Latest AWS notes to create an EC2 Instance

Step 1:
Created CentOs ec2 instance on aws.

Step2: Before to start use this two commands

yum install deltarpm epel-release
yum update

Step3: sudo yum install wget

Install wget if it is not supporting.

Step4: Download and Install java

wget --no-cookies --no-check-certificate --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u141-b15/336fa29ff2bb4ef291e347e091f7f4a7/jdk-8u141-linux-x64.tar.gz"

Step5:
Move java setup into opt folder

Step6: cd /opt/
tar xzf jdk-8u141-linux-x64.tar.gz

Step7:alternatives --install /usr/bin/java java /opt/jdk1.8.0_141/bin/java 2
Step8:
alternatives --config java

step9: **export JAVA_HOME=/opt/jdk1.8.0_141**
Step10: **export JRE_HOME=/opt/jdk1.8.0_141/jre**
Step11: Setup Path export PATH=$PATH:/opt/jdk1.8.0_141/bin:/opt/jdk1.8.0_141/jre/bin

Tomcat Setup on centos Instance:

Step1: sudo groupadd tomcat
Step2:
sudo useradd -M -s /bin/nologin -g tomcat -d /opt/tomcat tomcat

Step3: wget http://www-us.apache.org/dist/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz
step4: sudo mkdir /opt/tomcat
Step 5: sudo tar xvf apache-tomcat-8*tar.gz -C /opt/tomcat --strip-components=1

Change to the Tomcat installation path:

cd /opt/tomcat

Give the tomcat group ownership over the entire installation directory:

sudo chgrp -R tomcat /opt/tomcat

Next, give the tomcat group read access to the conf directory and all of its contents, and execute access to the directory itself:

sudo chmod -R g+r conf
sudo chmod g+x conf

Then make the tomcat user the owner of the webapps, work, temp, and logs directories:

sudo chown -R tomcat webapps/ work/ temp/ logs/

Now that the proper permissions are set up, let's set up a Systemd unit file.

Step6: Install Systemd Unit File

Because we want to be able to run Tomcat as a service, we will set up a Tomcat Systemd unit file .

Create and open the unit file by running this command:

sudo vi /etc/systemd/system/tomcat.service

Paste in the following script. You may also want to modify the memory allocation settings that are specified in CATALINA_OPTS:

/etc/systemd/system/tomcat.service

```
# Systemd unit file for tomcat
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target
OnFailure=unit-status-mail@%n.service

[Service]
Type=forking

Environment=JAVA_HOME=/opt/jdk1.8.0_141/jre
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms256m -Xmx512m -XX:MaxPermSize=128m'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat
Group=tomcat

[Install]
WantedBy=multi-user.target
```

Save and exit. This script tells the server to run the Tomcat service as the tomcat user, with the settings specified.

Now reload Systemd to load the Tomcat unit file:

```
sudo systemctl daemon-reload
```

Now you can start the Tomcat service with this systemctl command:

```
sudo systemctl start tomcat
```

Check that the service successfully started by typing:

sudo systemctl status tomcat

If you want to enable the Tomcat service, so it starts on server boot, run this command:

sudo systemctl enable tomcat

MongoDB installation guide on ec2.

Step1: Please follow this site
https://tecadmin.net/install-mongodb-on-centos-rhel-and-fedora/

Step2:

Start MongoDB without access control.

mongod --port 27017 --dbpath /data/db1

Connect to the instance.

mongo --port 27017

Create the user administrator.

db.createUser(

```
{

  user: "clubShopAdmin",

  pwd: "clubshop@123",

  roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]

} )
```

Re-start the MongoDB instance with access control.

mongod --auth --port 27017 --dbpath /data/db1

Connect and authenticate as the user administrator

To authenticate during connection

mongo --port 27017 -u "clubShopAdmin" -p "clubshop@123" --authenticationDatabase "admin"

To authenticate after connecting

mongo --port 27017

use admin
db.auth("clubShopAdmin", "clubshop@123")

Craete additions user:

use clubshopDb

db.createUser(

```
{

  user: "clubshopAdmin",

  pwd: "clunShop123",

  roles: [ { role: "readWrite", db: "clubshopDb" },

        { role: "read", db: "clubShopReporting" } ]

}
```

```
)
```

```
mongo --port 27017 -u "clubshopAdmin" -p "clunShop123" --authenticationDatabase "clubshopDb"
```

```
use clubshopDb
```

```
db.auth("clubshopAdmin", "clunShop123" )
```

If you want you can create another user for TESTING only having access READ only. Using same process.

Enable authentication on Mongodb by editing /etc/mongod.conf. It should have the following line

```
security:
  authorization: enabled
```

Create a addition user on Mongodb Use MongoChef for it from your laptop

Congratulatiosn you creted mongo setup on aws successfully !!!