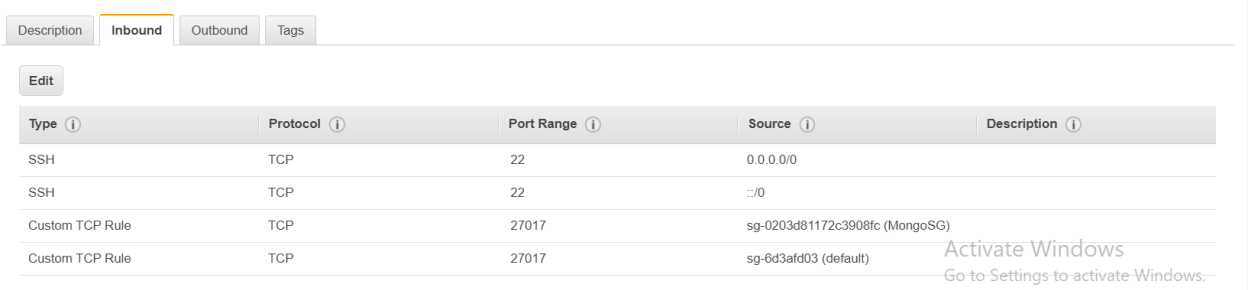
Mongo Replication on RHEL

Documentation by Aniket Rao

1. Create a security group with the ports 22 and 27017 enabled. In the source for Port 22 it should be 0.0.0.0/0 and :/0 but for Port 27017 it should be the security group created and default security group as shown below:



Description | Inbound | Outbound | Tags

Edit

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	
Custom TCP Rule	TCP	27017	sg-0203d81172c3908fc (MongoSG)	
Custom TCP Rule	TCP	27017	sg-6d3afd03 (default)	

2. Launch a RHEL 8 instance and during setup in “Configure Instance” add in the following Bash Script in Advanced Details ---> User data so that everything is setup and installed in your instance before hand

```
#!/bin/bash
        #set debug mode
        set -x

        ins=$(cat <<- END
        [mongodb-org-4.0]
        name=MongoDB Repository
        baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/4.0/x86_64/
        gpgcheck=1
        enabled=1
        gpgkey=https://www.mongodb.org/static/pgp/server-4.0.asc
        END
        )

        echo "$ins" | sudo tee /etc/yum.repos.d/mongodb-org-4.0.repo

        yes Y | sudo yum install -y mongodb-org

        hugepages=$(cat <<- EOP
        #!/bin/bash
        ### BEGIN INIT INFO
        # Provides:        disable-transparent-hugepages
        # Required-Start:   $local_fs
        # Required-Stop:
        # X-Start-Before:   mongod mongodb-mms-automation-agent
        # Default-Start:    2 3 4 5
        # Default-Stop:     0 1 6
```

```
# Short-Description: Disable Linux transparent huge pages
# Description:       Disable Linux transparent huge pages, to improve
#                    database performance.
### END INIT INFO
case \$1 in
  start)
    if [ -d /sys/kernel/mm/transparent_hugepage ]; then
      thp_path=/sys/kernel/mm/transparent_hugepage
    elif [ -d /sys/kernel/mm/redhat_transparent_hugepage ]; then
      thp_path=/sys/kernel/mm/redhat_transparent_hugepage
    else
      return 0
    fi
    echo 'never' > \${thp_path}/enabled
    echo 'never' > \${thp_path}/defrag
    re='^[0-1]+\$'
    if [[ \$(cat \${thp_path}/khugepaged/defrag) =~ \$re ]]
    then
      # RHEL 7
      echo 0  > \${thp_path}/khugepaged/defrag
    else
      # RHEL 6
      echo 'no' > \${thp_path}/khugepaged/defrag
    fi
    unset re
    unset thp_path
    ;;
esac
EOP
)

echo "$hugepages" | sudo tee /etc/init.d/disable-transparent-hugepages

sudo chmod 755 /etc/init.d/disable-transparent-hugepages

sudo chkconfig --add disable-transparent-hugepages

softlim=$(cat <<- END
mongod soft nproc 64000
END
)
echo "$softlim" | sudo tee -a /etc/security/limits.conf

mongoconf=$(cat <<- END
# mongod.conf
# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/
# where to write logging data.
```

```
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log
# how the process runs
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /var/run/mongodb/mongod.pid  # location of pidfile
  timeZoneInfo: /usr/share/zoneinfo
# Where and how to store data.
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true
  engine: wiredTiger
  wiredTiger:
    collectionConfig:
      blockCompressor: none
# network interfaces
net:
  port: 27017
  bindIp: mongo.domain.net  # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, alternatively, use the net.bindIpAll setting.
  ssl:
    mode: requireSSL
    PEMKeyFile: /etc/ssl/mongo_ssl/mongodb.pem
    CAFile: /etc/ssl/mongo_ssl/CA.pem
    allowInvalidCertificates: true
    allowInvalidHostnames: true
#security:
#  authorization: enabled
#  keyFile: /var/lib/mongo/rsetkey
#operationProfiling:
#replication:
#  replSetName: TestRS-0
#sharding:
## Enterprise-Only Options
#auditLog:
#snmp:
END
)
echo "$mongoconf" | sudo tee /etc/mongod.conf

mkdir /home/ec2-user/mongo_ssl
cd /home/ec2-user//mongo_ssl
openssl req -out CA.pem -new -x509 -days 365 -keyout CAPrivKey.pem -subj
"/C=IN/ST=karnataka/O=Organisation/CN=*.domain.net/emailAddress=user@domain.com" -nodes

echo "00" > serial_num.srl # two random digits number
```

```
openssl genrsa -out mongodb.key 2048
openssl req -key mongodb.key -new -out mongodb.req -subj
"/C=IN/ST=karnataka/O=Organisation/CN=server/CN=*.domain.net/emailAddress=user@domain.com" -nodes
openssl x509 -req -in mongodb.req -CA CA.pem -CAkey CAPrivKey.pem -CAserial serial_num.srl -out mongodb.crt -days 365
cat mongodb.key mongodb.crt > mongodb.pem
openssl verify -CAfile CA.pem mongodb.pem

openssl genrsa -out mclient.key 2048
openssl req -key mclient.key -new -out mclient.req -subj "/C=IN/ST=karnataka/O=Organisation/CN=client/emailAddress=user@domain.com" -
nodes
openssl x509 -req -in mclient.req -CA CA.pem -CAkey CAPrivKey.pem -CAserial serial_num.srl -out mclient.crt -days 365
cat mclient.key mclient.crt > mclient.pem
openssl verify -CAfile CA.pem mclient.pem

#once everything is completed let's move it to /etc/ssl/ folder

sudo cp -R /home/ec2-user/mongo_ssl /etc/ssl
cd mongo_ssl
sudo chown mongod:mongod *

sudo openssl rand -base64 756 |sudo tee /var/lib/mongo/rsetkey
sudo chmod 400 /var/lib/mongo/rsetkey
sudo chown mongod:mongod /var/lib/mongo/rsetkey

sudo service mongod start

sudo chkconfig mongod on
```

3. Select the Security Group you created for this instance and launch the instance

4. Go to Route 53 and create a private host zone.

   - Give a domain name In my example I gave domain.net
   - Type will be private. When private is selected, it will ask for VPC, select the region you are working in and then click create
   - Now create a sub domain but clicking on “Create Set Record”
   - In Name add a sub domain name for ex: mongo1

- In value add Private IP of the instance and click create.

5. Create Image of the instance to create a replica instances and avoiding re-installation of packages.

6. Launch 2 separate instances with the image. Create two Set Record in Route 53, mongo2.domain.net and mongo3.domain.net and in value add Private IP's in respective Set Records.

7. SSH Login in Mongo1 and go to /etc/mongod.conf
   $ vi /etc/mongod.conf

8. Change the Bind IP value to mongo1.domain.net

9. Similarly change it in Mongo 2 and Mongo 3.

10. Restart mongo Services
    $ service mongod restart

11. To login to mongo use the following:
    $ mongo mongo1.domain.net –ssl –sslCAFile /etc/ssl/mongo_ssl/CA.pem –sslPEMKeyFile /etc/ssl/mongo_ssl/mclient.pem

12. Similarly do it for other two instances

13. Try restarting Mongo Services in the other two instances. It will throw an error. Reboot of instance using reboot command will fix it.

14. Login in to the Primary Mongo (In this case mongo1)

    $ mongo mongo1.domain.net –-ssl –-sslCAFile /etc/ssl/mongo_ssl/CA.pem –-sslPEMKeyFile /etc/ssl/mongo_ssl/mclient.pem

15. Use the database "admin"

```
>  use admin
```

Switched to db admin

```
> db.createUser({"user": "admin", "pwd": "yourpassword", "roles": [{"role": "root", "db": "admin"}]})
```

You will get a Successfully added user comment. If error occurs, check for syntax.

To authenticate created user

```
> db.auth("admin", "yourpassword")
```

1

If you get 1 as output, it is correct.

16. In /etc/mongod.conf Uncomment the following lines in all 3 instances
    - Security
    - Authorization
    - Keyfile
    - Replication
    - ReplicaName

17. Set the same replica name in all the instances in /etc/mongod.conf

18. Restart mongo services in all the instances.

19. To login into mongo1 with created user and password:

    $ mongo –u admin –p yourpassword mongo1.domain.net/admin –-ssl –-sslCAFile /etc/ssl/mongo_ssl/CA.pem –-sslPEMKeyFile /etc/ssl/mongo_ssl/mclient.pem

20. Add the following command in Mongo

```
rs.initiate(
              {
                 _id: "TestRS-0", #Find this name in /etc/mongod.conf in ReplicaName
                 members: [
                    { _id: 0, host : "mongo1.domain.net:27017" },
                    { _id: 1, host : "mongo2.domain.net:27017" },
                    { _id: 2, host : "mongo3.domain.net:27017", "arbiterOnly": true}
                 ]
              }
           )
```

21. To check whether it took into effect

    > rs.status()

22. Now you can view contents of the database from any of the 3 instances using the following command:

    $ mongo –u admin –p yourpassword mongo1.domain.net/admin –ssl –sslCAFile /etc/ssl/mongo_ssl/CA.pem –sslKeyFile /etc/ssl/mongo_ssl/mclient.pem

    Here mongo1 is primary, mongo2 is secondary and mongo3 is arbitrary.

References:

1. https://medium.com/@greeshu.renu/how-to-setup-three-member-replica-set-on-amazon-ec2-mongodb-60f0aaddcf32
2. https://gist.github.com/Greeshu/a5833afa286147d7672e975c798e8691
3. https://www.youtube.com/watch?v=yp7VJihdvTM