Oracle® Communications Messaging Server

Installation and Configuration Guide for Cassandra Message Store

8.1

F15147-02

January 2025

ORACLE®

Oracle Communications Messaging Server Installation and Configuration Guide for Cassandra Message Store, 8.1

F15147-02

Contents

1   Cassandra Message Store Installation and Configuration Overview

2   Planning Your Cassandra Message Store Installation

3   Developing a Cassandra Message Store Architecture

4 Cassandra Message Store System Requirements

5 Cassandra Message Store Pre-Installation Tasks

6 Installing Cassandra Message Store

7   Cassandra Message Store Post-Installation Tasks

8   Migrating to Cassandra Message Store

Preface

This guide provides instructions for installing and configuring Oracle Communications Messaging Server Cassandra message store. Where appropriate, this guide refers to *Messaging Server Installation and Configuration Guide* for concepts and tasks pertaining to the overall installation of Messaging Server software.

Audience

This document is intended for system administrators or software technicians who install and configure Messaging Server. This guide assumes you are familiar with the following topics:

- Messaging protocols, such as IMAP and SMTP
- Oracle Directory Server Enterprise Edition and LDAP
- System administration and networking
- Cassandra

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

Cassandra Message Store Installation and Configuration Overview

This chapter provides an overview of the Oracle Communications Messaging Server Cassandra message store installation and configuration process.

Overview of Cassandra Message Store Installed Software

During the Cassandra message store installation process, you install and configure the following software:

- On Cassandra nodes:
  - Oracle Java SE Runtime Environment (JRE) or OpenJDK

    If you are using SSL and need to use the Java **keytool** utility, you must install Java SE Development Kit (JDK).

  - Python

  - Apache Cassandra

- On Messaging Server message access tier, MTA, and MMP hosts:
  - Messaging Server software

  - Optional: Java is required on message access tier hosts if they also run the Indexed Search Convertor (ISC).

See *Messaging Server Installation and Configuration Guide* for information about additional Messaging Server components that you must install and configure, and other software dependencies.

> **Note:**
>
> You cannot run both Cassandra message store and classic message store in the same Messaging Server deployment. That is, you cannot mix Cassandra and classic message stores in the same deployment.

Overview of the Message Store Installation Procedure

The installation procedure follows these steps:

1. Plan your installation. When planning your installation, do the following:

   - Determine the scale of your implementation, for example, a small development system, or a large production system.

   - Determine how many physical machines you need, and which software components to install on each machine.

- Plan the system topology, for example, how the system components connect to each other over the network.

2. Review system requirements. System requirements include:

   - Hardware requirements, such as disk space.

   - System software requirements, such as operating system (OS) versions and OS patch requirements.

   - Information requirements, such as IP addresses and host names.

3. Install and configure Apache Cassandra.

4. Install and configure Messaging Server and configure the new message store.

5. Perform post-installation configuration tasks.

6. Verify the installation.

7. Continue with the rest of your Messaging Server installation.

Messaging Server Installation Options

You install Messaging Server software by running the installer in either interactive or silent mode. When you run the installer in silent mode, you are running a non-interactive session. The installation inputs are taken from the following sources:

- A silent installation file

- Command-line arguments

- Default settings

You can use silent mode to install multiple instances of the same software component and configuration without having to manually run an interactive installation for each instance.

For more information, see the discussion on running a non-interactive session in *Messaging Server Installation and Configuration Guide*.

Directory Placeholders Used in This Guide

Table 1-1 lists the placeholders that are used in this guide:

Table 1-1   Messaging Server Directory Placeholders

Placeholder	Directory
MessagingServer_home	Specifies the installation location for the Messaging Server software. The default is **/opt/sun/comms/messaging64**.
InstallRoot	Specifies the installation location for other Communications Suite software. The default is **/opt/sun/comms**.
ConfigRoot	Specifies the location of the configuration files. The default is **/var/opt/sun/comms/messaging64/config**.

2

Planning Your Cassandra Message Store Installation

This chapter describes how to plan for installing the Oracle Communications Messaging Server Cassandra message store.

About Cassandra

Apache Cassandra is a free and open-source NoSQL database designed for:

- Large amounts of data
- Use on commodity Linux servers
- Continuous read/write availability
- High availability with no single point of failure
- Multiple data centers to ensure constant uptime

For more information, see the Apache Cassandra documentation.

About the Cassandra Message Store

In Messaging Server, the Cassandra message store consists of *keyspaces*. A keyspace, in a NoSQL data store, is a namespace that defines data replication on nodes. (It resembles the schema concept in Relational database management systems.) The Messaging Server Cassandra message store consists of the following keyspaces:

1. **ms_msg**: Contains the email message content. It can be very large.

2. **ms_mbox**: Contains user and mailbox metadata. It is relatively small and has a lot of mutations.

3. **ms_cache**: Contains cache tables for converted message content.

After Messaging Server has been installed and configured, the Cassandra message store keyspaces are created on one or more data centers once the Messaging Server stored processes start for the first time. When deploying the Cassandra message store, the recommendation is to use multiple Cassandra clusters. See "Understanding the Cassandra Message Store Architecture" for more information.

About Elasticsearch Indexing and Search

Apache Cassandra is the database platform hosting the Cassandra message store and Elasticsearch is the indexing and search engine.

The following component is Oracle extensions to Elasticsearch , which is open source software. This extension is packaged and with Messaging Server:

- Indexed Search Converter (ISC): Deployed as a standalone web application, the ISC processes the entire message body—headers, attachments, and plain text body—and generates plain text content for each of the message parts. It stores this converted content

in a persistent Cassandra cache table, which is later indexed in Elasticsearch cluster. The cache enables the ISC to avoid processing the same contents and attachments multiple times (for example, when there are multiple recipients, email forwards, and so on).

You must deploy the ISC on the message access tier hosts.

Email messages have two paths to the Cassandra message store:

1. Messages that arrive through LMTP are processed by ISC in the message access layer. The ISC pre-converts the header, body and attachments to plain text and stores them in cache. This is also referred to as "pre-conversion". This data is then indexed in Elasticsearch cluster.

2. Messages that arrive in the store through the IMAP **APPEND** command or the **imsrestore** command are to be processed by ISC on the message access tier host(s). In this scenario, the indexing is triggered twice:

   a. In the first trigger, all email fields, except the body and header, are immediately indexed. The body conversion and indexing happen asynchronously.

   b. The second trigger happens after the message is converted and cached.

Planning Your Cassandra Message Store Installation

See the topic on planning your Messaging Server installation in *Messaging Server Installation and Configuration Guide* for general planning information. You must consider the following items before installing Cassandra message store:

- Cassandra software: You must have Apache Cassandra as the database for the Cassandra message store.

- Operating system: You must install Cassandra nodes on Linux. Solaris OS is not supported for Cassandra nodes.

- Messaging Server version: Cassandra nodes and message access tier hosts must run Messaging Server 8.1. Front-end hosts must run at least Messaging Server 8.0.1. Additionally, you must use Messaging Server Unified Configuration to administer message access tier hosts. Legacy configuration is not supported on these hosts.

- Store protocol: You must use Local Mail Transfer Protocol (LMTP) on the message access layer to deliver messages into the Cassandra message store.

- Notifications: You must use Event Notification Service (ENS) for notifications on the message access tier hosts. (Support for Java Message Queue (JMQ) is deprecated.)

- Store delivery: You must configure store affinity groups by using the **proxy:***mailhost***.storehostlist** Unified Configuration option for robust Cassandra message store delivery.

- Message store storage: Use of solid-state drives (SSDs) is recommended for nodes running Cassandra. For actual storage requirements, see "Hardware Requirements".

System Deployment Planning

This section contains system-level planning topics you must consider before installing Cassandra message store:

Planning for High Availability

Cassandra database is designed to be highly available with no single point of failure. Thus, you do not need to install additional cluster software to create a high availability solution for the message store itself.

Planning for Disaster Recovery

When planning for disaster recovery, consider deploying Cassandra across multiple data centers in different geographic locations. Cassandra enables replicating and synchronizing data across multiple data centers. You can use different data centers as a live backup, or as a failover site.

Using Load Balancing

Load balancers balance network connections uniformly or by algorithm across multiple servers. You cannot use load balancers on the message store or directory masters. You can use them for connections to MMPs, Oracle Convergence, MTAs, directory consumers, and with the MTA's use of anti-virus and anti-spam products.

Planning Backup Strategies

Cassandra uses data replication to stores copies (replicas) of data on multiple nodes to ensure that data is not lost if the system crashes, hardware fails, and so on. Nevertheless, backing up and restoring data is an important administrative task for your Messaging Server deployment. To back up the Cassandra database, use the snapshot feature, which can back up all keyspaces, a single keyspace, or an entire cluster. For more information, see the Cassandra backup and restore documentation.

About Installing a Secure System

You can secure your Messaging Server infrastructure by first determining your firewall or DMZ architecture. You can also configure Secure Sockets Layer (SSL) on Messaging Server front-end hosts. You must also protect Messaging Server individual components.

For Cassandra message store, keep the following security items in mind:

- Messaging Server does not use SSL or authentication for the Cassandra Query Language (CQL) connections from the message access tier (the IMAP/LMTP-server/store-CLI-tools) to Cassandra. Instead, keeping those hosts together on a physically secure network that blocks external access to CQL is recommended.

- To secure data at rest on Cassandra nodes, refer to Apache Cassandra for more information.

- You can optionally secure the connection from the LMTP server to the Indexing Search Convertor (ISC) with SSL.

- ENS can be secured by using SSL and authentication support that was introduced in Messaging Server 8.0.1.

For more information, see the discussion on secure installation and configuration of Messaging Server in the *Messaging Server Security Guide*.

3

Developing a Cassandra Message Store Architecture

This chapter provides information about how to design your Oracle Communications Messaging Server Cassandra message store architecture, and information about how Messaging Server components are distributed across hardware and software resources. For more information about general concepts, see the "Developing a Messaging Server Architecture" chapter in *Messaging Server Installation and Configuration Guide*.

Understanding the Cassandra Message Store Architecture

In general, you never deploy the Cassandra message store on a single node. Instead, you deploy the Cassandra message store on multiple nodes in a *data center*. In Cassandra, the term *data center* is a grouping of nodes, configured together for replication purposes. (A *cluster* contains one or more data centers, which can span physical locations.)

A Cassandra node can be installed on a Docker container, a virtual machine (VM), or a physical host. Using containers provides more flexibility with your deployment than using physical hosts. For example, if a physical host contains extra bandwidth, you can increase throughput by installing more Cassandra instances within containers.

In a Messaging Server deployment, containers are recommended for installing Cassandra. However, if a physical host can support only a single Cassandra instance, do not use a container.

Each node can read and write messages that are sent to the data center. One node is chosen to be *coordinator* for each read or write operation, and forwards the read or write operation to appropriate *replicas* for the operation's partition key. Cassandra uses token aware routing, so the coordinator for an operation usually holds one copy of the relevant data.

Cassandra uses replicas on multiple nodes to ensure data reliability and fault tolerance. A *replication factor* (on a per-keyspace basis) determines the number of replicas. A replication factor of 1 means that there is only one copy of each row on one node. A replication factor of 2 means two copies of each row, where each copy is on a different node. When you install the Cassandra message store, Messaging Server automatically sets the replication factor on the Cassandra keyspaces. You can control the replication factor through Messaging Server configuration options. For more information, see the replication factor options in Table 4-3.

Internode communications between nodes occurs through a peer-to-peer protocol called *gossip*. Through gossip, nodes periodically exchange state information about themselves and about other nodes they know about. To facilitate gossip communications and bootstrap nodes in a data center, you designate some nodes as *seed nodes*. The seed node acts as a point of contact for other nodes to find each other and learn the topology of the data center. In addition, it is recommended to configure *snitches*, which determine which data centers and racks the nodes belong to. It is recommended to configure the Cassandra option **GossipingPropertyFileSnitch** as **endpoint_snitch**. Snitches inform Cassandra about the network topology so that requests are routed efficiently and enable Cassandra to distribute replicas by grouping machines into data centers and racks. You also use Message Server data center and replication factor options, to configure racks and data centers. For more information, see Table 4-3.

For more information about Cassandra architecture, see Cassandra documentation.

Overview of Cassandra Message Store Logical Architecture

In general terms, the message store holds and stores user mail. Sometimes it is referred to as a "back end." The message store also refers to the Message Access Components including the IMAP server and the POP server. Starting with Messaging Server 8.0.2, the Message Access Components have been separated out from the message store into their own logical layer, referred to as the message access layer.

The original Messaging Server message store is based on storing mailbox caches, mailbox indexes and messages in the file system, and use of Oracle Berkeley Database (BDB) for other metadata. The Cassandra message store, based on Cassandra database, changes the approach to how you architect the store components of your Messaging Server deployment. At a high-level, Figure 3-1 shows the Cassandra message store logical architecture.

Figure 3-1   Cassandra Message Store Logical Architecture
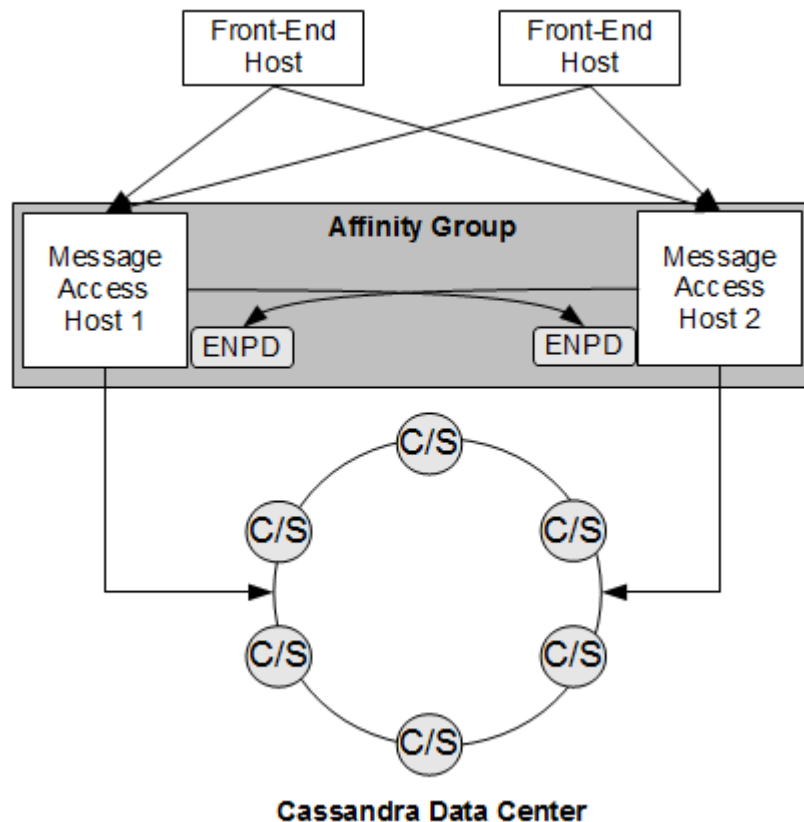
The preceding figure shows the following components of the Cassandra message store logical architecture:

*   Front-end hosts: The front-end hosts, which can be Message Transfer Agent (MTA), Messaging Multiplexor (MMP), LMTP client, or Webmail (**mshttpd**) hosts, communicate with the message access hosts.

*   Message access hosts: The message access hosts run the **imapd** process and LMTP server. The message access hosts also run the Event Notification Service (ENS) **enpd** process for horizontal scalability in the form of a store affinity group. Store affinity groups

enable routing to an alternate LMTP server in the list if the first one listed is unavailable. Store notifications relevant to the store affinity group are published to all ENPD servers in the store affinity group. Other connections to a store affinity group (POP, IMAP, LMTP server) can be handled by any host in the store affinity group.

> **✎ Note:**
>
> Only two message access hosts are required in a store affinity group. This differs from classic store automatic failover, which requires three hosts in a store affinity group.

The message access hosts also run the Indexed Search Converter (ISC) process, which pre-converts messages before delivery to the message store.

- Cassandra data center: The data center is a collection of Cassandra hosts, which store the email data.
- Elasticsearch cluster: The cluster is a collection of one or more nodes to index the email data.

Deploying the Cassandra Message Store on Data Centers

Figure 3-2 shows the logical architecture of a Cassandra message store on a single data center.

Figure 3-2   Message Store Deployed Upon a Single Cassandra Data Center
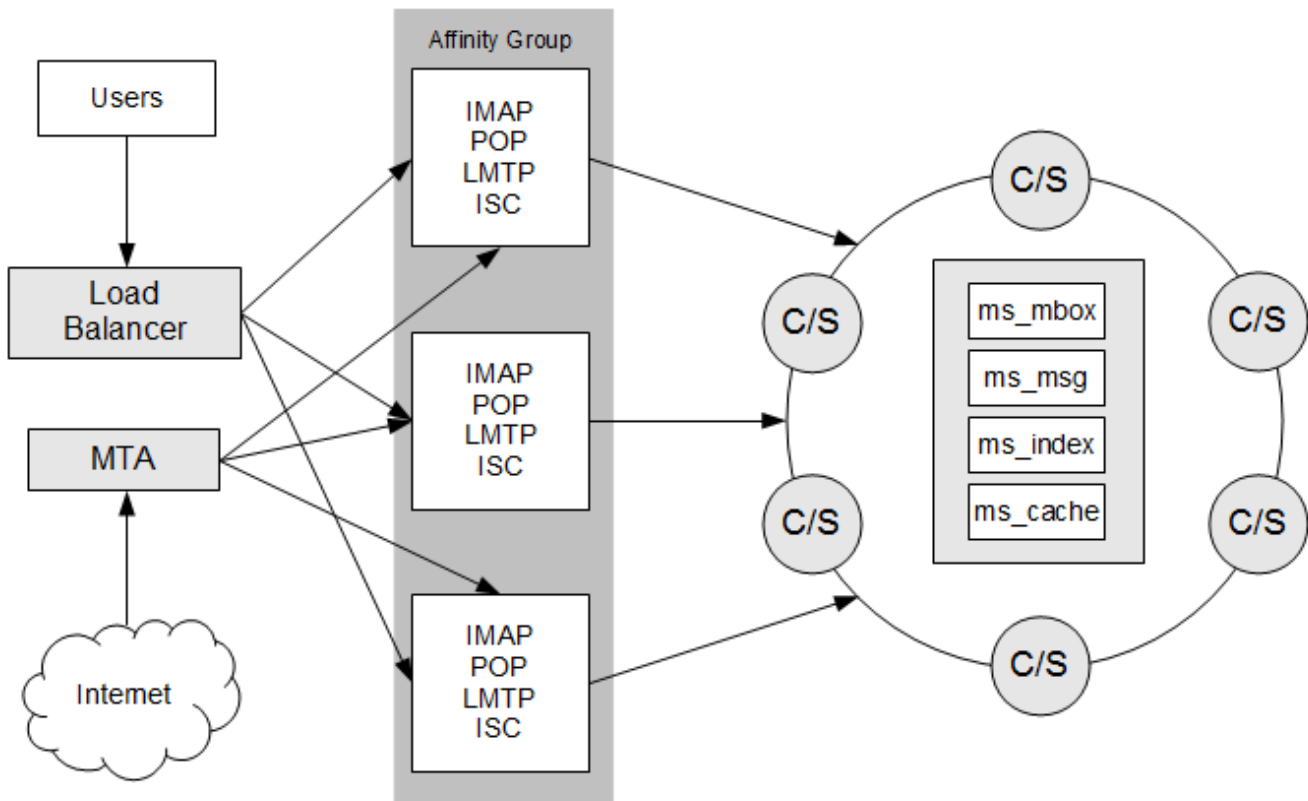
This figure shows a Messaging Server Cassandra message store deployment that consists of a load balancer; a Message Transfer Agent (MTA) host for incoming traffic; multiple Messaging Server message access hosts in a single affinity group; a single Cassandra data center; and Elasticsearch cluster. Each Cassandra node (indicated by a "C" in the figure) in the data center contains the three keyspaces that make up the message store and indexing and search component: **ms_mbox**, **ms_msg**, and **ms_cache**.

> **Note:**
>
> See Figure 3-1 for more details on the logical architecture of the message access tier and ENS, which is not depicted in this figure.

When you perform the initial configuration of Messaging Server, or when the service startup of the **stored** process happens when in Cassandra message store mode, by default, it creates all three keyspaces on a Cassandra node on one data center called **DC1**.

In general, use a single data center for a proof of concept deployment, and not for production purposes. While you can use a single store affinity group deployment, the scale limits of a single-affinity deployment are unknown. In the event ENS notifications become a scale bottleneck, the recommendation is to add a new store affinity group. When there is more than one store affinity group, you must have an MMP between the load balancer and the IMAP/LMTP/ISC tier.

Figure 3-3 shows the logical architecture of a Cassandra message store on multiple data centers and clusters.

Figure 3-3   Message Store Deployed Upon Multiple Cassandra Clusters
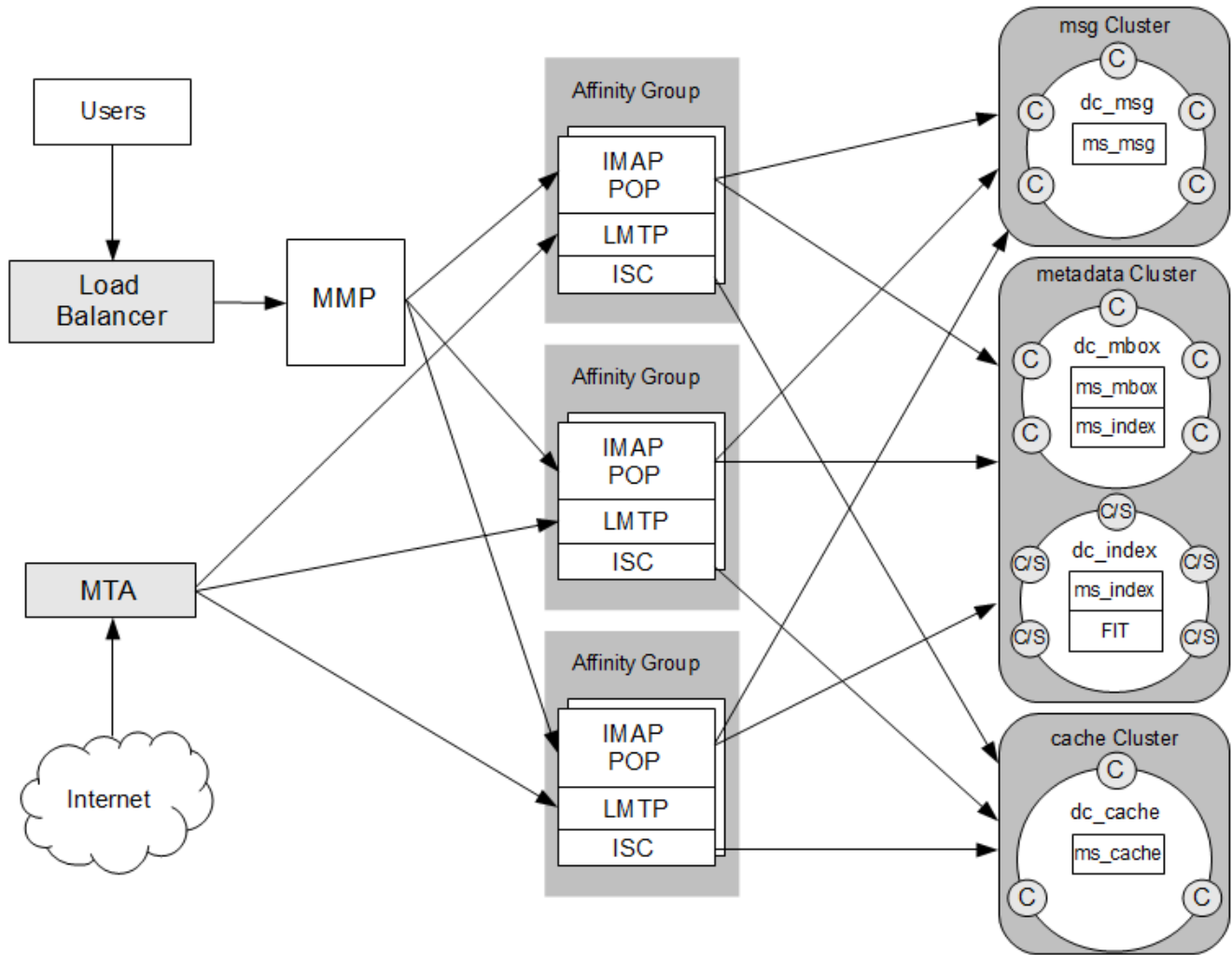
This figure shows a Messaging Server Cassandra message store deployment that consists of a load balancer; a Message Transfer Agent (MTA) host for incoming traffic; a Messaging Multiplexor (MMP) host to act as the connection point to the multiple back-end message stores; multiple Messaging Server front-end hosts, configured as three store affinity groups (and running ENS), for message access and delivery handling; and three Cassandra clusters. The clusters contain at least one Cassandra data center, indicated by "**dc_**_name_." Each Cassandra node is indicated by "C" in the figure. The **dc_msg** data center contains the **ms_msg** keyspace. The **dc_mbox** data center contains the **ms_mbox** keyspace. The **dc_cache** data center contains the **ms_cache** keyspace. Because the cache data can be regenerated, only a single **ms_cache** keyspace is needed; disaster recovery is unnecessary. Also, the **ms_cache** keyspace is only needed by the Indexed Search Converter (ISC). The Cassandra replication factor ensures there are _N_ copies of data distributed across the nodes in the data center.

> **Note:**
>
> See Figure 3-1 for more details on the logical architecture of the message access tier and ENS, which is not depicted in this figure.

Planning for a Highly Available Cassandra Message Store Deployment

Unlike previous versions of Messaging Server, where you need to install and manage clustering software to obtain a highly available message store, the Cassandra message store is itself designed to provide continuous uptime. Cassandra relies the concept of installing multiple nodes with no single point of failure. Because of its peer-to-peer distributed system, where data is handled by all nodes in a cluster, Cassandra is able to address the problem of failures. Thus, the need to use additional clustering software is rendered unnecessary (nor is it supported).

At a high-level, Cassandra operates in the following way to ensure data consistency across all nodes in a cluster:

1. Uses a peer-to-peer protocol called gossip for nodes to exchange state information.

2. Ensures data durability through use of a a sequentially written commit log on each node.

3. Indexes and writes data to an in-memory structure, called a memtable, which is similar to a write-back cache.

4. When the in-memory structure becomes full, Cassandra writes the data to disk. Cassandra automatically partitions and replicates the writes throughout the cluster.

For more information on Cassandra architecture, see the Cassandra documentation at:

http://cassandra.apache.org/doc/latest/architecture/index.html

High availability is achieved on the Messaging Server front-end tier (Message Transfer Agent (MTA), Messaging Multiplexor (MMP), and **mshttpd** daemon) through the use of load balancers, or MX records (MTA-submit). High availability is achieved on the message access tier (IMAP, POP, LMTP server, ENPD, and ISC) with store affinity groups. High availability on the back-end tier (Cassandra and ISC) is achieved through Cassandra itself.

You must configure store affinity groups consistently on all front-end tier and message access tier hosts. Store affinity groups enable the MMP to switch to an alternate IMAP server that can perform read/write IMAP operations for the user if one of the message access tier hosts goes down. Classic message store only supports failover (with read-only on backup IMAP servers), while Cassandra store supports read/write for all IMAP servers in the affinity group.

Store notifications are broadcast to all **enpd** servers in the store affinity group (which then relays them to all subscribed IMAP servers in the affinity group). Without store affinity groups, all store notifications would have to be broadcast to all *enpd* servers in the deployment, which would eventually pose a scalability problem.

For more information on configuring store affinity groups, see "Installing and Configuring the Messaging Server Software".

4

Cassandra Message Store System Requirements

This chapter describes the software, hardware, and operating system requirements for installing Oracle Communications Messaging Server Cassandra message store hosts. See *Messaging Server Installation and Configuration Guide* for the software, hardware, and operating system requirements for Messaging Server front-end and message access tier hosts.

Software Requirements

This section describes the software requirements for installing Cassandra message store.

Supported Operating Systems

Table 4-1 lists operating systems that support Cassandra message store. We recommend that you run the latest software update with the latest recommended patch set.

Table 4-1   Supported Operating Systems

Operating System	CPU	Required Patches
Oracle Linux 7.2 Red Hat Linux 7.2	x64	See the Oracle Linux and Red Hat Enterprise Linux documentation for patch information.

Required Software

Table 4-2 lists software required for installing and running Cassandra message store. See *Messaging Server Installation and Configuration Guide* for other requirements related to Messaging Server itself.

Table 4-2   Software Requirements

Product	Version	Notes
Oracle Java SE Runtime Environment 8 (JRE) or OpenJDK 8	1.8.0_40 minimum	Required on message access tier hosts if they also run the Indexed Search Convertor (ISC). If you are using SSL and need to use the Java **keytool** utility, you must install Java SE Development Kit (JDK).
Python	2.7.x	Required to install Cassandra hosts.
Apache Cassandra	3.11	You can download it from Cassandra website.

Hardware Requirements

Cassandra node storage requirement is:

- Message data center nodes: 5 TB when using SSD (utilization should be less than 70 percent)

Additionally, see the table on hardware requirements for installing Cassandra in the Cassandra documentation.

The number and configuration of the systems that you employ for your Messaging Server installation depends on the scale and the type of deployment you have planned.

Cassandra nodes that are configured with the **ms_msg** keyspace require the most disk capacity to store the message content. Cassandra nodes configured with the **ms_cache** keyspace collectively should have enough disk capacity for 14 days of converted data stored in the conversion cache.

> **Note:**
>
> The sizing estimates in this section assume proper application configuration and tuning, in a manner consistent with leading practices of Oracle Communications consulting and performance engineering. This information is provided for informational purposes only and is not intended to be, nor shall it be construed as a commitment to deliver Oracle programs or services. This document shall not form the basis for any type of binding representation by Oracle and shall not be construed as containing express or implied warranties of any kind. You understand that information contained in this document will not be a part of any agreement for Oracle programs and services. Business parameters and operating environments vary substantially from customer to customer and as such not all factors, which may impact sizing, have been accounted for in this documentation.

Time Synchronization Requirements

Configure Network Time Protocol (NTP) to ensure that time is synchronized across your deployment.

Cassandra nodes must synchronize their clocks between all hosts accessing Cassandra. All the Cassandra hosts and all the IMAP/LMTP-server tier hosts must have fully synchronized clocks or the deployment becomes unsupportable (both Cassandra-level errors and application errors related to IMAP QRESYNC and CONDSTORE [RFC 7162] will result).

Information Requirements

This section describes the information needed before installing and configuring Cassandra message store.

Connection Information

You need the IP addresses of the Cassandra nodes that act as **cassconnectpoints** when installing and configuring the message access tier nodes.

Data Center and Replication Factor Requirements

Table 4-3 shows the Messaging Server options you use to configure racks and data centers when installing and configuring message access tier hosts.

Table 4-3   Data Center and Replication Configuration Options

Keyspace	Data Center Option	Replication Factor Option
ms_cache	store.cascachedc	store.cascacherf
ms_mbox	store.casmetadc	store.casmetarf
ms_msg	store.casmsgdc	store.casmsgrf

Store Affinity Group Requirements

You must understand the store affinity group layout, and how each LDAP **mailHost** value translates to two or more access tier hosts. For more information, see the topic on the **storehostlist** option in *Messaging Server Reference*.

ORACLE®

5

Cassandra Message Store Pre-Installation Tasks

This chapter provides information on the pre-installation tasks you must complete on Cassandra nodes before you can install Messaging Server software.

Summary of General Pre-Installation Tasks

The following list summarizes the general pre-installation tasks you must complete before installing any Messaging Server component.

- Create a UNIX system user and group for Messaging Server, and set permissions for the directories and files owned by that user.
- Check that DNS is running and configured properly for the Messaging Server host.
- Check the number of file descriptors for Linux, and if this number is less than 16384, you need to increase the value.
- Install Oracle Directory Server Enterprise Edition, if your site does not currently have Directory Server deployed.

See the chapter titled "Messaging Server Pre-Installation Tasks" in *Messaging Server Installation and Configuration Guide* for detailed information.

The following list summarizes the pre-installation tasks you must complete on Cassandra nodes:

- Installing Java
- Installing Python
- Installing Apache Cassandra

Installing Java

To install Java, see "Prerequisites" on the Cassandra web site at:

`http://cassandra.apache.org/doc/latest/getting_started/installing.html`

> **✐ Note:**
>
> The **JAVA_HOME/bin** directory must be in the **PATH** environment variable.

Installing Python

To install Python, see the Python documentation at:

`https://docs.python.org/2/installing/`

Be sure to use the version of Python that is supported by the version of Cassandra that you are installing.

Installing Apache Cassandra

The tasks to install Apache Cassandra are:

- Downloading the Apache Cassandra Software
- Installing the Apache Cassandra Software
- Installing Elasticsearch Cluster
- Setting Up the Cassandra Cluster
- Changing Initial Cassandra Settings
- Changing Initial Tuning Settings

Downloading the Apache Cassandra Software

To download the Cassandra software:

1. Download the Cassandra software from the Cassandra download site, located at:

   `http://cassandra.apache.org/download/`

2. Copy the installer file to your Cassandra message store hosts.

Installing the Apache Cassandra Software

To install Cassandra software:

1. On each Cassandra node, install the Cassandra software, and verify that Cassandra is running.

   For more information, see the Cassandra installation documentation at:

   `http://cassandra.apache.org/doc/latest/getting_started/installing.html#`

2. Ensure that for Oracle Linux 6.x and later, the 32-bit versions of the **glibc** libraries are installed.

   For more information, see the Cassandra documentation at:

   `http://cassandra.apache.org/doc/latest/getting_started/installing.html#`

3. Optionally, install **msstatbot**, a monitoring solution for Cassandra. For more information, see the *Messaging Server System Administrator's Guide*.

Installing Elasticsearch Cluster

To install Elasticsearch cluster, see the documentation at: `https://www.elastic.co/guide/en/elasticsearch/reference/current/install-elasticsearch.html`

Setting Up the Cassandra Cluster

To set up the Cassandra cluster, see the Cassandra documentation:

- For a single data center, see:

  `http://cassandra.apache.org/doc/latest/configuration/index.html`

- For multiple data centers, see:

  http://cassandra.apache.org/doc/latest/configuration/index.html

When setting up multiple data centers, the Messaging Server recommendation, which minimizes the overhead in replicating and repairing Cassandra keyspaces across all data centers, is to configure four data centers in three clusters with keyspaces arranged as shown in Table 5-1.

Table 5-1   Recommended Multiple Data Centers and Clusters Configuration

Data Center Name and Node Types	Keyspaces	Cluster Configuration
DC_MSG, Cassandra nodes	**ms_msg**	**Cluster Content**
DC_META, Cassandra nodes	**ms_mbox**, **ms_index**	Combined with **DC_INDEX** into **Cluster Metadata**
DC_CACHE, Cassandra nodes	**ms_cache**	**Cluster Cache**

Cluster settings, such as the cluster name and seed nodes, are defined in the **cassandra.yaml** file. See the following section for more information.

To support more concurrent index updates, the ratio of **DC_META** nodes to **DC_INDEX** nodes should be at least 1 to 2.

Changing Initial Cassandra Settings

On each Cassandra node, optimize the Cassandra installation by following the recommendations in the Cassandra documentation.

Changing Initial Tuning Settings

On each Cassandra node, change the configuration files described in this section so that the node operates correctly in the Cassandra message store deployment.

cassandra.yaml File

Make the changes in this section to the **/etc/cassandra/cassandra.yaml** file.

For all nodes, to enable separate clusters for better performance, specify **cluster_name**.

Make the following changes to the **num_tokens** setting:

- **DC_MSG**, **DC_META**, and **DC_CACHE** nodes:

  num_tokens: 256

To improve performance, locate data on SSD drives:

- **data_file_directories**:

  /var/lib/cassandra/data

- **commitlog_directory**:

  /var/lib/cassandra/commitlog

- **saved_caches_directory**:

  /var/lib/cassandra/saved_caches

- **hints_directory**:

```
/var/lib/cassandra/hints
```

To support large mailbox and message, increase the **commitlog** size:

```
commitlog_segment_size_in_mb: 256
```

To specify seed nodes, you must use two nodes from each data center in the cluster, preferably located on different racks, so that each cluster has different seeds, for example:

- **DC_MSG** cluster:

  ```
  seeds: "192.0.2.12,192.0.2.24"
  ```

- **DC_META** cluster:

  ```
  seeds: "192.0.2.1,192.0.2.2,192.0.2.10,192.0.2.3"
  ```

- **DC_CACHE** cluster:

  ```
  seeds: "192.0.2.14,192.0.2.7"
  ```

For all nodes, make the following change to improve performance:

```
memtable_flush_writers: 8
```

For all nodes, specify **listen_address**, **rpc_address**, **native_transport_address**, and so on, according to your deployment.

cassandra-env.sh File

For all nodes, to specify the location of the heap dump, make the following change to the **/etc/cassandra/cassandra-env.sh** file:

```
export CASSANDRA_HEAPDUMP_DIR=/scratch/heapdump
```

jvm.options File

For **DC_MSGDC_META**, and **DC_CACHE** nodes, make the following heap size changes:

```
-Xms32G
-Xmx32G
```

For **DC_META** nodes, to improve performance, make the following heap size changes to improve performance:

```
-Xms16G
-Xmx16G
```

For all nodes, make the following changes to improve performance:

```
-XX:InitiatingHeapOccupancyPercent=70
-XX:ParallelGCThreads=12
-XX:ConcGCThreads=12
```

For all nodes, print garbage collection measurements, which are useful for monitoring system performance:

```
-XX:+PrintGCDetails
-XX:+PrintGCDateStamps
-XX:+PrintHeapAtGC
-XX:+PrintTenuringDistribution
-XX:+PrintGCApplicationStoppedTime
-XX:+PrintPromotionFailure
-XX:PrintFLSStatistics=1
```

ORACLE

```
-Xloggc:/var/log/cassandra/gc.log
-XX:+UseGCLogFileRotation
-XX:NumberOfGCLogFiles=10
-XX:GCLogFileSize=10M
```

cassandra-rackdc.properties File

For all nodes, make the following changes to the **cassandra-rackdc.properties** file.

1. Configure the endpoint snitch:

   ```
   endpoint_snitch: GossipingPropertyFileSnitch
   ```

2. Set the data center and rack names as appropriate:

   ```
   dc=mydc
   rack=myrac
   ```

   For example, for a node in **DC_CACHE** in a physical rack in one location, set
   dc=DC_CACHE and **rack=RAC1**. And, for another node in **DC_CACHE** in a physical rack
   in another location, set **dc=DC_CACHE** and **rack=RAC2**.

   > **Note:**
   >
   > Data center and rack names are case sensitive.

6

Installing Cassandra Message Store

This chapter describes how to install and configure Oracle Communications Messaging Server software on Cassandra message store nodes, message access tier hosts, and front-end tier hosts. For more information on the Messaging Server install command, see *Messaging Server Installation and Configuration Guide*.

About Messaging Server Components

When you install the Cassandra message store, you install and configure one or more of the following components:

- **Cassandra Node.** Consists of the Cassandra database. The Cassandra keyspaces and schema are created whenever the Messaging Server **stored** process starts on any message access node that points to this Cassandra data center.

- **Message Access Tier.** Consists of a set of components that store, retrieve, and manipulate messages for mail clients. When using Cassandra message store, you configure at least IMAP and the LMTP server components on this tier. The message access tier also must be affinity-group aware for notification publishing.

- **Front-End Tier.** Consists of hosts that connect to the message access tier affinity group based on a user's LDAP **mailHost** attribute. For Cassandra message store, this includes the Message Transfer Agent (MTA), Messaging Multiplexor (MMP), LMTP Client, and **mshttpd** daemon.

> ✎ **Note:**
>
> Messaging Server front-end tier components—Message Transfer Agent and Messaging Multiplexor (MMP) hosts, LMTP clients, and Webmail Server (**mshttpd**) hosts—must be running at least Messaging Server 8.0.1.

The Cassandra message store changes the semantics of the **mailHost** attribute in LDAP. Instead of referring to a physical host, in Messaging Server 8.0.2 "mail host" refers to a *store affinity group*. You configure the store affinity group by using the **proxy:***mailhost***.storehostlist msconfig** option. You must configure store affinity groups with the same **storehostlist** on all front-end and access-layer hosts in the deployment. Store affinity groups enable the MTA to route to an alternate LMTP server in the list if the first one listed is unavailable. Affinity groups are used to distribute ENS delivery events so that IMAP IDLE works without the need to broadcast all delivery events to all IMAP servers.

> ✎ **Note:**
>
> The store affinity group concept is also used by the Messaging Server 8.0.1 automatic failover (without cluster) feature. That is why it is supported for Messaging Server 8.0.1 front-end hosts to communicate with Cassandra message access-tier hosts.

About Unified Configuration

You must use Unified Configuration to configure and administer Messaging Server on the message access tier hosts.

Unified Configuration is an improved, streamlined process to configure and administer Messaging Server. Unlike in legacy configurations, Unified Configuration uses validation to verify configuration accuracy, and employs a single tool to configure the entire Messaging Server configuration (with a few exceptions). For more information, see the discussion on Unified Configuration in *Messaging Server System Administrator's Guide*.

Before Installing Messaging Server

You must have already completed the following steps on all Messaging Server components:

1. Ensure that DNS is running and configured properly.

2. Ensure you have sufficient file descriptors on Linux.

3. Make sure you do not configure conflicting port numbers on a host when various components run on a single machine.

4. Preparing Directory Server by running the **comm_dssetup.pl** script against it.

For more information on completing these tasks, see "Messaging Server Pre-Installation Tasks" in *Messaging Server Installation and Configuration Guide*.

The following steps must be completed for Cassandra nodes:

1. Install Java.

   You also need to install Java on the message access tier hosts if they run the Indexed Search Convertor (ISC), in addition to the normal processes (**imap**, **enpd**, and **lmtps**).

2. Install Python.

3. Install Apache Cassandra software.

4. Start Apache Cassandra as a transactional (Cassandra) node.

   See Cassandra Message Store Pre-Installation Tasks for more information.

Using a Configuration Management Tool to Simplify Installation and Configuration

Currently, Messaging Server does not supply its own tool for configuration management. However, you can use a configuration management tool, such as Chef or Puppet, to simplify the installation and management of Messaging Server across multiple servers. For more information, see the topic on using a configuration management tool with Messaging Server in *Messaging Server Installation and Configuration Guide*.

Installing and Configuring Messaging Server

The tasks to install the front-end and message access tiers, and the Cassandra message store are as follows:

- Downloading the Messaging Server Software

- [Installing and Configuring the Messaging Server Software](#)

Downloading the Messaging Server Software

1. Download the media pack for Oracle Communications Messaging Server software and for Messaging-CAS software from the Oracle software delivery website, located at:

   http://edelivery.oracle.com/

2. Copy the Messaging Server ZIP file to a temporary directory on your Messaging Server hosts and extract the files.

Installing and Configuring the Messaging Server Software

You must install and configure the Cassandra message store, and associated components, in the following order, to ensure proper operation:

- [Installing Message Access Tier Hosts](#)
- [Installing MTA Tier Hosts](#)
- [Installing MMP Hosts](#)

> **Note:**
>
> You must use ENS for notifications with the Cassandra message store. The default when you run the **init-config** command is for ENS to be enabled, so no additional steps are necessary to enable ENS. However, manual store affinity group configuration is required to have ENS tolerate host failures.

> **Caution:**
>
> When completing the steps in this section, do not start Messaging Server until instructed to do so. Otherwise, errors might result.

Installing Message Access Tier Hosts

The message access tier hosts must run at least Messaging Server 8.0.2 and be configured for Unified Configuration.

To install and configure Messaging Server software on the message access tier hosts:

1. Log in as or become the superuser (**root**).

2. Use the **rpm -i** command to install Messaging Server software.

   See *Messaging Server Installation and Configuration Guide* for more information.

3. Create the Cassandra configuration:

   ```
   cd MessagingServer_home/bin
   init-config
   ```

> **Note:**
>
> By default, the **init-config** command also performs the necessary configuration to set up LMTP pre-conversion of messages. Round robin load balancing does not exist for hosts specified as ISC servers to perform LMTP pre-conversion. Instead, the first available server in list is used.

4. Configure required Cassandra message store options by using the **CasAccessTier.rcp** recipe file.

   This recipe enables the message store, sets the message store type to Cassandra, sets replication factors and data centers, sets connection points, sets up the LMTP server, sets up store affinity groups, and disables unwanted Messaging Server processes.

   a. Change to the *MessagingServer_home***/lib/recipes** directory.

   b. Edit the **CasAccessTier.rcp** recipe for your deployment to configure the access tier hosts. Relevant options that the **CasAccessTier.rcp** file sets are:

   ```
   store.enable = 1
   store.purge.enable = 1
   store.serviceadmingroupdn = Service_Administrators_Group_DN
   store.dbtype = cassandra
   store.casconnectpoints = IP_address_CSnode1 IP_address_CSnode2 ...
   store.casmetarf = 3
   store.casmsgrf = 3
   store.cascacherf = 1
   store.cascachedc = DC3
   store.casmetadc = DC1
   store.casmetadc = Datacenter Name
   store.cascachedc = Datacenter Name
   store.casmsgdc = Datacenter Name
   ```

   > **Note:**
   >
   > The **store.enable**, **store.purge.enable**, and **store.serviceadmingroupdn** options should be set automatically by the **init-config** command.

   c. Run your edited recipe file:

   ```
   msconfig run CasAccessTier.rcp
   ```

5. If Cassandra cluster authentication is required, use the **msconfig** command to set the **store.casusername** and **store.caspassword** options.

6. (Optional) Configure how the ISC is invoked.

   By default, the ISC is configured with the **iscimmediate** option, for LMTP pre-conversion. See *Messaging Server Reference* for more information on the **iscdeferred** option to delay ISC operations until after delivery to the message store.

Installing MTA Tier Hosts

The MTA tier hosts must run Messaging Server 8.0.1 or greater and be configured for Unified Configuration (for store affinity group support).

To install and configure Messaging Server software on the MTA tier (Submission/Relay) hosts:

1. Log in as or become the superuser (**root**).

2. Use the **rpm -i** command to install Messaging Server software.

   See *Messaging Server Installation and Configuration Guide* for more information about the **rpm -i** command.

3. Create the configuration:

   ```
   cd MessagingServer_home/bin
   init-config
   ```

   Running the **init-config** command automatically sets up the LMTP channel correctly on the MTA tier hosts.

4. Configure the MTA by copying the **CasFrontEnd.rcp** recipe from the *MessagingServer_home*/**lib/recipes** directory to the *ConfigRoot*/**recipes** directory, then editing the file.

   When configuring an MTA host, make sure to:

   - Disable the message store

   - Disable the LMTP server

   - Disable **mshttpd**

   For the store affinity group:

   - Specify a list of fully qualified server names that have access to the same message store data. The first host name in the list should be the default master for the message store while the other hosts in the list should be configured to act as failover hosts for the master. When a **mailHost** attribute is found in LDAP, it is first resolved through this configuration. The store affinity group settings (the **storehostlist** option values) must be the same on all hosts in the deployment It is not permitted for the same server name to appear in more than one **storehostlist**. For the Cassandra message store, you can reorder the **storehostlist** on different servers as long as the content is the same.

   > **Note:**
   >
   > The MTA ignores the **storehostlist** option unless the **affinitylist** option is set.

5. Run your edited recipe file:

   ```
   msconfig run CasFrontEnd.rcp
   ```

Installing MMP Hosts

The MMP tier is required if there is more than one store affinity group. The classic message store and Cassandra message store must be in different store affinity groups if both are present in the deployment for migration purposes.

The MMP tier hosts must run Messaging Server 8.0.1 or greater and be configured for Unified Configuration (for store affinity group support).

To install and configure Messaging Server software on the Cassandra MMP tier hosts:

1. Log in as or become the superuser (**root**).

2. Use the **rpm -i** command to install Messaging Server software.

See *Messaging Server Installation and Configuration Guide* for more information about the **rpm -i** command.

3. Create the configuration:

```
cd MessagingServer_home/bin
init-config
```

Running the **init-config** command automatically sets up the LMTP channel correctly on the MTA tier hosts.

4. Configure the MMP by copying the **CasFrontEnd.rcp** recipe from the *MessagingServer_home***/lib/recipes** directory to the *ConfigRoot***/recipes** directory, then editing the file.

When configuring an MMP host, make sure to:

- Disable the message store

- Disable the LMTP server

- Disable **mshttpd**

- Enable the MMP (**mmp.enable** option)

- Uncomment the **delete_** options

For the store affinity group:

- Specify a list of fully qualified server names that have access to the same message store data. The first host name in the list should be the default master for the message store while the other hosts in the list should be configured to act as failover hosts for the master. When a **mailHost** attribute is found in LDAP, it is first resolved through this configuration. The store affinity group settings (the **storehostlist** option values) must be the same on all hosts in the deployment It is not permitted for the same server name to appear in more than one **storehostlist**. For the Cassandra message store, you can reorder the **storehostlist** on different servers as long as the content is the same.

5. Run your edited recipe file:

```
msconfig run CasFrontEnd.rcp
```

Using a Non-MMP IMAP Proxy

If you want to use a different IMAP proxy, that proxy must support IMAP LOGIN referrals (RFC 2221) and for robustness needs to support a facility to handle the store affinity groups.

Configuring a Remote Indexed Search Converter

To configure a remote Indexed Search Converter (ISC), where the ISC is not running on the same host as LMTP, run the following command:

```
msconfig set isc_client.ischosts ISC_host_name1 ISC_host_name2 ...
```

ISC is used by both cassandra store and classic store with elastic search. ISC requires read access to the message store and read/write access to the conversion cache to perform conversion. When remote ISC is used with classic store, the message store and conversion cache directories must be on a shared filesystem (e.g., NFS) and mounted at the same path on both the message store and ISC systems.

Enabling SSL

You can configure the ISC to communicate using Secure Sockets Layer (SSL). To enable SSL, you set the appropriate ISC configuration options, and generate a certificate to import into the certificate database. For detailed steps, see the topic on configuring the ISC component for SSL in *Messaging Server Security Guide*.

Starting the Cassandra Message Store Across Tiers

After you have installed Messaging Server software and configured the Cassandra message store (across all tiers), start the tiers in the following order:

1. Cassandra nodes: Cassandra should already be started. When starting Messaging Server, only the **isc** process should start.

2. Message access hosts: When starting Messaging Server, the appropriate message access services (**imapd**, **stored**, **isc**, and so on) should start.

3. Front-end hosts: When starting Messaging Server, the appropriate front-end services (MMP, MTA, **mshttp**, and so on) should start.

Restarting a Cassandra Cluster

A restarted Cassandra process on a rebooted Cassandra node takes a long time to join the Cassandra cluster. Therefore, when restarting a cluster, you must follow specific steps to avoid read failures in the logs and Cassandra taking a long time to start on the Cassandra nodes. See the topic on restarting a Cassandra message store cluster in *Messaging Server System Administrator's*.

7

Cassandra Message Store Post-Installation Tasks

This chapter describes the post-installation tasks that you must complete to finish the Cassandra message store installation, and how to add and remove additional Cassandra nodes.

Verifying the Installation

To verify the Cassandra message store installation was successful:

1. Verify the Cassandra core was created by logging in to the Cassandra administration console at:

   ```
   http://host:8983/cassandra
   ```

   If the administration console cannot be reached, check the **system.log** file on the Cassandra host for errors. By default, this file is located at **/var/log/cassandra/system.log**.

2. Verify the Indexed Search Converter (ISC) process launched correctly by accessing the statistics servlet at:

   ```
   http://host:8070/converter/statistics
   ```

   If the ISC cannot be reached, check the ISC log file for errors. By default, this file is located at **/opt/sun/comms/messaging64/log/isc.log.***.

Scaling Your Cassandra Message Store Deployment Horizontally

To scale your Cassandra message store deployment horizontally by adding more nodes to an existing deployment, see *Messaging Server System Administrator's Guide*.

For adding a node, and adding a data center, also see the Cassandra documentation at:

```
http://cassandra.apache.org/doc/latest/operating/topo_changes.html
```

8

Migrating to Cassandra Message Store

This chapter describes how to migrate from a Messaging Server 8.0.1 or prior "classic" message store to Messaging Server 8.0.2 and Cassandra message store. It assumes that you have chosen a target deployment, and have developed an architectural design and deployment plan.

Overview of Migrating to Cassandra Message Store

The high-level steps to migrate to Cassandra Message store include:

1. Upgrading current front-end Message Transfer Agent (MTA) hosts, Messaging Multiplexor (MMP) hosts, and Webmail Server (**mshttpd**) hosts to at least Messaging Server 8.0.1.

   You must upgrade these hosts to at least Messaging Server 8.0.1 to support store affinity groups.

   The Cassandra message store changes the semantics of the **mailHost** attribute in LDAP. Instead of referring to a physical host, in Messaging Server 8.0.2 "mail host" refers to a *store affinity group*. You configure the store affinity group with the **proxy:***maihost***.storehostlist msconfig** option. You must configure store affinity groups on front-end MTAs for robust Cassandra message store delivery. Store affinity groups enable the MTA to route to an alternate LMTP server in the list if the first one listed is unavailable. Affinity groups are used to distribute ENS delivery events so that IMAP IDLE works without the need to broadcast all delivery events to all IMAP servers.

2. If not already using LMTP, enabling LMTP server on the classic message store host. When you enable LMTP server, you must restart the dispatcher on each classic store.

   For more information, see "Installing Message Access Tier Hosts".

3. If not already using LMTP for delivery, switch front-end MTAs to use LMTP for delivery. When you switch front-end MTAs to use LMTP, you must run the **imsimta restart smtp** command on those MTAs whose delivery options you changed.

   For more information, see "Installing MTA Tier Hosts".

4. Installing and configuring the Cassandra message store.

   For more information, see "Cassandra Message Store Pre-Installation Tasks" and "Installing and Configuring the Messaging Server Software".

5. Using the **rehostuser** command to migrate users from classic store to Cassandra store.

   For more information, see the **rehostuser** topic in *Messaging Server System Administrator's Guide*.

   For example, the following **rehostuser** command migrates user **joe** to affinity group **ngstore** using message access server **ma1**.

   ```
   rehostuser -u joe -d ngstore -R ma1
   ```

   The **-d** *target* for the **rehostuser** command becomes the user's **mailHost** attribute. For the end users to have access to their email after migration, that *target* must either match one of the host names in the **storehostlist** option for that affinity group, or must match the

name of the affinity group. Use of the **-R** option is only required if the **-d** *target* is not a resolvable host name.

> **Note:**
>
> The **storehostlist** option for an affinity group must be set to the same value on all front-end MTA, MMP, and **mshttpd** hosts, and all message access-tier hosts. Use fully qualified domain names when referring to hosts, both with the **rehostuser** command and in the **storehostlist** option.