Take Home Midterm Exam

CS157C

Yulan Jin

1. Set up 3 nodes in AWS:
2. Create security groups:

mongodb-internal-access: groupID: sg-0e3508f23fe58a16c

shard server and config server can be accessed within this group, limit inboud source to this group

port number 27017-27019

1. Create three instances:

My IP address: IPv4: 75.3.240.208

Instance: descriptions: security groups:

for both mongodb-internal-access and default groups, edit inbound rules:

myIP: 75.3.240.208/32, port number 22, SSH

To convenience the process of setting up another two shard servers:

Create an image of this instance, as a source of creating other instances.

Instance – actions – create image. After creation, it is under image – AMIS.

Click that image – Launch – start configuration – Add tags: shard-server-2

Configure security group: default and mongodb-internal-access

1. Access instances through SSH
2. Shard-server-1:

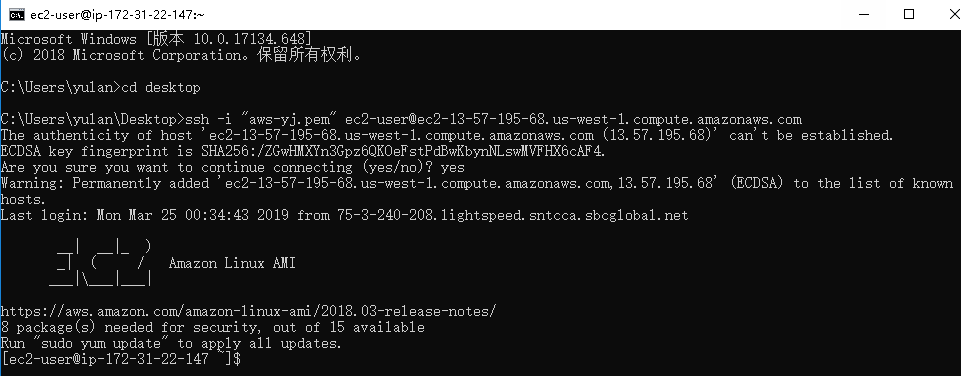
Find the directory that contains my key-pair “aws-yj.pem” in my laptop, and use

**ssh -i "aws-yj.pem" ec2-user@ec2-13-57-195-68.us-west-1.compute.amazonaws.com**

to connect to the instance.

IPv4 Public IP: 13.57.195.68

Private IPs: 172.31.22.147

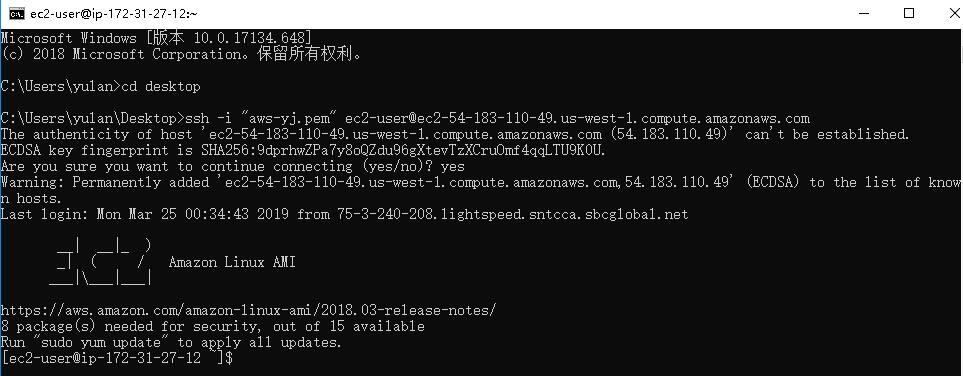


1. Shard-server-2:

**ssh -i "aws-yj.pem" ec2-user@ec2-54-183-110-49.us-west-1.compute.amazonaws.com**

IPv4 Public IP: 54.183.110.49

Private IPs: 172.31.27.12

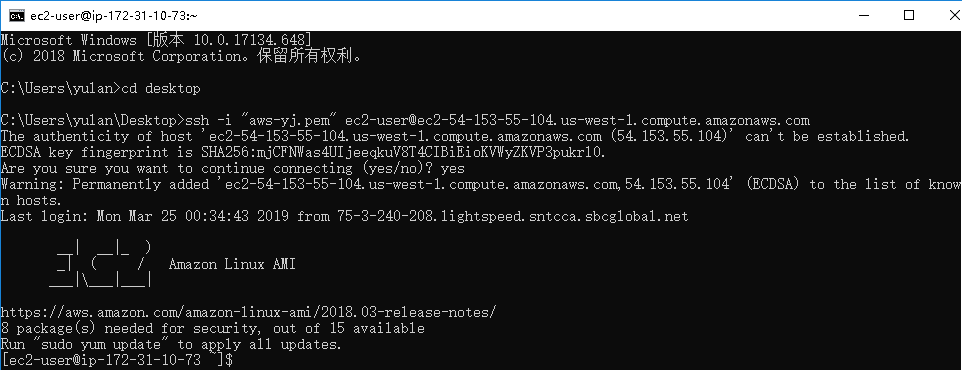


1. Shard-server-3:

ssh -i "aws-yj.pem" [ec2-user@ec2-13-57-244-8.us-west-1.compute.amazonaws.com](mailto:ec2-user@ec2-13-57-244-8.us-west-1.compute.amazonaws.com)

IPv4 Public IP: 54.153.55.104

Private IPs: 172.31.10.73



1. Install MongoDB in each node

Install mongodb on Amazon Linux:

1. Sudo nano /etc/yum.repos.d/mongodb-org-4.0.repo

Ctrol + O: write out on disk

1. Sudo yum -y update && sudo yum install -y mongodb-org

Sudo chkconfig mongod on

1. Create directory to store data in each node

vi /etc/mongod.conf

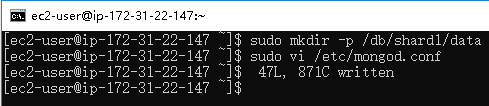
change port number to 27018

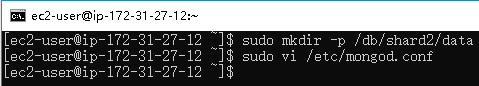
sharding:

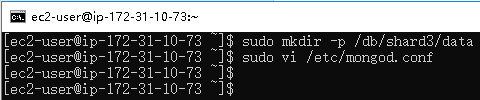
clusterRole: shardsvr

dbPath: /db/shard1/data

bindIP: change to its own private IP







1. Public and private IP of addresses of three AWS instances:
2. Shard-server-1:

IPv4 Public IP: 13.57.195.68

Private IPs: 172.31.22.147

1. Shard-server-2:

IPv4 Public IP: 54.183.110.49

Private IPs: 172.31.27.12

1. Shard-server-3:

IPv4 Public IP: 54.153.55.104

Private IPs: 172.31.10.73

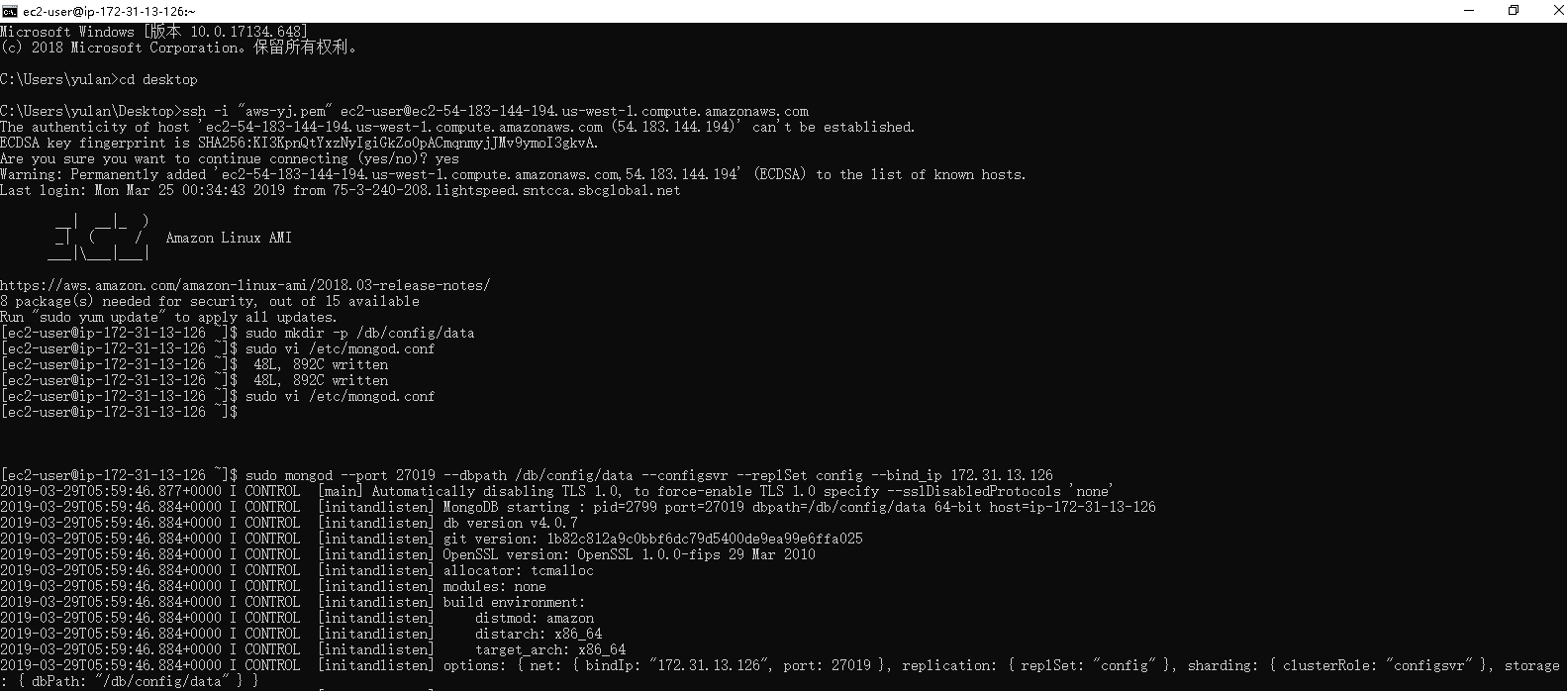
6. Set up and launch three config servers in a replica set.

config-server-1:

IPv4: Public IP: 54.183.144.194

Private IP: 172.31.13.126

**ssh -i "aws-yj.pem" ec2-user@ec2-54-183-144-194.us-west-1.compute.amazonaws.com**



sudo vi /etc/mongod.conf

change dbPath: /db/config/data

change bindIP to 172.31.10.236 (its own private IP)

change port number to 27019

replication:

replSetName: config

sharding:

clusterRole: configsvr

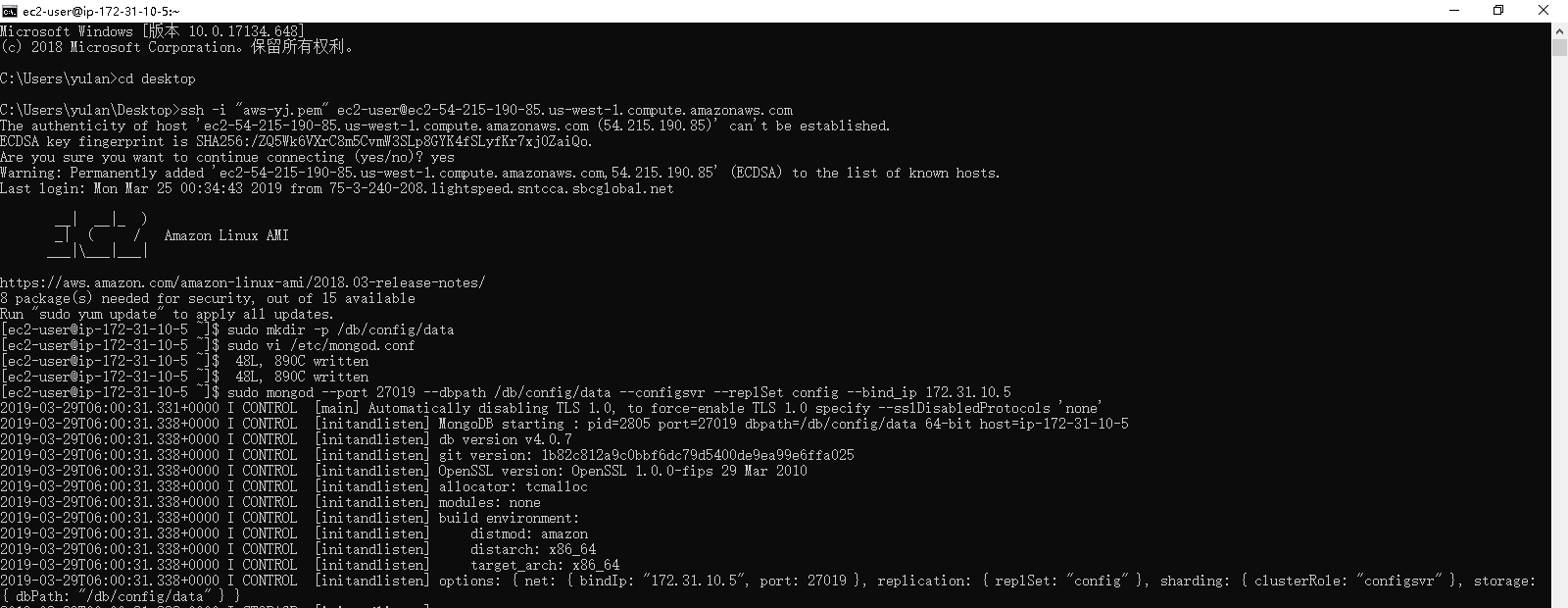
Then we set config-server-2 and config-server-3.

Config-server-2:

**ssh -i "aws-yj.pem" ec2-user@ec2-54-215-190-85.us-west-1.compute.amazonaws.com**

IPv4 Public IP: 54.215.190.85

Private IPs: 172.31.10.5

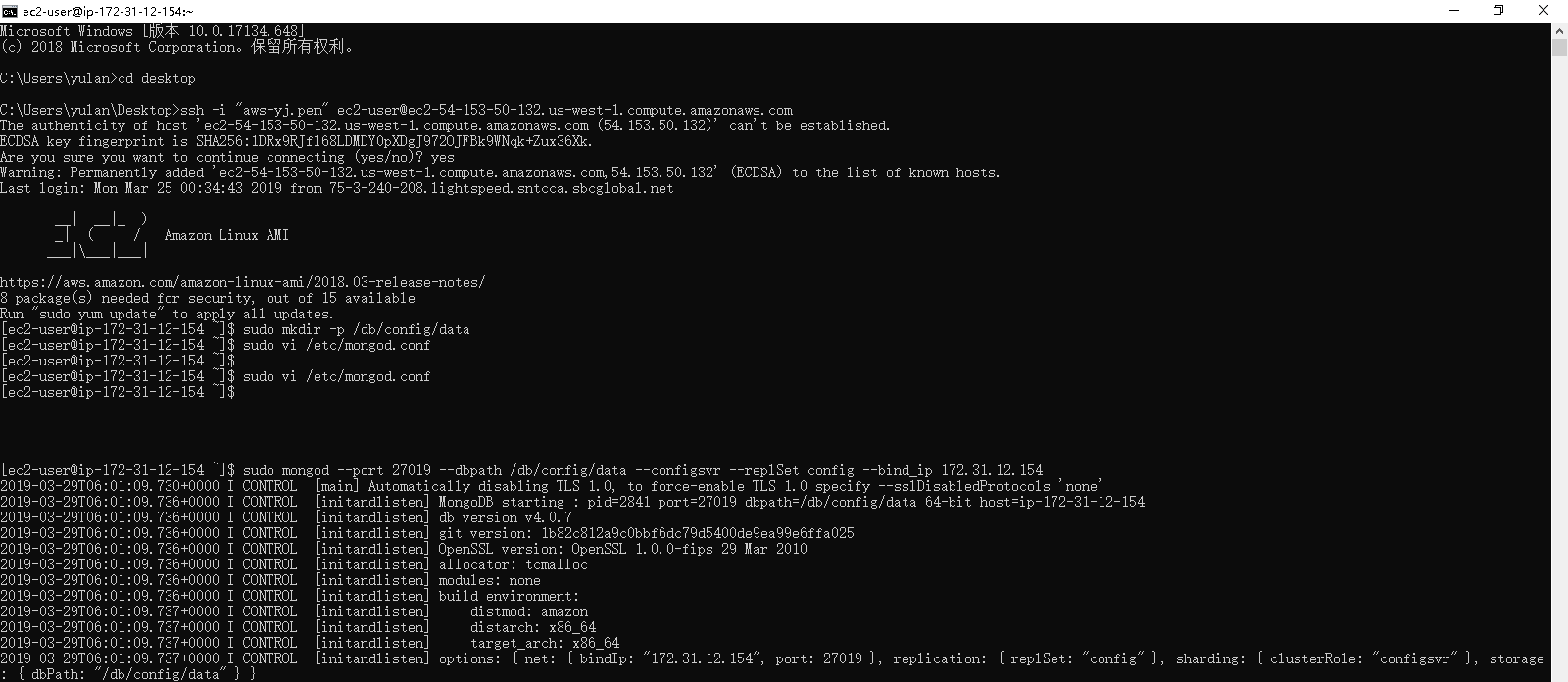


Config-server-3:

**ssh -i "aws-yj.pem" ec2-user@ec2-54-153-50-132.us-west-1.compute.amazonaws.com**

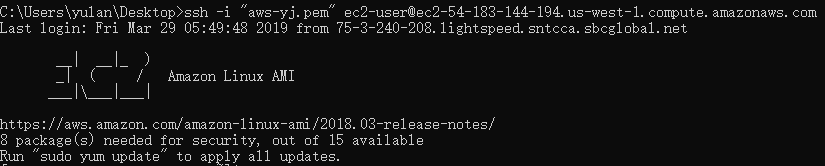
IPv4 Public IP: 54.153.50.132

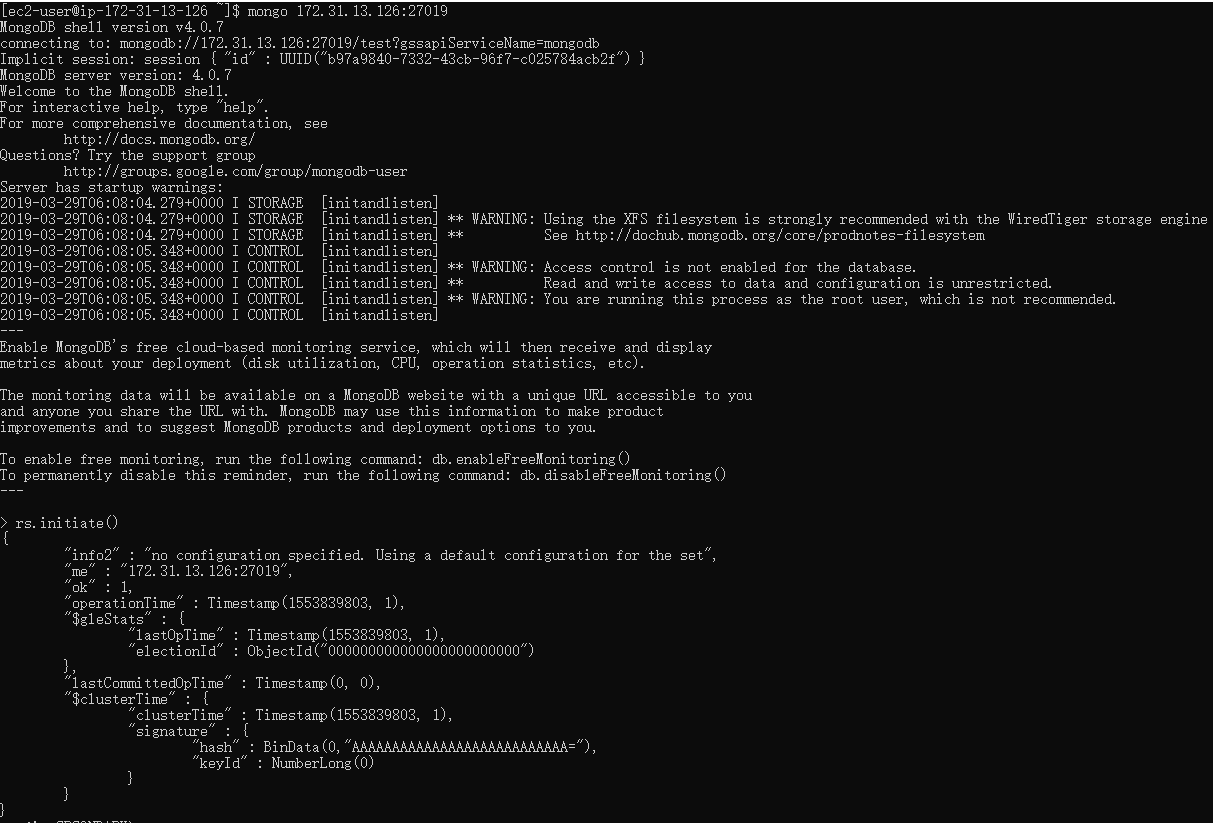
Private IPs: 172.31.12.154



Open a new temporary window, connect with the first config server

**ssh -i "aws-yj.pem" ec2-user@ec2-54-183-144-194.us-west-1.compute.amazonaws.com**











7. Connect mongos to each config server

mongos:

**ssh -i "aws-yj.pem" ec2-user@ec2-13-56-163-192.us-west-1.compute.amazonaws.com**

IPv4 Public IP: 13.56.163.192

Private IPs: 172.31.16.5

Comment out storage section for mongod.conf of mongos:

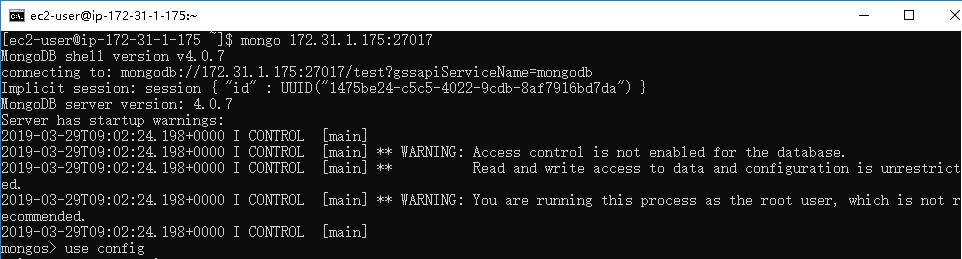
Sudo vi /etc/mongod.conf

Port: 27017

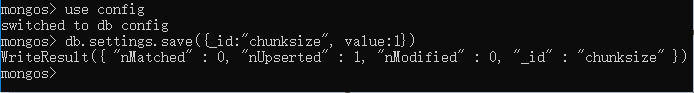
bindIP: 172.31.16.5

sharding:  
 configDB: config/172.31.13.126:27019,172.31.10.5:27019,172.31.12.154:27019

sudo mongos --config /etc/mongod.conf --fork --logpath /var/log/mongodb/mongod.log

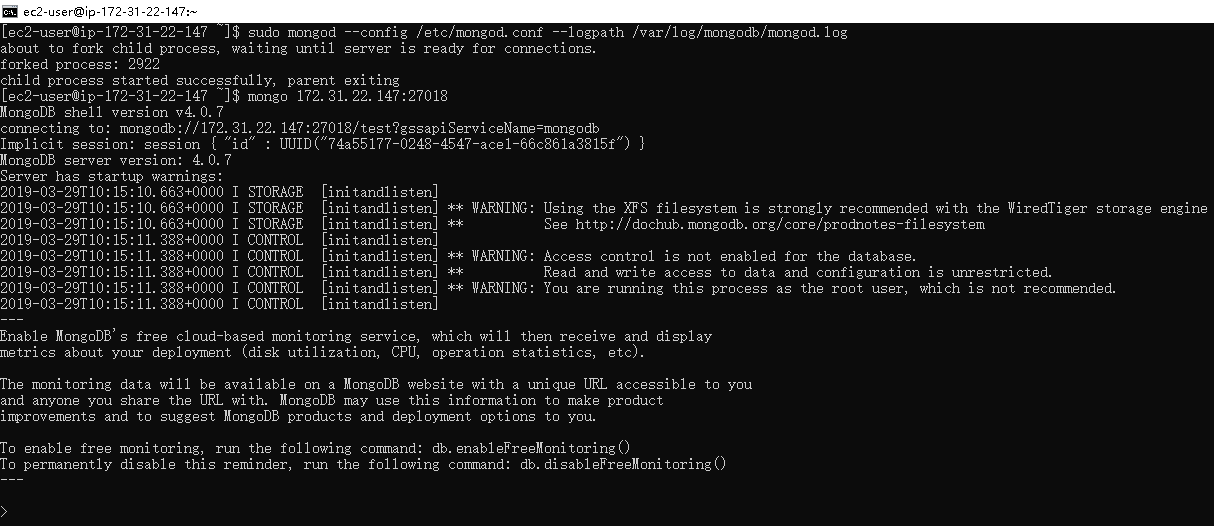


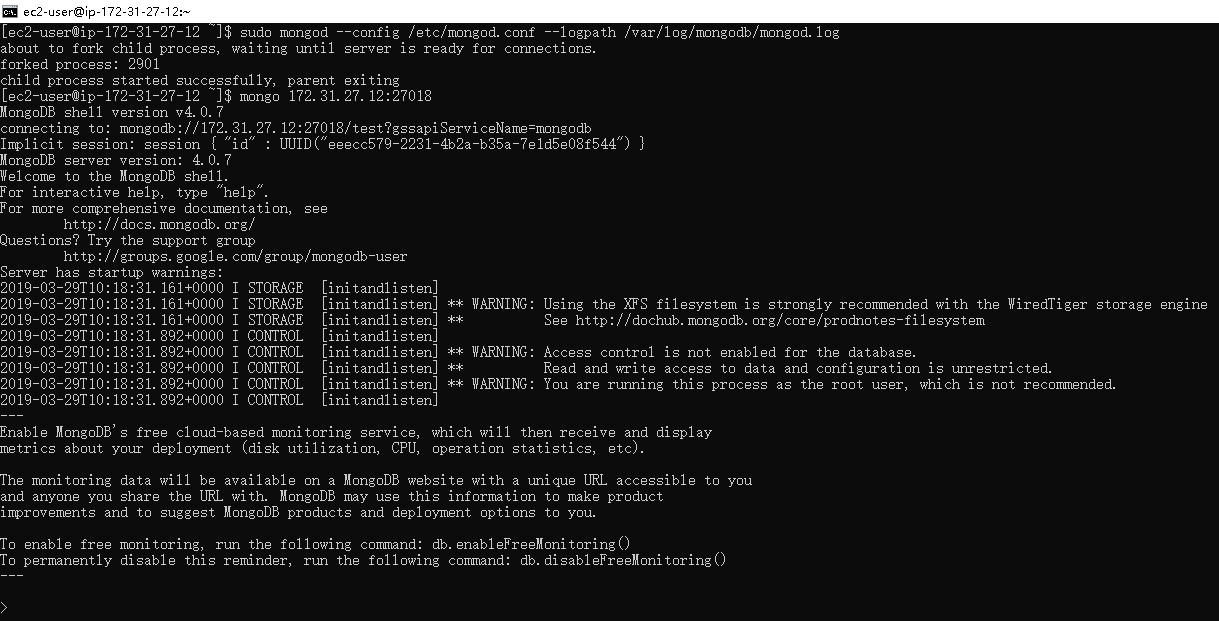
Set chunksize to 1 inmongos:

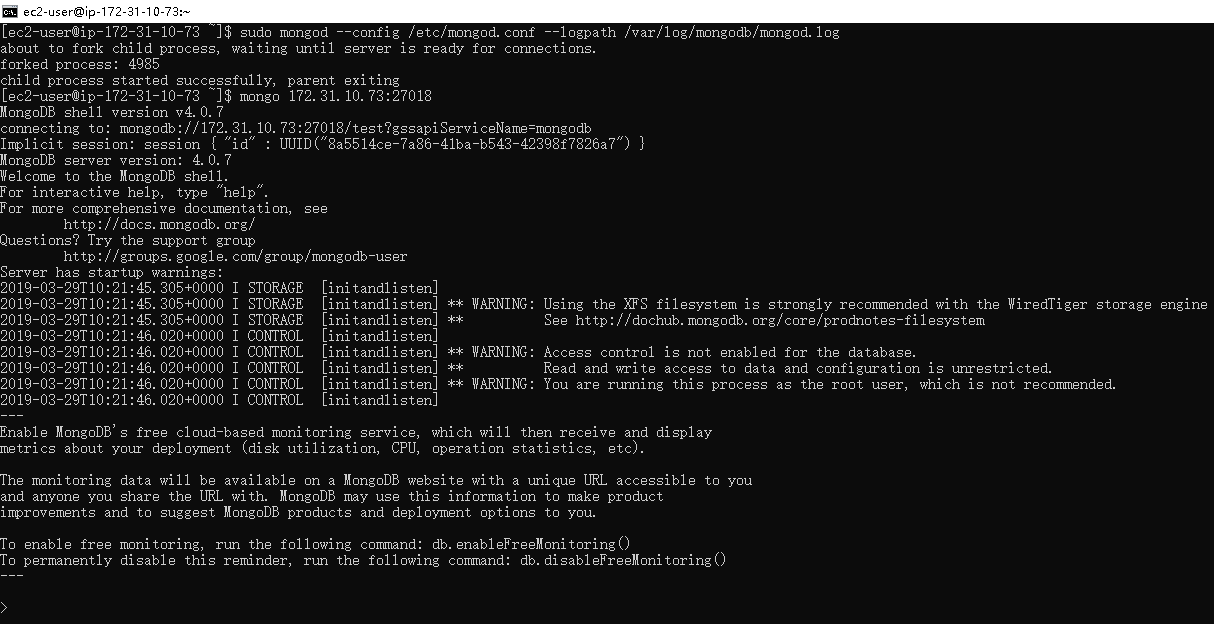


8.

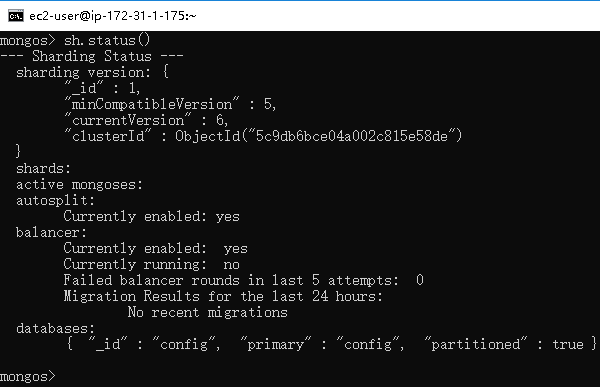
Set up and launch each of three shards: make sure to include the result of sh.status() before adding shards which will be done in the next task.







In mongos, run sh.status():



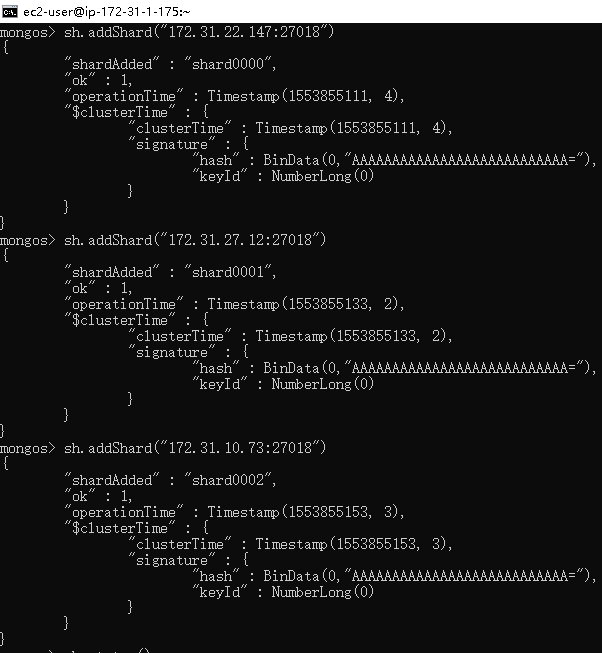
9. Add shards: make sure to include the result of sh.status() after adding shards

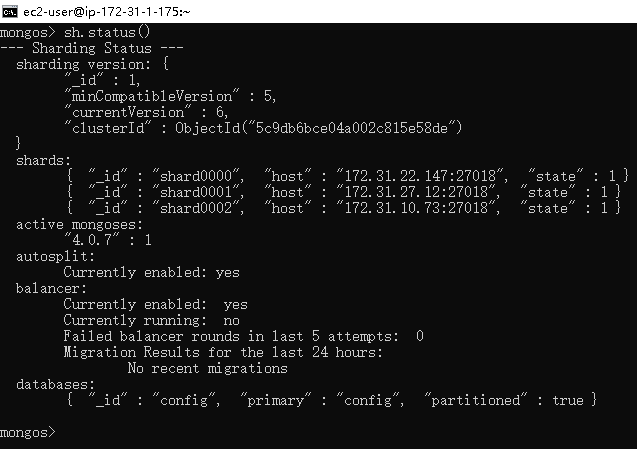
sh.addShard(“172.31.15.68:27018”)

sh.addShard(“172.31.15.221:27018”)

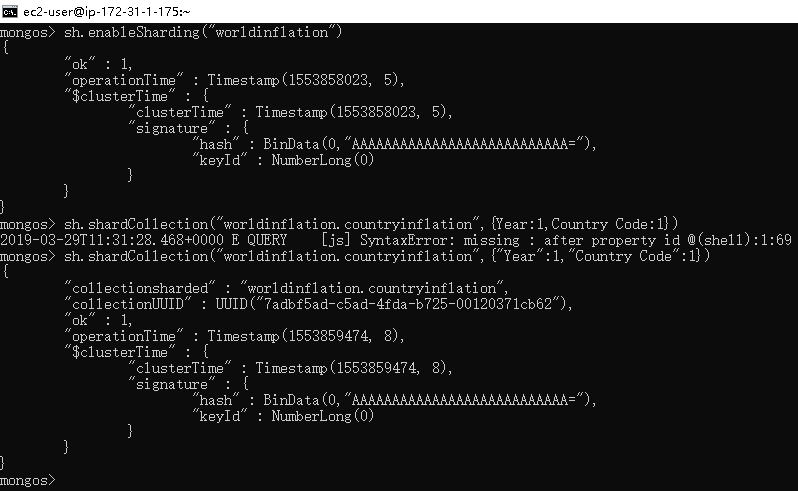
sh.addShard(“172.31.15.251:27018”)

In mongos:





10. (5points) Enable shards: explain the nature of the shard key (ascending, random, or location based) and the sharding strategy (range-based or hash based)



The index is a compound key of (“Year”, “Country Code”). It satisfies large shard key cardinality, low shard key frequency, and non-monotonically changing shard keys. Therefore, it uses range-based sharding strategy. Note this, the year decides the main order, in the original database, although for a given country code, years of documents are in increasing order, the whole order of years is not monotonically increasing. I choose year to mainly decide the order of documents to facilitate queries.

Then, since the documents are not ordered in the increasing order of years, I will say the nature of the shard key is random. Not all newly added documents will be in the max chunk.

11. Populate data in the cluster using a public data set : Explain your collection and include the code to populate data and sh.status() after populating data. Specify the URL to the dataset.

<https://pkgstore.datahub.io/core/inflation/inflation-consumer_json/data/f928d7e34e8287e66ae42f280d628efc/inflation-consumer_json.json>

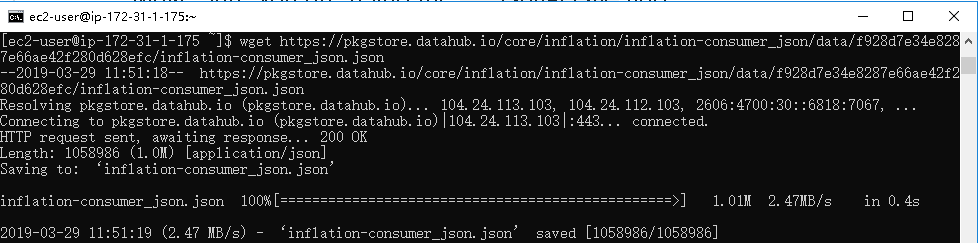
Link to the json file of the dataset

<https://datahub.io/core/inflation>

Link to topics of the collection

The collection contains documents containing fields: country, country code, inflation, and year.

Download data:



Import data:

