

## Lab: MongoDB-Java

### Part A. Setup a MongoDB that allows remote access

1. Create VM instances <https://cloud.google.com/compute/docs/quickstart-linux>
  - a. Name: mongo-remote
  - b. Region: us-west1, zone: us-west1-a
  - c. 1vCPU with 3.75 GB memory, n1-standard-1. Customize if you know your requirements well
  - d. Ubuntu 18.04 LTS, use default settings for other options
  - e. Default access
  - f. Allow http traffic
2. Wait for boot up
3. Install MongoDB (<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/>)
4. Create a database testDB and a database user test

```
mongo
use testDB
db.createUser( {
  user: "test",
  pwd: "csis3300",
  roles: [ {role: "readWrite", db: "testDB"},
           {role: "dbAdmin", db: "testDB"} ]
})
```

You can use show users to check.

5. Open MongoDB access up to all IPs
  - a. Edit MongoDB config file  
`sudo nano /etc/mongod.conf`
  - b. Look for the net line and comment out the bindIp line under it, which is currently limiting MongoDB connections to localhost:





```
# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1  <- change to 0.0.0.0
```

## 6. Setup firewall on virtual machine

### a. View network details of mongo-remote

Filter VM instances

Columns

<input type="checkbox"/> Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input type="checkbox"/>  csis3300-text	us-west1-b		10.138.0.2 (nic0)	35.230.89.115 	SSH ▾ ⋮
<input type="checkbox"/>  mongo-remote	us-west1-a		10.138.0.4 (nic0)	35.247.121.99 	SSH ▾ ⋮

Start

Stop

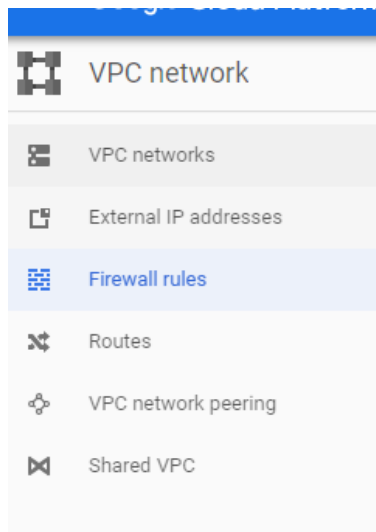
Reset

Delete

View network details

View logs

### b. Click Firewall rules



c. Create Firewall Rule

[←](#) Firewall rule details [EDIT](#) [DELETE](#)

---

**mongodb**  
**Description**

**Logs**  
Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. [Learn more](#)  
☐ On  
☒ Off

**Network**  
default

**Priority** [?](#)  
Priority can be 0–65535 [Check priority of other firewall rules](#)  

1000

**Direction**  
Ingress

**Action on match**  
Allow

**Targets**  

All instances in the network ▼

**Source filter** [?](#)  

IP ranges ▼

**Source IP ranges** [?](#)  

0.0.0.0/0 ✕

**Second source filter** [?](#)  

None ▼

**Protocols and ports**  
☐ Allow all  
☒ Specified protocols and ports  

tcp:27017

[⌵ Disable rule](#)

Save

Cancel

7. Restart the virtual machine and bring up the mongod service
8. On another virtual machine, type the following command to verify that remote access has been enabled.

```
mongo -u test -p csis3300 --host 35.247.121.99 --authenticationDatabase testDB
```

35.247.121.99 is the external IP address of mongo-remote. This IP address may change when you restart the virtual machine.



## Part B. MongoDB-Java examples

1. Download the MongoDB Driver  
<http://central.maven.org/maven2/org/mongodb/mongo-java-driver/3.9.1/>
2. Open Eclipse and create a new project mongo-java.
3. Add the MongoDB Driver to the Build Path.
4. Create collections
5. Insert Documents
6. Query

## References

<http://mongodb.github.io/mongo-java-driver/3.10/driver/getting-started/quick-start/>  
[https://www.tutorialspoint.com/mongodb/mongodb\\_java.htm](https://www.tutorialspoint.com/mongodb/mongodb_java.htm)  
<https://www.codejava.net/java-se/jdbc/java-connecting-to-mongodb-database-examples>  
<http://mongodb.github.io/mongo-java-driver/3.10/driver/tutorials/perform-read-operations/>