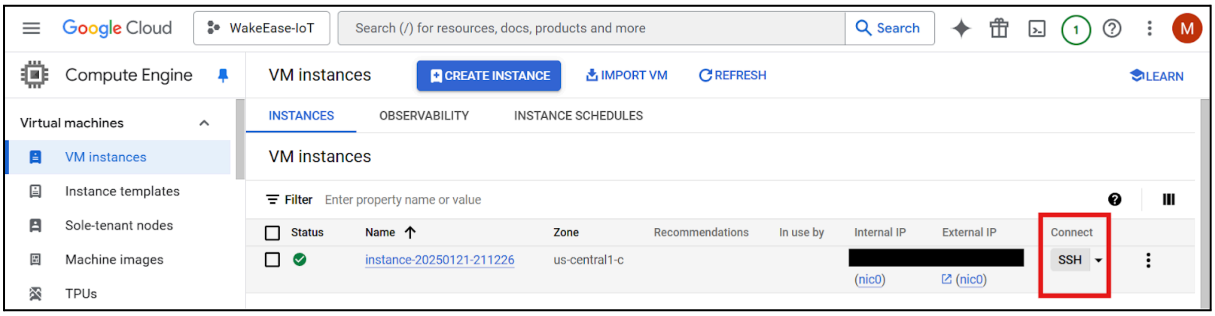
Setting Up MongoDB and MongoDB Charts

Step 1: Launch SSH

1. Click the ***SSH*** button next to your VM instance to launch the SSH-in-browser.



□	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	instance-20250121-211226	us-central1-c			[redacted] (nic0)	(nic0)	SSH

Step 2: Update the System

1. Run the following commands to update and upgrade your system:

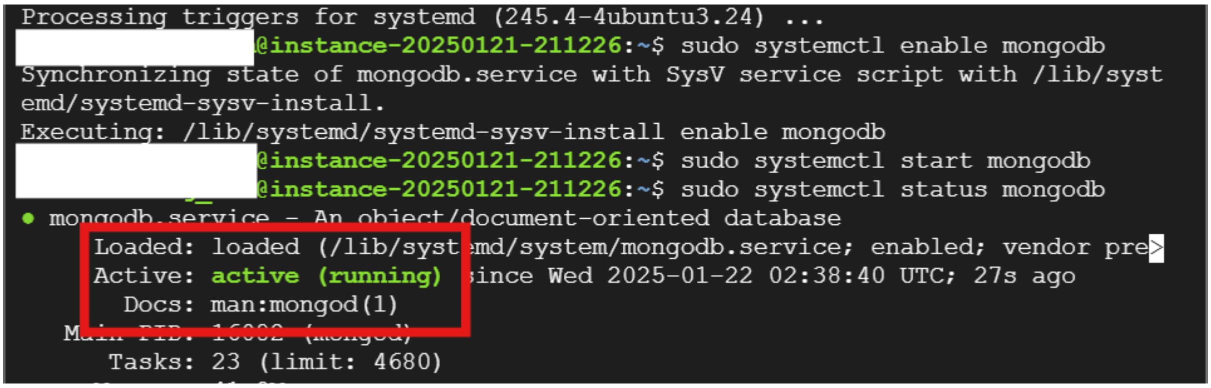
- `sudo apt-get update`
- `sudo apt-get upgrade`

Step 3: Install MongoDB

1. Run the following commands to install MongoDB:

- `sudo apt-get install -y mongodb`
- `sudo systemctl enable mongodb`
- `sudo systemctl start mongodb`
- `sudo systemctl status mongodb`

Note: Ensure that the status shows Active: active running



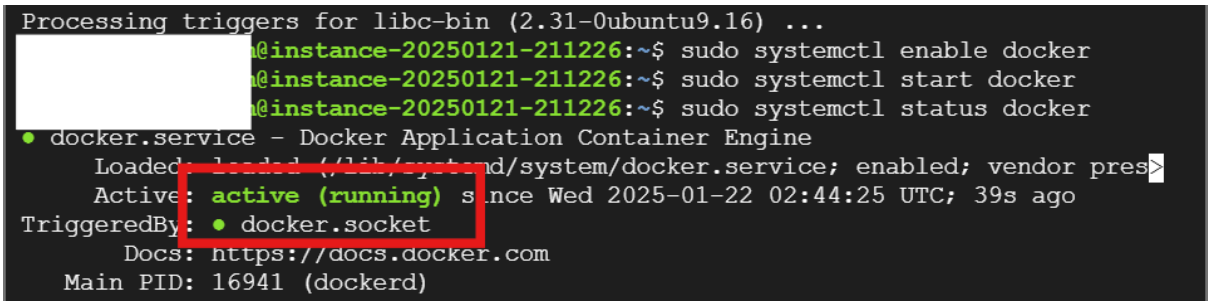
```
Processing triggers for systemd (245.4-4ubuntu3.24) ...
[redacted]@instance-20250121-211226:~$ sudo systemctl enable mongodb
Synchronizing state of mongodb.service with SysV service script with /lib/syst
emd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mongodb
[redacted]@instance-20250121-211226:~$ sudo systemctl start mongodb
@instance-20250121-211226:~$ sudo systemctl status mongodb
● mongodb.service - An object/document-oriented database
     Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor pre>
     Active: active (running) since Wed 2025-01-22 02:38:40 UTC; 27s ago
       Docs: man:mongod(1)
   Main PID: 16082 (mongod)
      Tasks: 23 (limit: 4680)
```

Step 4: Install Docker

1. Open a new SSH terminal and run the following commands to install Docker:

- `sudo apt install -y docker.io`
- `sudo systemctl enable docker`
- `sudo systemctl start docker`
- `sudo systemctl status docker`

Note: Ensure that the status shows Active: active running



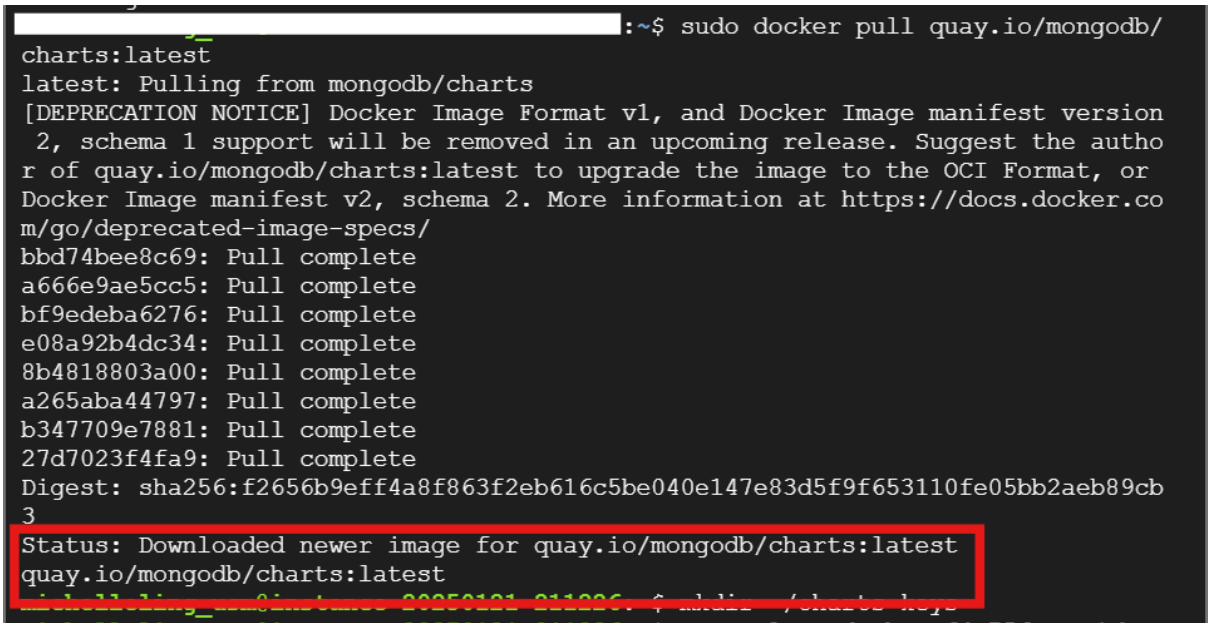
```
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
[redacted]@instance-20250121-211226:~$ sudo systemctl enable docker
@instance-20250121-211226:~$ sudo systemctl start docker
@instance-20250121-211226:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor pres>
     Active: active (running) since Wed 2025-01-22 02:44:25 UTC; 39s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 16941 (dockerd)
```

Step 5: Pull MongoDB Charts Image

1. Open a new SSH terminal, then run the following command to pull the MongoDB Charts image:

- `sudo docker pull quay.io/mongodb/charts:latest`

Note: Ensure that the status shows Downloaded newer image ...



```
[redacted]:~$ sudo docker pull quay.io/mongodb/
charts:latest
latest: Pulling from mongodb/charts
[DEPRECATION NOTICE] Docker Image Format v1, and Docker Image manifest version
 2, schema 1 support will be removed in an upcoming release. Suggest the autho
r of quay.io/mongodb/charts:latest to upgrade the image to the OCI Format, or
Docker Image manifest v2, schema 2. More information at https://docs.docker.co
m/go/deprecated-image-specs/
bbd74bee8c69: Pull complete
a666e9ae5cc5: Pull complete
bf9edeba6276: Pull complete
e08a92b4dc34: Pull complete
8b4818803a00: Pull complete
a265aba44797: Pull complete
b347709e7881: Pull complete
27d7023f4fa9: Pull complete
Digest: sha256:f2656b9eff4a8f863f2eb616c5be040e147e83d5f9f653110fe05bb2aeb89cb
3
Status: Downloaded newer image for quay.io/mongodb/charts:latest
quay.io/mongodb/charts:latest
```

Step 6: Generate Encryption Key for MongoDB Charts

1. Create a directory for the encryption key:
   - `mkdir ~/charts-keys`
2. Generate the encryption key:
   - `openssl rand -base64 756 > ~/charts-keys/mongodb-charts.key`
3. Set the correct permissions for the key file:
   - `sudo chmod 600 ~/charts-keys/mongodb-charts.key`

Step 7: Edit MongoDB Configuration File

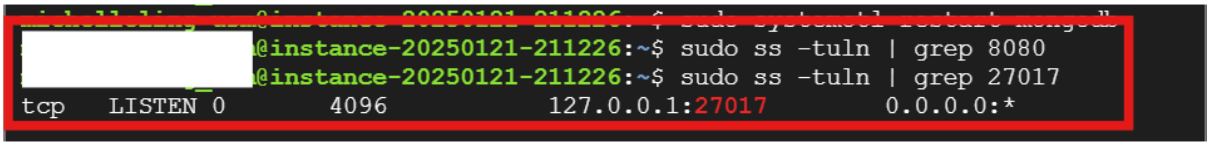
1. Open the MongoDB configuration file:
   - `sudo nano /etc/mongod.conf`
2. Replace the file content with:

```
storage:
  dbPath: /var/lib/mongodb
  journal:
    enabled: true

systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

net:
  port: 27017
  bindIp: 0.0.0.0  # For remote access

security:
  authorization: "enabled"

processManagement:
  timeZoneInfo: /usr/share/zoneinfo
```

3. Save and close the file:
   - **CTRL + O**, then **ENTER** to save.
   - **CTRL + X** to exit.
4. Check if the file is updated:
   - `sudo nano /etc/mongod.conf`
5. Restart MongoDB to apply changes:
   - `sudo systemctl restart mongodb`

Step 8: Check Firewall Settings

1. Ensure your Google Cloud Firewall allows **Port 27017** for MongoDB and **Port 8080** is available for MongoDB Charts dashboard.
2. Check if **Port 8080** is free:
   - `sudo ss -tuln | grep 8080`
3. Verify MongoDB is listening on **Port 27017**:
   - `sudo ss -tuln | grep 27017`
4. The following screenshot shows the expected output for both ports:

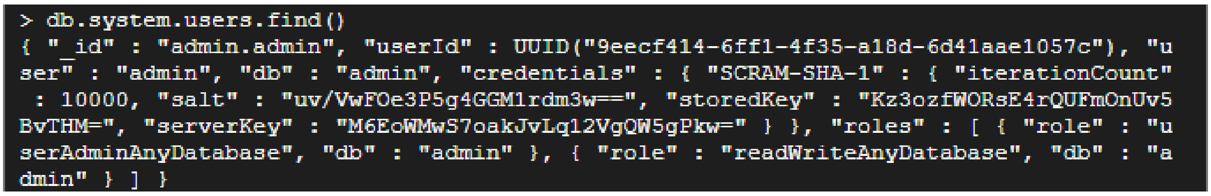


```
[redacted]@instance-20250121-211226:~$ sudo ss -tuln | grep 8080
[redacted]@instance-20250121-211226:~$ sudo ss -tuln | grep 27017
tcp   LISTEN 0      4096          127.0.0.1:27017        0.0.0.0:*
```

Step 9: Create Admin User for Docker Image

1. Open the MongoDB shell:
   - `mongo --host 127.0.0.1 --port 27017`
2. In MongoDB shell, switch to **admin** database
   - `use admin`
3. Create admin user credentials for Docker:

   -
     ```
     db.createUser({
       user: "admin",
       pwd: "password",
       roles: [
         { role: "userAdminAnyDatabase", db: "admin" },
         "readWriteAnyDatabase"
       ]
     })
     ```
4. You can verify the admin user has been created using this command:
   - `db.system.users.find()`



```
> db.system.users.find()
{ "_id" : "admin.admin", "userId" : UUID("9eecf414-6ff1-4f35-a18d-6d41aae1057c"), "u
ser" : "admin", "db" : "admin", "credentials" : { "SCRAM-SHA-1" : { "iterationCount"
 : 10000, "salt" : "uv/VwFOe3P5g4GGM1rdm3w==", "storedKey" : "Kz3ozfWORsE4rQUFmOnUv5
BvTHM=", "serverKey" : "M6EoWMwS7oakJvLq12VgQW5gPkw=" } }, "roles" : [ { "role" : "u
serAdminAnyDatabase", "db" : "admin" }, { "role" : "readWriteAnyDatabase", "db" : "a
dmin" } ] }
```

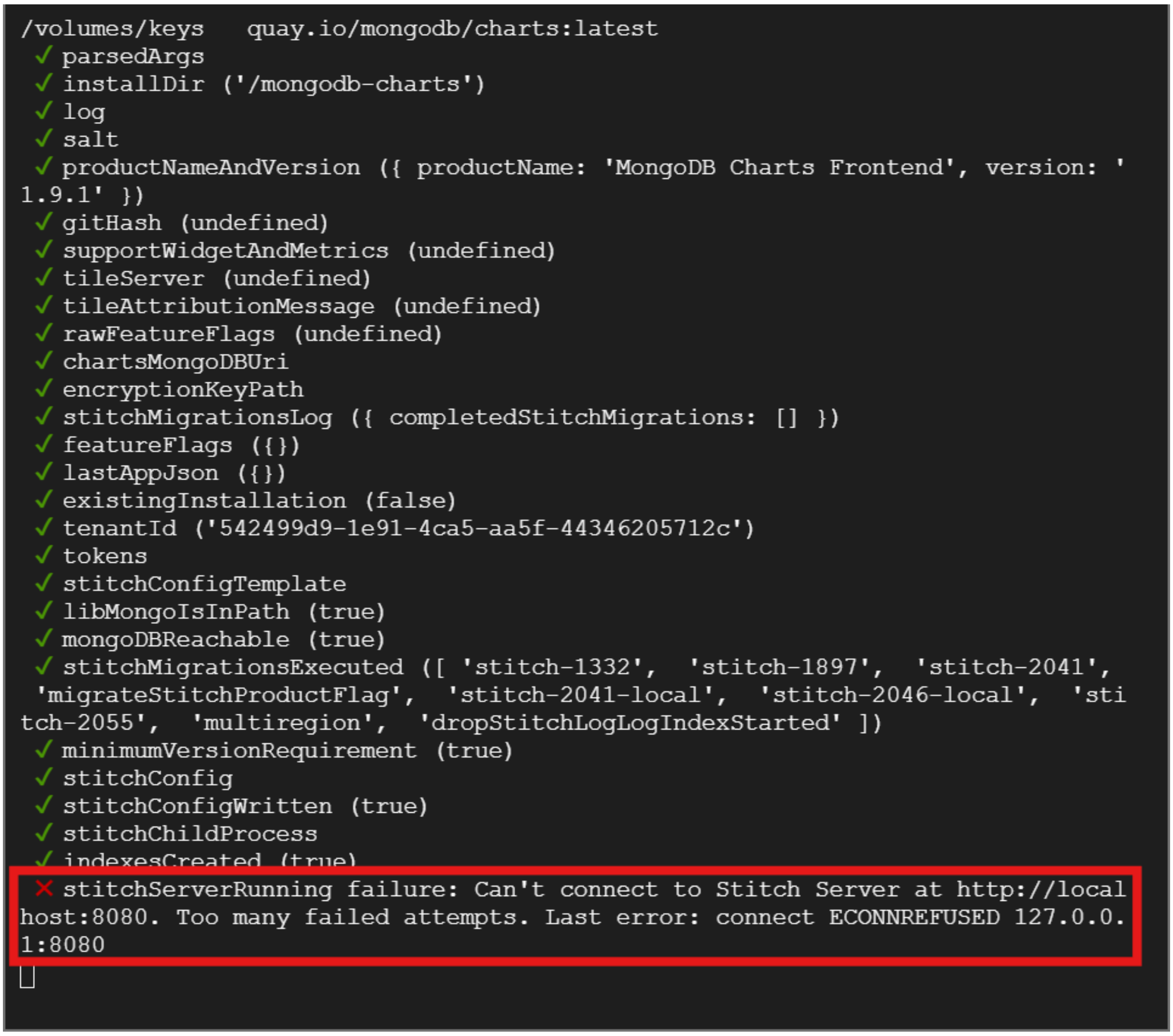
5. Exit MongoDB:
   - `exit` or `[CTRL + C]`

Step 10: Run MongoDB Charts with Docker

1. Start MongoDB Charts:

-
  ```
  sudo docker run --network host -it \
    -e CHARTS_MONGODB_URI="mongodb://admin:password@127.0.0.1:27017" \
    -e CHARTS_USER="admin" \
    -e CHARTS_PASS="password" \
    -v ~/charts-keys:/mongodb-charts/volumes/keys \
    quay.io/mongodb/charts:latest
  ```

2. If you encounter this error, terminate the process [**`CTRL + C`**] and proceed to next step.



```
/volumes/keys   quay.io/mongodb/charts:latest
 ✓ parsedArgs
 ✓ installDir ('/mongodb-charts')
 ✓ log
 ✓ salt
 ✓ productNameAndVersion ({ productName: 'MongoDB Charts Frontend', version: '
1.9.1' })
 ✓ gitHash (undefined)
 ✓ supportWidgetAndMetrics (undefined)
 ✓ tileServer (undefined)
 ✓ tileAttributionMessage (undefined)
 ✓ rawFeatureFlags (undefined)
 ✓ chartsMongoDBUri
 ✓ encryptionKeyPath
 ✓ stitchMigrationsLog ({ completedStitchMigrations: [] })
 ✓ featureFlags ({})
 ✓ lastAppJson ({})
 ✓ existingInstallation (false)
 ✓ tenantId ('542499d9-1e91-4ca5-aa5f-44346205712c')
 ✓ tokens
 ✓ stitchConfigTemplate
 ✓ libMongoIsInPath (true)
 ✓ mongoDBReachable (true)
 ✓ stitchMigrationsExecuted ([ 'stitch-1332',  'stitch-1897',  'stitch-2041',
 'migrateStitchProductFlag',  'stitch-2041-local',  'stitch-2046-local',  'sti
tch-2055',  'multiregion',  'dropStitchLogLogIndexStarted' ])
 ✓ minimumVersionRequirement (true)
 ✓ stitchConfig
 ✓ stitchConfigWritten (true)
 ✓ stitchChildProcess
 ✓ indexesCreated (true)
 ✗ stitchServerRunning failure: Can't connect to Stitch Server at http://local
host:8080. Too many failed attempts. Last error: connect ECONNREFUSED 127.0.0.
1:8080
```

Step 11: Troubleshoot Previous Error

1. Remove and recreate the charts-keys directory:

- `rm -rf ~/charts-keys`
- `mkdir ~/charts-keys`

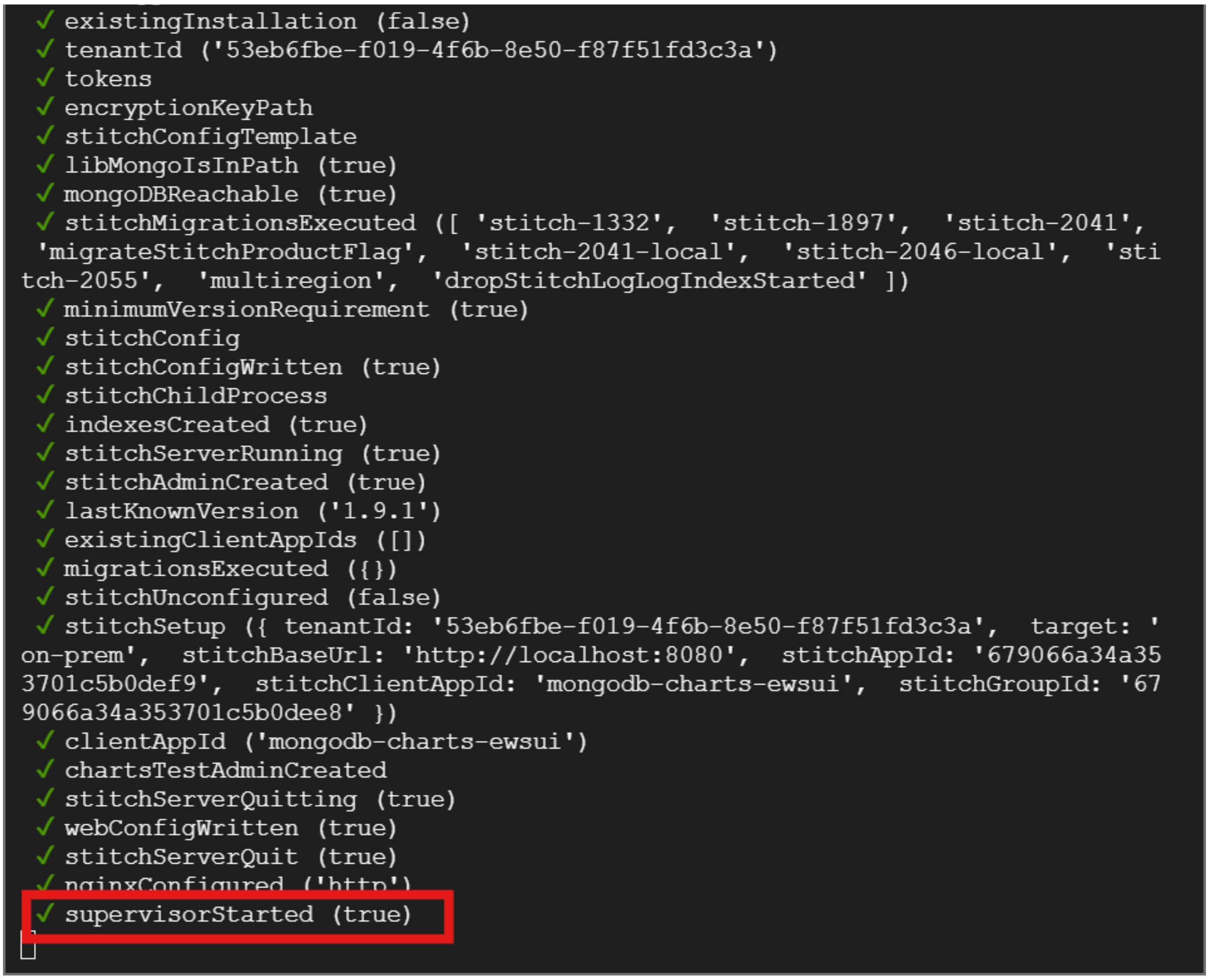
2. Set the correct permissions:

- `sudo chown -R $USER:$USER ~/charts-keys`
- `sudo chmod -R 755 ~/charts-keys`

3. Restart MongoDB Charts:

-
```
sudo docker run --network host -it \
  -e CHARTS_MONGODB_URI="mongodb://admin:password@127.0.0.1:27017" \
  -e CHARTS_USER="admin" \
  -e CHARTS_PASS="password" \
  -v ~/charts-keys:/mongodb-charts/volumes/keys \
  quay.io/mongodb/charts:latest
```

4. Once ✔ supervisorStarted (true) message appears, it means MongoDB Charts is ready to access via the browser. Proceed to the next step.



```
 ✓ existingInstallation (false)
 ✓ tenantId ('53eb6fbe-f019-4f6b-8e50-f87f51fd3c3a')
 ✓ tokens
 ✓ encryptionKeyPath
 ✓ stitchConfigTemplate
 ✓ libMongoIsInPath (true)
 ✓ mongoDBReachable (true)
 ✓ stitchMigrationsExecuted ([ 'stitch-1332',  'stitch-1897',  'stitch-2041',
 'migrateStitchProductFlag',  'stitch-2041-local',  'stitch-2046-local',  'sti
tch-2055',  'multiregion',  'dropStitchLogLogIndexStarted' ])
 ✓ minimumVersionRequirement (true)
 ✓ stitchConfig
 ✓ stitchConfigWritten (true)
 ✓ stitchChildProcess
 ✓ indexesCreated (true)
 ✓ stitchServerRunning (true)
 ✓ stitchAdminCreated (true)
 ✓ lastKnownVersion ('1.9.1')
 ✓ existingClientAppIds ([])
 ✓ migrationsExecuted ({})
 ✓ stitchUnconfigured (false)
 ✓ stitchSetup ({ tenantId: '53eb6fbe-f019-4f6b-8e50-f87f51fd3c3a',  target: '
on-prem',  stitchBaseUrl: 'http://localhost:8080',  stitchAppId: '679066a34a35
3701c5b0def9',  stitchClientAppId: 'mongodb-charts-ewsui',  stitchGroupId: '67
9066a34a353701c5b0dee8' })
 ✓ clientAppId ('mongodb-charts-ewsui')
 ✓ chartsTestAdminCreated
 ✓ stitchServerQuitting (true)
 ✓ webConfigWritten (true)
 ✓ stitchServerQuit (true)
 ✓ nginxConfigured ('http')
 ✓ supervisorStarted (true)
```

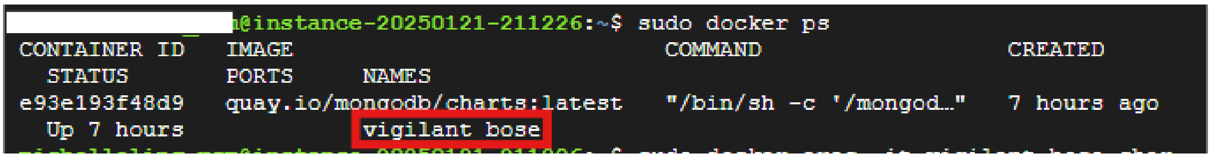
Step 12: Add Admin User for MongoDB Charts

Note: MongoDB Charts requires the username to be in email format (e.g., user@example.com).

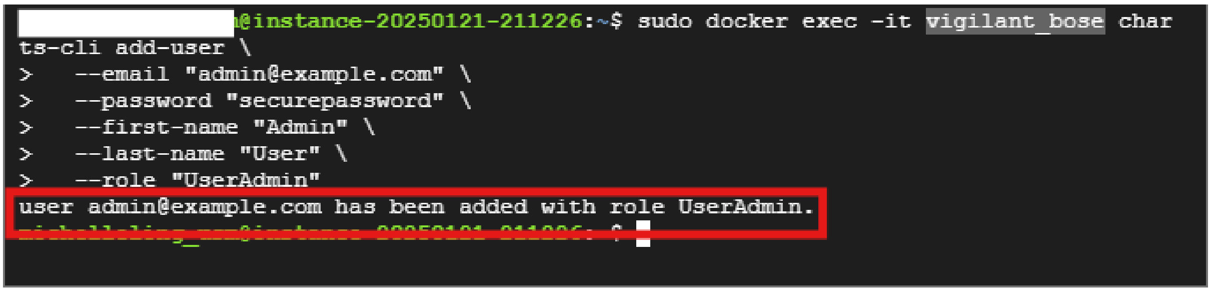
1. Open a new SSH terminal and run the following command to identify the `CONTAINER_NAME` for the active Docker container:
   - `sudo docker ps`
2. To add the admin user, replace `CONTAINER_NAME` with the name of the container obtained from the sudo docker ps output and execute the following command:

   -
     ```
     sudo docker exec -it <CONTAINER_NAME> charts-cli add-user \
       --email "admin@example.com" \
       --password "securepassword" \
       --first-name "Admin" \
       --last-name "User" \
       --role "UserAdmin"
     ```
3. Example:
   - From the sudo docker ps output, if the `CONTAINER_NAME` is **vigilant_bose**, the command would look like this:

   -
     ```
     sudo docker exec -it vigilant_bose charts-cli add-user \
       --email "admin@example.com" \
       --password "securepassword" \
       --first-name "Admin" \
       --last-name "User" \
       --role "UserAdmin"
     ```



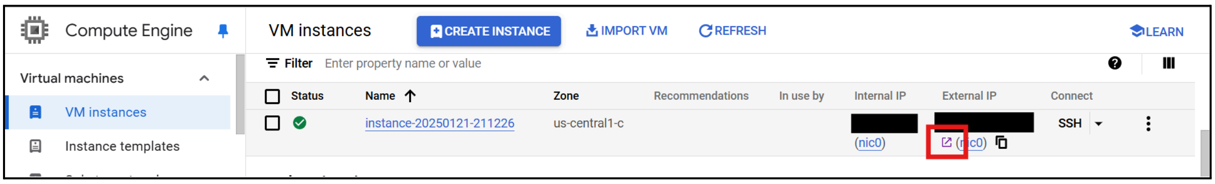
```
[redacted]@instance-20250121-211226:~$ sudo docker ps
CONTAINER ID   IMAGE                          COMMAND                   CREATED
  STATUS       PORTS      NAMES
e93e193f48d9   quay.io/mongodb/charts:latest  "/bin/sh -c '/mongod…"    7 hours ago
  Up 7 hours              vigilant_bose
```



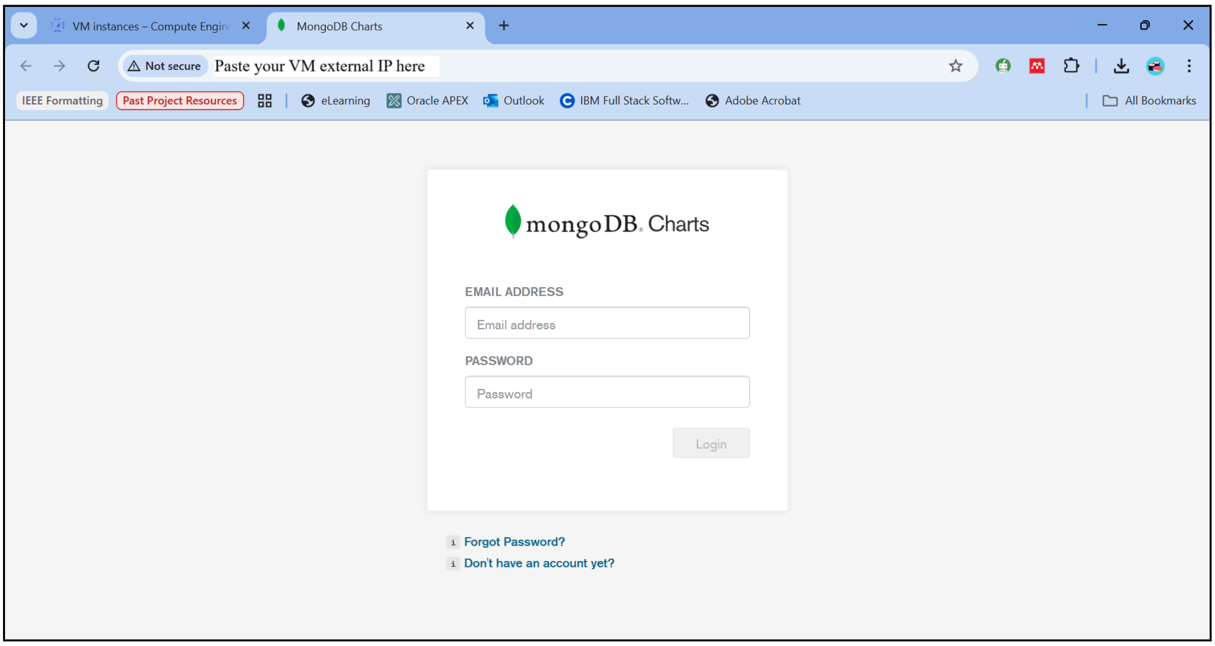
```
[redacted]@instance-20250121-211226:~$ sudo docker exec -it vigilant_bose char
ts-cli add-user \
>    --email "admin@example.com" \
>    --password "securepassword" \
>    --first-name "Admin" \
>    --last-name "User" \
>    --role "UserAdmin"
user admin@example.com has been added with role UserAdmin.
```

Step 13: Access MongoDB Charts Dashboard

1. Go to Google Cloud Platform.
2. On the **VM Instance**, click the external IP to open the MongoDB Charts dashboard.
3. If it doesn't launch, try copying and pasting the IP into your browser's address bar.
4. Once the page loads, MongoDB Charts is ready!
5. Use the following credentials to log in (created at step 12):
   - **Username**: admin@example.com
   - **Password**: securepassword



□	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	instance-20250121-211226	us-central1-c			[redacted] (nic0)	[redacted] (nic0)	SSH



Paste your VM external IP here

EMAIL ADDRESS

Email address

PASSWORD

Password