ThreatConnect® Upgrade Guide: Containerized Deployment

Software Version 7.9

Technical Guide

April 23, 2025

10033-11 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 upgrade ThreatConnect® on a ThreatConnect instance that is running in a containerized solution using Docker® or Podman.

Important: The containerized deployment was tested on AlmaLinux OS™ and is the preferred deployment method for all production and non-production systems starting with ThreatConnect version 7.5. For instructions on how to upgrade ThreatConnect and keep it running directly on an operating system (OS), see *ThreatConnect Upgrade Guide: Operating System Deployment*.

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

Upgrade Steps

Important: All of the steps in this guide apply to all Docker and Podman deployments.

Step 1: Upgrade Docker Files

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

Important If you have changed any of the following files locally, you must back them up, because they will get overwritten during the upgrade:

- `nginx.conf`
- `opensearch_internal_users.yml`
- `postgres.conf`
- `redis.conf`

1. Upgrade the ThreatConnect Docker ZIP file to the latest version (replace the `<version number>` placeholder value with the version number for the ThreatConnect version to which you are upgrading):

```
Unset
cd /opt
unzip -o /tmp/threatconnect-docker-v<version number>.zip
```

2. Redo any local changes that you had in the following files, taking care not to overwrite incoming updates from ThreatConnect:
    - `nginx.conf`
    - `opensearch_internal_users.yml`
    - `postgres.conf`
    - `redis.conf`

Step 2: Fix Shell Scripts

Note: You must complete this step on all hosts that 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 3: Restore Your Environment File

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

Restore your ThreatConnect Docker `.env` file (replace the `<//secure location>` placeholder value):

```
Unset
scp <//secure location>/.env /opt/threatconnect-docker/.env
```

Step 4: Update the TC_VERSION Variable

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

Update `TC_VERSION` to the latest ThreatConnect version in your `.env` file (replace the `<version number>` placeholder value with the version number for the ThreatConnect version to which you are upgrading):

```
Unset
TC_VERSION=v<version number>
```

Step 5: Add New Environment Variables

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

1. Add the following environment variables with appropriate values to the `.env` file:

```
Unset
# OpenSearch Credentials [Required].
OPENSEARCH_USERNAME=
OPENSEARCH_PASSWORD=
# Opensearch Logs. An absolute path to Opensearch Logs, owned by user 1000
```

```
OPENSEARCH_LOGS=
# OpenSearch bootstrap.memory_lock. Set to false in rootless mode. Default:
true
OPENSEARCH_MEM_LOCK=
# OpenSearch Snapshot location
OPENSEARCH_SNAPSHOTS=
# NGINX DNS Resolver [Required]
#     Docker: 127.0.0.11 (default)
#     Podman: 10.89.0.1

NGINX_DNS_RESOLVER=127.0.0.11
```

2. Remove the following environment variable from the `.env` file:

```
Unset
REDIS_ARGS=
```

Step 6: Log Into ThreatConnect's ECR

Log into ThreatConnect's Elastic® Container Registry (ECR):

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
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 7: Stop and Remove Containers

Run the following command to stop and remove the service containers:

```
Unset
docker-compose down
docker-compose pull
```

Step 8: Create Newly Supported OpenSearch Folder Mounts

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

1. Create the OpenSearch mount folders:

```
Unset
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:

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

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

```
Unset
OPENSEARCH_LOGS=/opt/threatconnect-docker/opensearch-logs
OPENSEARCH_SNAPSHOTS=/opt/threatconnect-docker/opensearch-snapshots
```

Step 9: Start OpenSearch

Note: You must complete this step on the host that runs 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
```

Important: If you receive an "Unauthorized" error while testing the connection to OpenSearch, run the following script to reset the OpenSearch password:

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

If you reset the OpenSearch password, you must 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.

Step 10: Start ThreatConnect

Start each of the following services in the following order: **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.

Start tc-mon

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

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

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

Step 11: 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 or Podman 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 12: Re-create the Search Index

Important: If you are **upgrading from ThreatConnect 7.7.0 or later**, skip this step.

Follow these steps to re-create the 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. Click **CREATE SEARCH INDEX** at the top right of the **Settings** tab.

4. On the **Setup** tab of the **Search Index Configuration** window, select the **Perform search indexing on database source** and **Load file contents into search index** checkboxes, and then click **INITIALIZE**.

   Note: The **Perform search indexing on database source** checkbox determines whether to index objects that exist in the ThreatConnect database, and the **Load file contents into search index** checkbox determines whether to index objects that exist in document storage.

Step 13: Update the Search Index

Important: If you are **upgrading from ThreatConnect 7.7.0 or later**, skip this step.

Update the search index on the host that is running the OpenSearch Docker container (replace the `<opensearch username>` and `<opensearch password>` placeholder values):

```
Unset
export OPENSEARCH_USER=<opensearch username>
export OPENSEARCH_PASS=<opensearch password>
curl -sk -XPOST
"https://$OPENSEARCH_USER:$OPENSEARCH_PASS@localhost:9200/orgs/_close"
curl -sk -XPUT
"https://$OPENSEARCH_USER:$OPENSEARCH_PASS@localhost:9200/orgs/_settings" -H
'Content-Type: application/json' -d'
{
  "analysis.analyzer": {
    "default_search": {
      "filter": [
        "lowercase",
        "asciifolding"
      ],
      "type": "custom",
      "tokenizer": "whitespace"
    }
  }
}'
curl -sk -XPOST
"https://$OPENSEARCH_USER:$OPENSEARCH_PASS@localhost:9200/orgs/_open"
```

Step 14: 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

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.