



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Web Services and Applications Deployment Guide

Installing and Deploying Cassandra 2.2

4/7/2025

---

## Contents

- 1 Installing and Deploying Cassandra 2.2
  - 1.1 Installing Cassandra
  - 1.2 Configuring Cassandra
  - 1.3 Verifying the Cassandra installation
  - 1.4 Next step

# Installing and Deploying Cassandra 2.2

## Installing Cassandra

Complete this procedure for each Cassandra node.

### Prerequisites

- If you are using Web Services and Application v8.5.2.41 or earlier, a 2.1.4 version of the [Cassandra distribution](#) needs to be downloaded first. This package includes the `cassandra-cli` tool that you need to load the schema into your Cassandra 2.2 cluster.
- You have installed the latest [Java SE Development Toolkit 8](#). For more information, refer to the [Java documentation](#).

### Start

1. [Download the latest 2.2.x version of Cassandra](#).
2. Copy the Cassandra archive to the installation directory. For example, `/usr/local`
3. Use a tar utility to extract the files. For example, `tar -zxvf apache-cassandra-2.2.7-bin.tar.gz`
4. Add directories for data, commitlog, and saved\_caches. You can create these directories anywhere or in the default locations configured in the `Cassandra_install_dir/conf/cassandra.yaml` file. For example:
  - `/var/lib/cassandra/data`
  - `/var/lib/cassandra/commitlog`
  - `/var/lib/cassandra/saved_caches`
5. Add a directory for logging. You can create this directory anywhere, such as `/var/log/cassandra/`.

### End

## Configuring Cassandra

The procedures below describe how to create the Cassandra keyspace for the following scenarios:

- Development: 1 Cassandra node (appropriate for a development or lab environment)
- Single Data Center: 1 data center with a minimum of three Cassandra nodes

### Important

For more complex Cassandra deployments, please consult with Genesys

Select a tab below for the procedure that matches your deployment scenario.

## Development

### Configuring Cassandra (1 Cassandra node)

### Important

The files modified in this procedure are typically found in the ***Cassandra\_install\_dir/conf*** directory.

#### Prerequisites

- [Installing Cassandra](#)

#### Start

1. Modify the **cassandra.yaml** file:
  - a. Set seeds to the list of host name of the node. For example: `-seeds: "127.0.0.1"`
  - b. Set `listen_address` and `rpc_address` to the host name.
  - c. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra](#).
  - d. Set the `start_rpc` parameter to `true`.
5. Save your changes and close the file.
6. Open the **log4j-server.properties** file and set the `log4j.appender.R.File` property to the directory you created in Step 5 of [Installing Cassandra](#).
7. Save your changes and close the file.

#### End

## Single Data Center

## Configuring Cassandra (1 data center)

Complete the steps below for each node.

### Important

The files modified in this procedure are typically found in the **`Cassandra_install_dir/conf`** directory.

### Prerequisites

- [Installing Cassandra](#)

### Start

1. Modify the **`cassandra.yaml`** file:
  - a. Set the `cluster_name`. It must be the same name on all nodes.
  - b. Set the `initial_token` according to the node's place in ring. It must be one of the following:

```
Node #1: -9223372036854775808
Node #2: -3074457345618258603
Node #3: 3074457345618258602
```

### Important

The tokens shown here can be used for a three-node Cassandra cluster in a single data center. If you are using a different topology or cluster size, [consult the Cassandra documentation](#).

- c. Set `seeds` to the list of host names of all nodes. For example: `-seeds: "node1, node2, node3"`
  - d. Set `listen_address` and `rpc_address` to the host name.
  - e. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra](#).
  - f. Change `endpoint_snitch` to `PropertyFileSnitch`.
7. Save your changes and close the file.
  8. Open the **`log4j-server.properties`** file and set the `log4j.appender.R.File` property to the directory you created in Step 5 of [Installing Cassandra](#).
  9. Save your changes and close the file.
  10. Open the **`cassandra-topology.properties`** file and update for your cluster topology. For each node in your cluster, add the following line:

```
[node]=[datacenter]:[rack]
```

Where:

- *[node]* is the IP address of the node.
- *[datacenter]* is the name of the data center for this node.
- *[rack]* is the name of the rack for this node.

The following is a sample **cassandra-topology.properties** file for a Single Data Center scenario:

```
192.0.2.10=datacenter1:rack1
192.0.2.11=datacenter1:rack1
192.0.2.12=datacenter1:rack1
```

11. Save your changes and close the file.

**End**

## Verifying the Cassandra installation

### Prerequisites

- [Configuring Cassandra](#)

### Start

1. Start all Cassandra nodes using the following command: *Cassandra\_install\_dir/bin/cassandra*
2. Use the nodetool utility to verify that all nodes are connected by entering the following command: *Cassandra\_install\_dir/bin/nodetool -h Cassandra\_host ring*

The following is sample output for a Single Data Center scenario with three Cassandra nodes:

```
/genesys/apache-cassandra-1.2/bin$ ./nodetool ring
Address      DC           Rack  Status  State  Load      Owns      Token
192.0.2.10   datacenter1 rack1  Up      Normal 14.97 MB  100.00%  -9223372036854775808
192.0.2.11   datacenter1 rack1  Up      Normal 14.97 MB  100.00%  -3074457345618258603
192.0.2.12   datacenter1 rack1  Up      Normal 14.97 MB  100.00%  3074457345618258602
```

The following is sample output for a Development scenario with a single Cassandra node:

```
/genesys/apache-cassandra-2.2/bin$ ./nodetool ring
Address      DC           Rack  Status  State  Load      Effective-
Ownership Token
127.0.0.1    datacenter1 rack1  Up      Normal 1.89 MB
100.00%     76880863635469966884037445232169973201
```

**End**

## Next step

- [Installing Web Services and Applications](#)