Installing Apache Cassandra 3 - EOL 2023

Cassandra 3 will reach its end of life in 2023. Teamwork Cloud 2021x Refresh 2 Hot Fix 5 works with both Cassandra 3 and 4. Upgrades and new deployments should install Cassandra 4. Cassandra 3 should only be installed for data migration or recovery.

Installing OpenJDK 8 (for Cassandra)

Installing OpenJDK 8 (for Cassandra)

```
yum -y install java-1.8.0-openjdk
```

Use this only if you need to install Oracle Java 8u202 (used with Cassandra) instead of OpenJDK

From the Java version list, please check that the recommended Oracle JVM version is compatible with the Teamwork Cloud version you are using. In order to consolidate all of the installed applications in a single location, we will be installing them under */opt/local/java*. To facilitate deployment, you may deploy using the associated script (**install_java_202.sh)**. Oracle no longer allows direct download of their JDK, so it must be downloaded offline and placed in the same location as the **install** scripts. The installation script extracts it into the proper location, invokes the alternative command to point the system to this instance (you may need to select it when prompted), and creates entries in */etc/environment*. Upon completing the installation, issue the following command:

```
java -version
```

You should receive output such as the following:

```
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
```

If properly installed, you will see Java identified as Java HotSpot(TM)

install_java_202.sh

```
#!/bin/bash
echo "========================================="
echo "Installing Oracle Java 1.8.0_202"
echo "========================================="
echo ""
echo "  Oracle Java can no longer be downloaded directly due to new authentication requirements"
echo ""
echo "  After manually downloading jdk-8u202-linux-x64.tar.gz, copy it to this directory"
echo ""
echo "  Latest version is available from https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.
html"
echo ""
echo "  Archive downloads available from https://www.oracle.com/java/technologies/javase/javase8-archive-
downloads.html"
echo ""
read -p -"Press any key to continue, Ctl-C to exit ...: " -n1 -s
echo ""
echo "========================================="
sudo mkdir -p /opt/local/java
sudo tar xzf jdk-8u202-linux-x64.tar.gz -C /opt/local/java
cd /opt/local/java/jdk1.8.0_202/
sudo alternatives --install /usr/bin/java java /opt/local/java/jdk1.8.0_202/bin/java 2
sudo alternatives --config java
sudo alternatives --install /usr/bin/jar jar /opt/local/java/jdk1.8.0_202/bin/jar 2
sudo alternatives --install /usr/bin/javac javac /opt/local/java/jdk1.8.0_202/bin/javac 2
sudo alternatives --set jar /opt/local/java/jdk1.8.0_202/bin/jar
sudo alternatives --set javac /opt/local/java/jdk1.8.0_202/bin/javac
sudo echo 'JAVA_HOME=/opt/local/java/jdk1.8.0_202' > /etc/environment
sudo chown -R root:root /opt/local/java/jdk1.8.0_202
```

Installing Apache Cassandra 3.11.x

The deployment script for Cassandra removes Datastax Community Edition 2.2.x as well as OpsCenter and the Datastax Agent (which are not compatible with Cassandra 3.x), downloads and installs Cassandra 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. To install, execute the installation script (**install_cassandra<version_number>_<os_version>.sh**).

install_cassandra3_11_centos7.sh

```
#!/bin/bash
echo "========================================="
echo "Installing Apache Cassandra 3.11.x"
echo "========================================="
echo "Removing Datastax Community Edition"
yum remove -y datastax-agent
yum remove -y opscenter
yum remove -y cassandra22-tools
yum remove -y cassandra22
yum remove -y dsc22
rm -f /etc/yum.repos.d/datastax.repo
echo "Creating Apache Cassandra Repository File"
echo "[cassandra]" > /etc/yum.repos.d/cassandra.repo
echo "name=Apache Cassandra" >> /etc/yum.repos.d/cassandra.repo
echo "baseurl=http://www.apache.org/dist/cassandra/redhat/311x/" >> /etc/yum.repos.d/cassandra.repo
echo "gpgcheck=1" >> /etc/yum.repos.d/cassandra.repo
echo "repo_gpgcheck=1" >> /etc/yum.repos.d/cassandra.repo
echo "gpgkey=https://www.apache.org/dist/cassandra/KEYS" >> /etc/yum.repos.d/cassandra.repo
yum install -y epel-release
yum install -y cassandra
yum install -y cassandra-tools
yum install -y jemalloc
chkconfig --add cassandra
chkconfig cassandra on
echo "======================="
echo "Configuring firewall"
echo "======================="
FWZONE=$(firewall-cmd --get-default-zone)
echo "Discovered firewall zone $FWZONE"
cat <<EOF | tee /etc/firewalld/services/cassandra.xml
<?xml version="1.0" encoding="utf-8"?>
<service version="1.0">
    <short>cassandra</short>
    <description>cassandra</description>
    <port port="7000" protocol="tcp"/>
    <port port="7001" protocol="tcp"/>
        <port port="9042" protocol="tcp"/>
        <port port="9160" protocol="tcp"/>
        <port port="9142" protocol="tcp"/>
</service>
EOF
sleep 5
firewall-cmd --zone=$FWZONE --remove-port=7000/tcp --permanent  &> /dev/null
firewall-cmd --zone=$FWZONE --remove-port=7001/tcp --permanent  &> /dev/null
firewall-cmd --zone=$FWZONE --remove-port=7199/tcp --permanent  &> /dev/null
firewall-cmd --zone=$FWZONE --remove-port=9042/tcp --permanent  &> /dev/null
firewall-cmd --zone=$FWZONE --remove-port=9160/tcp --permanent  &> /dev/null
firewall-cmd --zone=$FWZONE --remove-port=9142/tcp --permanent  &> /dev/null
firewall-cmd --zone=$FWZONE --add-service=cassandra --permanent
firewall-cmd --reload
echo "====================================================="
echo "Changing ownership of data and commit log directories"
echo "====================================================="
mkdir /data &> /dev/null
mkdir /logs &> /dev/null
chown cassandra:cassandra /data &> /dev/null
chown cassandra:cassandra /logs &> /dev/null
echo "====================================================="
echo "Making configuration file changes"
echo "====================================================="
IP_ADDRESS=$(ip route get 1 | awk '{print $NF;exit}')
```

```
HOSTNAME=$(hostname)
cp /etc/cassandra/default.conf/cassandra.yaml /etc/cassandra/default.conf/cassandra.yaml.backup
cp /etc/cassandra/default.conf/cassandra.yaml ./cassandra.yaml.template
sed -i "s/ - seeds: \"127.0.0.1\"/ - seeds: \"$IP_ADDRESS\"/g" cassandra.yaml.template
sed -i "s/listen_address:.*/listen_address: $IP_ADDRESS/g" cassandra.yaml.template
sed -i "s/# broadcast_rpc_address:.*/broadcast_rpc_address: $IP_ADDRESS/g" cassandra.yaml.template
sed -i "s/broadcast_rpc_address:.*/broadcast_rpc_address: $IP_ADDRESS/g" cassandra.yaml.template
sed -i "s/# commitlog_total_space_in_mb:.*/commitlog_total_space_in_mb: 8192/g" cassandra.yaml.template
sed -i "s/commitlog_total_space_in_mb:.*/commitlog_total_space_in_mb: 8192/g" cassandra.yaml.template
sed -i "s/^rpc_address:.*/rpc_address: 0.0.0.0/g" cassandra.yaml.template
sed -i "s/start_rpc:.*/start_rpc: true/g" cassandra.yaml.template
sed -i "s/thrift_framed_transport_size_in_mb:.*/thrift_framed_transport_size_in_mb: 100/g" cassandra.yaml.
template
sed -i "s/commitlog_segment_size_in_mb:.*/commitlog_segment_size_in_mb: 192/g" cassandra.yaml.template
sed -i "s/read_request_timeout_in_ms:.*/read_request_timeout_in_ms: 1800000/g" cassandra.yaml.template
sed -i "s/range_request_timeout_in_ms:.*/range_request_timeout_in_ms: 1800000/g" cassandra.yaml.template
sed -i "s/write_request_timeout_in_ms:.*/write_request_timeout_in_ms: 1800000/g" cassandra.yaml.template
sed -i "s/cas_contention_timeout_in_ms:.*/cas_contention_timeout_in_ms: 1000/g" cassandra.yaml.template
sed -i "s/truncate_request_timeout_in_ms:.*/truncate_request_timeout_in_ms: 1800000/g" cassandra.yaml.template
sed -i "s/request_timeout_in_ms:.*/request_timeout_in_ms: 1800000/g" cassandra.yaml.template
sed -i "s/batch_size_warn_threshold_in_kb:.*/batch_size_warn_threshold_in_kb: 3000/g" cassandra.yaml.template
sed -i "s/batch_size_fail_threshold_in_kb:.*/batch_size_fail_threshold_in_kb: 5000/g" cassandra.yaml.template
sed -i '/data_file_directories:.*/!b;n;c\ \ \ \ - \/data\/data' cassandra.yaml.template
sed -i "s/hints_directory:.*/hints_directory: \/data\/hints/g" cassandra.yaml.template
sed -i "s/commitlog_directory:.*/commitlog_directory: \/logs\/commitlog/g" cassandra.yaml.template
sed -i "s/saved_caches_directory:.*/saved_caches_directory: \/data\/saved_caches/g" cassandra.yaml.template
\cp -fR ./cassandra.yaml.template /etc/cassandra/default.conf/cassandra.yaml
# Apply fix to systemd vulnerability preventing service control of cassandra
cat << EOF > /etc/systemd/system/cassandra.service
[Unit]
Description=Apache Cassandra
After=network.target

[Service]
PIDFile=/var/run/cassandra/cassandra.pid
User=cassandra
Group=cassandra
ExecStart=/usr/sbin/cassandra -f -p /var/run/cassandra/cassandra.pid
Restart=always
LimitNOFILE=100000
LimitMEMLOCK=infinity
LimitNPROC=32768
[Install]
WantedBy=multi-user.target
EOF
chkconfig --del cassandra
systemctl daemon-reload
systemctl enable cassandra
```

Configuring cassandra.yaml

Now, proceed to edit */etc/cassandra/default.conf/cassandra.yaml:*

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


The script above makes all of the changes to the configuration files stated below. However, please verify that all of them have been made.

The first items we will be editing relate to the IP address of the Cassandra node and communication settings. In our diagram above, this IP address is **192.168.130.10**. You will need to search for 3 keys in the configuration file and modify them accordingly. The seeds parameter is a comma-delimited list containing all of the seeds in the Cassandra cluster. Since our cluster consists of only a single node, it contains only one entry - our IP address. The other 2 parameters contain the IP address on which Cassandra listens for connections and the IP address to broadcast to other Cassandra nodes in the cluster. The **broadcast_rpc_address** may be commented out using a # character. If so, remove the "#" and make sure there are no leading spaces.

Additionally, we need to set **rpc_address** to *0.0.0.0* (meaning, it will listen to rpc requests on all interfaces), and **start_rpc** to *true* (so it will process rpc requests).

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

The next set of parameters controls thresholds to ensure that the data being sent is processed properly.

```
thrift_framed_transport_size_in_mb: 100
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
```

If you have installed your commit log in its own partition, the default commit log size will be the lesser of ¼ of the partition size of 8GB. In order to ensure that the recommended 8GB is used, you must uncomment the **commitlog_total_space_in_mb**, such that it will show as below. However, if you are uncommenting this value, please ensure that the partition has enough space to accommodate an 8GB commit log.

```
commitlog_total_space_in_mb: 8192
```

The next step is to point the data to the new locations. There are 4 entries that will be modified: **data_file_directories**, **commitlog_directory**, **hints_directory**, and **saved_caches_directory**. Search for these keys and edit them as follows:

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

After you have made these changes, save the **cassandra.yaml** file. Now, start the related services, as follows:

```
systemctl start cassandra
```

Now, proceed to check if Cassandra is running. To do this, issue the following command:

```
nodetool status
```

If the service is running, you will receive output such as below:

```
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
```

If the service is fully operational, the first 2 characters on the last line will state "**UN**", indicating the node's status is **U**p, and its state is **N**ormal.