ThreatConnect® Installation Guide: Containerized Deployment

Software Version 7.7

Technical Guide

September 27, 2024

10032-04 EN Rev. B

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®.

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: The `.env` file holds all passwords and configurations for the Docker deployment. Once the Docker 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	16	2	200
Containerized Database	32	4	100

[1]Allocated to ThreatConnect Docker 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.

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.

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

Step 1: 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 2: 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 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**).

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

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

```
Unset
/opt/threatconnect-docker/certs/trusted/
```

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):

```
Unset
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:

```
Unset
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):

```
Unset
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):

```
Unset
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names

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

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

```
Unset
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`:

```
Unset
cat my-root-ca.pem >> fullchain.pem
```

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

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

Step 5: 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 6: 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 7: Install AWS CLI

Amazon Web Services® Command Line Interface (AWS 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 8: 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 9: 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
```

Step 10: 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**).

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 Docker will use (`1000`):

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

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

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

Step 11: 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**).

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:

```
Unset
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 Docker will use (`1000`) and can be written to by the **tc-job** user your ThreatConnect Docker will use (`1001`):

```
Unset
# Create the tc-job user with UID 1001
groupadd -g 1001 tc-job
useradd tc-job -u 1001 -g 1001 -m -s /bin/bash
echo "tc-job-pass123" | passwd tc-job --stdin

# Update permissions as follows:
chgrp -R 1000 /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 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/
```

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

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

Step 12: 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:

```
Unset
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. Run the following command to 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. Extract and load the database schema on the database server:

```
Unset
./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**:

```
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 13: 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
```

Step 14: 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. 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.

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
```

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}
    ]
}
```