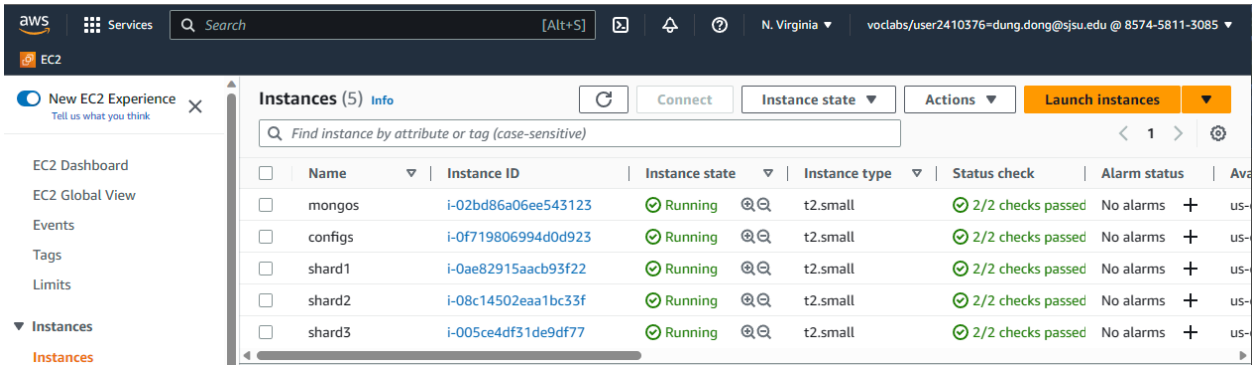
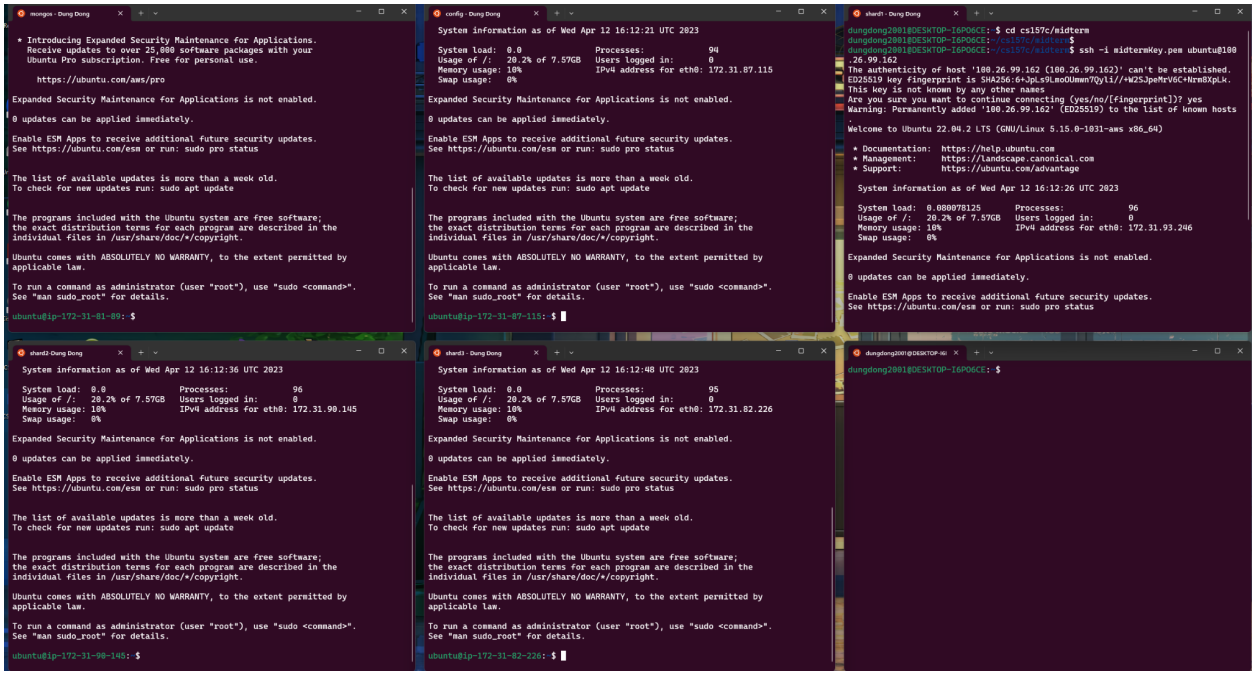
1. Create 5 EC2 instances by clicking Launch Instance → Create RSA key (midtermKey.pem) → Launch Instance. Set Inbound / Outbound traffic to All traffic.
    a. Mongos:
    b. Configs:
    c. Shard1:
    d. Shard2:
    e. Shard3:

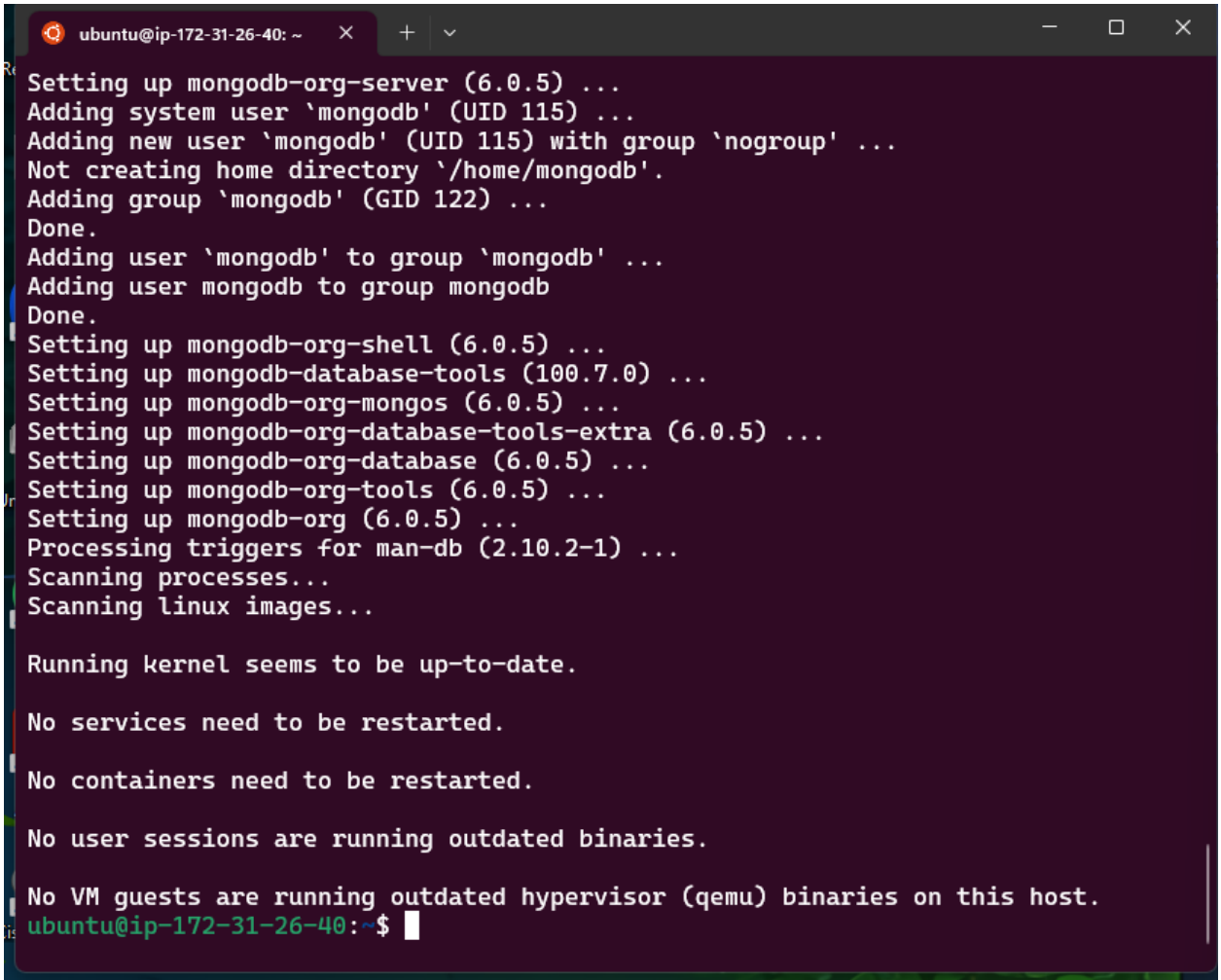


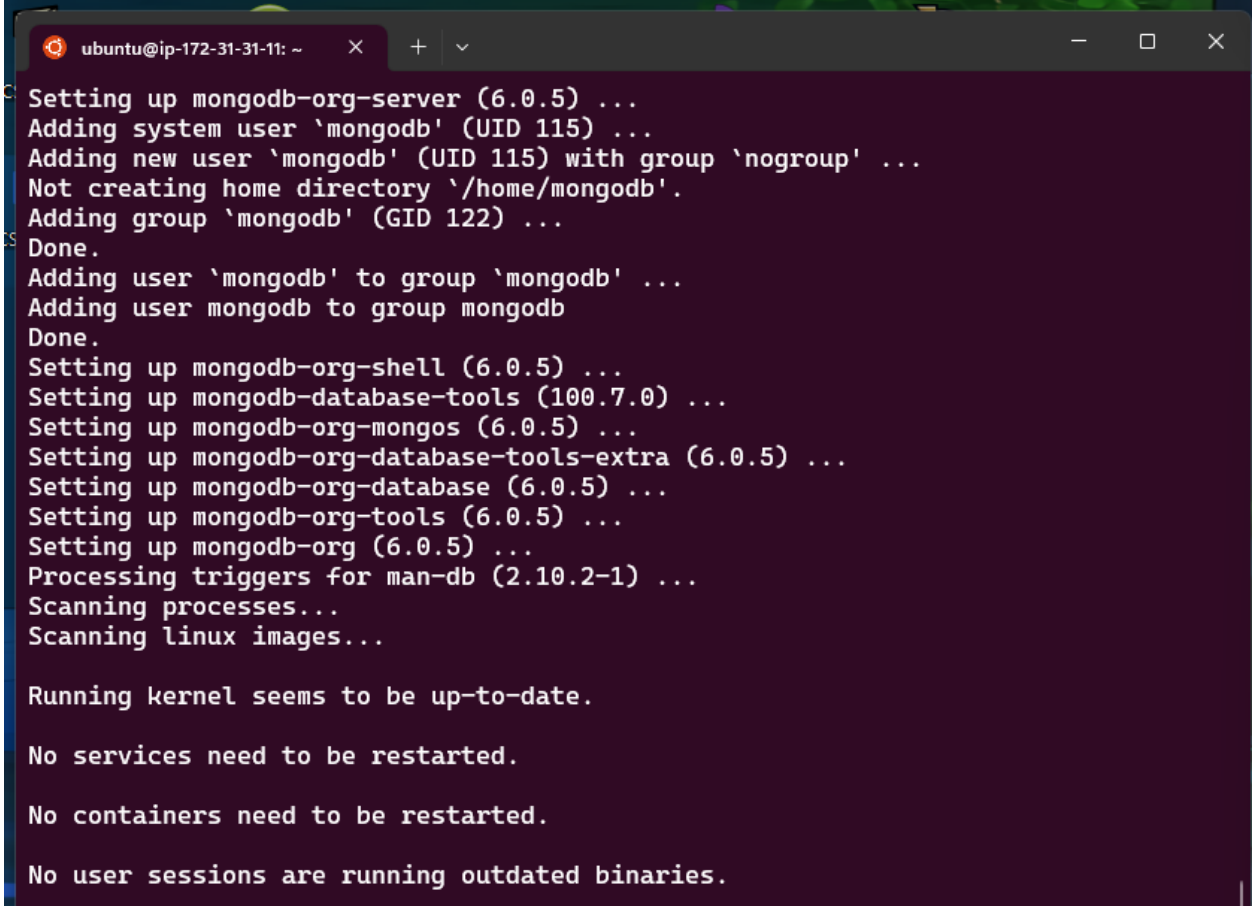
2. Give permissions to the key files created in step 1: chmod 600 midtermKey.pem
Access to servers: ssh -i midtermKey.pem ubuntu@<Public IPv4 address>

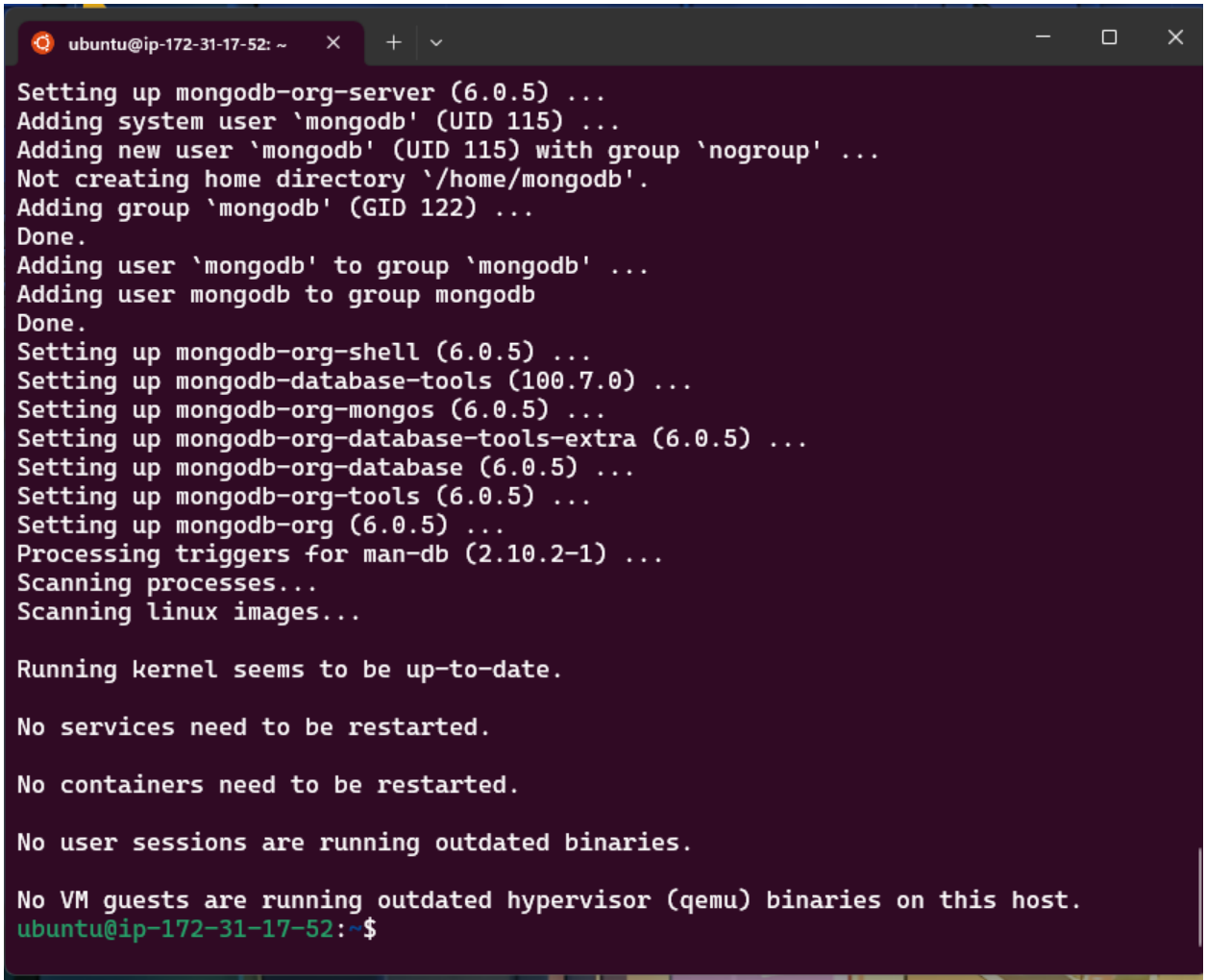


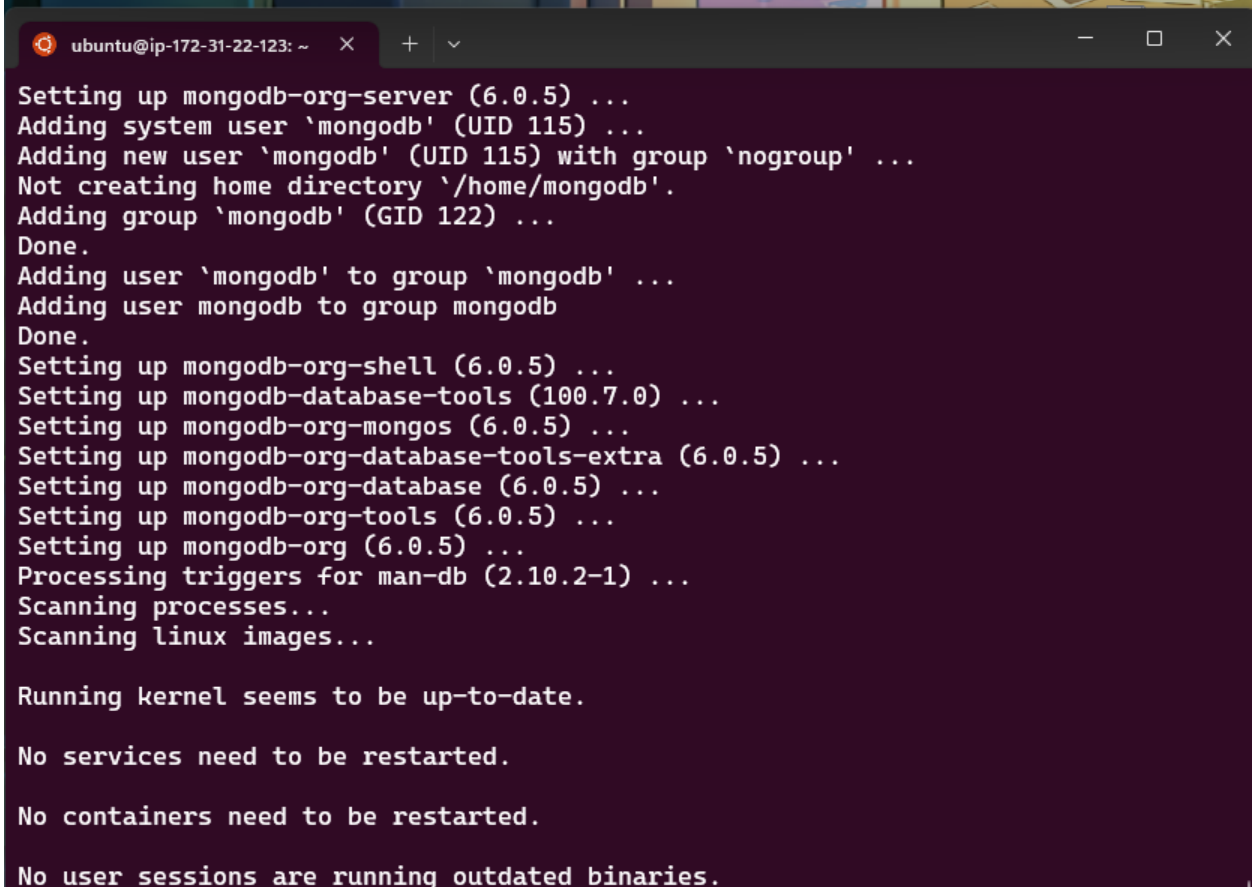
3. Follow steps on [Install MongoDB Community Edition on Ubuntu — MongoDB Manual] to install mongoDB
However, the first step gets an error saying that “gpg: WARNING: no command supplied. Trying to guess what you mean …”. I checked the key-list and it signed for MongoDB 5.0 not 6.0.

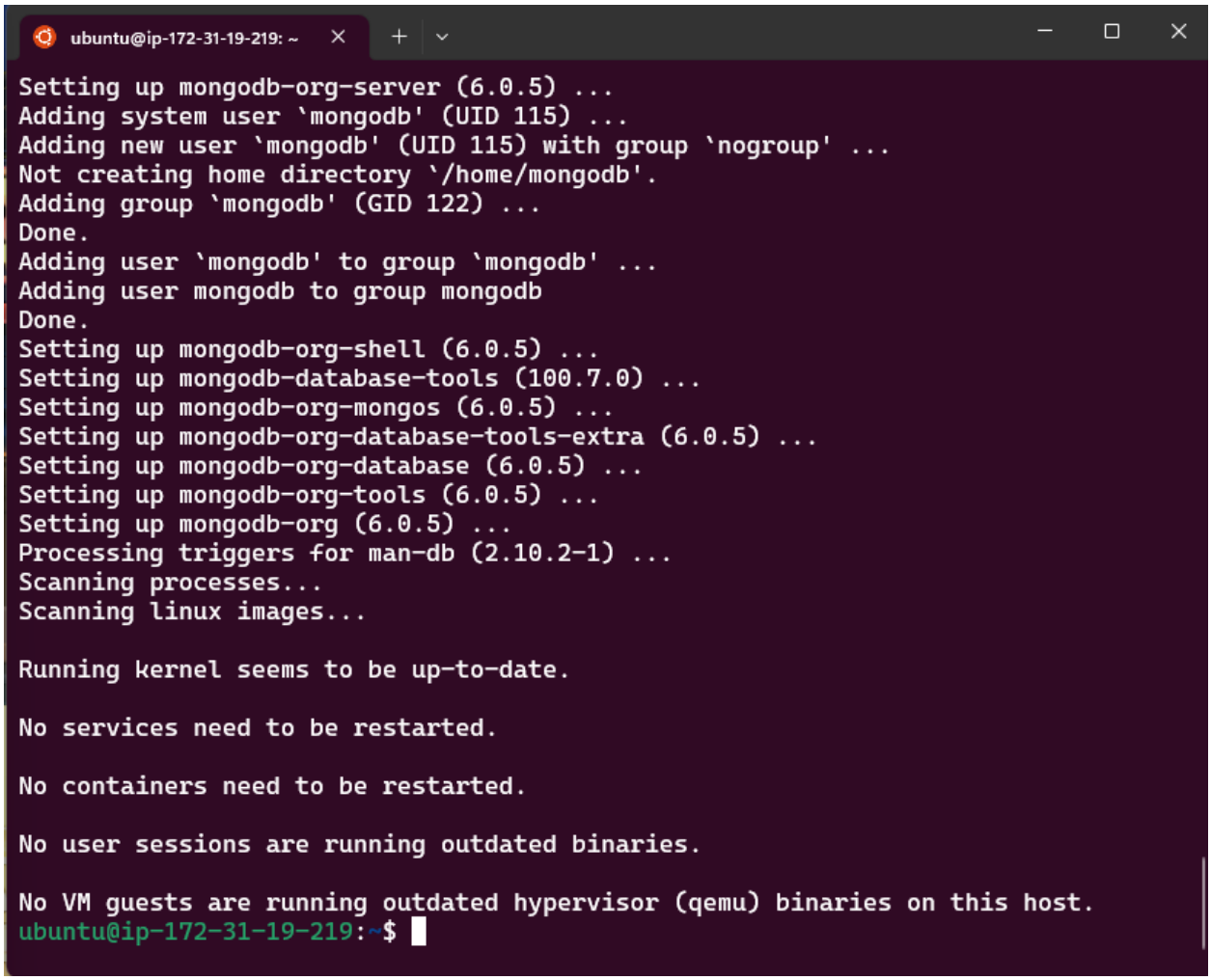
Thus, I had to replace it with “curl -sS https://pgp.mongodb.com/server-6.0.asc | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/mongodb-6.0.gpg” for the first step.


ubuntu@ip-172-31-26-40: ~
Setting up mongodb-org-server (6.0.5) ...
Adding system user `mongodb' (UID 115) ...
Adding new user `mongodb' (UID 115) with group `nogroup' ...
Not creating home directory `/home/mongodb'.
Adding group `mongodb' (GID 122) ...
Done.
Adding user `mongodb' to group `mongodb' ...
Adding user mongodb to group mongodb
Done.
Setting up mongodb-org-shell (6.0.5) ...
Setting up mongodb-database-tools (100.7.0) ...
Setting up mongodb-org-mongos (6.0.5) ...
Setting up mongodb-org-database-tools-extra (6.0.5) ...
Setting up mongodb-org-database (6.0.5) ...
Setting up mongodb-org-tools (6.0.5) ...
Setting up mongodb-org (6.0.5) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-26-40:~$


ubuntu@ip-172-31-31-11: ~
Setting up mongodb-org-server (6.0.5) ...
Adding system user `mongodb' (UID 115) ...
Adding new user `mongodb' (UID 115) with group `nogroup' ...
Not creating home directory `/home/mongodb'.
Adding group `mongodb' (GID 122) ...
Done.
Adding user `mongodb' to group `mongodb' ...
Adding user mongodb to group mongodb
Done.
Setting up mongodb-org-shell (6.0.5) ...
Setting up mongodb-database-tools (100.7.0) ...
Setting up mongodb-org-mongos (6.0.5) ...
Setting up mongodb-org-database-tools-extra (6.0.5) ...
Setting up mongodb-org-database (6.0.5) ...
Setting up mongodb-org-tools (6.0.5) ...
Setting up mongodb-org (6.0.5) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

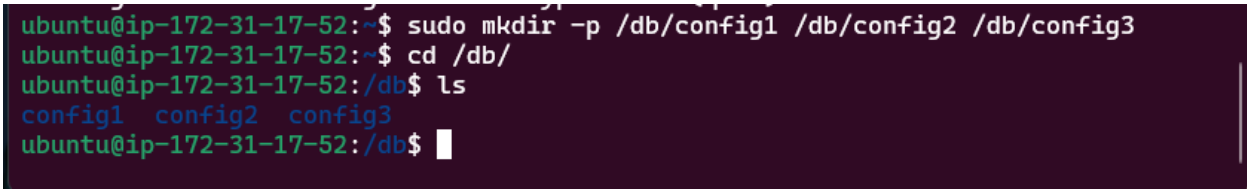

ubuntu@ip-172-31-17-52: ~
Setting up mongodb-org-server (6.0.5) ...
Adding system user `mongodb' (UID 115) ...
Adding new user `mongodb' (UID 115) with group `nogroup' ...
Not creating home directory `/home/mongodb'.
Adding group `mongodb' (GID 122) ...
Done.
Adding user `mongodb' to group `mongodb' ...
Adding user mongodb to group mongodb
Done.
Setting up mongodb-org-shell (6.0.5) ...
Setting up mongodb-database-tools (100.7.0) ...
Setting up mongodb-org-mongos (6.0.5) ...
Setting up mongodb-org-database-tools-extra (6.0.5) ...
Setting up mongodb-org-database (6.0.5) ...
Setting up mongodb-org-tools (6.0.5) ...
Setting up mongodb-org (6.0.5) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-17-52:~$


ubuntu@ip-172-31-22-123: ~
Setting up mongodb-org-server (6.0.5) ...
Adding system user `mongodb' (UID 115) ...
Adding new user `mongodb' (UID 115) with group `nogroup' ...
Not creating home directory `/home/mongodb'.
Adding group `mongodb' (GID 122) ...
Done.
Adding user `mongodb' to group `mongodb' ...
Adding user mongodb to group mongodb
Done.
Setting up mongodb-org-shell (6.0.5) ...
Setting up mongodb-database-tools (100.7.0) ...
Setting up mongodb-org-mongos (6.0.5) ...
Setting up mongodb-org-database-tools-extra (6.0.5) ...
Setting up mongodb-org-database (6.0.5) ...
Setting up mongodb-org-tools (6.0.5) ...
Setting up mongodb-org (6.0.5) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

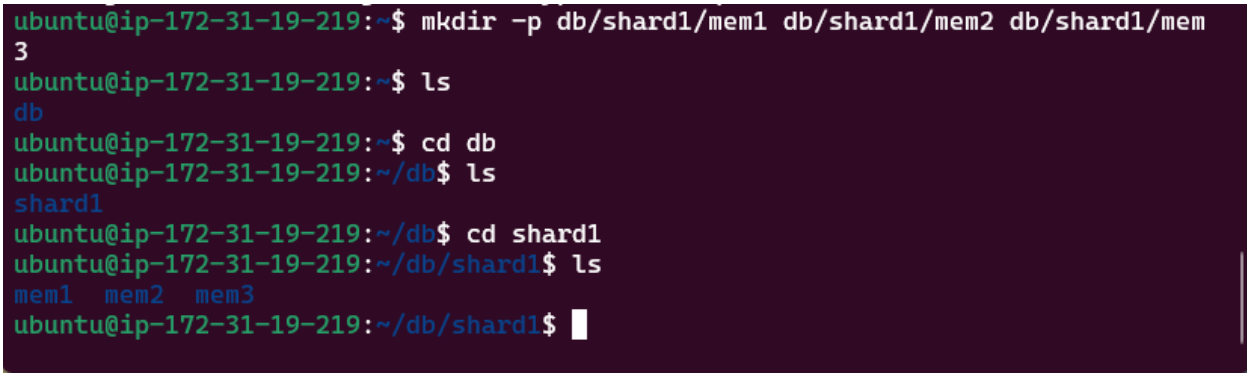


4.

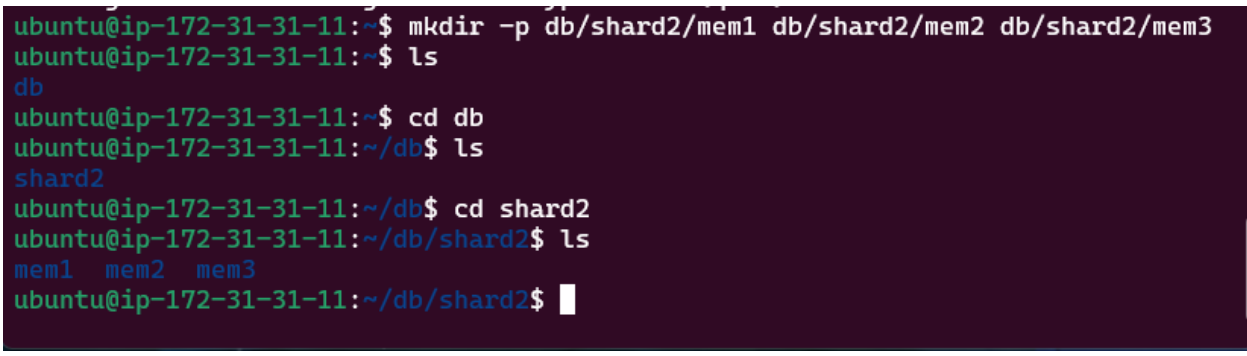
a. Config: sudo mkdir -p /db/config1 /db/config2 /db/config3



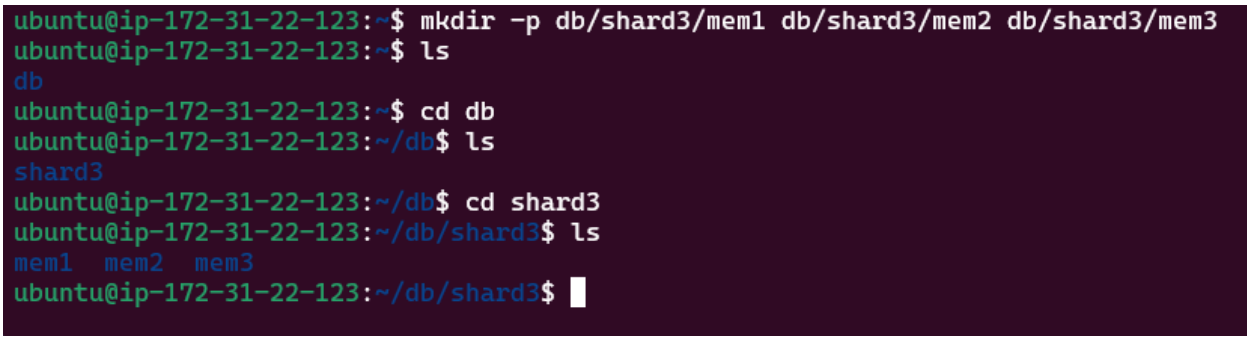
b. Shard1: mkdir -p db/shard1/mem1 db/shard1/mem2 db/shard1/mem3



c. Shard2: mkdir -p db/shard2/mem1 db/shard2/mem2 db/shard2/mem3



d. Shard3: mkdir -p db/shard3/mem1 db/shard3/mem2 db/shard3/mem3



5.

a. Mongos:
   i. Public IPv4 address: 54.226.173.1
   ii. Private IPv4 address: 172.31.26.40
   iii. DNS: ec2-54-226-173-1.compute-1.amazonaws.com

b. Configs:
   i. Public IPv4 address: 3.94.196.162
   ii. Private IPv4 address: 172.31.17.52
   iii. DNS: ec2-3-94-196-162.compute-1.amazonaws.com

c. Shard1:
   i. Public IPv4 address: 34.230.29.212
   ii. Private IPv4 address: 172.31.19.219

iii. DNS: ec2-34-230-29-212.compute-1.amazonaws.com

d. Shard2:

i. Public IPv4 address: 34.226.213.111

ii. Private IPv4 address: 172.31.31.11

iii. DNS: ec2-34-226-213-111.compute-1.amazonaws.com

e. Shard3:

i. Public IPv4 address: 54.226.251.91

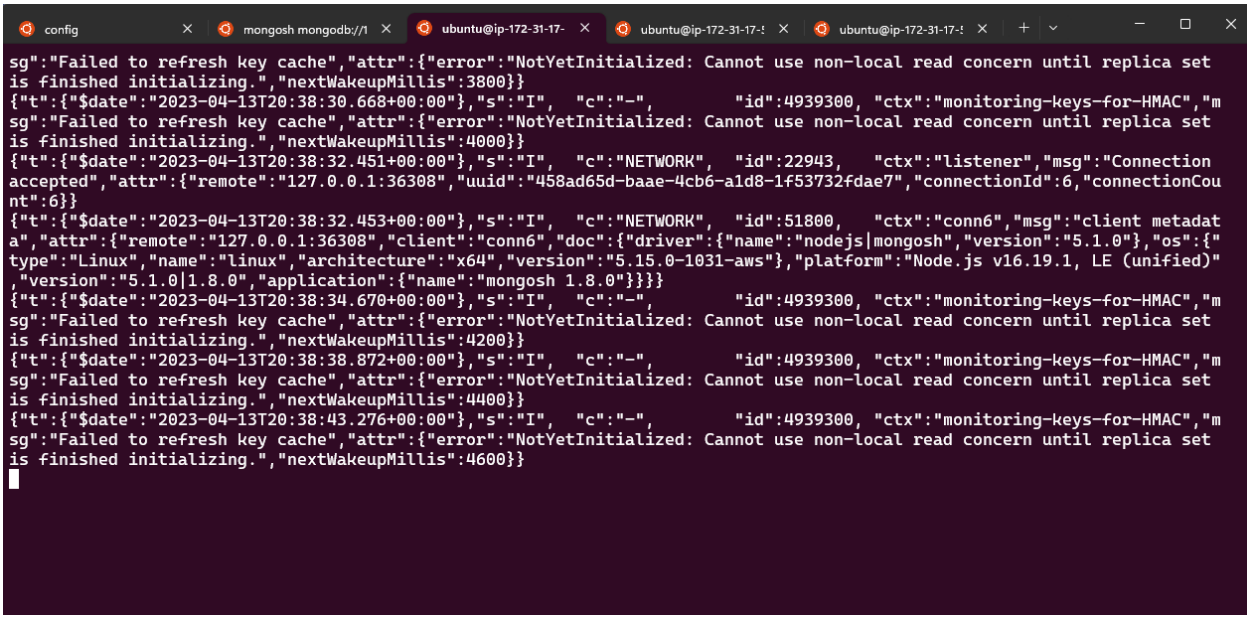
ii. Private IPv4 address: 172.31.22.123

iii. DNS: ec2-54-226-251-91.compute-1.amazonaws.com

6.

a. Config 1:

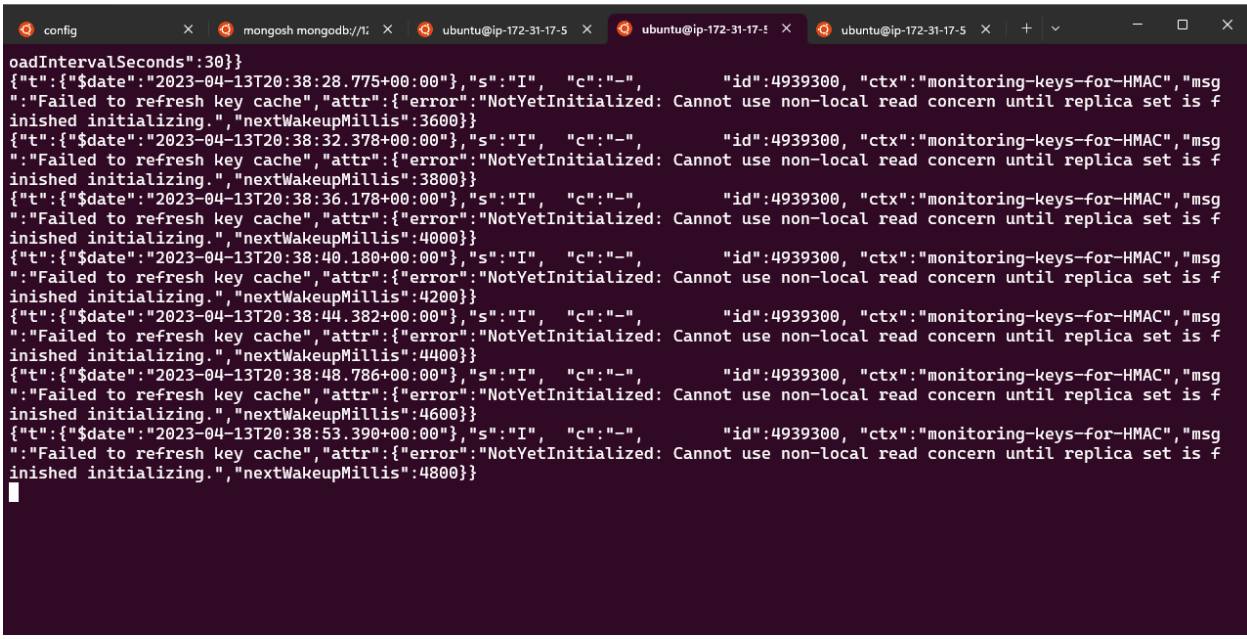
sudo mongod --port 27031 --configsvr --replSet config --dbpath /db/config1 --bind_ip localhost,ec2-107-22-129-133.compute-1.amazonaws.com



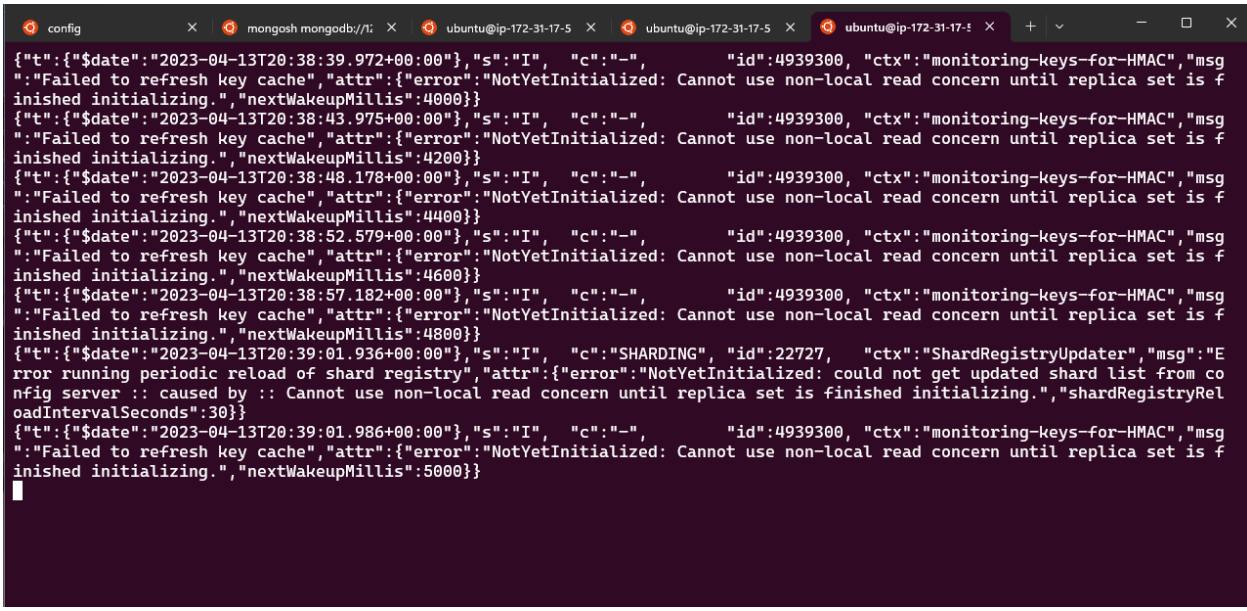
b. Config 2:

sudo mongod --port 27032 --configsvr --replSet config --dbpath /db/config2 --bind_ip

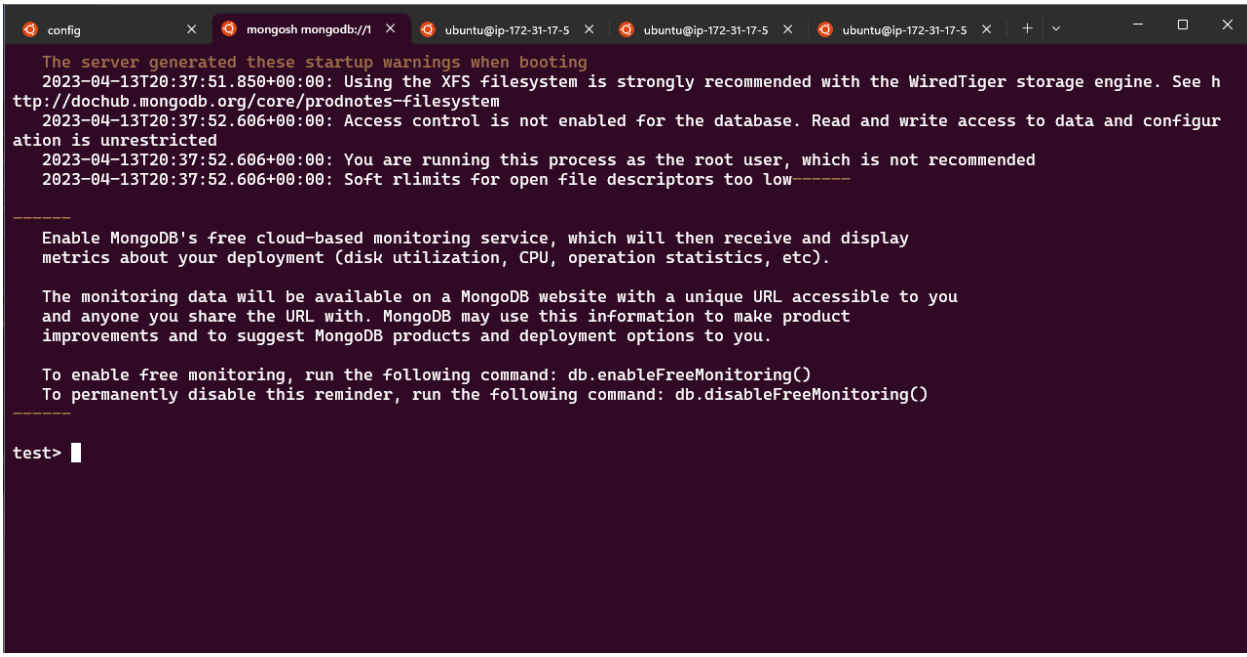
localhost,ec2-107-22-129-133.compute-1.amazonaws.com



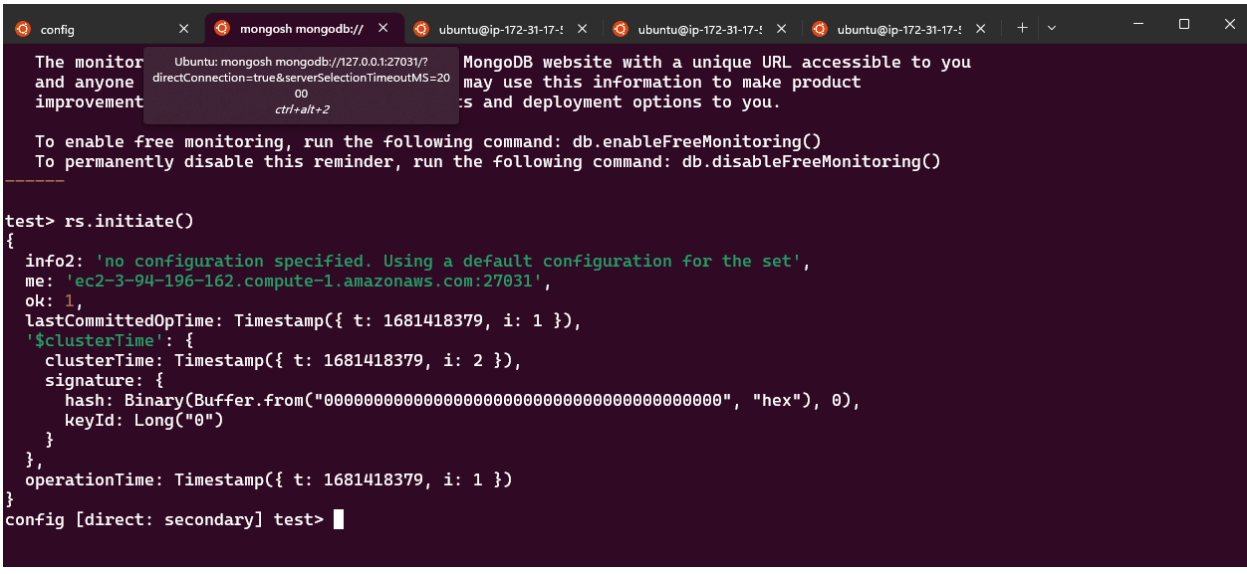
c. Config 3:
sudo mongod --port 27033 --configsvr --replSet config --dbpath /db/config3 --bind_ip localhost,ec2-107-22-129-133.compute-1.amazonaws.com



d. Connect to one of the config servers:
mongosh --port 27031



e. Initiate the replica set:
rs.initiate()

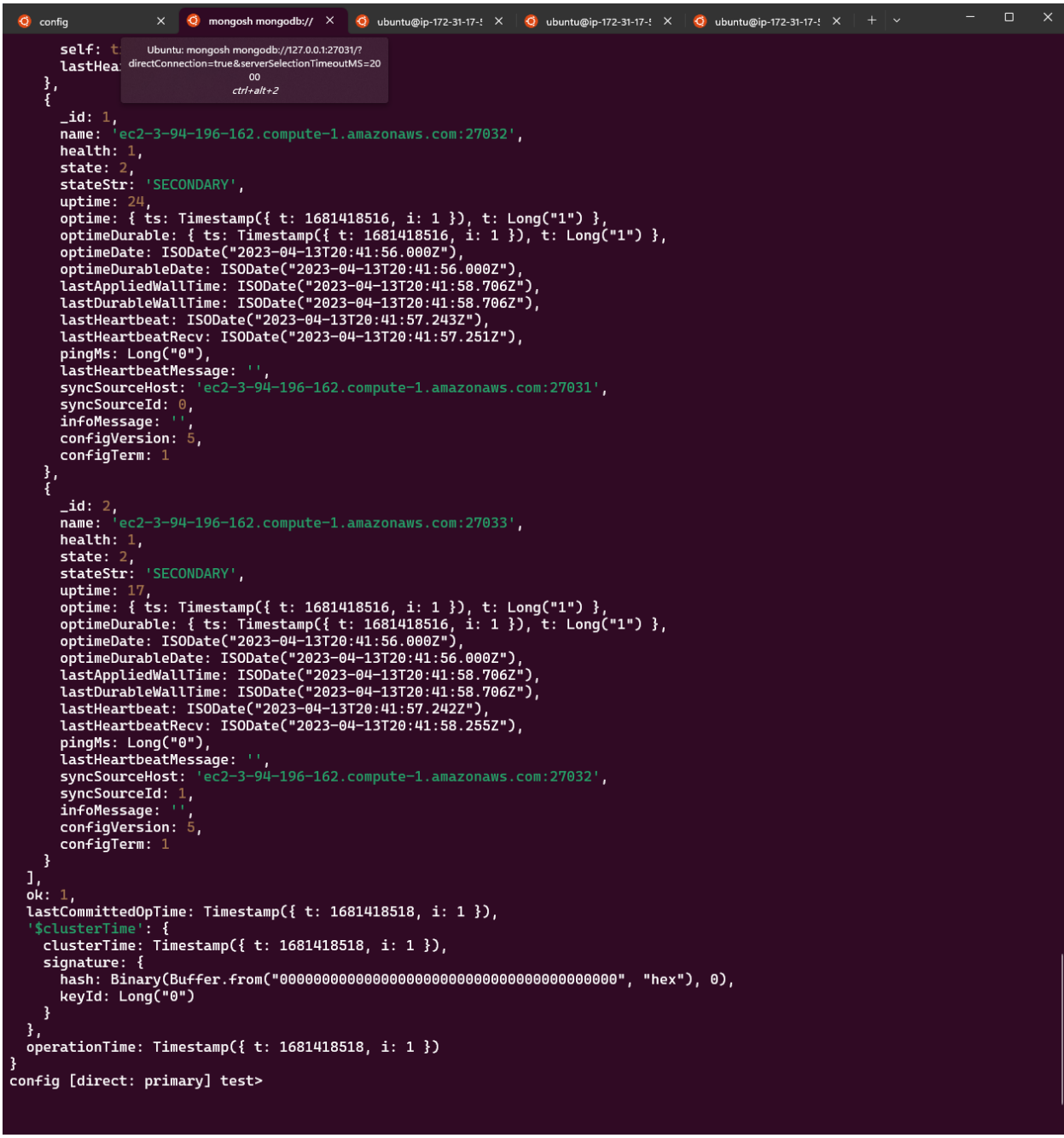


f. Check status:
rs.status()


config
mongosh mongodb://
ubuntu@ip-172-31-17-
ubuntu@ip-172-31-17-
ubuntu@ip-172-31-17-
Ubuntu: mongosh mongodb://127.0.0.1:27031/?directConnection=true&serverSelectionTimeoutMS=2000
ctrl+alt+2
config [direct:
{
set: 'config',
date: ISODate(
myState: 1,
term: Long("1"),
syncSourceHost: '',
syncSourceId: -1,
configsvr: true,
heartbeatIntervalMillis: Long("2000"),
majorityVoteCount: 1,
writeMajorityCount: 1,
votingMembersCount: 1,
writableVotingMembersCount: 1,
optimes: {
lastCommittedOpTime: { ts: Timestamp({ t: 1681418435, i: 1 }), t: Long("1") },
lastCommittedWallTime: ISODate("2023-04-13T20:40:35.682Z"),
readConcernMajorityOpTime: { ts: Timestamp({ t: 1681418435, i: 1 }), t: Long("1") },
appliedOpTime: { ts: Timestamp({ t: 1681418435, i: 1 }), t: Long("1") },
durableOpTime: { ts: Timestamp({ t: 1681418435, i: 1 }), t: Long("1") },
lastAppliedWallTime: ISODate("2023-04-13T20:40:35.682Z"),
lastDurableWallTime: ISODate("2023-04-13T20:40:35.682Z")
},
lastStableRecoveryTimestamp: Timestamp({ t: 1681418379, i: 1 }),
electionCandidateMetrics: {
lastElectionReason: 'electionTimeout',
lastElectionDate: ISODate("2023-04-13T20:39:39.404Z"),
electionTerm: Long("1"),
lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1681418379, i: 1 }), t: Long("-1") },
lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1681418379, i: 1 }), t: Long("-1") },
numVotesNeeded: 1,
priorityAtElection: 1,
electionTimeoutMillis: Long("10000"),
newTermStartDate: ISODate("2023-04-13T20:39:39.458Z"),
wMajorityWriteAvailabilityDate: ISODate("2023-04-13T20:39:39.664Z")
},
members: [
{
_id: 0,
name: 'ec2-3-94-196-162.compute-1.amazonaws.com:27031',
health: 1,
state: 1,
stateStr: 'PRIMARY',
uptime: 165,
optime: { ts: Timestamp({ t: 1681418435, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T20:40:35.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T20:40:35.682Z"),
lastDurableWallTime: ISODate("2023-04-13T20:40:35.682Z"),
syncSourceHost: '',
syncSourceId: -1,
infoMessage: '',
electionTime: Timestamp({ t: 1681418379, i: 2 }),
electionDate: ISODate("2023-04-13T20:39:39.000Z"),
configVersion: 1,
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
}
],
ok: 1,
lastCommittedOpTime: Timestamp({ t: 1681418435, i: 1 }),
'$clusterTime': {
clusterTime: Timestamp({ t: 1681418435, i: 1 }),
signature: {
hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex"), 0),
keyId: Long("0")
}
},
operationTime: Timestamp({ t: 1681418435, i: 1 })
}
config [direct: primary] test>

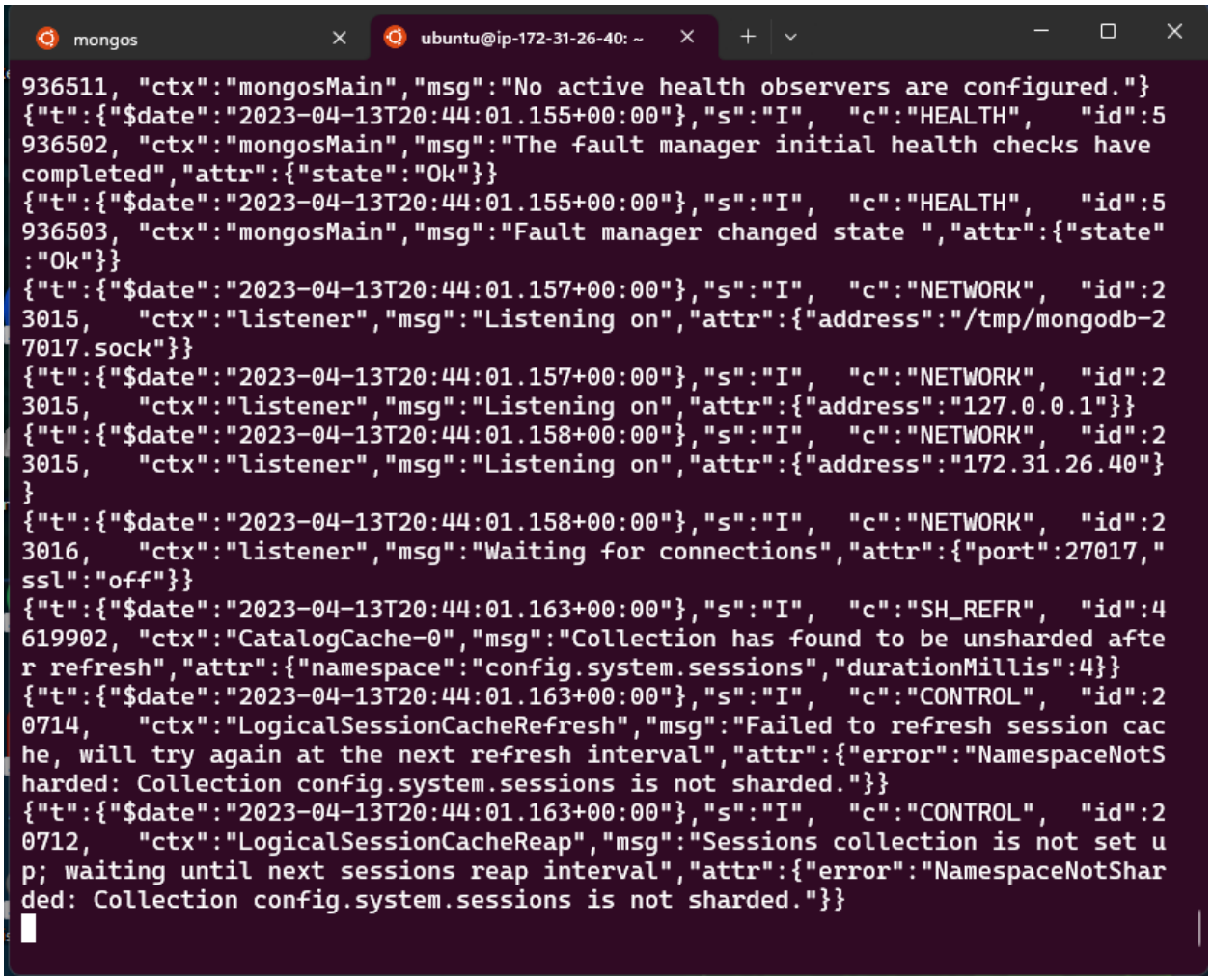
g. Add other members to the set and check for status again:
rs.add("ec2-3-86-247-48.compute-1.amazonaws.com:27032")
rs.add("ec2-3-86-247-48.compute-1.amazonaws.com:27033")
rs.status()


config
mongosh mongodb://
ubuntu@ip-172-31-17-
ubuntu@ip-172-31-17-
ubuntu@ip-172-31-17-
Ubuntu: mongosh mongodb://127.0.0.1:27031/?directConnection=true&serverSelectionTimeoutMS=2000
ctrl+alt+2
operationTime
1 })
}
config [direct:
{
set: 'config',
date: ISODate("2023-04-13T20:41:58.914Z"),
myState: 1,
term: Long("1"),
syncSourceHost: '',
syncSourceId: -1,
configsvr: true,
heartbeatIntervalMillis: Long("2000"),
majorityVoteCount: 2,
writeMajorityCount: 2,
votingMembersCount: 3,
writableVotingMembersCount: 3,
optimes: {
lastCommittedOpTime: { ts: Timestamp({ t: 1681418518, i: 1 }), t: Long("1") },
lastCommittedWallTime: ISODate("2023-04-13T20:41:58.706Z"),
readConcernMajorityOpTime: { ts: Timestamp({ t: 1681418518, i: 1 }), t: Long("1") },
appliedOpTime: { ts: Timestamp({ t: 1681418518, i: 1 }), t: Long("1") },
durableOpTime: { ts: Timestamp({ t: 1681418518, i: 1 }), t: Long("1") },
lastAppliedWallTime: ISODate("2023-04-13T20:41:58.706Z"),
lastDurableWallTime: ISODate("2023-04-13T20:41:58.706Z")
},
lastStableRecoveryTimestamp: Timestamp({ t: 1681418497, i: 2 }),
electionCandidateMetrics: {
lastElectionReason: 'electionTimeout',
lastElectionDate: ISODate("2023-04-13T20:39:39.404Z"),
electionTerm: Long("1"),
lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1681418379, i: 1 }), t: Long("-1") },
lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1681418379, i: 1 }), t: Long("-1") },
numVotesNeeded: 1,
priorityAtElection: 1,
electionTimeoutMillis: Long("10000"),
newTermStartDate: ISODate("2023-04-13T20:39:39.458Z"),
wMajorityWriteAvailabilityDate: ISODate("2023-04-13T20:39:39.664Z")
},
members: [
{
_id: 0,
name: 'ec2-3-94-196-162.compute-1.amazonaws.com:27031',
health: 1,
state: 1,
stateStr: 'PRIMARY',
uptime: 247,
optime: { ts: Timestamp({ t: 1681418518, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T20:41:58.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T20:41:58.706Z"),
lastDurableWallTime: ISODate("2023-04-13T20:41:58.706Z"),
syncSourceHost: '',
syncSourceId: -1,
infoMessage: '',
electionTime: Timestamp({ t: 1681418379, i: 2 }),
electionDate: ISODate("2023-04-13T20:39:39.000Z"),
configVersion: 5,
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},

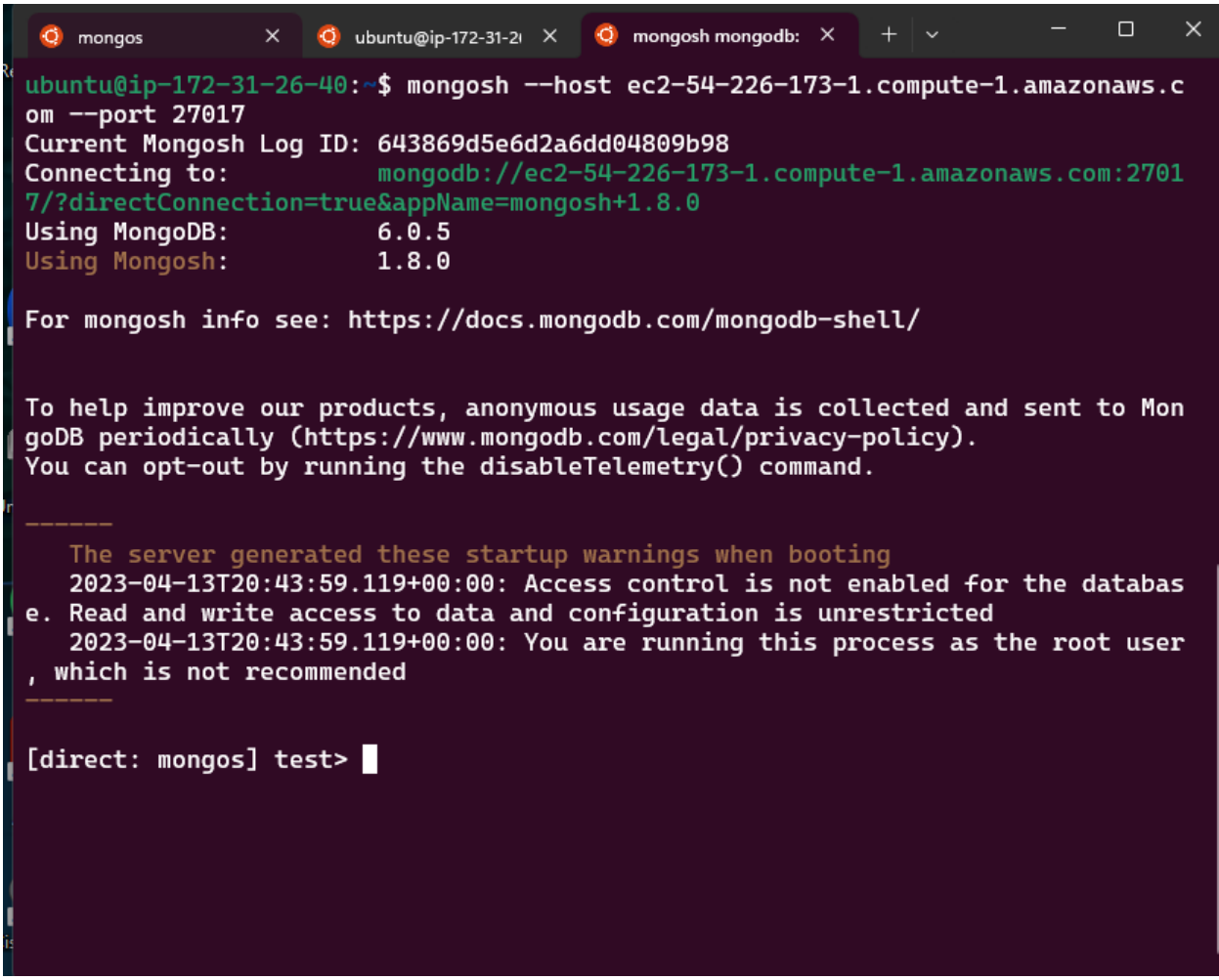


7. Start mongos & connect to config servers from mongos:
   sudo mongos --configdb config/ec2-3-86-247-48.compute-1.amazonaws.com:27031,ec2-3-86-247-48.compute-1.amazonaws.com:27032,ec2-3-86-247-48.compute-1.amazonaws.com:27033 --bind_ip

localhost,ec2-54-152-90-237.compute-1.amazonaws.com



a. Connect to config server:
mongosh --host ec2-54-152-90-237.compute-1.amazonaws.com --port 27017

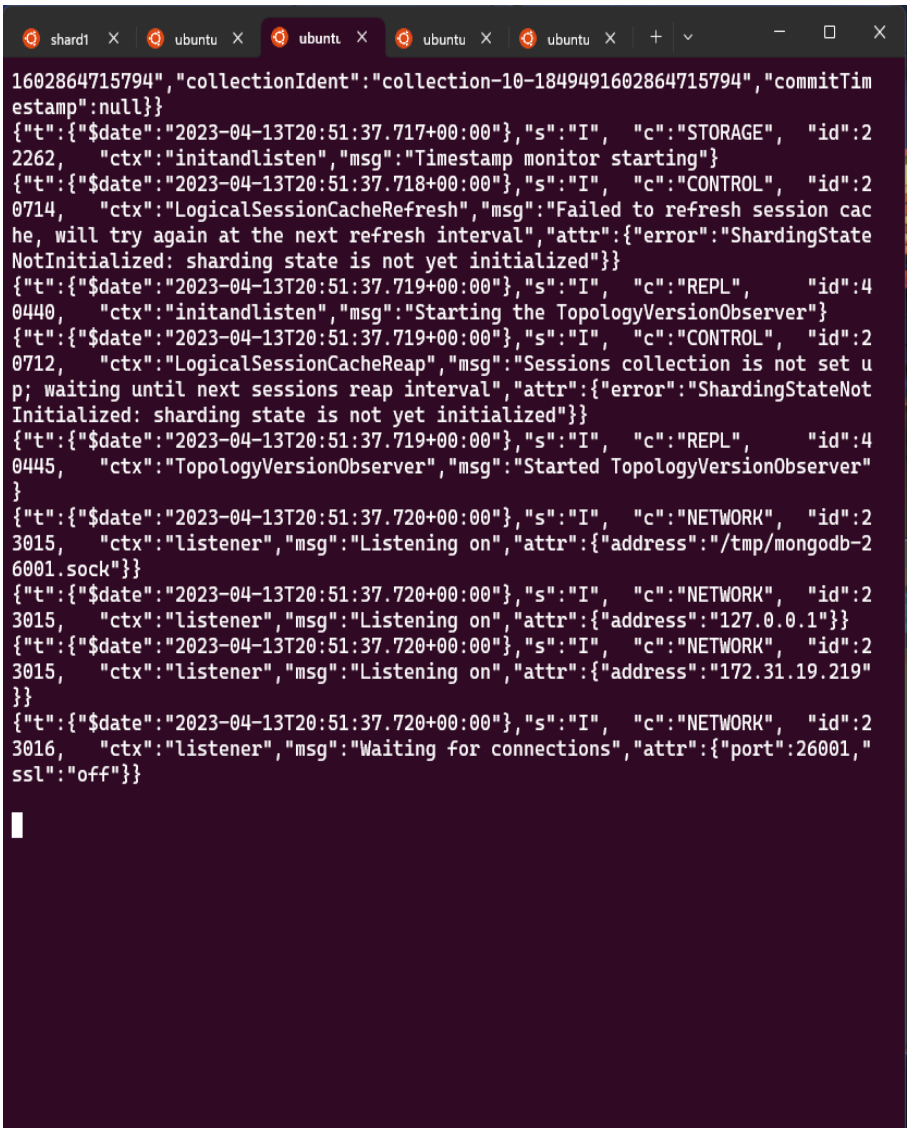


8.

a. **Shard1**:

i. **Start shard1 servers**:
Shard 1.1:
sudo mongod --port 26001 --shardsvr --bind_ip localhost,ec2-34-230-29-212.compute-1.amazonaws.com --replSet shard1

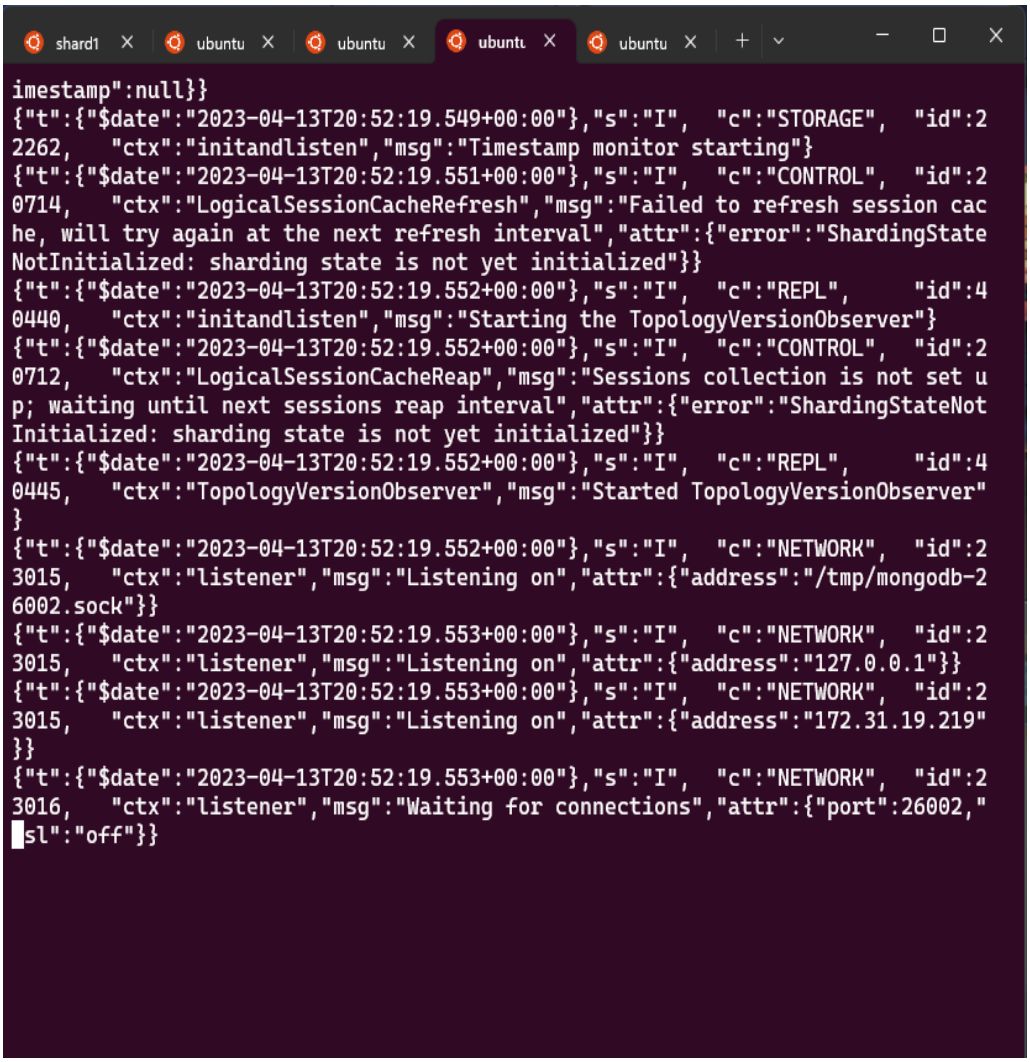
--dbpath db/shard1/mem1



Shard 1.2:

sudo mongod --port 26002 --shardsvr --bind_ip localhost,ec2-34-230-29-212.compute-1.amazonaws.com --replSet shard1

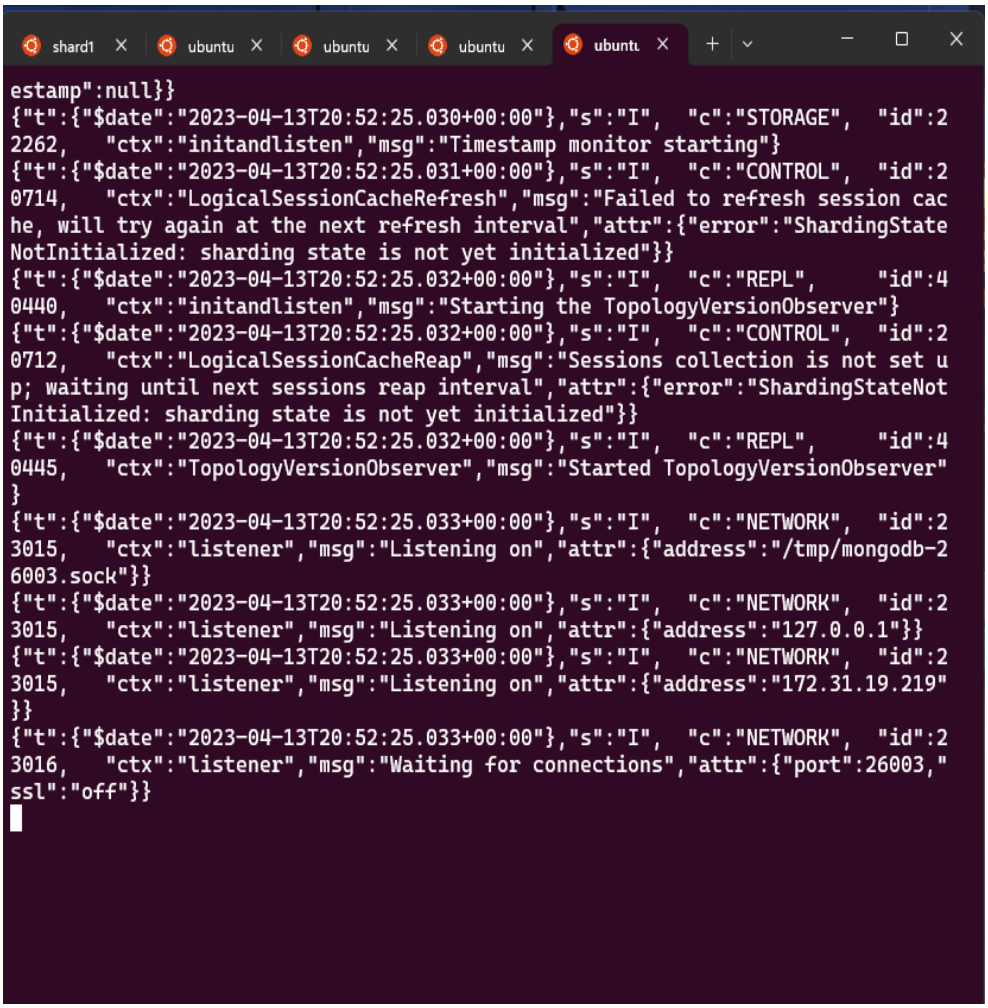
--dbpath db/shard1/mem2



Shard 1.3:

sudo mongod --port 26003 --shardsvr --bind_ip localhost,ec2-34-230-29-212.compute-1.amazonaws.com --replSet shard1

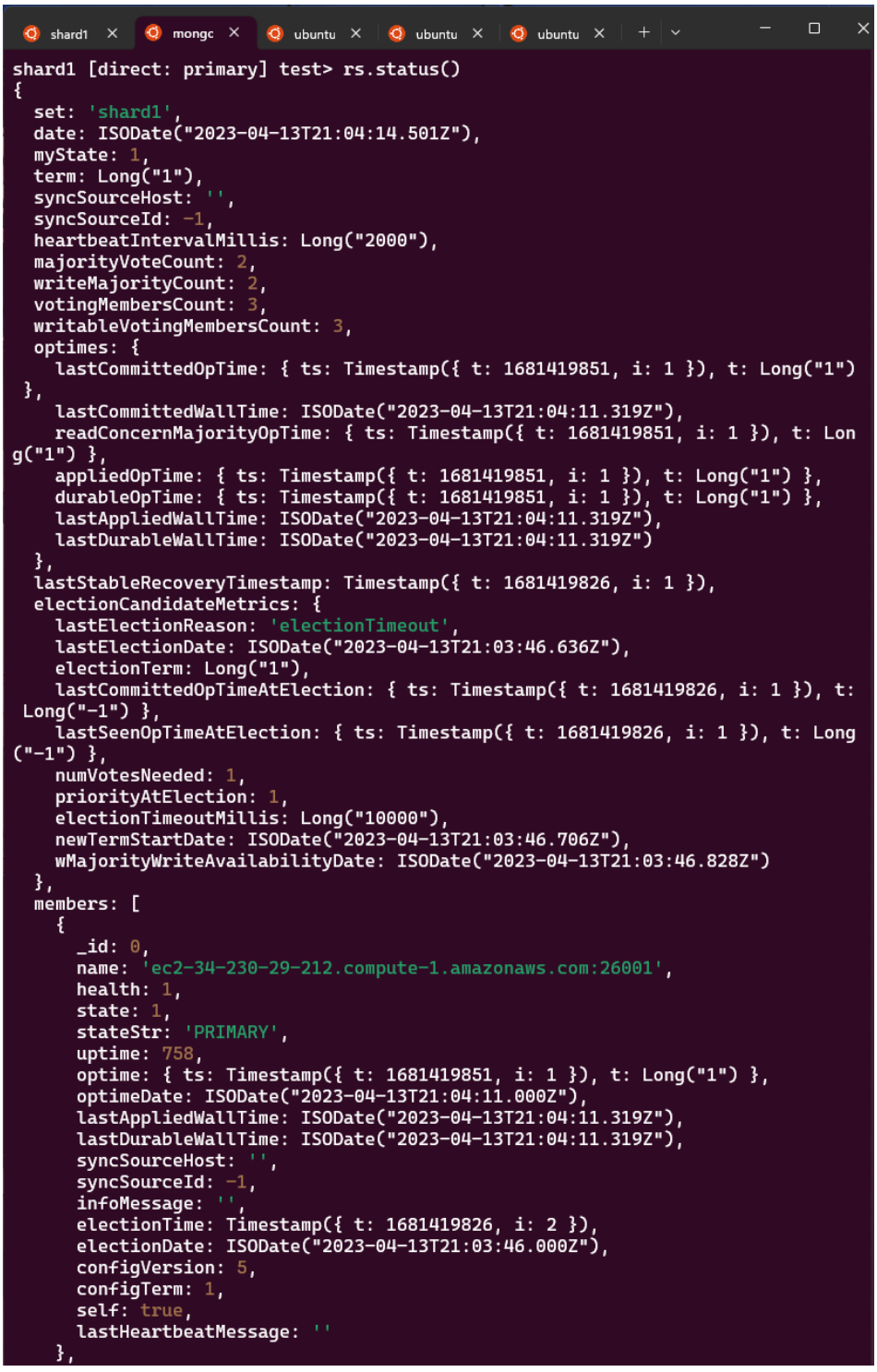
--dbpath db/shard1/mem3

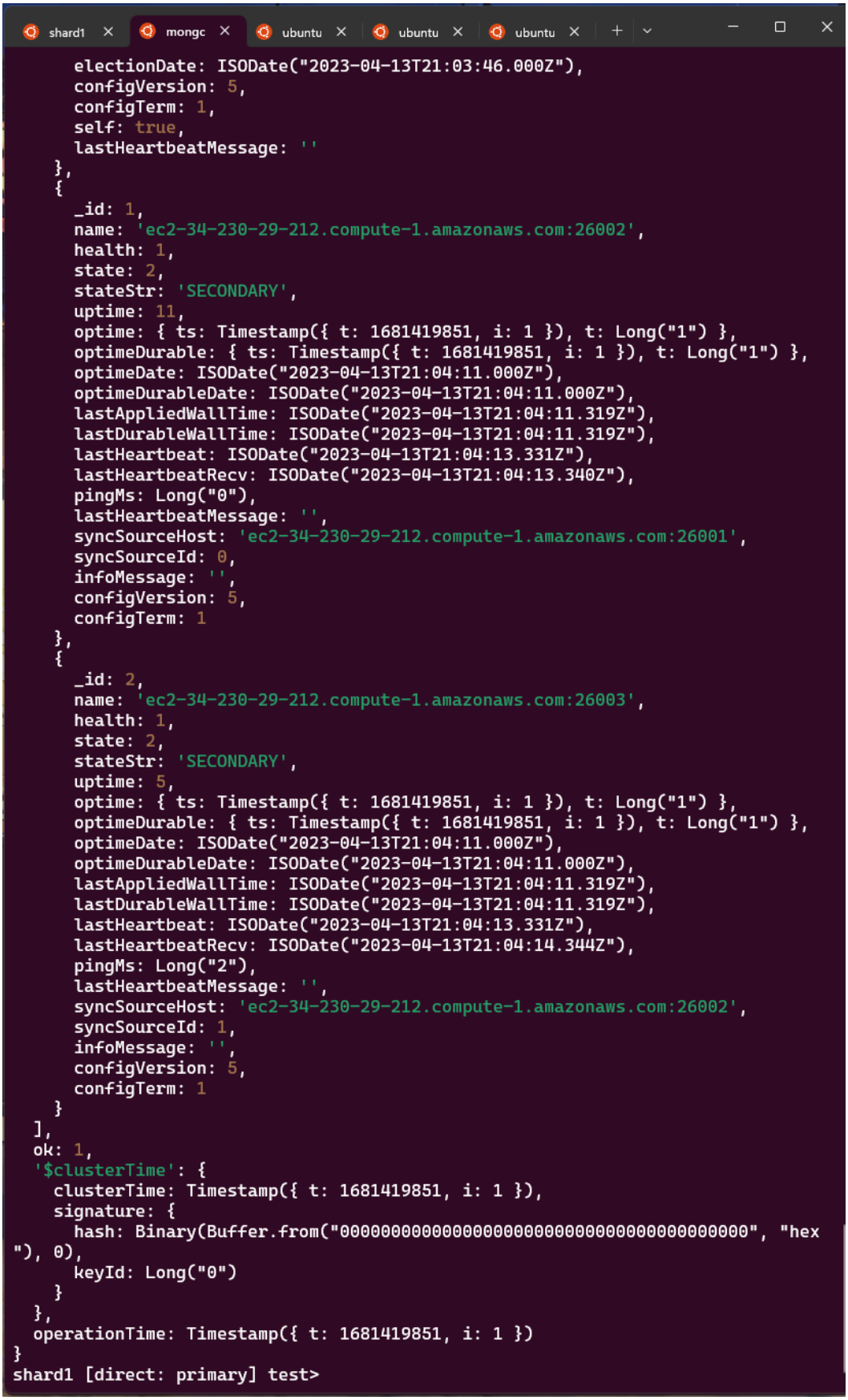


ii. Connect to one of those servers and set up replica set:

mongosh --port 26001
rs.initiate()
rs.status()
rs.add("ec2-34-230-29-212.compute-1.amazonaws.com:26002")
rs.add("ec2-34-230-29-212.compute-1.amazonaws.com:26003")

rs.status()


shard1
mongc
ubuntu
ubuntu
ubuntu
shard1 [direct: primary] test> rs.status()
{
set: 'shard1',
date: ISODate("2023-04-13T21:04:14.501Z"),
myState: 1,
term: Long("1"),
syncSourceHost: '',
syncSourceId: -1,
heartbeatIntervalMillis: Long("2000"),
majorityVoteCount: 2,
writeMajorityCount: 2,
votingMembersCount: 3,
writableVotingMembersCount: 3,
optimes: {
lastCommittedOpTime: { ts: Timestamp({ t: 1681419851, i: 1 }), t: Long("1")
},
lastCommittedWallTime: ISODate("2023-04-13T21:04:11.319Z"),
readConcernMajorityOpTime: { ts: Timestamp({ t: 1681419851, i: 1 }), t: Lon
g("1") },
appliedOpTime: { ts: Timestamp({ t: 1681419851, i: 1 }), t: Long("1") },
durableOpTime: { ts: Timestamp({ t: 1681419851, i: 1 }), t: Long("1") },
lastAppliedWallTime: ISODate("2023-04-13T21:04:11.319Z"),
lastDurableWallTime: ISODate("2023-04-13T21:04:11.319Z")
},
lastStableRecoveryTimestamp: Timestamp({ t: 1681419826, i: 1 }),
electionCandidateMetrics: {
lastElectionReason: 'electionTimeout',
lastElectionDate: ISODate("2023-04-13T21:03:46.636Z"),
electionTerm: Long("1"),
lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1681419826, i: 1 }), t:
Long("-1") },
lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1681419826, i: 1 }), t: Long
("-1") },
numVotesNeeded: 1,
priorityAtElection: 1,
electionTimeoutMillis: Long("10000"),
newTermStartDate: ISODate("2023-04-13T21:03:46.706Z"),
wMajorityWriteAvailabilityDate: ISODate("2023-04-13T21:03:46.828Z")
},
members: [
{
_id: 0,
name: 'ec2-34-230-29-212.compute-1.amazonaws.com:26001',
health: 1,
state: 1,
stateStr: 'PRIMARY',
uptime: 758,
optime: { ts: Timestamp({ t: 1681419851, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T21:04:11.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T21:04:11.319Z"),
lastDurableWallTime: ISODate("2023-04-13T21:04:11.319Z"),
syncSourceHost: '',
syncSourceId: -1,
infoMessage: '',
electionTime: Timestamp({ t: 1681419826, i: 2 }),
electionDate: ISODate("2023-04-13T21:03:46.000Z"),
configVersion: 5,
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},


shard1
mongc
ubuntu
ubuntu
ubuntu
electionDate: ISODate("2023-04-13T21:03:46.000Z"),
configVersion: 5,
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},
{
_id: 1,
name: 'ec2-34-230-29-212.compute-1.amazonaws.com:26002',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 11,
optime: { ts: Timestamp({ t: 1681419851, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681419851, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T21:04:11.000Z"),
optimeDurableDate: ISODate("2023-04-13T21:04:11.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T21:04:11.319Z"),
lastDurableWallTime: ISODate("2023-04-13T21:04:11.319Z"),
lastHeartbeat: ISODate("2023-04-13T21:04:13.331Z"),
lastHeartbeatRecv: ISODate("2023-04-13T21:04:13.340Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-34-230-29-212.compute-1.amazonaws.com:26001',
syncSourceId: 0,
infoMessage: '',
configVersion: 5,
configTerm: 1
},
{
_id: 2,
name: 'ec2-34-230-29-212.compute-1.amazonaws.com:26003',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 5,
optime: { ts: Timestamp({ t: 1681419851, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681419851, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T21:04:11.000Z"),
optimeDurableDate: ISODate("2023-04-13T21:04:11.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T21:04:11.319Z"),
lastDurableWallTime: ISODate("2023-04-13T21:04:11.319Z"),
lastHeartbeat: ISODate("2023-04-13T21:04:13.331Z"),
lastHeartbeatRecv: ISODate("2023-04-13T21:04:14.344Z"),
pingMs: Long("2"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-34-230-29-212.compute-1.amazonaws.com:26002',
syncSourceId: 1,
infoMessage: '',
configVersion: 5,
configTerm: 1
}
],
ok: 1,
'$clusterTime': {
clusterTime: Timestamp({ t: 1681419851, i: 1 }),
signature: {
hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex
"), 0),
keyId: Long("0")
}
},
operationTime: Timestamp({ t: 1681419851, i: 1 })
}
shard1 [direct: primary] test>

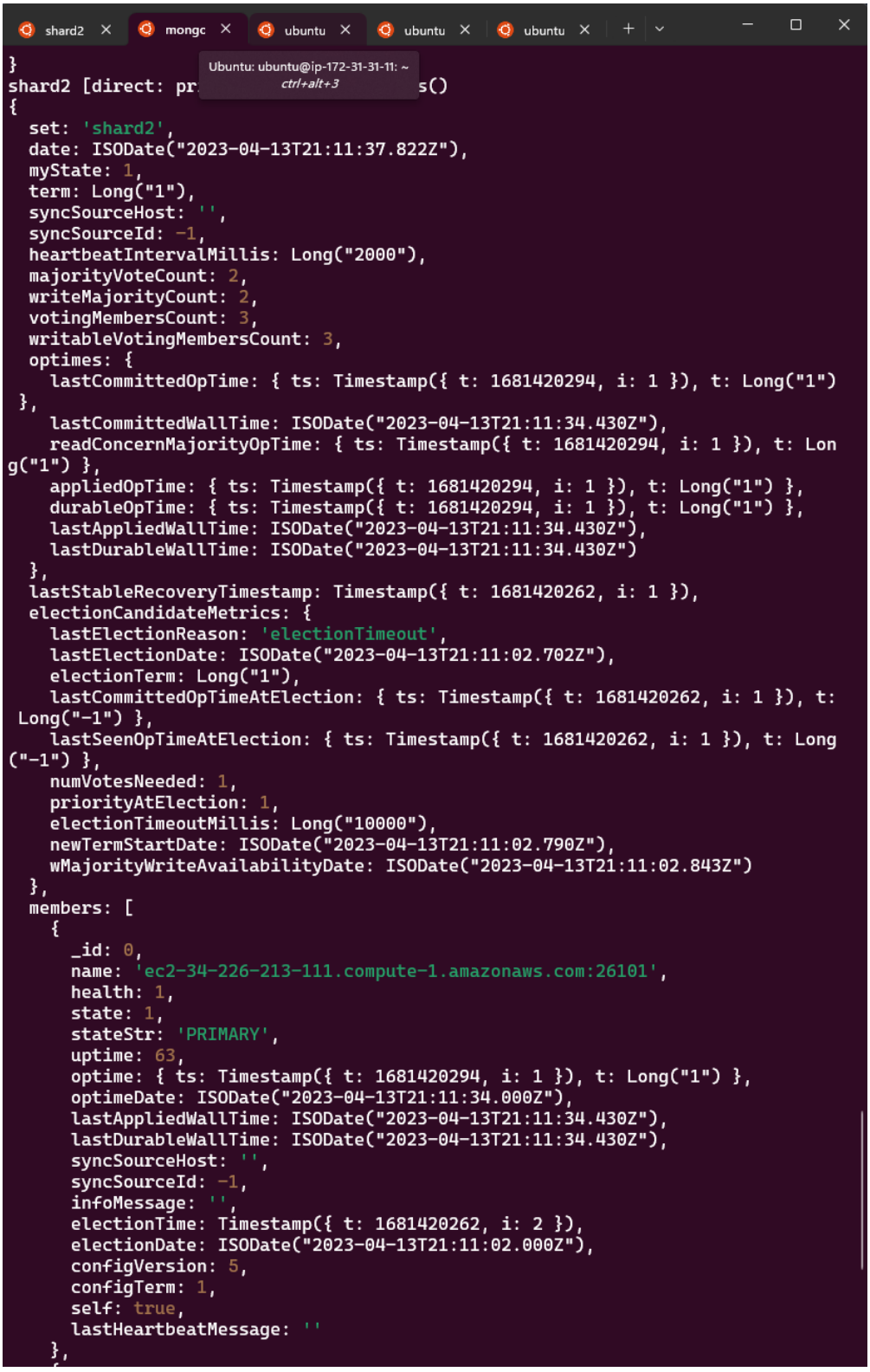
b. **Shard 2:**

sudo mongod --port 26101 --shardsvr --bind_ip localhost,ec2-34-226-213-111.compute-1.amazonaws.com --replSet shard2 --dbpath db/shard2/mem1

sudo mongod --port 26102 --shardsvr --bind_ip localhost,ec2-34-226-213-111.compute-1.amazonaws.com --replSet shard2 --dbpath db/shard2/mem2

sudo mongod --port 26103 --shardsvr --bind_ip localhost,ec2-34-226-213-111.compute-1.amazonaws.com --replSet shard2 --dbpath db/shard2/mem3

mongosh --port 26101
rs.initiate()
rs.status()
rs.add("ec2-34-226-213-111.compute-1.amazonaws.com:26102")
rs.add("ec2-34-226-213-111.compute-1.amazonaws.com:26103")
rs.status()


shard2
mongc
ubuntu
ubuntu
ubuntu
Ubuntu: ubuntu@ip-172-31-31-11: ~
ctrl+alt+3
}
shard2 [direct: pr
s()
{
set: 'shard2',
date: ISODate("2023-04-13T21:11:37.822Z"),
myState: 1,
term: Long("1"),
syncSourceHost: '',
syncSourceId: -1,
heartbeatIntervalMillis: Long("2000"),
majorityVoteCount: 2,
writeMajorityCount: 2,
votingMembersCount: 3,
writableVotingMembersCount: 3,
optimes: {
lastCommittedOpTime: { ts: Timestamp({ t: 1681420294, i: 1 }), t: Long("1")
},
lastCommittedWallTime: ISODate("2023-04-13T21:11:34.430Z"),
readConcernMajorityOpTime: { ts: Timestamp({ t: 1681420294, i: 1 }), t: Lon
g("1") },
appliedOpTime: { ts: Timestamp({ t: 1681420294, i: 1 }), t: Long("1") },
durableOpTime: { ts: Timestamp({ t: 1681420294, i: 1 }), t: Long("1") },
lastAppliedWallTime: ISODate("2023-04-13T21:11:34.430Z"),
lastDurableWallTime: ISODate("2023-04-13T21:11:34.430Z")
},
lastStableRecoveryTimestamp: Timestamp({ t: 1681420262, i: 1 }),
electionCandidateMetrics: {
lastElectionReason: 'electionTimeout',
lastElectionDate: ISODate("2023-04-13T21:11:02.702Z"),
electionTerm: Long("1"),
lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1681420262, i: 1 }), t:
Long("-1") },
lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1681420262, i: 1 }), t: Long
("-1") },
numVotesNeeded: 1,
priorityAtElection: 1,
electionTimeoutMillis: Long("10000"),
newTermStartDate: ISODate("2023-04-13T21:11:02.790Z"),
wMajorityWriteAvailabilityDate: ISODate("2023-04-13T21:11:02.843Z")
},
members: [
{
_id: 0,
name: 'ec2-34-226-213-111.compute-1.amazonaws.com:26101',
health: 1,
state: 1,
stateStr: 'PRIMARY',
uptime: 63,
optime: { ts: Timestamp({ t: 1681420294, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T21:11:34.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T21:11:34.430Z"),
lastDurableWallTime: ISODate("2023-04-13T21:11:34.430Z"),
syncSourceHost: '',
syncSourceId: -1,
infoMessage: '',
electionTime: Timestamp({ t: 1681420262, i: 2 }),
electionDate: ISODate("2023-04-13T21:11:02.000Z"),
configVersion: 5,
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},


shard2
mongc
ubuntu
ubuntu
ubuntu
Ubuntu: ubuntu@ip-172-31-31-11: ~
ctrl+alt+3
electionDate
T21:11:02.000Z"),
configVersi
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},
{
_id: 1,
name: 'ec2-34-226-213-111.compute-1.amazonaws.com:26102',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 9,
optime: { ts: Timestamp({ t: 1681420294, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681420294, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T21:11:34.000Z"),
optimeDurableDate: ISODate("2023-04-13T21:11:34.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T21:11:34.430Z"),
lastDurableWallTime: ISODate("2023-04-13T21:11:34.430Z"),
lastHeartbeat: ISODate("2023-04-13T21:11:36.436Z"),
lastHeartbeatRecv: ISODate("2023-04-13T21:11:36.447Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-34-226-213-111.compute-1.amazonaws.com:26101',
syncSourceId: 0,
infoMessage: '',
configVersion: 5,
configTerm: 1
},
{
_id: 2,
name: 'ec2-34-226-213-111.compute-1.amazonaws.com:26103',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 5,
optime: { ts: Timestamp({ t: 1681420294, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681420294, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T21:11:34.000Z"),
optimeDurableDate: ISODate("2023-04-13T21:11:34.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T21:11:34.430Z"),
lastDurableWallTime: ISODate("2023-04-13T21:11:34.430Z"),
lastHeartbeat: ISODate("2023-04-13T21:11:36.438Z"),
lastHeartbeatRecv: ISODate("2023-04-13T21:11:36.948Z"),
pingMs: Long("2"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-34-226-213-111.compute-1.amazonaws.com:26102',
syncSourceId: 1,
infoMessage: '',
configVersion: 5,
configTerm: 1
}
],
ok: 1,
'$clusterTime': {
clusterTime: Timestamp({ t: 1681420294, i: 1 }),
signature: {
hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex
"), 0),
keyId: Long("0")
}
},
operationTime: Timestamp({ t: 1681420294, i: 1 })
}
shard2 [direct: primary] test>

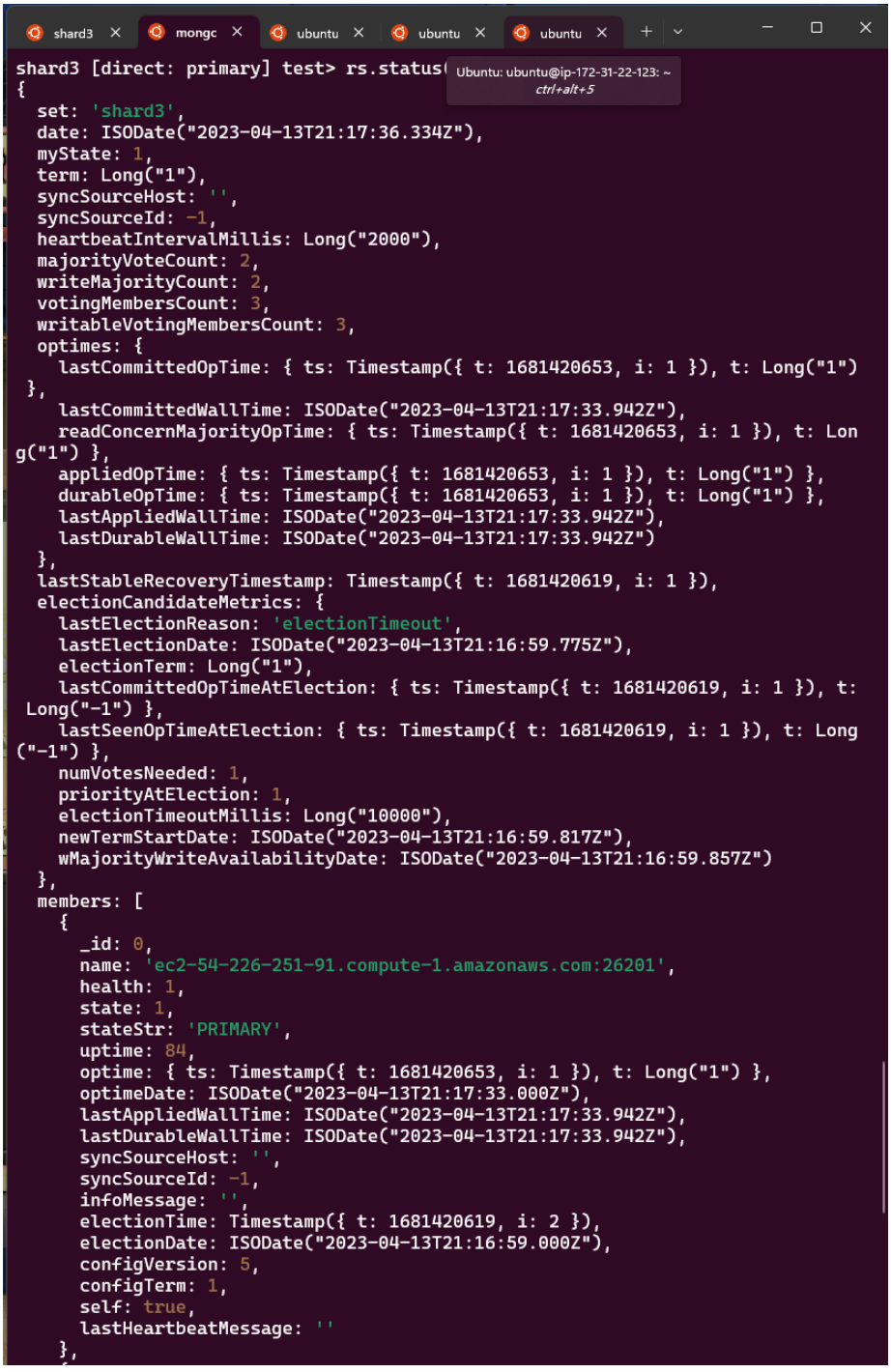
c. **Shard 3:**

sudo mongod --port 26201 --shardsvr --bind_ip localhost,ec2-54-226-251-91.compute-1.amazonaws.com --replSet shard3 --dbpath db/shard3/mem1

sudo mongod --port 26202 --shardsvr --bind_ip localhost,ec2-54-226-251-91.compute-1.amazonaws.com --replSet shard3 --dbpath db/shard3/mem2

sudo mongod --port 26203 --shardsvr --bind_ip localhost,ec2-54-226-251-91.compute-1.amazonaws.com --replSet shard3 --dbpath db/shard3/mem3

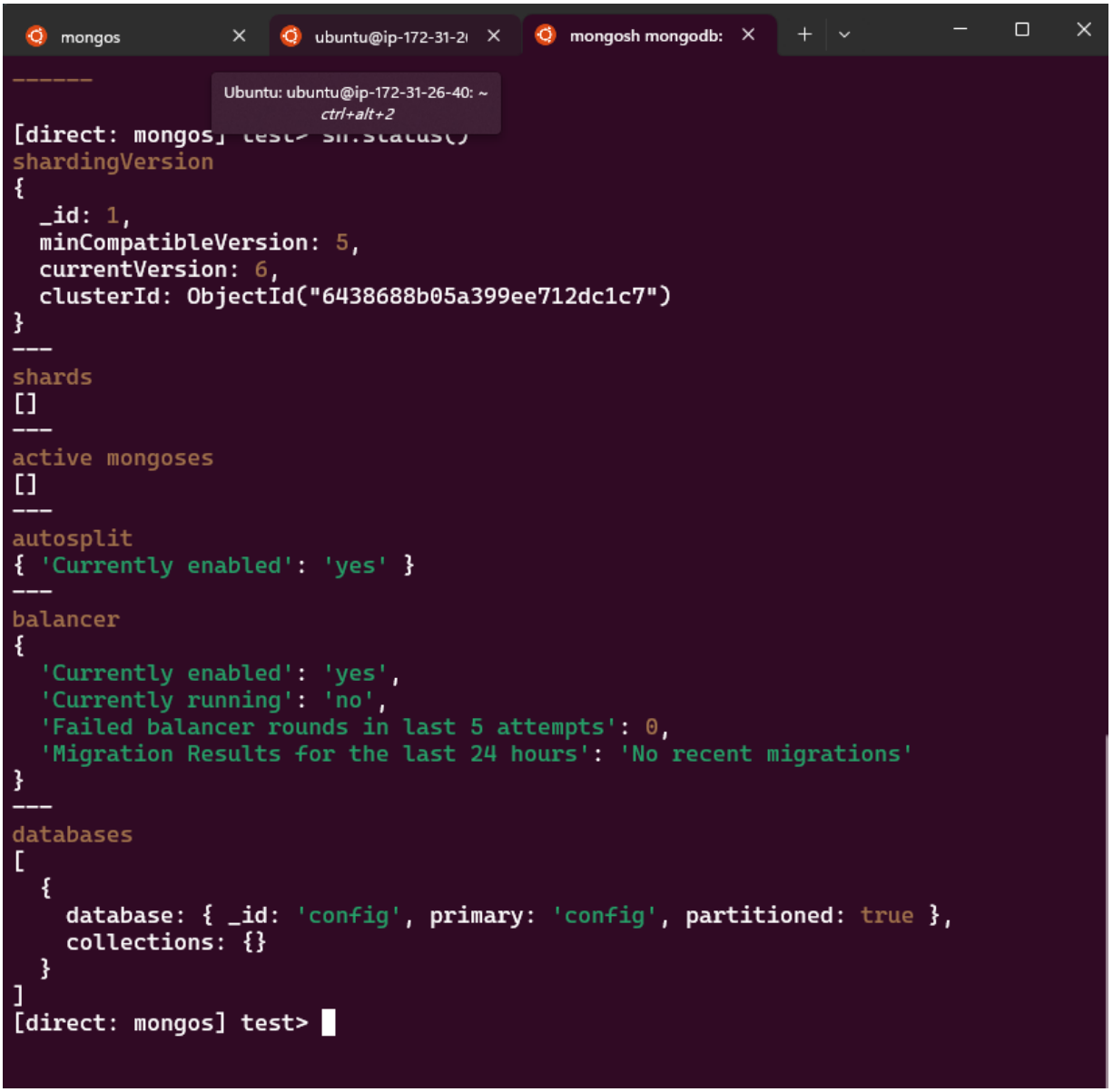
mongosh --port 26201
rs.initiate()
rs.status()
rs.add("ec2-54-226-251-91.compute-1.amazonaws.com:26202")
rs.add("ec2-54-226-251-91.compute-1.amazonaws.com:26203")
rs.status()




```
shard3 [direct: primary] test> rs.status(
{
  set: 'shard3',
  date: ISODate("2023-04-13T21:17:36.334Z"),
  myState: 1,
  term: Long("1"),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long("2000"),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1681420653, i: 1 }), t: Long("1")
 },
    lastCommittedWallTime: ISODate("2023-04-13T21:17:33.942Z"),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1681420653, i: 1 }), t: Lon
g("1") },
    appliedOpTime: { ts: Timestamp({ t: 1681420653, i: 1 }), t: Long("1") },
    durableOpTime: { ts: Timestamp({ t: 1681420653, i: 1 }), t: Long("1") },
    lastAppliedWallTime: ISODate("2023-04-13T21:17:33.942Z"),
    lastDurableWallTime: ISODate("2023-04-13T21:17:33.942Z")
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1681420619, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate("2023-04-13T21:16:59.775Z"),
    electionTerm: Long("1"),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1681420619, i: 1 }), t:
 Long("-1") },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1681420619, i: 1 }), t: Long
("-1") },
    numVotesNeeded: 1,
    priorityAtElection: 1,
    electionTimeoutMillis: Long("10000"),
    newTermStartDate: ISODate("2023-04-13T21:16:59.817Z"),
    wMajorityWriteAvailabilityDate: ISODate("2023-04-13T21:16:59.857Z")
  },
  members: [
    {
      _id: 0,
      name: 'ec2-54-226-251-91.compute-1.amazonaws.com:26201',
      health: 1,
      state: 1,
      stateStr: 'PRIMARY',
      uptime: 84,
      optime: { ts: Timestamp({ t: 1681420653, i: 1 }), t: Long("1") },
      optimeDate: ISODate("2023-04-13T21:17:33.000Z"),
      lastAppliedWallTime: ISODate("2023-04-13T21:17:33.942Z"),
      lastDurableWallTime: ISODate("2023-04-13T21:17:33.942Z"),
      syncSourceHost: '',
      syncSourceId: -1,
      infoMessage: '',
      electionTime: Timestamp({ t: 1681420619, i: 2 }),
      electionDate: ISODate("2023-04-13T21:16:59.000Z"),
      configVersion: 5,
      configTerm: 1,
      self: true,
      lastHeartbeatMessage: ''
    },
```


shard3
mongc
ubuntu
ubuntu
ubuntu
Ubuntu: ubuntu@ip-172-31-22-123: ~
ctrl+alt+5
},
{
_id: 1,
name: 'ec2-54-226-251-91.compute-1.amazonaws.com:26202',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 8,
optime: { ts: Timestamp({ t: 1681420653, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681420653, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T21:17:33.000Z"),
optimeDurableDate: ISODate("2023-04-13T21:17:33.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T21:17:33.942Z"),
lastDurableWallTime: ISODate("2023-04-13T21:17:33.942Z"),
lastHeartbeat: ISODate("2023-04-13T21:17:35.948Z"),
lastHeartbeatRecv: ISODate("2023-04-13T21:17:35.964Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-54-226-251-91.compute-1.amazonaws.com:26201',
syncSourceId: 0,
infoMessage: '',
configVersion: 5,
configTerm: 1
},
{
_id: 2,
name: 'ec2-54-226-251-91.compute-1.amazonaws.com:26203',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 4,
optime: { ts: Timestamp({ t: 1681420653, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681420653, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T21:17:33.000Z"),
optimeDurableDate: ISODate("2023-04-13T21:17:33.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T21:17:33.942Z"),
lastDurableWallTime: ISODate("2023-04-13T21:17:33.942Z"),
lastHeartbeat: ISODate("2023-04-13T21:17:35.951Z"),
lastHeartbeatRecv: ISODate("2023-04-13T21:17:34.459Z"),
pingMs: Long("2"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-54-226-251-91.compute-1.amazonaws.com:26202',
syncSourceId: 1,
infoMessage: '',
configVersion: 5,
configTerm: 1
}
],
ok: 1,
'$clusterTime': {
clusterTime: Timestamp({ t: 1681420653, i: 1 }),
signature: {
hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex
"), 0),
keyId: Long("0")
}
},
operationTime: Timestamp({ t: 1681420653, i: 1 })
}
shard3 [direct: primary] test>

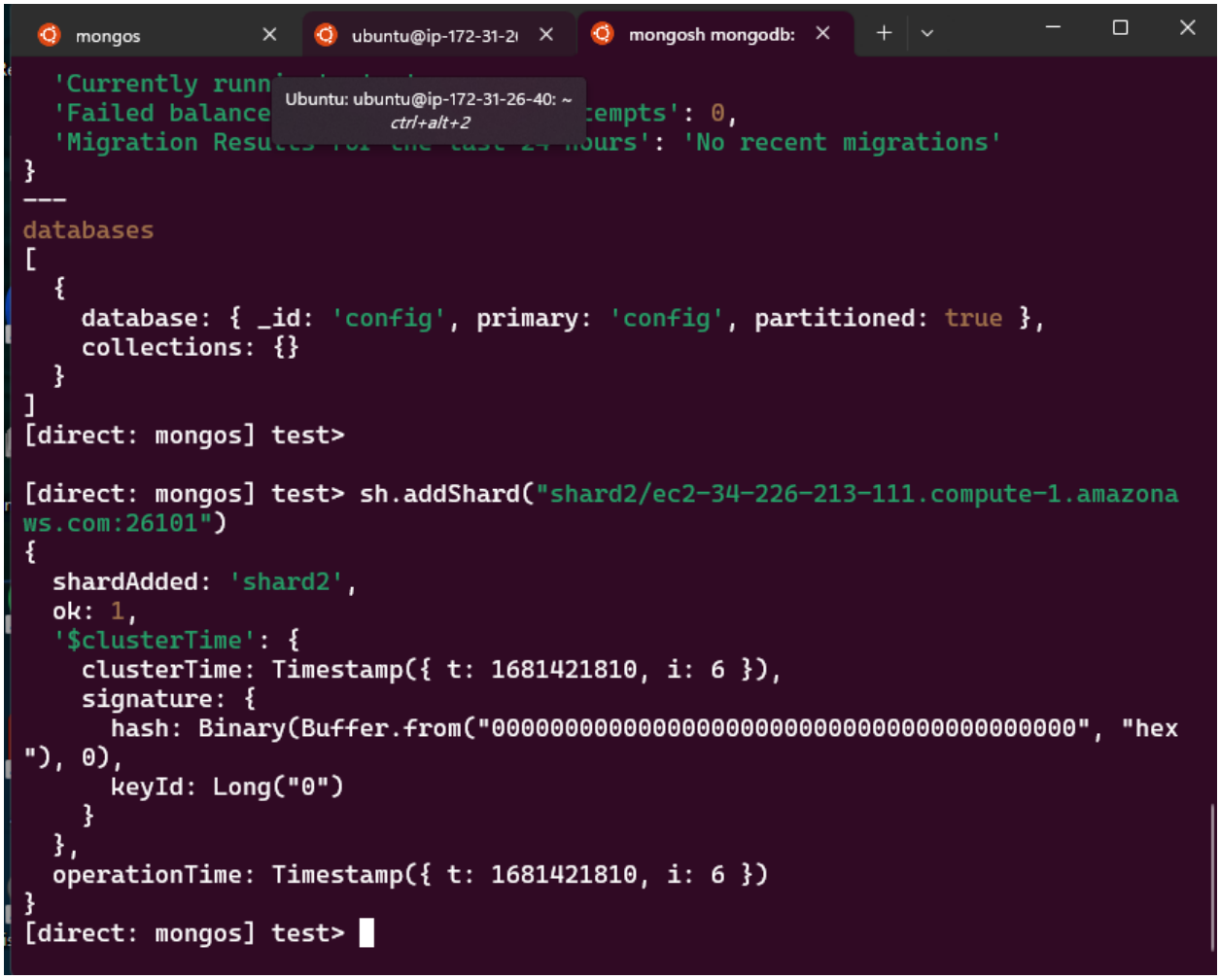
d. **Before adding shard:**
sh.status()



9. **Adding Shards:**
Shard 1:
sh.addShard("shard1/ec2-34-230-29-212.compute-1.amazonaws.com:26001")

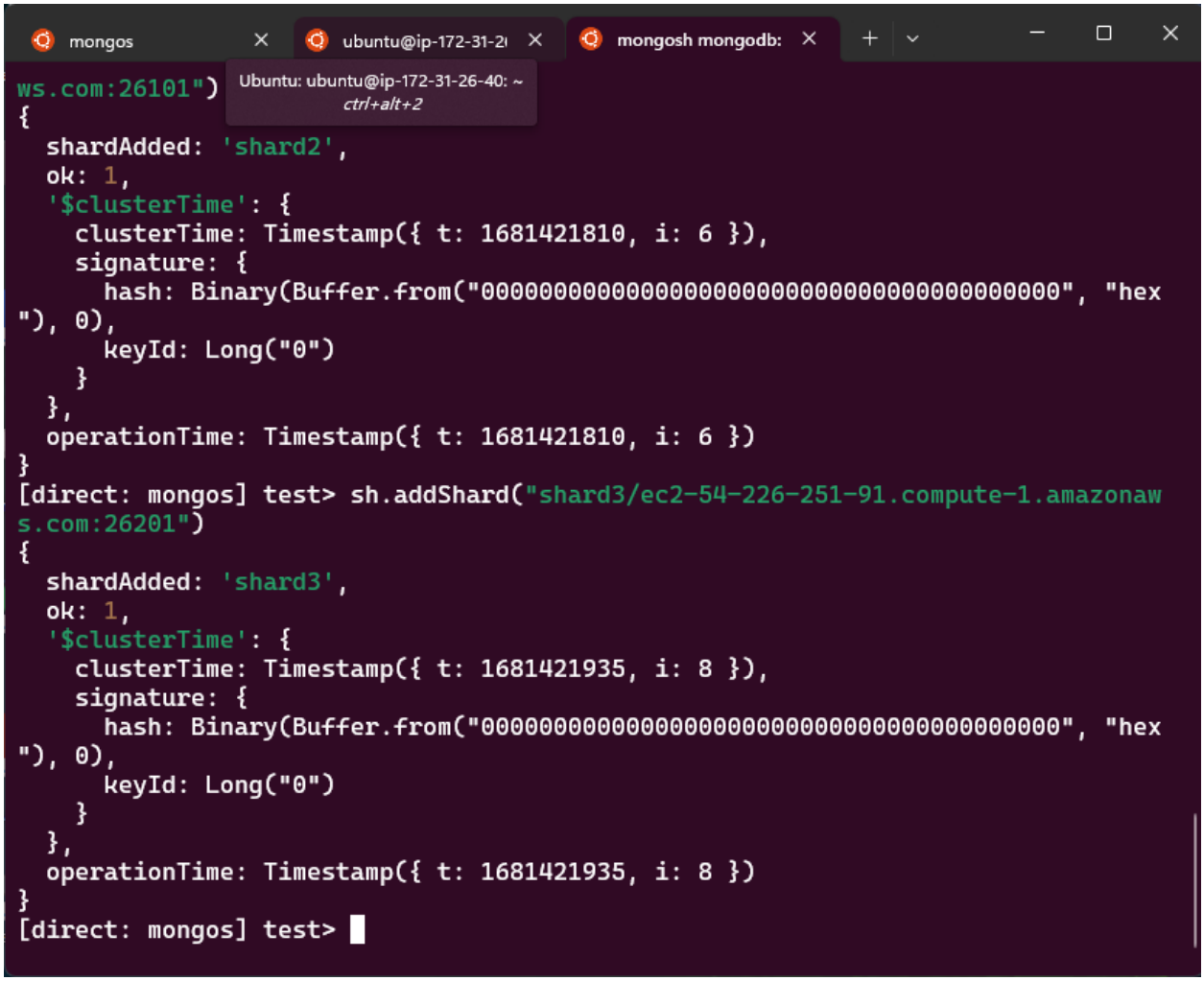
Shard 2:

sh.addShard("shard2/ec2-34-226-213-111.compute-1.amazonaws.com:26101")



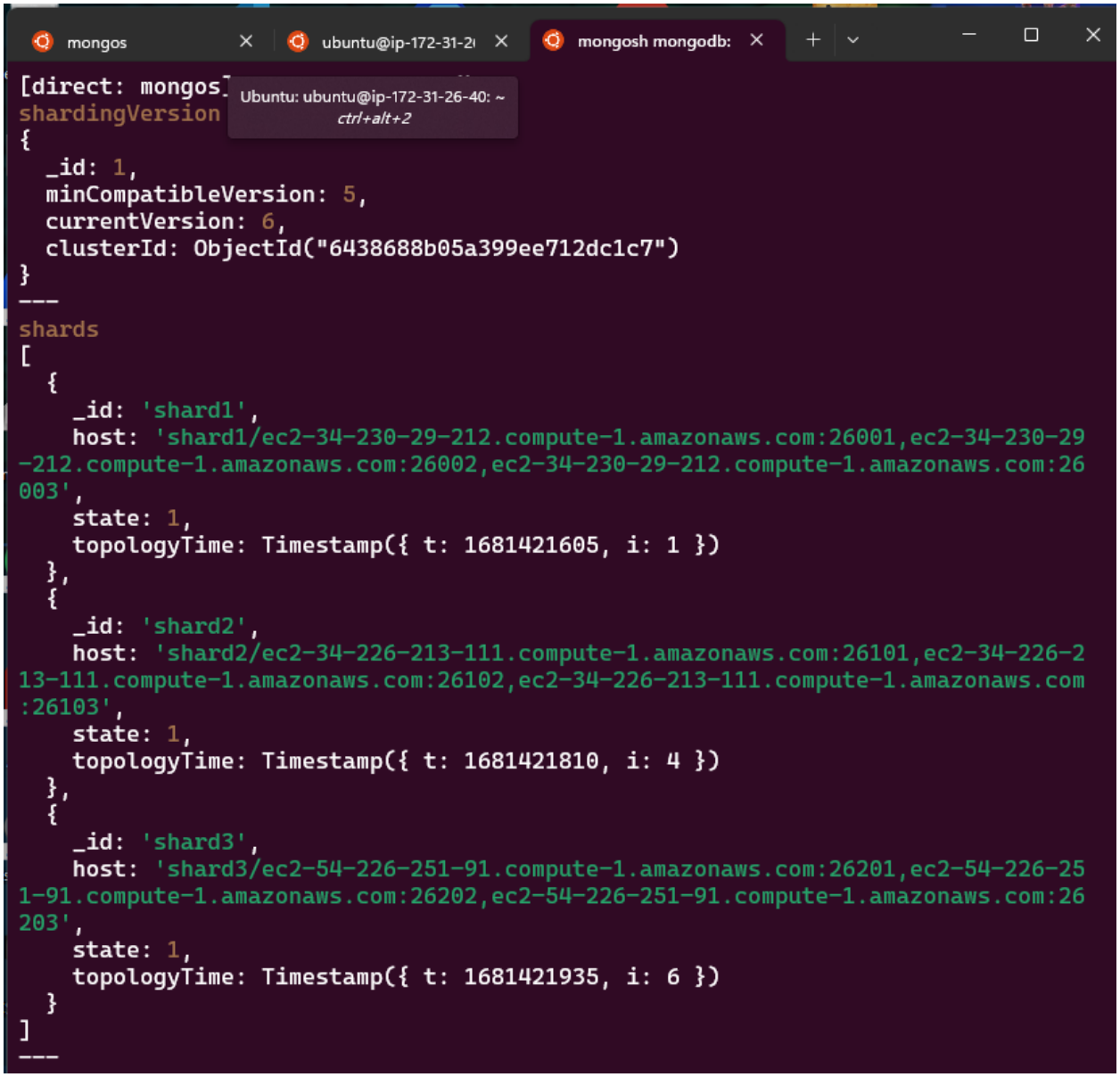
Shard 3:

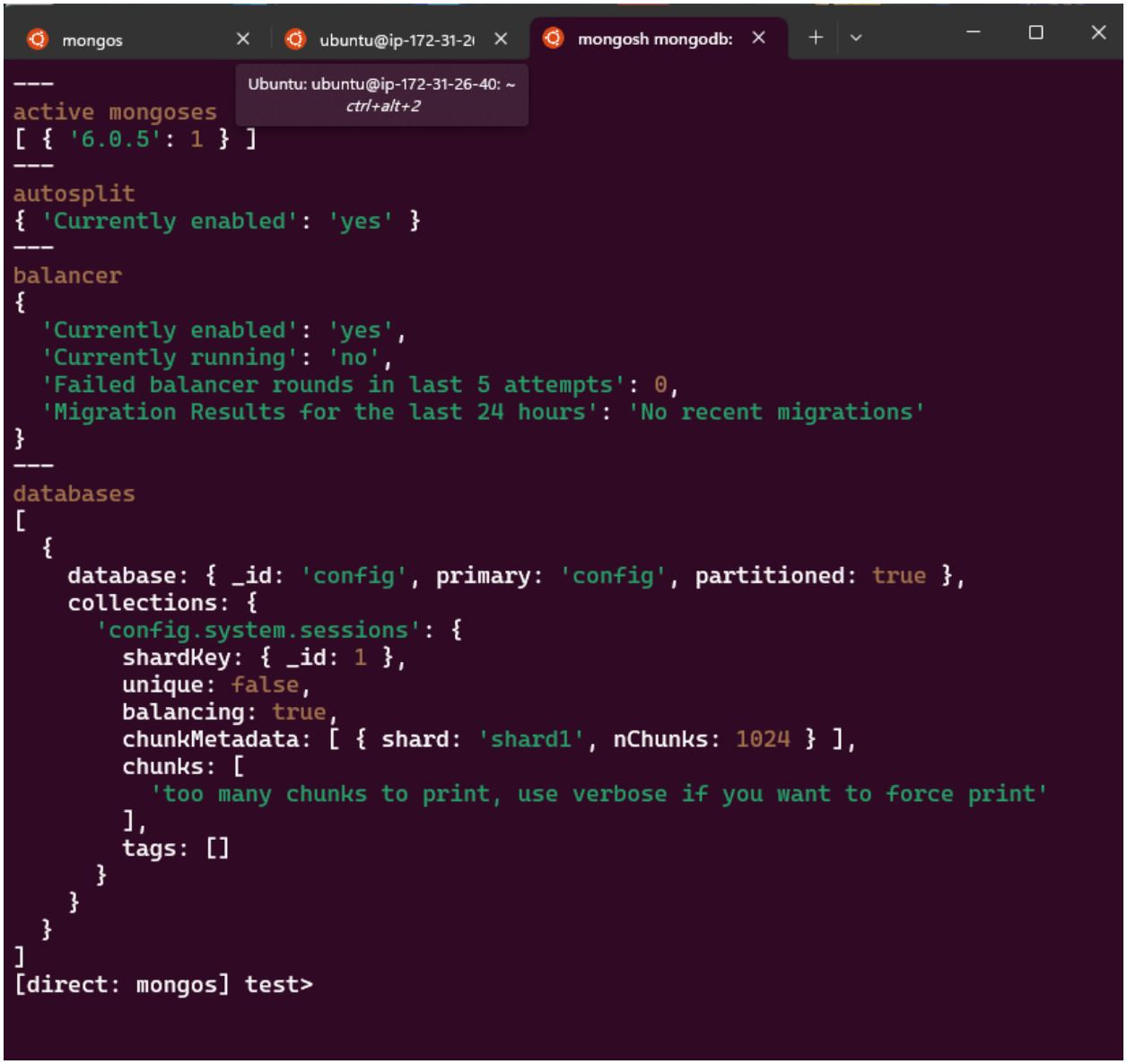
sh.addShard("shard3/ec2-54-226-251-91.compute-1.amazonaws.com:26201")



Shard status after adding shards :

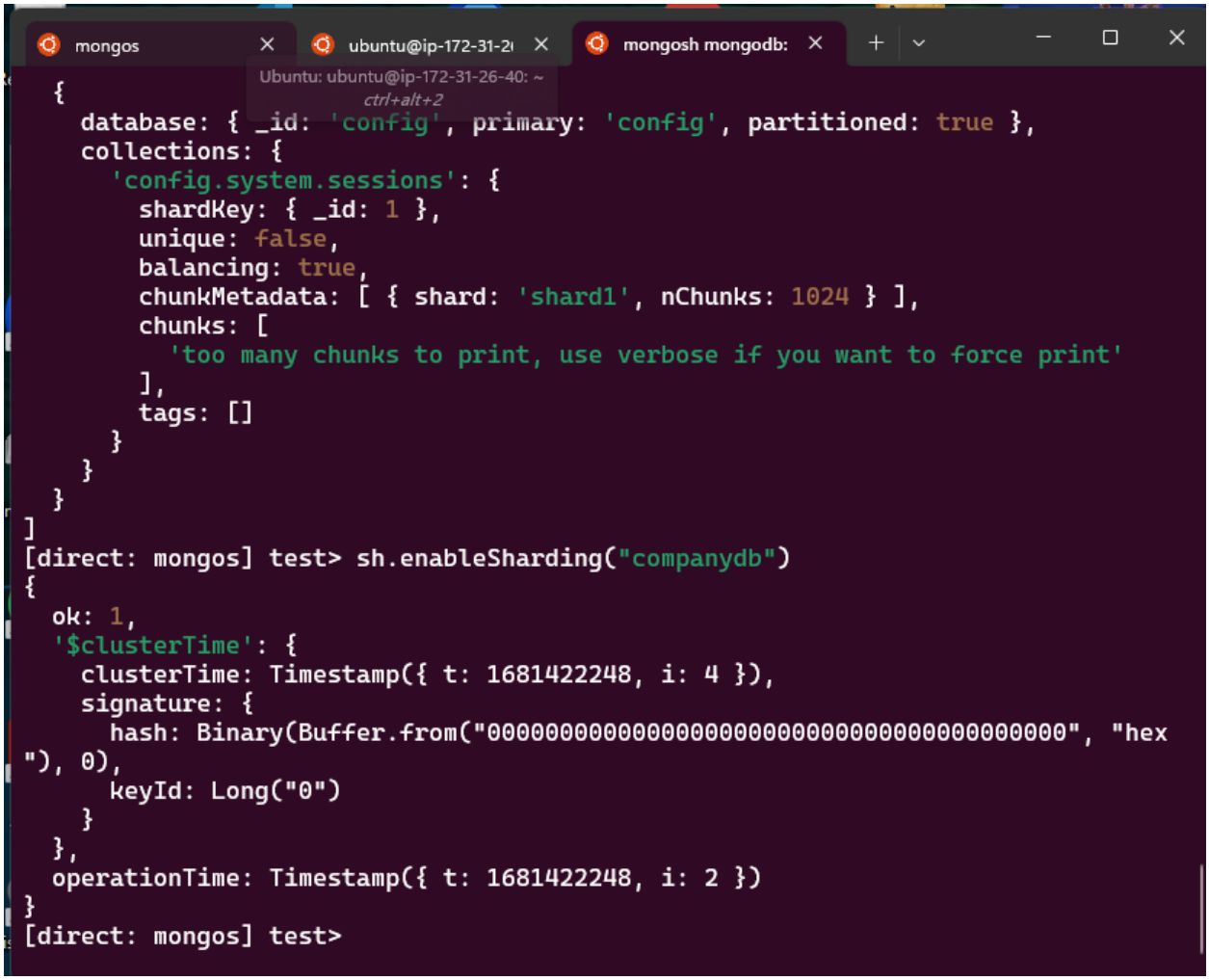
sh.status()


mongos
ubuntu@ip-172-31-2
mongosh mongodb:
[direct: mongos]
Ubuntu: ubuntu@ip-172-31-26-40: ~
ctrl+alt+2
shardingVersion
{
_id: 1,
minCompatibleVersion: 5,
currentVersion: 6,
clusterId: ObjectId("6438688b05a399ee712dc1c7")
}
---
shards
[
{
_id: 'shard1',
host: 'shard1/ec2-34-230-29-212.compute-1.amazonaws.com:26001,ec2-34-230-29-212.compute-1.amazonaws.com:26002,ec2-34-230-29-212.compute-1.amazonaws.com:26003',
state: 1,
topologyTime: Timestamp({ t: 1681421605, i: 1 })
},
{
_id: 'shard2',
host: 'shard2/ec2-34-226-213-111.compute-1.amazonaws.com:26101,ec2-34-226-213-111.compute-1.amazonaws.com:26102,ec2-34-226-213-111.compute-1.amazonaws.com:26103',
state: 1,
topologyTime: Timestamp({ t: 1681421810, i: 4 })
},
{
_id: 'shard3',
host: 'shard3/ec2-54-226-251-91.compute-1.amazonaws.com:26201,ec2-54-226-251-91.compute-1.amazonaws.com:26202,ec2-54-226-251-91.compute-1.amazonaws.com:26203',
state: 1,
topologyTime: Timestamp({ t: 1681421935, i: 6 })
}
]


mongos
ubuntu@ip-172-31-2
mongosh mongodb:
Ubuntu: ubuntu@ip-172-31-26-40: ~
ctrl+alt+2
---
active mongoses
[{ '6.0.5': 1 }]
---
autosplit
{ 'Currently enabled': 'yes' }
---
balancer
{
'Currently enabled': 'yes',
'Currently running': 'no',
'Failed balancer rounds in last 5 attempts': 0,
'Migration Results for the last 24 hours': 'No recent migrations'
}
---
databases
[
{
database: { _id: 'config', primary: 'config', partitioned: true },
collections: {
'config.system.sessions': {
shardKey: { _id: 1 },
unique: false,
balancing: true,
chunkMetadata: [{ shard: 'shard1', nChunks: 1024 }],
chunks: [
'too many chunks to print, use verbose if you want to force print'
],
tags: []
}
}
}
]
[direct: mongos] test>

10. **Enable sharding:**

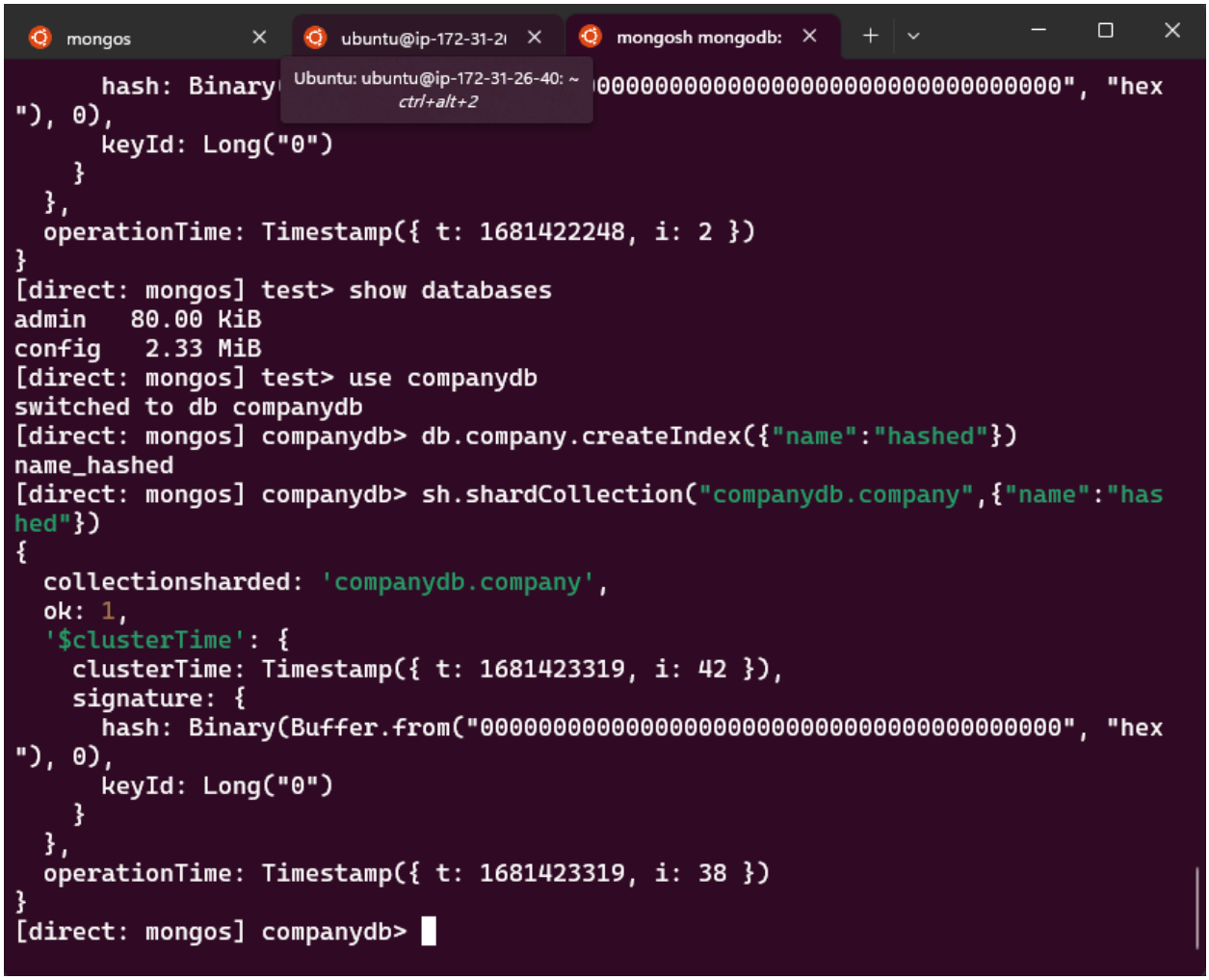
sh.enableSharding("companydb")



Create hashed-index using company's name and enable shard-key:

db.company.createIndex({"name":"hashed"})

sh.shardCollection("companydb.company",{"name":"hashed"})

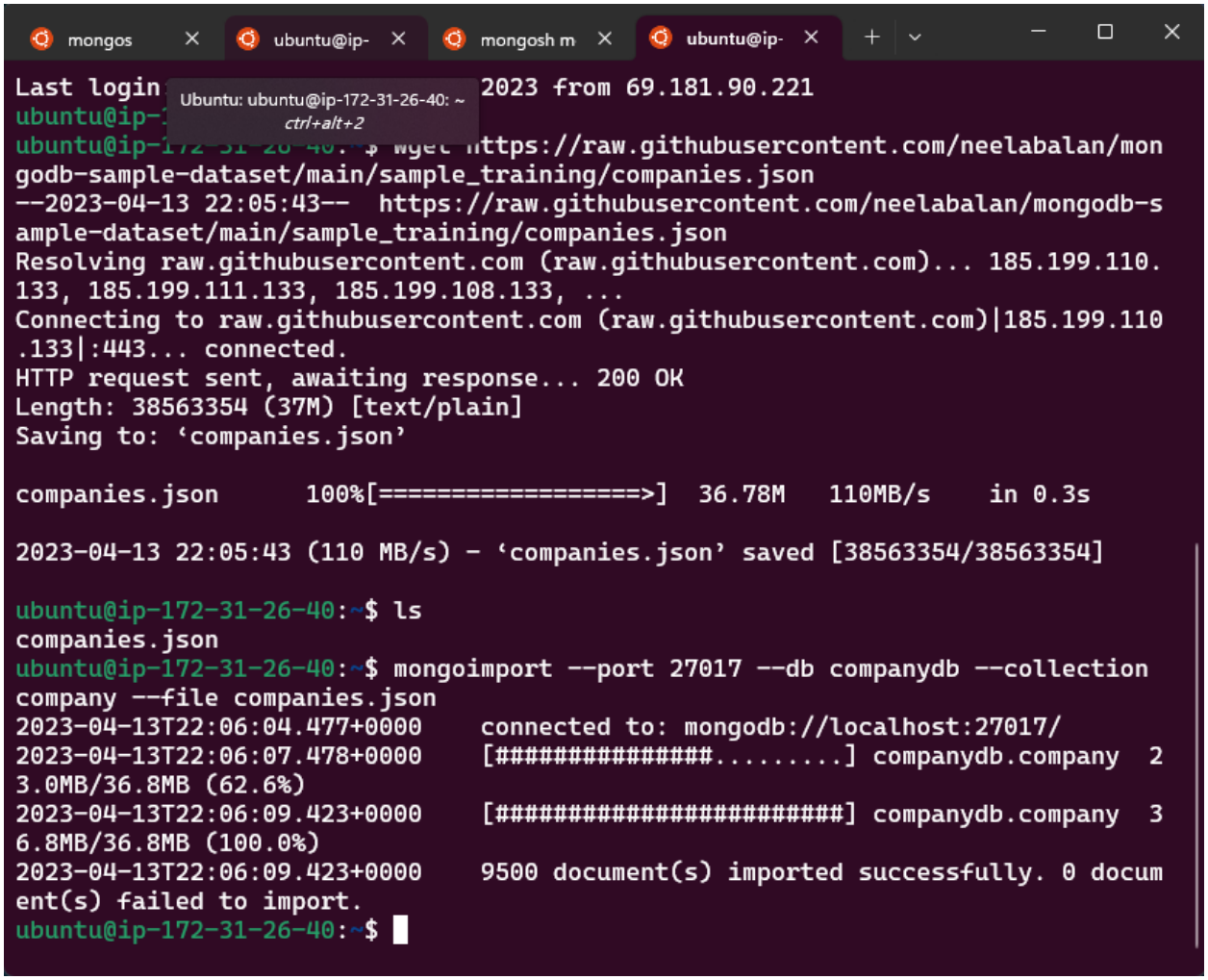


11. Dataset: [mongodb-sample-dataset/companies.json at main · neelabalan/mongodb-sample-dataset · GitHub]

Download and Import dataset:

wget https://raw.githubusercontent.com/neelabalan/mongodb-sample-dataset/main/sample_training/companies.json

mongoimport --port 27017 --db companydb --collection company --file companies.json



12.1: Find companies founded between 2010 and 2022

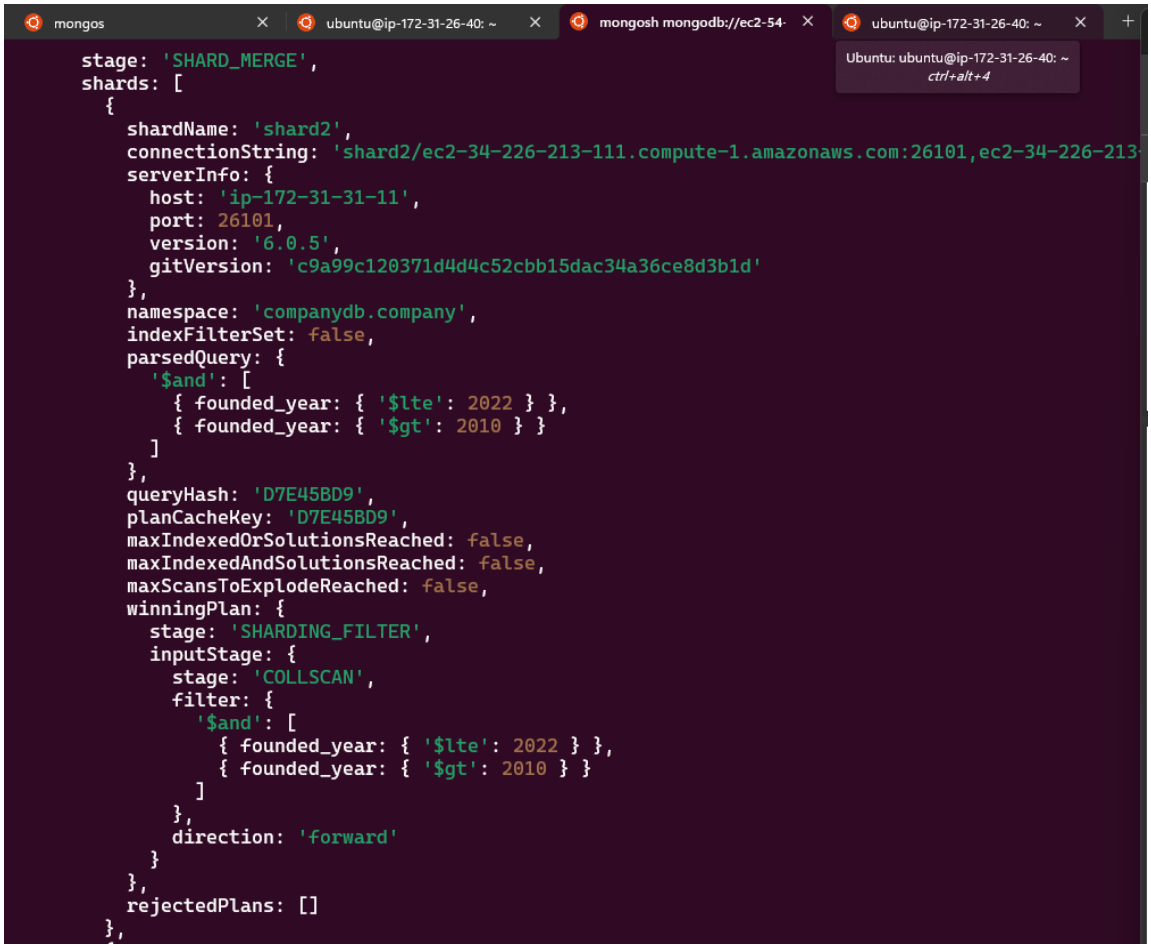
db.company.find({"founded_year":{$gt: 2010, $lte:2022}})

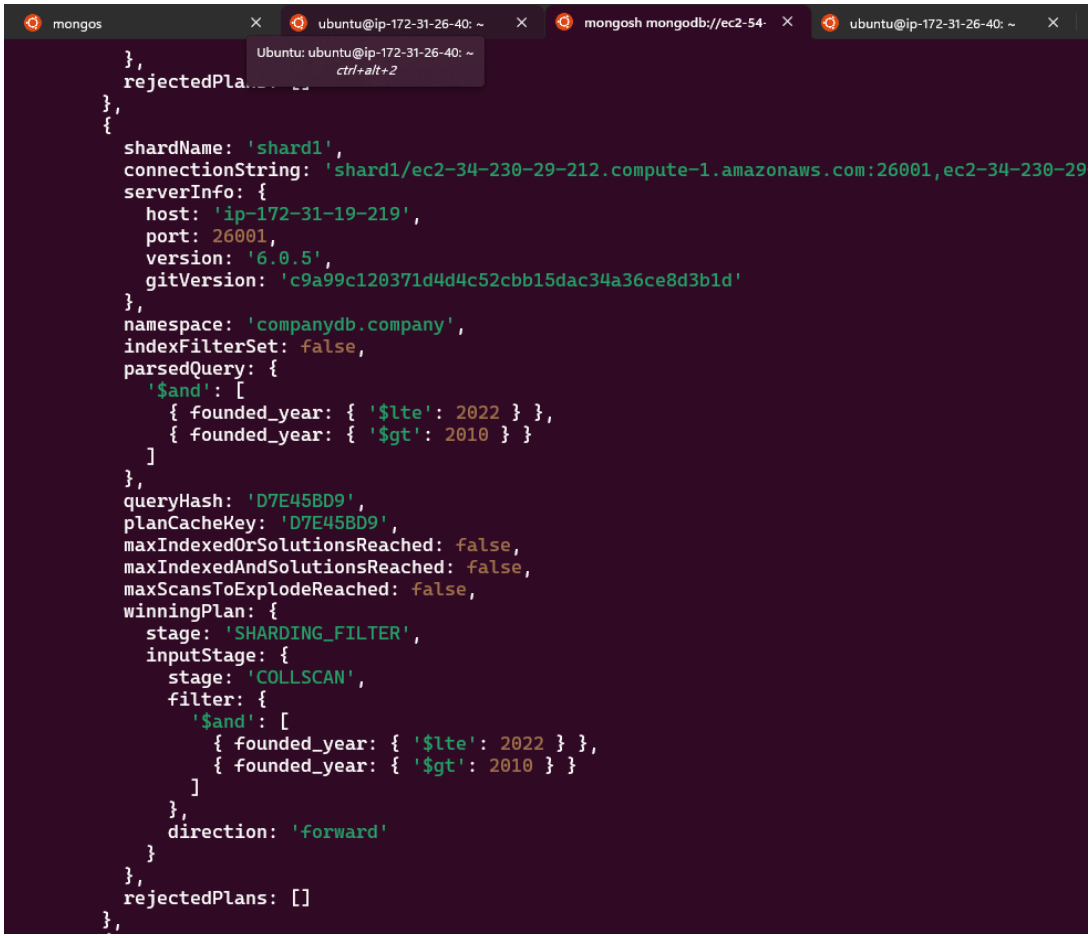
Result:


mongos
ubuntu@ip-
mongosh m
ubuntu@ip-
Ubuntu: ubuntu@ip-172-31-26-40: ~
ctrl+alt+2
partne
},
{
_id: ObjectId("52cdef7c4bab8bd675297fb1"),
name: 'headr',
permalink: 'headr',
crunchbase_url: 'http://www.crunchbase.com/company/headr',
homepage_url: 'http://www.headr.com',
blog_url: 'http://tripodsocial.tumblr.com/',
blog_feed_url: '',
twitter_username: 'tripodsocial',
category_code: 'web',
number_of_employees: 8,
founded_year: 2012,
founded_month: 1,
founded_day: null,
deadpooled_year: null,
deadpooled_month: null,
deadpooled_day: null,
deadpooled_url: '',
tag_list: '',
alias_list: null,
email_address: 'info@headr.com',
phone_number: '+49-511-336337-0 ',
description: 'Free Facebook video chat app.',
created_at: 'Sun Sep 02 22:13:39 UTC 2007',
updated_at: 'Tue Jan 22 06:31:44 UTC 2013',
overview: ' <a href="http://vive.me/" title="vive" rel="nofollow">vive lets you video chat with your friends and meet amazing people on Facebook! Sh
are your awesome, ask for advice, laugh and learn, be inspired! ',
image: {
available_sizes: [
[
[147, 73],
'assets/images/resized/0019/3143/193143v1-max-150x150.png'
],
[
[147, 73],
'assets/images/resized/0019/3143/193143v1-max-250x250.png'
],
[
[147, 73],
'assets/images/resized/0019/3143/193143v1-max-450x450.png'
]
],
attribution: null
},
products: [],
relationships: [
{
is_past: false,
title: 'CEO',
person: {
first_name: 'Arnd',
last_name: 'Aschentrup',
permalink: 'arnd-aschentrup'
}
},
{
is_past: false,
title: 'CPO',
person: {
first_name: 'Matthias',
last_name: 'Kleimann',
permalink: 'matthias-kleimann'

Shards served the query: **shard 1, shard 2, shard 3**

db.company.find({"founded_year":{$gt: 2010, $lte:2022}}).explain()


mongos
ubuntu@ip-172-31-26-40: ~
mongosh mongodb://ec2-54·
ubuntu@ip-172-31-26-40: ~
Ubuntu: ubuntu@ip-172-31-26-40: ~
ctrl+alt+4
stage: 'SHARD_MERGE',
shards: [
{
shardName: 'shard2',
connectionString: 'shard2/ec2-34-226-213-111.compute-1.amazonaws.com:26101,ec2-34-226-213
serverInfo: {
host: 'ip-172-31-31-11',
port: 26101,
version: '6.0.5',
gitVersion: 'c9a99c120371d4d4c52cbb15dac34a36ce8d3b1d'
},
namespace: 'companydb.company',
indexFilterSet: false,
parsedQuery: {
'$and': [
{ founded_year: { '$lte': 2022 } },
{ founded_year: { '$gt': 2010 } }
]
},
queryHash: 'D7E45BD9',
planCacheKey: 'D7E45BD9',
maxIndexedOrSolutionsReached: false,
maxIndexedAndSolutionsReached: false,
maxScansToExplodeReached: false,
winningPlan: {
stage: 'SHARDING_FILTER',
inputStage: {
stage: 'COLLSCAN',
filter: {
'$and': [
{ founded_year: { '$lte': 2022 } },
{ founded_year: { '$gt': 2010 } }
]
},
direction: 'forward'
}
},
rejectedPlans: []
},



Execution time: 12

db.company.find({"founded_year":{$gt: 2010, $lte:2022}}).explain("executionStats").executionStats.executionTimeMillis

```
[direct: mongos] companydb> db.company.find({"founded_year":{$gt: 2010, $lte:20
22}}).explain("executionStats").executionStats.executionTimeMillis
12
[direct: mongos] companydb>
```

12.2: Find companies that offices in San Jose founded after 2000

db.company.find({"offices": {$elemMatch: {"city": "San Jose"}}, founded_year: {$gt: 2000}}, {_id: 1, name: 1, founded_year: 1, offices: 1})


mongos
ubuntu@ip-
mongosh m
ubuntu@ip-
Ubuntu: ubuntu@ip-172-31-26-40: ~
ctrl+alt+2
{
_id: O bd67529846c"),
name: 'Asterpix',
founded_year: 2006,
offices: [
{
description: null,
address1: '4340 Stevens Creek Blvd #191',
address2: '',
zip_code: '95129',
city: 'San Jose',
state_code: 'CA',
country_code: 'USA',
latitude: 37.321798,
longitude: -121.979453
}
]
},
{
_id: ObjectId("52cdef7c4bab8bd67529854c"),
name: 'Wrike',
founded_year: 2006,
offices: [
{
description: '',
address1: '',
address2: '',
zip_code: '95113',
city: 'San Jose',
state_code: 'CA',
country_code: 'USA',
latitude: 37.3973787,
longitude: -122.0879281
}
]
},
{
_id: ObjectId("52cdef7d4bab8bd675298d7a"),
name: 'Codefast',
founded_year: 2004,
offices: [
{
description: null,
address1: '',
address2: '',
zip_code: '',
city: 'San Jose',
state_code: 'CA',
country_code: 'USA',
latitude: 37.316466,
longitude: -121.873881
}
]
},
{
_id: ObjectId("52cdef7d4bab8bd6752993eb"),
name: 'Roamware',
founded_year: 2002,
offices: [
{
description: 'Corporate Headquarters',
address1: '3031 Tisch Way',
address2: 'Suite 1000',
zip_code: '95128',
city: 'San Jose',
state_code: 'CA',

Shards served the query: shard 1, shard 2, shard 3:

db.company.find({"offices": {$elemMatch: {"city": "San Jose"}}, founded_year: {$gt: 2000}}, {_id: 1, name: 1, founded_year: 1, offices: 1}).explain()


mongos
ubuntu@ip-
mongosh m
ubuntu@ip-
Ubuntu: ubuntu@ip-172-31-26-40: ~
ctrl+alt+2
ard2/ec2-34-226-213-111.compute-1.amazonaws.com:
26101,ec2-34-226-213-111.compute-1.amazonaws.com:26102,ec2-34-226-213-111.compu
te-1.amazonaws.com:26103',
serverInfo: {
host: 'ip-172-31-31-11',
port: 26101,
version: '6.0.5',
gitVersion: 'c9a99c120371d4d4c52cbb15dac34a36ce8d3b1d'
},
namespace: 'companydb.company',
indexFilterSet: false,
parsedQuery: {
'$and': [
{
offices: { '$elemMatch': { city: { '$eq': 'San Jose' } } }
},
{ founded_year: { '$gt': 2000 } }
]
},
queryHash: 'B8E7A422',
planCacheKey: 'B8E7A422',
maxIndexedOrSolutionsReached: false,
maxIndexedAndSolutionsReached: false,
maxScansToExplodeReached: false,
winningPlan: {
stage: 'PROJECTION_SIMPLE',
transformBy: { _id: 1, name: 1, founded_year: 1, offices: 1 },
inputStage: {
stage: 'SHARDING_FILTER',
inputStage: {
stage: 'COLLSCAN',
filter: {
'$and': [
{
offices: { '$elemMatch': { city: { '$eq': 'San Jose' } } }
}
},
{ founded_year: { '$gt': 2000 } }
]
},
direction: 'forward'
}
}
},
rejectedPlans: []
},
{
shardName: 'shard3',
connectionString: 'shard3/ec2-54-226-251-91.compute-1.amazonaws.com:2
6201,ec2-54-226-251-91.compute-1.amazonaws.com:26202,ec2-54-226-251-91.compute-
1.amazonaws.com:26203',
serverInfo: {
host: 'ip-172-31-22-123',
port: 26201,
version: '6.0.5',
gitVersion: 'c9a99c120371d4d4c52cbb15dac34a36ce8d3b1d'
},
namespace: 'companydb.company',
indexFilterSet: false,
parsedQuery: {
'$and': [
{
offices: { '$elemMatch': { city: { '$eq': 'San Jose' } } }
},
{ founded_year: { '$gt': 2000 } }

Execution Time: 7

db.company.find({"offices": {$elemMatch: {"city": "San Jose"}}, founded_year: {$gt: 2000}}, {_id: 1, name: 1, founded_year: 1, offices: 1}).explain("executionStats").executionStats.executionTimeMillis

```
[direct: mongos] companydb> db.company.find({"offices": {$elemMatch: {"city": "
San Jose"}}, founded_year: {$gt: 2000}}, {_id: 1, name: 1, founded_year: 1,  of
fices: 1}).explain("executionStats").executionStats.executionTimeMillis
7
[direct: mongos] companydb>
```

12.3: Find all companies that has founded year in: 2012 and 2013:

db.company.find({founded_year: {$in: [2013,2012]}},{_id: 1, name: 1, founded_year: 1})

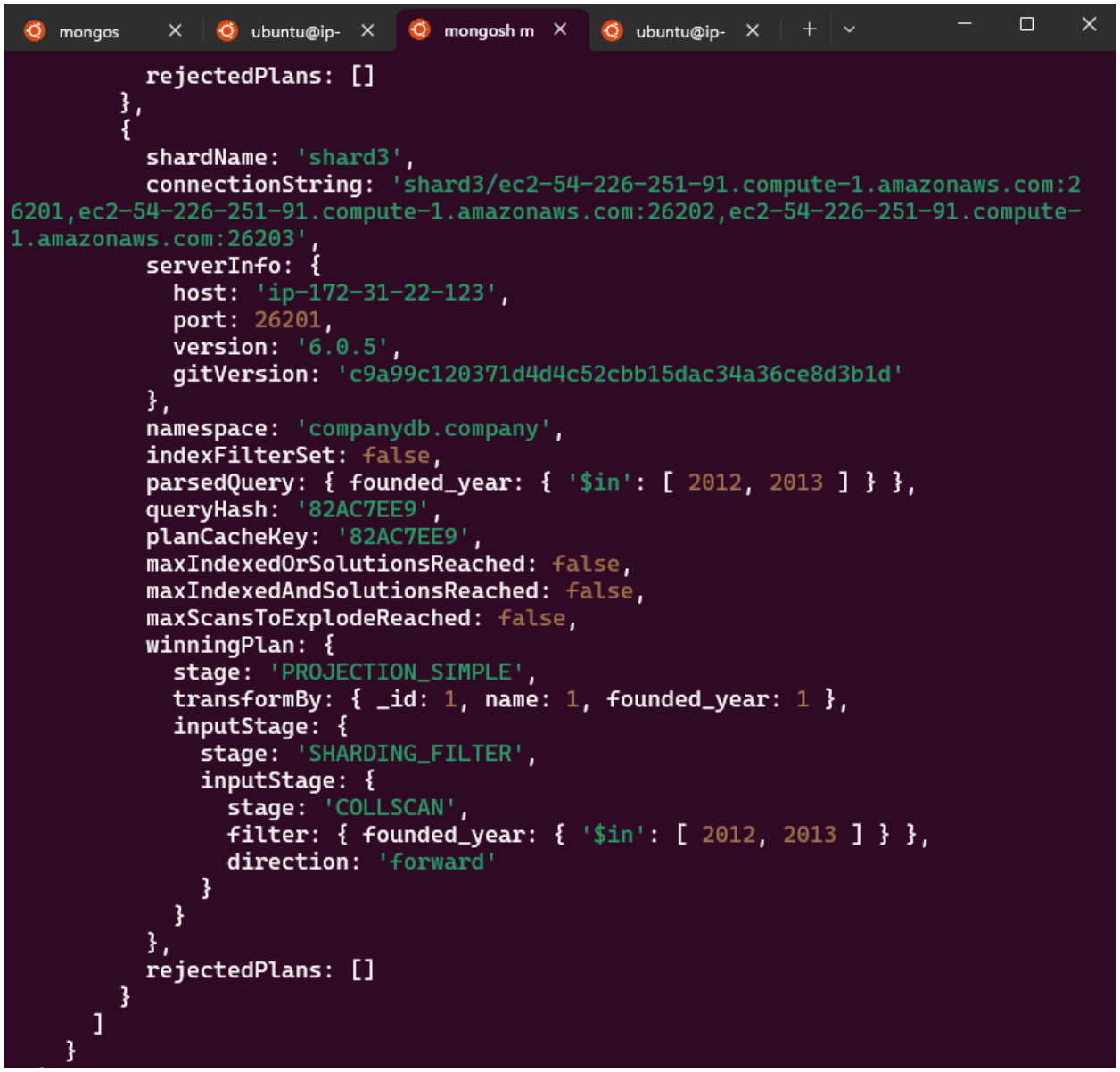
Result:


mongos
ubuntu@ip-
mongosh m
ubuntu@ip-
Ubuntu: mongosh mongodb://
ec2-54-226-173-1.compute-1.amazonaws.com:27017/?
directConnection=true
ctrl+alt+3
name: 'CPXi',
founded_year: 2
}
]
Type "it" for more
[direct: mongos] companydb> db.company.find({founded_year: {$in: [2013,2012]}},
{_id: 1, name: 1, founded_year: 1})
[
{
_id: ObjectId("52cdef7c4bab8bd6752983b6"),
name: 'Mobiluck',
founded_year: 2012
},
{
_id: ObjectId("52cdef7c4bab8bd675298509"),
name: 'Skydeck',
founded_year: 2012
},
{
_id: ObjectId("52cdef7d4bab8bd6752992f7"),
name: 'Bling Easy',
founded_year: 2012
},
{
_id: ObjectId("52cdef7d4bab8bd675299609"),
name: 'Non-Member Films',
founded_year: 2012
},
{
_id: ObjectId("52cdef7d4bab8bd675299ed8"),
name: 'PriceHub',
founded_year: 2012
},
{
_id: ObjectId("52cdef7e4bab8bd67529b254"),
name: 'Pikk',
founded_year: 2013
},
{
_id: ObjectId("52cdef7e4bab8bd67529b58d"),
name: 'Pict',
founded_year: 2012
},
{
_id: ObjectId("52cdef7e4bab8bd67529b681"),
name: 'Plug in SEO',
founded_year: 2012
},
{
_id: ObjectId("52cdef7e4bab8bd67529b8ab"),
name: 'Veewow',
founded_year: 2012
},
{
_id: ObjectId("52cdef7e4bab8bd67529bd37"),
name: 'Red Ambiental',
founded_year: 2012
},
{
_id: ObjectId("52cdef7c4bab8bd675297fb1"),
name: 'headr',
founded_year: 2012
},
{
_id: ObjectId("52cdef7c4bab8bd6752981a0"),
name: 'Pinger',

Shards served the query: shard 1, shard 2, shard 3

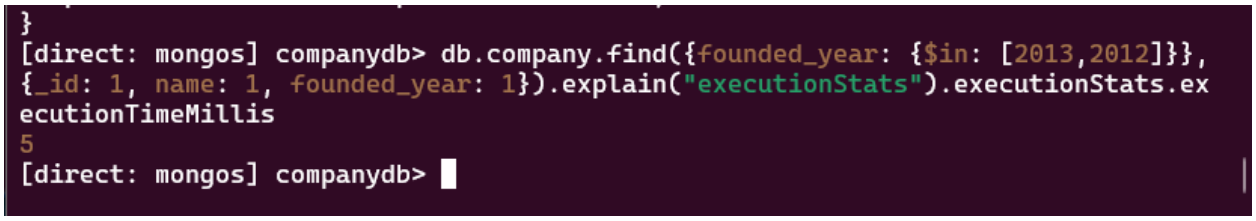
db.company.find({founded_year: {$in: [2013,2012]}},{_id: 1, name: 1, founded_year: 1}).explain()


mongos
ubuntu@ip-
mongosh m
ubuntu@ip-
Ubuntu: ubuntu@ip-172-31-26-40: ~
ctrl+alt+2
shardName: 'shard1',
connectionString: 'shard1/ec2-34-230-29-212.compute-1.amazonaws.com:26001,ec2-34-230-29-212.compute-1.amazonaws.com:26002,ec2-34-230-29-212.compute-1.amazonaws.com:26003',
serverInfo: {
host: 'ip-172-31-19-219',
port: 26001,
version: '6.0.5',
gitVersion: 'c9a99c120371d4d4c52cbb15dac34a36ce8d3b1d'
},
namespace: 'companydb.company',
indexFilterSet: false,
parsedQuery: { founded_year: { '$in': [2012, 2013] } },
queryHash: '82AC7EE9',
planCacheKey: '82AC7EE9',
maxIndexedOrSolutionsReached: false,
maxIndexedAndSolutionsReached: false,
maxScansToExplodeReached: false,
winningPlan: {
stage: 'PROJECTION_SIMPLE',
transformBy: { _id: 1, name: 1, founded_year: 1 },
inputStage: {
stage: 'SHARDING_FILTER',
inputStage: {
stage: 'COLLSCAN',
filter: { founded_year: { '$in': [2012, 2013] } },
direction: 'forward'
}
}
},
rejectedPlans: []
},
{
shardName: 'shard2',
connectionString: 'shard2/ec2-34-226-213-111.compute-1.amazonaws.com:26101,ec2-34-226-213-111.compute-1.amazonaws.com:26102,ec2-34-226-213-111.compute-1.amazonaws.com:26103',
serverInfo: {
host: 'ip-172-31-31-11',
port: 26101,
version: '6.0.5',
gitVersion: 'c9a99c120371d4d4c52cbb15dac34a36ce8d3b1d'
},



Execution Time: 5

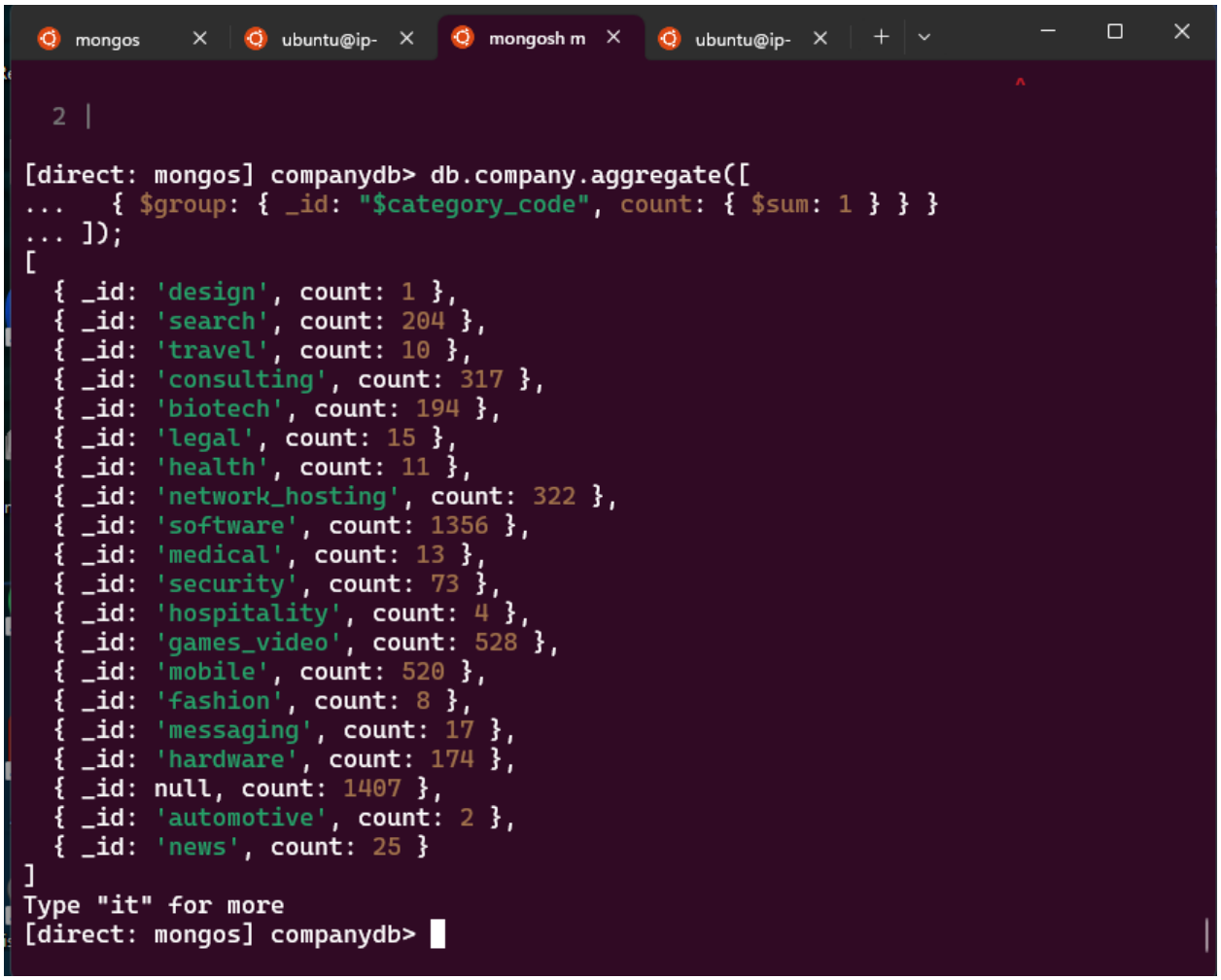
db.company.find({founded_year: {$in: [2013,2012]}},{_id: 1, name: 1, founded_year: 1}).explain("executionStats").executionStats.executionTimeMillis



12.4: Find number of companies in each category_code:

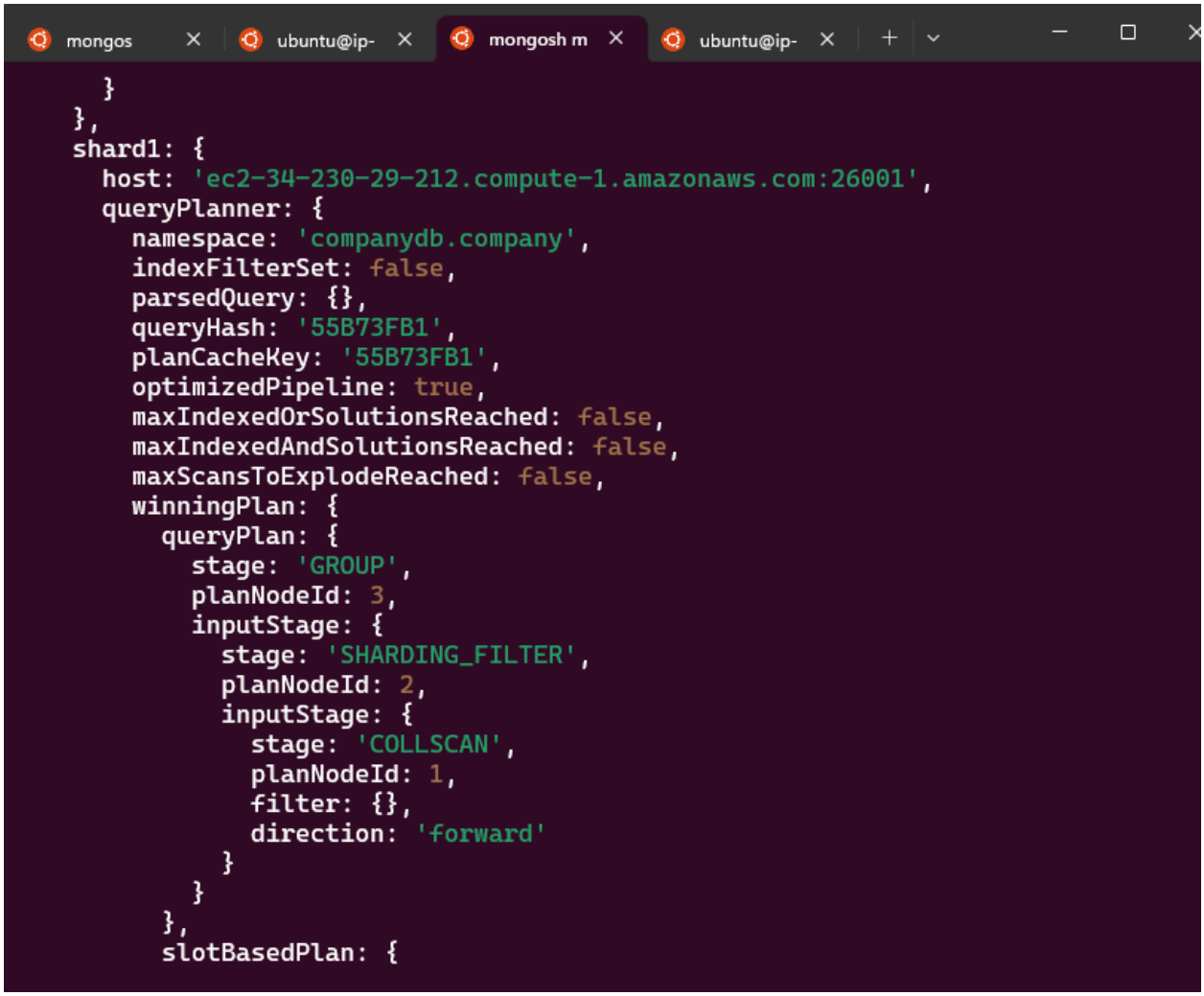
Result:

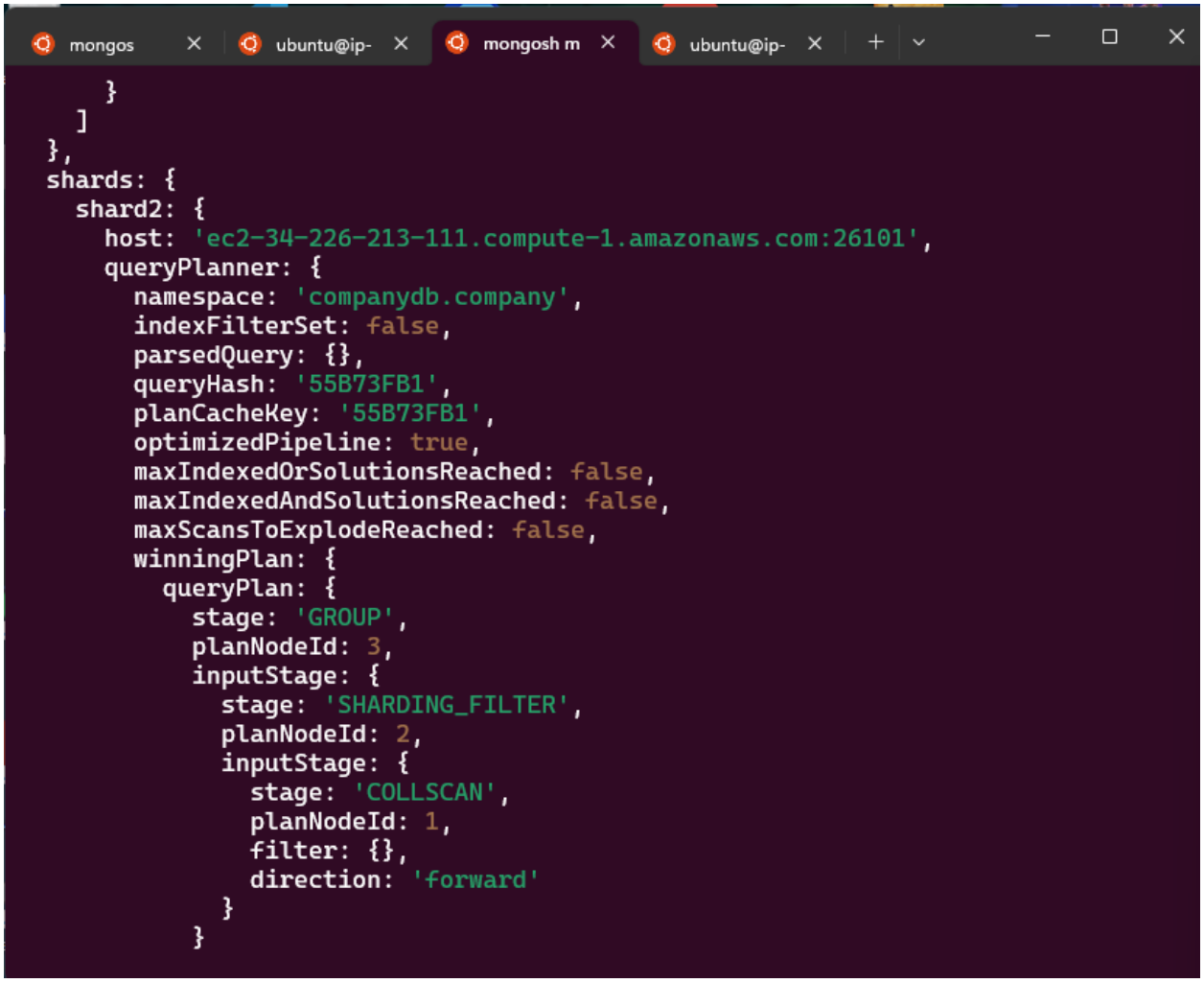
db.company.aggregate([{ $group: { _id: "$category_code", count: { $sum: 1 } } }]);



Shards served the query: shard 1, shard 2, shard 3

db.company.aggregate([{ $group: { _id: "$category_code", count: { $sum: 1 } } }]).explain()





Execution Time: 176ms

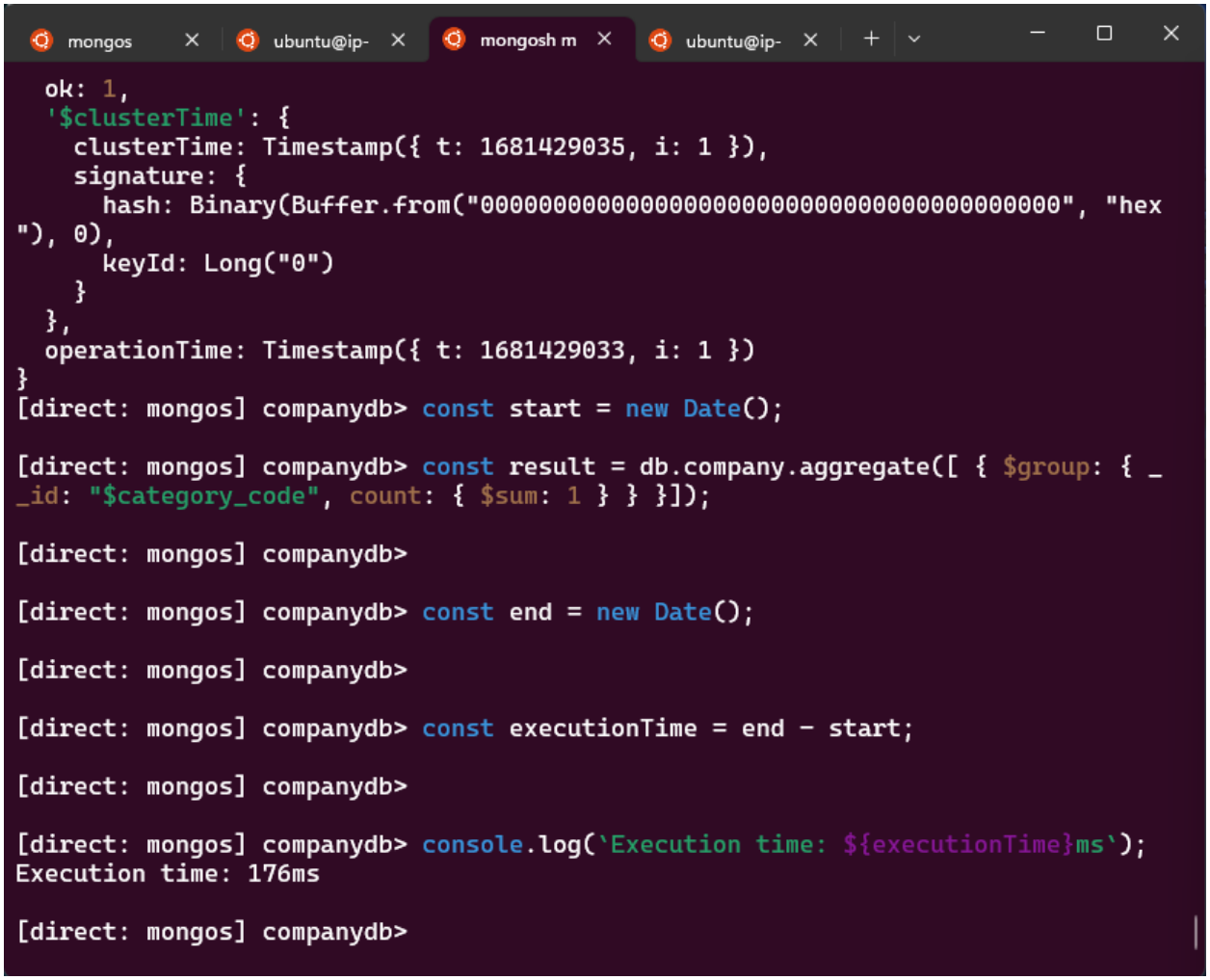
const start = new Date();

const result = *db.company.aggregate([{ $group: { _id: "$category_code", count: { $sum: 1 } } }]);*

const end = new Date();

const executionTime = end - start;

console.log(`Execution time: ${executionTime}ms`);



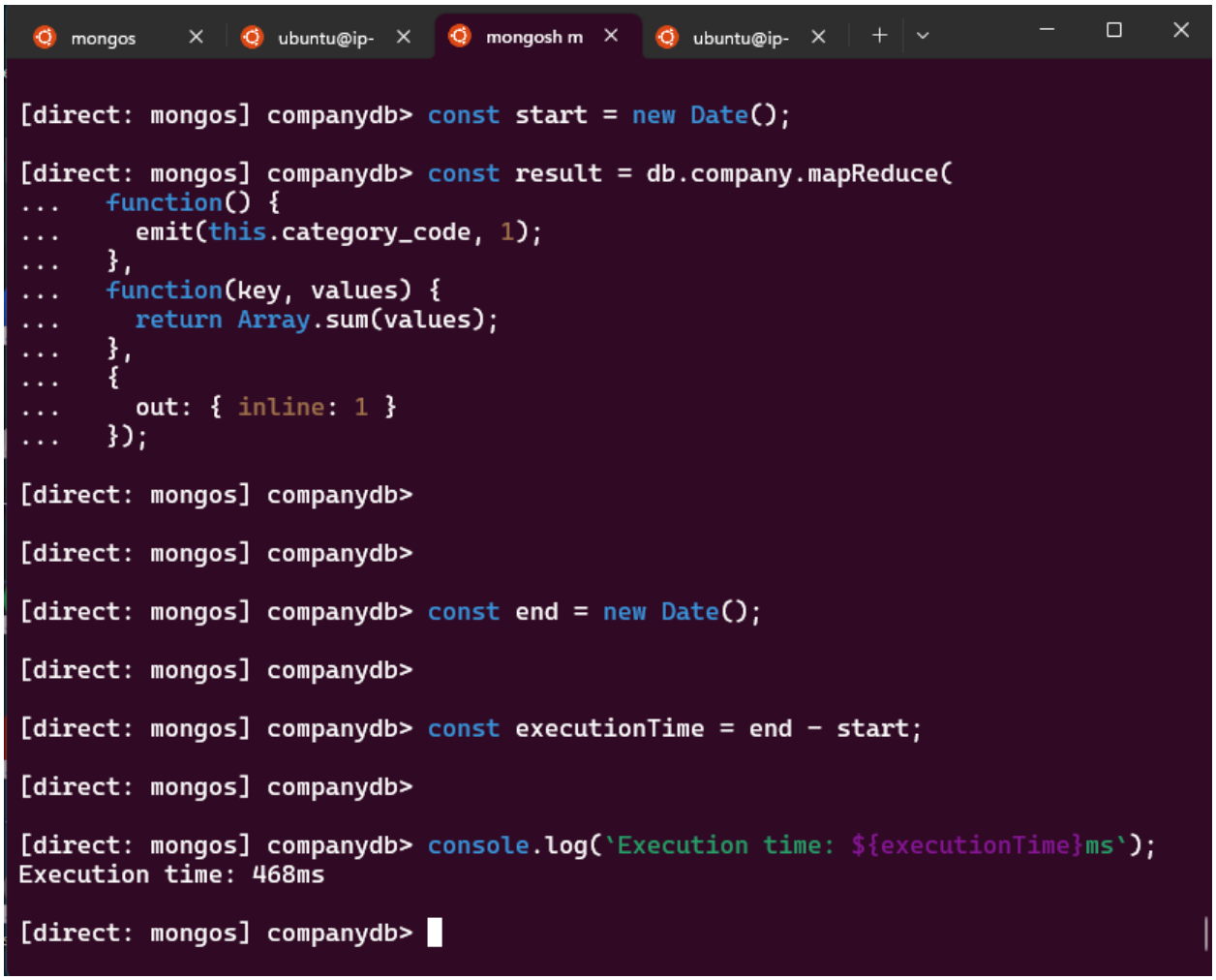
12.5: Map Reduce: Find numbers of companies in each category_code:

```
db.company.mapReduce(
 function() {
  emit(this.category_code, 1);
 },
 function(key, values) {
  return Array.sum(values);
 },
 {
  out: { inline: 1 }
 });
```

Execution Time: 468ms

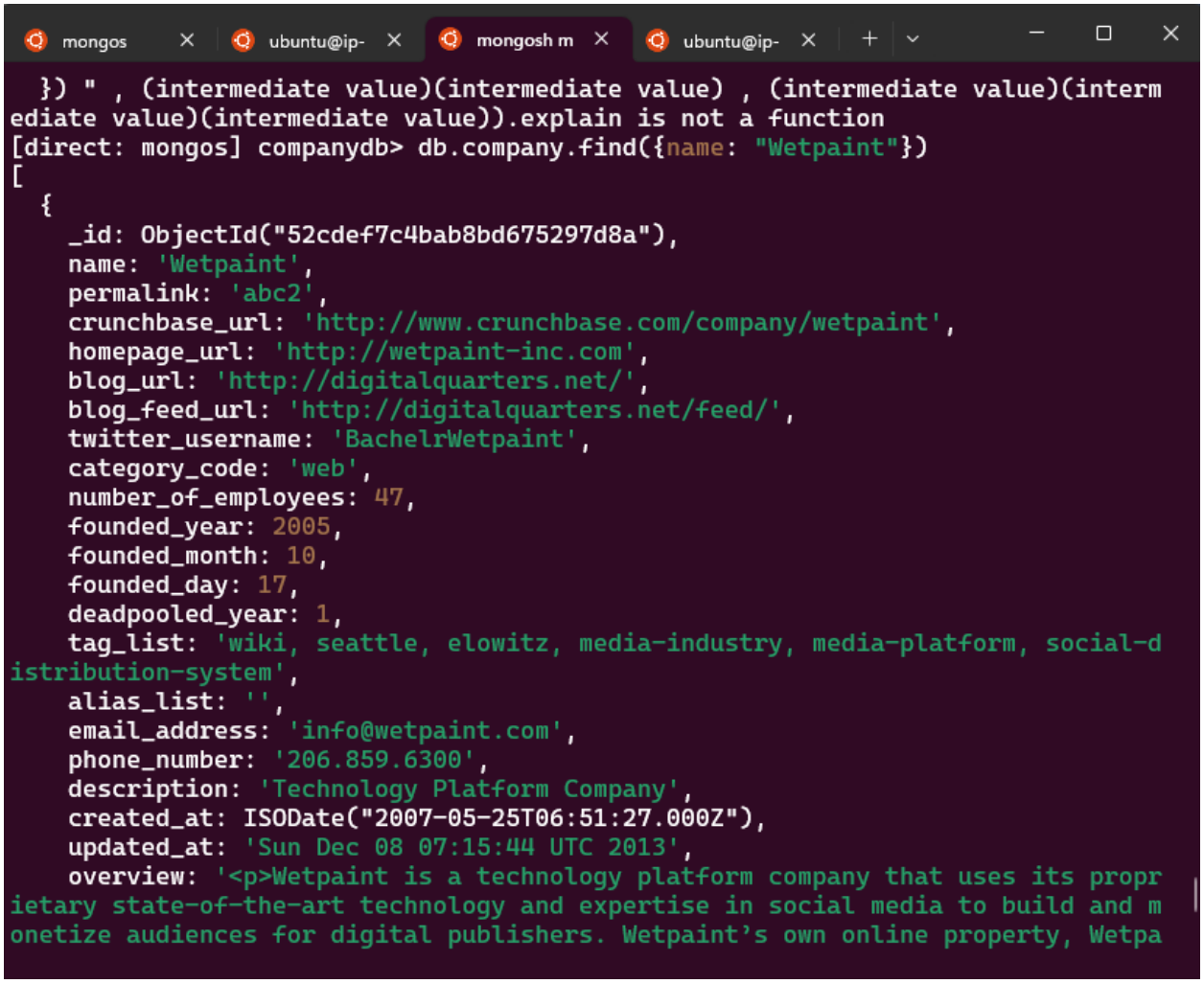
```
const start = new Date();
const result = db.company.mapReduce(
  function() {
    emit(this.category_code, 1);
  },
  function(key, values) {
    return Array.sum(values);
  },
  {
```

```
  out: { inline: 1 }
});

const end = new Date();

const executionTime = end - start;

console.log(`Execution time: ${executionTime}ms`);
```



12.6: Update a company named 'Wetpaint', founded_year to become 2012

Before update:



After update:

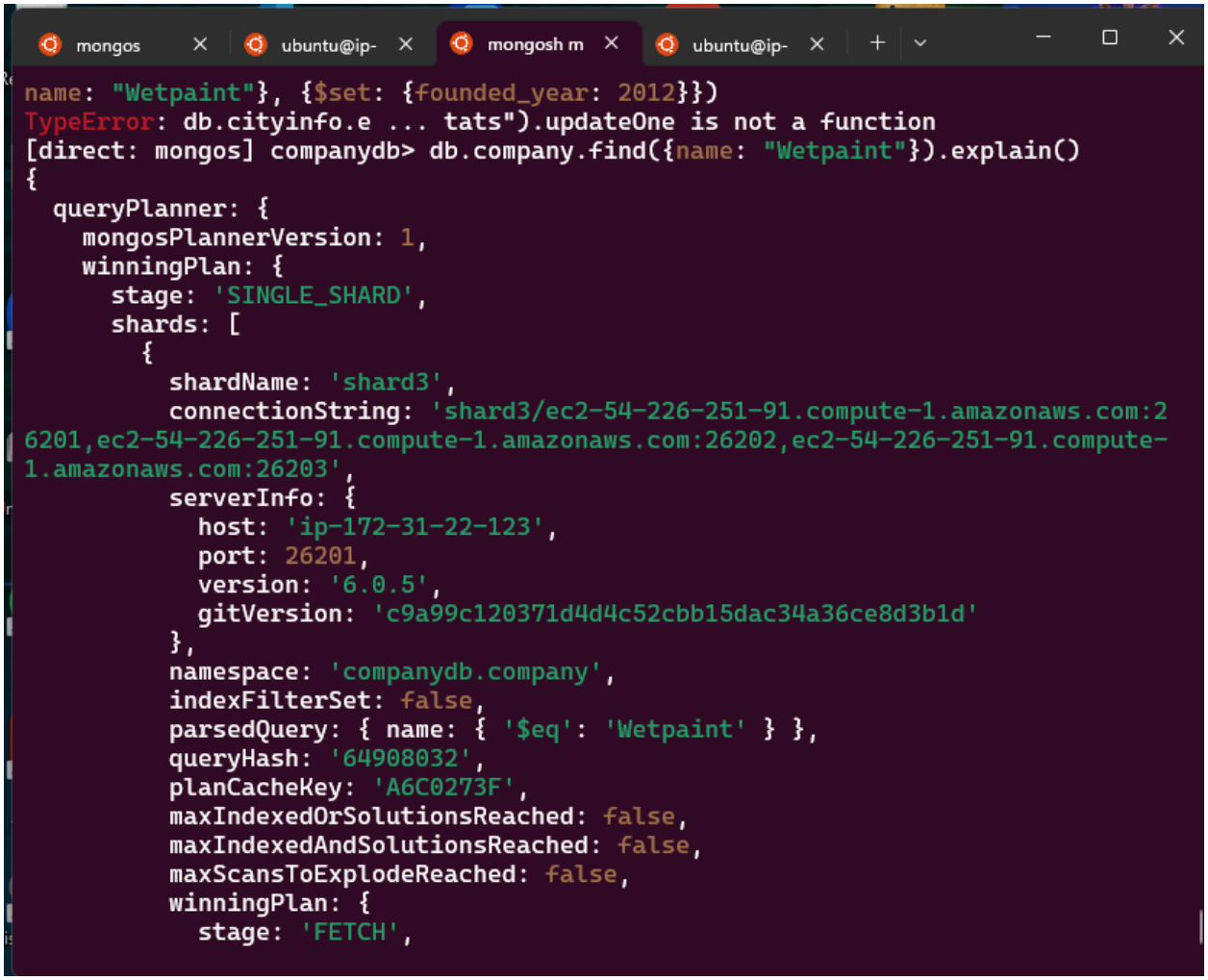
db.company.updateOne({name: "Wetpaint"}, {$set: {founded_year: 2012}})

```
[direct: mongos] companydb> db.company.updateOne({name: "Wetpaint"}, {$set: {fo
unded_year: 2012}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
[direct: mongos] companydb>
```

mongos | ubuntu@ip- | mongosh m | ubuntu@ip-

```
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
[direct: mongos] companydb> db.company.find({name: "Wetpaint"})
[
  {
    _id: ObjectId("52cdef7c4bab8bd675297d8a"),
    name: 'Wetpaint',
    permalink: 'abc2',
    crunchbase_url: 'http://www.crunchbase.com/company/wetpaint',
    homepage_url: 'http://wetpaint-inc.com',
    blog_url: 'http://digitalquarters.net/',
    blog_feed_url: 'http://digitalquarters.net/feed/',
    twitter_username: 'BachelrWetpaint',
    category_code: 'web',
    number_of_employees: 47,
    founded_year: 2012,
    founded_month: 10,
    founded_day: 17,
    deadpooled_year: 1,
    tag_list: 'wiki, seattle, elowitz, media-industry, media-platform, social-d
istribution-system',
    alias_list: '',
    email_address: 'info@wetpaint.com',
    phone_number: '206.859.6300',
    description: 'Technology Platform Company',
    created_at: ISODate("2007-05-25T06:51:27.000Z"),
    updated_at: 'Sun Dec 08 07:15:44 UTC 2013',
    overview: '<p>Wetpaint is a technology platform company that uses its propr
```

Shard served the query: Shard 3

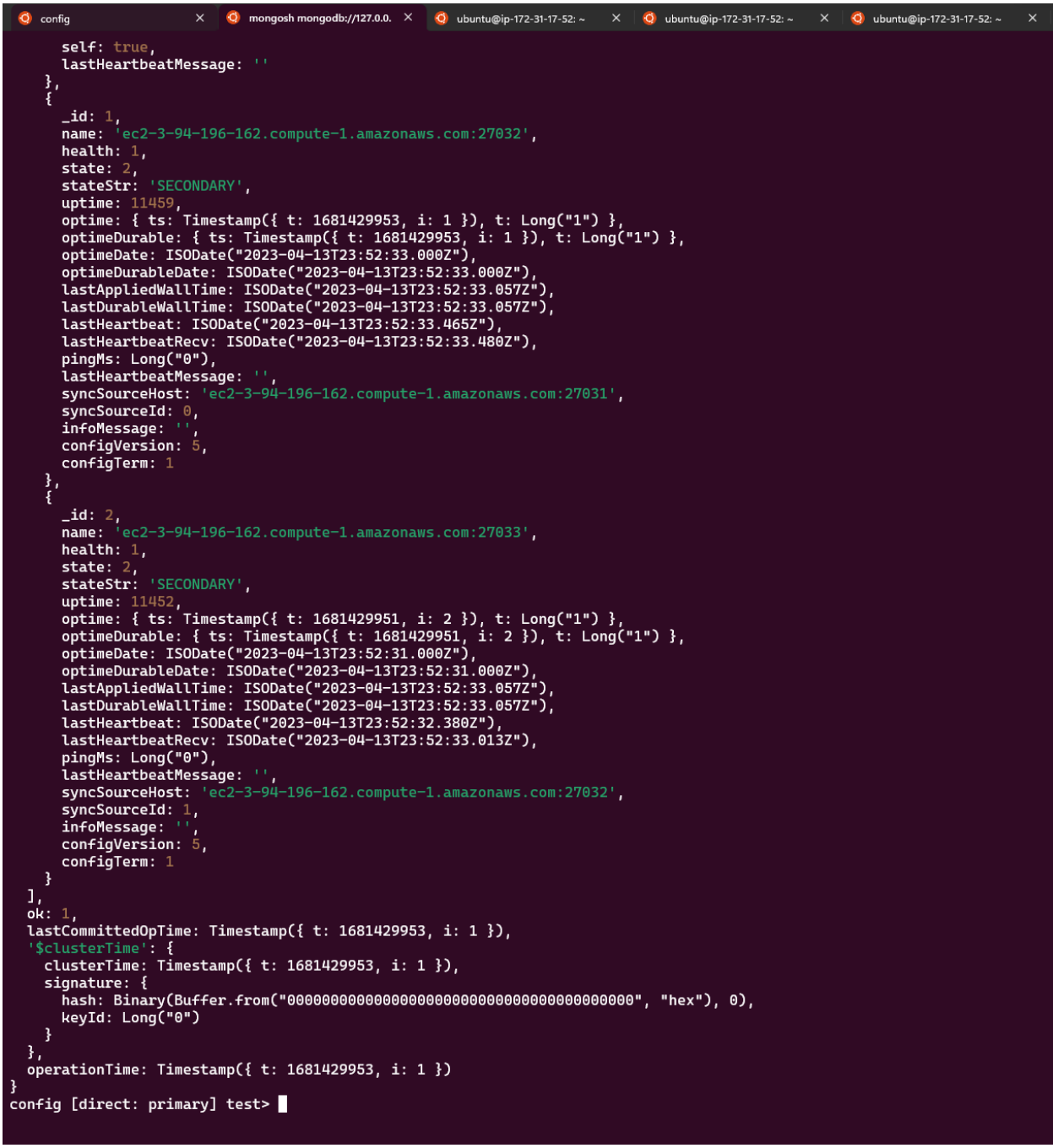


13. rs.status()

a. Config:

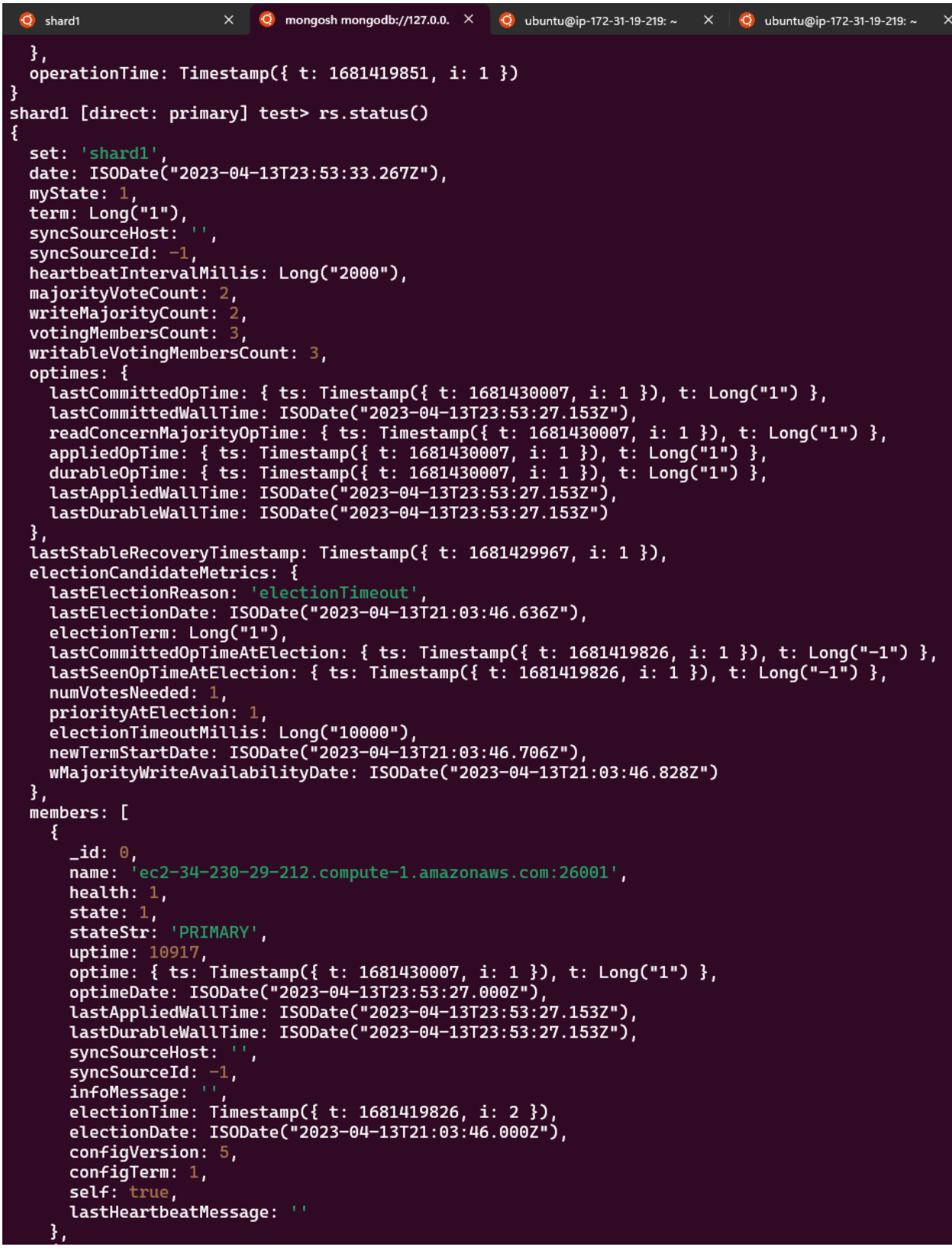
config | mongosh mongodb://127.0.0. | ubuntu@ip-172-31-17-52: ~ | ubuntu@ip-172-31-17-52: ~ | ubuntu@ip-172-31-17-52: ~

```
}
config [direct: primary] test>
config [direct: primary] test> rs.status()
{
  set: 'config',
  date: ISODate("2023-04-13T23:52:33.595Z"),
  myState: 1,
  term: Long("1"),
  syncSourceHost: '',
  syncSourceId: -1,
  configsvr: true,
  heartbeatIntervalMillis: Long("2000"),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1681429953, i: 1 }), t: Long("1") },
    lastCommittedWallTime: ISODate("2023-04-13T23:52:33.057Z"),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1681429953, i: 1 }), t: Long("1") },
    appliedOpTime: { ts: Timestamp({ t: 1681429953, i: 1 }), t: Long("1") },
    durableOpTime: { ts: Timestamp({ t: 1681429953, i: 1 }), t: Long("1") },
    lastAppliedWallTime: ISODate("2023-04-13T23:52:33.057Z"),
    lastDurableWallTime: ISODate("2023-04-13T23:52:33.057Z")
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1681429904, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate("2023-04-13T20:39:39.404Z"),
    electionTerm: Long("1"),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1681418379, i: 1 }), t: Long("-1") },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1681418379, i: 1 }), t: Long("-1") },
    numVotesNeeded: 1,
    priorityAtElection: 1,
    electionTimeoutMillis: Long("10000"),
    newTermStartDate: ISODate("2023-04-13T20:39:39.458Z"),
    wMajorityWriteAvailabilityDate: ISODate("2023-04-13T20:39:39.664Z")
  },
  members: [
    {
      _id: 0,
      name: 'ec2-3-94-196-162.compute-1.amazonaws.com:27031',
      health: 1,
      state: 1,
      stateStr: 'PRIMARY',
      uptime: 11682,
      optime: { ts: Timestamp({ t: 1681429953, i: 1 }), t: Long("1") },
      optimeDate: ISODate("2023-04-13T23:52:33.000Z"),
      lastAppliedWallTime: ISODate("2023-04-13T23:52:33.057Z"),
      lastDurableWallTime: ISODate("2023-04-13T23:52:33.057Z"),
      syncSourceHost: '',
      syncSourceId: -1,
      infoMessage: '',
      electionTime: Timestamp({ t: 1681418379, i: 2 }),
      electionDate: ISODate("2023-04-13T20:39:39.000Z"),
      configVersion: 5,
      configTerm: 1,
      self: true,
      lastHeartbeatMessage: ''
    },
```

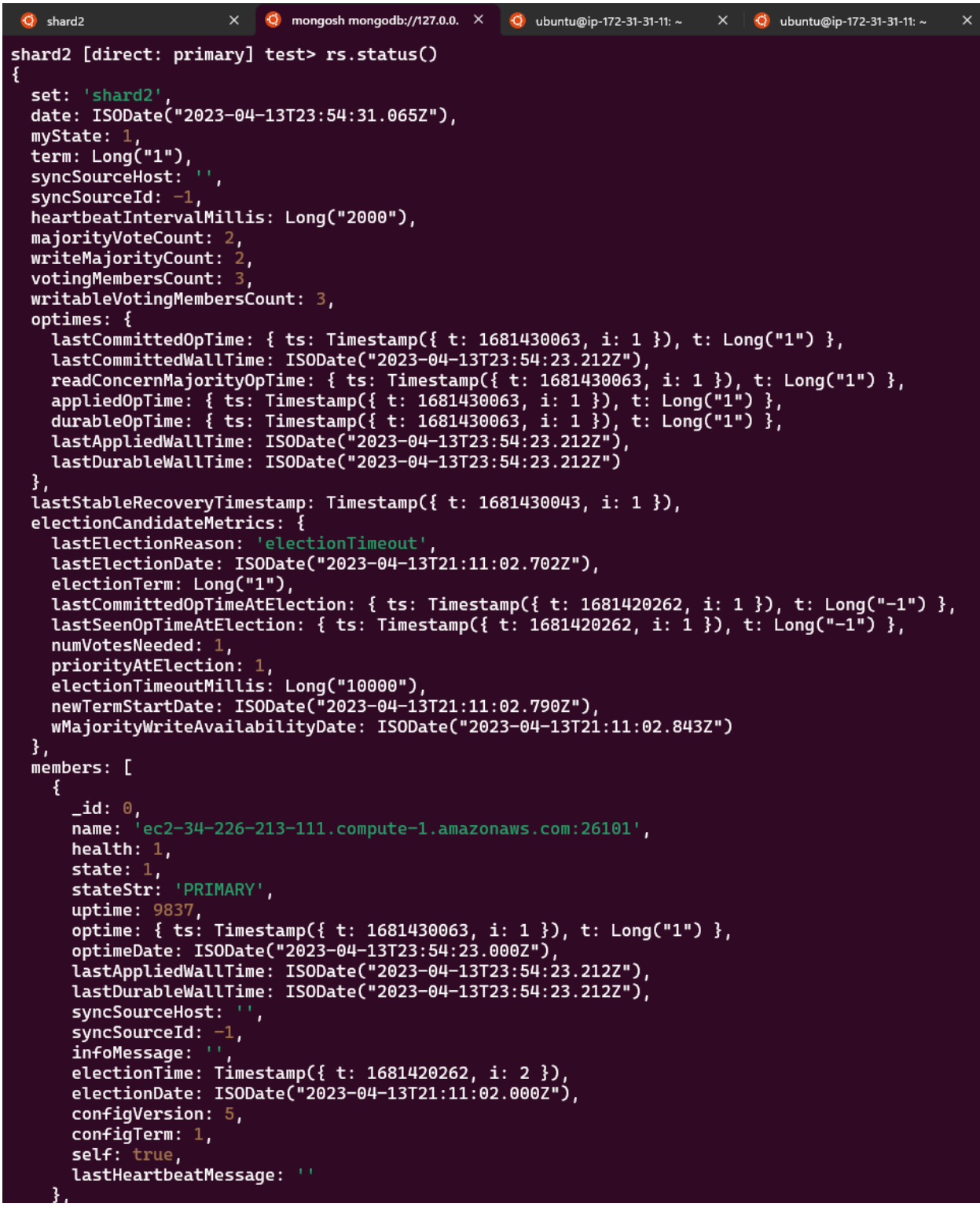

config
mongosh mongodb://127.0.0.
ubuntu@ip-172-31-17-52: ~
ubuntu@ip-172-31-17-52: ~
ubuntu@ip-172-31-17-52: ~
self: true,
lastHeartbeatMessage: ''
},
{
_id: 1,
name: 'ec2-3-94-196-162.compute-1.amazonaws.com:27032',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 11459,
optime: { ts: Timestamp({ t: 1681429953, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681429953, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:52:33.000Z"),
optimeDurableDate: ISODate("2023-04-13T23:52:33.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:52:33.057Z"),
lastDurableWallTime: ISODate("2023-04-13T23:52:33.057Z"),
lastHeartbeat: ISODate("2023-04-13T23:52:33.465Z"),
lastHeartbeatRecv: ISODate("2023-04-13T23:52:33.480Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-3-94-196-162.compute-1.amazonaws.com:27031',
syncSourceId: 0,
infoMessage: '',
configVersion: 5,
configTerm: 1
},
{
_id: 2,
name: 'ec2-3-94-196-162.compute-1.amazonaws.com:27033',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 11452,
optime: { ts: Timestamp({ t: 1681429951, i: 2 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681429951, i: 2 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:52:31.000Z"),
optimeDurableDate: ISODate("2023-04-13T23:52:31.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:52:33.057Z"),
lastDurableWallTime: ISODate("2023-04-13T23:52:33.057Z"),
lastHeartbeat: ISODate("2023-04-13T23:52:32.380Z"),
lastHeartbeatRecv: ISODate("2023-04-13T23:52:33.013Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-3-94-196-162.compute-1.amazonaws.com:27032',
syncSourceId: 1,
infoMessage: '',
configVersion: 5,
configTerm: 1
}
],
ok: 1,
lastCommittedOpTime: Timestamp({ t: 1681429953, i: 1 }),
'$clusterTime': {
clusterTime: Timestamp({ t: 1681429953, i: 1 }),
signature: {
hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex"), 0),
keyId: Long("0")
}
},
operationTime: Timestamp({ t: 1681429953, i: 1 })
}
config [direct: primary] test>

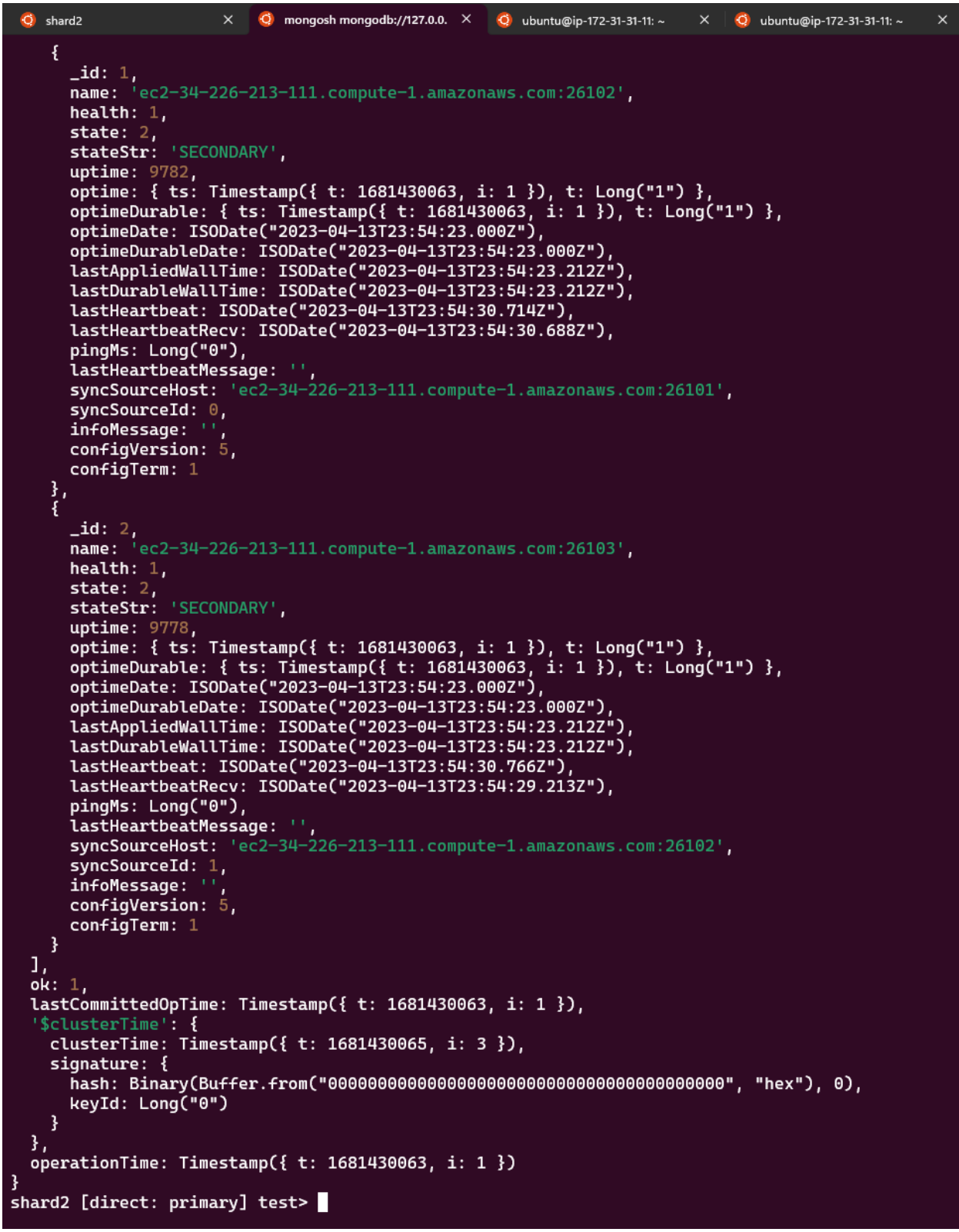
b. Shard 1:


shard1
mongosh mongodb://127.0.0.
ubuntu@ip-172-31-19-219: ~
ubuntu@ip-172-31-19-219: ~
{
_id: 1,
name: 'ec2-34-230-29-212.compute-1.amazonaws.com:26002',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 10170,
optime: { ts: Timestamp({ t: 1681430007, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681430007, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:53:27.000Z"),
optimeDurableDate: ISODate("2023-04-13T23:53:27.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:53:27.153Z"),
lastDurableWallTime: ISODate("2023-04-13T23:53:27.153Z"),
lastHeartbeat: ISODate("2023-04-13T23:53:32.431Z"),
lastHeartbeatRecv: ISODate("2023-04-13T23:53:32.462Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-34-230-29-212.compute-1.amazonaws.com:26001',
syncSourceId: 0,
infoMessage: '',
configVersion: 5,
configTerm: 1
},
{
_id: 2,
name: 'ec2-34-230-29-212.compute-1.amazonaws.com:26003',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 10163,
optime: { ts: Timestamp({ t: 1681430007, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681430007, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:53:27.000Z"),
optimeDurableDate: ISODate("2023-04-13T23:53:27.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:53:27.153Z"),
lastDurableWallTime: ISODate("2023-04-13T23:53:27.153Z"),
lastHeartbeat: ISODate("2023-04-13T23:53:32.359Z"),
lastHeartbeatRecv: ISODate("2023-04-13T23:53:32.907Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-34-230-29-212.compute-1.amazonaws.com:26002',
syncSourceId: 1,
infoMessage: '',
configVersion: 5,
configTerm: 1
}
],
ok: 1,
lastCommittedOpTime: Timestamp({ t: 1681430007, i: 1 }),
'$clusterTime': {
clusterTime: Timestamp({ t: 1681430007, i: 2 }),
signature: {
hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex"), 0),
keyId: Long("0")
}
},
operationTime: Timestamp({ t: 1681430007, i: 1 })
}
shard1 [direct: primary] test>

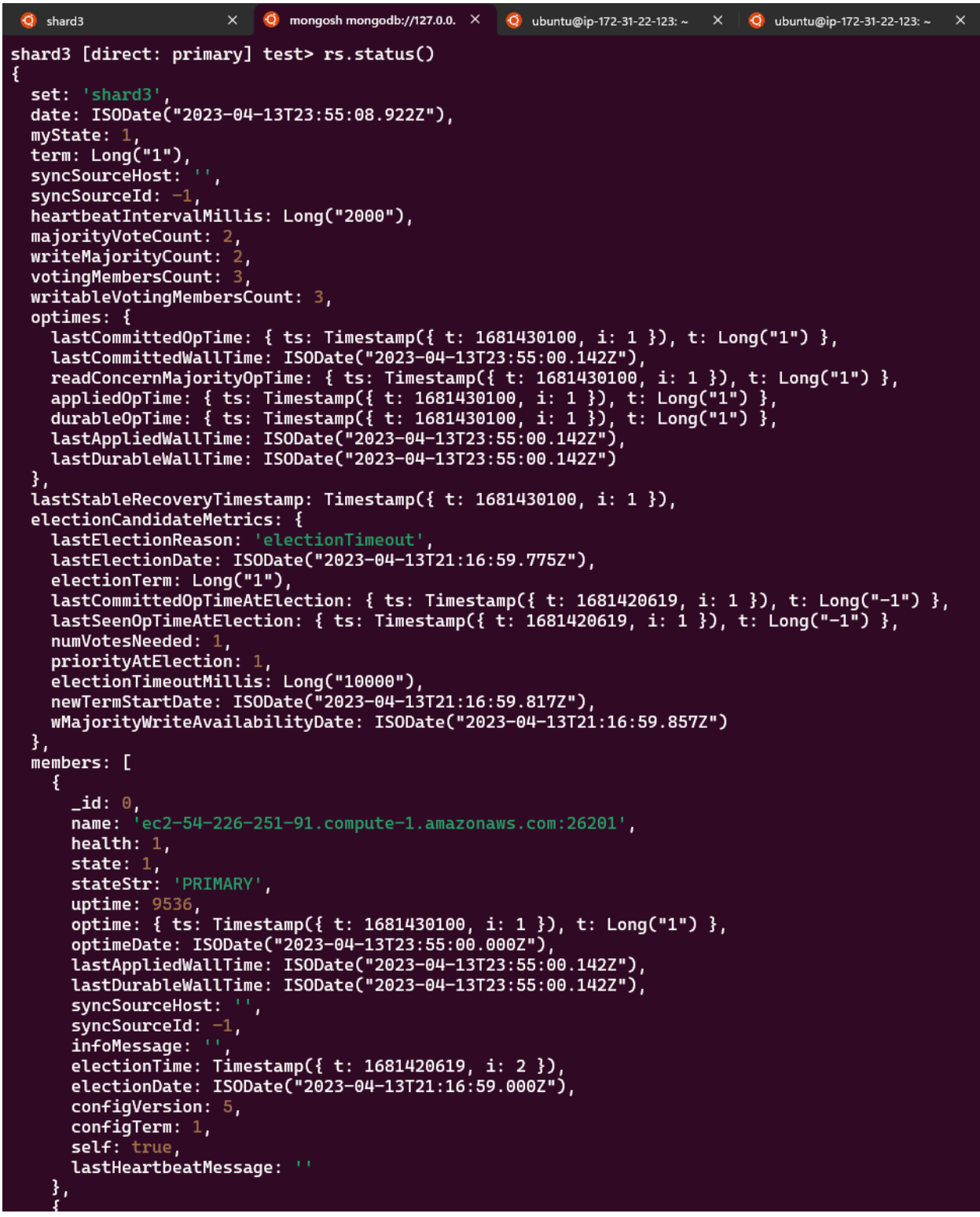

shard1
mongosh mongodb://127.0.0.
ubuntu@ip-172-31-19-219: ~
ubuntu@ip-172-31-19-219: ~
},
operationTime: Timestamp({ t: 1681419851, i: 1 })
}
shard1 [direct: primary] test> rs.status()
{
set: 'shard1',
date: ISODate("2023-04-13T23:53:33.267Z"),
myState: 1,
term: Long("1"),
syncSourceHost: '',
syncSourceId: -1,
heartbeatIntervalMillis: Long("2000"),
majorityVoteCount: 2,
writeMajorityCount: 2,
votingMembersCount: 3,
writableVotingMembersCount: 3,
optimes: {
lastCommittedOpTime: { ts: Timestamp({ t: 1681430007, i: 1 }), t: Long("1") },
lastCommittedWallTime: ISODate("2023-04-13T23:53:27.153Z"),
readConcernMajorityOpTime: { ts: Timestamp({ t: 1681430007, i: 1 }), t: Long("1") },
appliedOpTime: { ts: Timestamp({ t: 1681430007, i: 1 }), t: Long("1") },
durableOpTime: { ts: Timestamp({ t: 1681430007, i: 1 }), t: Long("1") },
lastAppliedWallTime: ISODate("2023-04-13T23:53:27.153Z"),
lastDurableWallTime: ISODate("2023-04-13T23:53:27.153Z")
},
lastStableRecoveryTimestamp: Timestamp({ t: 1681429967, i: 1 }),
electionCandidateMetrics: {
lastElectionReason: 'electionTimeout',
lastElectionDate: ISODate("2023-04-13T21:03:46.636Z"),
electionTerm: Long("1"),
lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1681419826, i: 1 }), t: Long("-1") },
lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1681419826, i: 1 }), t: Long("-1") },
numVotesNeeded: 1,
priorityAtElection: 1,
electionTimeoutMillis: Long("10000"),
newTermStartDate: ISODate("2023-04-13T21:03:46.706Z"),
wMajorityWriteAvailabilityDate: ISODate("2023-04-13T21:03:46.828Z")
},
members: [
{
_id: 0,
name: 'ec2-34-230-29-212.compute-1.amazonaws.com:26001',
health: 1,
state: 1,
stateStr: 'PRIMARY',
uptime: 10917,
optime: { ts: Timestamp({ t: 1681430007, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:53:27.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:53:27.153Z"),
lastDurableWallTime: ISODate("2023-04-13T23:53:27.153Z"),
syncSourceHost: '',
syncSourceId: -1,
infoMessage: '',
electionTime: Timestamp({ t: 1681419826, i: 2 }),
electionDate: ISODate("2023-04-13T21:03:46.000Z"),
configVersion: 5,
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},

c. Shard 2:


shard2
mongosh mongodb://127.0.0.
ubuntu@ip-172-31-31-11: ~
ubuntu@ip-172-31-31-11: ~
shard2 [direct: primary] test> rs.status()
{
set: 'shard2',
date: ISODate("2023-04-13T23:54:31.065Z"),
myState: 1,
term: Long("1"),
syncSourceHost: '',
syncSourceId: -1,
heartbeatIntervalMillis: Long("2000"),
majorityVoteCount: 2,
writeMajorityCount: 2,
votingMembersCount: 3,
writableVotingMembersCount: 3,
optimes: {
lastCommittedOpTime: { ts: Timestamp({ t: 1681430063, i: 1 }), t: Long("1") },
lastCommittedWallTime: ISODate("2023-04-13T23:54:23.212Z"),
readConcernMajorityOpTime: { ts: Timestamp({ t: 1681430063, i: 1 }), t: Long("1") },
appliedOpTime: { ts: Timestamp({ t: 1681430063, i: 1 }), t: Long("1") },
durableOpTime: { ts: Timestamp({ t: 1681430063, i: 1 }), t: Long("1") },
lastAppliedWallTime: ISODate("2023-04-13T23:54:23.212Z"),
lastDurableWallTime: ISODate("2023-04-13T23:54:23.212Z")
},
lastStableRecoveryTimestamp: Timestamp({ t: 1681430043, i: 1 }),
electionCandidateMetrics: {
lastElectionReason: 'electionTimeout',
lastElectionDate: ISODate("2023-04-13T21:11:02.702Z"),
electionTerm: Long("1"),
lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1681420262, i: 1 }), t: Long("-1") },
lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1681420262, i: 1 }), t: Long("-1") },
numVotesNeeded: 1,
priorityAtElection: 1,
electionTimeoutMillis: Long("10000"),
newTermStartDate: ISODate("2023-04-13T21:11:02.790Z"),
wMajorityWriteAvailabilityDate: ISODate("2023-04-13T21:11:02.843Z")
},
members: [
{
_id: 0,
name: 'ec2-34-226-213-111.compute-1.amazonaws.com:26101',
health: 1,
state: 1,
stateStr: 'PRIMARY',
uptime: 9837,
optime: { ts: Timestamp({ t: 1681430063, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:54:23.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:54:23.212Z"),
lastDurableWallTime: ISODate("2023-04-13T23:54:23.212Z"),
syncSourceHost: '',
syncSourceId: -1,
infoMessage: '',
electionTime: Timestamp({ t: 1681420262, i: 2 }),
electionDate: ISODate("2023-04-13T21:11:02.000Z"),
configVersion: 5,
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},


shard2
mongosh mongodb://127.0.0.
ubuntu@ip-172-31-31-11: ~
ubuntu@ip-172-31-31-11: ~
{
_id: 1,
name: 'ec2-34-226-213-111.compute-1.amazonaws.com:26102',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 9782,
optime: { ts: Timestamp({ t: 1681430063, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681430063, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:54:23.000Z"),
optimeDurableDate: ISODate("2023-04-13T23:54:23.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:54:23.212Z"),
lastDurableWallTime: ISODate("2023-04-13T23:54:23.212Z"),
lastHeartbeat: ISODate("2023-04-13T23:54:30.714Z"),
lastHeartbeatRecv: ISODate("2023-04-13T23:54:30.688Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-34-226-213-111.compute-1.amazonaws.com:26101',
syncSourceId: 0,
infoMessage: '',
configVersion: 5,
configTerm: 1
},
{
_id: 2,
name: 'ec2-34-226-213-111.compute-1.amazonaws.com:26103',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 9778,
optime: { ts: Timestamp({ t: 1681430063, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681430063, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:54:23.000Z"),
optimeDurableDate: ISODate("2023-04-13T23:54:23.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:54:23.212Z"),
lastDurableWallTime: ISODate("2023-04-13T23:54:23.212Z"),
lastHeartbeat: ISODate("2023-04-13T23:54:30.766Z"),
lastHeartbeatRecv: ISODate("2023-04-13T23:54:29.213Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-34-226-213-111.compute-1.amazonaws.com:26102',
syncSourceId: 1,
infoMessage: '',
configVersion: 5,
configTerm: 1
}
],
ok: 1,
lastCommittedOpTime: Timestamp({ t: 1681430063, i: 1 }),
'$clusterTime': {
clusterTime: Timestamp({ t: 1681430065, i: 3 }),
signature: {
hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex"), 0),
keyId: Long("0")
}
},
operationTime: Timestamp({ t: 1681430063, i: 1 })
}
shard2 [direct: primary] test>

d. Shard 3:


shard3
mongosh mongodb://127.0.0.
ubuntu@ip-172-31-22-123: ~
ubuntu@ip-172-31-22-123: ~
shard3 [direct: primary] test> rs.status()
{
set: 'shard3',
date: ISODate("2023-04-13T23:55:08.922Z"),
myState: 1,
term: Long("1"),
syncSourceHost: '',
syncSourceId: -1,
heartbeatIntervalMillis: Long("2000"),
majorityVoteCount: 2,
writeMajorityCount: 2,
votingMembersCount: 3,
writableVotingMembersCount: 3,
optimes: {
lastCommittedOpTime: { ts: Timestamp({ t: 1681430100, i: 1 }), t: Long("1") },
lastCommittedWallTime: ISODate("2023-04-13T23:55:00.142Z"),
readConcernMajorityOpTime: { ts: Timestamp({ t: 1681430100, i: 1 }), t: Long("1") },
appliedOpTime: { ts: Timestamp({ t: 1681430100, i: 1 }), t: Long("1") },
durableOpTime: { ts: Timestamp({ t: 1681430100, i: 1 }), t: Long("1") },
lastAppliedWallTime: ISODate("2023-04-13T23:55:00.142Z"),
lastDurableWallTime: ISODate("2023-04-13T23:55:00.142Z")
},
lastStableRecoveryTimestamp: Timestamp({ t: 1681430100, i: 1 }),
electionCandidateMetrics: {
lastElectionReason: 'electionTimeout',
lastElectionDate: ISODate("2023-04-13T21:16:59.775Z"),
electionTerm: Long("1"),
lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1681420619, i: 1 }), t: Long("-1") },
lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1681420619, i: 1 }), t: Long("-1") },
numVotesNeeded: 1,
priorityAtElection: 1,
electionTimeoutMillis: Long("10000"),
newTermStartDate: ISODate("2023-04-13T21:16:59.817Z"),
wMajorityWriteAvailabilityDate: ISODate("2023-04-13T21:16:59.857Z")
},
members: [
{
_id: 0,
name: 'ec2-54-226-251-91.compute-1.amazonaws.com:26201',
health: 1,
state: 1,
stateStr: 'PRIMARY',
uptime: 9536,
optime: { ts: Timestamp({ t: 1681430100, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:55:00.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:55:00.142Z"),
lastDurableWallTime: ISODate("2023-04-13T23:55:00.142Z"),
syncSourceHost: '',
syncSourceId: -1,
infoMessage: '',
electionTime: Timestamp({ t: 1681420619, i: 2 }),
electionDate: ISODate("2023-04-13T21:16:59.000Z"),
configVersion: 5,
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},
{


shard3
mongosh mongodb://127.0.0.
ubuntu@ip-172-31-22-123: ~
ubuntu@ip-172-31-22-123: ~
{
_id: 1,
name: 'ec2-54-226-251-91.compute-1.amazonaws.com:26202',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 9461,
optime: { ts: Timestamp({ t: 1681430100, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681430100, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:55:00.000Z"),
optimeDurableDate: ISODate("2023-04-13T23:55:00.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:55:00.142Z"),
lastDurableWallTime: ISODate("2023-04-13T23:55:00.142Z"),
lastHeartbeat: ISODate("2023-04-13T23:55:08.303Z"),
lastHeartbeatRecv: ISODate("2023-04-13T23:55:08.485Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-54-226-251-91.compute-1.amazonaws.com:26201',
syncSourceId: 0,
infoMessage: '',
configVersion: 5,
configTerm: 1
},
{
_id: 2,
name: 'ec2-54-226-251-91.compute-1.amazonaws.com:26203',
health: 1,
state: 2,
stateStr: 'SECONDARY',
uptime: 9456,
optime: { ts: Timestamp({ t: 1681430100, i: 1 }), t: Long("1") },
optimeDurable: { ts: Timestamp({ t: 1681430100, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-04-13T23:55:00.000Z"),
optimeDurableDate: ISODate("2023-04-13T23:55:00.000Z"),
lastAppliedWallTime: ISODate("2023-04-13T23:55:00.142Z"),
lastDurableWallTime: ISODate("2023-04-13T23:55:00.142Z"),
lastHeartbeat: ISODate("2023-04-13T23:55:08.478Z"),
lastHeartbeatRecv: ISODate("2023-04-13T23:55:08.621Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: 'ec2-54-226-251-91.compute-1.amazonaws.com:26202',
syncSourceId: 1,
infoMessage: '',
configVersion: 5,
configTerm: 1
}
],
ok: 1,
lastCommittedOpTime: Timestamp({ t: 1681430100, i: 1 }),
'$clusterTime': {
clusterTime: Timestamp({ t: 1681430106, i: 2 }),
signature: {
hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex"), 0),
keyId: Long("0")
}
},
operationTime: Timestamp({ t: 1681430100, i: 1 })
}
shard3 [direct: primary] test>

14.

node0: mongos: ec2-54-226-173-1.compute-1.amazonaws.com
node0:port#27017 mongos server

node1: configs: ec2-3-94-196-162.compute-1.amazonaws.com
node1:port#27031 config server PRIMARY
node1:port#27032 config server SECONDARY
node1:port#27033 config server SECONDARY

node2: shard1: ec2-34-230-29-212.compute-1.amazonaws.com
node2:port#26001 shard1 server PRIMARY
node2:port#26002 shard1 server SECONDARY
node2:port#26003 shard1 server SECONDARY

node3: shard2: ec2-34-226-213-111.compute-1.amazonaws.com
node3:port#26101 shard2 server PRIMARY
node3:port#26102 shard2 server SECONDARY
node3:port#26103 shard2 server SECONDARY

node4: shard3: ec2-54-226-251-91.compute-1.amazonaws.com
node4:port#26201 shard3 server PRIMARY
node4:port#26202 shard3 server SECONDARY
node4:port#26203 shard3 server SECONDARY