ThreatConnect® Migration Guide: Containerized Deployment

Software Version 7.7

Technical Guide

September 24, 2024

10034-04 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®. The only thing that is not included in the Docker environment is the database, which will not be touched during the upgrade process.

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

1. Shut down ThreatConnect and its supporting services:

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

2. Create a folder on the new host with a name such as `/threatconnect-data.` This folder will be a repository for data copied over from the old ThreatConnect system in the next step.

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

3. Place the following items into the `/threatconnect-data` folder:
    - OpenSearch Data
        - Located at `//<old OpenSearch host>/etc/opensearch/opensearch-security/internal_users.yml`
        - OpenSearch data need to be copied only to the host intended to run OpenSearch.
    - OpenSearch `internal_users.yml`
        - Located at `//<old OpenSearch host>/etc/opensearch/opensearch-security/internal_users.yml`
        - OpenSearch `internal_users.yml` needs to be copied only to the host intended to 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™
    - Located at `//<old ThreatConnect host>/path/to/threatconnect/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**).
- ThreatConnect Storage
    - Located at `//<old ThreatConnect host>/path/to/threatconnect/storage`
    - If you intend to run messaging (**tc-mon**), application (**tc-app**), and Playbooks (**tc-job**) Dockers 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.
- 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

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 7.6.3, run the following commands:

```
Unset
cd /opt
unzip ThreatConnect-Docker-v7.6.3.zip
cd /opt/threatconnect-docker
```

Step 3: Update Docker 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 `config/license.xml`:

```
Unset
cp /threatconnect-data/license.xml /opt/threatconnect-docker/config/
```

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

```
Unset
mkdir -p /opt/threatconnect-docker/certs/trusted
cp /threatconnect-data/fullchain.pem /opt/threatconnect-docker/certs/
cp /threatconnect-data/privkey.pem /opt/threatconnect-docker/certs/
```

2. If applicable, add trusted certificates to the `certs` folder (e.g., a SAML™ IDP certificate):

```
Unset
cp /threatconnect-data/trusted/* /opt/threatconnect-docker/certs/trusted/
```

3. 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

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

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 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). However, if you are on a system that is not connected to the internet (i.e., an air-gapped system), you do not need to install AWS CLI; instead, see the "Air Gap System" section for further instruction.

1. Install AWS CLI:

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

2. Configure AWS CLI using the credentials your ThreatConnect Customer Success Manager shared with you. If located in Europe, replace `us-east-1` with `eu-central-1` before running the following commands:

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

3. Log into ThreatConnect's ECR. If located in Europe, replace `us-east-1` with `eu-central-1` before running the following commands:

```
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 9: 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 'vm.max_map_count=262144' >> /etc/sysctl.d/99-sysctl.conf
sysctl -p
```

Step 10: 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$//' docker-entrypoint.d/00_init.sh
chmod 755 docker-entrypoint.d/00_init.sh
sed -i 's/\r$//' 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
```

Step 11: Update Firewall Ports

Note: You must complete this step on the hosts that previously ran ThreatConnect.

Remove forward ports from the previous ThreatConnect installation:

```
Unset
firewall-cmd --permanent
--remove-forward-port=port=25:proto=tcp:toport=2500:toaddr=
firewall-cmd --permanent
--remove-forward-port=port=80:proto=tcp:toport=8080:toaddr=
firewall-cmd --permanent
--remove-forward-port=port=443:proto=tcp:toport=8443:toaddr=
firewall-cmd --reload
```

To verify whether the forward ports were removed successfully, run the following command:

```
Unset
firewall-cmd --list-all
```

If the forward ports were removed, you will see output similar to the following:

```
Unset
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens192
  sources:
  services: cockpit dhcpv6-client smtp ssh
  ports: 5432/tcp 6379/tcp 9200/tcp 9600/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Step 12: Configure OpenSearch Data

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

1. 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
```

2. Set `OPENSEARCH_DATA` in your `.env` file as follows:

```
Unset
OPENSEARCH_DATA=/threatconnect-data/opensearch-data
```

3. Secure OpenSearch.
4. Replace `config/opensearch_internal_users.yml` with your own internal users file, as in the following example:

```
Unset
cp /threatconnect-data/internal_users.yml \
       /opt/threatconnect-docker/config/opensearch_internal_users.yml
```

5. Retrieve the encrypted username and password values from the database and update `OPENSEARCH_USERNAME` and `OPENSEARCH_PASSWORD` in your `.env` file as follows:

```
Unset
select name,value from systemconfig
       where name='searchAdminUsername' or name='searchAdminPassword';
```

6. Set `OPENSEARCH_SECURITY_DISABLED` and `OPENSEARCH_SECURITY_ENABLED` in your `.env` file as follows:

```
Unset
OPENSEARCH_SECURITY_DISABLED=false
OPENSEARCH_SECURITY_ENABLED=true
```

Step 13: 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**).

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
```

Then run the following command:

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

Step 14: Configure TC Exchange Data

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

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

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

Then run the following commands:

```
Unset
# Update permissions as follows:
chown 1000:1000 -R /threatconnect-data/exchange

# correct octal permissions
find /threatconnect-data/exchange -type f -exec chmod 644 -- {} +
find /threatconnect-data/exchange -type d -exec chmod 755 -- {} +

# set ACLs
setfacl -Rm u:1001:rx /threatconnect-data/exchange/programs/
setfacl -Rm u:1001:rwx /threatconnect-data/exchange/jobs/
setfacl -Rm u:1000:rwx /threatconnect-data/exchange/jobs/

# set new default ACLs
setfacl -Rdm u:1001:rx /threatconnect-data/exchange/programs/
setfacl -Rdm u:1001:rwx /threatconnect-data/exchange/jobs/
setfacl -Rdm u:1000:rwx /threatconnect-data/exchange/jobs/
```

Step 15: 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:
    - Execute `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. Test the installation (replace the `<opensearch password>` placeholder value):

```
Unset
curl -sku admin:<opensearch password>
https://localhost:9200/_cat/indices/orgs?v
```

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. Once Postgres is started, load your dump file:

```
Unset
cp /threatconnect-data/dump.sql /opt/threatconnect-docker/schema/
docker-compose exec postgres psql -U tcuser -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 tc-job
```

Step 16: 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 containers (switch to the appropriate host before running the following commands):

```
Unset
docker-compose stop tc-mon tc-app tc-job
docker-compose start tc-mon
#wait
docker-compose start tc-app
#wait
docker-compose start 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 17: 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 (replace all `<container name>` placeholder values):

```
Unset
docker restart <container name> <container name>
```

4. Tail monitor the ThreatConnect logs:

```
Unset
docker ps --format "table {{.Names}}" | grep -e "mon\|app\|job" | xargs -L 1 -P
`docker ps | wc -l` docker logs --since 15m -f
```

Appendix

Air Gap System

If you are on a system that is not connected to the internet (i.e., an air-gapped system), follow these steps to download and install the necessary Docker images:

1. Pull Docker images and save them for use on an air-gapped system. (Replace the `<version number>` placeholder values with the version number for the ThreatConnect version you are installing.) If located in Europe, replace `us-east-1` with `eu-central-1` before running the following commands:

```
Unset
docker save -o threatconnect-v<version number>.tar
373319941383.dkr.ecr.us-east-1.amazonaws.com/threatconnect:v<version number>
docker save -o opensearch-v2.6.0.tar
373319941383.dkr.ecr.us-east-1.amazonaws.com/opensearch:2.6.0
docker save -o redis-v6.2.6.tar redis:6.2.6
docker save -o postgres-v14.9.tar postgres:14.9
docker save -o nginx-v1.23.3.tar nginx:1.23.3
```

2. Copy the TAR file images to the air-gapped system.

3. Load the TAR file images into Docker (replace the `<version number>` placeholder value with the version number for the ThreatConnect version you are installing):

```
Unset
docker load -i threatconnect-v<version number>.tar
docker load -i opensearch-v2.6.0.tar
docker load -i redis-v6.2.6.tar
docker load -i postgres-v14.9.tar
docker load -i nginx-v1.23.3.tar
```

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 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}
    ]
}
```