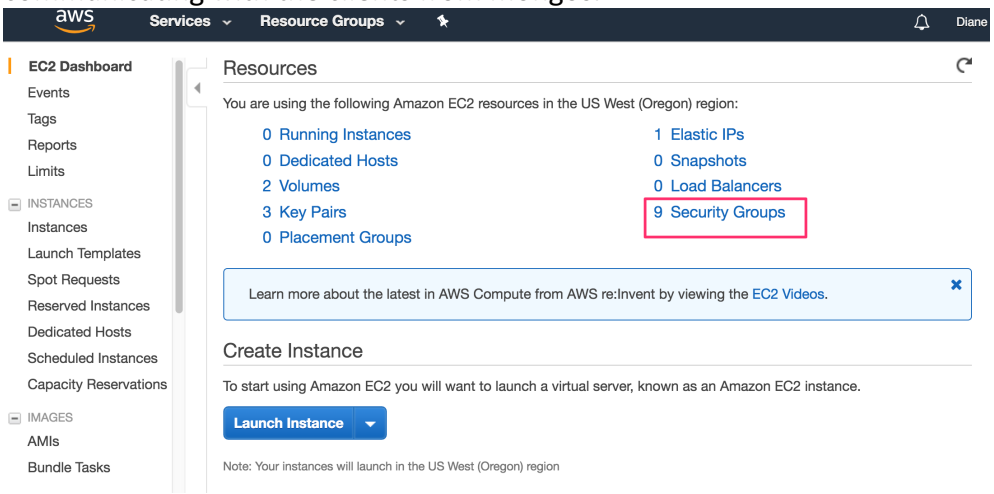
MSDS 697 Creating replicated sharded MongoDBs on AWS

Objective

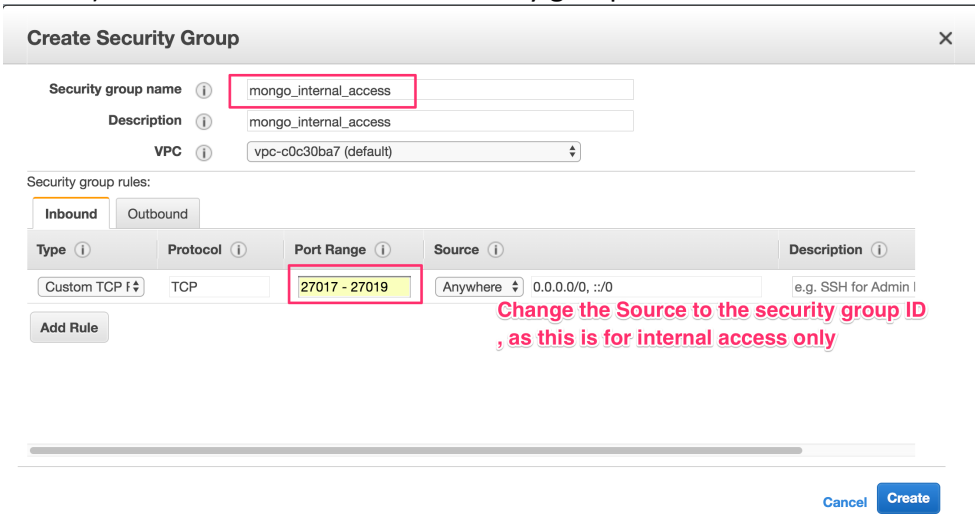
Create ten servers for replication and sharding - one server for the mongos and three servers each for the first replica set, the second replica set, and the config server replica set on AWS.

Steps

Step 1. Create two security groups on EC2 ➔ Security Groups – one for communicating between shards, replicated nodes, configuration server and mongos and the other for communicating with the clients from mongos.

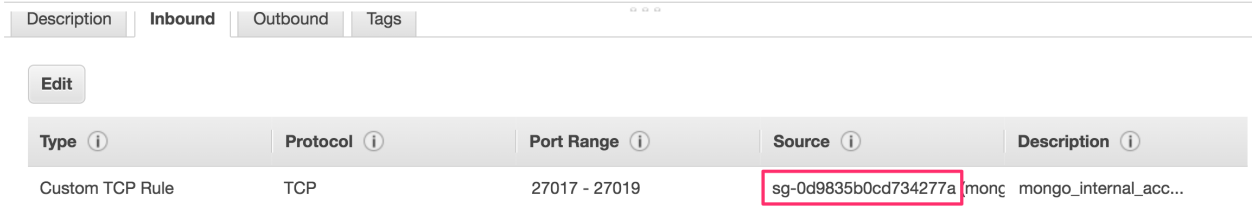


1.1) Create internal access security group

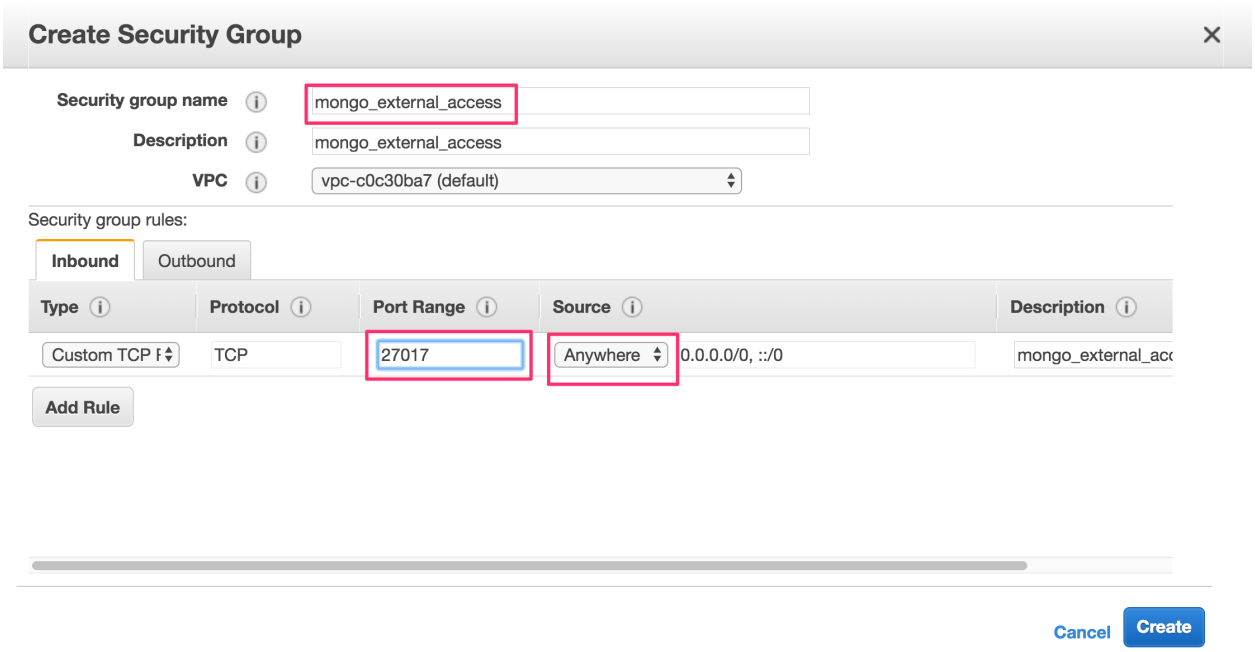


port 27017 - mongos
port 27018 - shards
port 27019 - config server

Once the security group is created, change the source as its security group ID.



1.2) Create external access security group



27107 – Clients to mongos.

Step 2. Launch 10 instances (1 for mongos, 3 for configuration, 3 for shard1, and 3 for shard2)

2.1) Choose an Amazon Linux (Storage optimized would be a better choice but $$).

a. Make sure to choose default and mongo_internal_access. (For mongos, also choose mongo_external_access.)

2.2) ssh to each instance and install MongoDB.

a. Follow steps mentioned in https://docs.mongodb.com/manual/tutorial/install-mongodb-on-amazon/

i. Create a /etc/yum.repos.d/mongodb-org-4.0.repo and edit.

ii. `sudo yum install -y mongodb-org`

iii. start mongod **(Except for the mongos server)**

`sudo service mongod start`

```
[ec2-user@ip-172-31-31-46 ~]$ sudo service mongod start
Starting mongod (via systemctl):                           [  OK  ]
```

2.3) Create an AMI to create multiple instances with the same configuration and repeat.

2.4) For each instance **(Except for mongos)**

a. Create /data/db :

`sudo mkdir -p /data/db`

```
[ec2-user@ip-172-31-31-46 ~]$ sudo mkdir -p /data/db
```

b. Change the owner of /data/db to mongod **(Except for mongos)** :

`sudo chown -R mongod:mongod /data/db`

```
[ec2-user@ip-172-31-31-46 ~]$ sudo chown -R mongod:mongod /data/db
```

Step 3. Create 3-node replica sets for 2 shards.

3.1) On /etc/mongod.conf of each node, update the configuration.

- Change
  ```
  storage:
    dbPath: /data/db
  ```
- Add `port: 27018`
- Add `bindIp: 0.0.0.0`
- Add
  ```
  replication:
    replSetName: rs1 (or rs2)
  ```
- Add
  ```
  sharding:
    clusterRole: shardsvr
  ```

```
# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
  dbPath: /data/db
  journal:
    enabled: true
#  engine:
#  mmapv1:
#  wiredTiger:

# 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

# network interfaces
net:
  port: 27018
  bindIp: 0.0.0.0  # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addr


#security:

#operationProfiling:

replication:
  replSetName: rs2

sharding:
  clusterRole: shardsvr
```

3.2) Start mongod on each node using

```
sudo mongod --config /etc/mongod.conf
[ec2-user@ip-172-31-31-46 ~]$ sudo mongod --config /etc/mongod.conf
```

3.3) Connect a mongo shell to one of the mongod instances and run rs.initiate() per shard.

```
sudo mongo --port 27018
config = {_id : "rs1 (or rs2)",
members:[{_id:0,host:"PRIVATE_IP_ADDRESS:27018"},
         {_id:1,host:"PRIVATE_IP_ADDRESS: 27018"},
         {_id:2,host:"PRIVATE_IP_ADDRESS: 27018"}]}
rs.initiate(config)
```

```
> config = {_id : "rs2", members:[{_id:0,host:"172.31.33.99:27018"},{_id:1,host:"172.31.45.208:27018"},{_id:2,host:"172.31.27.49:27018"}]}
{
        "_id" : "rs2",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "172.31.33.99:27018"
                },
                {
                        "_id" : 1,
                        "host" : "172.31.45.208:27018"
                },
                {
                        "_id" : 2,
                        "host" : "172.31.27.49:27018"
                }
        ]
}
> rs.initiate(config)
{
        "ok" : 1,
        "operationTime" : Timestamp(1546827166, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1546827166, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

Step 4. Create a 3-node replica set for config server.

4.1) On /etc/mongod.conf of each node, update the configuration.

- Change
  ```
  storage:
    dbPath: /data/db
  ```
- Add `port: 27019`
- Comment `#bindIp` line
- Add
  ```
  replication:
    replSetName: config_rs
  ```
- Add
  ```
  sharding:
    clusterRole: configsvr
  ```

```
# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
  dbPath: /data/db
  journal:
    enabled: true
#  engine:
#  mmapv1:
#  wiredTiger:

# 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

# network interfaces
net:
  port: 27019
  bindIp: 0.0.0.0  # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses


#security:

#operationProfiling:

replication:
  replSetName: config_rs

sharding:
  clusterRole: configsvr

## Enterprise-Only Options

#auditLog:

#snmp:
```

4.2) Start mongod using

```
sudo mongod --config /etc/mongod.conf
```

4.3) Connect a mongo shell to one of the mongod instances and run rs.initiate() per shard.

```
sudo mongo --port 27019
config = {_id : "config_rs",
members:[{_id:0,host:"PRIVATE_IP_ADDRESS:27019"},
         {_id:1,host:"PRIVATE_IP_ADDRESS: 27019"},
         {_id:2,host:"PRIVATE_IP_ADDRESS: 27019"}]}
rs.initiate(config)
```

```
> config = {_id : "config_rs", members:[{_id:0,host:"172.31.31.46:27019"},{_id:1,host:"172.31.25.251:27019"},{_id:2,host:"172.31.30.219:27019"}]}
{
        "_id" : "config_rs",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "172.31.31.46:27019"
                },
                {
                        "_id" : 1,
                        "host" : "172.31.25.251:27019"
                },
                {
                        "_id" : 2,
                        "host" : "172.31.30.219:27019"
                }
        ]
}
>
> rs.initiate(config)
{
        "ok" : 1,
        "operationTime" : Timestamp(1546838072, 1),
        "$gleStats" : {
                "lastOpTime" : Timestamp(1546838072, 1),
                "electionId" : ObjectId("000000000000000000000000")
        },
        "lastCommittedOpTime" : Timestamp(0, 0),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1546838072, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

Step 5. Configure monogos, add shards, and connect to clients.

5.1) On /etc/mongod.conf of each node, update the configuration.

- Comment the storage setting, as it is not going to store data.
- Comment #bindIp line.
- Add configDB under sharding.

```
sharding:
  configDB: congif_rs/PRIVATE_IP_ADDRESS:27019,
PRIVATE_IP_ADDRESS:27019,PRIVATE_IP_ADDRESS:27019
```

```
# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
#storage:
#  dbPath: /var/lib/mongo
#  journal:
#    enabled: true
#  engine:
#  mmapv1:
#  wiredTiger:

# 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

# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0  # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or,


#security:

#operationProfiling:

#replication:

sharding:
  configDB: config_rs/172.31.31.46:27019,172.31.25.251:27019,172.31.30.219:27019

## Enterprise-Only Options

#auditLog:
```

5.2) Start mongos using

```
sudo mongos --config /etc/mongod.conf
```

5.3) Connect a mongo shell

```
sudo mongo --port 27017
```

5.4) Add two shards and their replica sets.

```
sh.addShard("rs1/PRIVATE_IP_ADDRESS:27018,PRIVATE_IP_ADDRESS:27018,
PRIVATE_IP_ADDRESS:27018")
sh.addShard("rs2/PRIVATE_IP_ADDRESS:27018,PRIVATE_IP_ADDRESS:27018,
PRIVATE_IP_ADDRESS:27018")
```

```
mongos> sh.addShard("rs1/172.31.30.123:27018,172.31.29.92:27018,172.31.30.237:27018")
{
        "shardAdded" : "rs1",
        "ok" : 1,
        "operationTime" : Timestamp(1546839444, 6),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1546839444, 6),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.addShard("rs2/172.31.33.99:27018,172.31.45.208:27018,172.31.27.49:27018")
{
        "shardAdded" : "rs2",
        "ok" : 1,
        "operationTime" : Timestamp(1546839483, 5),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1546839483, 6),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

5.5) Enable sharding on the database and collection.

- Enable sharding for a database.

```
sh.enableSharding("mydb")
```

- Determine the shard key, create an index on the shard key and shard the collection.

```
use mydb
db.friend.createIndex( {"name":"hashed"} )
sh.shardCollection("mydb.friend", {"name":"hashed"})
```

Step 6. Load data from S3.

- configure ec2 using
  `aws configure`
- import using aws s3 cp and mongoimport (**should enable sharding on the db and collection first**)
  `aws s3 cp s3://bucketname/file - | mongoimport --db database_name(mydb) --collection collection_name(friend) --port 27017`

```
[ec2-user@ip-172-31-17-100 ~]$ aws s3 cp s3://usfca-msan694/world_bank_project.json - | mongoimport --db msds697 --collection world_bank_project --port 27017
2019-01-07T07:30:37.912+0000    connected to: localhost:27017
2019-01-07T07:30:38.812+0000    num failures: 15
: "http://www.worldbank.org/projects/P131763/first-programmatic-development-policy-loan?lang=en" } does not contain shard key for pattern { borrower: "hashed
" }

2019-01-07T07:30:38.814+0000    imported 485 documents
```

Reference :

1. Convert a Replica Set to a Sharded Cluster, MongoDB, https://docs.mongodb.com/manual/tutorial/convert-replica-set-to-replicated-shard-cluster/
2. Install MongoDB Community Edition on Amazon Linux, MongoDB, https://docs.mongodb.com/manual/tutorial/install-mongodb-on-amazon/
3. How to setup MongoDB Sharded Cluster with Replicasets on AWS, Cloud Buddy, https://m.youtube.com/watch?v=-_sLpMqVBZI