Guide for Setting Up Cassandra Cluster

1/16/2016

Contents

[1. 硬件参数 2](#_Toc480202474)

[2. 收集信息 2](#_Toc480202475)

[3. 开始之前 2](#_Toc480202476)

[3.1. 所需文件 2](#_Toc480202477)

[3.2. 所需工具 2](#_Toc480202478)

[4. 安装前置 3](#_Toc480202479)

[4.1. 更新Hostname 3](#_Toc480202480)

[5. 安装 Cassandra 3](#_Toc480202481)

[5.1. Installing the Oracle Java Virtual Machine 3](#_Toc480202482)

[5.2. Installing Cassandra 4](#_Toc480202483)

[5.3. Troubleshooting and Starting Cassandra 5](#_Toc480202484)

[5.4. Connecting to the Cluster 6](#_Toc480202485)

[6. 配置Cassandra节点 7](#_Toc480202486)

[6.1. 编辑节点配置 7](#_Toc480202487)

[6.2. 检查Cassandra服务状态 7](#_Toc480202488)

[7. 配置防火墙 8](#_Toc480202489)

[7.1. 更新防火墙配置 8](#_Toc480202490)

[7.2. 检查防火墙状态 9](#_Toc480202491)

[8. Cassandra security 9](#_Toc480202492)

[modify cassandra.yaml: 9](#_Toc480202493)

[9. 创建用户，表 10](#_Toc480202494)

# 硬件参数

|  |  |  |
| --- | --- | --- |
| **Item** | **Specification** | **Remarks** |
| RAM | 32GB |  |
| CPU | 8 Cores |  |
| OS Disk | 20 GB |  |
| Data Disk | 2048 GB | /var/lib/cassandra |
| Log Disk | 512 GB | /var/log/cassandra |

# 收集信息

Fill the form below

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Key** | **Value** | **Comments** |
| Cassandra Node 1 | IP Address |  | Seed node |
| Cassandra Node 2 | IP Address |  | Seed node |
| Cassandra Node 3 | IP Address |  |  |
| App Server 1 | IP Address |  |  |
| App Server 2 | IP Address |  |  |

# 开始之前

## 所需文件

确保你获得以下文件:

* 无

## 所需工具

准备以下工具:

* WinSCP (<http://winscp.net/> )
* PuTTY (<http://www.putty.org/>)
* Notepad ++ (可选, <https://notepad-plus-plus.org/>) (WinSCP与Notepad++的集成在这里<https://winscp.net/eng/docs/integration_editor>)

# 安装前置

本说明以以下假设为前提: 服务器满足硬件参数条件; Linux HyperV虚拟机集成服务或者VmWare虚拟机的VmWare Tools已安装.

关于Linux集成服务的更多信息请参阅:

<http://blogs.technet.com/b/virtualization/archive/2015/05/01/linux-integration-services-4-0-announcement.aspx>

## 更新Hostname

hostnamectl set-hostname **CassandraNodeName**

例如:

hostnamectl set-hostname Cassandra\_Node\_1

# 安装 Cassandra

## Installing the Oracle Java Virtual Machine

Cassandra requires that the Oracle Java SE Runtime Environment (JRE) be installed. So, in this step, you'll install and verify that it's the default JRE.

To make the Oracle JRE package available, you'll have to add a Personal Package Archives (PPA) using this command:

sudo add-apt-repository ppa:webupd8team/java

Update the package database:

sudo apt-get update

Then install the Oracle JRE. Installing this particular package not only installs it but also makes it the default JRE. When prompted, accept the license agreement:

sudo apt-get install oracle-java8-set-default

After installing it, verify that it's now the default JRE:

java -version

You should see output similar to the following:

Output

java version "1.8.0\_60"

Java(TM) SE Runtime Environment (build 1.8.0\_60-b27)

Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)

## Installing Cassandra

We'll install Cassandra using packages from the official Apache Software Foundation repositories, so start by adding the repo so that the packages are available to your system. Note that Cassandra 2.2.2 is the latest version at the time of this publication. Change the 22x to match the latest version. For example, use 23x if Cassandra 2.3 is the latest version:

echo "deb http://www.apache.org/dist/cassandra/debian 22x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list

The add the repo's source:

echo "deb-src http://www.apache.org/dist/cassandra/debian 22x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list

To avoid package signature warnings during package updates, we need to add three public keys from the Apache Software Foundation associated with the package repositories.

Add the first one using this pair of commands, which must be run one after the other:

gpg --keyserver pgp.mit.edu --recv-keys F758CE318D77295D

gpg --export --armor F758CE318D77295D | sudo apt-key add -

Then add the second key:

gpg --keyserver pgp.mit.edu --recv-keys 2B5C1B00

gpg --export --armor 2B5C1B00 | sudo apt-key add -

Then add the third:

gpg --keyserver pgp.mit.edu --recv-keys 0353B12C

gpg --export --armor 0353B12C | sudo apt-key add -

Update the package database once again:

sudo apt-get update

Finally, install Cassandra:

sudo apt-get install cassandra

## Troubleshooting and Starting Cassandra

Ordinarily, Cassandra should have been started automatically at this point. However, because of a bug, it does not. To confirm that it's not running, type:

sudo service cassandra status

If it is not running, the following output will be displayed:

Output

\* could not access pidfile for Cassandra

This is a well-known issue with the latest versions of Cassandra on Ubuntu. We'll try a few fixes. First, start by editing its init script. The parameter we're going to modify is on line 60 of that script, so open it using:

sudo vim /etc/init.d/cassandra

That line should read:

/etc/init.d/cassandra

CMD\_PATT="cassandra.+CassandraDaemon"

Change it to:

/etc/init.d/cassandra

CMD\_PATT="cassandra"

Close and save the file, then reboot the server:

sudo reboot

Or:

sudo shutdown -r now

After logging back in, Cassandra should now be running. Verify:

sudo service cassandra status

If you are successful, you will see:

Output

\* Cassandra is running

## Connecting to the Cluster

If you were able to successfully start Cassandra, check the status of the cluster:

sudo nodetool status

In the output, **UN** means it's **U**p and **N**ormal:

Output

Datacenter: datacenter1

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

Status=Up/Down

|/ State=Normal/Leaving/Joining/Moving

-- Address Load Tokens Owns Host ID Rack

UN 127.0.0.1 142.02 KB 256 ? 2053956d-7461-41e6-8dd2-0af59436f736 rack1

Note: Non-system keyspaces don't have the same replication settings, effective ownership information is meaningless

Then connect to it using its interactive command line interface cqlsh.

cqlsh

You will see it connect:

Output

Connected to Test Cluster at 127.0.0.1:9042.

[cqlsh 5.0.1 | Cassandra 2.2.2 | CQL spec 3.3.1 | Native protocol v4]

Use HELP for help.

cqlsh>

Type exit to quit:

exit

# 配置Cassandra节点

## 编辑节点配置

按如下修改: /etc/cassandra/cassandra.yaml

**listen\_address:** **本机IP 非127.0.0.1**如："172.16.3.1"

**rpc\_address:** **本机IP 非127.0.0.1**如："172.16.3.1"

**seeds: 本机IP 非127.0.0.1** 如："172.16.3.1"

重启后登录ubuntu

## 检查Cassandra服务状态

/usr/bin/nodetool status

输出应该如下:

Datacenter: datacenter1

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

Status=Up/Down

|/ State=Normal/Leaving/Joining/Moving

-- Address Load Tokens Owns (effective) Host ID Rack

UN 192.168.137.127 51.67 KB 256 100.0% 26acca41-8ffa-4962-96bd-d1d21a483b18 rack1

And after you’ve finished setting up all 3 nodes, you’ll see output:

Datacenter: datacenter1

=======================ss

Status=Up/Down

|/ State=Normal/Leaving/Joining/Moving

-- Address Load Tokens Owns (effective) Host ID Rack

UN 192.168.137.221 192.32 KB 256 67.0% 26acca41-8ffa-4962-96bd-d1d21a483b18 rack1

UN 192.168.137.222 168.46 KB 256 64.9% 487623dd-50d0-4c08-af97-0bb84fbe70cb rack1

UN 192.168.137.223 113.61 KB 256 68.1% b141c533-541f-40a8-b64e-c94fdfb2f1e7 rack1

# 配置防火墙

## 更新防火墙配置

apt install firewalld

firewall-cmd --permanent --zone=internal --add-port=8888/tcp

firewall-cmd --permanent --zone=internal --add-port=7000/tcp

firewall-cmd --permanent --zone=internal --add-port=7001/tcp

firewall-cmd --permanent --zone=internal --add-port=7199/tcp

firewall-cmd --permanent --zone=internal --add-port=9042/tcp

firewall-cmd --permanent --zone=internal --add-port=9160/tcp

firewall-cmd --permanent --zone=internal --add-port=61620/tcp

firewall-cmd --permanent --zone=internal --add-port=61621/tcp

firewall-cmd --permanent --zone=internal --add-source=**cassandra\_node\_1\_ip\_addr**

firewall-cmd --permanent --zone=internal --add-source=**cassandra\_node\_2\_ip\_addr**

firewall-cmd --permanent --zone=internal --add-source=**cassandra\_node\_3\_ip\_addr**

firewall-cmd --permanent --zone=internal --add-source=**app\_server\_ip1**

firewall-cmd --permanent --zone=internal --add-source=**app\_server\_ip2**

firewall-cmd --reload

## 检查防火墙状态

firewall-cmd --zone=internal --list-all

输出应该如下

internal

interfaces:

sources: ***ip\_addresses***

services: dhcpv6-client ipp-client mdns samba-client ssh

ports: **8888/tcp 7000/tcp 7001/tcp 7199/tcp 9042/tcp 9160/tcp 61620/tcp 61621/tcp**

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

# Cassandra security

 You could do the following steps:

 (To login and execute commands in cassandra, you can use my pretty CassandraGUI or ugly cqlsh)

## modify cassandra.yaml:

vim /etc/Cassandra/Cassandra.yaml

更改一下属性：

authenticator: PasswordAuthenticator

authorizer: CassandraAuthorizer

(for cluster)Increase the replication factor for the system\_auth keyspace to N (number of nodes).

restart cassandra service

login with cassandra/Cassandra

GUI登录方式：

ContactPoints: Cassandra.yaml 内listen\_address

Port:9042

UserName：cassandra

Password：cassandra

CMD登录：

cqlsh –u cassandra –p cassandra

执行：

alter user cassandra  with password '!honeywell1';

CMD登录：

cqlsh –u cassandra –p ‘!honeywell1’

执行：

CREATE USER IF NOT EXISTS bpspub WITH PASSWORD '!guomao1'  NOSUPERUSER;

GRANT MODIFY PERMISSION ON ALL KEYSPACES TO bpspub;

GRANT SELECT PERMISSION ON ALL KEYSPACES TO bpspub;

# 创建用户，表

在CassandraGUI中执行以下命令，创建用户，表：

路径：

$BPSR200\bpsr200\7-SourceCode\Platform\Deployment\DBScripts\Cassandra\ cassandra.txt

CREATE KEYSPACE IF NOT EXISTS AlarmData WITH replication = {

'class': 'SimpleStrategy',

'replication\_factor': '3'

}

AND durable\_writes = true;

CREATE TABLE IF NOT EXISTS AlarmData.AlarmHistory (

Id uuid,

AlarmName text,

AlarmDescription text,

AlarmDateTime timestamp,

AlarmMonth int,

AlarmType int,

AlarmLevel int,

SiteId int,

SubsystemId int,

EquipmentId int,

PointId int,

AlarmRootCause int,

Comment text,

AlarmInfos text,

AlarmRuleId uuid,

AlarmId text,

AlarmStatus int,

AlarmAckStatus int,

WorkOrders text,

PRIMARY KEY ((SiteId, AlarmMonth), AlarmDateTime, Id)

) WITH CLUSTERING ORDER BY (AlarmDateTime DESC);