

Installing Apache Cassandra

Teamwork Cloud uses Apache Cassandra, an open-source NoSQL distributed database. Before installing Teamwork Cloud, please follow the steps below to install Apache Cassandra.

To install Apache Cassandra

1. Install Apache Cassandra by executing the `install_cassandra<version_number>_<os_version>.sh` installation script.

Example

```
sudo ./install_cassandra4x_centos_rhel_7.sh
```



The script downloads and installs the necessary packages, Cassandra, and the Cassandra tools from the Apache Software Foundation repository, and creates the necessary firewall rules to allow proper operation both for a single node or a cluster installation. OpenJDK 8 will be installed with Cassandra. Additionally, the installation script will install OpenJDK 11 and set the system environment to use the newer Java 11.

2. Start Apache Cassandra by executing the following command:

```
sudo systemctl start cassandra
```

3. Check if Apache Cassandra is running by executing the following command:

```
nodetool -h ::FFFF:127.0.0.1 status
```



You can also use the following command:

```
nodetool -Dcom.sun.jndi.rmiURLParsing=legacy status
```

If Apache Cassandra is running, you should receive the output displayed below. If the service is fully operational, the first 2 characters of the last line are "UN", indicating that the node status is Up, and its state is Normal.

```
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens       Owns (effective)  Host ID                               Rack
UN  127.0.0.1    128.4 KB      256          100.0%           ea3f99eb-c4ad-4d13-95a1-80aec71b750f  rack1
```



Wait for a few minutes until Cassandra starts for the first time before checking if it is running. If Cassandra has not started yet, you will get the error: "No nodes present in the cluster. Has this node finished starting up?" This means that you need to give Cassandra more time to start.

4. If Apache Cassandra is not running or if you used installation options other than the one described in this chapter, optionally [configure Apache Cassandra](#).

Configuring Apache Cassandra (optional)

If you used other installation options and not the provided script or if Apache Cassandra does not start, configure it as described below.



Before starting, note that you do not need to configure Apache Cassandra if you installed it using the installation script we provided (`install_cassandra<version_number>_<os_version>.sh`). It should start without any additional configuration.

To configure Apache Cassandra

1. Edit the `cassandra.yaml` file by executing the following command:

```
sudo nano /etc/cassandra/default.conf/cassandra.yaml
```

2. Find the following parameters related to the Cassandra node IP address and communication settings, and change them as shown below:

Example

```
seeds: "192.168.130.10"
listen_address: 192.168.130.10
broadcast_rpc_address: 192.168.130.10
rpc_address: 0.0.0.0
```

- **seeds** - a comma-delimited list containing all of the seeds in the Cassandra cluster. Since our cluster consists of a single node, it contains only one entry - our IP address.
- **listen_address** - the IP address that Cassandra uses to listen for connections.
- **broadcast_rpc_address** - the IP address used to broadcast to other Cassandra nodes in the cluster. This parameter may be commented. In such case, remove "#" and make sure there are no leading spaces.
- **rpc_address** - when set to 0.0.0.0, Cassandra listens to rpc requests on all interfaces.

3. Find the following parameters that control thresholds to ensure that the data being sent is processed properly, and change them as shown below:

Example

```
commitlog_segment_size_in_mb: 192
read_request_timeout_in_ms: 1800000
range_request_timeout_in_ms: 1800000
write_request_timeout_in_ms: 1800000
cas_contention_timeout_in_ms: 1000
truncate_request_timeout_in_ms: 1800000
request_timeout_in_ms: 1800000
batch_size_warn_threshold_in_kb: 3000
batch_size_fail_threshold_in_kb: 5000
```

4. To ensure that the default commit log size is 8GB (recommended), uncomment the **commitlog_total_space_in_mb** parameter as show as below.

Example

```
commitlog_total_space_in_mb: 8192
```



Ensure that the partition where the commit log is installed has enough space to accommodate a commit log of 8GB.

5. To point the data to the appropriate locations, find the following parameters and change them as shown below:

Example

```
data_file_directories:
- /data/data
commitlog_directory: /logs/commitlog
hints_directory: /data/hints
saved_caches_directory: /data/saved_caches
```

6. Start Apache Cassandra by executing the following command:

```
sudo systemctl start cassandra
```

7. Check if Apache Cassandra is running by executing the following command:

```
nodetool status
```

If Apache Cassandra is running, you should receive the output displayed below. If the service is fully operational, the first 2 characters of the last line are "UN", indicating that the node status is Up, and its state is Normal.

Example

```
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load       Tokens     Owns (effective)  Host ID                               Rack
UN  127.0.0.1    128.4 KB   256        100.0%           ea3f99eb-c4ad-4d13-95a1-80aec71b750f  rack1
```

Configuring Cassandra JVM node (optional)

If you did not use the installation script or want to increase the RAM usage by Cassandra, make the following changes. Otherwise, these configuration changes are set automatically by the Cassandra installation script.



Configuration files are located in `/etc/cassandra/conf/`.

- By default, the maximum RAM usage for Cassandra is 8GB. To change the amount of RAM used by Cassandra, uncomment **-Xms4G (min)** and **-Xmx4G (max)** in the `jvm-server.options` file and specify their values.
- In the `jvm11-server.options` and `jvm8-server.options` files, comment all lines from "### CMS Settings" to "### G1 Settings".
- In the `jvm11-server.options` and `jvm8-server.options` files, uncomment the following lines:

```
#-XX:+UseG1GC
#-XX:MaxGCPauseMillis=500
```

- In the `jvm11-server.options` and `jvm8-server.options` files, uncomment the following lines and set the values to the physical CPU core count (the values of both parameters should be the same):

```
#-XX:ParallelGCThreads=16
#-XX:ConcGCThreads=16
```

- In the `jvm8-server.options` file, comment all lines from "### GC logging options" to the end of the file.
- Synchronize CPU clocks on all Cassandra cluster nodes. Otherwise, you may encounter issues when creating an empty Cassandra cluster.
- When using `cqlsh`, use Python 3.6.0 or a later version. Python 2.7 series is no longer supported.
- In the `logback.xml` file, comment the `<appender-ref ref="ASYNCDEBUGLOG" />` line. This will increase Cassandra's performance by disabling the debug log.

Tuning Linux to improve Apache Cassandra performance



If you install Teamwork Cloud using the `install_twc<os_version>.sh` script, Cassandra performance is tuned automatically. However, if you plan to use other installation options or if you need to set other parameters after running the script, you can do it manually as described in this section.

To improve Apache Cassandra performance

1. Open the `sysctl.conf` file by executing the following command:

```
sudo nano /etc/sysctl.conf
```

2. To configure the TCP settings, add the following tuning parameters to the file:

Example

```
net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.core.optmem_max=40960
net.core.default_qdisc=fq
net.core.somaxconn=4096
net.ipv4.conf.all.arp_notify = 1
net.ipv4.tcp_keepalive_time=60
net.ipv4.tcp_keepalive_probes=3
net.ipv4.tcp_keepalive_intvl=10
net.ipv4.tcp_mtu_probing=1
net.ipv4.tcp_rmem=4096 12582912 16777216
net.ipv4.tcp_wmem=4096 12582912 16777216
net.ipv4.tcp_max_syn_backlog=8096
net.ipv4.tcp_slow_start_after_idle = 0
net.ipv4.tcp_tw_reuse = 1
vm.max_map_count = 1048575
vm.swappiness = 0
vm.dirty_background_ratio=5
vm.dirty_ratio=80
vm.dirty_expire_centisecs = 12000
```

3. To apply the setting without rebooting, execute the following command:

```
sudo sysctl -p
```

For more information about tuning Linux, see [DSE 6.8 Administrator Guide](#).