



# ThreatConnect® Installation Guide: Containerized Deployment

Software Version 7.6

Technical Guide

June 18, 2024

10032-02 EN Rev. A



©2024 ThreatConnect, Inc.

ThreatConnect® is a registered trademark of ThreatConnect, Inc.

Amazon Web Services® and OpenSearch® are registered trademarks of Amazon Web Services.

Docker® is a registered trademark of Docker, Inc.

Elastic® is a registered trademark of Elasticsearch BV.

AlmaLinux OS™ is a trademark of Linux Foundation.

Java® is a registered trademark of Oracle Corporation.

Postgres® and PostgreSQL® are registered trademarks of the PostgreSQL Community Association of Canada.

Python® is a registered trademark of Python Software Foundation.

Redis® is a registered trademark of Redis Ltd.



# Table of Contents

---

<b>Overview</b> .....	<b>4</b>
<b>System Requirements</b> .....	<b>5</b>
Hardware .....	5
<b>Installing ThreatConnect</b> .....	<b>7</b>
Step 1: Configure Environment Variables .....	7
Step 2: Install the ThreatConnect License .....	7
Step 3: Create Certificates in the certs Folder .....	7
Step 4: Install Docker .....	9
Step 5: Install Docker Compose .....	9
Step 6: Install AWS CLI and Log Into ThreatConnect's ECR .....	10
Step 7: Increase vm.max_map_count .....	10
Step 8: Open Necessary Ports for ThreatConnect.....	11
Step 9: Reformat and Change Permissions on Shell Scripts.....	11
Step 10: Start ThreatConnect.....	12
Step 11: Monitor ThreatConnect .....	12
Step 12: Log Into ThreatConnect .....	13
Step 13: Configure OpenSearch on ThreatConnect .....	13
Step 14: Move the .env File to a Secure Location .....	13
Step 15: Restart and Monitor the ThreatConnect Containers.....	14



# 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, these software, along with ThreatConnect, are packaged together in a containerized solution using Docker®.

**Important:** 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. For instructions on installing ThreatConnect and having it run on an operating system (OS), see *ThreatConnect Installation Guide: Linux Operating System*.

**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

To install an On-Premises instance of ThreatConnect, the requirements in the following sections must be met.

## 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) <sup>1,2</sup>	Min CPU Cores / vCPUs (2GHz) <sup>2</sup>	Estimated Storage (GB) <sup>3,4</sup>
<b>ThreatConnect Application</b>	64	16	300
<b>Containerized Redis</b>	8	2	20
<b>Containerized OpenSearch</b>	16	2	200
<b>Containerized Database</b>	32	4	100

<sup>1</sup> Allocated to ThreatConnect Docker containers; the OS will need additional space.

<sup>2</sup> While Java virtual machines will be allocated memory, there is some allowance for additional memory available for Feed and Playbook Apps.

<sup>3</sup> High IOPS, ideally SSDs, are preferred.

<sup>4</sup> 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**

	<b>Highly Available Document Storage (usually network-mounted storage)</b>
<b>Document Storage</b>	Equal to the desired capacity of documents stored

**Table 3**

	<b>Memory Minimum (GB)</b>	<b>Memory Recommended (GB)</b>
<b>Swap Space</b>	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.



# Installing ThreatConnect

Follow all steps outlined in the following subsections to install ThreatConnect.

## Step 1: Configure Environment Variables

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.

## Step 2: Install the ThreatConnect License

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

## Step 3: Create Certificates in the certs Folder

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

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

For guidance on generating local certificates, see the following steps:

1. Create a certificate authority (CA). Before running the following commands, replace the placeholder `<Country>`, `<State>`, `<City>`, `<Organization>`, and `<Organizational Unit>` values:

```
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=<Organization>/OU=<Organizational
Unit>/CN=My Root Authority" -out my-root-ca.pem -days 3650
```

**Note:** If using custom CAs, set the filename in the `CUSTOM_CA_PEM_FILE` variable `.env` file. This file must reside in the `certs` folder.



2. Create a private key:

```
openssl genrsa -out privkey.pem 4096
```

3. Create a certificate signing request. Before running the following command, replace the placeholder `<Country>`, `<State>`, `<City>`, `<Organization>`, `<Organizational Unit>`, and `<FQDN/IP or Server>` values:

```
openssl req -new -sha256 -key privkey.pem -subj  
"/C=<Country>/ST=<State>/L=<City>/O=<Organization>/OU=<Organizational  
Unit>/CN=<FQDN/IP of Server>" -out <FQDN/IP of Server>.csr
```

4. Create a new file for alternate names (`alt-names.ext`). Before running the following commands, replace the placeholder `<FQDN/IP or Server>` value:

```
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. Before running the following command, replace the placeholder `<FQDN/IP or Server>` value:

```
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. Verify that the certificate contains the values entered for `C`, `ST`, `L`, `O`, `OU`, and `CN` in Step 1:

```
openssl x509 -in fullchain.pem -text -noout
```



## Step 4: 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 5: 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 6: 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
```

2. Run the following commands to configure the AWS CLI and enter the credentials provided to you by your ThreatConnect Customer Success Manager. Before running the commands, replace the placeholder `<access key id>` and `<secret access key>` values:

```
/usr/local/bin/aws configure
Access Key ID:<access key id>
Secret Access Key:<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 7: 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 8: Open Necessary Ports for ThreatConnect

Run the following commands to open the ports required to use ThreatConnect:

```
service firewalld start
firewall-cmd --permanent --zone=public --add-service=smtp && firewall-cmd --
permanent --zone=public --add-service=http && firewall-cmd --permanent --
zone=public --add-service=https && firewall-cmd --permanent --zone=public --add-
forward-port=port=25:proto=tcp:toport=2500 && firewall-cmd --permanent --
zone=public --add-forward-port=port=80:proto=tcp:toport=8080 && firewall-cmd --
permanent --zone=public --add-forward-port=port=443:proto=tcp:toport=8443 &&
firewall-cmd --permanent --direct --add-rule ipv4 nat OUTPUT 0 -p tcp -o lo --dport
443 -j REDIRECT --to-ports 8443 && firewall-cmd --permanent --zone=public --add-
port=62000/tcp && firewall-cmd --reload
firewall-cmd --permanent --add-port={5432/tcp,6379/tcp,9200/tcp,9600/tcp}
firewall-cmd --reload
```

## Step 9: Reformat and Change Permissions on Shell Scripts

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

```
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: Start 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. Start Postgres®:

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

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

```
./load_schema.sh
```

4. Start **tc-mon**:

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

5. Start **tc-app**:

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

6. Start **tc-job**:

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

## Step 11: Monitor ThreatConnect

Run the following command to monitor ThreatConnect:

```
docker-compose logs --tail=10 --follow tc-mon tc-app tc-job
```




## Step 12: Log Into ThreatConnect

Run the following command to log into ThreatConnect:

```
admin/password1
```

After logging in successfully, you will be prompted to change the password for the `admin` user account.

## Step 13: Configure OpenSearch on ThreatConnect

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.

5. (Optional) To view the health and status of the OpenSearch nodes, index, and cluster, select the **Health** tab.

**Warning:** Only *advanced users* should access the **Repair** and **Wipe** tabs, as the utilities on these tabs will delete existing data.

6. The indexing status will be logged in the log file specified for ThreatConnect. Review the log on the **Logs** tab of the **System Settings** screen.

## Step 14: 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 15: 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
```