Lab 6 Oracle Linux MongoDB

This tutorial provides guidance on the installation and use of MongoDB within the Oracle Linux VM we use throughout this course.

Begin by setting your network settings to NAT in Oracle Virtualbox for the Oracle Developer VM. Then, I recommend creating a new snapshot prior to attempting to install MongoDB that can be restored if issues occur.

As an alternative to the class VM you can search “Bitnami MongoDB Stack for Windows” to get a working installation for Windows of MongoDB as well. While our lab we will focus on Linux, Windows is a great alternative to learn how to use MongoDB. Once installed and running, you can jump to the section below where you create the first collection.

Virtualbox Escape Keys

Right Ctrl (Host Key) + c

By default, the Host key is the Right Ctrl key on the keyboard.

HOST KEY + L

RIGHT CTRL + L

HOST + F

Right CTRL key to release the mouse and keyboard

Host key + Del to send Ctrl+Alt+Del (to reboot the guest);

Host key + Backspace to send Ctrl+Alt+Backspace (to restart the graphical user interface of a Linux guest);

Host key + F1 (or other function keys) to simulate Ctrl+Alt+F1 (or other function keys, i.e. to switch between virtual terminals in a Linux guest).

Make sure you have an outside connection on your VM first as you need to install software from the Internet for this lab to work. We will be using a terminal window or bash in Oracle Linux. Open a terminal and sudo as root

? sudo su –

Next check your IP address and assure you have Internet access using the ping command, or open Firefox from the applications on the Oracle Linux VM

# ip addr show

# ping -c4 www.yahoo.com

Next use vi or another file editor to create a mongoDB repo in the /var/yum.repos.d directory. Currently version 4.2 of MongoDB is stable:

# vi /etc/yum.repos.d/mongodb-org-4.2.repo

In the file copy and paste this exactly:

[mongodb-org-4.2]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.2/x86\_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-4.2.asc

Once the repo has been added, install the mongodb-org package

# sudo yum install -y mongodb-org

The install should be successful, all packages should download and install without error.

Next, either add firewall rules or disable the firewall to avoid issues

To disable firewalld.

# systemctl stop firewalld

# systemctl disable firewalld

# firewall-cmd --state

Or to add rules:

# sudo firewall-cmd --add-port=27017/tcp --permanent

# sudo firewall-cmd --reload

Next, check the MongoDB data store location

$ sudo vi /etc/mongod.conf

To check the log files the .conf file will show the location, by default it is:

/var/log/mongodb/mongod.log

Now is a good time to check this log

Next, set the mongod service to start on boot.

# sudo systemctl enable mongod

# sudo systemctl status mongod

If everything worked up until this point we can now start the MongoDB by using the following command:

# service mongod start

If necessary (not now), you can stop the MongoDB by using the following command:

# service mongod stop

Restart the MongoDB by using the following command:

# service mongod restart

# Make sure it is running if it fails

sudo service mongod start

To create an initial collection go into the mongo DB using the command:

$ mongo

To display the database you are using, type db:

db

The operation should return test, which is the default database. To switch databases, issue the use <db> helper, as in the following example:

use <database>

# To create a new database enter "use" followed by the database name:

> use products;

# To insert records:

db.products.insert( { item: "present", qty: 5 } );

db.products.insert( { item: "gift", qty: 10 } );

# During the insert, mongod will create the \_id field and assign it a unique ObjectId value, as verified by the inserted document

In the following example, the document passed to the insert() method includes the \_id field. The value of \_id must be unique within the collection to avoid duplicate key error.

db.products.insert( { \_id: 15, item: "box", qty: 20 } );

# For an array

db.products.insert(

[

{ \_id: 11, item: "jacket", qty: 50, type: "winter" },

{ item: "gloves", qty: 20 },

{ item: "hats", qty: 25 }

{ item: "mittens", qty: 50 }

{ item: "boots", qty: 50 }

]

);

You can check the created collection by using the command show collections.

>show collections

>use products;

Instead of tables, Mongo uses collections and instead of rows, Mongo uses documents.

# To find results from the product collections (e.g. table) enter:

>db.products.find();

To find a specifid ID:

db.products.find({ "\_id" : ObjectId("5beca50f88890e8a310b7d5a") });

db.collection.deleteOne deletes the first document that matches the filter. Use a field that is part of a unique index such as \_id for precise deletions.

The following example deletes one object ID:

db.products.deleteOne( { "\_id" : ObjectId("5beca4ce88890e8a310b7d58") } );

Mongo documentation recommends putting queries in a try and catch block like:

try {

db.products.deleteOne( { "\_id" : ObjectId("5beca4ce88890e8a310b7d58") } );

} catch (e) {

print(e);

}

To delete the entire collection you would use:

db.collectionname.deleteMany({})

You can delete all and re-add to the collection to test.

db.products.insert(

[

{ item: "gloves", qty: 20, type: "winter" },

{ item: "hats", qty: 25, type: "winter" },

{ item: "mittens", qty: 50, type: "winter" },

{ item: "boots", qty: 50, type: "winter" },

{ item: "shorts", qty: 10, type: "summer" },

{ item: "sunglasses", qty: 15, type: "summer" },

]

);

>Exit

If you have reached this point without issue, congratulations on your first MongoDB implementation!

If necessary, you may need to update all packages. This can take a significant amount of time on the Oracle VM. Thus, you may want to do these updates at a later point.

Once your VM can connect to the Internet update the package manager using yum

# yum update

To list installed and available packages to see what versions you need

yum list installed

yum list available