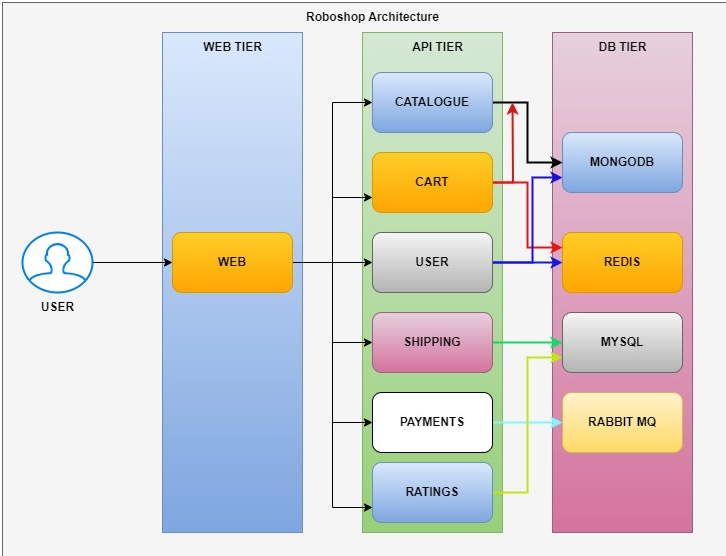
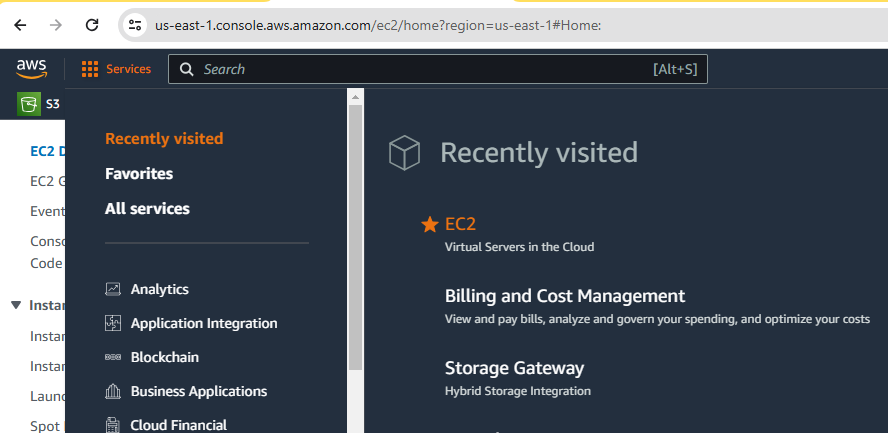
Session 7 – Roboshop Confiuration

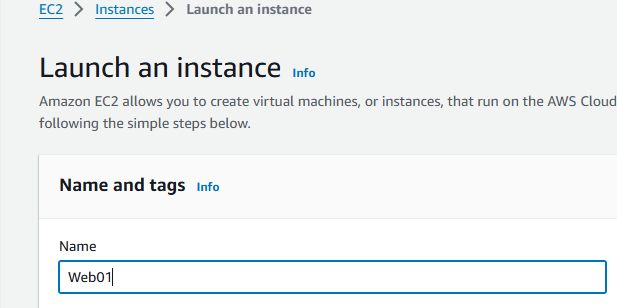


Click on ‘services’ in AWS Console and select EC2

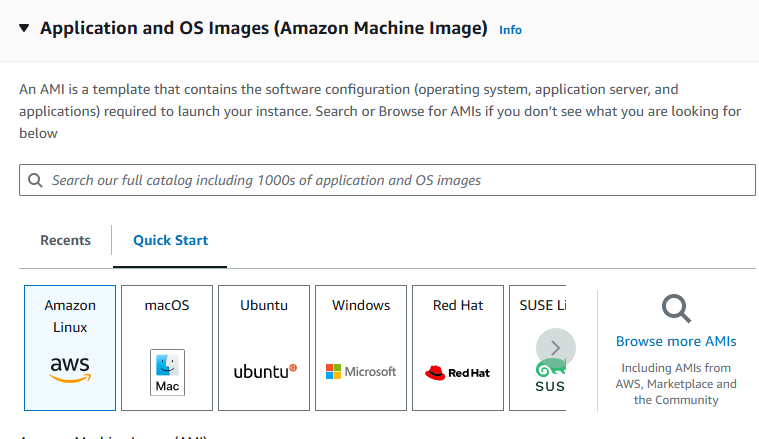


Click on ‘Launch Instance’ option:

Provide name of for EC2 instance as ‘Web01’ in the Name and Tags field

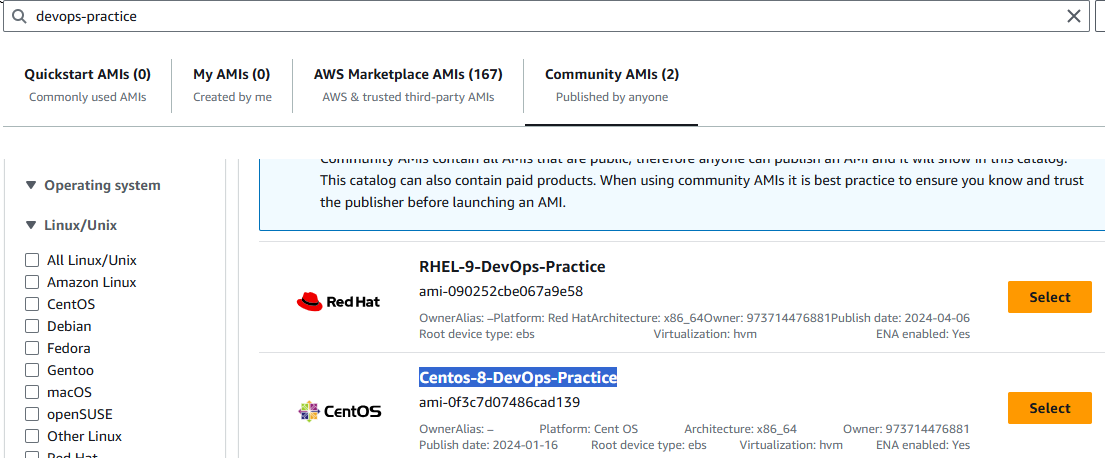


Click on “Browse more AMIs“ option



Provide “devops-practice” in the search box, click on ‘Community AMIs’

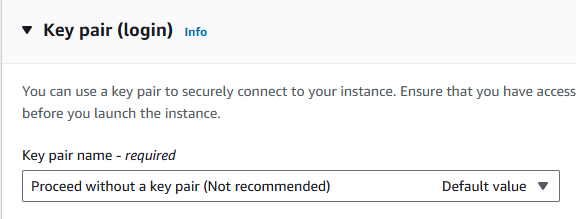
and select **‘Centos-8-DevOps-Practice’** AMI.



Credential details for the newly selected AMI **‘Centos-8-DevOps-Practice’**

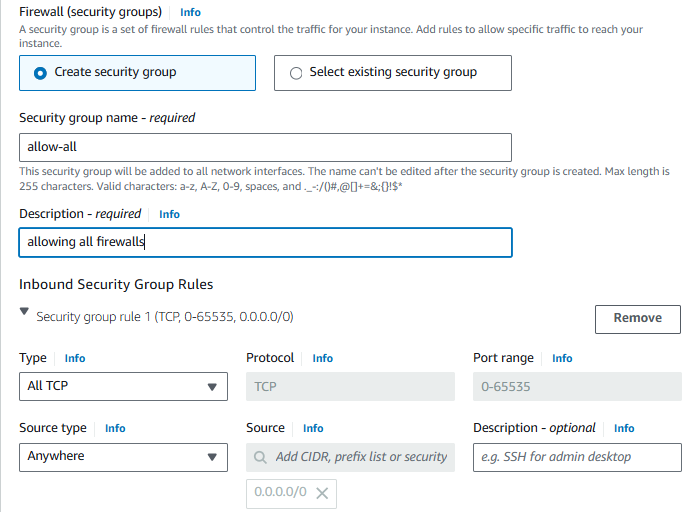
**Username/pwd: centos/DevOps321**

Select below option in the key-pair(login) section:



Here we are using username/password not the ‘key-pair’ to avoid time consume.

Create Security Group as below:



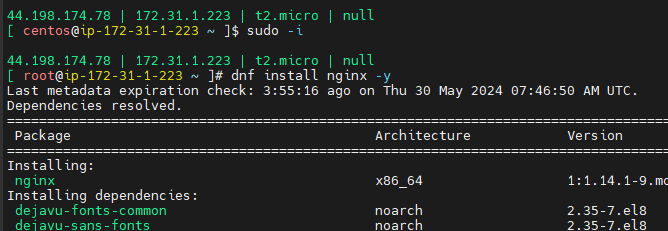
Click on Launch instance

Go to the Github repository ‘roboshop-documentation’ and configure the web tier

<https://github.com/arunnine/roboshop-documentation/blob/master/01-web.MD>

Install Nginx:

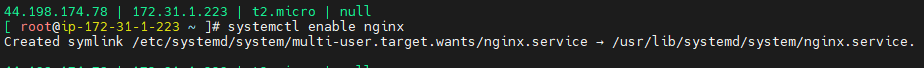
yum install nginx -y

****

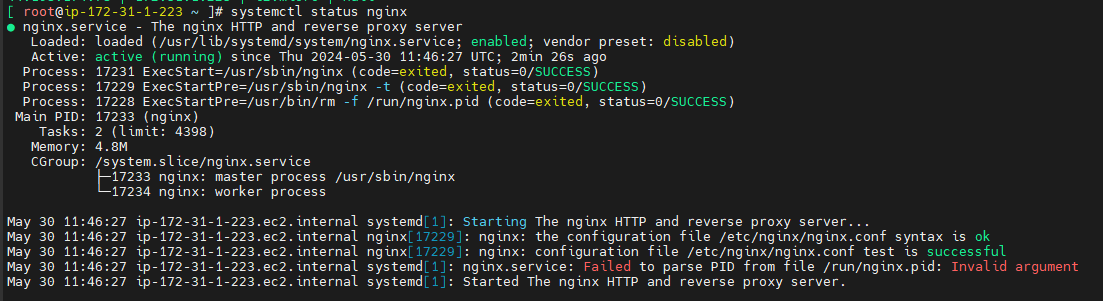
Start & Enable Nginx service:

systemctl enable nginx

systemctl start nginx

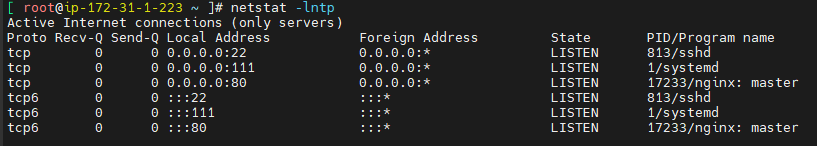






Use below command to check which port is open

netstat -lntp

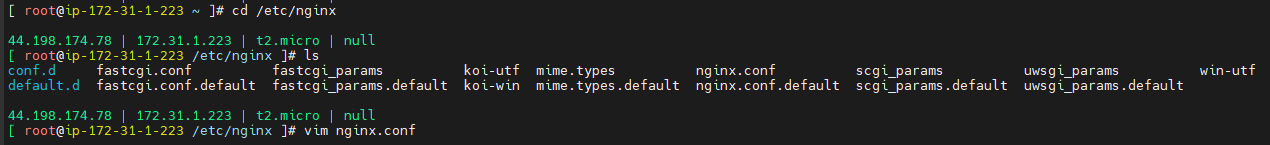


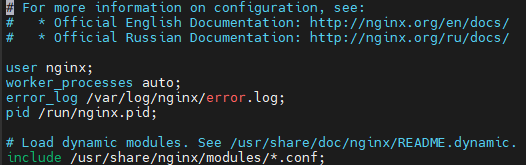
Try to access the service once over the browser and ensure you get some default content.

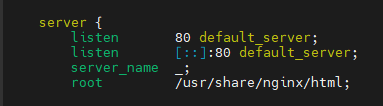
http://<public-IP>:80



To check nginx error log and installed location follow below commands:

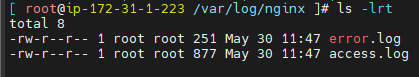






Error log:







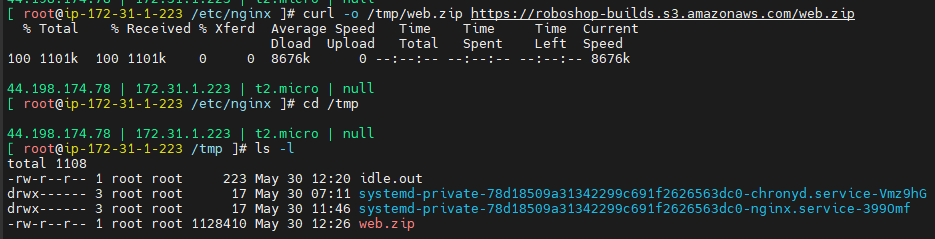
Remove the default content that web server is serving using below command.

rm -rf /usr/share/nginx/html/\*



Download the frontend content

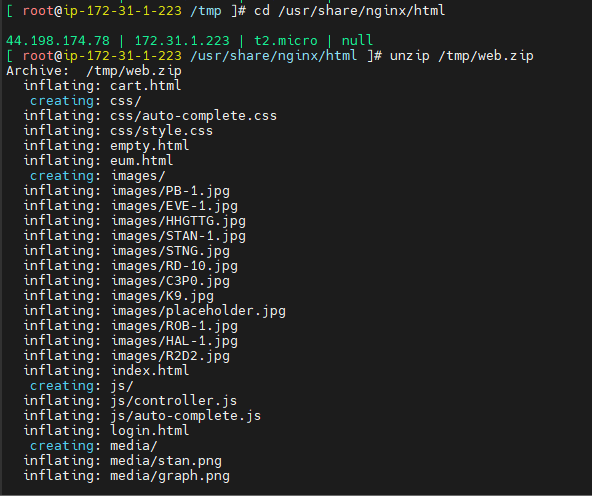
curl -o /tmp/web.zip https://roboshop-builds.s3.amazonaws.com/web.zip



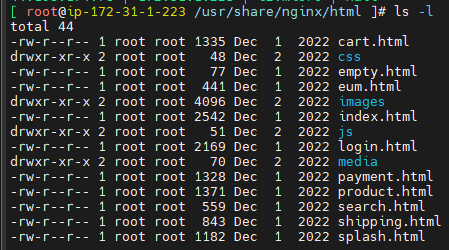
Extract the frontend content:

cd /usr/share/nginx/html

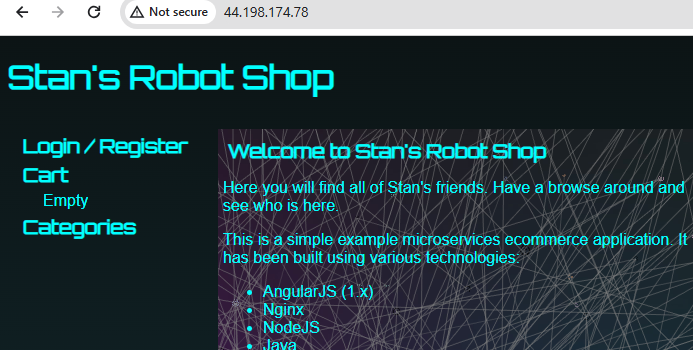
unzip /tmp/web.zip



We can see the default content replaced with roboshop content



Try to access the nginx service once more over the browser and ensure you get roboshop content.



Create Nginx Reverse Proxy Configuration.

vim /etc/nginx/default.d/roboshop.conf

Add the following content

vim /etc/nginx/default.d/+

proxy\_http\_version 1.1;

location /images/ {

expires 5s;

root /usr/share/nginx/html;

try\_files $uri /images/placeholder.jpg;

}

location /api/catalogue/ { proxy\_pass http://localhost:8080/; }

location /api/user/ { proxy\_pass http://localhost:8080/; }

location /api/cart/ { proxy\_pass http://localhost:8080/; }

location /api/shipping/ { proxy\_pass http://localhost:8080/; }

location /api/payment/ { proxy\_pass http://localhost:8080/; }

location /health {

stub\_status on;

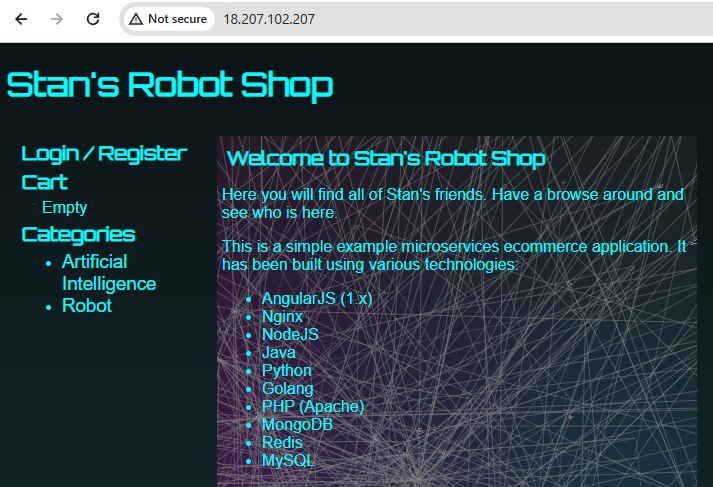
access\_log off;

}

Ensure you replace the localhost with the actual IP address of those component server. Word localhost is just used to avoid the failures on the Nginx Server.

Restart Nginx Service to load the changes of the configuration.

systemctl restart nginx



### MongoDB:

\* Developer has chosen the database MongoDB.

\* Hence, we are trying to install it up and configure it. </br>

\*\*NOTE: Versions of the DB Software you will get context from the developer, meaning we need to check with developer. Developer has shared the version information as MongoDB-4.x\*\*

Create a new EC2 instance for MongoDB with t3.medium bcoz database should have more capacity server.

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

Setup the MongoDB repo file

```

vim /etc/yum.repos.d/mongo.repo



Paste below content in the mongo.repo file

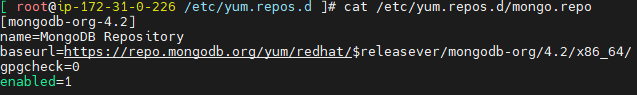
[mongodb-org-4.2]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.2/x86\_64/

gpgcheck=0

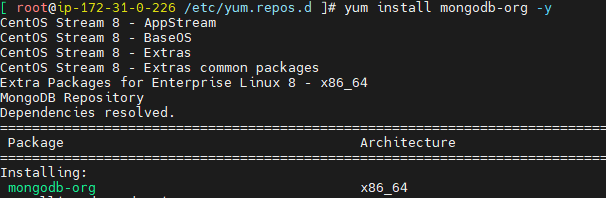
enabled=1



Install MongoDB

```

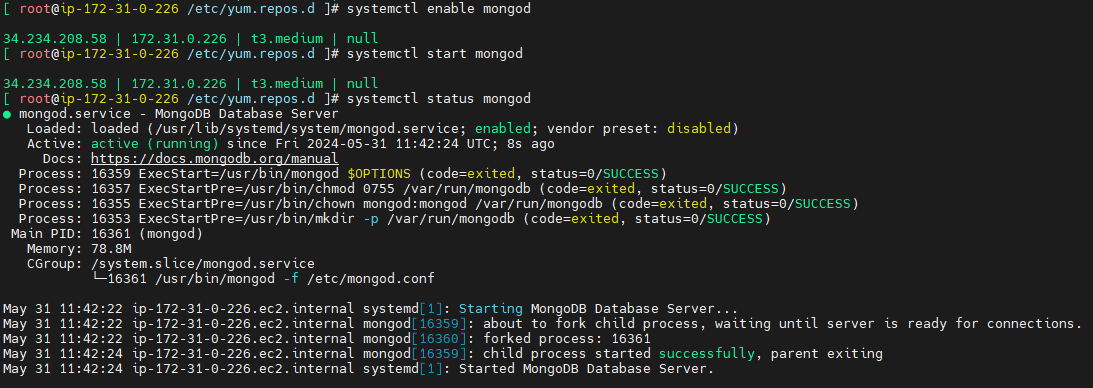
dnf install mongodb-org -y

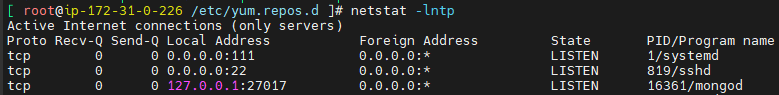


Start & Enable MongoDB Service

systemctl enable mongod

systemctl start mongod





Note: MongoDB will open port no 27017

Usually, MongoDB opens the port only to localhost (127.0.0.1), meaning this service can be accessed by the application that is hosted on this server only. However, we need to access this service to be accessed by another server (remote server), So we need to change the config accordingly.

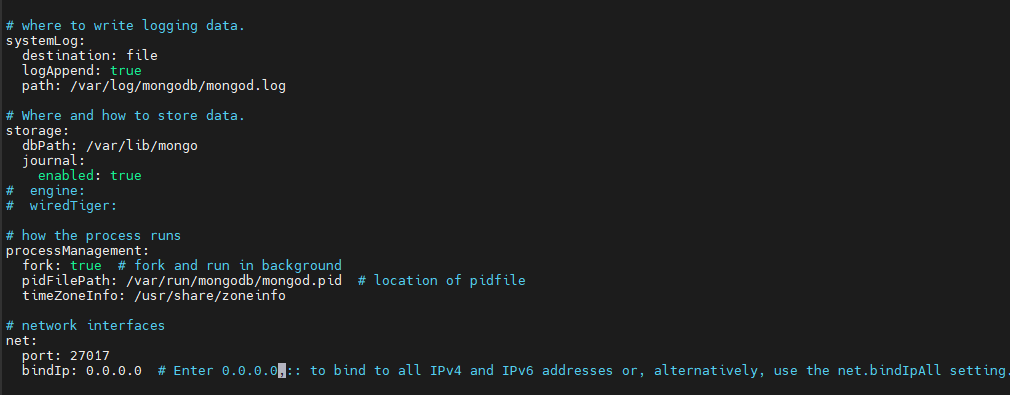
Update listen address from 127.0.0.1 to 0.0.0.0 in /etc/mongod.conf

You can edit file by using

```

vim /etc/mongod.conf

```

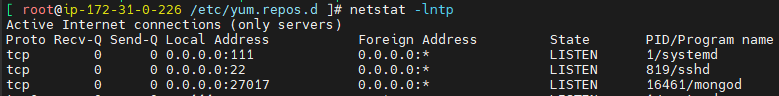


Restart the service

```

systemctl restart mongod

**```**

****

Now, MongoDB accepting connections from the internet.

Create a new EC2 instance for Catalogue.

**### Catalogue:**

Catalogue is a microservice that is responsible for serving the list of products that displays in roboshop application.

Developer has chosen NodeJs, check with developer which version of NodeJS is needed. Developer has set a context that it can work with NodeJS >18

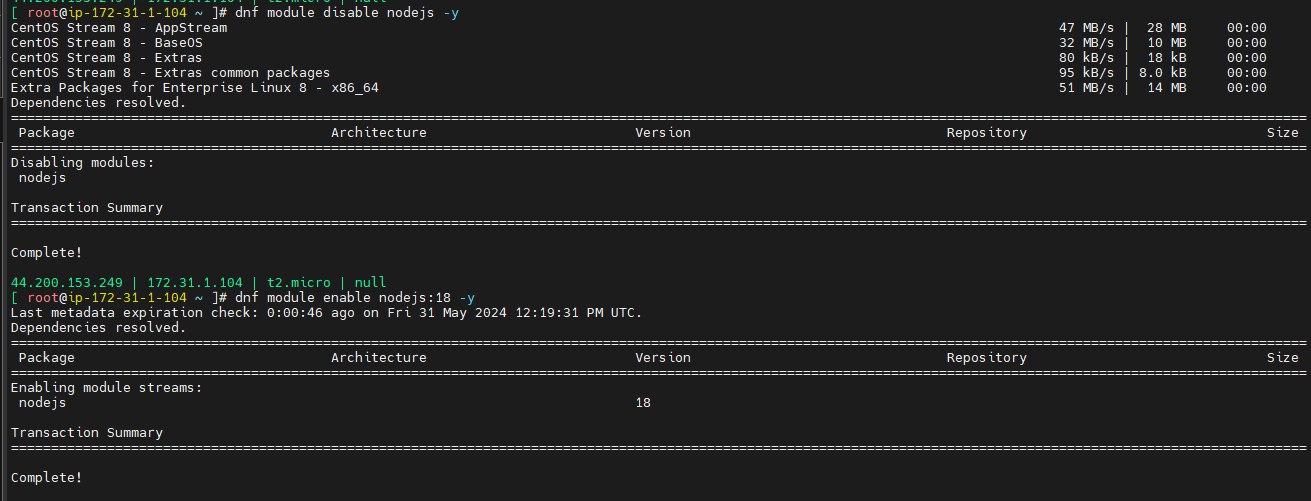
Install NodeJS, by default NodeJS 10 is available, we would like to enable 18 version and install list.

\*You can list modules by using dnf module list\*

```

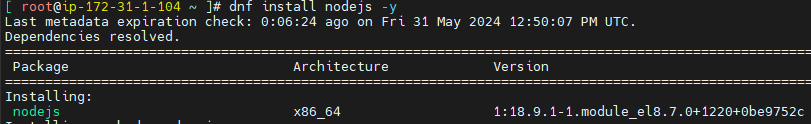
dnf module disable nodejs -y

dnf module enable nodejs:18 -y



Install NodeJS

dnf install nodejs -y



Configure the application.

Our application developed by the developer of our company and it is not having any RPM software just like other software, so we need to configure every step manually

Add application User

```

useradd roboshop

```

User roboshop is a function / daemon user to run the application. Apart from that we don't use this user to login to server.

Also, username roboshop has been picked because it more suits to our project name.

We keep application in one standard location. This is a usual practice that runs in the organization.

Let’s setup an app directory.

```

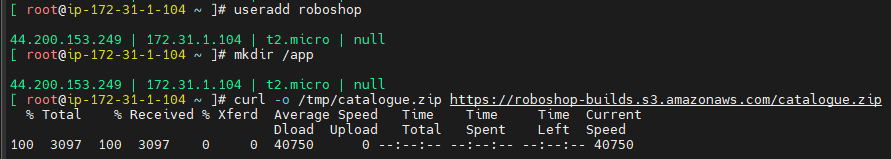
mkdir app

```

Download the application code to created app directory.

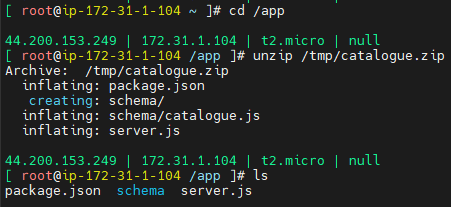
```

curl -o /tmp/catalogue.zip https://roboshop-builds.s3.amazonaws.com/catalogue.zip



cd /app

unzip /tmp/catalogue.zip



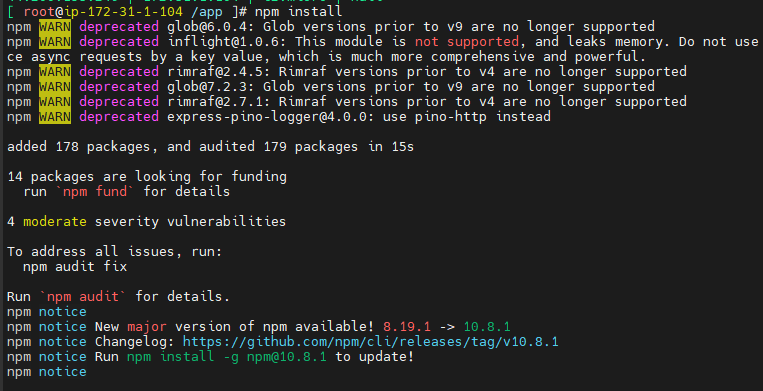
Every application is developed by development team will have some common softwares that they use as libraries. This application also has the same way of defined dependencies in the application configuration.

Let’s download the dependencies.

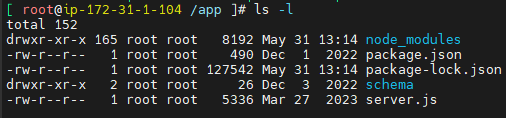
```

cd /app

npm install



node\_modules folder will be created as below



Note:

To find something in Linux use below command

find . -name “\*nginx\*”

We need to setup a new service in systemd so systemctl can manage this service

Setup SystemD Catalogue Service

```

vim /etc/systemd/system/catalogue.service



```

Note: Provide private IP address in the below MONGO\_URL section\*\*

[Unit]

Description = Catalogue Service

[Service]

User=roboshop

Environment=MONGO=true

Environment=MONGO\_URL="mongodb://<MONGODB-SERVER-IPADDRESS>:27017/catalogue"

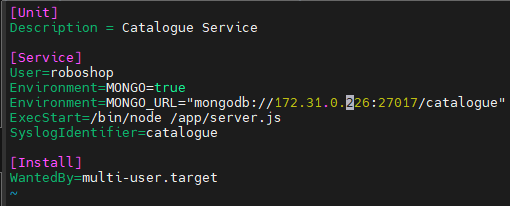
ExecStart=/bin/node /app/server.js

SyslogIdentifier=catalogue

[Install]

WantedBy=multi-user.target

```



Load the service.

```

systemctl daemon-reload

```

Start the service.

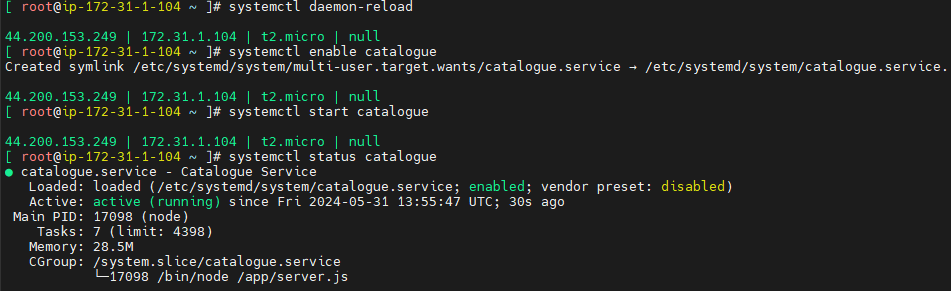
```

systemctl enable catalogue

```

```

systemctl start catalogue



Note: To check the logs, use below command

tail -f /var/log/messages

\* For the application to work fully functional we need to load schema to the Database.

\* Schemas are usually part of application code and developer will provide them as part of development.

We need to load the schema. To load schema, we need to install mongodb client.

To have it installed we can setup MongoDB repo and install mongodb-client

```

vim /etc/yum.repos.d/mongo.repo



```

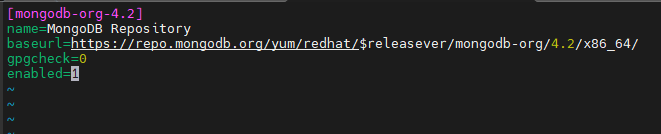
[mongodb-org-4.2]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.2/x86\_64/

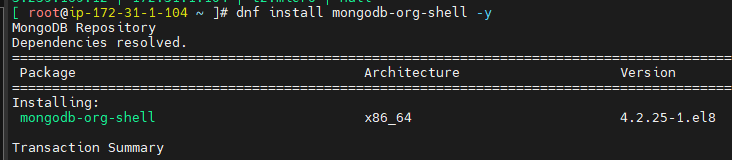
gpgcheck=0

enabled=1



```

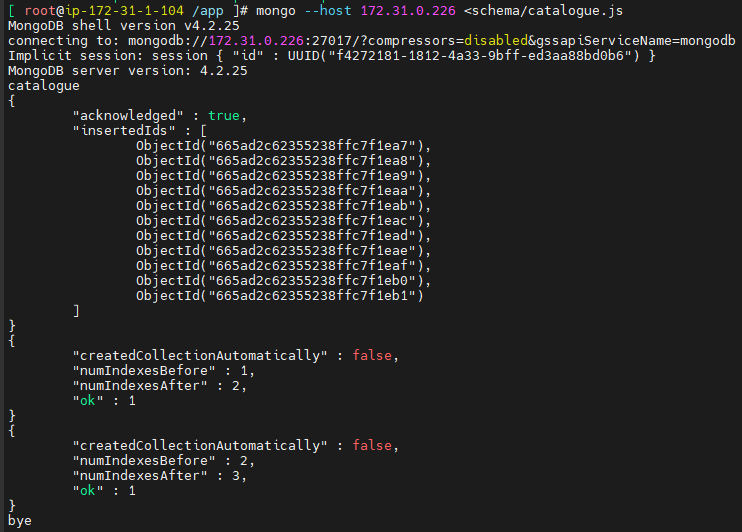
dnf install mongodb-org-shell -y



Load Schema

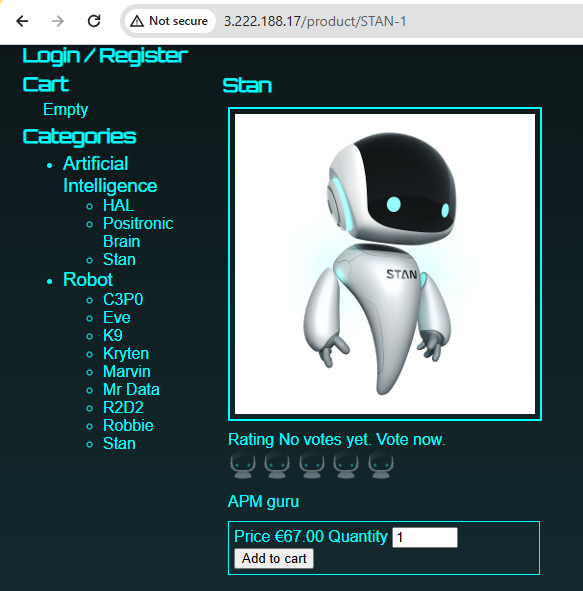
mongo --host MONGODB-SERVER-IPADDRESS </app/schema/catalogue.js

Note: Here provide mongo db server private IP address as below



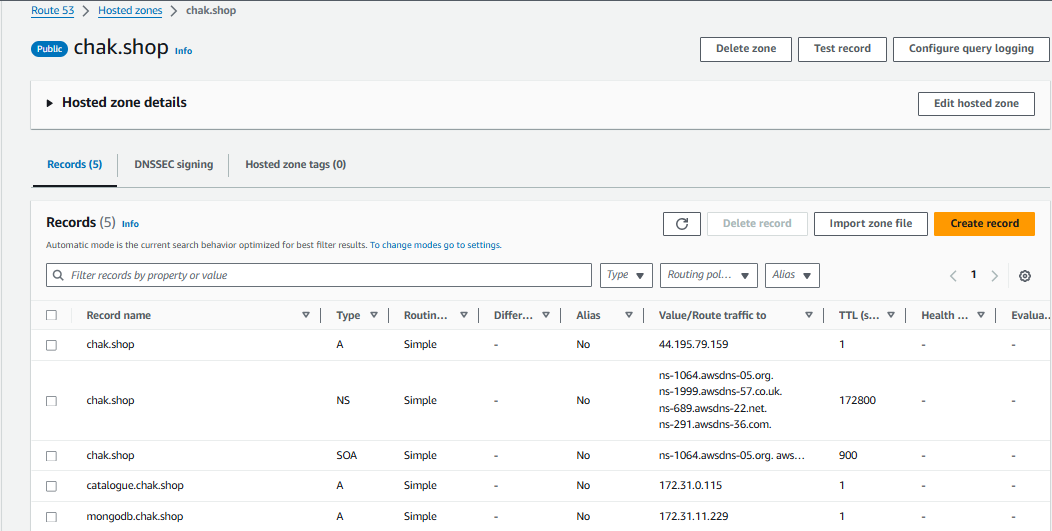


Now we can able to see the data in the webpage as below

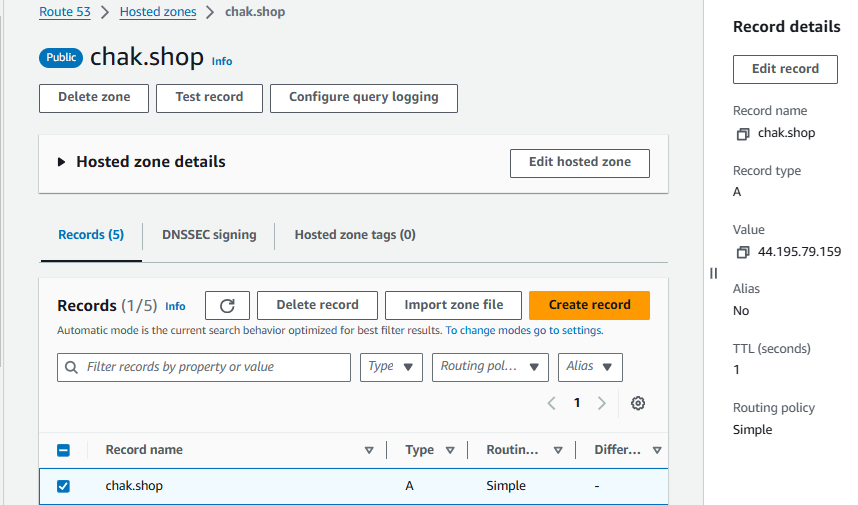


\*\*NOTE: You need to update catalogue server IP address in frontend(web server) configuration. Configuration file is /etc/nginx/default.d/roboshop.conf\*\*

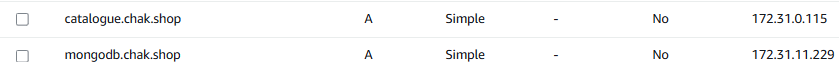




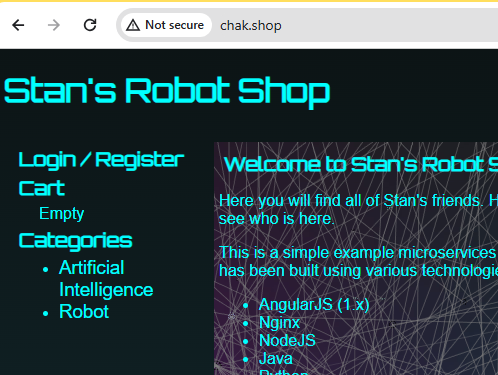
While creating the above record, we should provide public IP of the ‘Web’ server as the value to view the roboshop homepage.

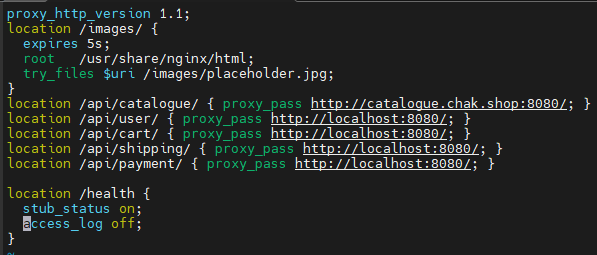


For below records, use private IP of the respective servers.



Browse the project with domain name as below:





Arun: chak.shop

|  |
| --- |
| ns13.domaincontrol.com |
| ns14.domaincontrol.com |

tail -f /var/log/messages 🡪 will show last 10 lines of the log message

less /var/log/messages 🡪 shows entire log, use ‘Shift+g’ to go to last line, gg to go to first line

Check for the roboshop interface logs 🡪 less /var/log/access.log (or) error.log

less /var/log/messages

MongoDB logs: cd /var/log/mongodb/mongod.log