configure Podman Home Container

Important: Steps 15 and 16 apply to rootless Podman deployments only. For Docker deployments and deployments running Podman as root, skip to Step 17.

Run the following commands to configure the Podman home container:

```
Unset
cd /opt/threatconnect-docker
mkdir run
echo "export XDG_DATA_HOME=/opt/threatconnect-docker/run" >>
/home/threatconnect/.bashrc
source /home/threatconnect/.bashrc
```

Step 17: 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`.

```
Unset
/usr/local/bin/aws configure
Access Key ID:****
Secret Access Key:****
Region:us-east-1
```

```
Unset
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 18: 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 logs and snapshots mount folders:

```
Unset
mkdir /threatconnect-data/opensearch-logs
mkdir /threatconnect-data/opensearch-snapshots
```

2. Copy or mount the OpenSearch data directory. Ensure `OPENSEARCH_DATA` has an absolute file path in your `.env` file and is owned by `1000:1000`, as in the following example:

```
Unset
chown 1000:1000 -R /threatconnect-data/opensearch-data
```

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

```
Unset
docker unshare chown 1000:1000 -R /threatconnect-data/opensearch-*
```

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

```
Unset
      OPENSEARCH_DATA=/threatconnect-data/opensearch-data
      OPENSEARCH_LOGS=/threatconnect-data/opensearch-logs
      OPENSEARCH_SNAPSHOTS=/threatconnect-data/opensearch-snapshots
```

5. Copy `/threatconnect-data/opensearch_internal_users.yml` to `/opt/threatconnect-docker/config/opensearch_internal_users.yml`.

6. Set OpenSearch credentials in your `.env` file to the same values as on the system you are migrating from:

```
Unset
OPENSEARCH_USERNAME=
OPENSEARCH_PASSWORD=
```

Step 19: 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**:

```
Unset
mkdir -p /threatconnect-data/logs/tc-mon
mkdir -p /threatconnect-data/logs/tc-app
mkdir -p /threatconnect-data/logs/tc-job
chown 1000:1000 -R /threatconnect-data/logs/
```

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

```
Unset
docker unshare chown 1000:1000 -R /threatconnect-data/logs
```

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

```
Unset
TC_MON_LOGS=/threatconnect-data/logs/tc-mon
TC_APP_LOGS=/threatconnect-data/logs/tc-app
TC_JOB_LOGS=/threatconnect-data/logs/tc-job
```

Step 20: 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. Ensure `TC_DOC_STORAGE` in your `.env` file has an absolute file path and is owned by `1000:1000`. For example, if your document storage folder is `/threatconnect-data/storage`, update `TC_DOC_STORAGE` in your `.env` file as follows:

```
Unset
TC_DOC_STORAGE=/threatconnect-data/storage
```

2. Then run the following command:

```
Unset
chown 1000:1000 -R /threatconnect-data/storage
```

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

```
Unset
docker unshare chown 1000:1000 -R /threatconnect-data/storage
```

Step 21: 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. If your TC Exchange folder is `/threatconnect-data/exchange`, update `TC_EXCHANGE` in your `.env` file as follows:

```
Unset
TC_EXCHANGE=/threatconnect-data/exchange
```

2. Run the following command:

```
Unset
chown 1000:1000 -R /threatconnect-data/exchange
```

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

```
Unset
docker unshare chown 1000:1000 -R /threatconnect-data/exchange
```

Step 22: 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.

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:

```
Unset
docker-compose up -d opensearch
```

2. Set the OpenSearch password only if you are running rootless Podman:

    Note: In a rootless Podman environment, the security admin script that processes `opensearch_internal_users.yml` does not get executed automatically.

```
Unset
./set_opensearch_password.sh
```

Start Postgres

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

1. Start Postgres:

```
Unset
docker-compose up -d postgres
```

2. Create the less-privileged user with which to run ThreatConnect:

```
Unset
./create_db_user.sh
```

3. Load your dump file:

```
Unset
cp /threatconnect-data/dump.sql /opt/threatconnect-docker/schema/
source .env
docker-compose exec postgres psql -U $DB_SUPER_USER -d threatconnect -f
/schema/dump.sql
```

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**:

```
Unset
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**.

```
Unset
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**:

```
Unset
docker-compose up -d nginx tc-job
```

Step 23: Fix Services

Open `//<old ThreatConnect host>/path/to/threatconnect/config/install.properties` and verify whether `serverType` is set to `FULL`.

serverType Is Not FULL

If `serverType` is set to anything other than `FULL`, this means you are migrating from a multi-server environment. To fix your Services directly in the database, contact your ThreatConnect Customer Support Engineer. To fix your Services manually in the ThreatConnect user interface, follow these steps:

1. Log into ThreatConnect with a System Administrator account.
2. Hover over **Playbooks** on the top navigation bar and select **Services**.
3. On the **Services** screen, click the ⋮ menu for a Service and select **Edit**.
4. On the **Configure** step of the **Edit Service** drawer, select **tc-job** in the **Launch Server** dropdown. Then click **NEXT**, followed by **SAVE**, to save your changes to the Service.
5. Repeat Steps 3–4 for each Service on your ThreatConnect instance.
6. Restart ThreatConnect.

serverType Is FULL

If `serverType` is set to `FULL`, follow these steps to correct Services directly in the database:

1. Run the following SQL to update the server name for all Services to be **tc-job**:

```
Unset
UPDATE AppCatalogItem SET id_ServiceServer =
  (SELECT id FROM ServiceServer WHERE UPPER(name) = 'TC-JOB' order by
datecreated desc limit 1)
WHERE id_ServiceServer IN (SELECT id FROM ServiceServer WHERE UPPER(name) =
  (select UPPER(name) from serviceserver where id in (select id_serviceserver
from appcatalogitem)))
AND EXISTS (SELECT id FROM ServiceServer WHERE UPPER(name) = 'TC-JOB');
```

2. Run the following SQL to delete the server from the old ThreatConnect deployment that no longer exists:

```
Unset
delete from serviceserver where id=1;
```

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 (switch to the appropriate host before running the commands):

```
Unset
docker restart tc-mon
docker restart tc-app
docker restart tc-job
```

4. Watch for output similar to the following to know when the Services start up:

```
Unset
docker-compose logs --tail=10 --follow tc-mon tc-app tc-job
tc-job-1  | 2024-08-13 16:57:24,317 INFO
[com.threatconnect.common.execution.service.manager.AbstractAppServicesManager]
(pool-30-thread-1) Launch App Requested: JS Report
```

```
tc-job-1  | 2024-08-13 16:57:24,320 INFO
[com.threatconnect.common.execution.service.manager.AbstractAppServicesManager]
(pool-30-thread-1) Handling launch on app: JS Report
tc-job-1  | 2024-08-13 16:57:30,739 INFO
[com.threatconnect.common.execution.service.manager.ServiceManager]
(pool-30-thread-1) Starting service manager for app:
8f08f5c6450ad5b0789b1f2e7b903302
```

Step 24: 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.

```
Unset
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:

```
Unset
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:

```
Unset
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 25: Rotate Nginx Container Access Logs

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

```
Unset
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):

```
Unset
/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:

```
Unset
logrotate -f /etc/logrotate.d/nginx
```

Appendix

Export Certificates

If needed, run the following commands to export the `tc cert` and `key` from `keystore.jks` on the old ThreatConnect host as `fullchain.pem` and `privkey.pem`, respectively (replace all `<password>` placeholder values):

```
Unset
keytool -importkeystore \
    -srckeystore /opt/threatconnect/config/keystore.jks \
    -srcstorepass <password> \
    -destkeystore keystore.p12 \
    -deststoretype PKCS12 \
    -srcalias tc \
    -deststorepass <password> -destkeypass <password>
openssl pkcs12 -in keystore.p12 -nokeys -out fullchain.pem -password
pass:<password>
openssl pkcs12 -in keystore.p12 -nodes -nocerts -out privkey.pem \
       -password pass:<password>
```

Export Postgres Dump File

Run the following command to generate a Postgres database dump file (replace the `<username>`, `<hostname>`, `<port>`, and `<dbname>` placeholder values):

```
Unset
pg_dump -U <username> -h <hostname> -p <port> <dbname> > /tmp/dump.sql
```

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`:

```
Unset
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**):

```
Unset
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**):

```
Unset
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:

```
Unset
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:

```
Unset
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**):

```
Unset
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**):

```
Unset
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:

```
Unset
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:

```
Unset
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:

```
Unset
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`:

```
Unset
{
    ...
    "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:

```
Unset
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.

```
Unset
./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.