ThreatConnect® Migration Guide: Containerized Deployment

Software Version 7.5

Technical Guide

May 7, 2024

10034-01 EN Rev. A

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

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

Table of Contents

Overview **4**

Upgrading ThreatConnect and Migrating to a Containerized Deployment **4**

Step 1: Shut Down ThreatConnect and Supporting Services 4

Step 2: Download the threatconnect-docker ZIP File 5

Step 3: Configure Environment Variables 5

Step 4: Install the ThreatConnect License 5

Step 5: Create Certificates in the certs Folder 5

Step 6: Export fullchain.pem and privkey.pem From Your Keystore 6

Step 7: Export Additional Trusted Certificates From Your Keystore 7

Step 8: Install AWS CLI and Log Into ThreatConnect's ECR 7

Step 9: Install Docker 8

Step 10: Install Docker Compose 8

Step 11: Increase vm.max_map_count 9

Step 12: Remove Forwarded Ports From the Previous ThreatConnect Installation 9

Step 13: Reformat and Change Permissions on Shell Scripts 9

Step 14: Copy or Mount OpenSearch Data 10

Step 15: Secure OpenSearch 10

Step 16: Copy or Mount Document Storage 10

Step 17: Start and Log Into ThreatConnect 11

Step 18: Move the .env File to a Secure Location 11

Step 19: Restart and Monitor the ThreatConnect Containers 11

Important: This guide describes how to upgrade ThreatConnect and migrate it to run in a containerized solution using Docker® instead of directly on the operating system (OS). To upgrade ThreatConnect and keep it running directly on an OS, see *ThreatConnect Upgrade Guide: Operating System Deployment*. Note that the containerization 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.

Overview

This guide describes how to upgrade ThreatConnect and migrate it to a containerized deployment. As of ThreatConnect version 7.5, you will no longer be required to install Java®, Python®, OpenSearch®, and Redis® during the ThreatConnect upgrade process. Instead, these software, along with ThreatConnect, are 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.

Upgrading ThreatConnect and Migrating to a Containerized Deployment

Follow all steps outlined in the following subsections to upgrade ThreatConnect and migrate it to a containerized deployment.

Step 1: Shut Down ThreatConnect and Supporting Services

Run the following commands to stop ThreatConnect and its supporting services from running:

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

Step 2: Download the threatconnect-docker ZIP File

Run the following commands to download and unzip the `threatconnect-docker` ZIP file for the current version of ThreatConnect. Before running the following commands, replace the placeholder `<version>` value with the corresponding ThreatConnect version number (e.g., `7.5.1`):

```
cd /opt
unzip threatconnect-docker-v<version>.zip
cd /opt/threatconnect-docker
```

Step 3: Configure Environment Variables

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

Note: The values of the variables in the `.env` file were all entered during the initial installation using the installer script (`setup.sh`).

```
cp .env.sample .env
```

Step 4: Install the ThreatConnect License

Place the ThreatConnect license file (`license.xml`) in the `config/` folder.

Step 5: Create Certificates in the certs Folder

Add the following required certificates to the `certs` folder:

- `fullchain.pem`
- `privatekey.pem`

Step 6: Export fullchain.pem and privkey.pem From Your Keystore

Run the following commands to export `fullchain.pem` and `privkey.pem` from `keystore.jks` to the `certs` folder. Before running the commands, replace all placeholder `<password>` values with the appropriate password for the corresponding variable.

Note: The following command assumes `keystore.jks` is located in `/opt/threatconnect/config/keystore.jks`. However, this location may be different from where `keystore.jks` is located in your previous ThreatConnect installation.

```
mkdir -p certs/trusted
cd certs
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>
```

Step 7: Export Additional Trusted Certificates From Your Keystore

If you have additional trusted certificates in your keystore, run the following commands to export them from `keystore.jks` to the `certs` folder. Before running the commands, replace both `<alias>` placeholder values with the alias that the certificate is listed under in the keystore, and replace the `<password>` placeholder value with the password for the keystore.

```
cd trusted
keytool -exportcert \
 -rfc \
 -alias <alias> \
 -file <alias>.pem \
 -keystore /opt/threatconnect/config/keystore.jks \
 -storepass <password> \
 -v
```

Step 8: Install AWS CLI and Log Into ThreatConnect's ECR

1. Run the following commands to install the Amazon Web Services® Command Line Interface (AWS CLI):

   ```
   curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
   "awscliv2.zip" &&\
   unzip awscliv2.zip &&\
   ./aws/install
   /usr/local/bin/aws configure
   Access Key ID:****
   Secret Access Key:****
   Region:us-east-1
   ```

2. Run the following commands to configure the AWS CLI and enter the credentials provided to you by your ThreatConnect Customer Success Manager:

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

3. Run the following command to log into ThreatConnect's Elastic® Container Registry (ECR). Note that this step requires the AWS CLI.

```
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: Install Docker

1. Run **one of the following commands** to install Docker:

```
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
yum install docker-ce docker-ce-cli containerd.io
```

```
dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
dnf install docker-ce docker-ce-cli containerd.io
```

2. Run the following commands to start and enable Docker:

```
systemctl start docker.service
systemctl enable docker.service
docker version
```

Step 10: Install Docker Compose

Run the following commands to install Docker Compose:

```
curl -SL https://github.com/docker/compose/releases/download/v2.24.5/docker-compose-linux-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 11: Increase vm.max_map_count

Run **one of the following commands** to increase the value of `vm.max_map_count` to `262144`:

```
echo 'vm.max_map_count=262144' >> /etc/sysctl.conf
```

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

Step 12: Remove Forwarded Ports From the Previous ThreatConnect Installation

Run the following commands to remove forwarded ports from the previous ThreatConnect installation:

```
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
```

Step 13: Reformat and Change Permissions on Shell Scripts

Run the following commands to reformat and change permissions on shell scripts:

```
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 14: Copy or Mount OpenSearch Data

Copy or mount the OpenSearch data directory to the environment in which you are installing the ThreatConnect Docker containers. Make sure the `OPENSEARCH_DATA` variable in your `.env` file has an absolute file path and is owned by `1000:1000`, as in the following example:

```
chown 1000:1000 -R /mnt/opensearch-data
```

Step 15: Secure OpenSearch

1. Replace `config/opensearch_internal_users.yml` with your initial users file, as in the following example:

   ```
   cp /etc/opensearch/opensearch-security/internal_users.yml
   config/opensearch_internal_users.yml
   ```

2. Retrieve the encrypted username and password values from the database and update the `OPENSEARCH_USERNAME` and `OPENSEARCH_PASSWORD` variables in your `.env` file:

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

3. Set the `OPENSEARCH_PROTOCOL`, `OPENSEARCH_SECURITY_DISABLED`, and `OPENSEARCH_SECURITY_ENABLED` variables in your `.env` file as follows:

   ```
   OPENSEARCH_PROTOCOL=https://
   OPENSEARCH_SECURITY_DISABLED=false
   OPENSEARCH_SECURITY_ENABLED=true
   ```

Step 16: Copy or Mount Document Storage

If your ThreatConnect document storage is local, copy or mount the document storage folder to the environment in which you are installing the ThreatConnect Docker containers. Make sure the `TC_DOC_STORAGE` variable in your `.env` file has an absolute file path and is owned by `1000:1000`, as in the following example:

```
chown 1000:1000 -R /mnt/tc-doc-storage
```

Step 17: Start and Log Into ThreatConnect

Start each of the following services in the order outlined in the accompanying steps. After starting each service, run `docker-compose logs --tail=10 --follow` to verify that the service started successfully before moving on to the next. Once a service is started, press **Ctrl-C**.

1. Start OpenSearch:

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

2. Extract and load the database schema (this must be done on the database server):

```
./load_schema.sh
```

3. Start **tc-mon**:

```
docker-compose up -d nginx redis tc-mon
```

4. Start **tc-app**:

```
docker-compose up -d nginx tc-app
```

5. Start **tc-job**:

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

6. Log into ThreatConnect.

Step 18: Move the .env File to a Secure Location

Move the `.env` file to a secure location (e.g., a server where passwords are stored).

Step 19: Restart and Monitor the ThreatConnect Containers

Docker Compose commands cannot be run without the `.env` file in place. To restart the ThreatConnect containers, use the `docker` command.

1. Run the following command to see the status of the ThreatConnect containers. Note that the container names are listed in the first column.

```
docker ps --format "table {{.Names}}\t{{.Image}}\t{{.Status}}"
```

2. Run the following command to restart the ThreatConnect containers. Before running the command, replace the placeholder `<container name>` values with the names of the ThreatConnect containers.

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

3. Run the following command to tail the ThreatConnect logs:

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