Install and Configure Cassandra for C3PO Core

Contents

Check the installation prerequisites 2

 Memory 2

CPU recommendations 2

Disk space 2

Software 2

Prepare for installation 2

Update kernel settings 3

Configure firewall port access 5

Install Cassandra nodes 5

Install Java 8 5

Create the Debian package 5

Update the configure settings for the node 6

Start the Cassandra service on the node 8

Validate Cassandra on the node 8

Configure remaining nodes 8

Create the database schema 9

Stop the Cassandra service on the node 9

Uninstall Cassandra 9

Uninstall Java 8 9

Reference 10

## Check the installation prerequisites

Verify that the system meets the following prerequisites for memory, CPU, disk space, and software.

### Memory

The more memory a Cassandra node has, the better its read performance. More RAM also allows memory tables (memtables) to hold more recently written data. Larger memtables lead to a fewer number of SSTables being flushed to disk and fewer files to scan from disk during a read. The ideal amount of RAM depends on the anticipated size of your hot data. For both dedicated hardware and virtual environments, the recommended memory is as follows.

* In Production, 32 GB to 512 GB, where the minimum is 8 GB
* In Development for non-loading testing environments, the minimum is 4 GB

### CPU recommendations

* Dedicated hardware in production environments, 16-core CPU processors are the current price-performance sweet spot
* Dedicated hardware in development for non-loading testing environments, 2-core CPU processors are sufficient

### Disk space

Cassandra writes data to disk when appending data to the commitlog for durability and when flushing meltable to SSTable data files for persistent storage. The commit log has a different access pattern (read/writes ratio) than the pattern for accessing data from SSTables. This access pattern is more important for spinning disks than for SSDs.

SSTables are periodically compacted. Compaction improves performance by merging and rewriting data and discarding old data. However, depending on the type of compaction and size of it, during compaction disk utilization and data directory, volume temporarily increases. For this reason, be sure to leave a sufficient amount of free disk space available on each node. Refer to this [link](http://docs.datastax.com/en/archived/cassandra/2.0/cassandra/architecture/architecturePlanningDiskCapacity_t.html) for more information about disk space.

### Software

* OS - Red hat Linux 6 / Ubuntu
* Use OpenJDK 7 or 8 (we are using OpenJDK 7)
* Python 2.7

## Prepare for installation

This document assumes the following architecture for the Cassandra test environment.

* 1 cluster
* 4 data centers
* 20 nodes

Gather the following information before you start the installation.

1. Choose a name for the cluster. The examples in this document use "HSS Cluster".
2. Gather a list of the IP addresses of the 20 nodes.
3. Select at least four nodes to be seed nodes (at a minimum, you need one seed node per data center). Note the IP addresses of the seed nodes.
4. Use the following snitch and replication strategy.

* Set GossipingPropertyFileSnitch, which is a configuration setting in the cassandra.yaml file for each node.
* Supply the replication factor per data center using NetworkTopologyStrategy when you create the keyspace.

1. Determine the naming convention for each data center and rack. For example, you could use the following convention: DC1, DC2 or DC100, DC200 and RAC1, RAC2 or R101, R102. Choose the names carefully; you cannot rename a data center.

### Update kernel settings

Make the following Kernel setting changes on each node before proceeding.

1. Synchronize the clocks on all nodes, using NTP (Network Time Protocol) or another method. Cassandra only overwrites a column if there is another version whose timestamp is more recent.
2. To handle thousands of concurrent connections used by Cassandra, optimize the Linux network stack by adding the following settings to /etc/sysctl.conf.

net.core.rmem\_max = 16777216

net.core.wmem\_max = 16777216

net.core.rmem\_default = 16777216

net.core.wmem\_default = 16777216

net.core.optmem\_max = 40960

net.ipv4.tcp\_rmem = 4096 87380 16777216

net.ipv4.tcp\_wmem = 4096 65536 16777216

After changing the settings, run one of the following commands (as appropriate for your distribution).

$ sudo sysctl -p /etc/sysctl.conf

$ sudo sysctl -p /etc/sysctl.d/filename.conf

1. Disable the zone\_reclaim\_mode on NUMA systems. The Linux kernel can be inconsistent in enabling/disabling zone\_reclaim\_mode, which can result in odd performance problems. To ensure that zone\_reclaim\_mode is disabled, run the following command.

$ echo 0 > /proc/sys/vm/zone\_reclaim\_mode

1. Set user resource limits. Use the ulimit -a command to view the current limits. Then, make the following changes.
   1. Include the following settings in the /etc/security/limits.d/cassandra.conf or /etc/security/limits.conf file.

<cassandra\_user> - memlock unlimited

<cassandra\_user> - nofile 100000

<cassandra\_user> - nproc 32768

<cassandra\_user> - as unlimited

* 1. If you run Cassandra as root, some Linux distributions (such as Ubuntu) require setting the limits for root explicitly instead of using cassandra\_user:

root - memlock unlimited

root - nofile 100000

root - nproc 32768

root - as unlimited

* 1. For RHEL 6.x-based systems, set the nproc limits in /etc/security/limits.d/90-nproc.conf.

cassandra\_user - nproc 32768

* 1. For installations on Debian and Ubuntu operating systems, by default, the pam\_limits.so module is not enabled. To enable it, edit the /etc/pam.d/su file and uncomment the following line. This change to the PAM configuration file ensures that the system reads the files in the /etc/security/limits.d directory.

session required pam\_limits.so

* 1. For all installations, add the following line to /etc/sysctl.conf:

vm.max\_map\_count = 1048575

* 1. To make the changes take effect, reboot the server or run the following command:

$ sudo sysctl -p

* 1. To confirm the limits are applied to the Cassandra process, run the following command where pid is the process ID of the currently running Cassandra process:

$ cat /proc/pid/limits

1. Disable swap. Failure to disable swap entirely can severely lower performance. Cassandra has multiple replicas and transparent failover. Therefore, when memory is low it is better for a replica to be killed immediately than go into swap. This allows the immediate redirection to a functioning replica instead of continuing to hit a replica with high latency due to swapping. If your system has a lot of DRAM, swapping still lowers performance significantly because the OS swaps out executable code so that more DRAM is available for caching disks.

$ sudo swapoff --all

To make this change permanent, remove all swap file entries from /etc/fstab.

### Configure firewall port access

If you have a firewall running on the nodes in your Cassandra cluster, you must open up the following ports on each node to allow bi-directional communication among the nodes. If you don't open these ports, the node acts as a stand-alone server instead of joining the database cluster.

| **Type** | **Port** | **Description** |
| --- | --- | --- |
| Public ports | 22 | SSH port |
| 8888 | OpsCenter website. The opscenterd daemon listens on this port for HTTP requests coming directly from the browser. |
| Cassandra inter-node ports | 7000 | Cassandra inter-node cluster communication. |
| 7001 | Cassandra SSL inter-node cluster communication. |
| 7199 | Cassandra JMX monitoring port |
| Cassandra client ports | 9042 | CQL native clients port |
| 9160 | Cassandra client port (Thrift) |
| Cassandra OpsCenter ports | 61620 | OpsCenter monitoring port. The opscenterd daemon listens on this port for TCP traffic coming from the agent. |
| 61621 | OpsCenter agent port. The agents listen on this port for SSL traffic initiated by OpsCenter. |

## Install Cassandra nodes

Because you must install Cassandra on 20 nodes, you will likely automate the installation using Ansible. As you create that script, use instructions similar to the following installation steps to create Debian packages.

### Install Java 8

1. Add the Java repository.

sudo add-apt-repository ppa:webupd8team/java

1. Update the package list:

sudo apt-get update

1. Install Java 8:

sudo apt-get install oracle-java8-installer

### Create the Debian package

1. Add the Apache repository of Cassandra to /etc/apt/sources.list.d/cassandra.sources.list, for example for version 2.1.x using the following command.

echo "deb http://www.apache.org/dist/cassandra/debian 21x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list

1. Add the Apache Cassandra repository keys:

curl https://www.apache.org/dist/cassandra/KEYS | sudo apt-key add -

1. Update the repositories using the following command.

sudo apt-get update

1. If you encounter a public key error similar to the following: *GPG error: http://www.apache.org 36x InRelease: The following signatures couldn't be verified because the public key is not available: NO\_PUBKEY A278B781FE4B2BDA*, set up the key and then repeat the command to update the repositories. The following command uses A278B781FE4B2BD as an example public key. You must set the actual public key provided in your message.

sudo apt-key adv --keyserver pool.sks-keyservers.net --recv-key A278B781FE4B2BDA

sudo apt-get update

1. Install Cassandra:

sudo apt-get install cassandra

Service to the node should start automatically.

1. To verify that Cassandra is running on the node, on the command line, execute the following command.

nodetool status

1. Before you can configure the node, you must stop the service. Because the Debian packages start the Cassandra service automatically, you must stop the server and clear the data. Doing this removes the default cluster\_name (Test Cluster) from the system table. All nodes must use the same cluster name.

$ sudo service cassandra stop

$ sudo rm -rf /var/lib/cassandra/data/system/\*

$ sudo rm -rf /var/lib/cassandra/commitlog/\*

$ sudo rm -rf /var/lib/cassandra/data/system\_trace/\*

$ sudo rm -rf /var/lib/cassandra/saved\_cache/\*

1. Run the package using the steps 1-7 on each of the remaining nodes.

### Update the configure settings for the node

1. With the Cassandra service stopped, update the **/etc/cassandra/cassandra.yaml** file with the settings shown in the following table. In the following table, replace <*node IP*> with the actual hostname or IP address for the node. Change only the settings in the following table.

| **cassandra.yaml Setting** | **Description** |
| --- | --- |
| cluster\_name: **"HSS Cluster"** | Supply your cluster name. |
| seed\_provider:  - class\_name: **org.apache.cassandra.locator.SimpleSeedProvider**  - seeds: **"<*node IP for dc 1*>", "<*node IP for dc 2*>", "<*node IP for dc 3*>", "<*node IP for dc 4*>"** | For class\_name, use the supplied class to implement the SeedProvider interface.  For seeds, list the same four (at a minimum) seed node IP addresses in a comma-delimited list that you set in the "[Prepare for installation](#Ref4792372981)" section of this document.  Each of the 20 nodes uses a local cassandra.yaml file. Use the same class name and four seed node IP addresses in every local file. |
| listen\_address: **"<*node IP*>"** | For <*node IP*>, supply the IP address for the current node. This is the node IP address (or host name) that Cassandra binds to and uses to connect to other nodes. |
| broadcast\_address: **"<*node IP*>"** | For <*node IP*>, supply the IP address for the current node. This is the IP address to broadcast or communicate with other nodes when you have a firewall between the nodes. (If you do not have a firewall between nodes, leave it blank to use the same setting as listen\_address.)  In a cluster setup where one or more nodes is behind a firewall and has a private IP address, listen\_address does not allow the hosts behind the firewalls to be discovered by other nodes.  broadcast\_address falls back to listen\_address when it is not stated. |
| rpc\_address: **"<*node IP*>"** | For <*node IP*>, supply the IP address for the current node. The address or interface that binds the Thrift RPC service and native transport server. |
| endpoint\_snitch: **GossipingPropertyFileSnitch** | Set this to GossipingPropertyFileSnitch, the full class name of the "snitch". The cassandra-rackdc.properties file defines the rack and data center for the local node. |

1. Update the **/etc/cassandra /cassandra-rackdc.properties** file with the settings shown in the following table.

|  |  |
| --- | --- |
| **cassandra-rackdc.properties Setting** | **Description** |
| dc=**<*data center name*>** | Supply the case-sensitive data center name for this node that you determined the "[Prepare for installation](#Ref4792372982)" section of this document |
| rack=**<*rack name*>** | Supply the case-sensitive rack name for this node that you determined in the ""[Prepare for installation](#Ref4792372983)" section of this document |
| prefer\_local=**true** | Remove the comment (#) from prefer\_local to use the local IP address when communication is not across different data centers. Using the "true" setting saves bandwidth. |

After changing these settings, you must start the node.

### Start the Cassandra service on the node

We recommend starting the seed nodes first, then starting the additional nodes.

1. To start the node, run the following command.

sudo service cassandra start

### Validate Cassandra on the node

1. Use the following command to verify that the node is running. You can run this command on any node.

nodetool status

The following output shows an example of text that appears on the screen when you run the status command.

Datacenter: DC1-west

====================

Status=Up/Down

|/ State=Normal/Leaving/Joining/Moving

--  Address        Load       Tokens  Owns    Host ID                             Rack

UN  10.78.137.154  31.88 MB   256     ?       4215682d-0836-4c70-9d54-c15e39e34471  RAC2

UN  10.78.137.141  27.69 MB   256     ?       17507af7-dead-4549-be39-2d97475d47b0  RAC1

UN  10.78.137.151  32.45 MB   256     ?       13309aa2-5fc4-4cea-a154-b26fdf09a3c7  RAC1

### Configure remaining nodes

For each of the remaining 19 modes, repeat the configuration changes to reflect the current node:

* In the cassandra.yaml file, change the **<*node IP*>**.
* In the cassandra-rackdc.properties file, change the data center and rack.

### Create the database schema

Refer to the [Cassandra Create Initial Schema](https://sites.sprint.com/network/technology/_layouts/15/WopiFrame.aspx?sourcedoc=/network/technology/IOT%2520NFV%2520Docs/Install%2520and%2520Configure/Cassandra/Cassandra%2520Initialize%2520Schema.docx&action=default) document and the vHSS\_schema.sh script available on the IOT NPV wiki on the Sprint network. After the schema is successfully created, the Cassandra instance can be used.

# Stop the Cassandra service on the node

Use the following step if you need to stop Cassandra on a node.

1. To stop a node, run the following command.

sudo service cassandra stop

# Uninstall Cassandra

The following procedure shows how to uninstall the Cassandra Debian packages.

1. Stop the Cassandra service.

$ sudo service cassandra stop

1. Verify that all services are stopped.

$ ps auwx | grep cassandra

1. If services are still running, use the PID to kill the service.

$ sudo kill *cassandra\_pid*

1. Remove the library and log directories.

$ sudo rm -r /var/lib/cassandra

$ sudo rm -r /var/log/cassandra

1. Remove the installation directories.

$ sudo apt-get remove "cassandra\*"

$ sudo apt-get purge "cassandra\*"

$ sudo apt-get autoremove

# Uninstall Java 8

The following procedure shows how to uninstall the Cassandra Debian packages.

1. Remove the Java package.

$ sudo apt-get remove oracle-java8-installer

1. Purge the Java package.

$ sudo apt-get purge oracle-java8-installer

1. Remove any unused packages.

$ sudo apt-get auto remove

# Reference

Datastax provides documentation for the open source versions of Cassandra. This includes everything about Cassandra, including architecture, installation, Cassandra Query Language (CQL), operations, and configuration.

<http://docs.datastax.com/en/cassandra/2.1/>