



Install Cisco WAE

This section contains the following topics:

- [Install and Configure Supervisor, on page 1](#)
- [Verify WAE Image, on page 2](#)
- [Install Cisco WAE, on page 3](#)
- [Install Multi WAE, on page 5](#)
- [Upgrade from Cisco WAE 7.x, on page 8](#)
- [Upgrade from Cisco WAE 7.x to Multi WAE, on page 9](#)
- [Install Cisco WAE License, on page 9](#)
- [Start and Stop Cisco WAE, on page 10](#)
- [Migrate Configurations from Cisco WAE 7.x, on page 11](#)
- [Update Packages or Templates, on page 11](#)
- [Troubleshoot a Cisco WAE Installation, on page 12](#)

Install and Configure Supervisor

Install and configure supervisor before installing WAE.



Note The following configuration steps work only when supervisor is installed using yum. If supervisor is installed using any other method, it has to be configured to run **supervisorctl** as a non root user.

Step 1 Install supervisor and verify.

```
sudo yum install -y epel-release
sudo yum install -y supervisor
supervisord -version
4.2.1
```

Step 2 Create directories with write permissions for the OS user running WAE.

```
sudo mkdir -p /opt/supervisor/run
sudo mkdir -p /opt/supervisor/log
sudo chown -R [USER-NAME]:[GROUP-NAME] /opt/supervisor
```

Step 3 Update supervisor configuration to not run as a root user.

Point the pid file to `/opt/supervisor/run/supervisor.pid` and user as the OS user running WAE.

Open `/etc/supervisord.conf` as root and edit.

- In the `[unix_http_server]` section:

- Change `;file=/var/run/supervisor/supervisor.sock` to `file=/opt/supervisor/run/supervisor.sock`
- Change `;chown=nobody:nogroup` to `chown=[USER-NAME]:[GROUP-NAME]`

- In the `[supervisord]` section:

- Change `;logfile=/var/log/supervisor/supervisord.log` to `logfile=/opt/supervisor/log/supervisord.log`.
- Change `;pidfile=/var/run/supervisord.pid` to `pidfile=/opt/supervisor/run/supervisord.pid`
- Change `;minfds=1024` to `minfds=1000000`
- Change `;minprocs=200` to `minprocs=257805`

Note Do not set the user under the `[supervisord]` section.

- In the `[supervisorctl]` section:

- Change `;serverurl=unix:///var/run/supervisor/supervisor.sock` to `serverurl=unix:///opt/supervisor/run/supervisor.sock`

Step 4 Start Supervisor.

```
sudo systemctl start supervisord
sudo supervisorctl status all
```

Step 5 Enable supervisor to start during system startup.

```
sudo systemctl enable supervisord
sudo systemctl status supervisord
```

Verify WAE Image

Step 1 Download the Cisco WAE 7.6.0 software package from [Cisco Download Software](#) site.

Step 2 The certificate and digital signature are both embedded in the downloaded file - `wae-linux-v7.6.0.signed.bin`.

Step 3 Run the self-extracting signed binary. This extracts the Release Binary and validates using the signature file.

Verification of signed image

```
[admin@wae-vm-21 workspace.signed]$ ./wae-linux-v7.6.0.signed.bin
Unpacking...
Verifying signature...
Downloading CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...
Successfully downloaded and verified crcam2.cer.
Downloading SubCA certificate from http://www.cisco.com/security/pki/certs/innertime.cer ...
Successfully downloaded and verified innertime.cer.
Successfully verified root, subca and end-entity certificate chain.
Successfully fetched a public key from WAE-CCO_RELEASE.cer.
Successfully verified the signature of wae-linux-v7.6.0.bin using WAE-CCO_RELEASE.cer
```

Step 4 The generated `wae-linux-v7.6.0.bin` is the Linux installer for WAE.

Install Cisco WAE

Before you begin



Note If you want to upgrade from an older WAE 7.x release to WAE 7.6.0, see [Upgrade from Cisco WAE 7.x, on page 8](#)

- If one does not yet exist, create a UNIX user (assigned to a group). You must be this UNIX user to run installation.
 - Make sure Java 11 and Python 3.6.x are installed on the system. `JAVA_HOME` environment variable is pointing to `jdk-11.0` and `/usr/bin/python3` must point to the installed python.
 - Make sure supervisor is installed and configured. See [Install and Configure Supervisor, on page 1](#).
 - Download and verify the digitally signed Cisco WAE 7.6.0 image. See [Verify WAE Image, on page 2](#).
 - Make sure that `requests.auth` python package is installed for the BW-OPT application to function in WAE.
-

Step 1 Stop WAE if running.

Step 2 Change permission of the install file using the command:

```
chmod +x wae-linux-v7.6.0.bin
```

Step 3 Run the installer specifying the target directory.

```
./wae-linux-v7.6.0.bin <wae-dir>
```

Step 4 Navigate to installation directory to source `waerc`. Setup environment and create a runtime directory specifying the path.

```
cd <wae-dir>
source waerc
wae-setup --dest <target-runtime-dir>
```

Step 5 You are prompted to set the Cisco WAE admin password.

```
WAE admin password:
Confirm password:
```

Step 6 After installing and setting up wae (i.e. after running `wae-setup`), create a soft link to the `wae.ini` file from inside `/etc/supervisord.d/` and add WAE config to supervisor.

```
sudo ln -sf <target-runtime-dir>/wae.ini /etc/supervisord.d/
```

- Note**
- Execute this step only after supervisor is installed and configured.
 - If you want to use an external-executable-nimo based network which needs `JAVA_HOME/JRE_HOME` to be set, edit the section `[program:waectl]` inside `target-runtime-dir/wae.ini` file to include `JAVA_HOME="valid_jdk_path"` inside environment.

For example, under `[program:waectl]` edit to add:

```
JAVA_HOME:environment=HOME="/home/wae", NCS_JAVA_VM_OPTIONS="-Xmx32G -Xms16G
-XX:+UseG1GC -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/home/wae/test/run/logs/
-Djava.io.tmpdir=/home/wae/test/run/work/", TMPDIR="/home/wae/test/run/work/",
JAVA_HOME="/usr/"
```

- For the new `wae.ini` changes to come into effect, execute `supervisorctl update`.

Step 7 Update supervisor configuration.

```
sudo supervisorctl update
```

Step 8 Start WAE process

```
sudo supervisorctl start wae:*
wae:zookeeper: started
wae:waectl: started
wae:kafka: started
wae:wae-monitor: started
```

- Note**
- `wae:waectl` is the WAE program.
 - `wae:kafka` and `wae:zookeeper` are required for traffic collection and internal messaging.
 - `wae:wae-monitor` is the monitoring service.
 - `wae:logrotate` is for log rotation.

Step 9 Check status of WAE process

```
sudo supervisorctl status
wae:kafka RUNNING pid 1540, uptime 28 days, 14:03:40
wae:logrotate RUNNING pid 1178, uptime 28 days, 15:10:11
wae:wae-monitor RUNNING pid 11520, uptime 0:00:12
wae:waectl RUNNING pid 1177, uptime 28 days, 15:10:11
wae:zookeeper RUNNING pid 1736, uptime 28 days, 14:03:39
```

- Note** To stop all WAE process, use the command:

```
sudo supervisorctl stop wae:*
```

Step 10 To migrate configurations from a WAE 7.x.x release to WAE 7.6.0 release, use the Cisco WAE upgrade script from [Cisco Download Software](#) site.

- Note** If the server/VM is restarted, all the WAE services are not restarted automatically and they will be in the stopped state. They can be started using the command mentioned in Step 8.
-

Install Multi WAE

Before you begin

- Install Ansible version 2.10.7 or higher based on python3. Use the following command:


```
sudo yum install ansible
```
- Install Java 11 on all remote hosts.
- Install Python3 on all remote hosts as well as in the host where playbooks are run.



-
- Note**
- On RHEL 8.4, run the playbook from a terminal where **waerc** is not sourced.
 - Restart WAE in the scale primary whenever the number of splits increases.
-

- Enable passwordless ssh between servers participating in multi WAE (including self ssh).
-

Step 1 Export the ansible.cfg. A custom ansible.cfg file is provided at **playbooks/ansible.cfg**. Use the command:

```
export ANSIBLE_CONFIG=<path-to-the-ansible-config-file>
```

Step 2 On the machine where you intend to run the playbook from, add the entry to self in **playbooks/known_hosts** file by doing an SSH to yourself. The Multi WAE installation only supports single **username** and **wae_dir** across different machines. You can also pass **ansible_ssh_user** from CLI while invoking the **ansible-playbook** command by passing **-u** flag.

```
ansible-playbook wae_install.yml -u <username> --ask-pass
```

Step 3 Add the following line at the end of the **playbooks/visudo** file to make sure you can run the sudo commands without password

```
<username> ALL=(ALL) NOPASSWD:ALL
```

Step 4 Modify the **playbooks/hosts** file to include the IP addresses of the machines. The **hosts** file has 3 groups: **[remote]**, **[primary]** and **[secondary]**.

```
[remote]
'element-1' ansible_ssh_user='TARGET_SSH_USER'
'element-2' ansible_ssh_user='TARGET_SSH_USER'
'element-3' ansible_ssh_user='TARGET_SSH_USER'
```

```
[primary]
'element-1' ansible_ssh_user='TARGET_SSH_USER'
```

```
[secondary]
'element-2' ansible_ssh_user='TARGET_SSH_USER'
```

where,

```
[remote] - indicate the set of hosts in which the playbooks are to be run
[primary] - is the host which should be set as primary when configuring HA. Must be one of the host
           present in [remote] group.
```

[secondary] - is the host which should be set as secondary when configuring HA. Must be one of the host present in [remote] group.

- Note**
- [remote] group is compulsory for every playbook execution.
 - [primary], and [secondary] groups are required only for **ha_config** playbook execution.

Step 5 Set the input parameters required by playbooks in **group_vars/all** file. The file is present in **playbooks/group_vars/all** (refer to the following table) and execute the playbook. The following table lists the details of the available playbooks:

Table 1: Ansible Playbook Details

Playbook	Description	Parameters	Usage
wae_install.yml	The wae_install.yml playbook installs WAE on remote machines by copying the WAE binary and performing relevant checks and tasks that are needed to get the server up and running using supervisor.	<ul style="list-style-type: none"> • WAE_USER_NAME: WAE user (sudo capable) preexisting on all the remote machines. • WAE_BIN_PATH: Absolute path to the WAE binary located on the machine where ansible-playbook is run. • WAE_DIR: Absolute path of the WAE directory which will hold wae-install and wae-run directories. • DELETE_SIGNED: Flag used to indicate if we need to clean up signed WAE image after install is complete. Default value is false. 	<pre>ansible-playbook wae_install.yml -i <path_to_inventory_file> --ask-pass</pre>
kafka_config.yml	The kafka_config.yml playbook deploys kafka on remote machines by setting the right configurations for the internal and external listeners.	<ul style="list-style-type: none"> • WAE_DIR: Absolute path of the WAE directory which will hold wae-install and wae-run directories 	<pre>ansible-playbook kafka_config.yml -i <path_to_inventory_file> --ask-pass</pre>

Playbook	Description	Parameters	Usage
ha_config.yml	The ha_config.yml playbook deploys HA between two nodes given WAE is running.	<ul style="list-style-type: none">• WAE_USER_NAME: WAE user (sudo capable) preexisting on all the remote machines.• WAE_DIR: Absolute path of the WAE directory which will hold wae-install and wae-run directories.• WAE_HA_XML_TEMPLATE: XML template containing WAE HA config to be loaded on CDB of the two nodes.	<pre>ansible-playbook ha_config.yml -i <path_to_inventory_file> --ask-pass</pre>

Playbook	Description	Parameters	Usage
load_config.yml	The <code>load_config.yml</code> playbook is intended to load the WAE configs on the remote WAE server.	<ul style="list-style-type: none"> • WAE_DIR: Absolute path of the WAE directory which will hold wae-install and wae-run directories. • WAE_CFGS_SRC_DIR: Absolute path of the directory where the configs are present on the machine where ansible-playbook is run. • WAE_CFGS: List of names of the config files. The files should be present at WAE_CFGS_SRC_DIR. • WAE_TMP_CFGS_DEST_DIR: Absolute path to a directory in remote machines where the config files will be copied. <p>Directory will be created if it does not exist. Default value is /tmp/wae_cfgs.</p>	<pre>ansible-playbook load_config.yml -i <path_to_inventory_file> --ask-pass</pre>

Upgrade from Cisco WAE 7.x

Before you begin

- Download the Cisco WAE upgrade script from [Cisco Download Software](#) site.
- Download and verify the digitally signed Cisco WAE 7.6.0 image. See [Verify WAE Image, on page 2](#).
- Make sure Java 11 and Python 3.6.x are installed on the system. `JAVA_HOME` environment variable is pointing to `jdk-11.0` and `/usr/bin/python3` must point to the installed python.
- Install `pexpect` using the following command:

```
sudo pip3 install pexpect
```
- Make sure supervisor is installed and configured. See [Install and Configure Supervisor, on page 1](#).

- Disable HA before doing an upgrade. Upgrade script does not handle any configurations related to specific functional packs present in the previous WAE installation. You can:
 - Remove the configurations related to functional packs before doing the upgrade, or
 - Install WAE manually (See [Install Cisco WAE, on page 3](#)), install the functional packs in the new WAE installation and then import the configurations (see [Migrate Configurations from Cisco WAE 7.x, on page 11](#)).

Step 1 Login to the machine where 7.x is installed.

Step 2 Run the `wae_upgrade` script.

Note The installation file passed as `--wae-bin` option is the image obtained after verifying the digitally signed Cisco WAE 7.6.0 image.

```
# ./wae_upgrade --upgrade --old-install-dir <WAE_7.x_INSTALL_DIR> --old-run-dir <WAE_7.x_RUN_DIR>
--new-install-dir <WAE_7.6.0_INSTALL_DIR> --new-run-dir <WAE_7.6.0_RUN_DIR> --cfg-dir
<dir_to_save_config> --wae-bin <WAE_7.6.0_INSTALLATION_FILE>
where
--old-install-dir    indicates the directory where 7.x WAE is installed
--old-run-dir        indicates the directory where the run time for 7.x WAE resides
--new-install-dir    indicates the directory where 7.6.0 WAE must be installed
--new-run-dir        indicates the directory where the run time for 7.6.0 WAE will reside
--cfg-dir            indicates the folder where the config is to be saved. This config will be changed
                    to match 7.6.0 and pushed to 7.6.0
--wae-bin            indicates the path to WAE 7.6.0 installation file.
```

Upgrade from Cisco WAE 7.x to Multi WAE

Step 1 Upgrade your WAE installation to Cisco WAE 7.6. See [Upgrade from Cisco WAE 7.x, on page 8](#).

Step 2 Use ansible playbook `load_config.yml` to configure agent and Nimo and manually configure Multi WAE on the upgraded WAE instance. See [Install Multi WAE, on page 5](#)

Step 3 Run upgrade script with export option to collect the config from the updated WAE instance. See [Migrate Configurations from Cisco WAE 7.x, on page 11](#).

Step 4 Use the `wae_install` and `load_config` playbooks to install and configure WAE in the other WAE instances

Install Cisco WAE License

A license is required to use all the features in Cisco WAE. If you have questions about obtaining a license, contact your Cisco support representative or system administrator.

Cisco WAE supports both Cisco Smart Licensing and traditional licensing. If you would like to convert from a traditional license to Smart Licensing, see your Cisco WAE account representative. For information on the differences between the two types of licensing, refer to the [Cisco Smart Licensing Overview](#) on Cisco.com.

For information on Cisco Smart Licensing, see "Smart Licensing" chapter in *Cisco WAE User Guide*.

Install Traditional License

To install a traditional license:

-
- Step 1** Run `license_install` tool, and pass the name of license file (with `.lic` extension). By default, the tool merges all features that are granted by the new license with those features in an existing license.
- ```
license_install -file <path>/<license_name>.lic
```
- Step 2** When prompted, enter the number that is associated with the directory in which you want to install the license.
- Note**
- If option 2 (`<wae-dir>/etc`) is selected, you need to reinstall the license when a new build is installed.
  - If option 1 (`/.cariden/etc`) is selected, reinstalling the license is not necessary unless the license is expired.
  - Once the license is installed, you can verify the installed licenses by running the `license_check` command.
- Step 3** Stop and start WAE for the installed license to be picked up.
- 

## Install Smart License

To install a smart license:

- 
- Step 1** See "Smart License" section in User Guide, to configure Smart License.
- Step 2** Stop and start WAE for the installed license to be picked up.
- 

## Start and Stop Cisco WAE

From the Cisco WAE run-time directory, enter the relevant Cisco WAE CLI command to start or stop Cisco WAE services:

- Start WAE
 

```
sudo supervisorctl start wae:*
wae:zookeeper: started
wae:waectl: started
wae:kafka: started
wae:wae-monitor: started
```
- Stop WAE
 

```
sudo supervisorctl stop wae:*
```

# Migrate Configurations from Cisco WAE 7.x

You can use the Cisco WAE upgrade script utility to migrate configurations from WAE 7.x.

## Before you begin

- Download the Cisco WAE upgrade script for migrating configurations from WAE 7.x to WAE 7.6.0 package from [Cisco Download Software](#) site.
- Install WAE 7.6.0 and start the WAE process before you proceed with migrating configurations. See [Install Cisco WAE, on page 3](#)
- Install `pexpect` using the following command:
 

```
sudo pip3 install pexpect
```
- Disable HA before doing an upgrade.
- Upgrade script does not handle any configurations related to specific functional packs present in the previous WAE installation. You can:
  - Remove the configurations related to functional packs before exporting them, or
  - Install the functional packs in the new WAE installation before importing the configurations.

---

**Step 1** To take a backup of the 7.x configuration, login to the machine where 7.x is installed, and run `wae_upgrade` script with `--export` option.

```
./wae_upgrade --export --install-dir <WAE_7.x_INSTALL_DIR> --run-dir <WAE_7.x_RUN_DIR> --cfg-dir <dir_to_save_exported_config>
```

Where:

```
--install-dir indicates the directory where 7.x WAE is installed
--run-dir indicates the directory where the run time for 7.x WAE resides
--cfg-dir indicates the folder where backup of 7.x configuration must reside
```

**Step 2** To restore the 7.x configuration to 7.6.0, login to the machine where 7.6.0 is installed, and run `wae_upgrade` script with `--import` option.

```
./wae_upgrade --import --install-dir <WAE_7.6.0_INSTALL_DIR> --run-dir <WAE_7.6.0_RUN_DIR> --cfg-dir <dir_to_import_saved_config>
```

Where:

```
--install-dir indicates the directory where 7.6.0 WAE is installed
--run-dir indicates the directory where the run time for 7.6.0 WAE resides
--cfg-dir indicates the folder where backup of 7.x configuration resides
```

---

## Update Packages or Templates

If any packages or templates are updated or added in the `<wae_run_time_directory>/packages` directory, request a package reload using the Cisco-style WAE CLI:

```
$ packages reload
```

For example, perform a package reload when you edit the `wae.conf` file.

## Troubleshoot a Cisco WAE Installation

To check the status of Cisco WAE, enter `sudo supervisorctl status`.

Cisco WAE comes with standard logging features in the YANG run time. Cisco WAE logs to multiple log files in the `<wae-run-time>/logs` directory.

The LDAP authentication logs are logged in `[wae-run-time]/logs/wae-ldap-auth.log` file. The tool located in `[wae-install-dir]/lib/exec/test-java-ssl-conn` is useful to test SSL connectivity for java applications like LDAP Authentication and EPNM notifications which provide useful information to debug certification issues.

The most useful log is `<wae-run-time>/logs/wae-java-vm.log`. Most Cisco WAE packages log to this file. Some Cisco WAE packages also log to `<wae-run-time>/logs/wae-python-vm-<package-name>.log`. The following example shows Python-VM based logs:

```
[wae@wae logs]$ pwd
/home/wae/wae-run/logs
[wae@host logs]$ ls -ltr wae-python-vm*
-rw-rw-r-- 1 wae wae 0 Feb 26 07:50 wae-python-vm-cisco-wae-opm-tte.log
-rw-rw-r-- 1 wae wae 0 Feb 26 07:50 wae-python-vm-cisco-wae-get-plan.log
-rw-rw-r-- 1 wae wae 0 Feb 26 07:50 wae-python-vm-cisco-wae-dmdmesh-creator-nimo.log
-rw-rw-r-- 1 wae wae 0 Feb 26 07:50 wae-python-vm-cisco-wae-layout-nimo.log
-rw-rw-r-- 1 wae wae 0 Feb 26 07:50 wae-python-vm-cisco-wae-opm-load-plan.log
-rw-rw-r-- 1 wae wae 0 Feb 26 07:50 wae-python-vm-cisco-wae-dmddeduct-nimo.log
-rw-rw-r-- 1 wae wae 0 Feb 26 07:50 wae-python-vm-cisco-wae-archive.log
-rw-rw-r-- 1 wae wae 2238 Feb 26 07:50 wae-python-vm.log
-rw-rw-r-- 1 wae wae 270 Feb 26 08:20 wae-python-vm-nso_wae_nodes_insert.log
```

By default, the log level is set to INFO. You can configure logging in the following ways:

- Define the log level of various logs in the run-time directory `wae.conf` file. For information about the `wae.conf` file, see the *Cisco WAE User Guide*.
- Use the Expert Mode to set logging capabilities for some network interface modules (NIMOs). For example, you can set logging capabilities such as topology NIMOs and the `lsp-snmp-nimo` module. For information about the Expert Mode, see the [Cisco WAE User Guide](#).
- Use the Cisco WAE CLI to define the log level for various NIMO components. To define the log level, enter the following command at the command line:

```
admin@wae% set java-vm java-logging logger <nimo-component> level <level-x>
```

Level types are `level-info`, `level-debug`, and `level-all`. The logs are saved to `wae-java-vm.log` and can be used for troubleshooting.

The following table lists basic NIMO components.

| NIMO Component | Description       |
|----------------|-------------------|
| com.cisco.wae  | General debugging |

| NIMO Component                       | Description                             |
|--------------------------------------|-----------------------------------------|
| com.cisco.wae.nimo.topo              | Topology-based NIMO debugging           |
| com.cisco.wae.nimo.lspconfig         | LSP configuration through NED debugging |
| com.cisco.wae.nimo.lsp               | LSP debugging                           |
| com.cisco.wae.nimo.snmptrafficpoller | SNMP traffic poller debugging           |
| com.cisco.wae.dare                   | Aggregation debugging                   |
| com.cisco.wae.nimo.optical           | Optical NIMO debugging                  |

### ssh: symbol lookup error

When the waerc file is sourced, ssh and scp commands may fail due to WAE specific openssl libraries set in the LD\_LIBRARY\_PATH environment variable by waerc. The error message `ssh: symbol lookup error: /lib64/libk5crypto.so.3: undefined symbol: Camellia_cbc_encrypt, version OPENSSL_1_1_0` may appear.

To resolve this issue, use any of the following steps:

- Use ssh, scp, or any other operation which uses openssl on a terminal session where waerc is not sourced.
- If you are using ssh or scp commands after sourcing waerc, then set the LD\_LIBRARY\_PATH environment variable to the value it had before sourcing waerc.

