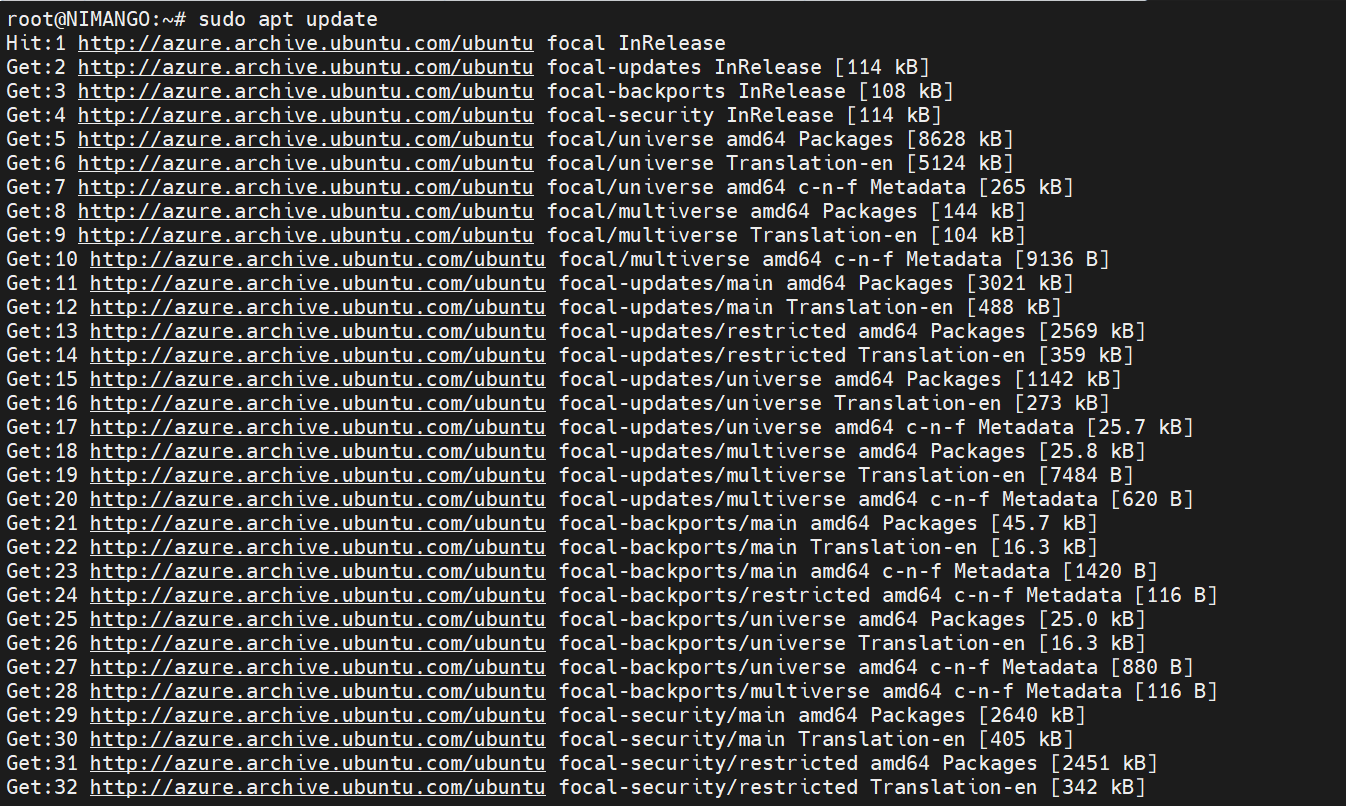
**MongoDB Replica Set Setup and Integrating with the Ops Manager for Backup**

**Prerequisites:**

* 3-node cluster environment with a primary node and two secondary nodes for Replica Set.
* The minimum configuration for each node is 1 vCPU and 3.5 GB RAM.
* The minimum configuration for the Ops Manager machine is 2 vCPU and 8 GB RAM.
* It is recommended to have all these machines in a single network.
* Make sure the security rules are updated so that all the nodes can communicate with each other seamlessly. Make sure the ports 27017 and 8080 are open between the nodes.

**Step 1: Install MongoDB**

Install MongoDB Enterprise on all nodes. Please note that MongoDB v5 is being used for this deployment as the latest version v8 had some issues.

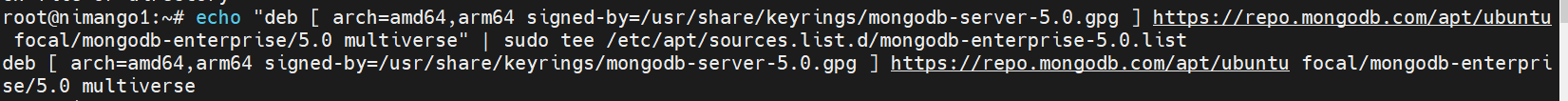
sudo apt-get update

sudo apt-get install software-properties-common gnupg apt-transport-https ca-certificates -y

A screenshot of a computer program

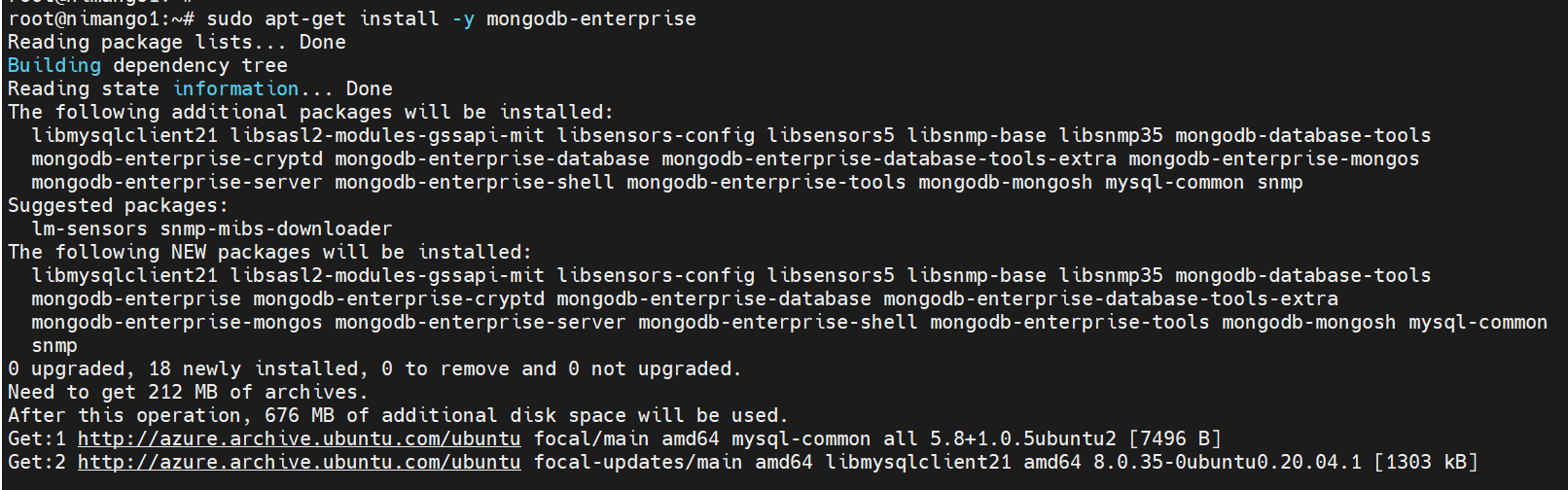
Description automatically generated

curl -fsSL https://pgp.mongodb.com/server-5.0.asc | sudo gpg -o /usr/share/keyrings/mongodb-server-5.0.gpg --dearmor

echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-5.0.gpg ] https://repo.mongodb.com/apt/ubuntu focal/mongodb-enterprise/5.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-enterprise-5.0.list

A screen shot of a computer

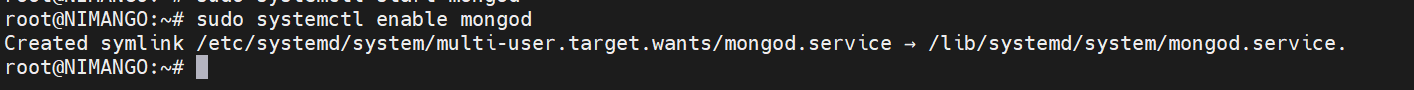
Description automatically generatedsudo apt-get update

sudo apt-get install -y mongodb-enterprise

sudo systemctl start mongod

A black screen with white text

Description automatically generated

sudo systemctl enable mongod

**Step 2: Create an Administrative User on the Master Node**

Create an administrative user on the master node:

A screen shot of a computer program

Description automatically generatedmongosh

A black rectangle with white text

Description automatically generateduse admin

db.createUser({

user: "admin-user",

pwd: passwordPrompt(),

roles: [{ role: "root", db: "admin" }, "readWriteAnyDatabase"]})

A computer screen shot of text

Description automatically generated

exit

**Step 3: Configure the Master Node**

Generate a key file for authentication:

sudo su –

A black background with white text

Description automatically generated

A screen shot of a computer

Description automatically generatedopenssl rand -base64 756 > /opt/keyfile

A black screen with white text

Description automatically generatedchmod 400 /opt/keyfile

A black background with white text

Description automatically generatedchown mongodb:mongodb /opt/keyfile

**Copy the key file to all the nodes in your cluster.**

scp /opt/keyfile root@<NodeIP>:/opt/

Note: Enable passwordless ssh

Access the default MongoDB configuration file and update the configuration file as shown, including the replica set name.

sudo vi /etc/mongod.conf

-----------------------------------------

net:

port: 27017

bindIp: 0.0.0.0

security:

keyFile: /opt/keyfile

replication:

replSetName: "replica01"A screen shot of a computer

Description automatically generated

**Step 4: Configure the Client Nodes**

Edit the configuration file on each client node and update the configuration for each node, including the bindIp and keyFile settings.

sudo vi /etc/mongod.conf

-----------------------------------------

net:

port: 27017

bindIp: 0.0.0.0

security:

keyFile: /opt/keyfile

replication:

replSetName: "replica01"

A screen shot of a computer

Description automatically generated

Apply the following ownership and permissions to the key file that was copied.

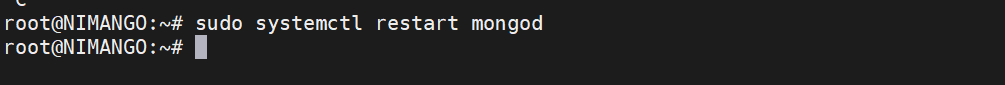
A black background with white text

Description automatically generatedchmod 400 /opt/keyfile

A black screen with white text

Description automatically generatedchown mongodb:mongodb /opt/keyfile

For the changes to take effect, restart the MongoDB service.

sudo systemctl restart mongod

sudo systemctl status mongod

A screen shot of a computer

Description automatically generated

Once the configuration changes are made, verify the connectivity using the below command on all the servers.

nc -zv <NodePublicIP> 27017

A black screen with white text

Description automatically generated

**Step 5: Initiate the Replica Set**

On the master node, initiate the replica set and add nodes:

A computer screen shot of text

Description automatically generatedmongosh -u admin-user -p --authenticationDatabase admin

A black screen with green text

Description automatically generatedrs.initiate()

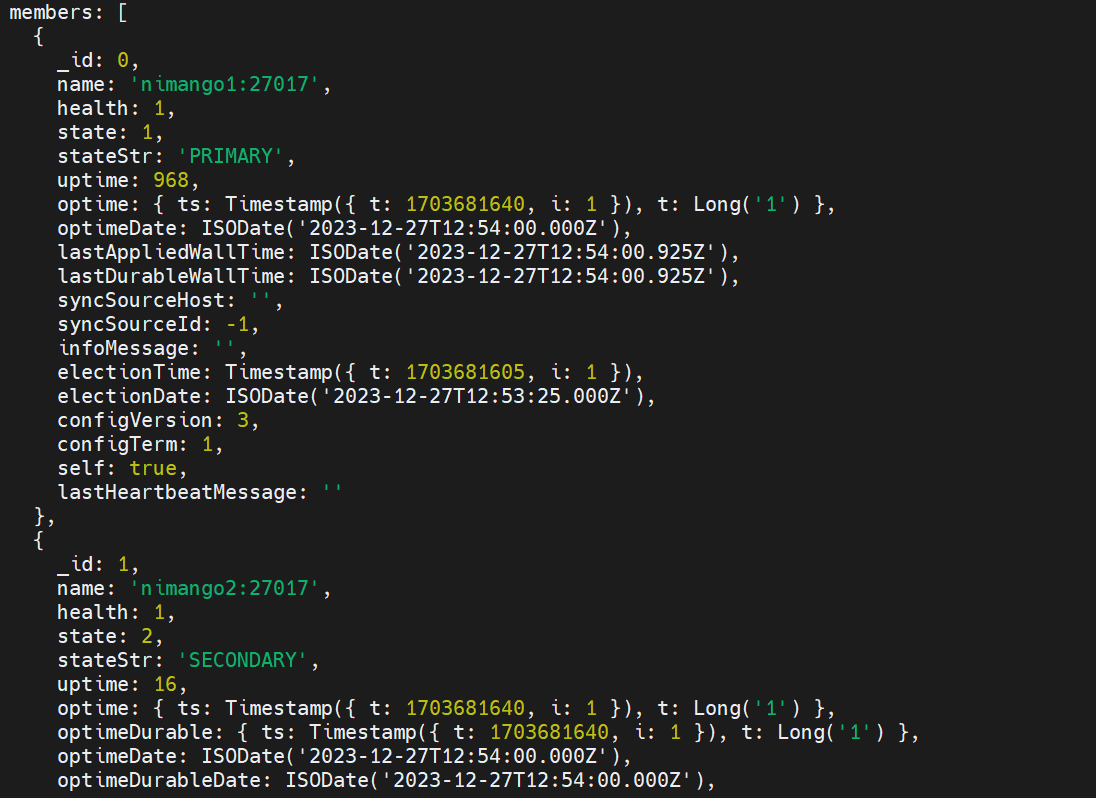
rs.add("<ClientNode1PublicIP>")

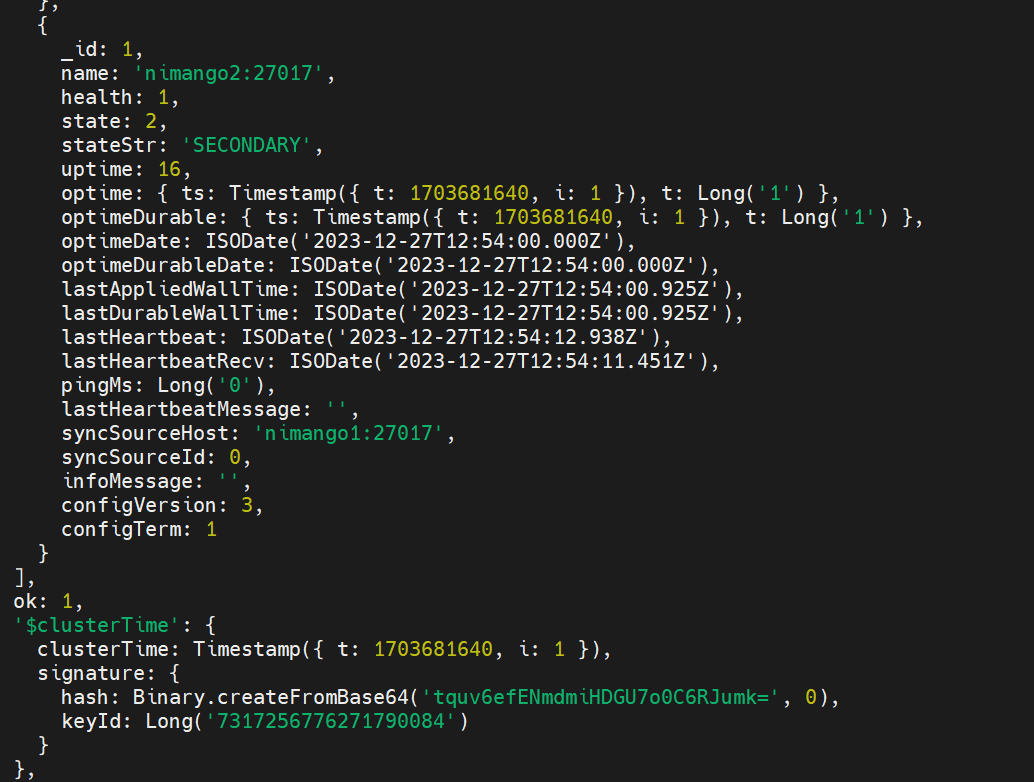
rs.add("<ClientNode2PublicIP>")

A computer screen with text and numbers

Description automatically generated

rs.status()





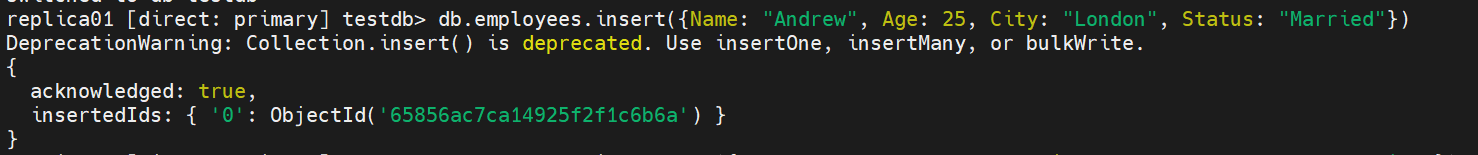
**Step 7: Verify Replication**

A computer screen shot of text

Description automatically generatedmongosh -u admin-user -p --authenticationDatabase admin

A black background with white text

Description automatically generateduse testdb

db.employees.insert({Name: "Andrew", Age: 25, City: "London", Status: "Married"})

A computer screen with green text

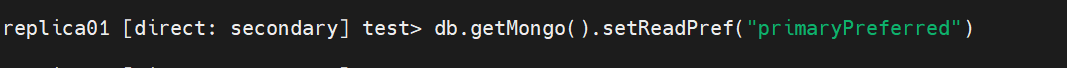
Description automatically generateddb.employees.find()

Now, head over to any client node and log into MongoDB as the Administrative user.

A computer screen shot of text

Description automatically generatedmongosh -u admin-user -p --authenticationDatabase admin

Once logged in, enable the secondary member to read operations on a per-connection basis using the following command

db.getMongo().setReadPref("primaryPreferred")

Finally, switch to the test database and list the documents in the collection.

A black background with white text

Description automatically generateduse testdb

db.employees.find()

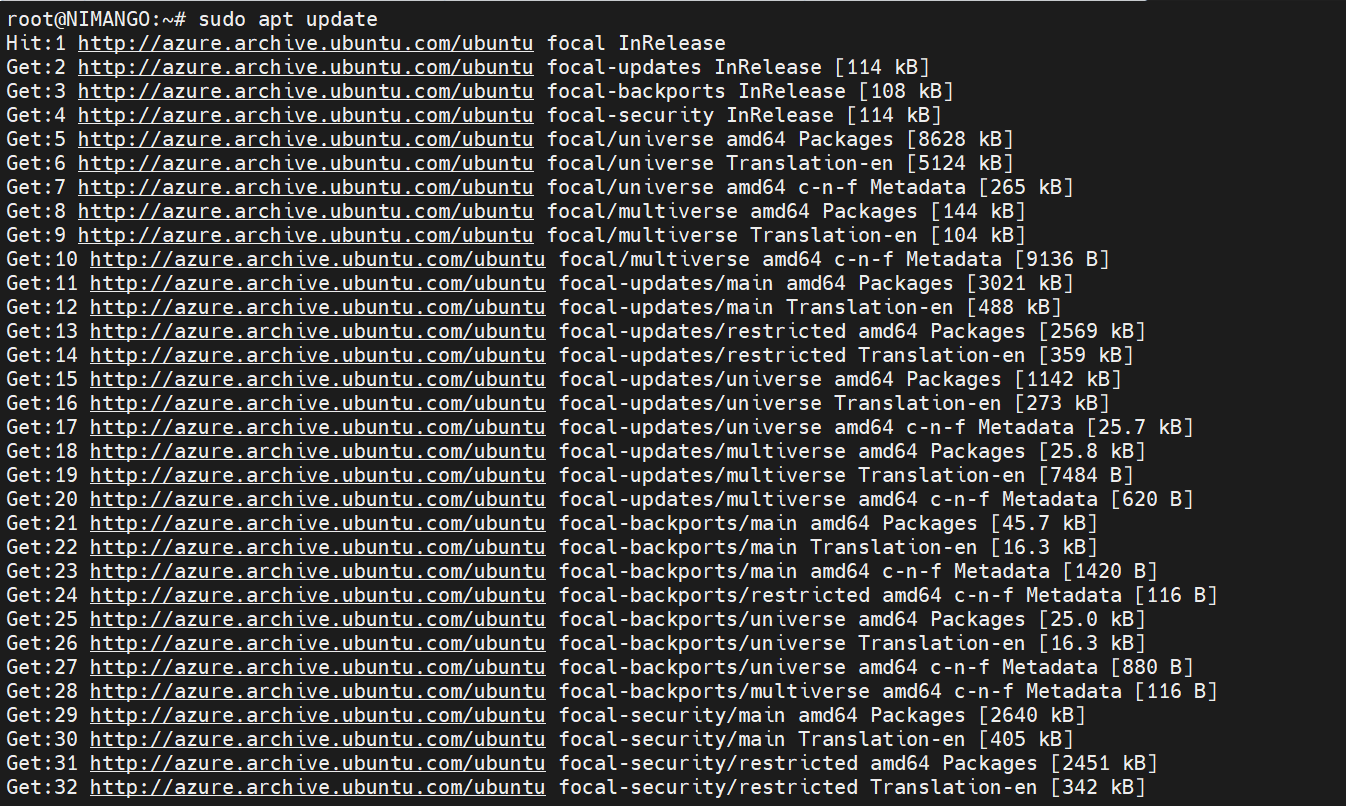
A computer screen with white and green text

Description automatically generated

**Ops Manager Installation**

**Install MongoDB**

sudo apt-get update

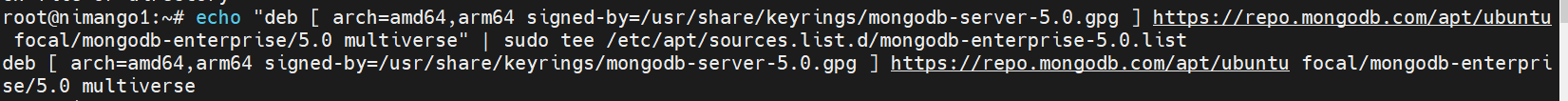


sudo apt-get install software-properties-common gnupg apt-transport-https ca-certificates -y

A screenshot of a computer program

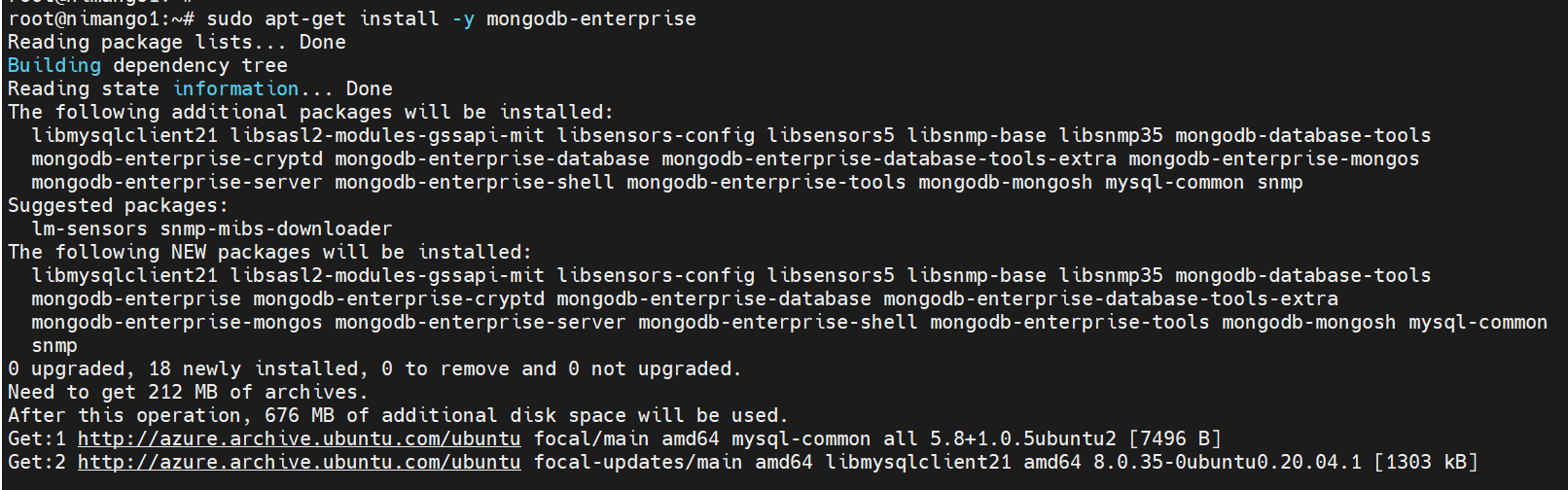
Description automatically generated

curl -fsSL https://pgp.mongodb.com/server-5.0.asc | sudo gpg -o /usr/share/keyrings/mongodb-server-5.0.gpg --dearmor

echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-5.0.gpg ] https://repo.mongodb.com/apt/ubuntu focal/mongodb-enterprise/5.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-enterprise-5.0.list

A screen shot of a computer

Description automatically generatedsudo apt-get update

sudo apt-get install -y mongodb-enterprise

sudo vi /etc/mongod.conf

-----------------------------------------

net:

port: 27017

bindIp: 0.0.0.0

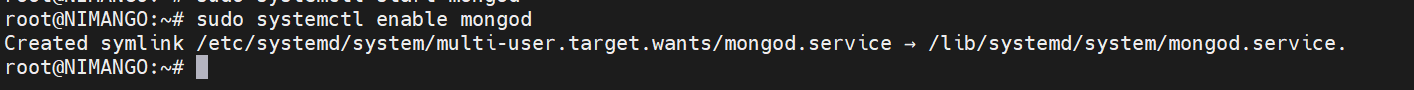
A screenshot of a computer

Description automatically generated

sudo systemctl start mongod

A black screen with white text

Description automatically generated

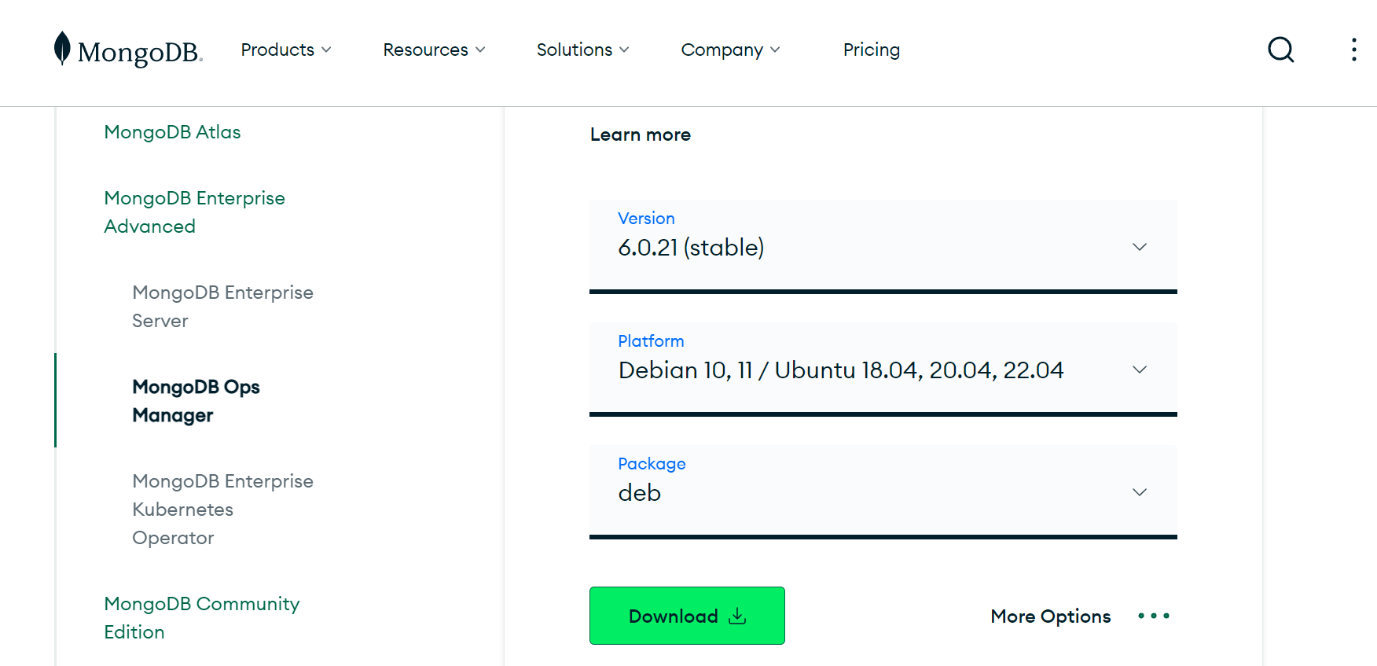
sudo systemctl enable mongod

Verify the connectivity using the below command on all the servers.

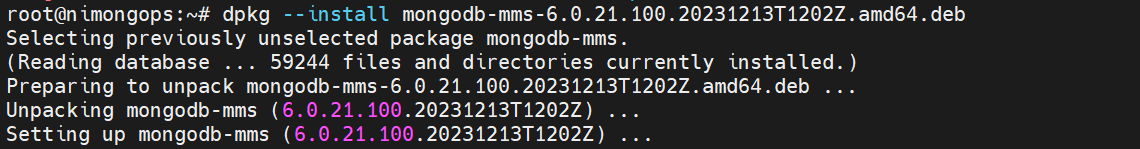
nc -zv <NodeIP> 27017

To install Ops Manager:

1. Download the latest version of the Ops Manager package.
2. Open your preferred browser to visit the [MongoDB Download Center](http://www.mongodb.com/try/download/ops-manager?jmp=docs).
3. From the Platforms drop-down menu, click Ubuntu 20.04.
4. From the Packages drop-down menu, click DEB for x86\_64 architecture.
5. Click Download.



1. The downloaded package is named mongodb-mms-<version>.x86\_64.deb, where <version> is the version number.
2. Install the .deb package by issuing the following command, where <version> is the version of the .deb package:

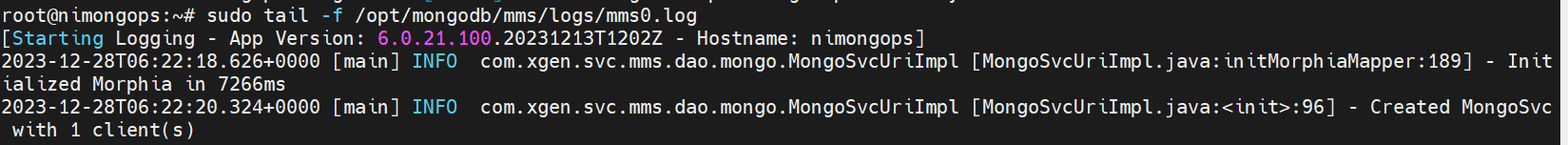
sudo dpkg --install mongodb-mms\_<version>\_x86\_64.deb

When installed, the base directory for the Ops Manager software is /opt/mongodb/mms/. The .deb package creates a new system user mongodb-mms under which the server will run.

1. Configure the Ops Manager connection to the Ops Manager Application Database.
2. On a server that is to run the Ops Manager, open /opt/mongodb/mms/conf/conf-mms.properties with root privileges and configure the settings described here, as appropriate. If you haven’t changed any of the default settings in the MongoDB server installation, do not change the settings.
3. Start Ops Manager.

sudo systemctl start mongodb-mms.service

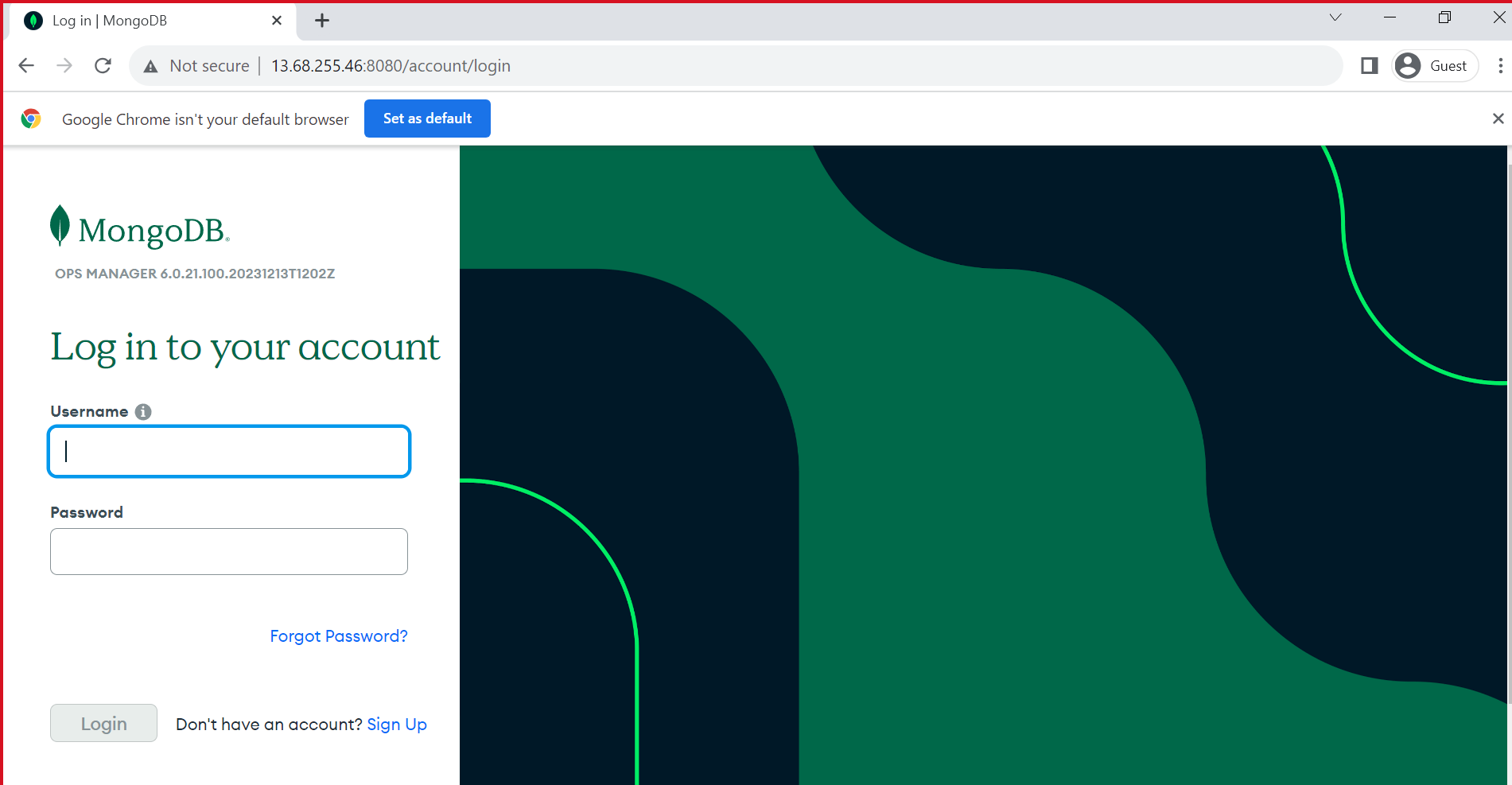
1. Check Ops Manager Logs to verify that the instance is up and running. It would take around 5 minutes for this.

sudo tail -f /opt/mongodb/mms/logs/mms0.log

1. Open the Ops Manager home page and register the first user.

Enter the following URL in a browser, where <host> is the fully qualified domain name of the server:

http://<OpsManagerHost>:8080



Click the Sign-Up link and follow the prompts to register the first user and create the first project. The first user is automatically assigned the Global Owner role.

1. Configure Ops Manager.

Ops Manager walks you through several configuration pages. Required settings are marked with a red asterisk. Enter information as appropriate. When configuration is complete, Ops Manager opens the Deployment page.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Add Existing MongoDB Processes to the Ops Manager
2. Navigate to the Deployment page for your project.

A screenshot of a computer

Description automatically generated

1. Click Add and select Existing MongoDB Deployment.

A screenshot of a computer

Description automatically generated

1. Follow the prompts to add the deployment. You have to provide the primary node hostname, port and authentication details.
2. Install the MongoDB Agent to Manage Deployments
   1. Log in to the Ops Manager Application.

A screenshot of a computer

Description automatically generated

* 1. Click Manage your existing deployment.

A screenshot of a computer

Description automatically generated

* 1. A screenshot of a computer

     Description automatically generatedFrom the Install Agent menu, click Debian 8/9/10/11, Ubuntu 16.X/18.X/20.x/22.x DEB.
  2. From a system shell on the host that will run the MongoDB Agent, issue the following curl command to download the installer for Ubuntu 18.04/20.04/22.04 or Debian 9/10/11 for 64-bit x86:

A screenshot of a computer

Description automatically generated

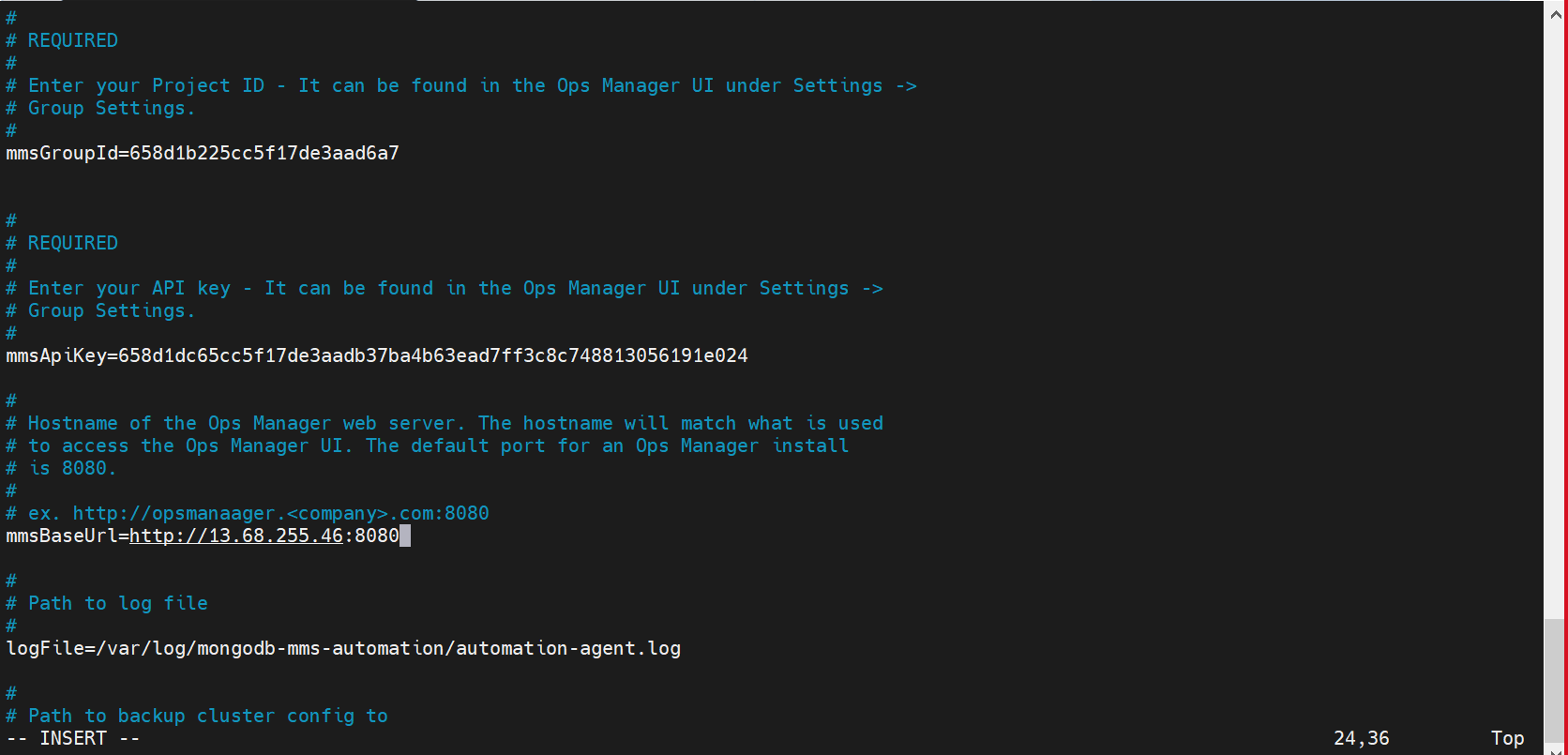
Copy and paste the commands in all the three server to install the MongoDB Agent

* 1. Edit the MongoDB Agent configuration file.

sudo vi /etc/mongodb-mms/automation-agent.config

* 1. Update the following configuration options:

1. mmsGroupId ProjectID of your project.
2. mmsApiKey Agent API key of your project.
3. mmsBaseUrl URL (hostname and port) of the Ops Manager Application.



A screenshot of a computer

Description automatically generated

* 1. Prepare the data directory.

The data directory stores MongoDB data. For an existing MongoDB deployment, ensure that the directory is owned by the mongodb user. If no MongoDB deployment exists, create the directory, and set the owner. To create a data directory and set the owner as the mongodb user:

sudo mkdir -p /data



 sudo chown mongodb:mongodb /data

* 1. Start the MongoDB Agent.

 sudo systemctl start mongodb-mms-automation-agent.service

* 1. A screenshot of a computer

     Description automatically generatedClick Verify Agent to make sure the MongoDB Agent can connect to the Ops Manager.
  2. Click Done.
  3. Repeat the installation procedure for each MongoDB host.

1. Back up a Deployment
2. Replica Set Requirements

A Replica Set must:

1. Be monitored by the Ops Manager.
2. Run MongoDB Enterprise with an FCV of 4.2 or later.
3. Have an active primary node.
4. A screenshot of a computer

   Description automatically generatedHave one node with WiredTiger set as its storage engine.
5. If you have not yet enabled Ops Manager Backup, click Begin Setup and complete the wizard.

For the HEAD directory, create a directory in Mongo Ops Manager VM (e.g.: /home/<username>/backup) and provide access to the mongodb-mms user using the below command.

sudo chown mongodb-mms:mongodb-mms /home/<username>/backup



A screenshot of a computer

Description automatically generated

1. Create an administrative user on the mongo ops manager mongodb:

A screen shot of a computer

Description automatically generated mongosh

use admin

A black background with white text

Description automatically generated

db.createUser({

user: "admin-user",

pwd: passwordPrompt(),

roles: [{ role: "root", db: "admin" }, "readWriteAnyDatabase"]

A computer screen with white text

Description automatically generated})

exit



1. Return to Projects and click on Continuous Backup from the left menu.
2. Click on Begin Setup and select your replica set.
3. A screenshot of a computer

   Description automatically generatedStart backing up the process.