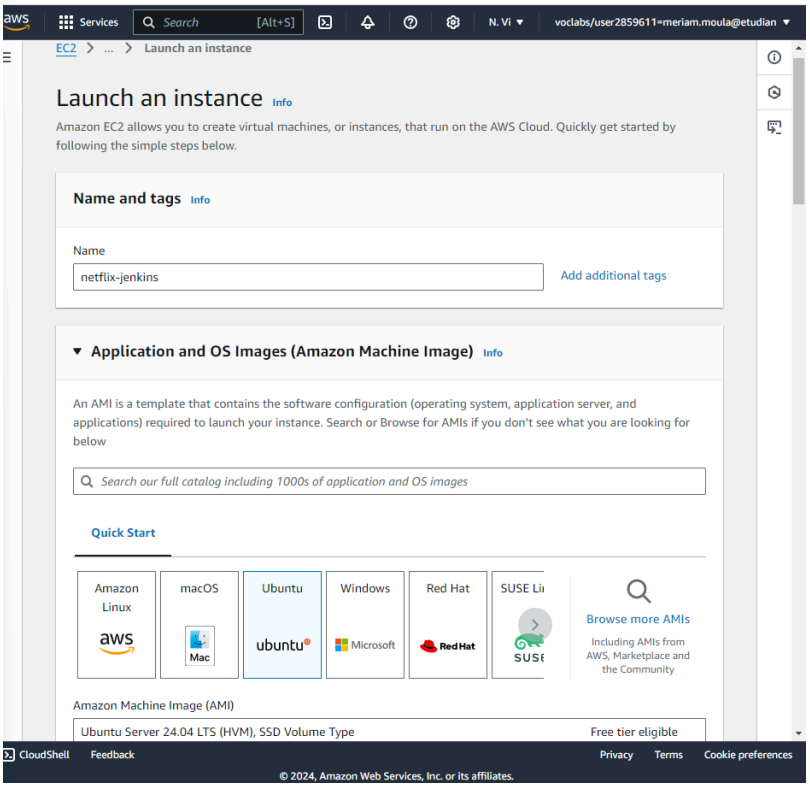
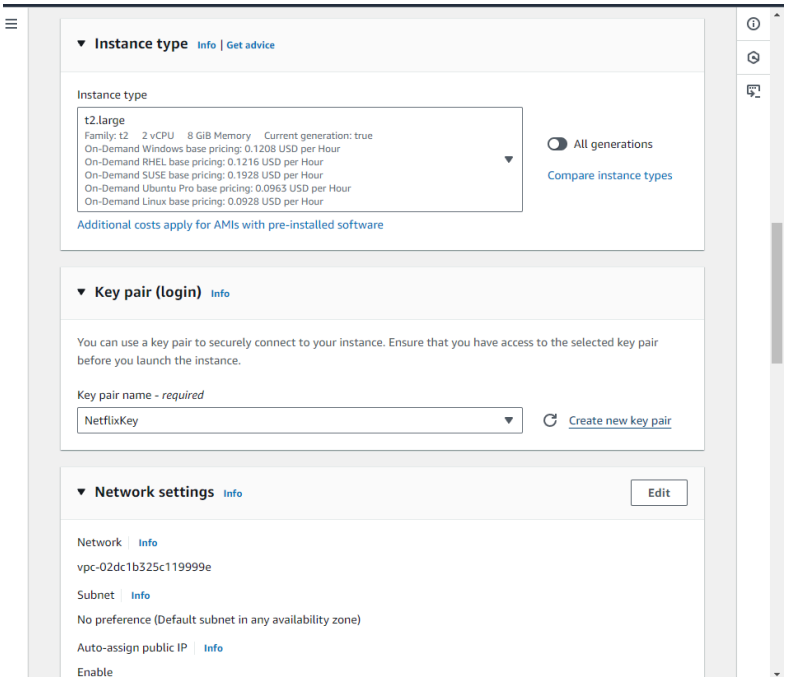
Note : Any key or secret informations or password will be removed or destroyed later , please make sure to use your own

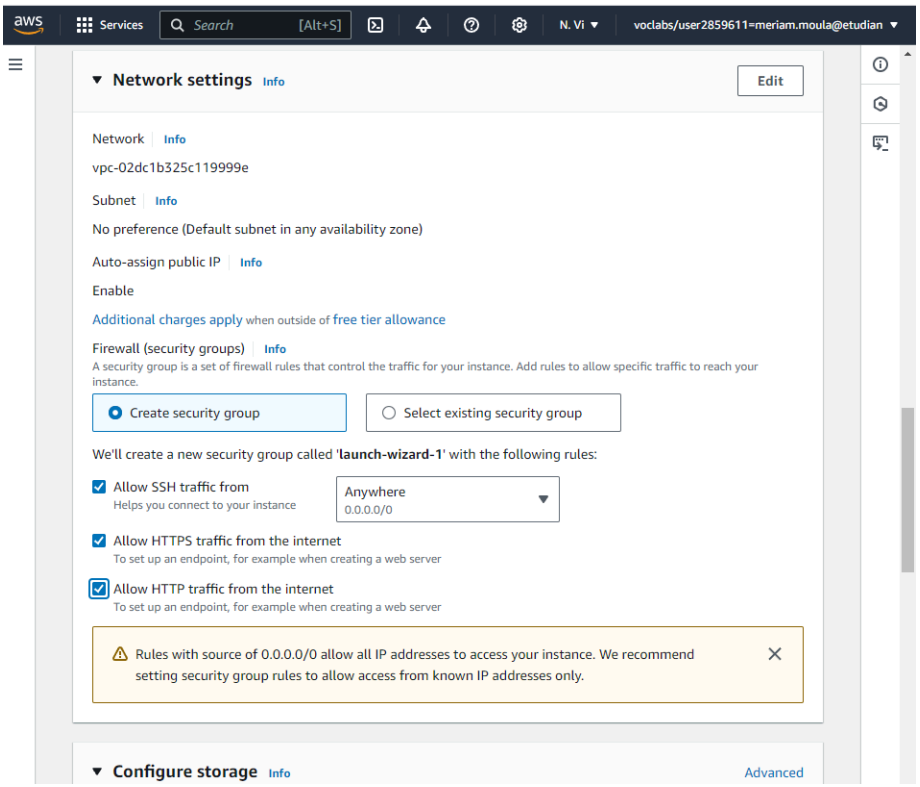
Initial Setup and Deployment

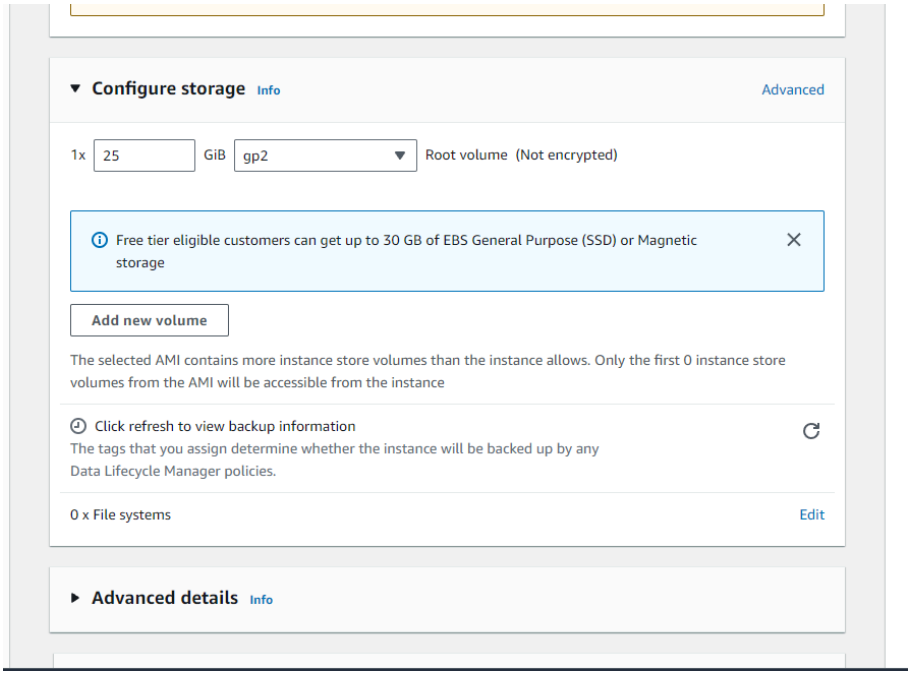
We will be working on an EC2 instance , where we will have our jenkins and SonarQube and Trivy running

Creating and launching the EC2 instance :




Instance type Info | Get advice
Instance type
t2.large
Family: t2 2 vCPU 8 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.1208 USD per Hour
On-Demand RHEL base pricing: 0.1216 USD per Hour
On-Demand SUSE base pricing: 0.1928 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0963 USD per Hour
On-Demand Linux base pricing: 0.0928 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
NetflixKey
Create new key pair
Network settings Info
Edit
Network Info
vpc-02dc1b325c119999e
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable

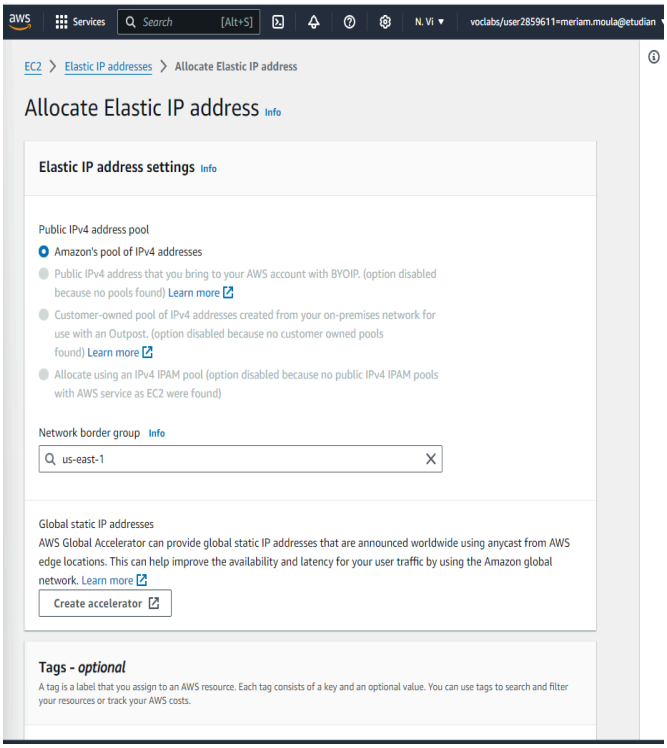

aws
Services
Search
[Alt+S]
N. Vi
voclabs/user2859611=meriam.moula@etudian
Network settings Info
Edit
Network Info
vpc-02dc1b325c119999e
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-1' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Configure storage Info
Advanced

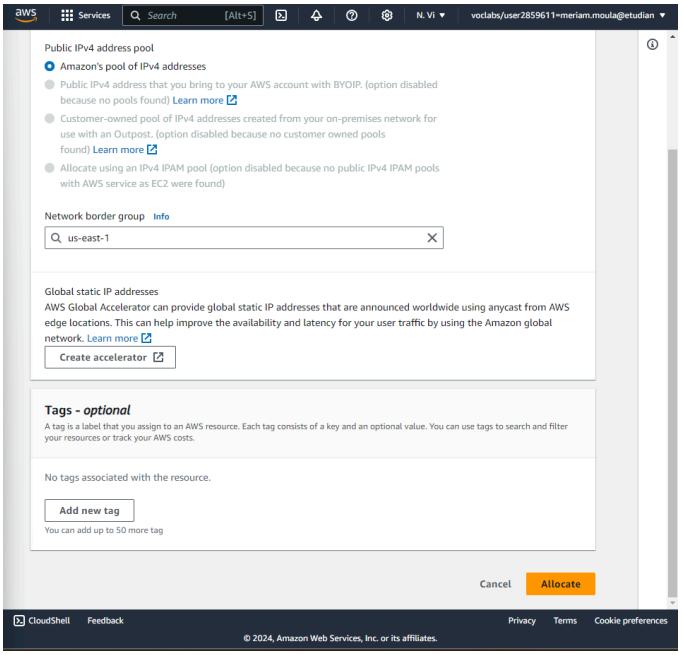


Creating and attaching an elastic IP address :

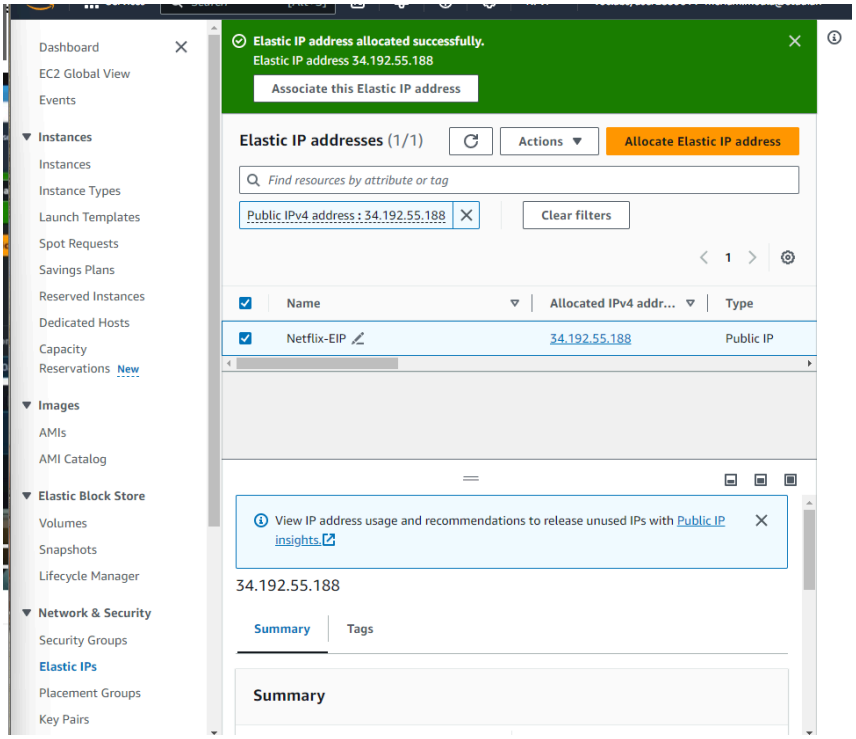
We will attach to the instance an elastic ip address so that we still have the same ip address even after stopping the instance

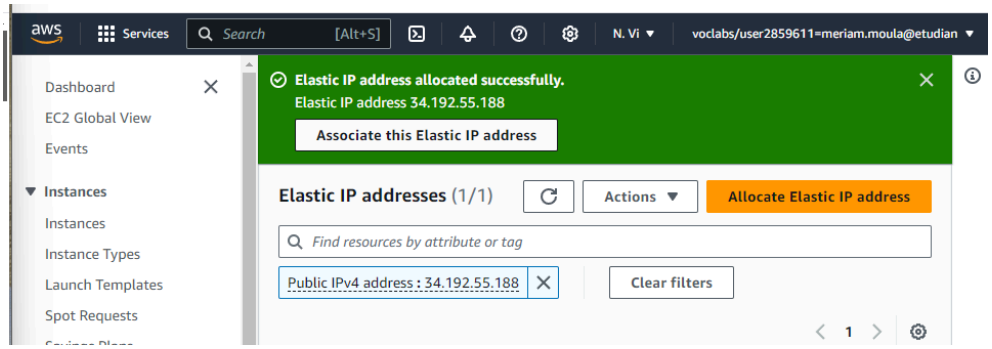
Allocating the EIP:



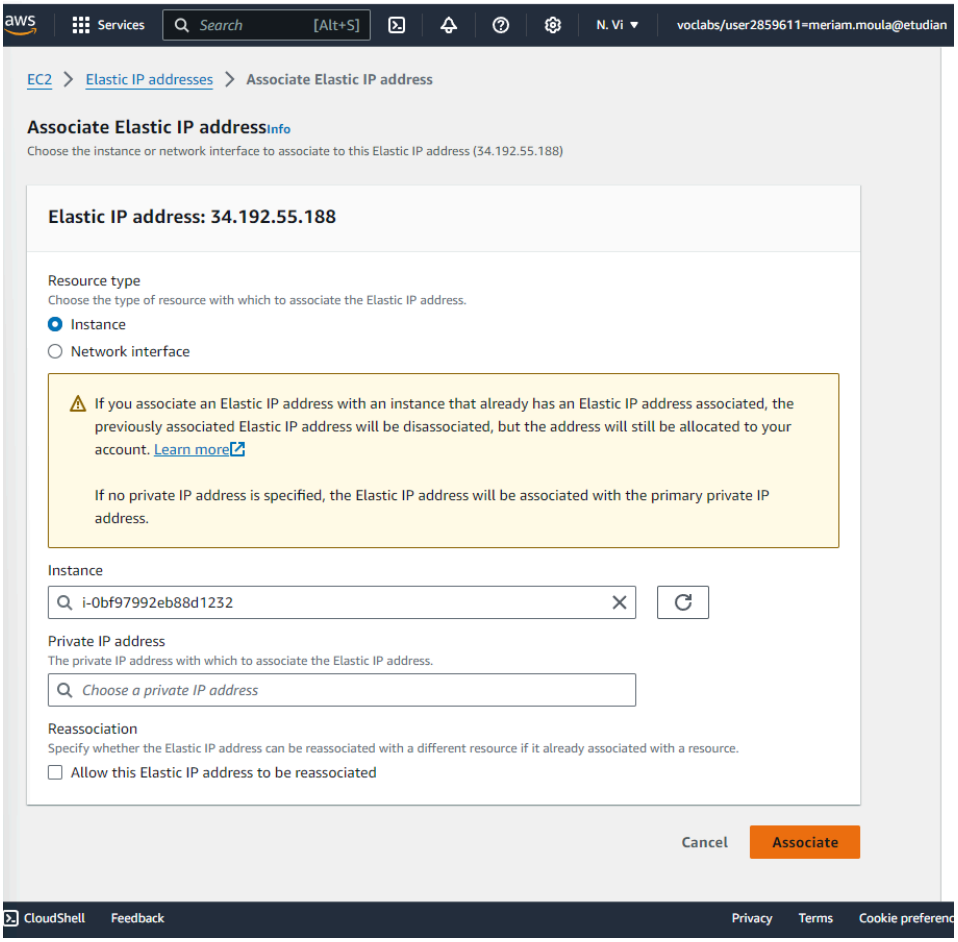


Renaming the EIP:



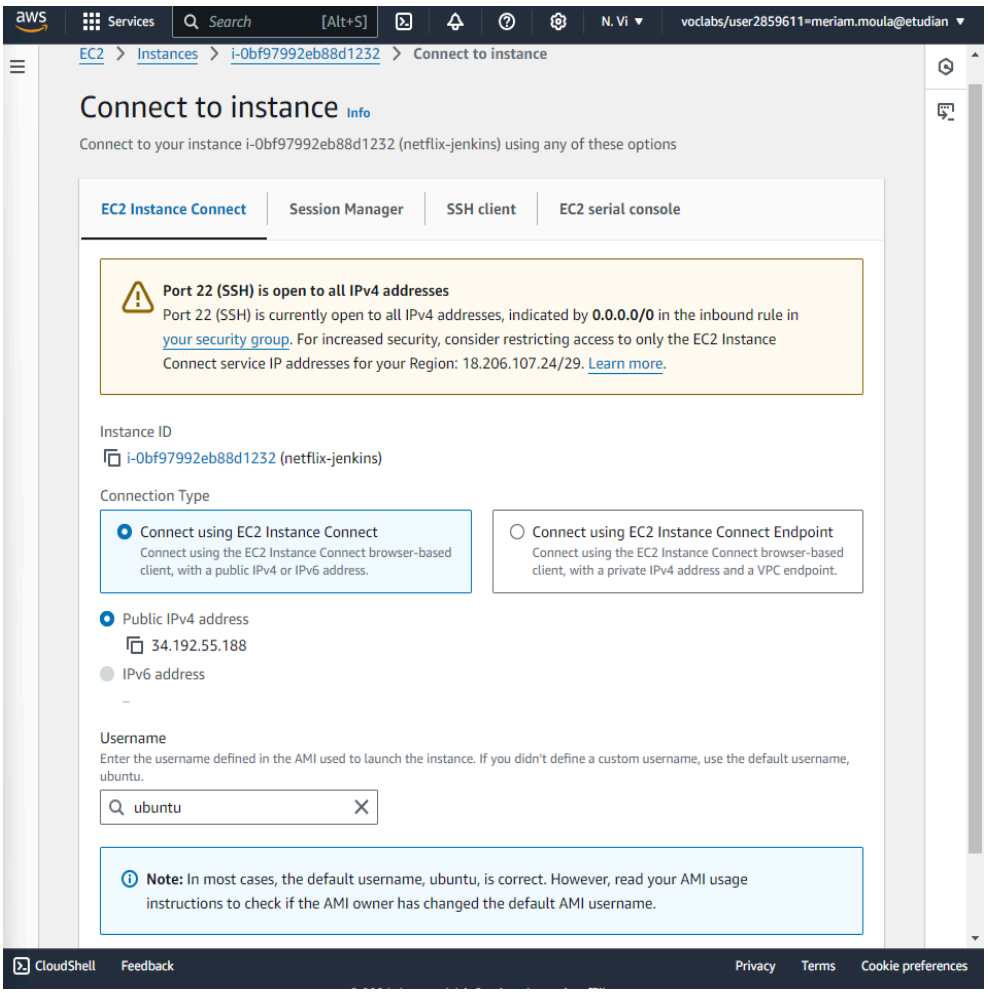


Associating The IP address To the instance

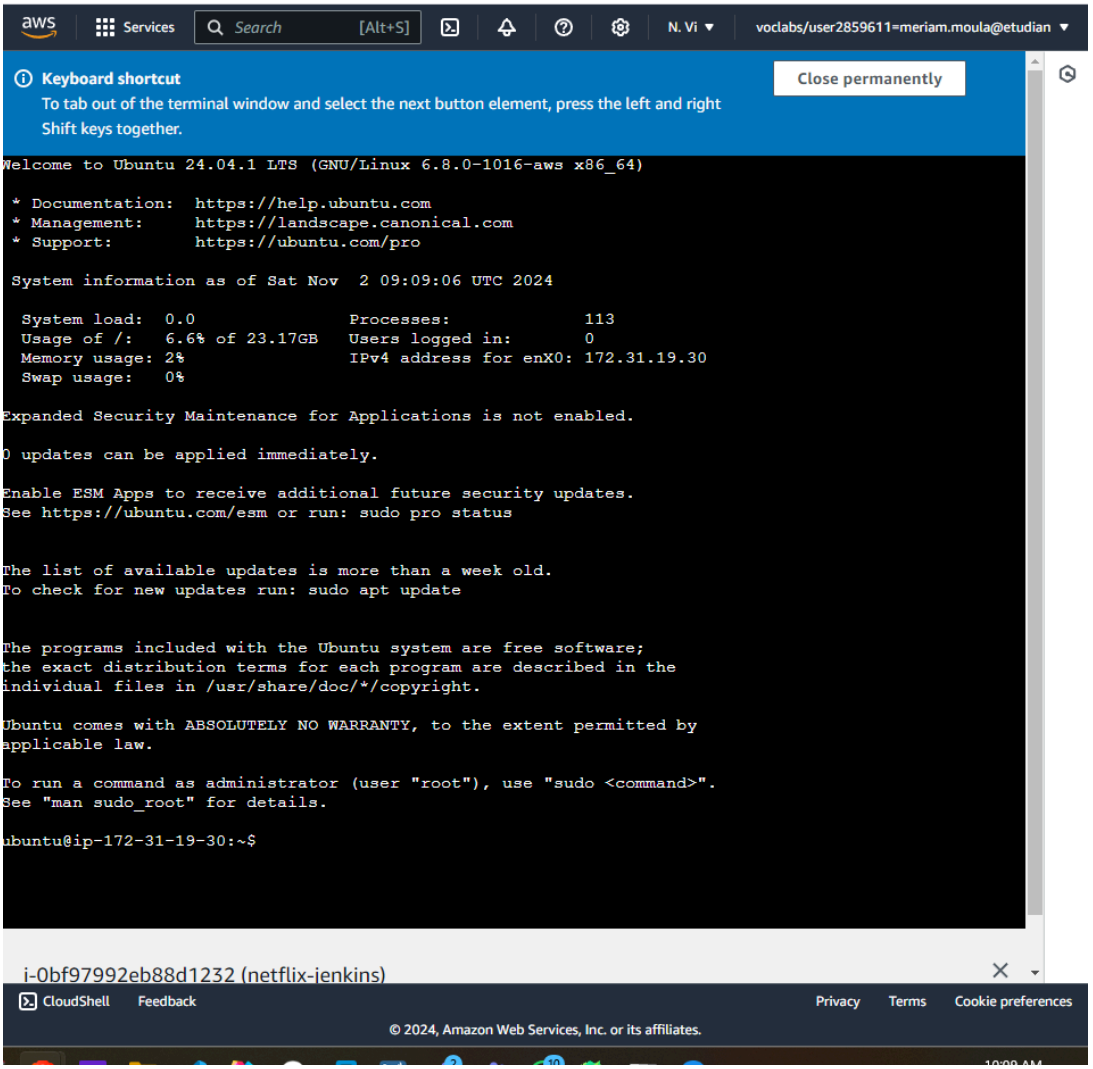


Connecting to the EC2 instance

Now we will be connecting to our EC2 instance :
there are four different ways to do that ,we can choose the one that suits us the most ,
I will be using EC2 Instance Connect which is supported by Ubuntu



When pressing on the Connect button we will ourselves inside of the EC2 machine :
The first thing you do when you are inside of the server is to update the packages by running this command : sudo apt update -y
And then you clone the repo to your machine using the git clone command



After cloning the project you can run it either with npm or docker using the docker image in Dockerfile
However Docker is not installed on the EC2 instance

Installing Docker on our instance :

Run this command to install Docker :

sudo apt-get update #Updating the packages
sudo apt-get install docker.io -y #Installing Docker
sudo usermod -aG docker $USER # Adding docker to the sudo group ,so that we get to run docker commands without sudo commands Replace with your system's username, e.g., 'ubuntu'
newgrp docker
sudo chmod 777 /var/run/docker.sock #Giving the permissions on this docker socket

Once this is done , you can access docker ,create docker images and run containers

Verifying the Docker installation:

You can verify the installation using this command :

docker --version

```
ubuntu@ip-172-31-19-30:~/DevSecOps-Project$ ls
Dockerfile  README.md   package.json  public  tsconfig.json       vercel.json     yarn.lock
Kubernetes  index.html  pipeline.txt  src     tsconfig.node.json  vite.config.ts
ubuntu@ip-172-31-19-30:~/DevSecOps-Project$
```

Running the application on our container :

Since we have the Dockerfile we can run our application as a container

We should make sure that our application is running so that we can integrate security

We use this command to build the image 👍

docker build -t netflix . # -t is for the tag and the . so that the building is from the current directory

We can verify that the image is built now with running this command : docker images

We can see that there is a netflix image created

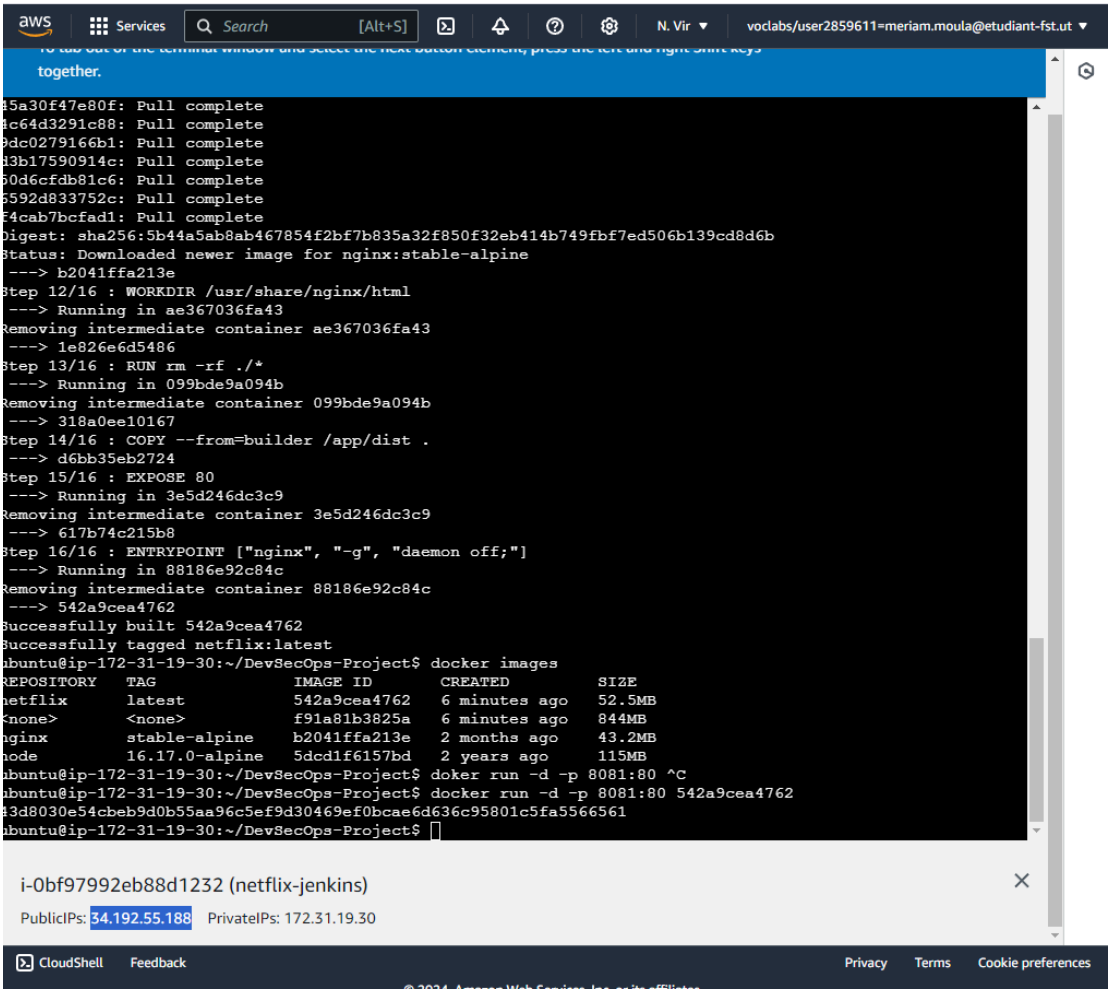
```
ubuntu@ip-172-31-19-30:~/DevSecOps-Project$ docker images
REPOSITORY   TAG              IMAGE ID       CREATED          SIZE
netflix      latest           542a9cea4762   6 minutes ago    52.5MB
<none>       <none>           f91a81b3825a   6 minutes ago    844MB
nginx        stable-alpine    b2041ffa213e   2 months ago     43.2MB
node         16.17.0-alpine   5dcd1f6157bd   2 years ago      115MB
```

We will run our application on a container :

use this command : docker run -d -p 8081:80 542a9cea4762 #Replace with your image id

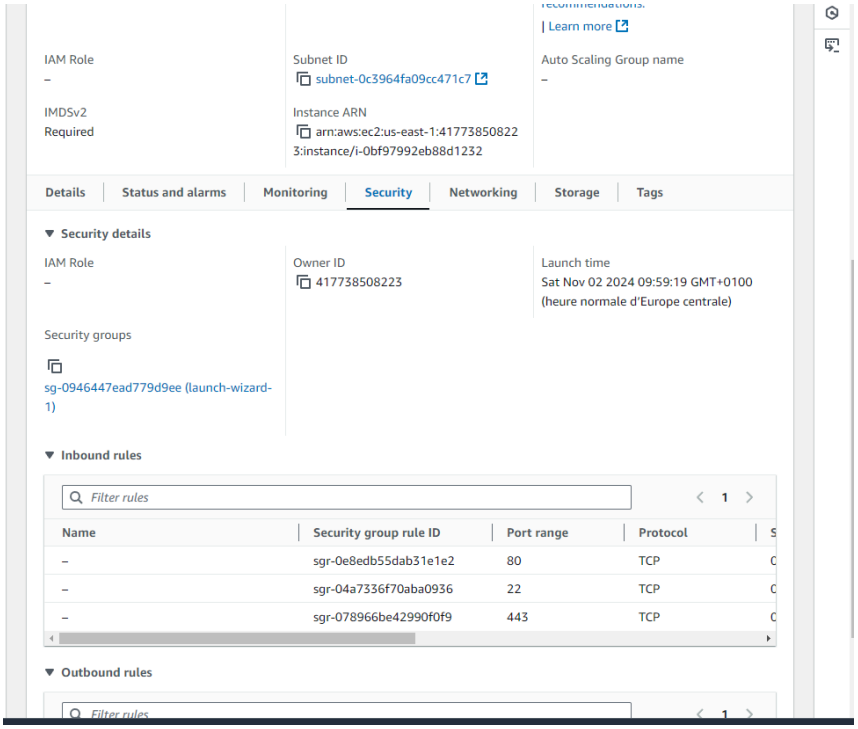
Accessing the application :

Use the ip down in the screen to access the app (Instead of localhost) 👍
and add the port that our EC2 machine would be using which is 8081

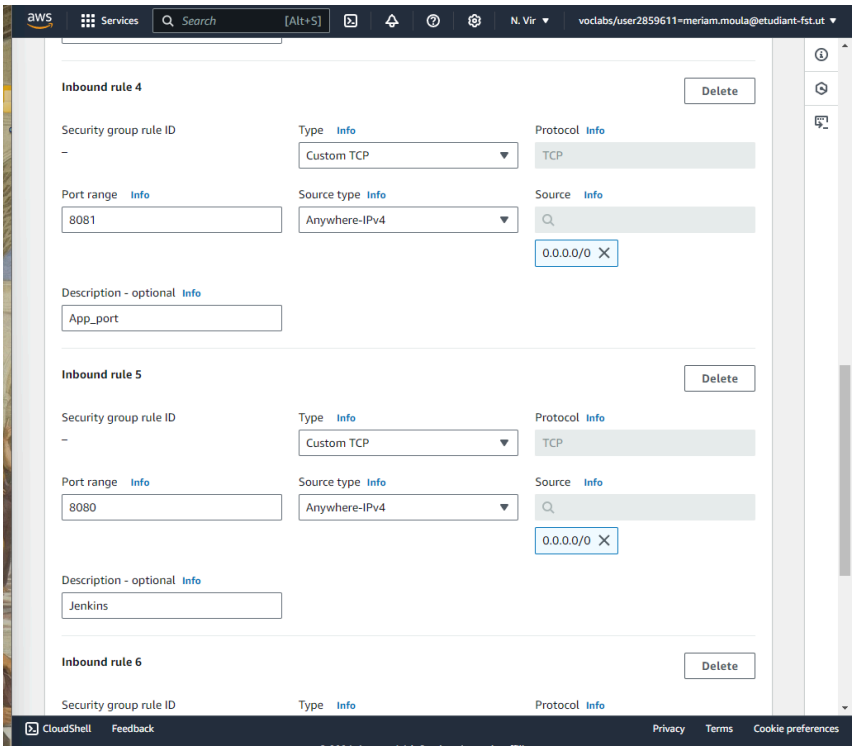


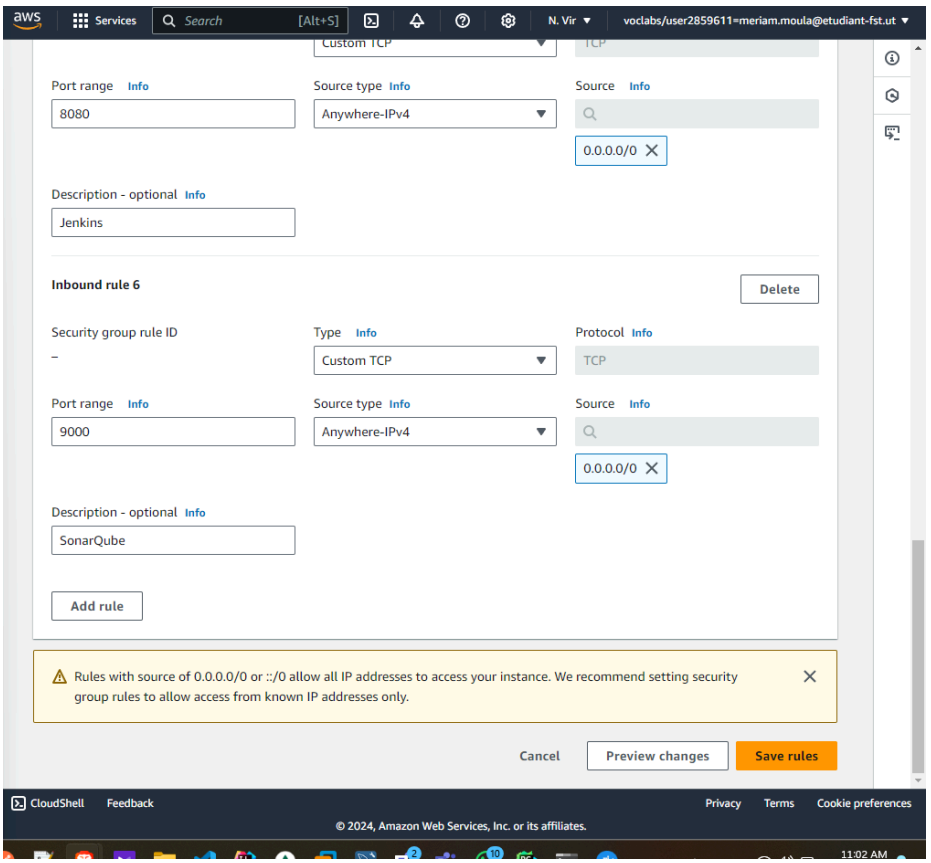
Setting the security groups:

The access will only work if we have port 8081 in our security group .
So in our EC2 instance , access the security group and add the rules



In here will add the inbound rules for the application that we are running , Jenkins and SonarQube





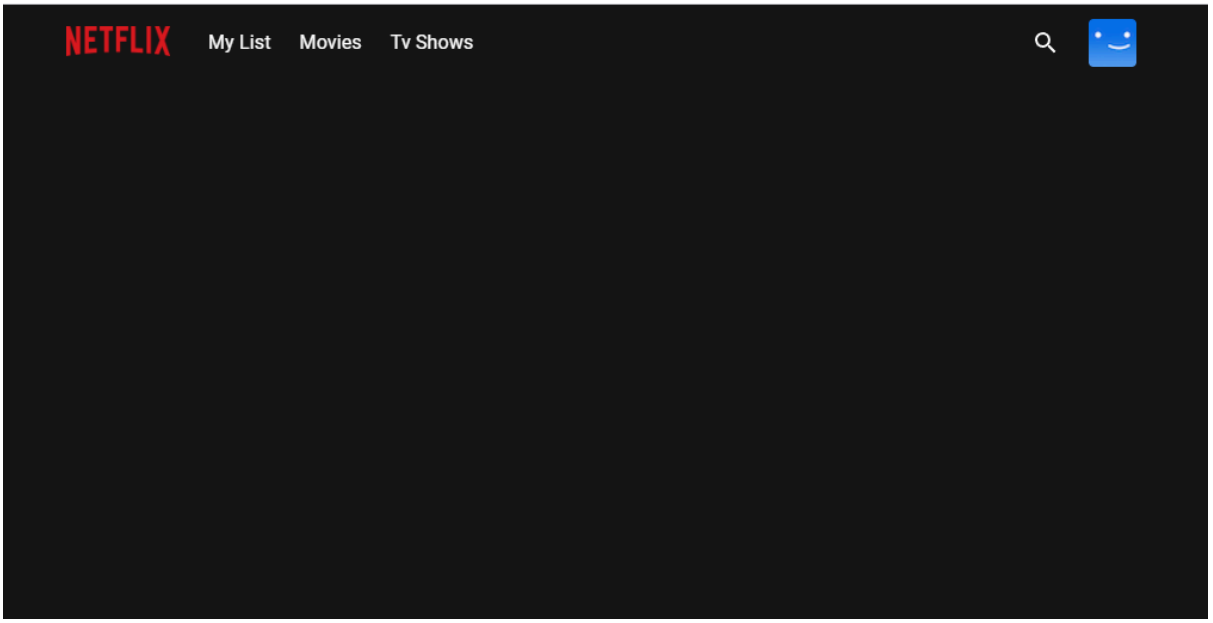
Adding the necessary Keys and credentials :

Now getting back the application
In the Dockerfile we can notice that there is this one argument that is asking you to have a TMDB api key

Dockerfile > ENV

```
1  FROM node:16.17.0-alpine as builder
2  WORKDIR /app
3  COPY ./package.json .
4  COPY ./yarn.lock .
5  RUN yarn install
6  COPY . .
7  ARG TMDB_V3_API_KEY
8  ENV VITE_APP_TMDB_V3_API_KEY=${TMDB_V3_API_KEY}
9  ENV VITE_APP_API_ENDPOINT_URL="https://api.themoviedb.org/
10 RUN yarn build
11
12 FROM nginx:stable-alpine
13 WORKDIR /usr/share/nginx/html
14 RUN rm -rf ./*
15 COPY --from=builder /app/dist .
16 EXPOSE 80
17 ENTRYPOINT ["nginx", "-g", "daemon off;"]
```

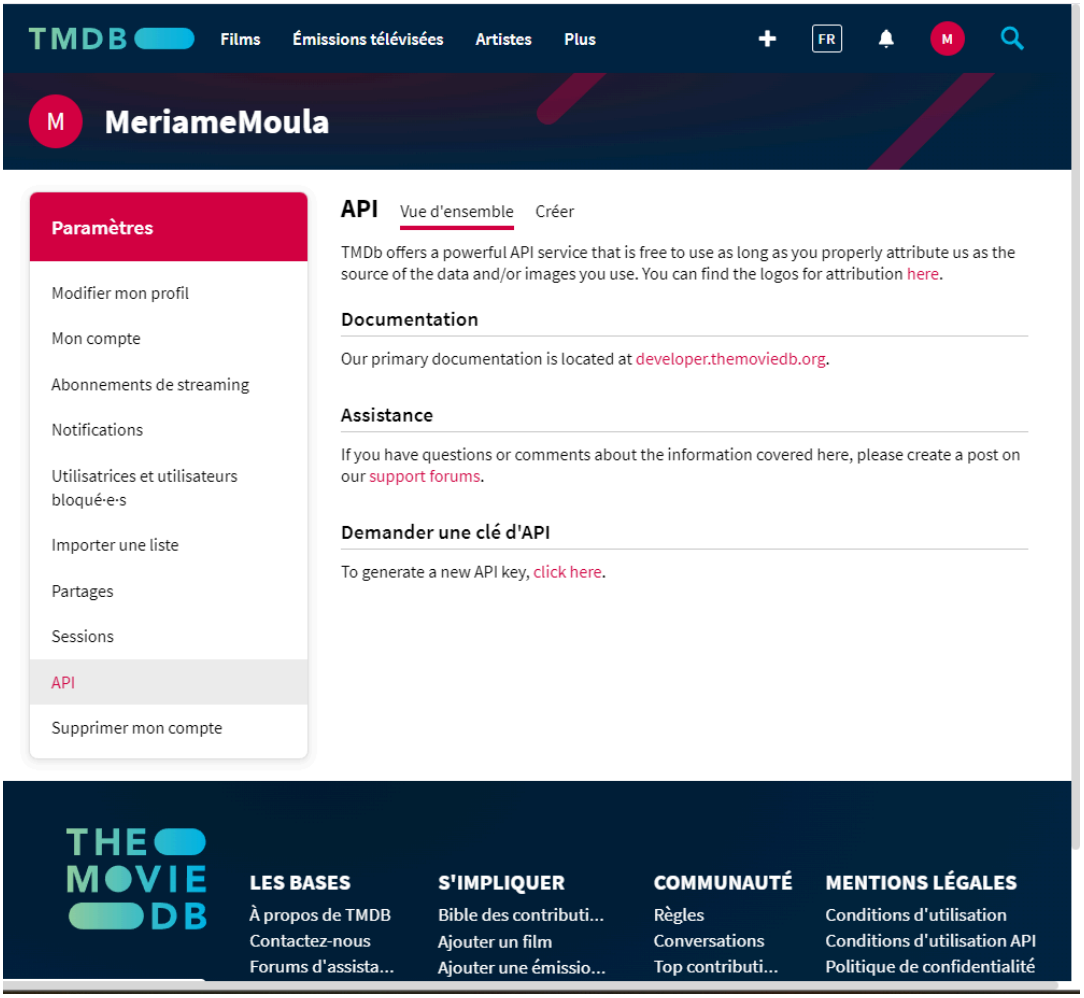
Unless we have this key , we are going to have a blank page like this :



Adding the API KEY :

Get the API Key:

- Open a web browser and navigate to TMDB (The Movie Database) website.
- Click on "Login" and create an account.
- Once logged in, go to your profile and select "Settings."
- Click on "API" from the left-side panel.
- Create a new API key by clicking "Create" and accepting the terms and conditions.
- Provide the required basic details and click "Submit."
- You will receive your TMDB API key.



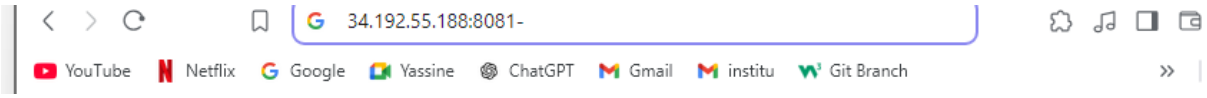
Now we run this command , If I have a server running am going to stop it and remove it , you can verify with this command :

docker ps
docker stop <container-id>
docker rm <container-id>

Now we recreate the image again using this command ,which is a build command but also passing an argument which is the api key :

docker build --build-arg TMDB_V3_API_KEY=<your-api-key> -t netflix .
we can verify the existence of the new image using
docker images
and then run the container with the new image
docker run -d -p 8081:80 netflix

Make sure to remove the /browse from the path



Security

Moving now to the Sec Part :
We will be using two main tools : SonarQube and Trivy

SonarQube

SonarQube is a tool used for **continuous code quality and security analysis**. It scans codebases to identify potential issues, such as bugs, security vulnerabilities, code smells, and duplications. SonarQube integrates with various programming languages and CI/CD pipelines, providing a dashboard with detailed insights on code quality, maintainability, and reliability. It's valuable for teams aiming to maintain high standards and improve code quality over time.

Trivy

Trivy is an **open-source vulnerability scanner** designed to detect security issues in container images, file systems, and Git repositories. It scans for known vulnerabilities in dependencies and also checks for misconfigurations, which can help prevent potential security risks before deployment. Trivy is lightweight and fast, making it popular for DevSecOps practices, and can easily integrate into CI/CD pipelines for automated security checks.

Installing SonarQube:

We are deploying sonarQube as a container
Here is the command we will be using

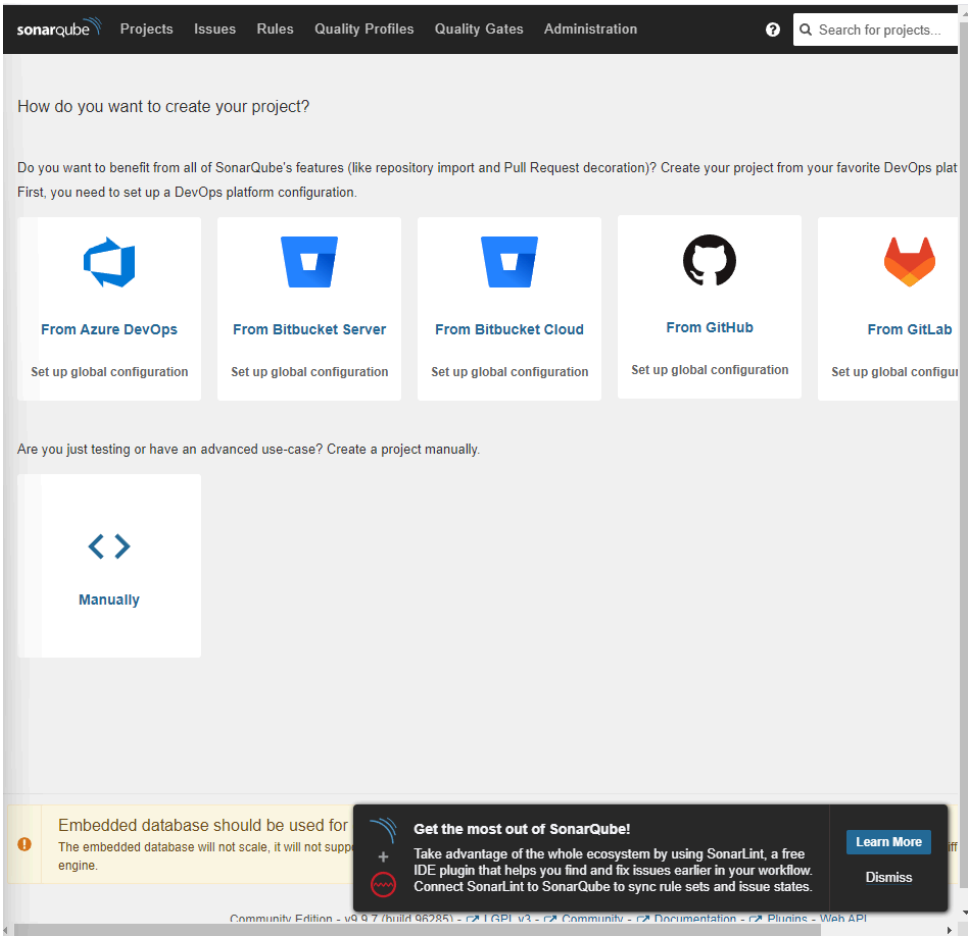
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community

Always make sure that you have the access to these ports with security groups
In case your EC2 instance is shut donw you can always re-start it with this command :
docker start sonar
we can confirm that the containers are running with docker ps

Accessing SonarQube:

we can access the sonarQube UI with ip address and the port
it should look like this : http://34.192.55.188:9000/

NOTE:
The default username and password are admin & admin
you need to change them before logging in
the UI will look like this



Installing Trivy :

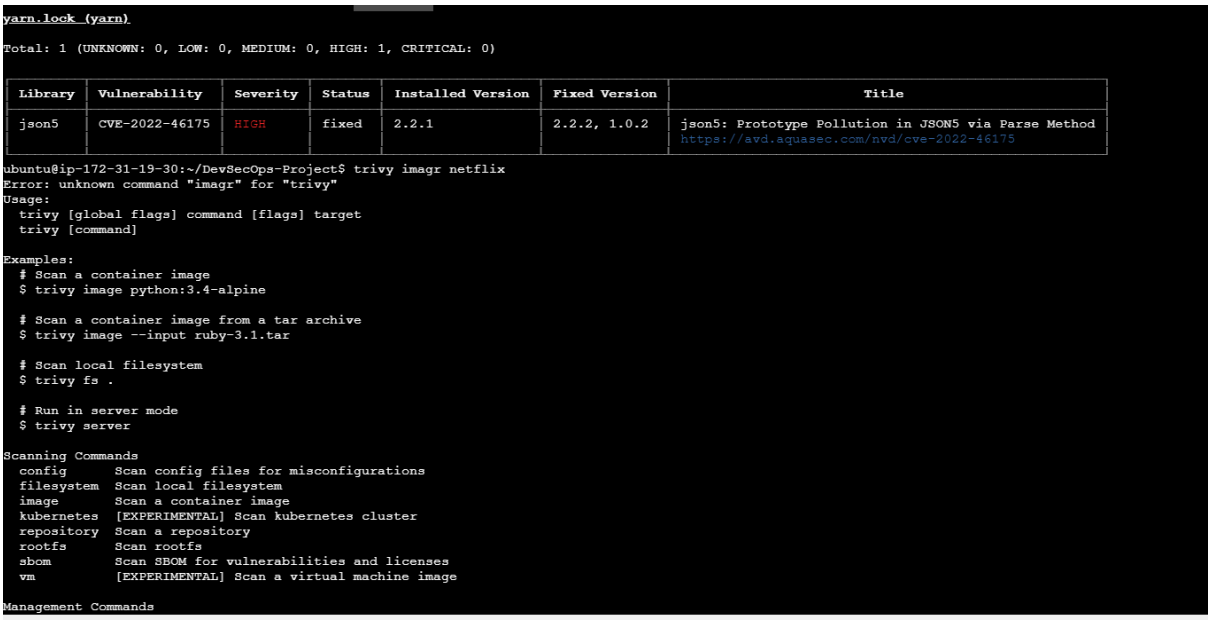
To install Trivy:
you can run this command 👍

sudo apt-get install wget apt-transport-https gnupg lsb-release
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -
echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list
sudo apt-get update
sudo apt-get install trivy

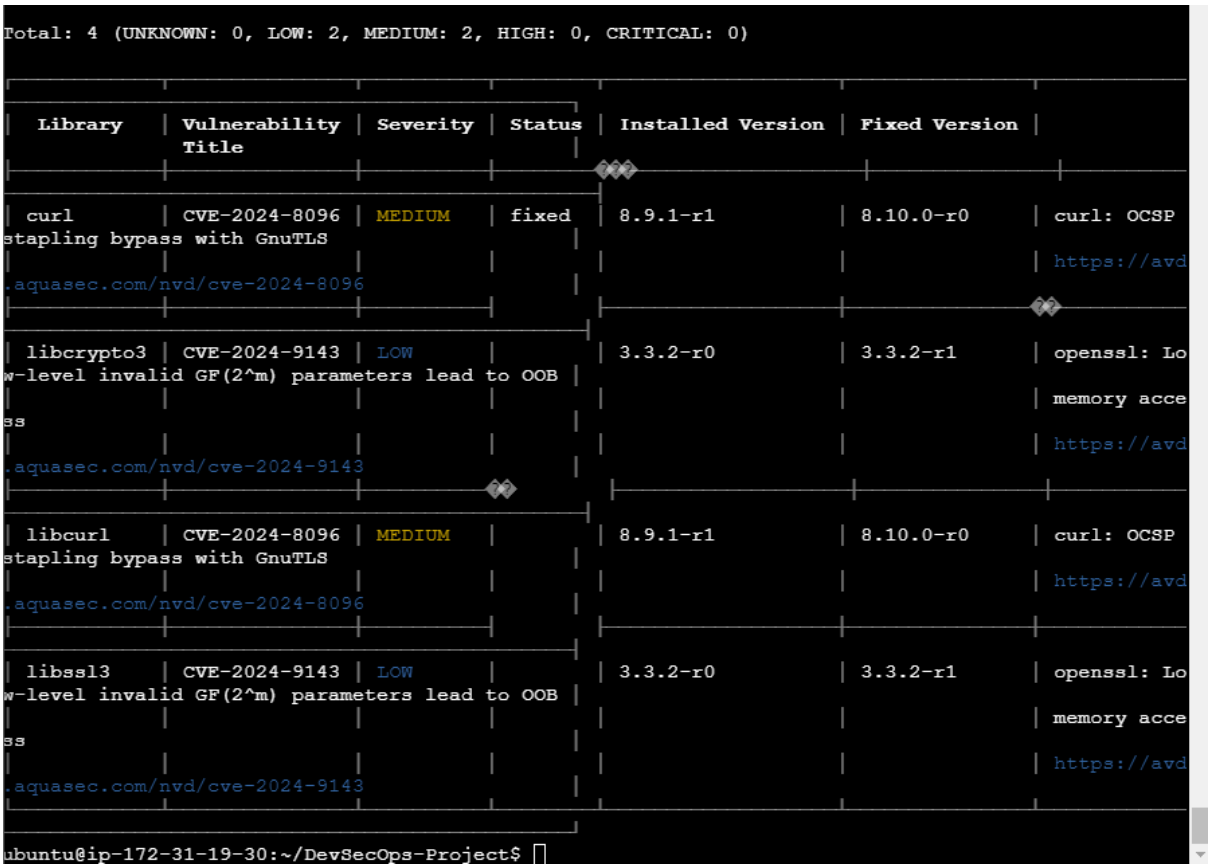
Now we can use trivy to scan ou folder using :

trivy fs .

Scanning with Trivy :



or we can use it to scan the image
trivy image <image-id>



CI/CD Setup:

Moving now to the Ops phase :

CI/CD Setup

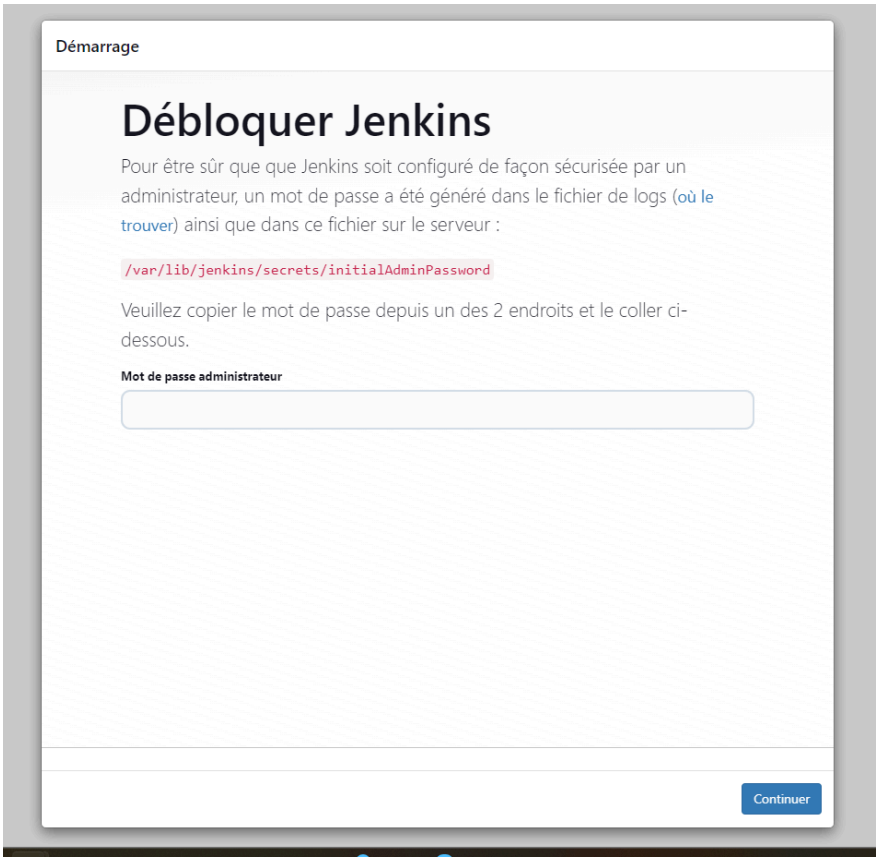
We will be using **Jenkins** for this section , and in order to install it you need to have java installed as well .

Here is the command that we will be using :

```
sudo apt update
sudo apt install fontconfig openjdk-17-jre
java -version
openjdk version "17.0.8" 2023-07-18
OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)

#jenkins
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

Accessing Jenkins:

We can now access the jenkins UI :



NOTE:
The password is stored in that file on our server , so we can use this command to access this file : sudo cat/var/lib/jenkins/secrets/initialAdminPassworddd

and use the password in the given file to access jenkins

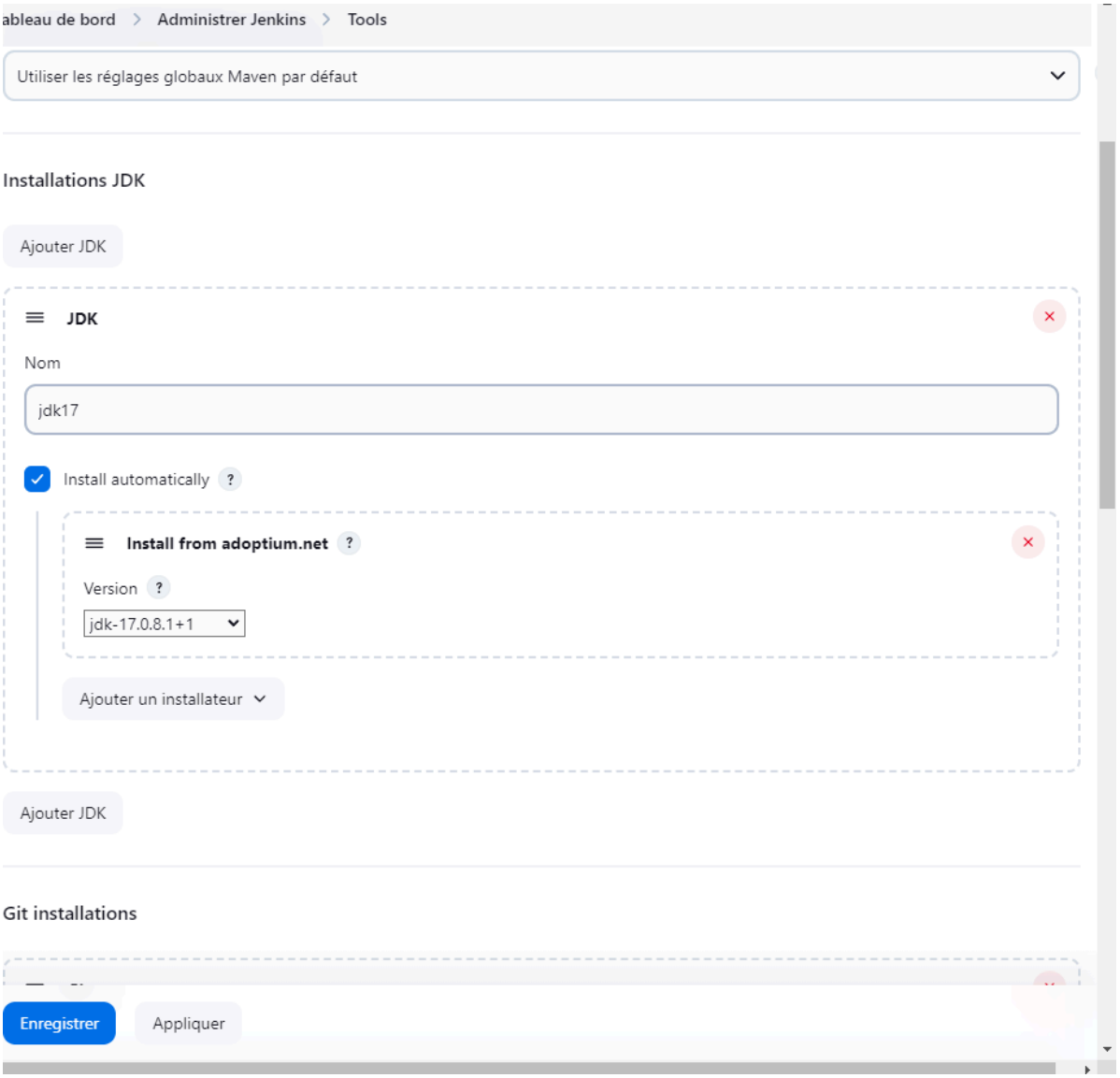
Installing plugins on Jenkins:

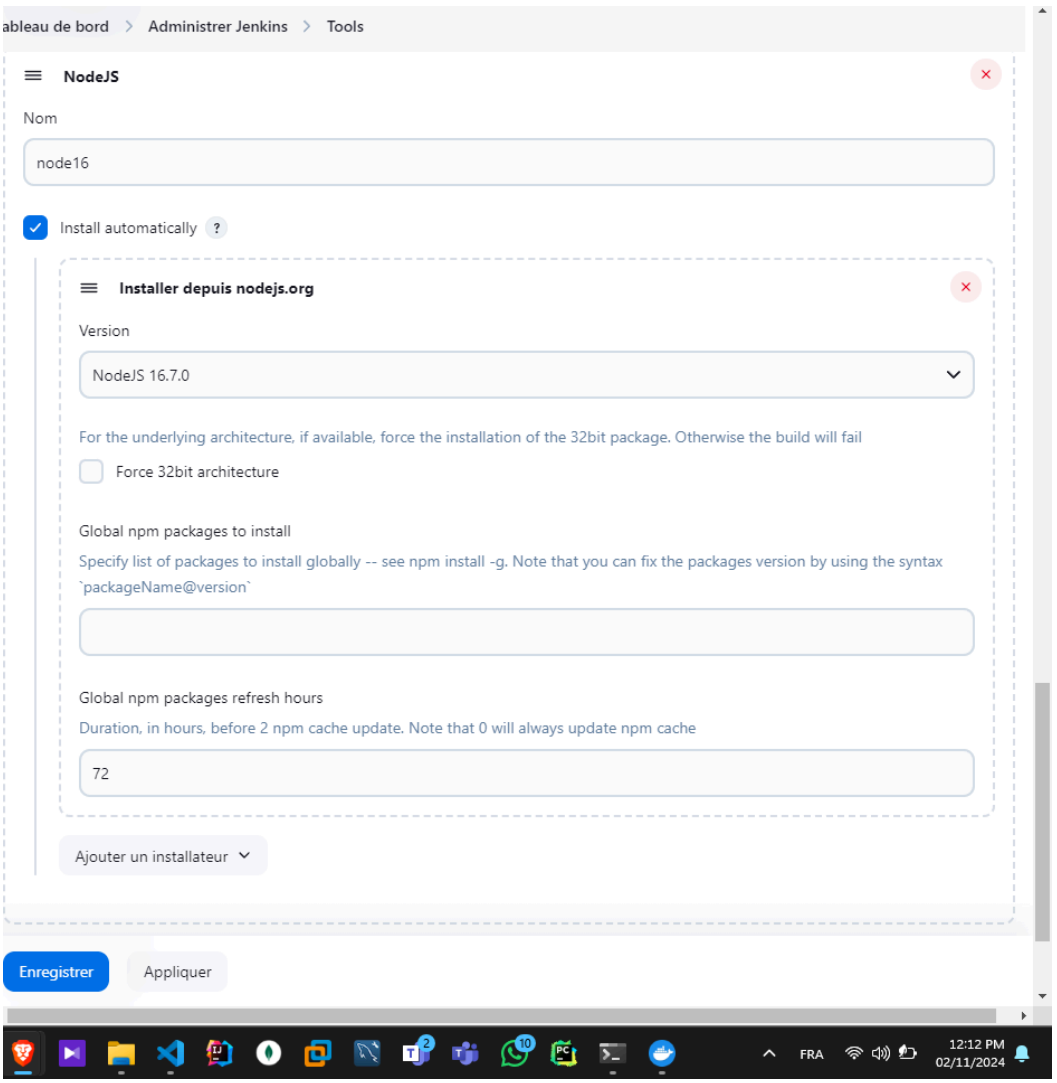
Install Necessary Plugins in Jenkins:
Goto Manage Jenkins →Plugins → Available Plugins →
Install below plugins
1 Eclipse Temurin Installer (Install without restart)
2 SonarQube Scanner (Install without restart)
3 NodeJs Plugin (Install Without restart)
4 Email Extension Plugin

for the tools make sure to name them as the same name in the script

Configure Java and Nodejs in Global Tool Configuration

Goto Manage Jenkins → Tools → Install JDK(17) and NodeJs(16)→ Click on Apply and Save

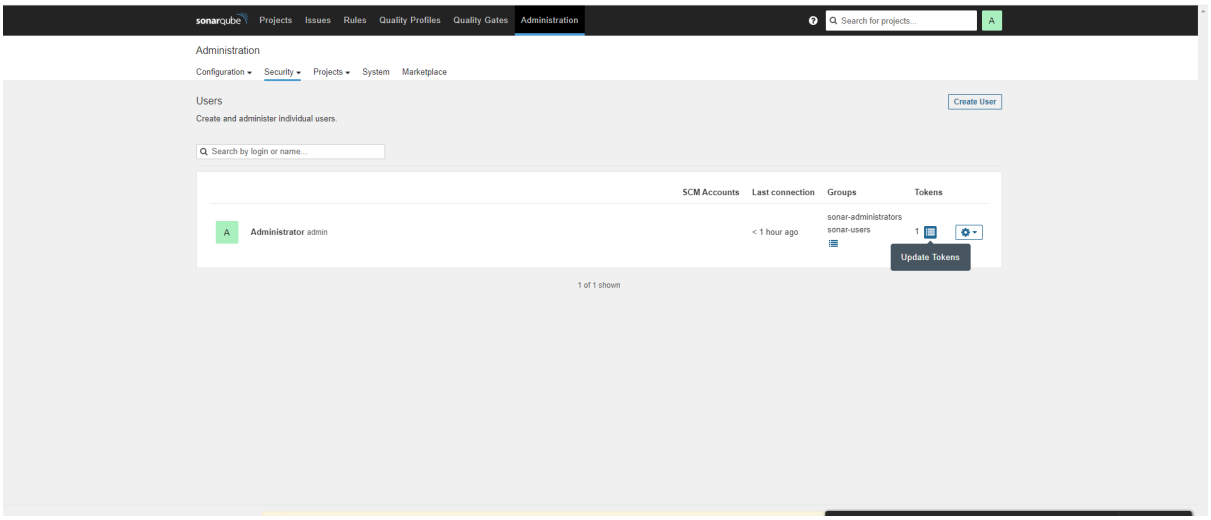




SonarQube:

Creating and saving the token:

Create the token
GO to SonarQube
Administration->Security -> Users->Update tokes -> Naming and generating the token



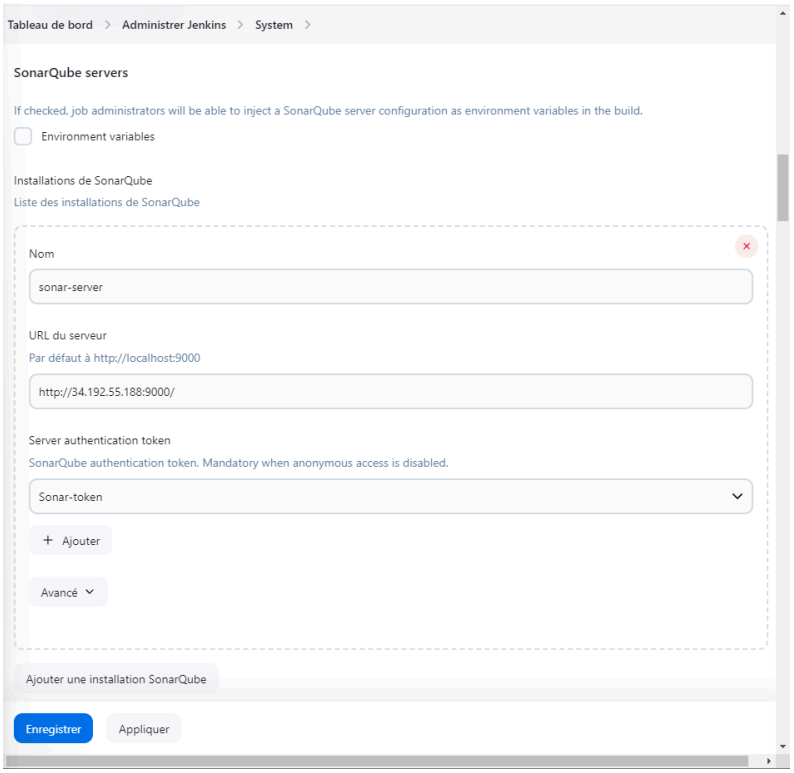
Goto Jenkins Dashboard → Manage Jenkins → Credentials → Add Secret Text. It should look like this

After adding sonar token

Click on Apply and Save

The Configure System option is used in Jenkins to configure different server

Global Tool Configuration is used to configure different tools that we install using Plugins



Adding the sonar scanner tool :

We will install a sonar scanner in the tools

Installations SonarScanner for MSBuild

Installations SonarScanner for MSBuild Edited

Installations SonarQube Scanner

Ajouter SonarQube Scanner

SonarQube Scanner

Name

sonar-scanner

Install automatically ?

Installer depuis Maven Central

Version

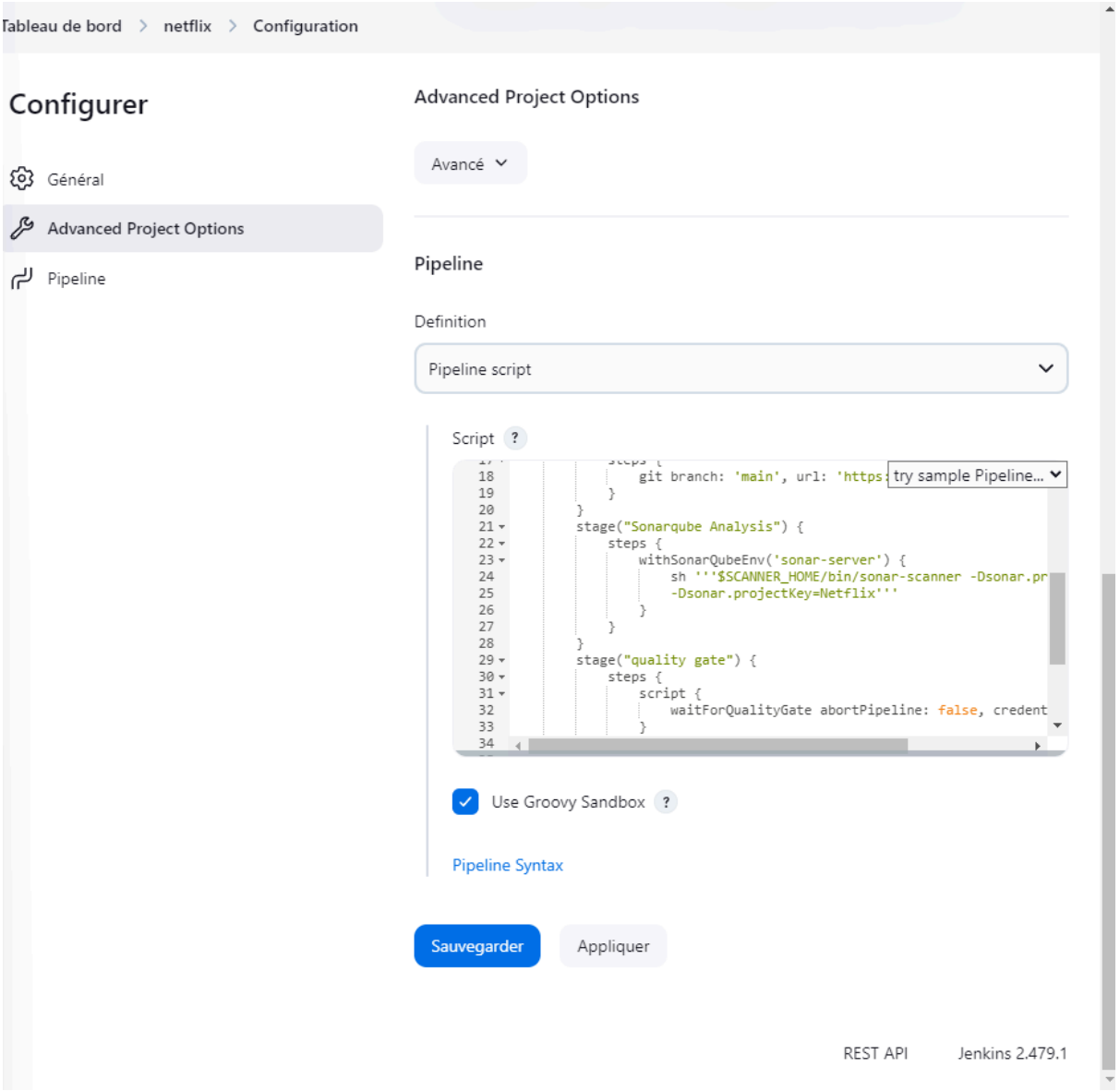
SonarQube Scanner 5.0.1.3006

Ajouter un installateur

Ajouter SonarQube Scanner

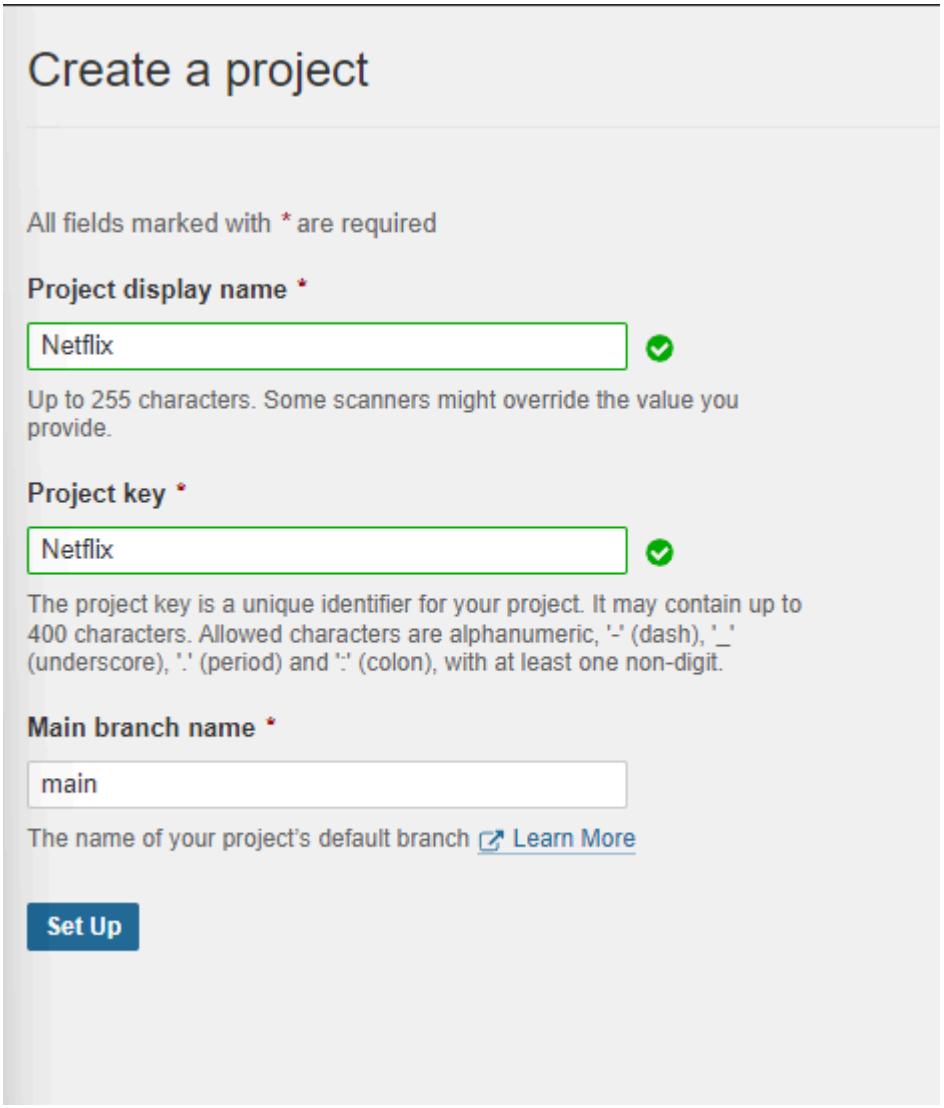
Creating the pipeline:

Create a CI/CD pipeline in Jenkins to automate your application deployment.

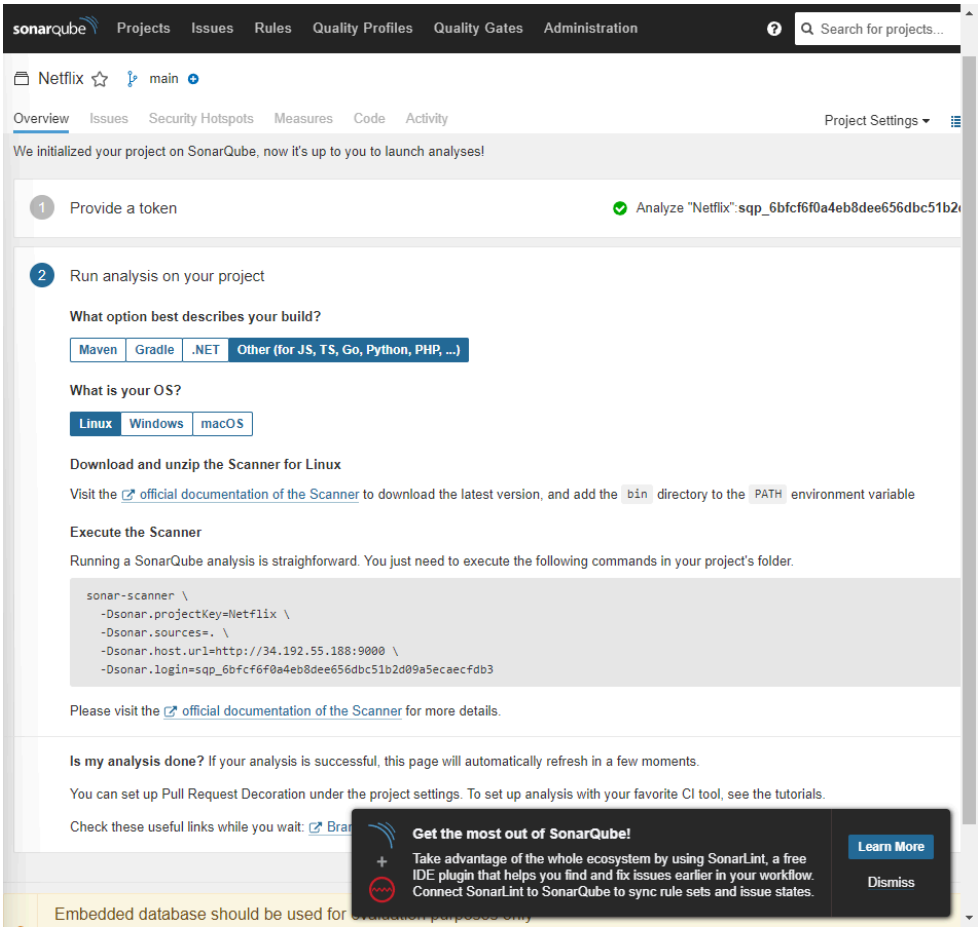


Adding a project on SonarQube:

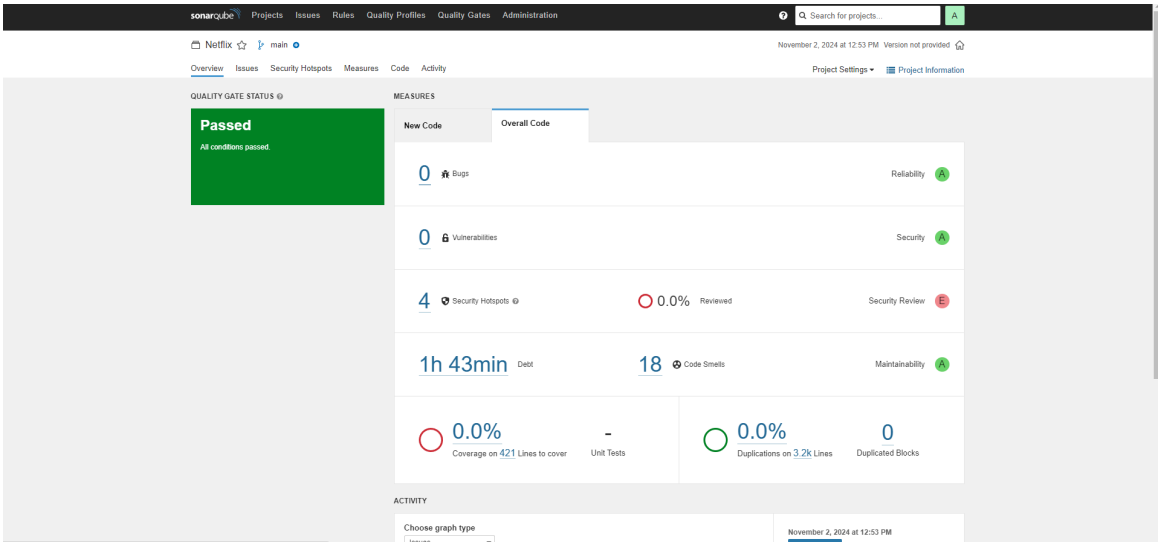
In the sonarQube analysis we can see that there is an existing project named Netflix
So let s make that happen
Go to sonarQube -> Projects -> Mnually



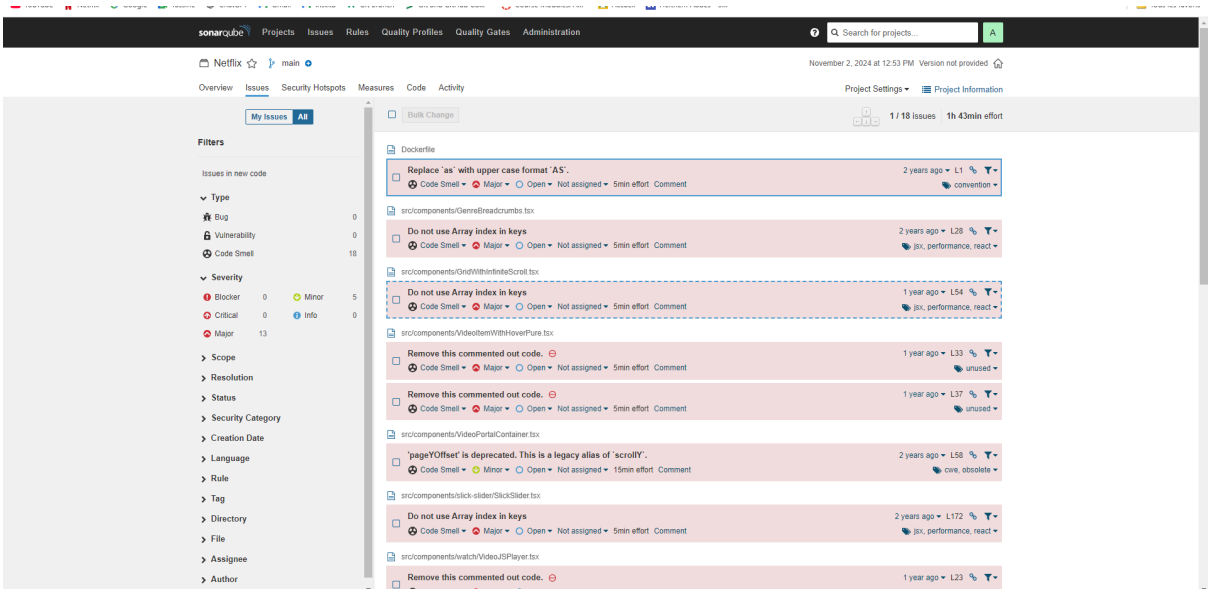
go to locally
This is what you should get after choosing the OS and the code , we will get the commands that we will be using



The page will change the build is correct



Here we can see all the informations about the issues



Quality Gate:

In SonarQube, a **Quality Gate** is a set of conditions that a project must meet to pass the quality check during analysis. It serves as a threshold for determining whether the code is considered "healthy" based on specified metrics. The Quality Gate can include a variety of criteria, such as:

Common Criteria for Quality Gates

1. **Coverage**: The percentage of code that is covered by unit tests.
2. **Bugs**: The number of bugs detected in the code.
3. **Vulnerabilities**: The number of security vulnerabilities identified.
4. **Code Smells**: The number of maintainability issues found in the code.
5. **Duplicated Lines**: The percentage of duplicated code in the project.

How Quality Gates Work

1. **Configuration**: Quality Gates are configured in the SonarQube interface. You can create a new Quality Gate or modify existing ones to suit the requirements of your project.
2. **Analysis**: When a project is analyzed, SonarQube checks the code against the defined Quality Gate conditions.
3. **Result**: After the analysis, SonarQube assigns a status (e.g., **Passed** or **Failed**) to the Quality Gate based on whether the project meets the specified criteria.

Install Dependency-Check and Docker Tools in Jenkins

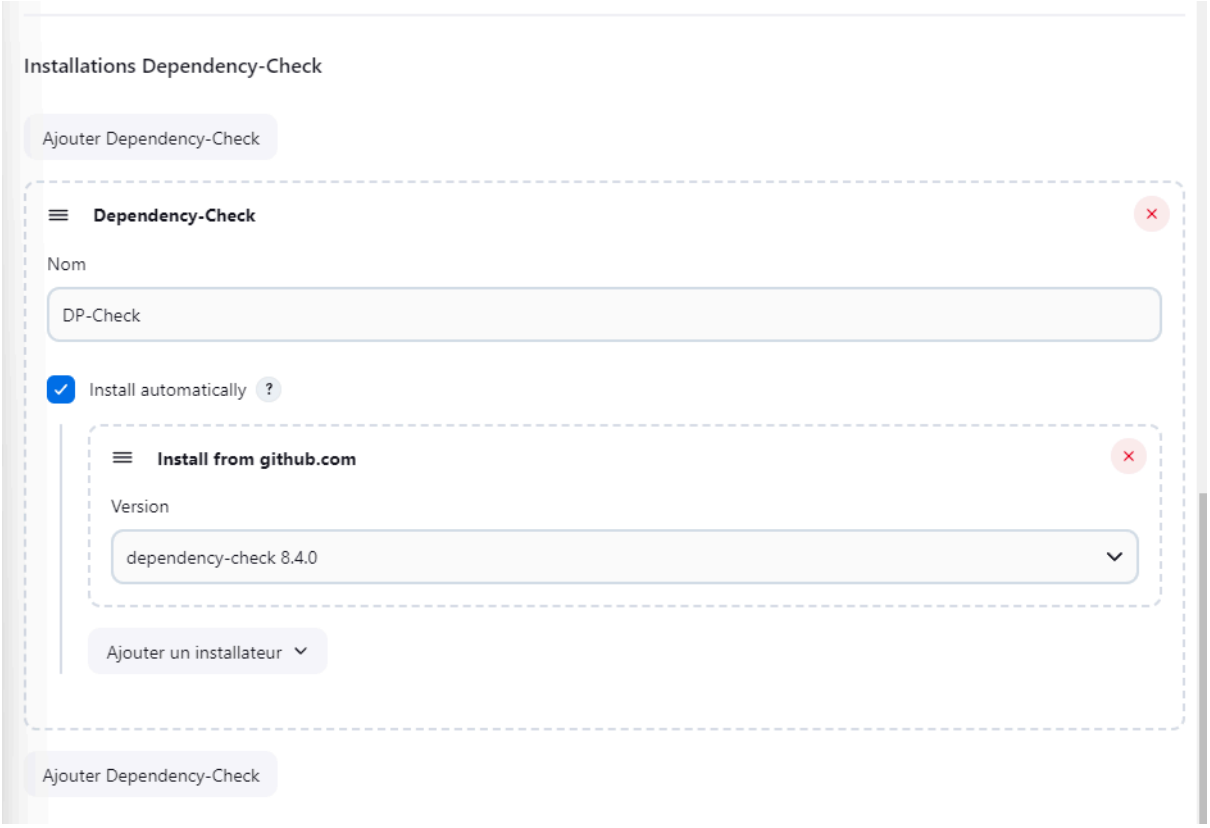
Install Dependency-Check and Docker Tools in Jenkins

Install Dependency-Check Plugin:

- Go to "Dashboard" in your Jenkins web interface.
- Navigate to "Manage Jenkins" → "Manage Plugins."
- Click on the "Available" tab and search for "OWASP Dependency-Check."
- Check the checkbox for "OWASP Dependency-Check" and click on the "Install without restart" button.

Configure Dependency-Check Tool:

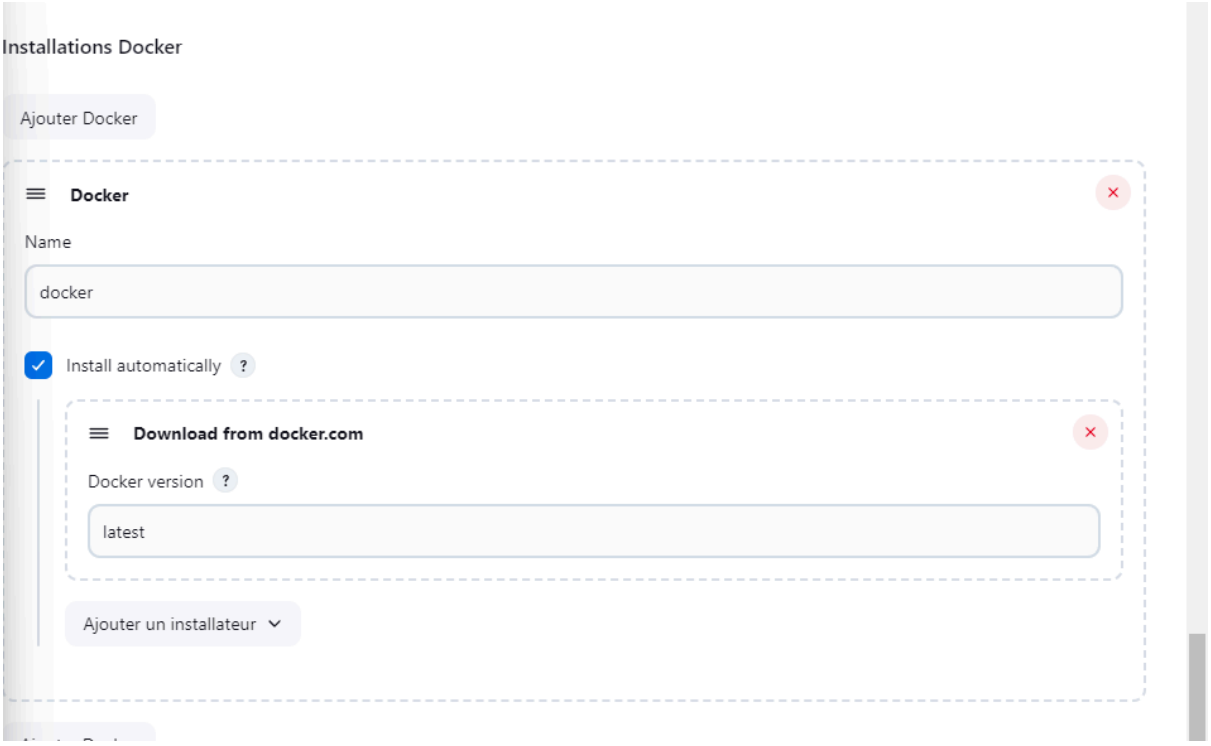
- After installing the Dependency-Check plugin, you need to configure the tool.
- Go to "Dashboard" → "Manage Jenkins" → "Global Tool Configuration."
- Find the section for "OWASP Dependency-Check."
- Add the tool's name, e.g., "DP-Check."
- Save your settings.



Install Docker Tools and Docker Plugins:

- Go to "Dashboard" in your Jenkins web interface.
- Navigate to "Manage Jenkins" → "Manage Plugins."
- Click on the "Available" tab and search for "Docker."
- Check the following Docker-related plugins:
  - Docker
  - Docker Commons
  - Docker Pipeline
  - Docker API
  - docker-build-step

- Click on the "Install without restart" button to install these plugins.



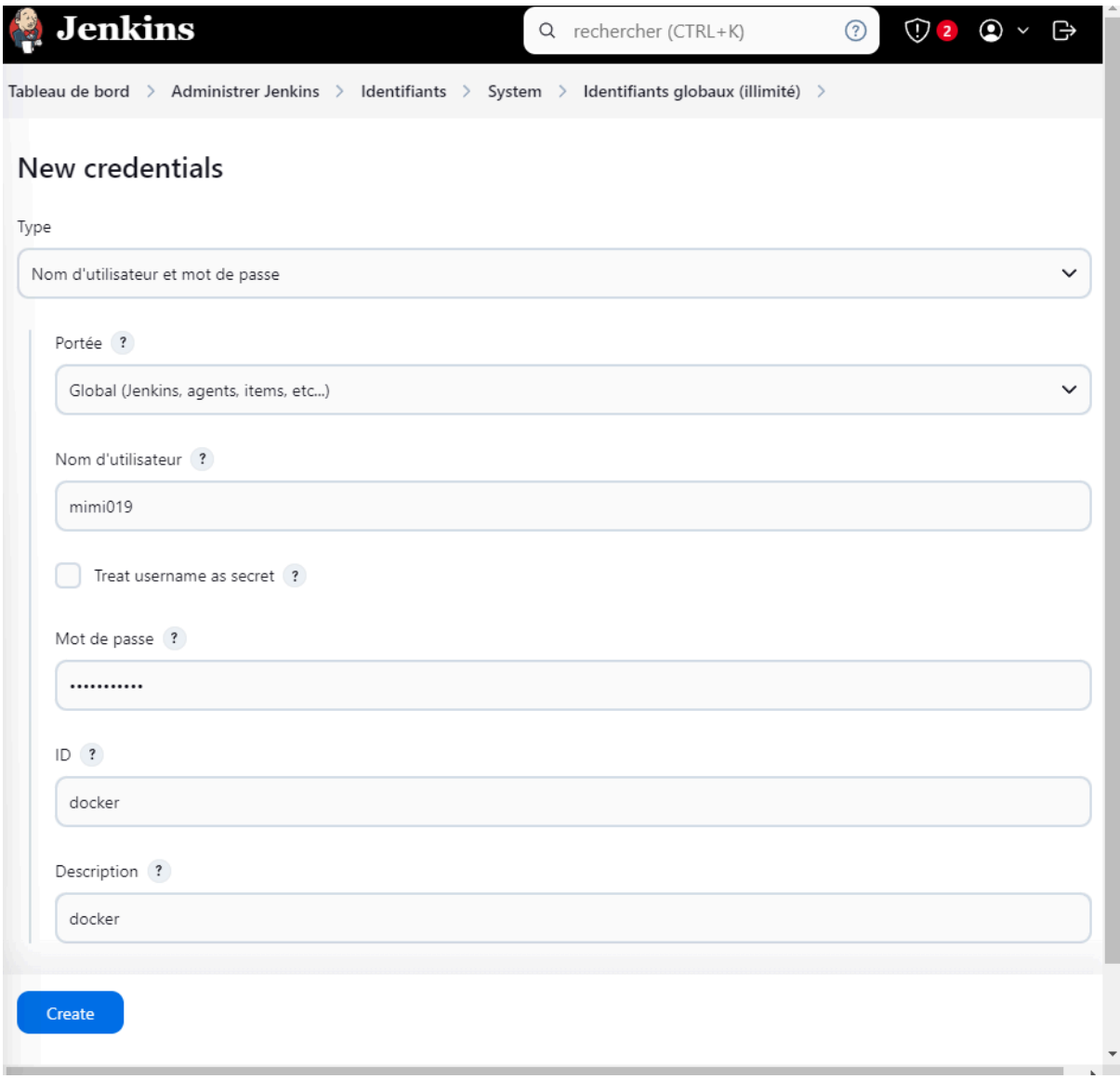
Adding credentials:

Add DockerHub Credentials:

- To securely handle DockerHub credentials in your Jenkins pipeline, follow these steps:
    - Go to "Dashboard" → "Manage Jenkins" → "Manage Credentials."
    - Click on "System" and then "Global credentials (unrestricted)."
    - Click on "Add Credentials" on the left side.
    - Choose "Secret text" as the kind of credentials.
    - Enter your DockerHub credentials (Username and Password) and give the credentials an ID (e.g., "docker").
    - Click "OK" to save your DockerHub credentials.
- Run these commands on your EC2 instance so that the docker stage work :

  `sudo su`

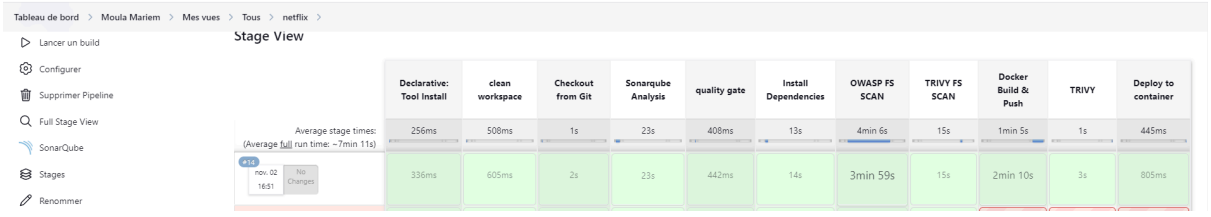
  `sudo usermod -aG docker jenkins`
  `sudo systemctl restart jenkins`



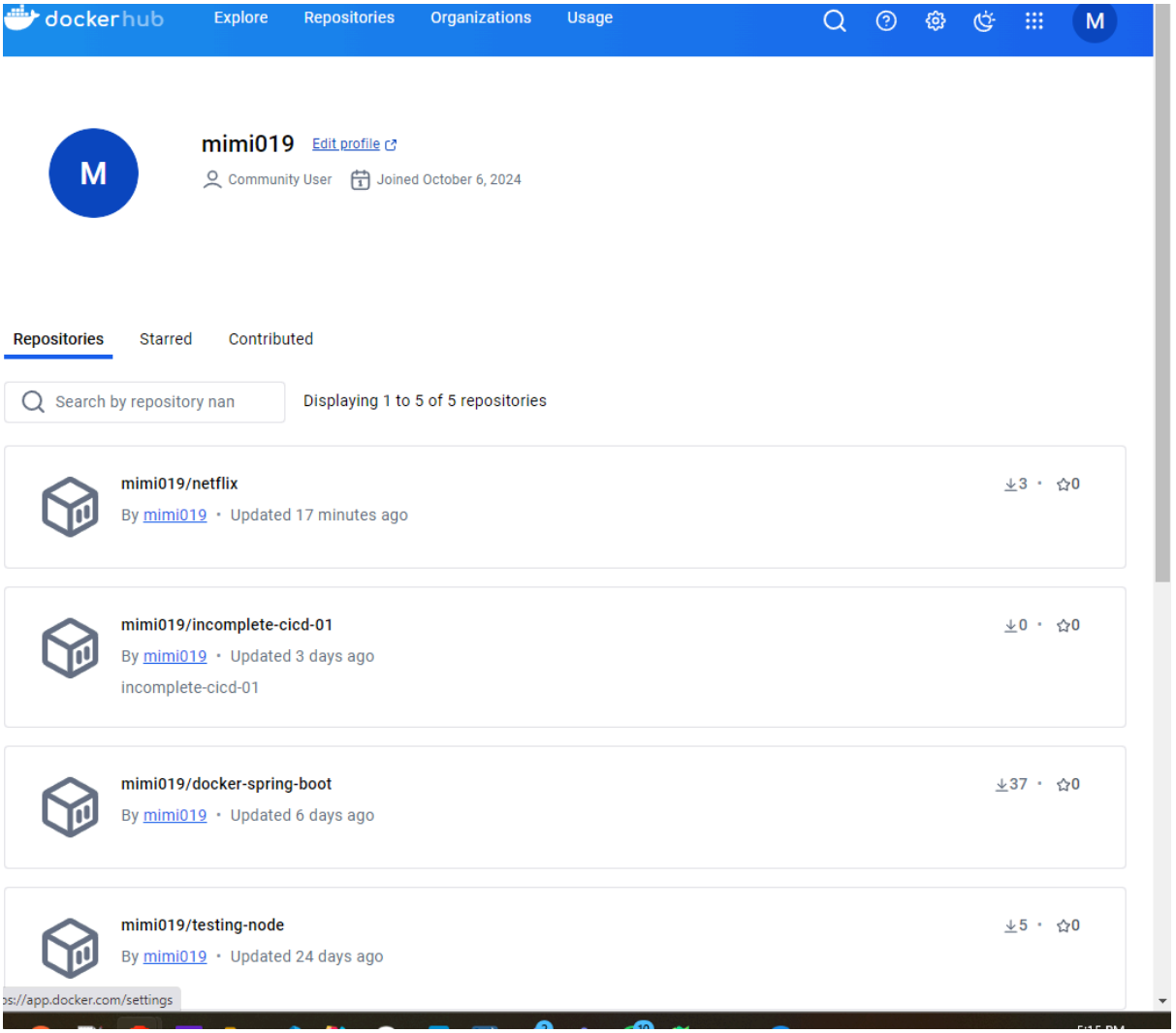
Now, you have installed the Dependency-Check plugin, configured the tool, and added Docker-related plugins along with your DockerHub credentials in Jenkins. You can now proceed with configuring your Jenkins pipeline to include these tools and credentials in your CI/CD process.

Building :

Since in the script we re in there building and running the container then i will stop the container and then remove it
Now that the pipeline turned green ,



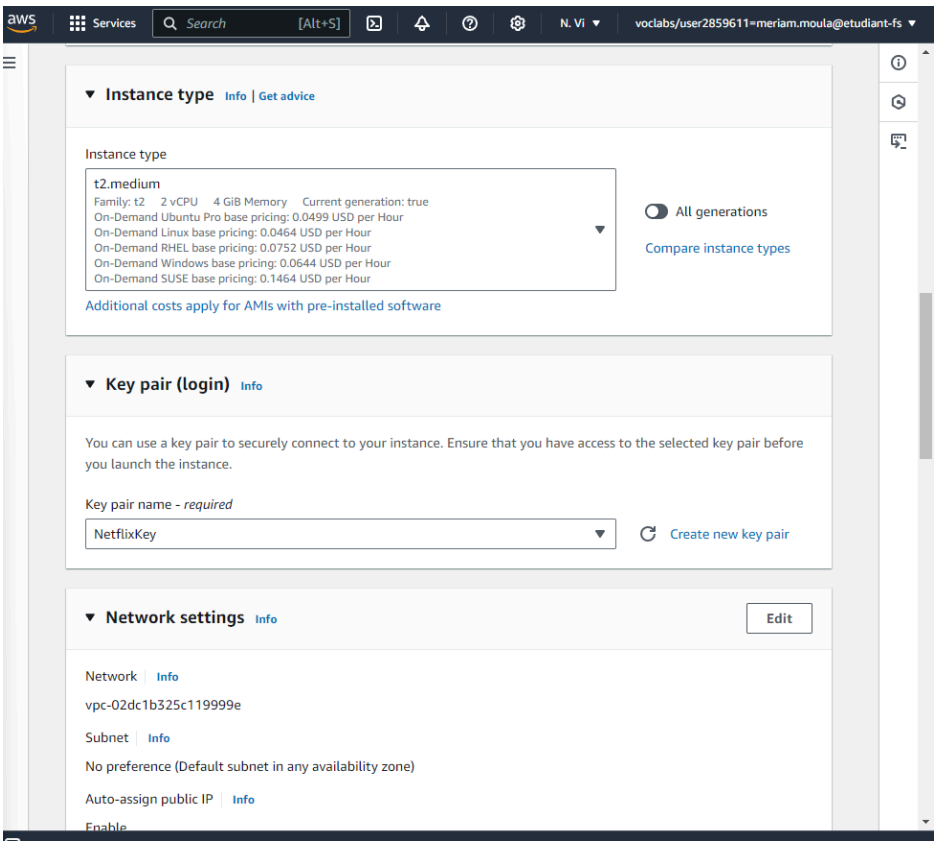
we can see that we have an update on our docker hub repo

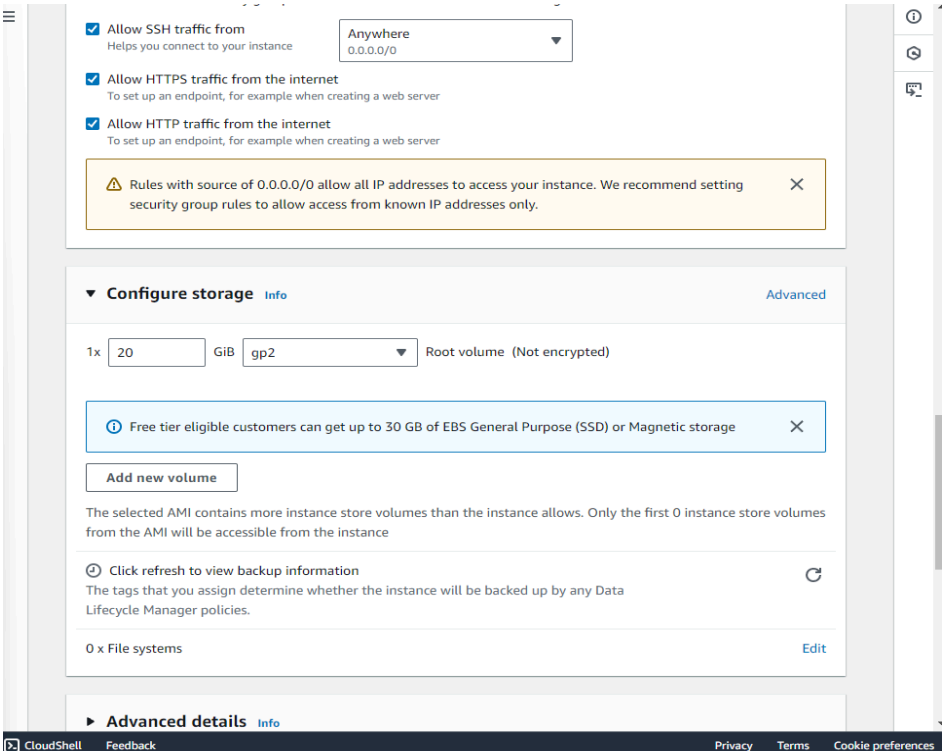


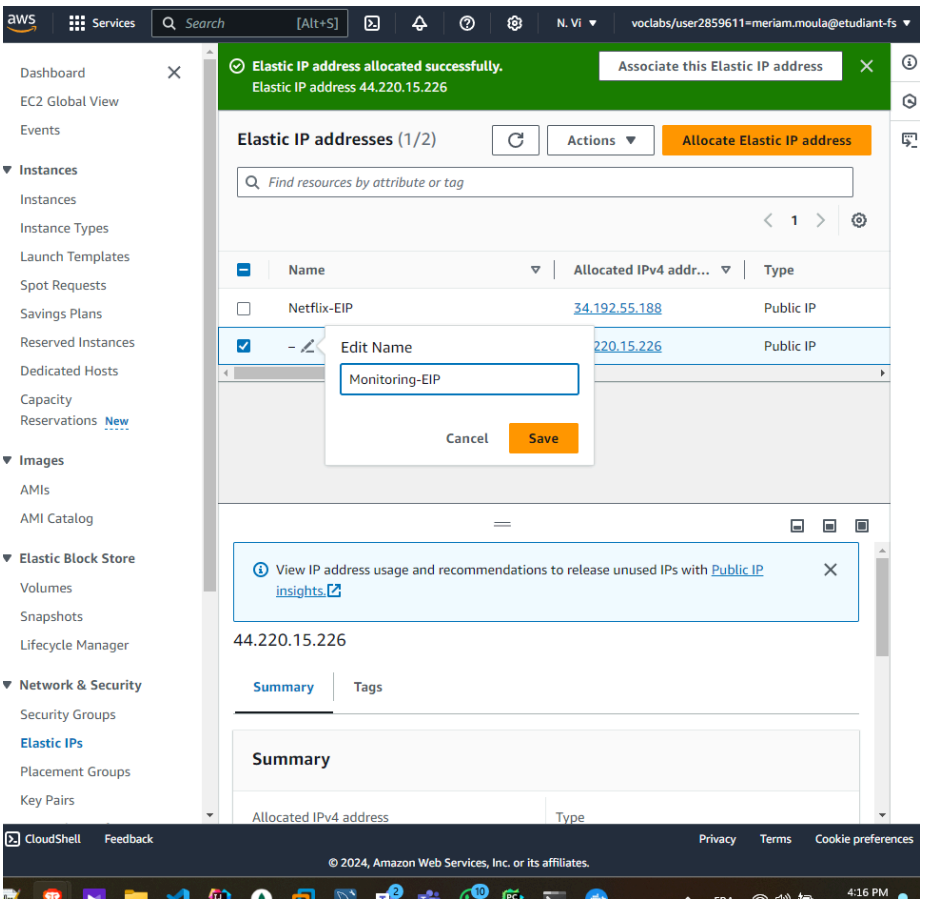
Monitoring

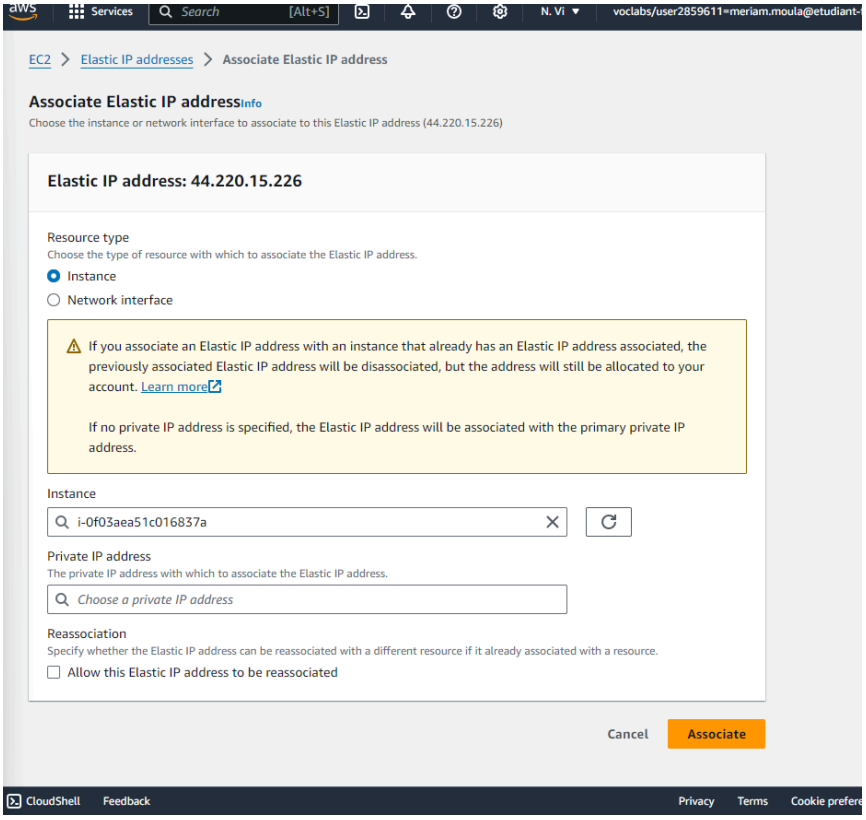
Creating instances for monitoring :

We are having a separate server because the current one is not enough
We are going with t2.medium because prometheus require a lot of storage and ram


aws
Services
Search
[Alt+S]
N. Vi
voclabs/user2859611=meriam.moula@etudiant-fs
Instance type Info | Get advice
Instance type
t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
NetflixKey
Create new key pair
Network settings Info
Edit
Network Info
vpc-02dc1b325c119999e
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable


Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Configure storage Info
Advanced
1x
20
GiB
gp2
Root volume (Not encrypted)
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage
Add new volume
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance
Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.
0 x File systems
Edit
Advanced details Info
CloudShell
Feedback
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
N. Vi
voclabs/user2859611=meriam.moula@etudiant-fs
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Elastic IP address allocated successfully.
Elastic IP address 44.220.15.226
Associate this Elastic IP address
Elastic IP addresses (1/2)
Actions
Allocate Elastic IP address
Find resources by attribute or tag
Name
Allocated IPv4 addr...
Type
Netflix-EIP
34.192.55.188
Public IP
Edit Name
Monitoring-EIP
Cancel
Save
220.15.226
Public IP
View IP address usage and recommendations to release unused IPs with Public IP insights.
44.220.15.226
Summary
Tags
Summary
Allocated IPv4 address
Type
CloudShell
Feedback
Privacy
Terms
Cookie preferences
© 2024, Amazon Web Services, Inc. or its affiliates.
4:16 PM


Services
Search
[Alt+S]
N. Vi
voclabs/user2859611=meriam.moula@etudiant-
EC2
Elastic IP addresses
Associate Elastic IP address
Associate Elastic IP address Info
Choose the instance or network interface to associate to this Elastic IP address (44.220.15.226)
Elastic IP address: 44.220.15.226
Resource type
Choose the type of resource with which to associate the Elastic IP address.
Instance
Network interface
If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. Learn more
If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.
Instance
i-0f03aea51c016837a
Private IP address
The private IP address with which to associate the Elastic IP address.
Choose a private IP address
Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
Allow this Elastic IP address to be reassociated
Cancel
Associate
CloudShell
Feedback
Privacy
Terms
Cookie prefere

Installing Prometheus and Grafana:

Set up Prometheus and Grafana to monitor your application.

Installing Prometheus:

First, create a dedicated Linux user for Prometheus and download Prometheus:

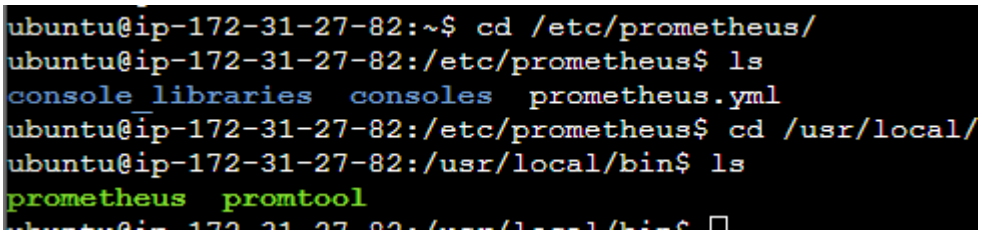
sudo useradd --system --no-create-home --shell /bin/false prometheus
wget
https://github.com/prometheus/prometheus/releases/download/v2.47.1/prometheus-2.47.1.linux-amd64.tar.gz

To unzip the files and remove all the unnecessary files
Extract Prometheus files, move them, and create directories:

tar -xvf prometheus-2.47.1.linux-amd64.tar.gz #Unzipping
cd prometheus-2.47.1.linux-amd64/ #Going inside the folder
sudo mkdir -p /data /etc/prometheus #Move all the necessary files
sudo mv prometheus promtool /usr/local/bin/
sudo mv consoles/ console_libraries/ /etc/prometheus/
sudo mv prometheus.yml /etc/prometheus/prometheus.yml

Set ownership for directories:
sudo chown -R prometheus:prometheus /etc/prometheus/ /data/

Now if you run this , it will change the dir : cd /etc/prometheus/



The `prometheus.yml` file is the main configuration file for Prometheus, an open-source monitoring and alerting toolkit. This file defines how Prometheus collects metrics, where to scrape them from, and various settings that control its behavior.
By configuring this file, you define how Prometheus will gather data from various services, making it a crucial component in your monitoring setup.
Node Exporter is a crucial component in the Prometheus ecosystem used for monitoring system metrics on a machine. It collects various hardware and OS metrics from the host machine and exposes them in a format that Prometheus can scrape

Create a systemd unit configuration file for Prometheus:

sudo nano /etc/systemd/system/prometheus.service

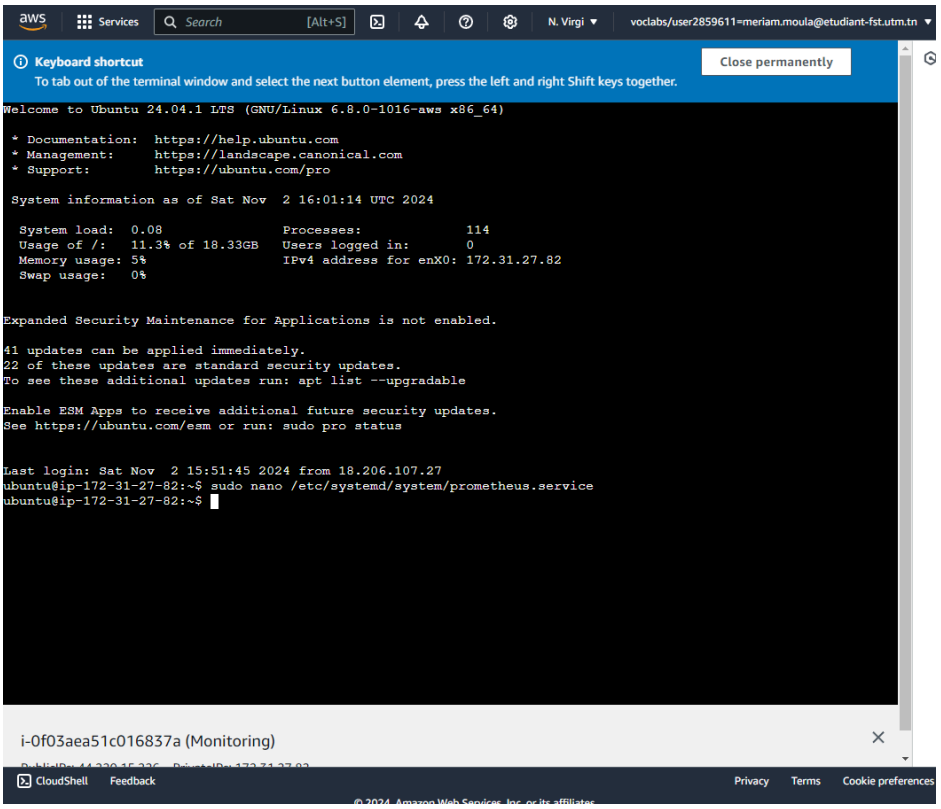
Updating the prometheus.service

Add the following content to the prometheus.service file:

```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

[Service]
User=prometheus
Group=prometheus
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/prometheus \
  --config.file=/etc/prometheus/prometheus.yml \
  --storage.tsdb.path=/data \
  --web.console.templates=/etc/prometheus/consoles \
  --web.console.libraries=/etc/prometheus/console_libraries \
  --web.listen-address=0.0.0.0:9090 \
  --web.enable-lifecycle

[Install]
```

WantedBy=multi-user.target

Note:
ctrl+o
ctrl +x



Here's a brief explanation of the key parts in this `prometheus.service` file:

- `User` and `Group` specify the Linux user and group under which Prometheus will run.
- `ExecStart` is where you specify the Prometheus binary path, the location of the configuration file (`prometheus.yml`), the storage directory, and other settings.
- `web.listen-address` configures Prometheus to listen on all network interfaces on port 9090.
- `web.enable-lifecycle` allows for management of Prometheus through API calls.
-

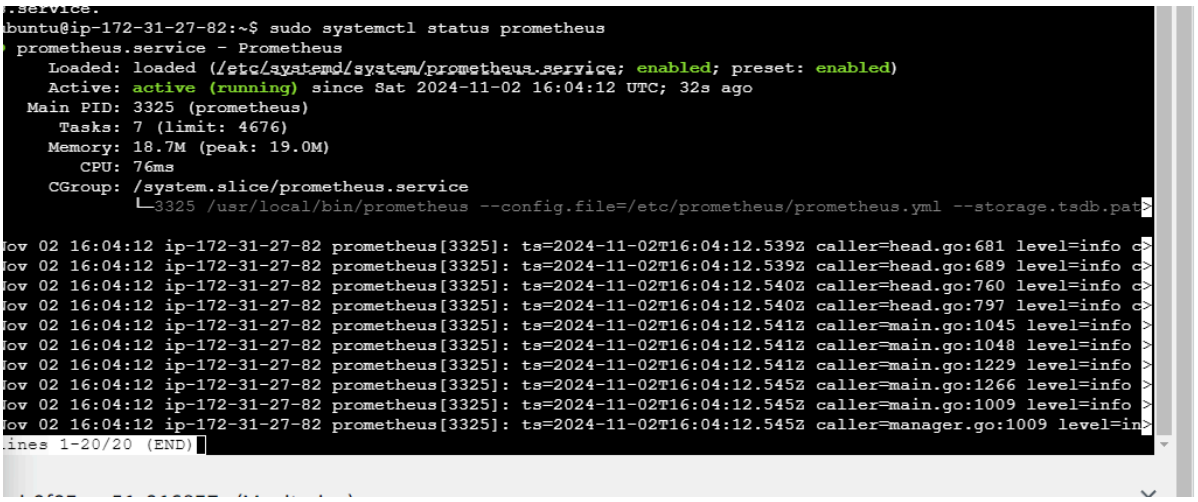
Enable and start Prometheus:

sudo systemctl enable prometheus
sudo systemctl start prometheus

```
ubuntu@ip-172-31-27-82:~$ sudo systemctl enable prometheus
sudo systemctl start prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheu
s.service.
ubuntu@ip-172-31-27-82:~$
```

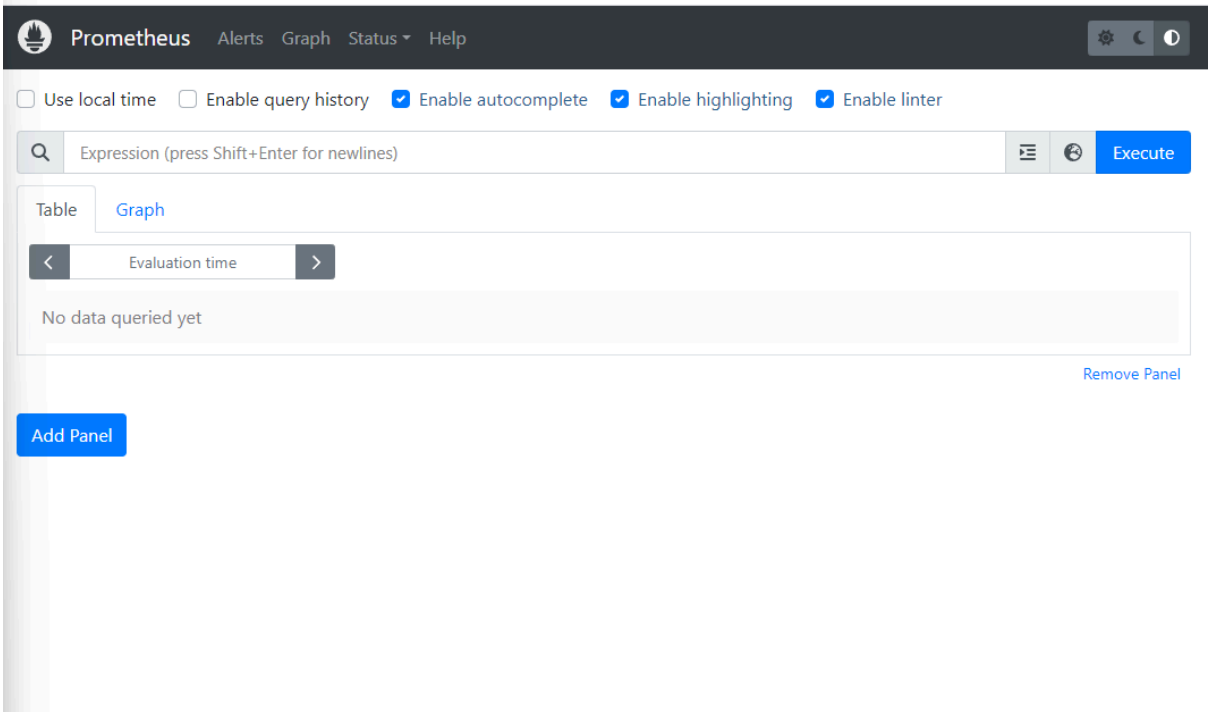
Verify Prometheus's status:

Run : sudo systemctl status prometheus
We can seee that the server is UP and running



we can access it on port 9090 , `http://<your-server-ip>:9090`

So make sure that your instance has access to the port 9090



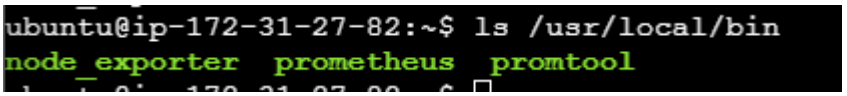
Installing Node Exporter:

Create a system user for Node Exporter and download Node Exporter:

sudo useradd --system --no-create-home --shell /bin/false node_exporter
wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz
we can see that there is a node file created

Extract Node Exporter files, move the binary, and clean up:

tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz
sudo mv node_exporter-1.6.1.linux-amd64/node_exporter /usr/local/bin/
rm -rf node_exporter*



Create a systemd unit configuration file for Node Exporter:

sudo nano /etc/systemd/system/node_exporter.service

Add the following content to the node_exporter.service file:

```
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

[Service]
User=node_exporter
Group=node_exporter
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/node_exporter --collector.logind

[Install]
WantedBy=multi-user.target
```

Note: Use ctrl+x to exit and save file

Enable and start Node Exporter:

```
sudo systemctl enable node_exporter
sudo systemctl start node_exporter
```

Verify the Node Exporter's status:

```
sudo systemctl status node_exporter
```

Configure Prometheus Plugin Integration:

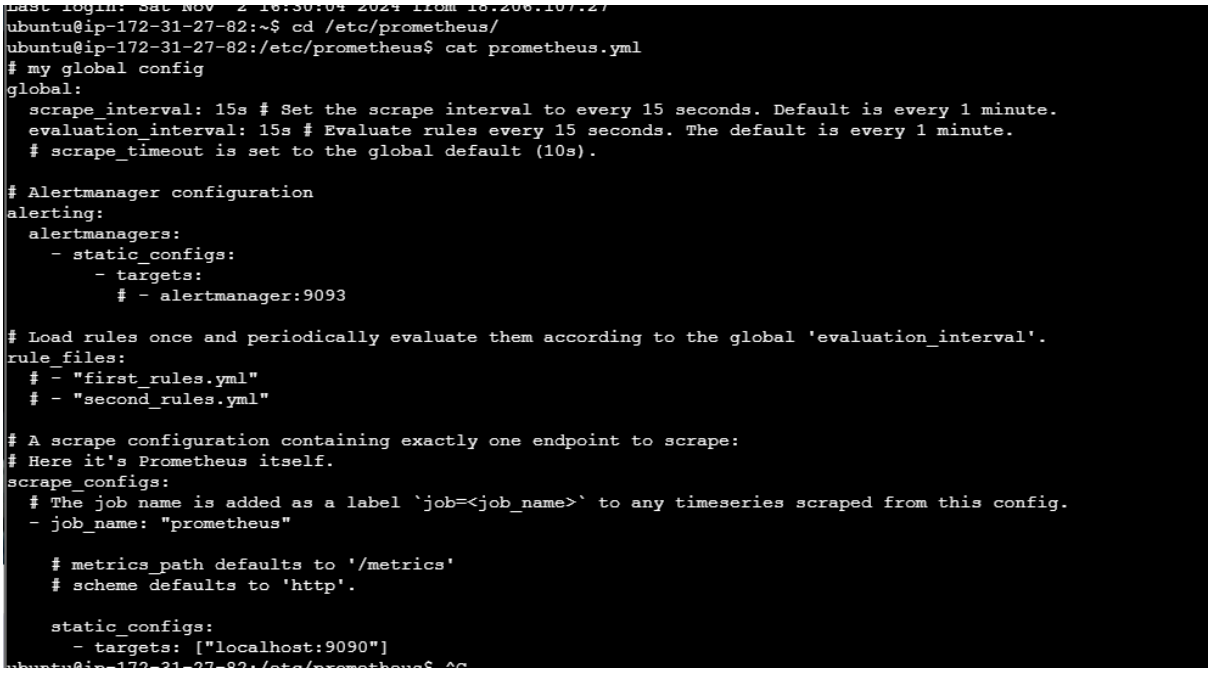
Integrate Jenkins with Prometheus to monitor the CI/CD pipeline.

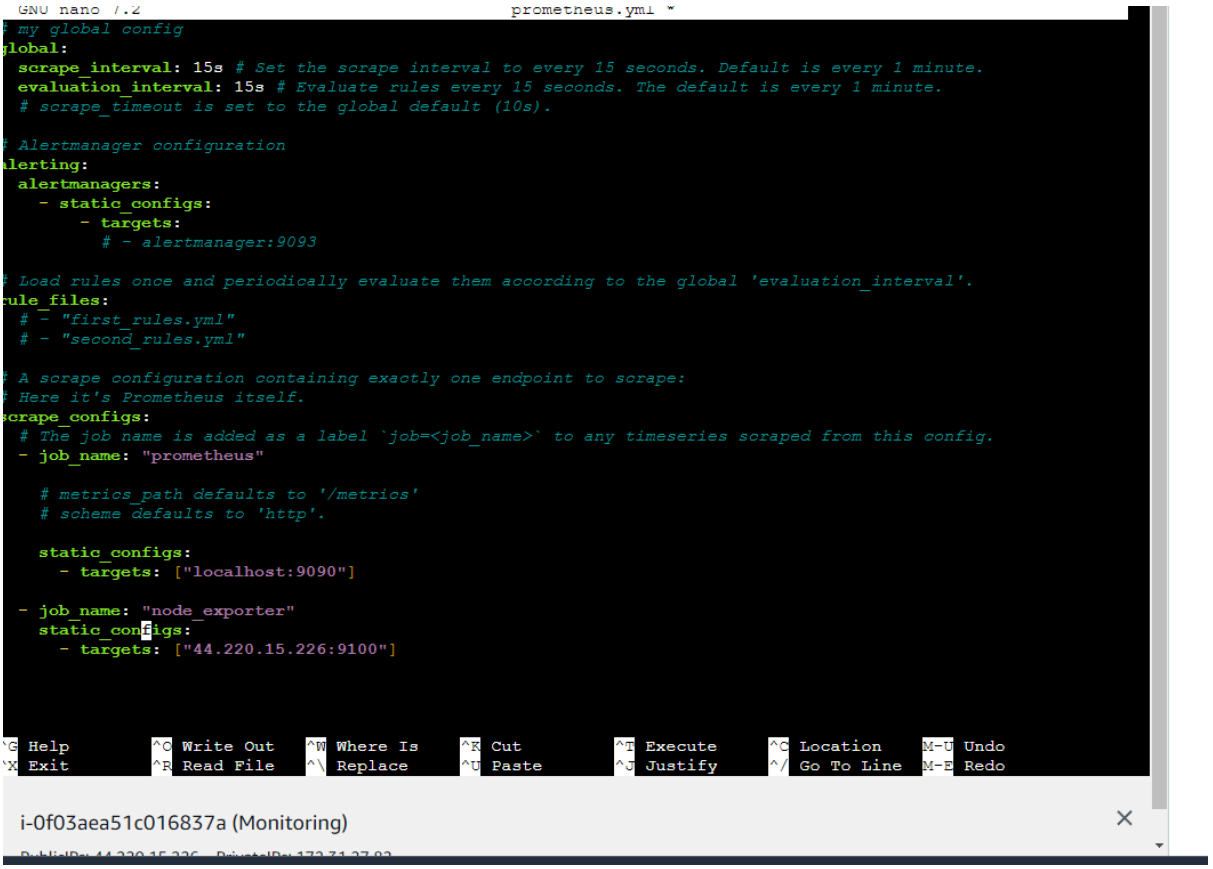
Prometheus Configuration:

To configure Prometheus to scrape metrics from Node Exporter and Jenkins, you need to modify the **`prometheus.yml`** file. Here is an example `prometheus.yml` configuration for your setup:

cd /etc/prometheus/
cat prometheus.yml

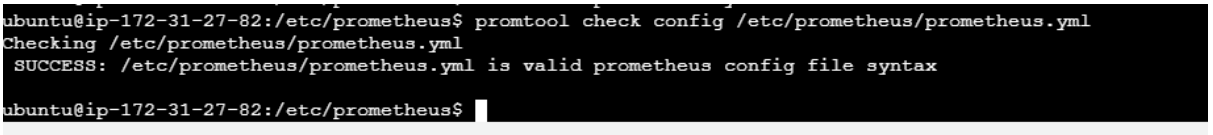
We will be modifying the file that we should always modify when we try to monitor using:
sudo nano prometheus.yml
we will add a job for our node exporter





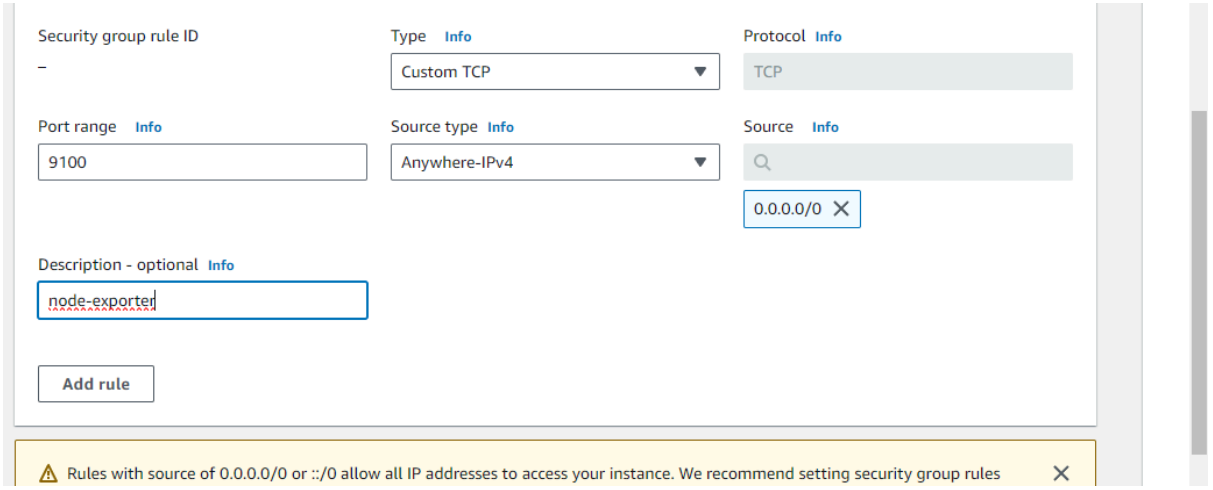
Check the validity of the configuration file:

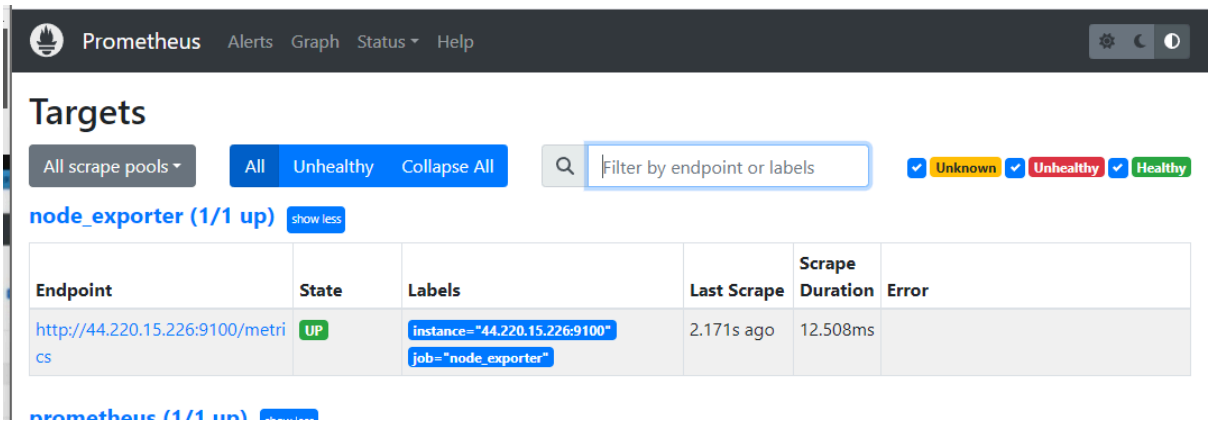
promtool check config /etc/prometheus/prometheus.yml



Reload the Prometheus configuration without restarting:

curl -X POST http://localhost:9090/-/reload
After reloading i should be able to see a target in my prometheus UI
Make sure to have port 9100 on our Security Group





Grafana :

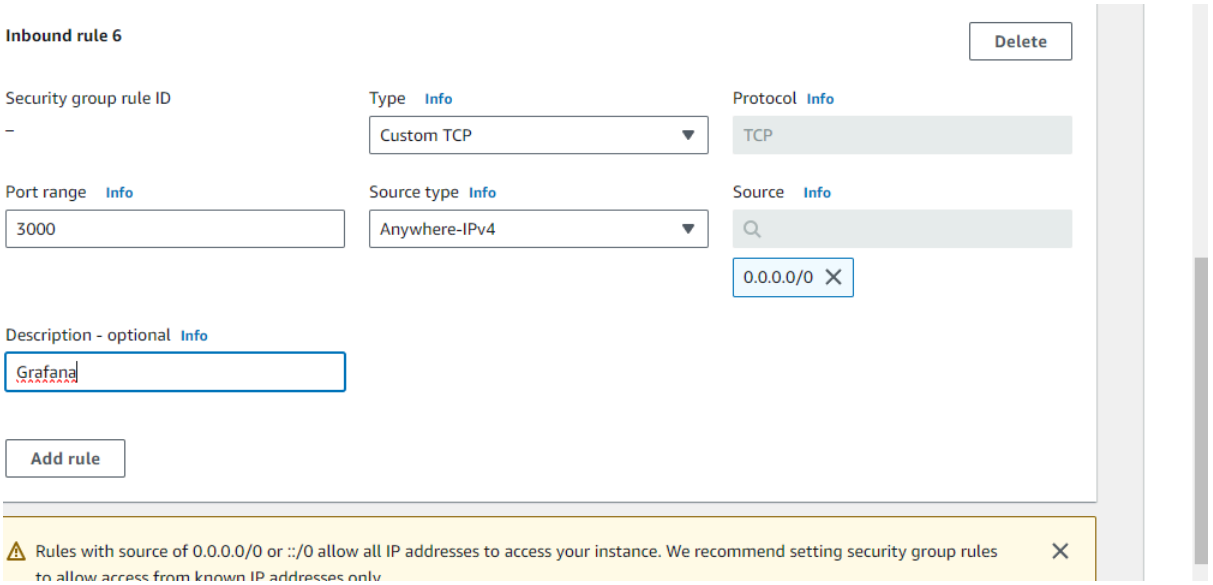
Install Grafana on Ubuntu 22.04 and Set it up to Work with Prometheus

Install Dependencies:

First, ensure that all necessary dependencies are installed:

sudo apt-get update
sudo apt-get install -y apt-transport-https software-properties-common

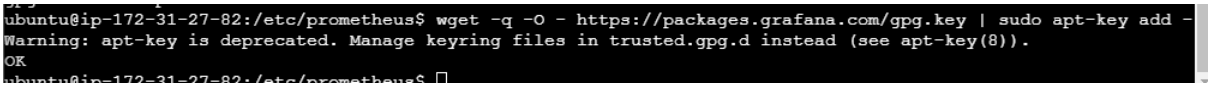
Update the security group for Grafana



Add the GPG Key:

Add the GPG key for Grafana:

wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -



Add Grafana Repository:

Add the repository for Grafana stable releases:
echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list

Update and Install Grafana:

Update the package list and install Grafana:
sudo apt-get update
sudo apt-get -y install grafana

Enable and Start Grafana Service:

To automatically start Grafana after a reboot, enable the service:

sudo systemctl enable grafana-server

Start Grafana:

sudo systemctl start grafana-server

Check Grafana Status:

Verify the status of the Grafana service to ensure it's running correctly:

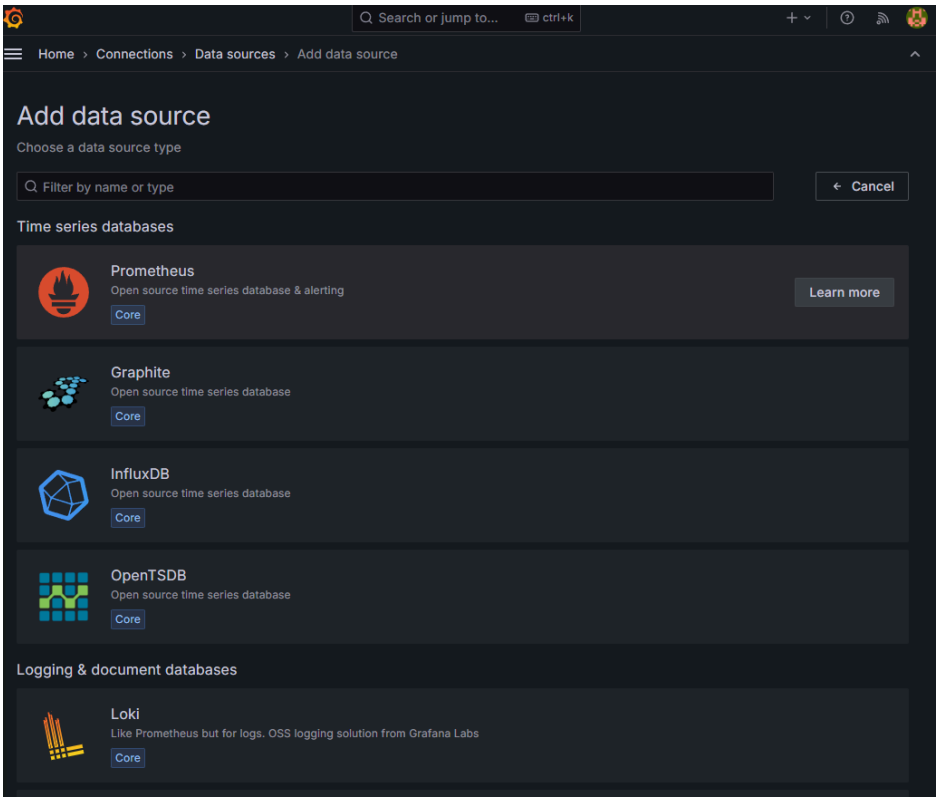
sudo systemctl status grafana-server

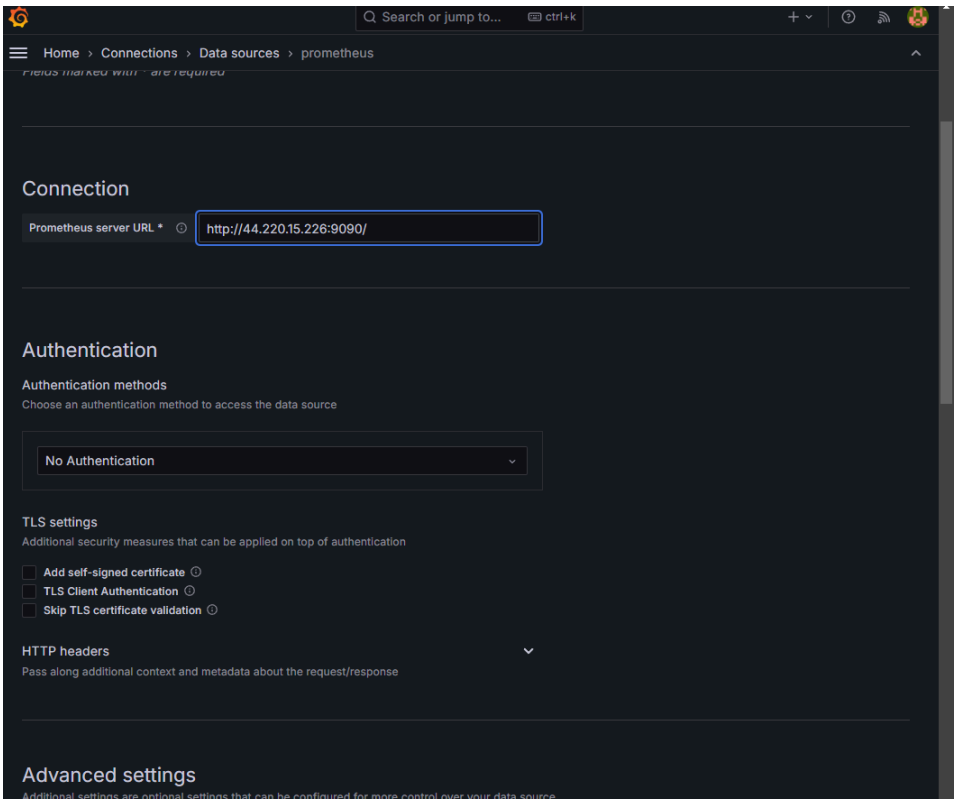
Access Grafana Web Interface:

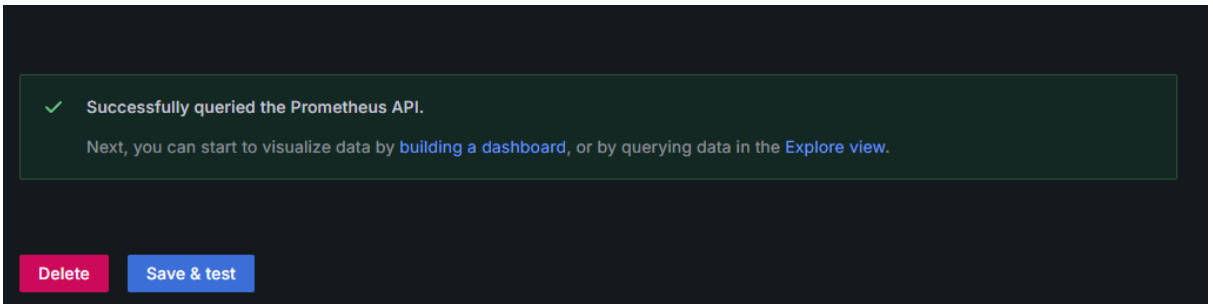
Open a web browser and navigate to Grafana using your server's IP address

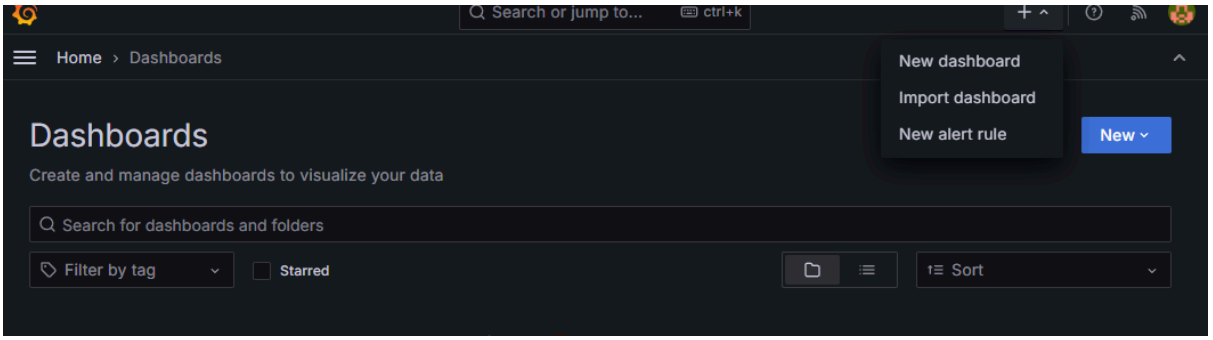
The default port for Grafana is 3000. For example:
`http://<your-server-ip>:3000`

The default password and username are admin
Choose Promethues as a data source




Search or jump to... ctrl+k
Home › Connections › Data sources › prometheus
Connection
Prometheus server URL *
http://44.220.15.226:9090/
Authentication
Authentication methods
Choose an authentication method to access the data source
No Authentication
TLS settings
Additional security measures that can be applied on top of authentication
Add self-signed certificate
TLS Client Authentication
Skip TLS certificate validation
HTTP headers
Pass along additional context and metadata about the request/response
Advanced settings


Successfully queried the Prometheus API.
Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.
Delete
Save & test


Search or jump to... ctrl+k
Home › Dashboards
New dashboard
Import dashboard
New alert rule
New
Dashboards
Create and manage dashboards to visualize your data
Search for dashboards and folders
Filter by tag
Starred
Sort

Google node exporter grafana dashboard

Tous Images Vidéos Actualités Livres Web Finance Outils

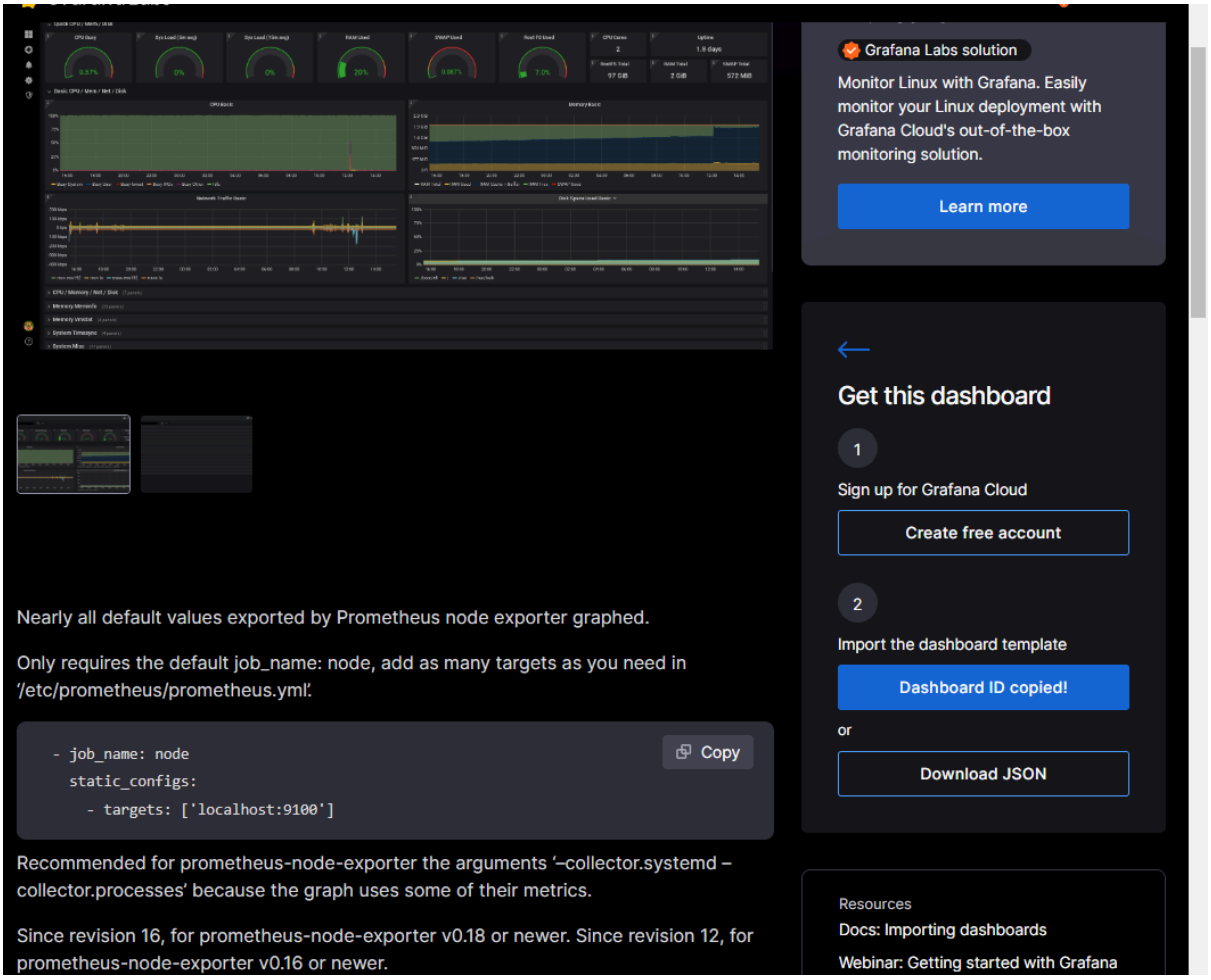
Grafana
https://grafana.com › dashboards · Traduire cette page

Node Exporter Full | Grafana Labs

Easily monitor your Linux deployment with **Grafana** Cloud's out-of-the-box monitoring solution. Learn more. Get this **dashboard**. 1. Sign up for **Grafana** Cloud.

Grafana
https://grafana.com › dashboards · Traduire cette page

Node Exporter Quickstart and Dashboard



Nearly all default values exported by Prometheus node exporter graphed.

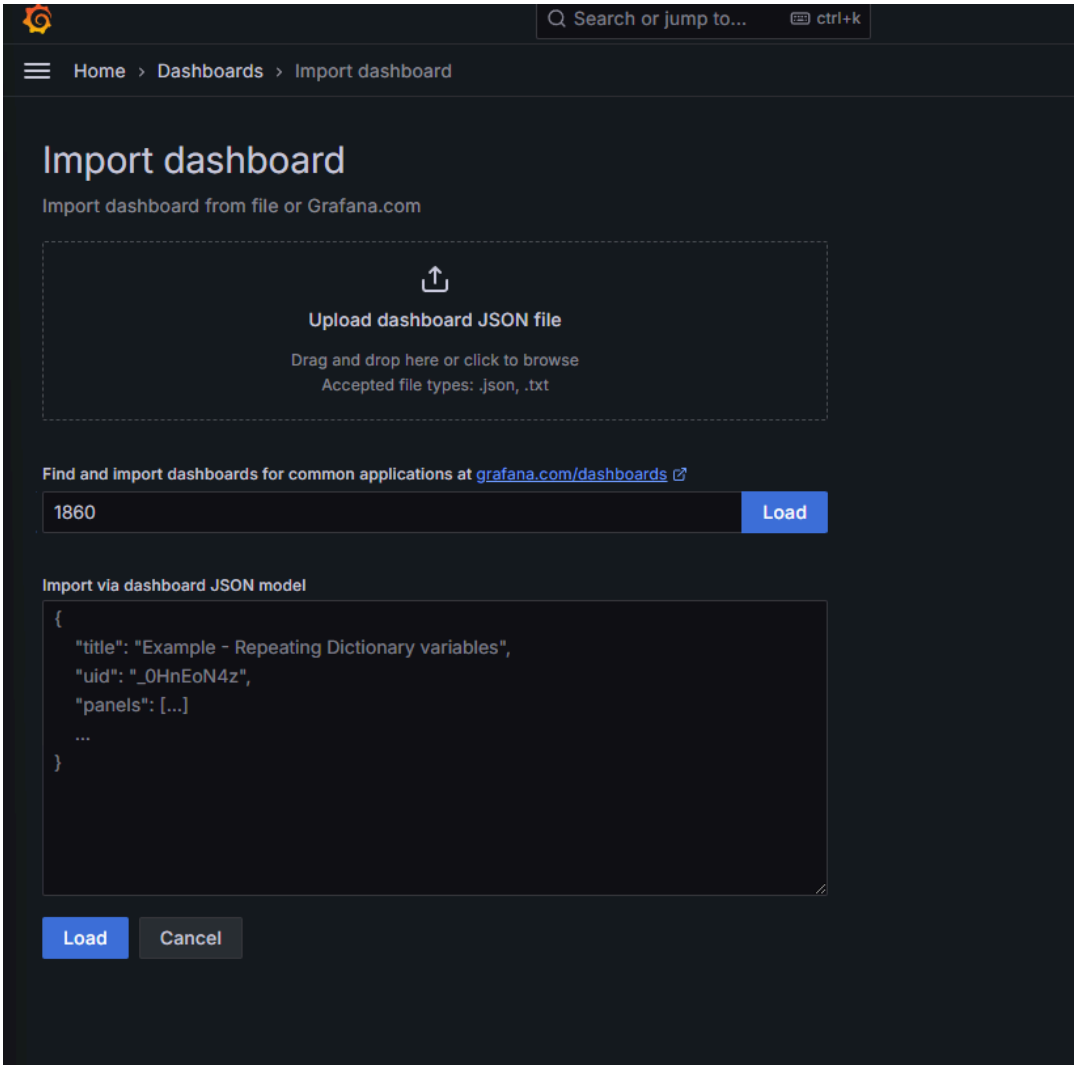
Only requires the default job_name: node, add as many targets as you need in '/etc/prometheus/prometheus.yml'.

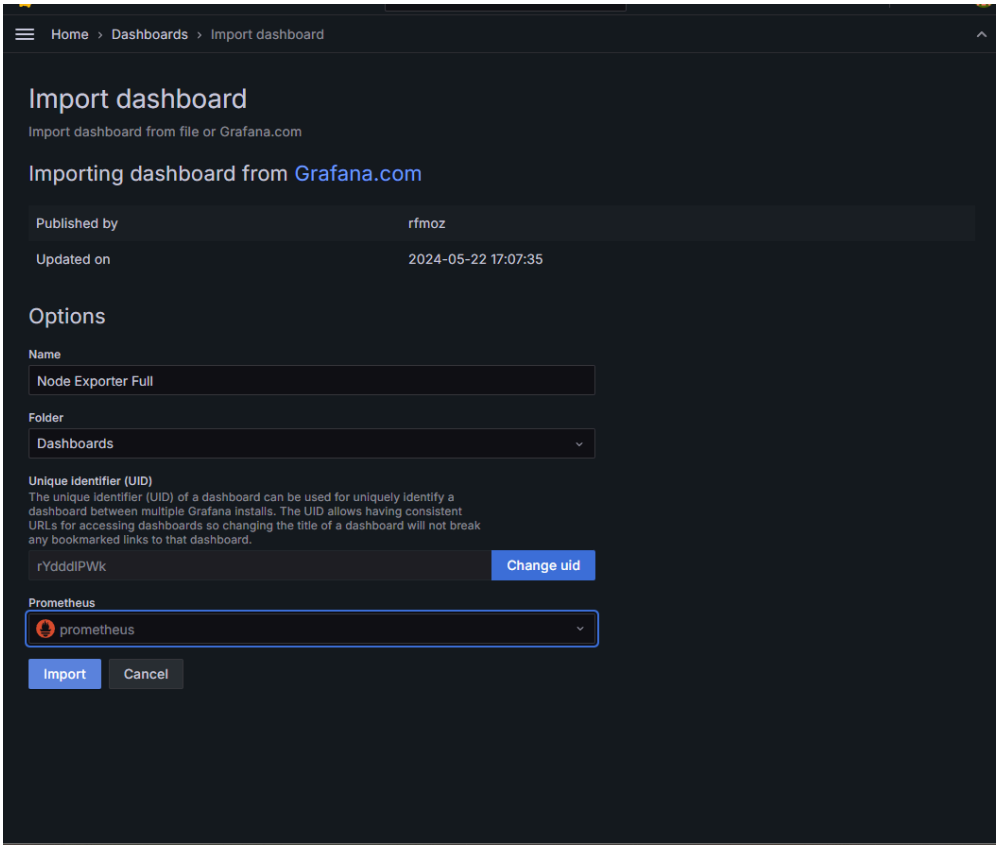
```
- job_name: node
  static_configs:
    - targets: ['localhost:9100']
```

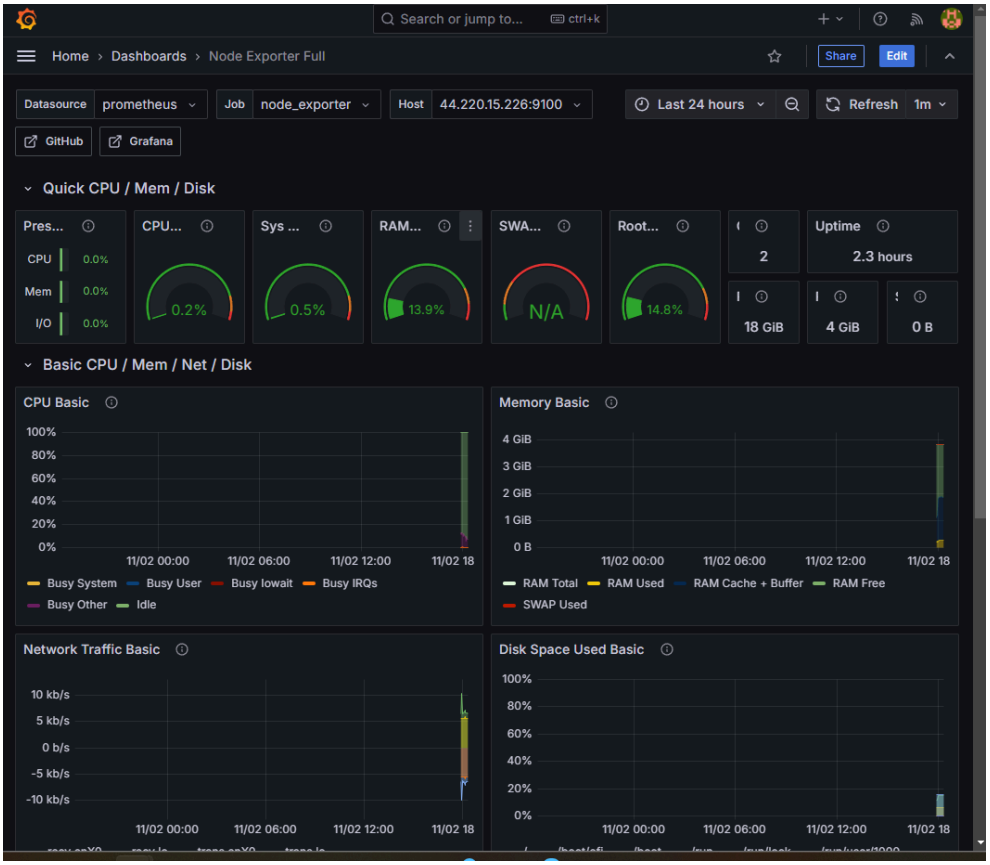
Copy

Recommended for prometheus-node-exporter the arguments '–collector.systemd –collector.processes' because the graph uses some of their metrics.

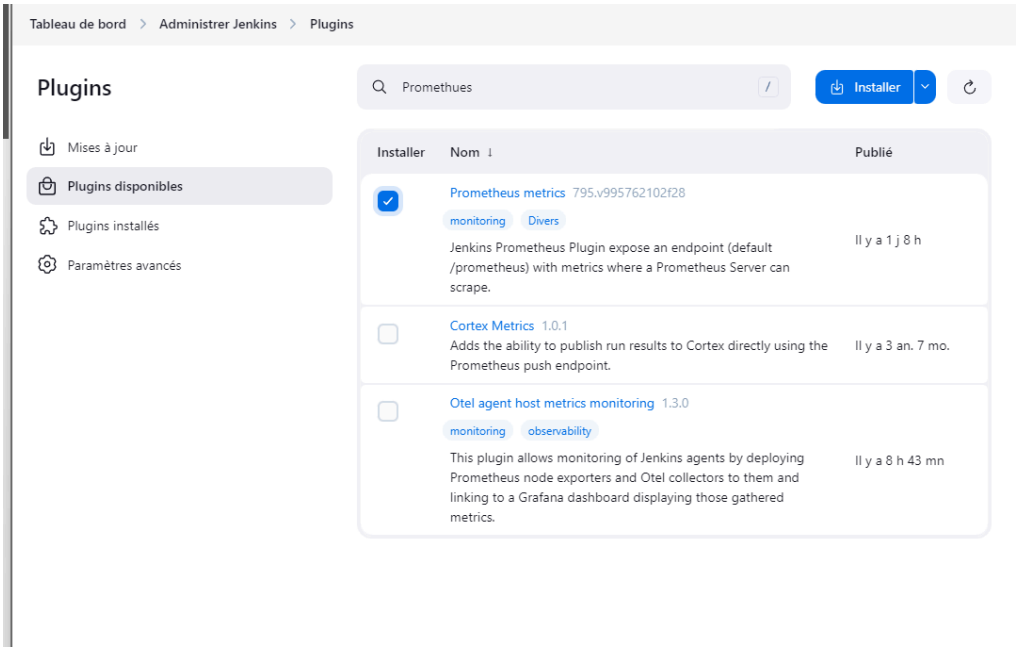
Since revision 16, for prometheus-node-exporter v0.18 or newer. Since revision 12, for prometheus-node-exporter v0.16 or newer.


Search or jump to...
ctrl+k
Home › Dashboards › Import dashboard
Import dashboard
Import dashboard from file or Grafana.com
Upload dashboard JSON file
Drag and drop here or click to browse
Accepted file types: .json, .txt
Find and import dashboards for common applications at grafana.com/dashboards
1860
Load
Import via dashboard JSON model
{
"title": "Example - Repeating Dictionary variables",
"uid": "_0HnEoN4z",
"panels": [...]
...
}
Load
Cancel





Since we need to monitor jenkins we need to install some plugins and add the jenkins in yaml file

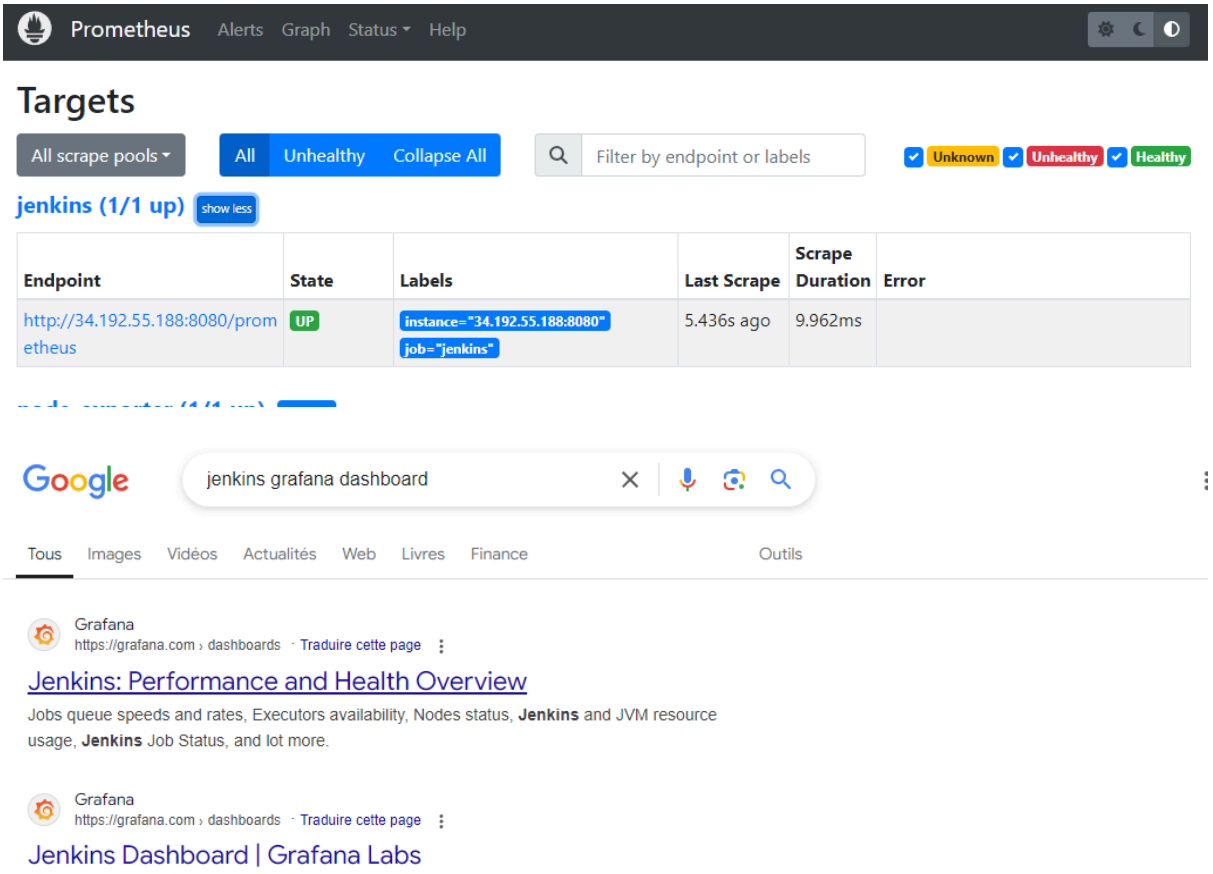


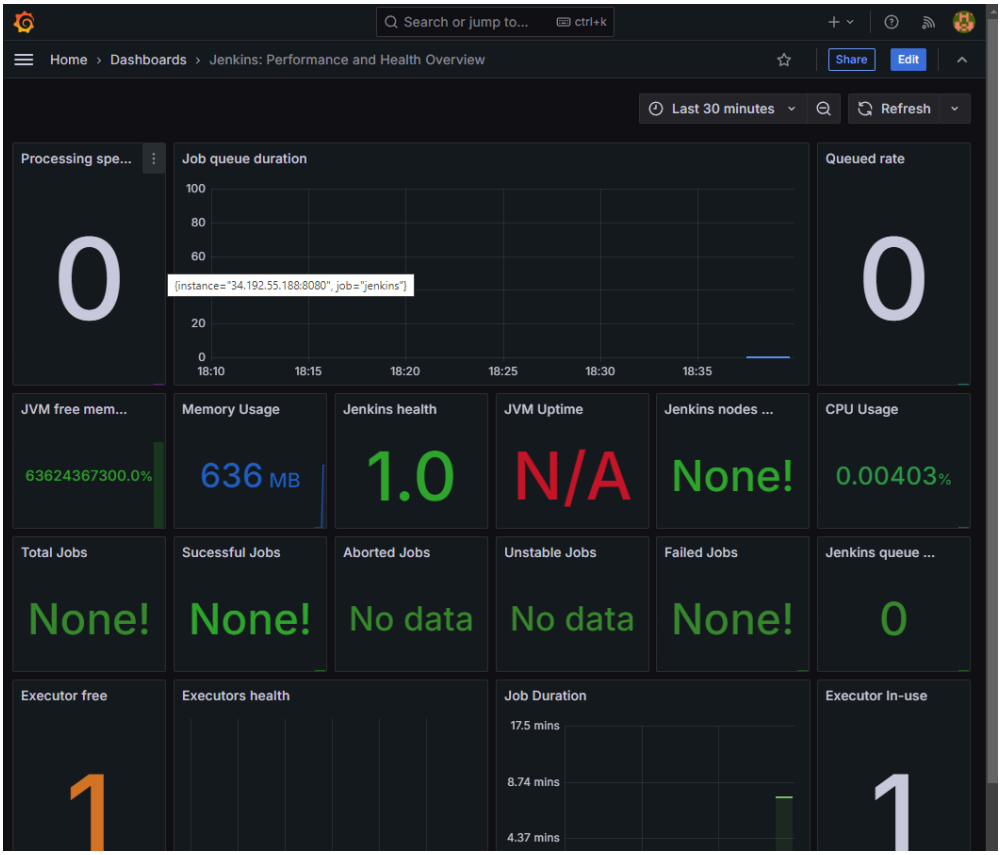
cd /etc/prometheus/
sudo nano prometheus.yml

```
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1
minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1
minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global
'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from
this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "node_exporter"
    static_configs:
      - targets: ["44.220.15.226:9100"]
  - job_name: 'jenkins'
    metrics_path: '/prometheus'
    static_configs:
- targets: ["44.220.15.226:9100"]
  - job_name: 'jenkins'
    metrics_path: '/prometheus'
    static_configs:
      - targets: ['localhost:8080']
```

Check the validity of the configuration file:

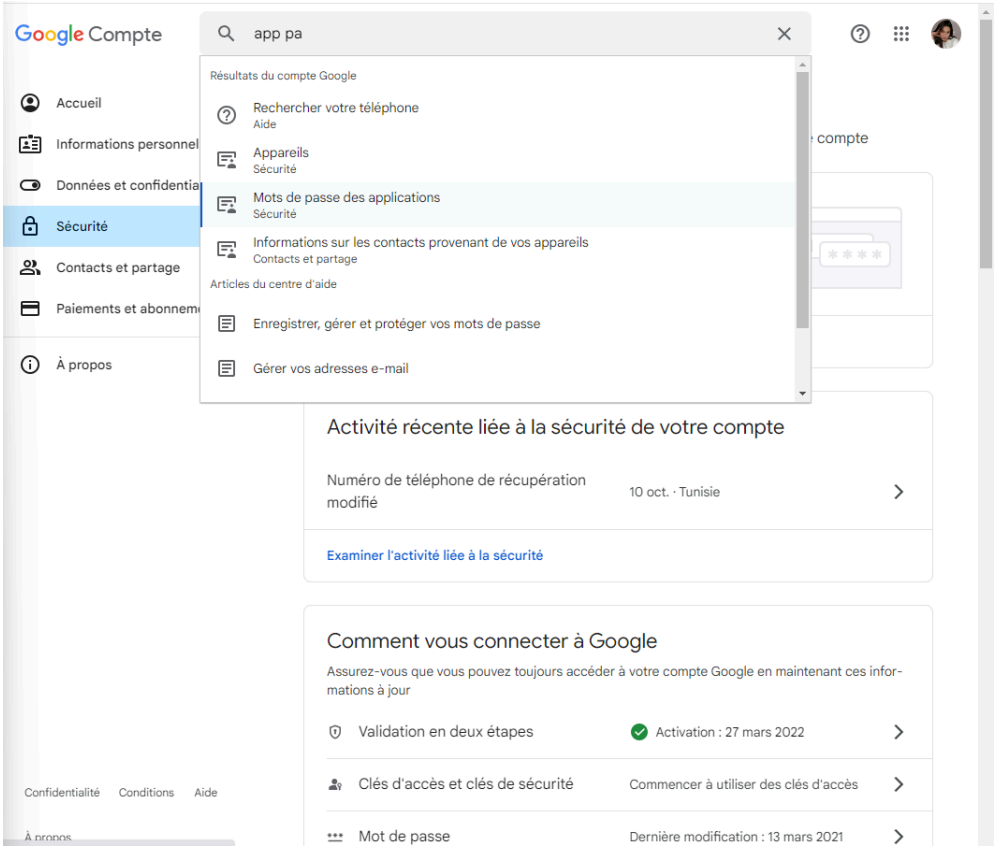
promtool check config /etc/prometheus/prometheus.yml

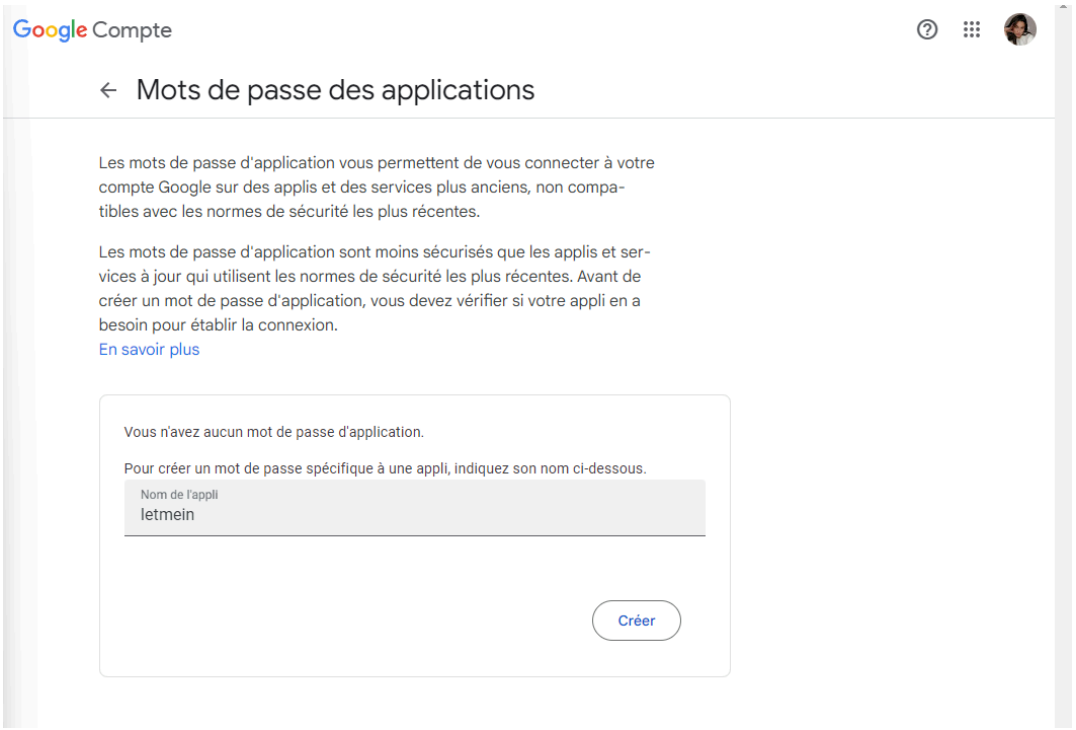


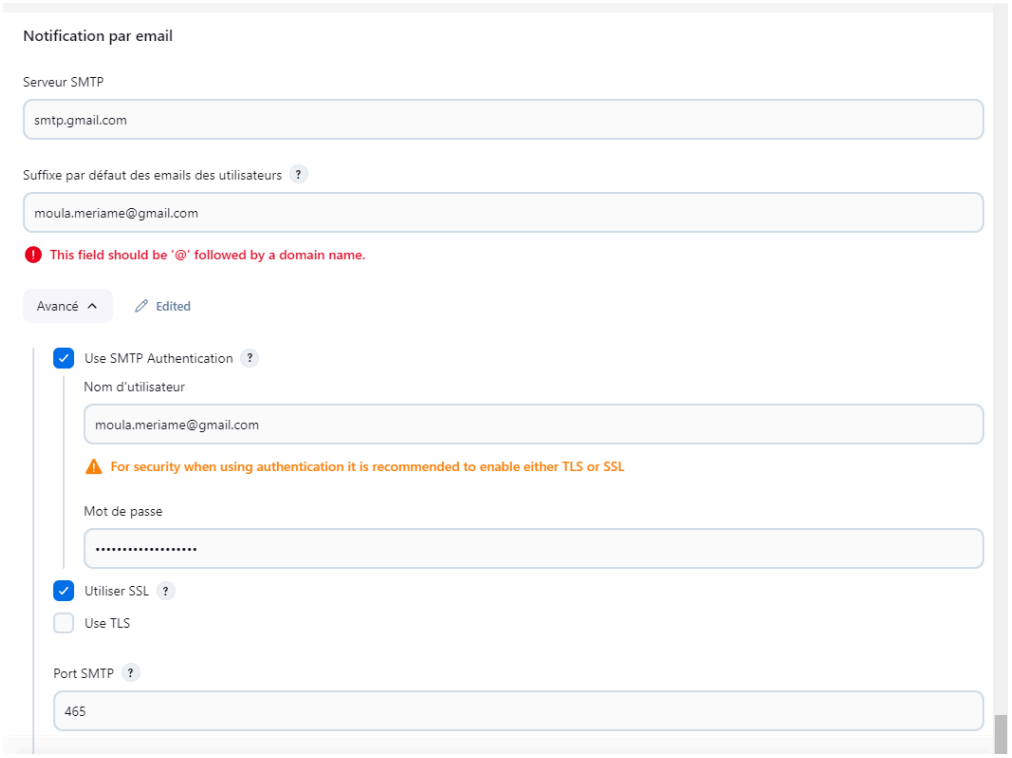


Enabling notifications:

You need to have a gmail account


Google Compte
app pa
Résultats du compte Google
Rechercher votre téléphone
Aide
Appareils
Sécurité
Mots de passe des applications
Sécurité
Informations sur les contacts provenant de vos appareils
Contacts et partage
Articles du centre d'aide
Enregistrer, gérer et protéger vos mots de passe
Gérer vos adresses e-mail
Accueil
Informations personnel
Données et confidentia
Sécurité
Contacts et partage
Paiements et abonnem
À propos
compte
Activité récente liée à la sécurité de votre compte
Numéro de téléphone de récupération modifié
10 oct. · Tunisie
Examiner l'activité liée à la sécurité
Comment vous connecter à Google
Assurez-vous que vous pouvez toujours accéder à votre compte Google en maintenant ces informations à jour
Validation en deux étapes
Activation : 27 mars 2022
Clés d'accès et clés de sécurité
Commencer à utiliser des clés d'accès
Mot de passe
Dernière modification : 13 mars 2021
Confidentialité
Conditions
Aide
À propos


Google Compte
Mots de passe des applications
Les mots de passe d'application vous permettent de vous connecter à votre compte Google sur des applis et des services plus anciens, non compatibles avec les normes de sécurité les plus récentes.
Les mots de passe d'application sont moins sécurisés que les applis et services à jour qui utilisent les normes de sécurité les plus récentes. Avant de créer un mot de passe d'application, vous devez vérifier si votre appli en a besoin pour établir la connexion.
En savoir plus
Vous n'avez aucun mot de passe d'application.
Pour créer un mot de passe spécifique à une appli, indiquez son nom ci-dessous.
Nom de l'appli
letmein
Créer


Notification par email
Serveur SMTP
smtp.gmail.com
Suffixe par défaut des emails des utilisateurs
moula.meriame@gmail.com
This field should be '@' followed by a domain name.
Avancé
Edited
Use SMTP Authentication
Nom d'utilisateur
moula.meriame@gmail.com
For security when using authentication it is recommended to enable either TLS or SSL
Mot de passe
Utiliser SSL
Use TLS
Port SMTP
465

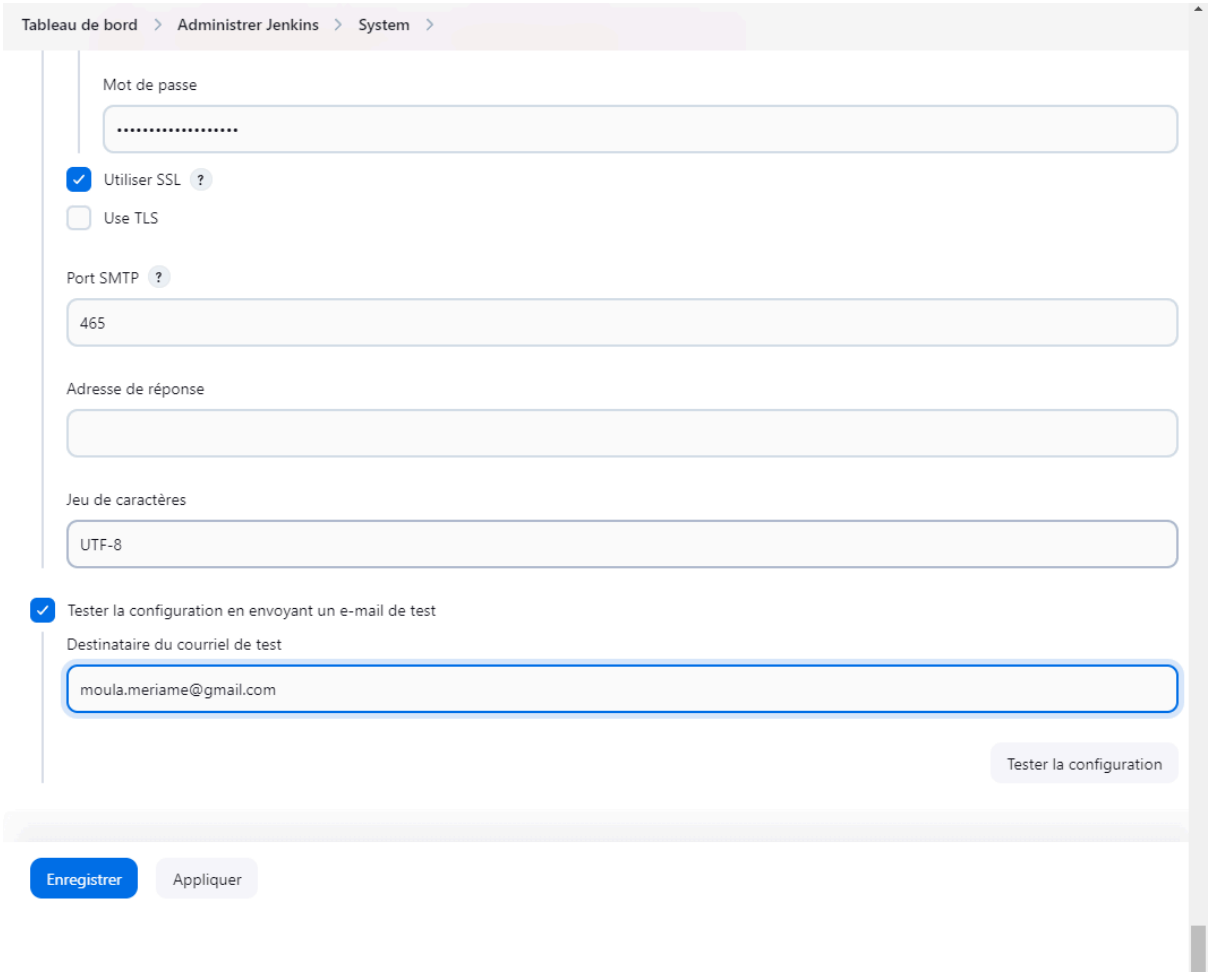
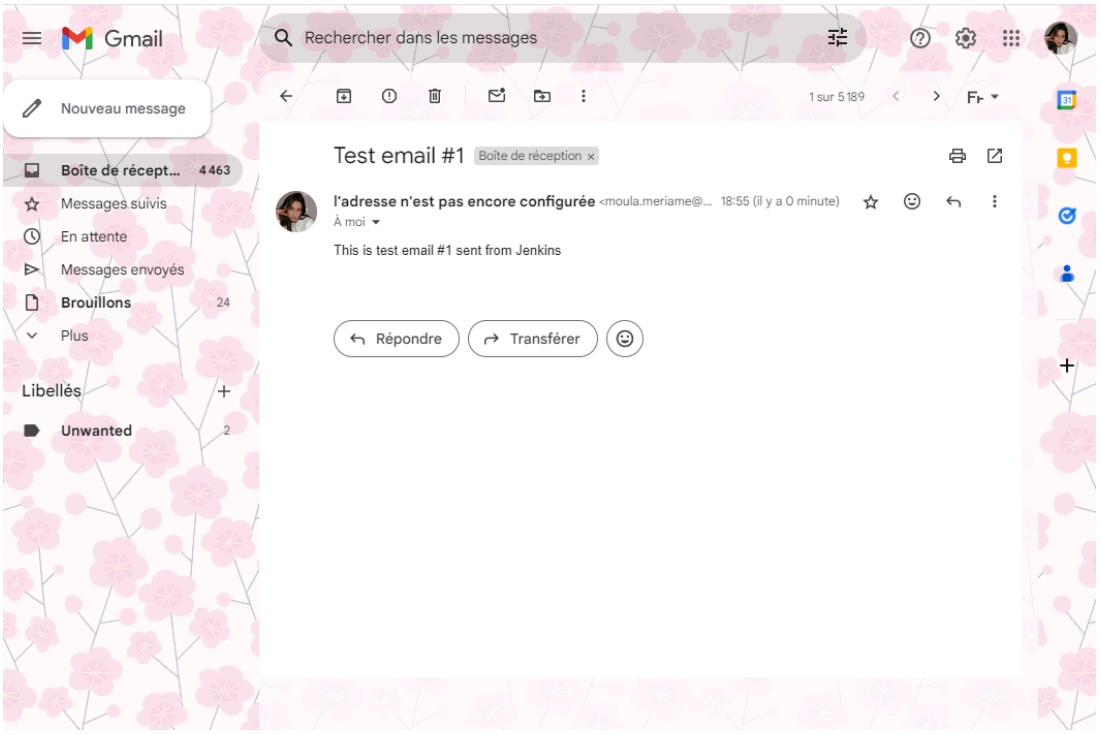
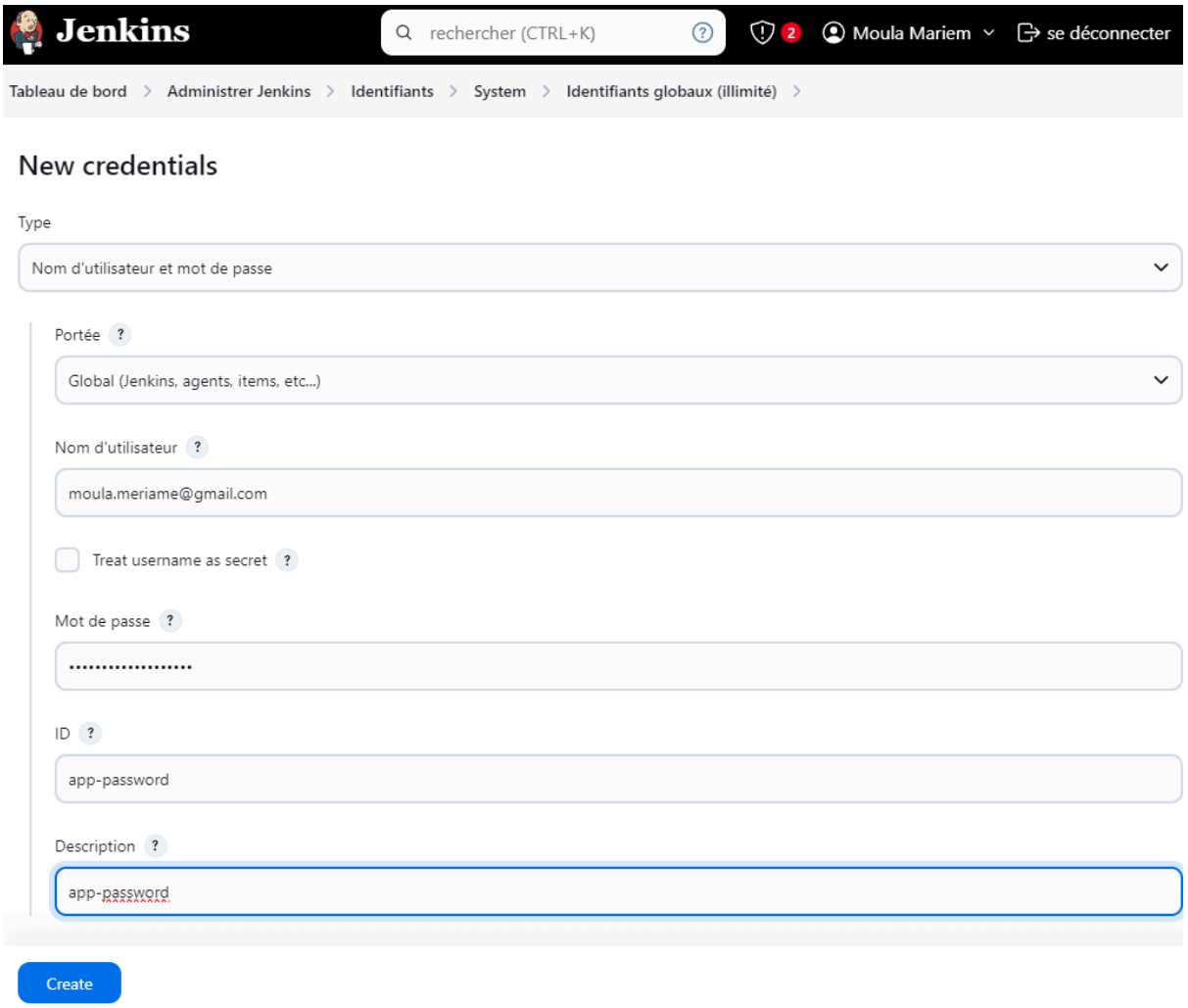


Tableau de bord
Administrer Jenkins
System
Mot de passe
Utiliser SSL
Use TLS
Port SMTP
465
Adresse de réponse
Jeu de caractères
UTF-8
Tester la configuration en envoyant un e-mail de test
Destinataire du courriel de test
moula.meriame@gmail.com
Tester la configuration
Enregistrer
Appliquer


Gmail
Rechercher dans les messages
Nouveau message
Boîte de récept... 4 463
Messages suivis
En attente
Messages envoyés
Brouillons 24
Plus
Libellés
Unwanted 2
1 sur 5 189
Test email #1
Boîte de réception ×
l'adresse n'est pas encore configurée <moula.meriame@... 18:55 (il y a 0 minute)
À moi
This is test email #1 sent from Jenkins
Répondre
Transférer

Make sure to add the credentials in the jenkins :



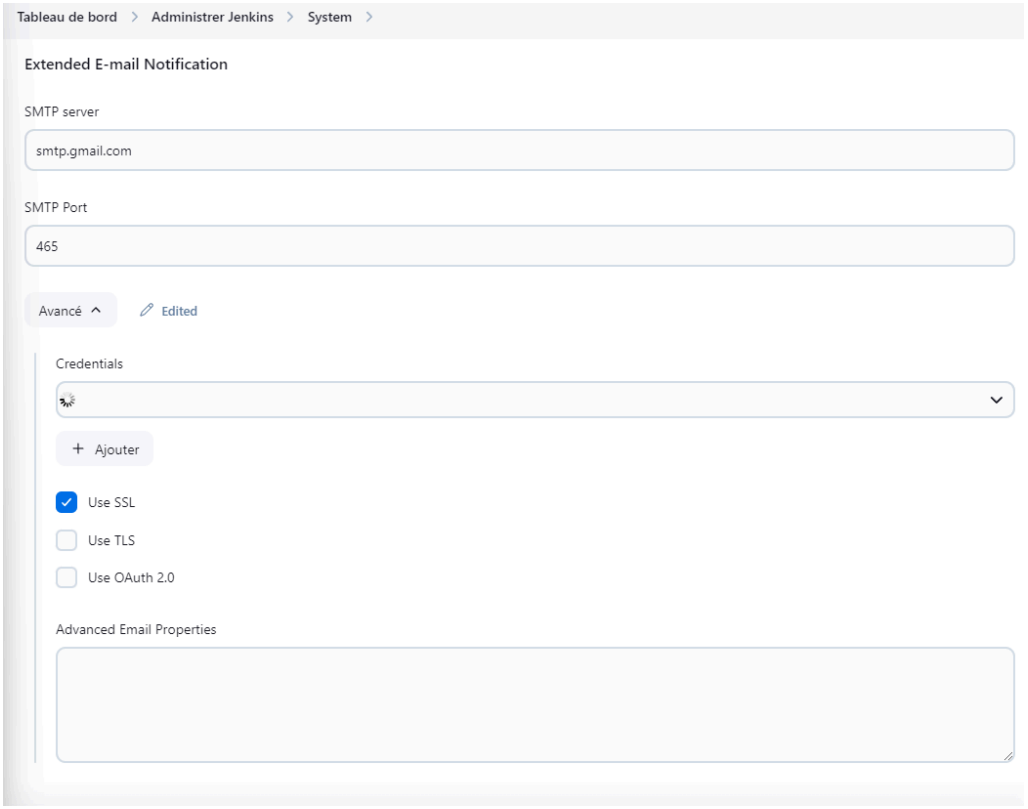
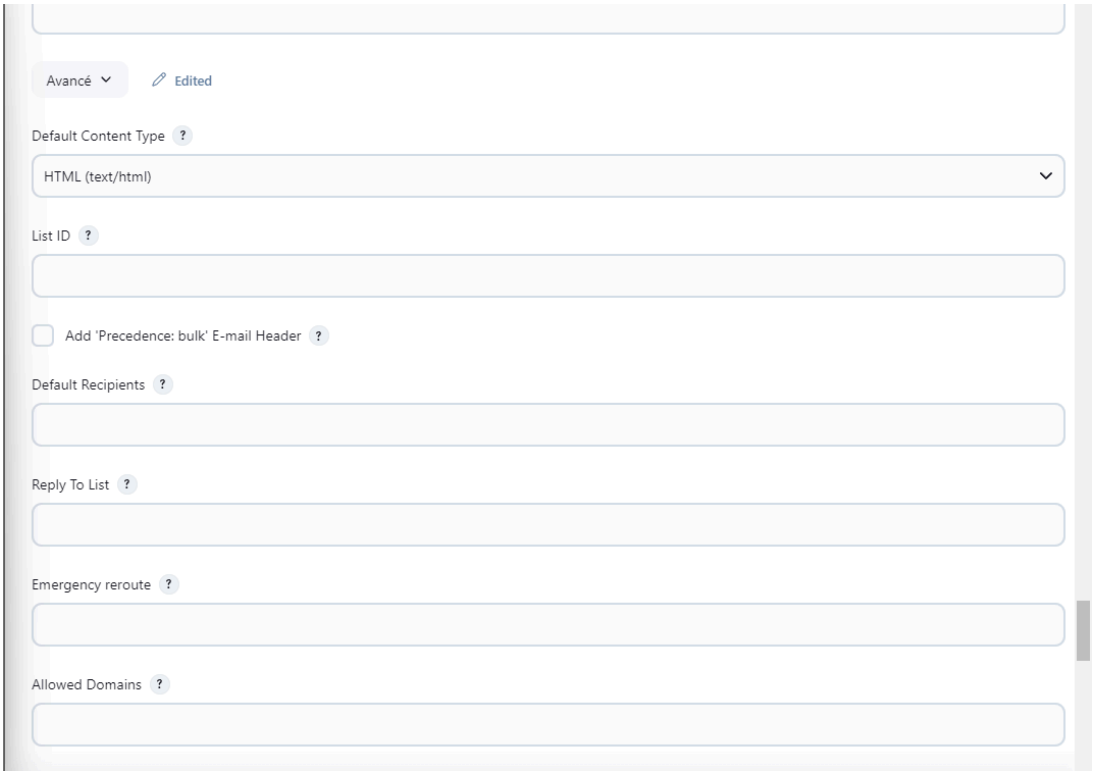
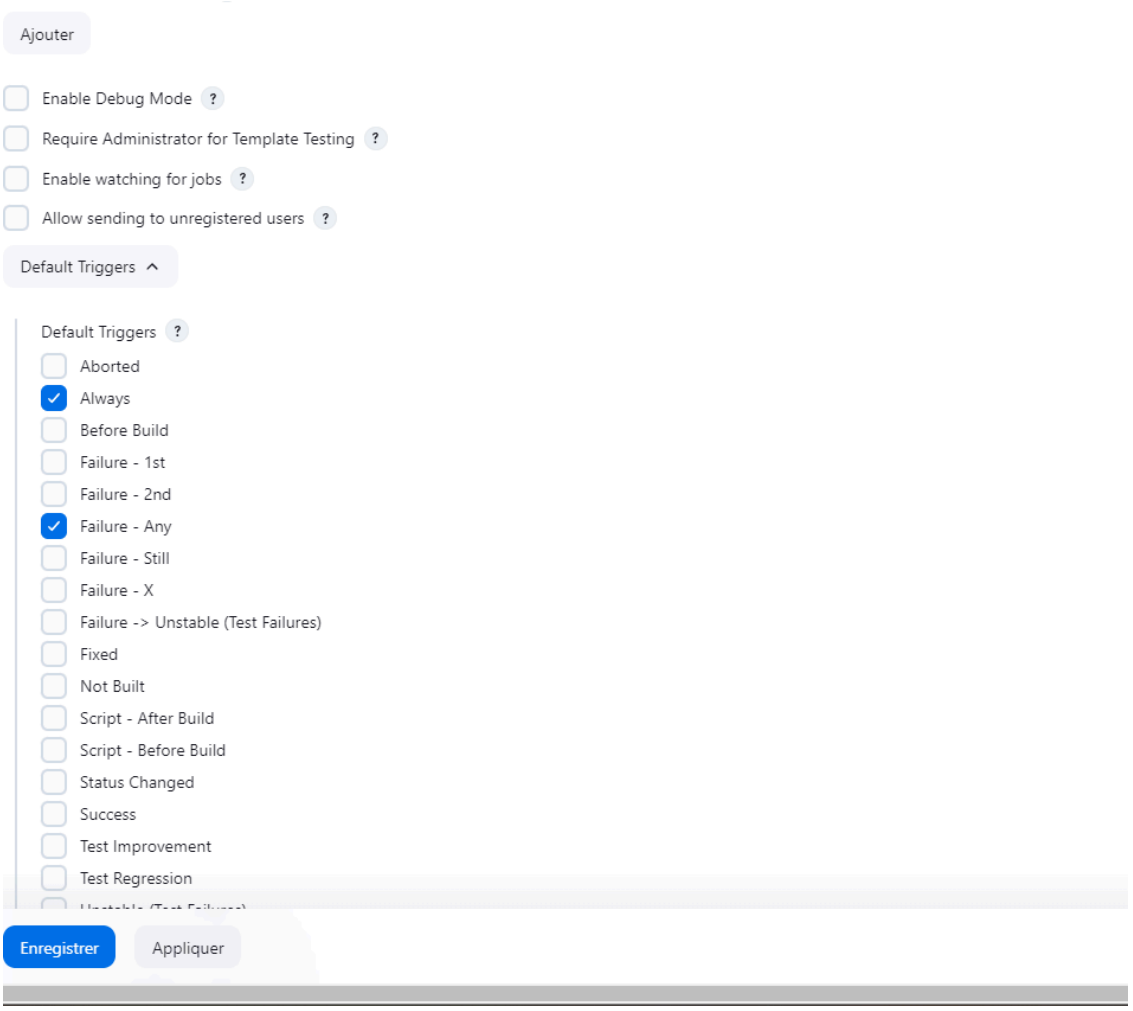


Tableau de bord > Administrer Jenkins > System >
Extended E-mail Notification
SMTP server
smtp.gmail.com
SMTP Port
465
Avancé
Edited
Credentials
Ajouter
Use SSL
Use TLS
Use OAuth 2.0
Advanced Email Properties


Avancé
Edited
Default Content Type
HTML (text/html)
List ID
Add 'Precedence: bulk' E-mail Header
Default Recipients
Reply To List
Emergency reroute
Allowed Domains



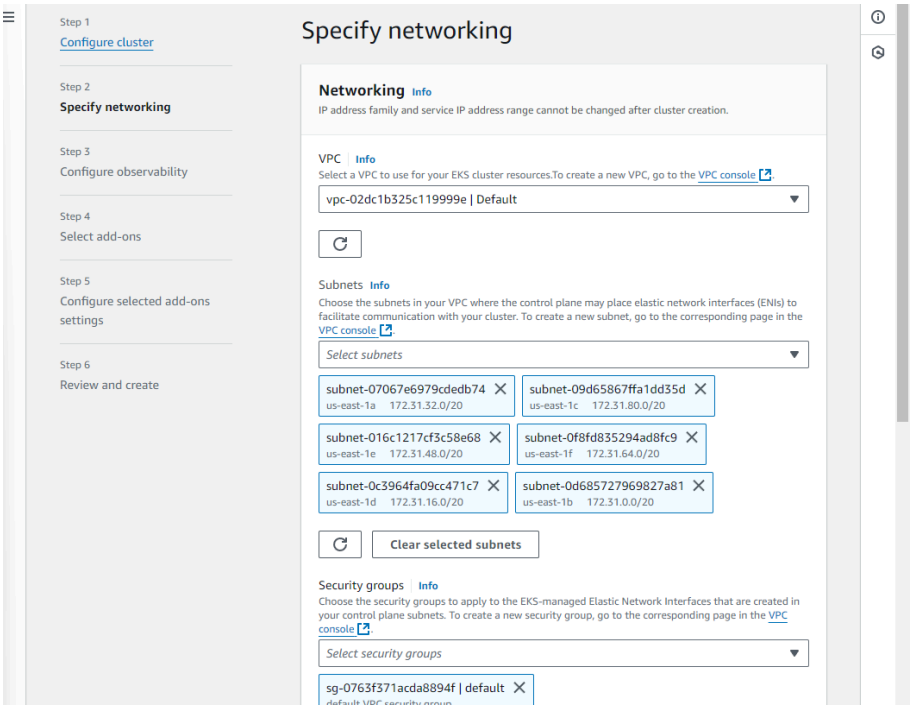
Updating the jenkins script

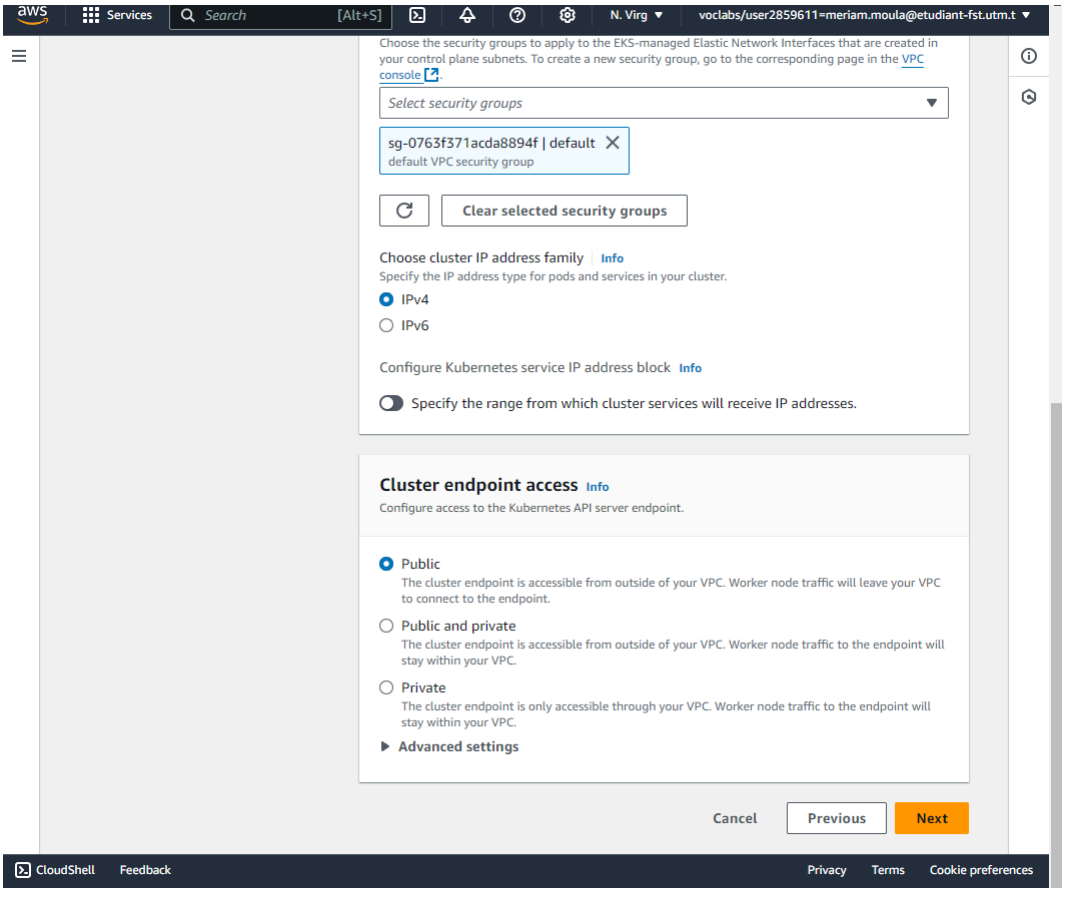
Kubernetes

Now moving to deploying using EKS

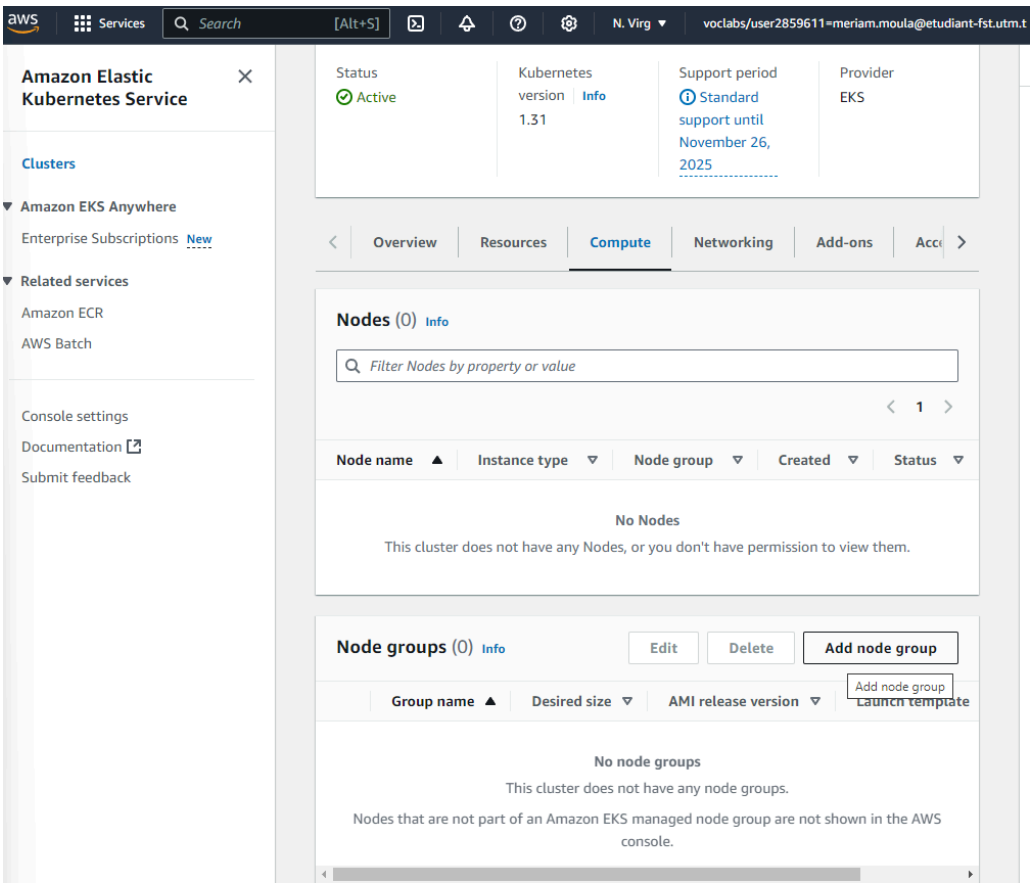
Create Kubernetes Cluster with Nodegroups

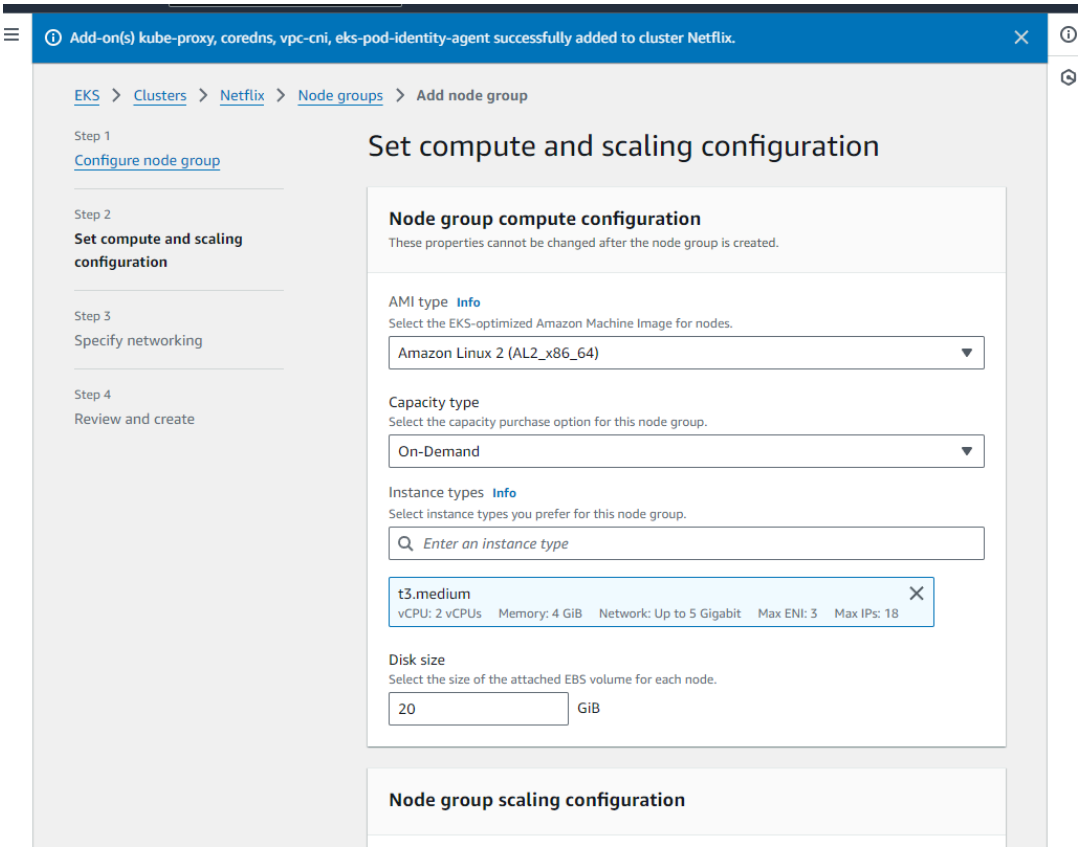
In this phase, you'll set up a Kubernetes cluster with node groups. This will provide a scalable environment to deploy and manage your applications.

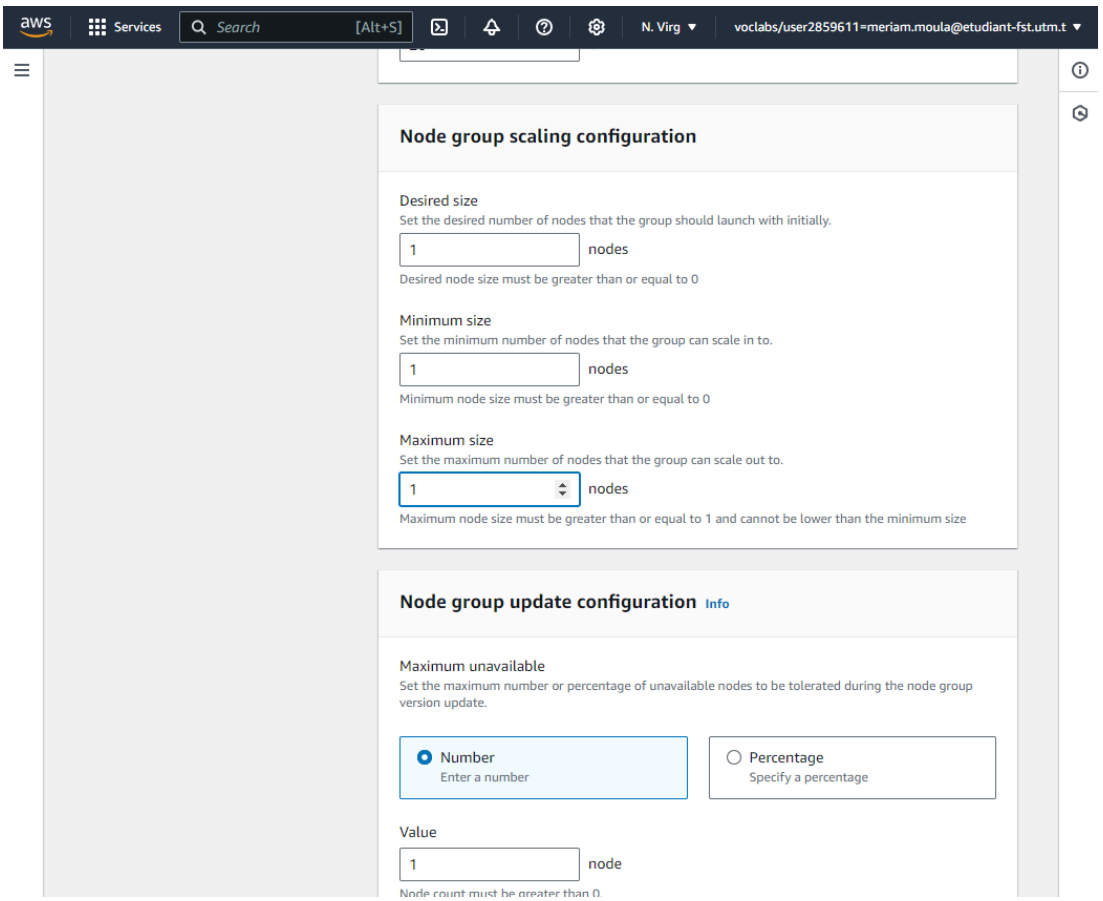




Once the cluster is created , we move to creating node groups


aws
Services
Search
[Alt+S]
N. Virg
voclabs/user2859611=meriam.moula@etudiant-fst.utm.t
Amazon Elastic Kubernetes Service
Clusters
Amazon EKS Anywhere
Enterprise Subscriptions New
Related services
Amazon ECR
AWS Batch
Console settings
Documentation
Submit feedback
Status
Active
Kubernetes version Info
1.31
Support period
Standard support until November 26, 2025
Provider
EKS
Overview
Resources
Compute
Networking
Add-ons
Nodes (0) Info
Filter Nodes by property or value
Node name
Instance type
Node group
Created
Status
No Nodes
This cluster does not have any Nodes, or you don't have permission to view them.
Node groups (0) Info
Edit
Delete
Add node group
Add node group
Group name
Desired size
AMI release version
Launch template
No node groups
This cluster does not have any node groups.
Nodes that are not part of an Amazon EKS managed node group are not shown in the AWS console.


Add-on(s) kube-proxy, coredns, vpc-cni, eks-pod-identity-agent successfully added to cluster Netflix.
EKS > Clusters > Netflix > Node groups > Add node group
Step 1
Configure node group
Step 2
Set compute and scaling configuration
Step 3
Specify networking
Step 4
Review and create
Set compute and scaling configuration
Node group compute configuration
These properties cannot be changed after the node group is created.
AMI type Info
Select the EKS-optimized Amazon Machine Image for nodes.
Amazon Linux 2 (AL2_x86_64)
Capacity type
Select the capacity purchase option for this node group.
On-Demand
Instance types Info
Select instance types you prefer for this node group.
Enter an instance type
t3.medium
vCPU: 2 vCPUs
Memory: 4 GiB
Network: Up to 5 Gigabit
Max ENI: 3
Max IPs: 18
Disk size
Select the size of the attached EBS volume for each node.
20
GiB
Node group scaling configuration



Logging to aws with our local cli using our aws credentials :

aws configure set aws_access_key_id
aws configure set aws_secret_access_key
aws configure set aws_session_token
verify with command aws sts get-caller-identity and aws configure get region
To Add a new context
aws eks update-kubeconfig --name Netflix --region us-east-1
kubectl get ns

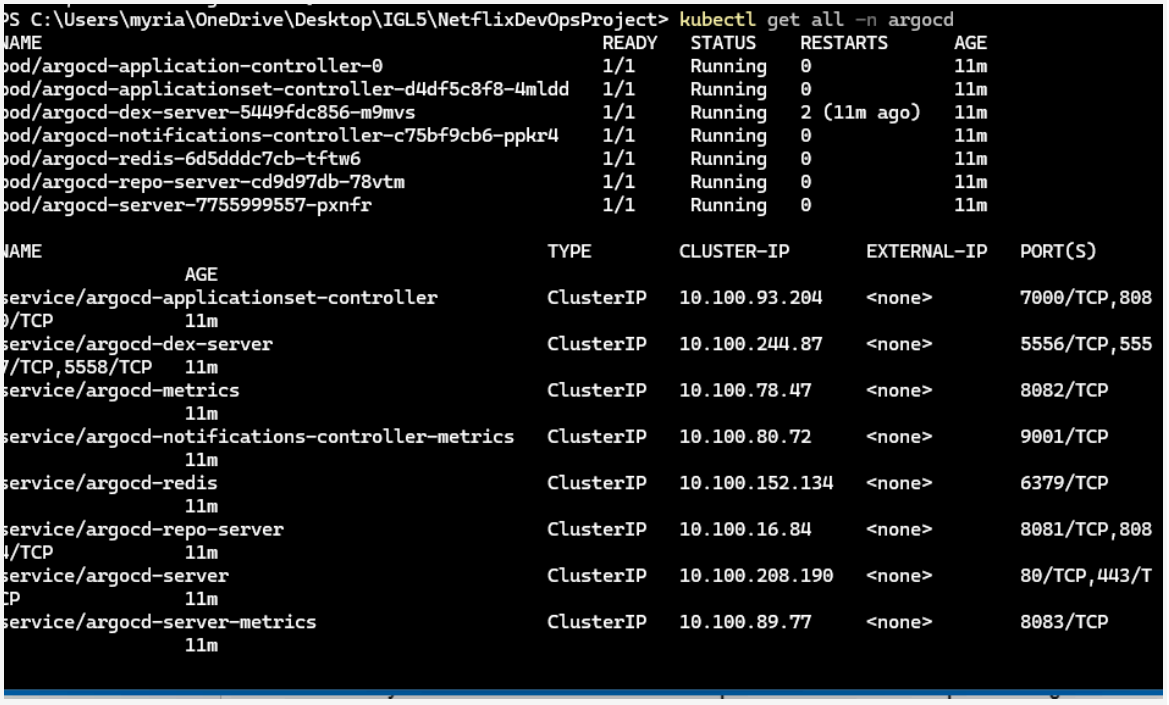
```
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> aws eks update-kubeconfig --name Netflix --
region us-east-1
Added new context arn:aws:eks:us-east-1:417738508223:cluster/Netflix to C:\Users\myria\.kube\config
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> kubectl get ns
NAME              STATUS   AGE
default           Active   23m
kube-node-lease   Active   23m
kube-public       Active   23m
kube-system       Active   23m
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> kubectl get pods
No resources found in default namespace.
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject>
```

Deploy Application with ArgoCD

Install ArgoCD on local machine:

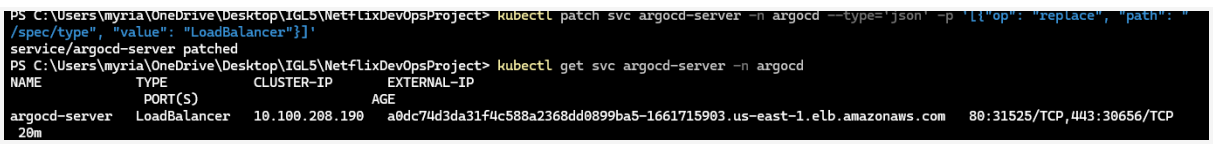
You can install ArgoCD on your Kubernetes cluster by following the instructions provided in https://argo-cd.readthedocs.io/en/stable/getting_started/

kubectl create namespace argocd
kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
kubectl get all -n argocd



Exposing our argocd server

kubectl patch svc argocd-server -n argocd --type='json' -p '[{"op": "replace", "path": "/spec/type", "value": "LoadBalancer"}]'
You can verify the changes by checking the service details again to confirm that its type is now set to LoadBalancer:
kubectl get svc argocd-server -n argocd



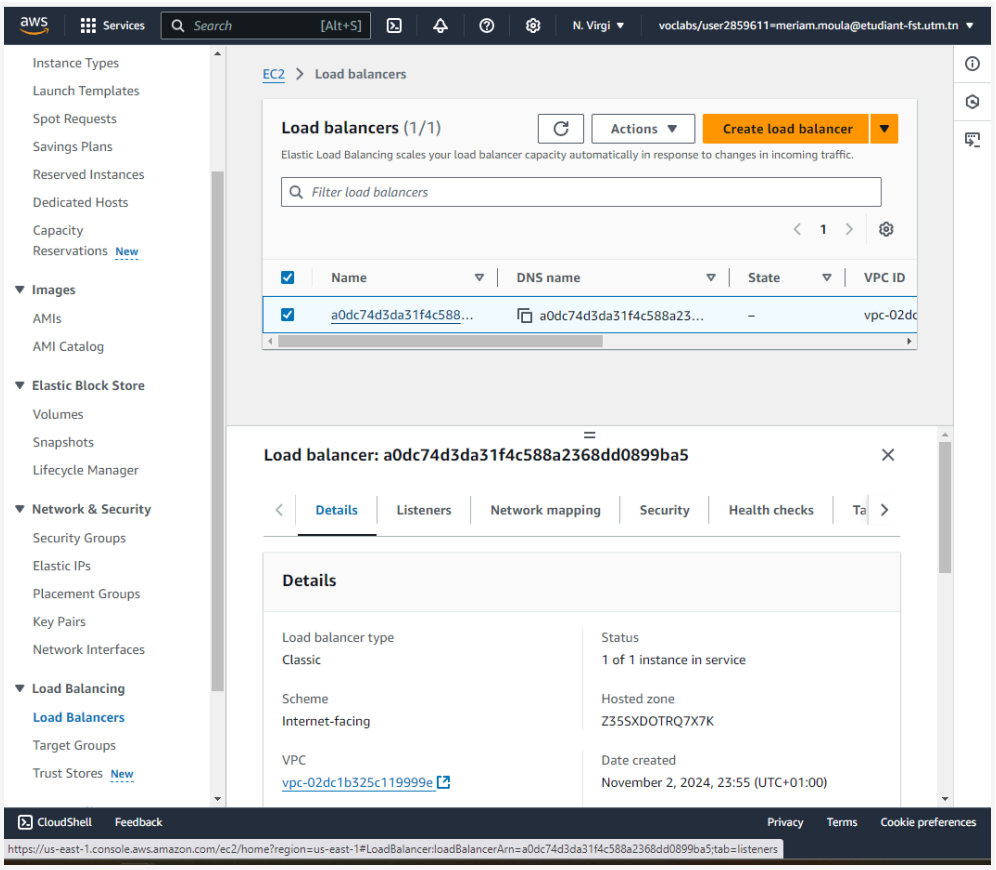
Patching the service to type LoadBalancer in your Kubernetes cluster should trigger the provisioning of a load balancer in your cloud provider, in this case, AWS, if your Kubernetes cluster is configured to work with AWS and you have the necessary permissions.

How It Works

1. Service Type: When you change the service type to `LoadBalancer`, Kubernetes interacts with the underlying cloud provider (AWS in this case) to provision an external load balancer.
2. Cloud Provider Integration: This integration must be correctly configured in your Kubernetes cluster to work with AWS. Typically, this means that your cluster was created using a tool that sets up these integrations (like EKS - Amazon Elastic Kubernetes Service) and has the appropriate IAM roles and policies in place.

AWS Console:

- Log in to the AWS Management Console.
- Navigate to the EC2 Dashboard.
- Click on Load Balancers under the "Load Balancing" section.
- Look for a load balancer with a name that includes `argocd-server` or similar.




```
$ARGOCD_SERVER = (kubectl get svc argocd-server -n argocd -o json | ConvertFrom-Json).status.loadBalancer.ingress[0].hostname
```

This command does the following:

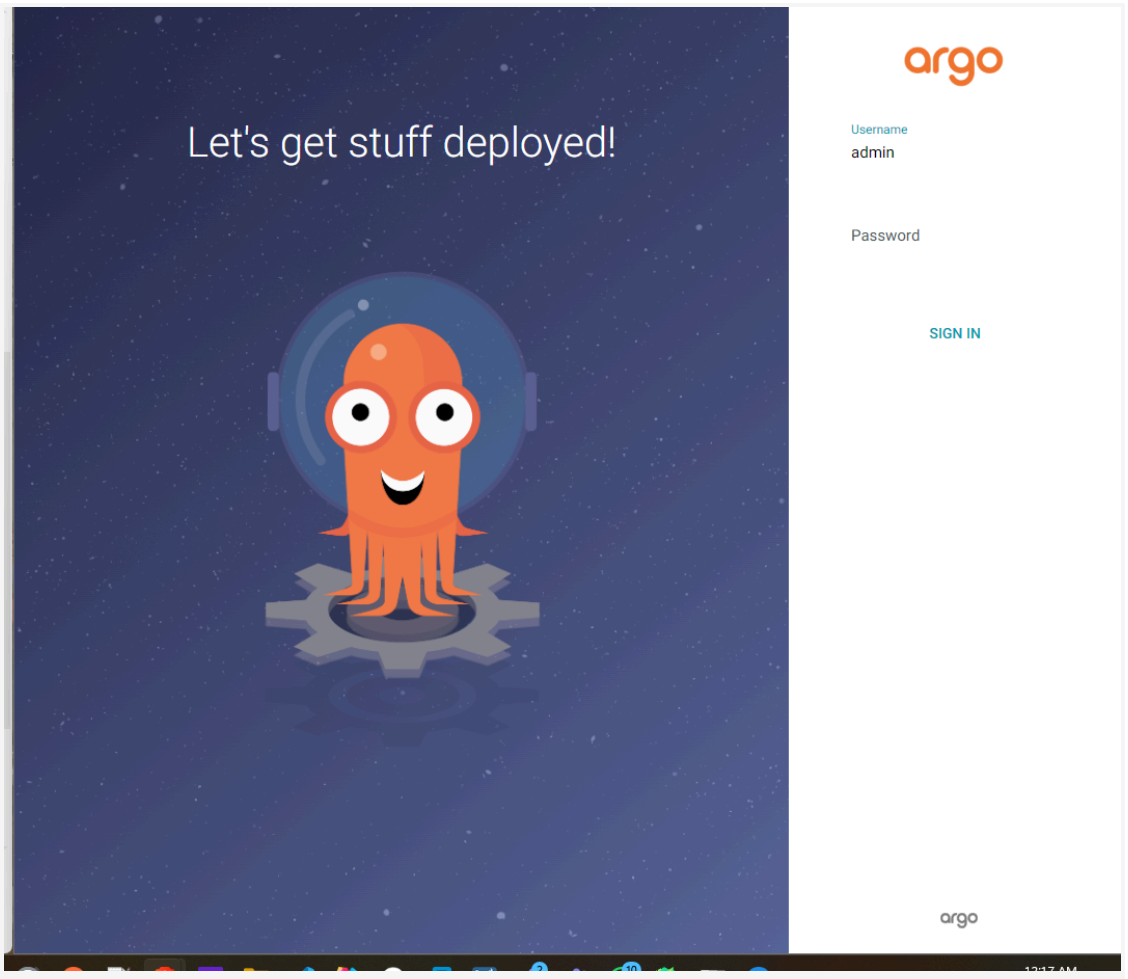
1. Runs `kubectl get svc argocd-server -n argocd -o json` to retrieve the service information in JSON format.
2. Pipes the JSON output to `ConvertFrom-Json`, which allows PowerShell to work with the JSON data as an object.
3. Accesses the nested property `.status.loadBalancer.ingress[0].hostname` and assigns it to the `$ARGOCD_SERVER` variable.

```
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> $ARGOCD_SERVER = (kubectl get s
c argocd-server -n argocd -o json | ConvertFrom-Json).status.loadBalancer.ingress[0].hostname
```

```
echo $ARGOCD_SERVER
```

```
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> echo ^C
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> echo $ARGOCD_SERVER
a0dc74d3da31f4c588a2368dd0899ba5-1661715903.us-east-1.elb.amazonaws.com
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject>
```

This will give me the endpoint to ArgoCD
a0dc74d3da31f4c588a2368dd0899ba5-1661715903.us-east-1.elb.amazonaws.com



To obtain the password we need to run this command

```
$ARGO_PWD =
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String((kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}")))
```

Explanation:

1. `kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}"` retrieves the base64-encoded password.

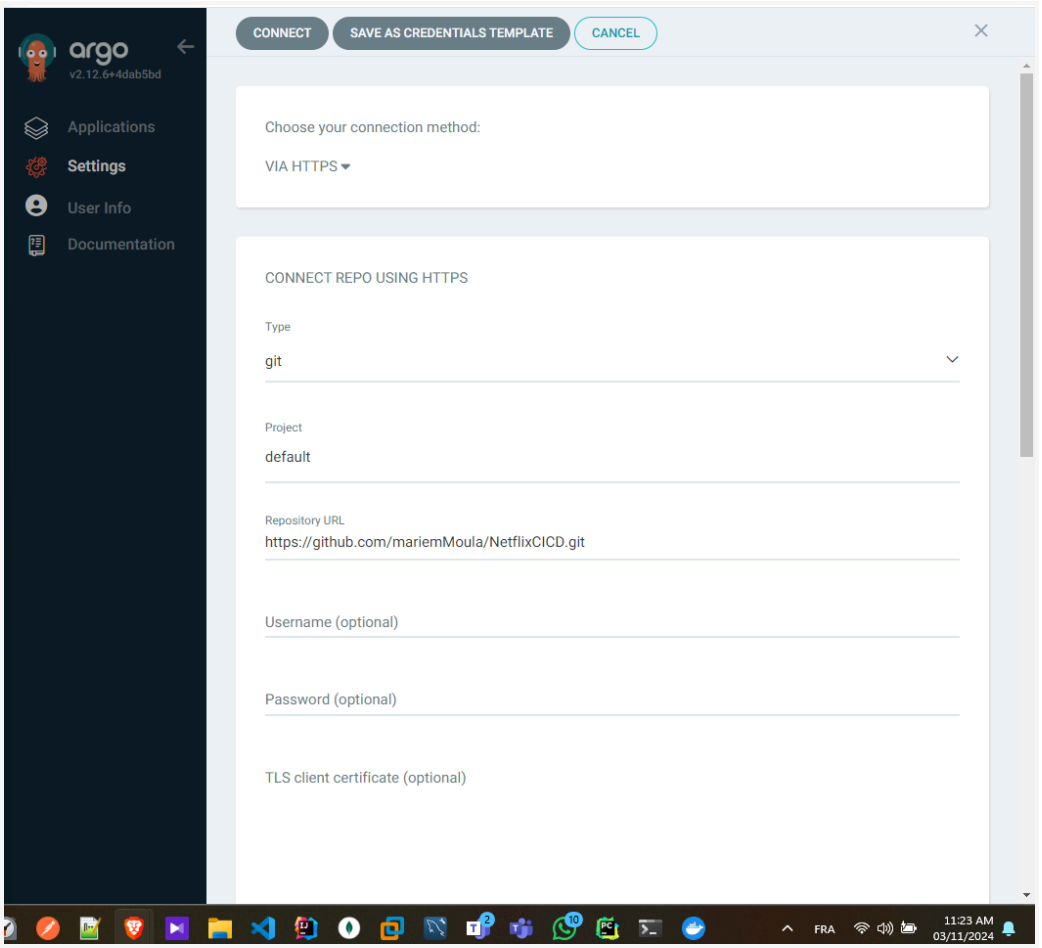
2. `[System.Convert]::FromBase64String(...)` decodes the base64 string.
3. `[System.Text.Encoding]::UTF8.GetString(...)` converts the decoded bytes to a UTF-8 string, making it human-readable.

After running this command, you can view the password by echoing $ARGO_PWD:

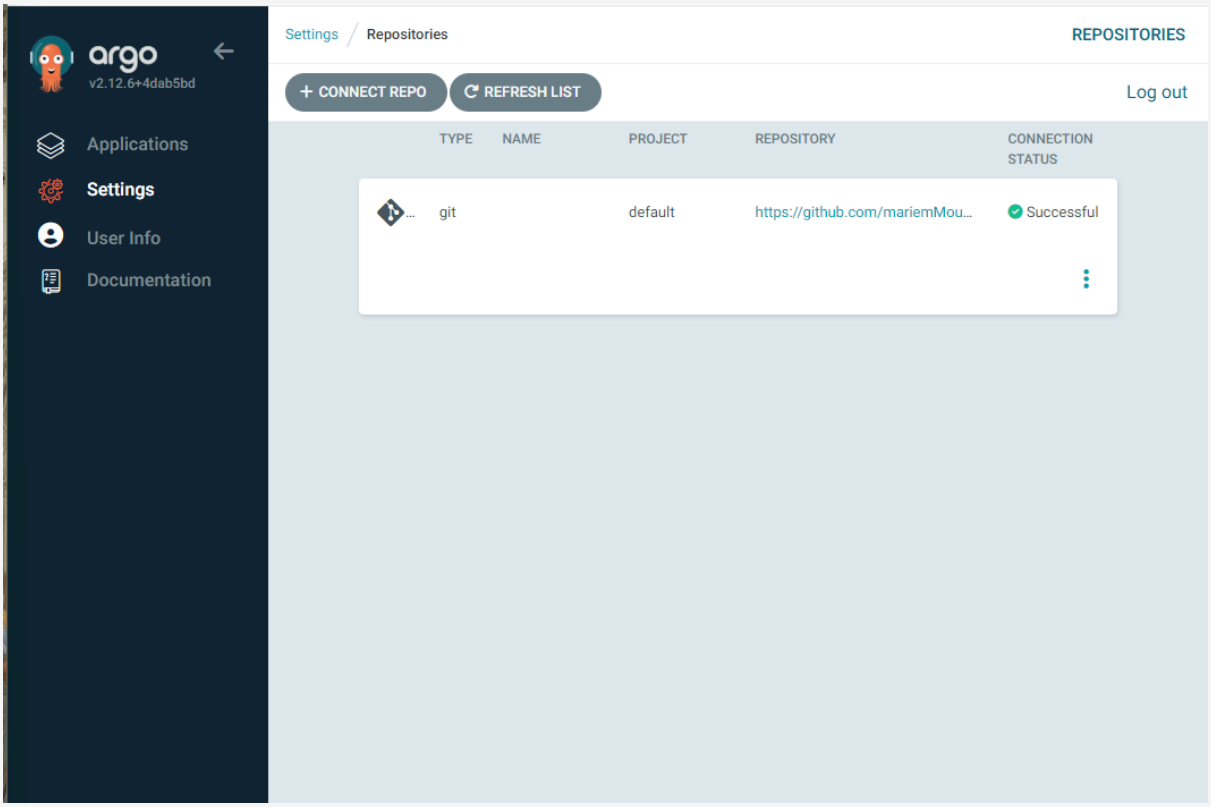
```
Write-Output $ARGO_PWD
```

```
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> $ARGO_PWD = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String((kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}")))
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> Write-Output $ARGO_PWD
6Uen5C361LZUWEfI
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject>
```

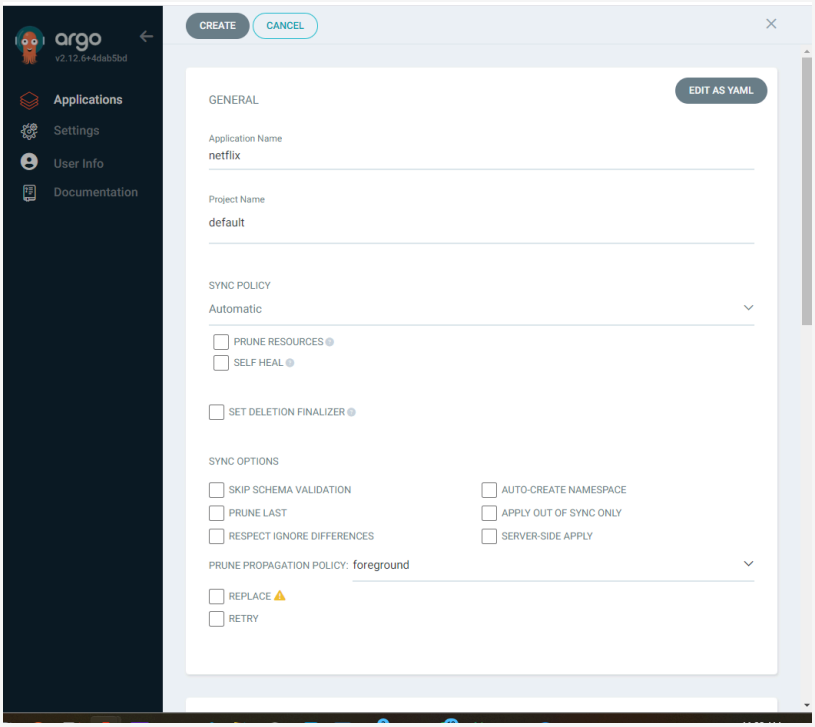
You don't need a password since it's a public repo



After connecting , it should look like this

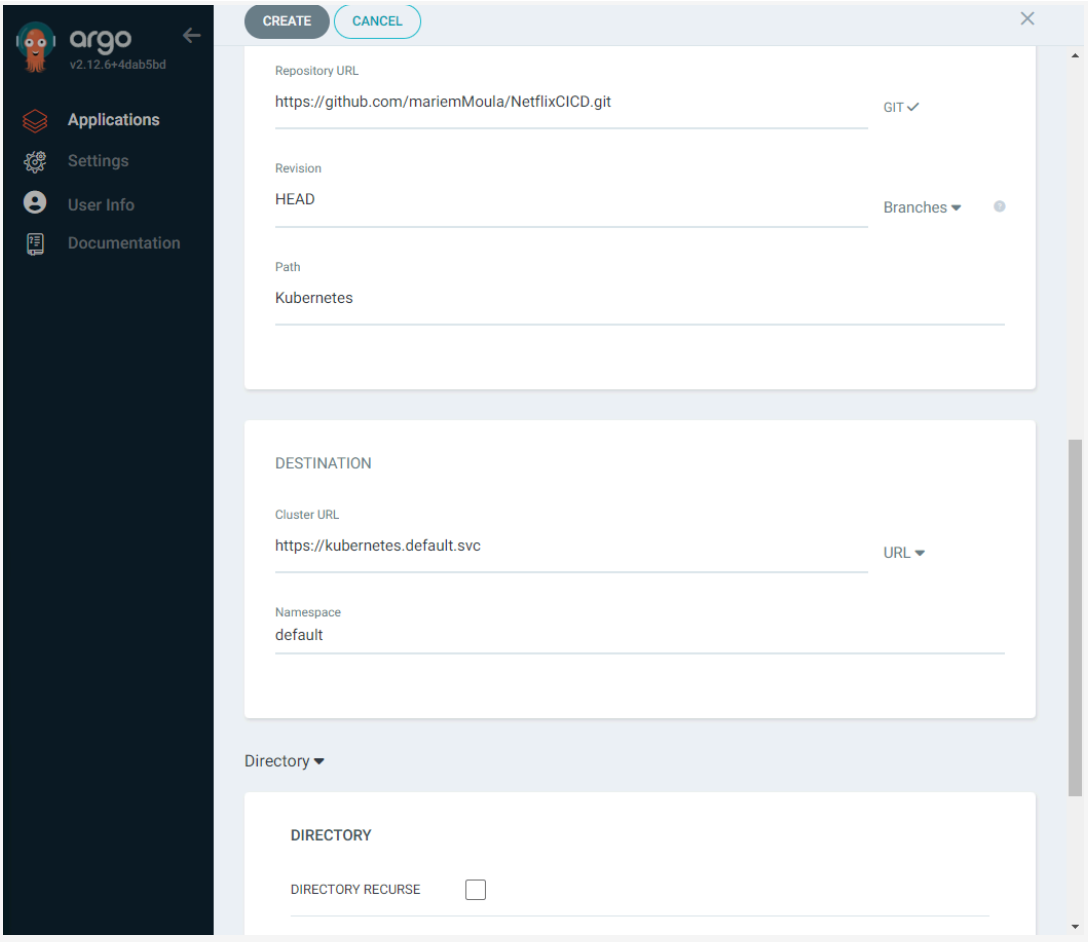


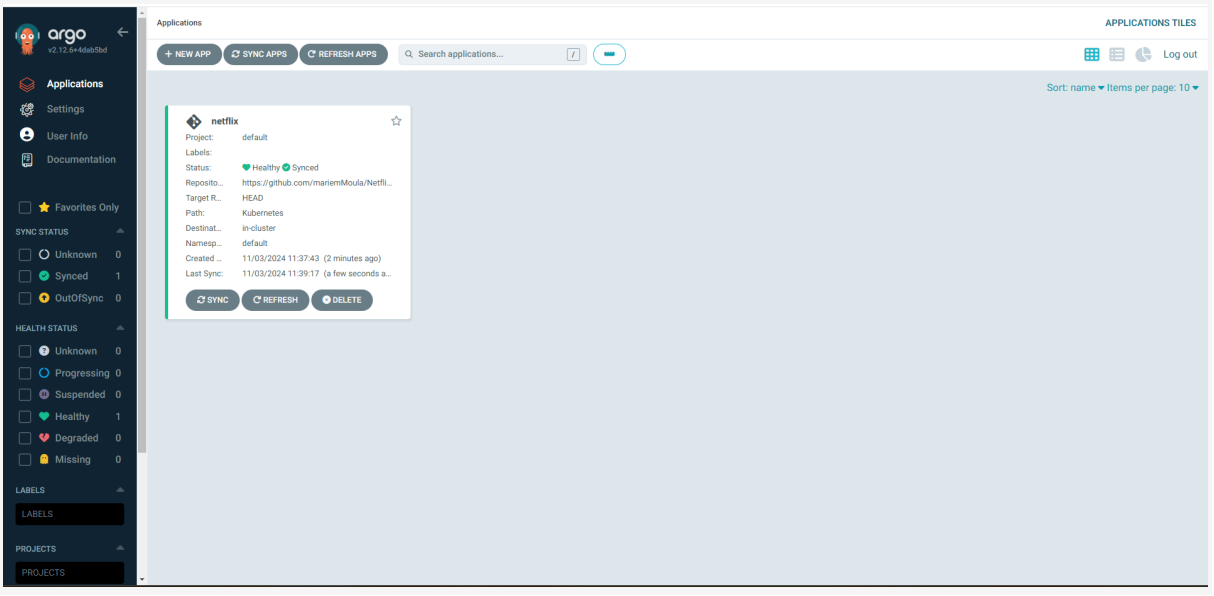
Now we need to create an app which will fetch the data to our Kubernetes folder in the project

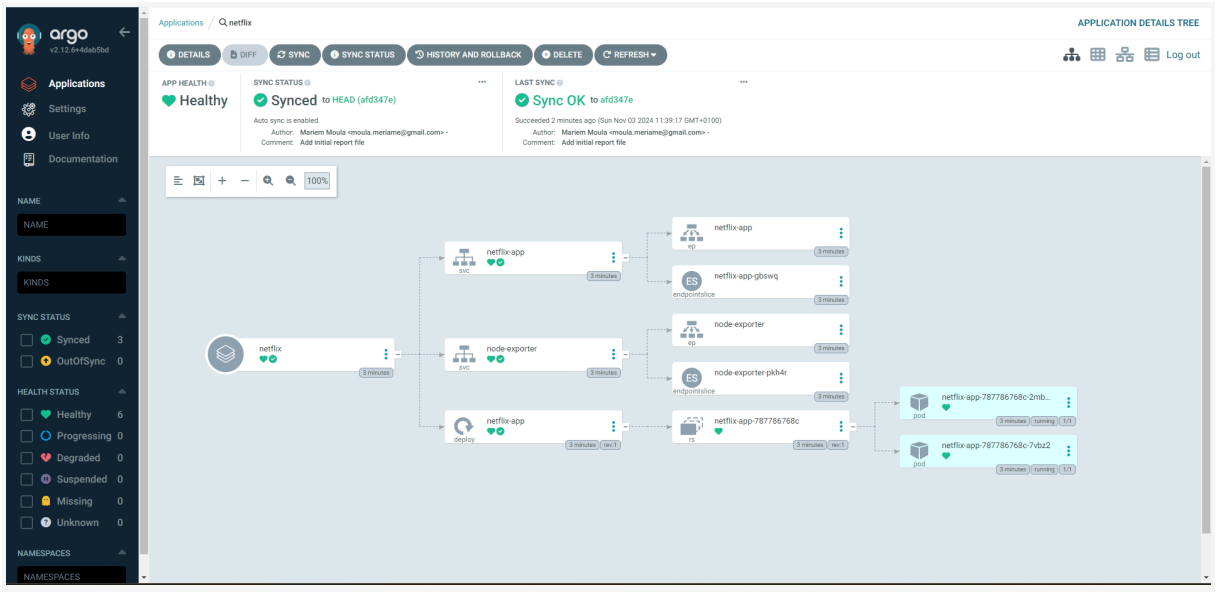


To GET THE NAME SPACE RUN THIS COMMAND : kubectl get ns

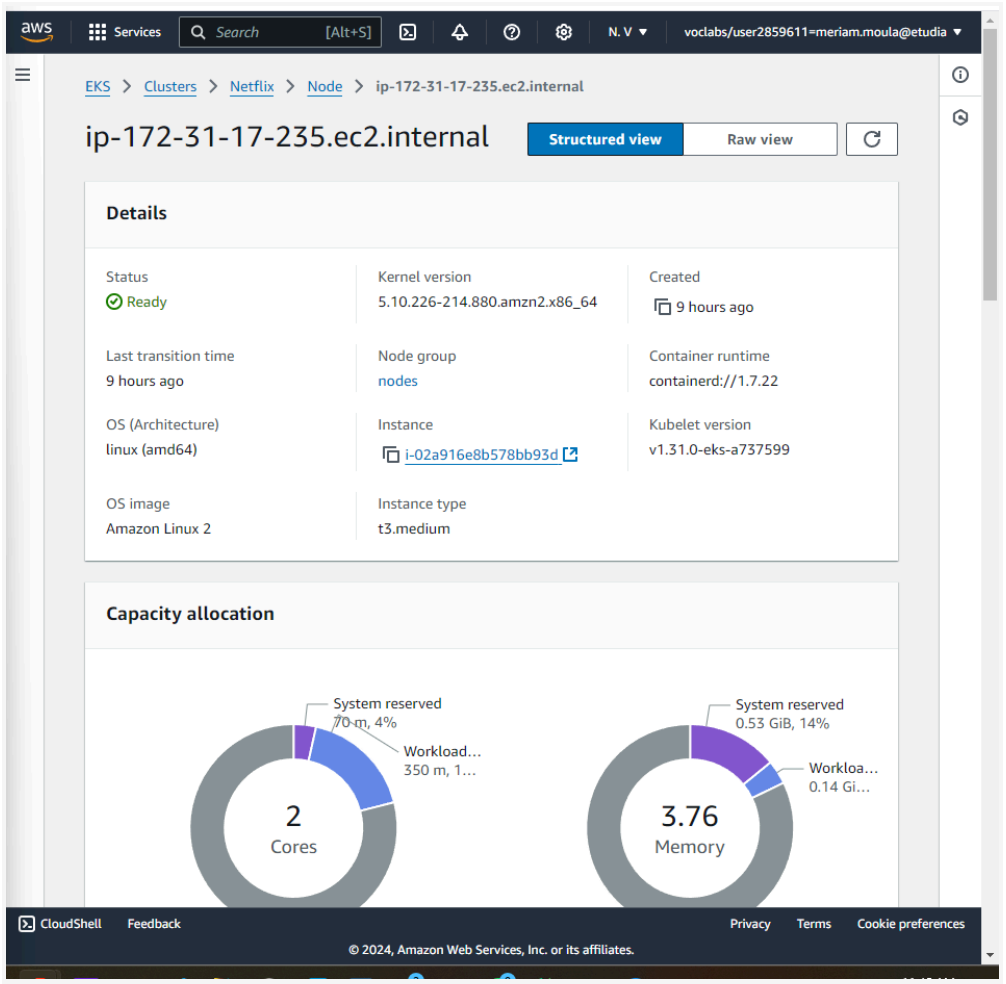
```
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> kubectl get ns
NAME                   STATUS   AGE
argocd                 Active   12h
default                Active   12h
kube-node-lease        Active   12h
kube-public            Active   12h
kube-system            Active   12h
```

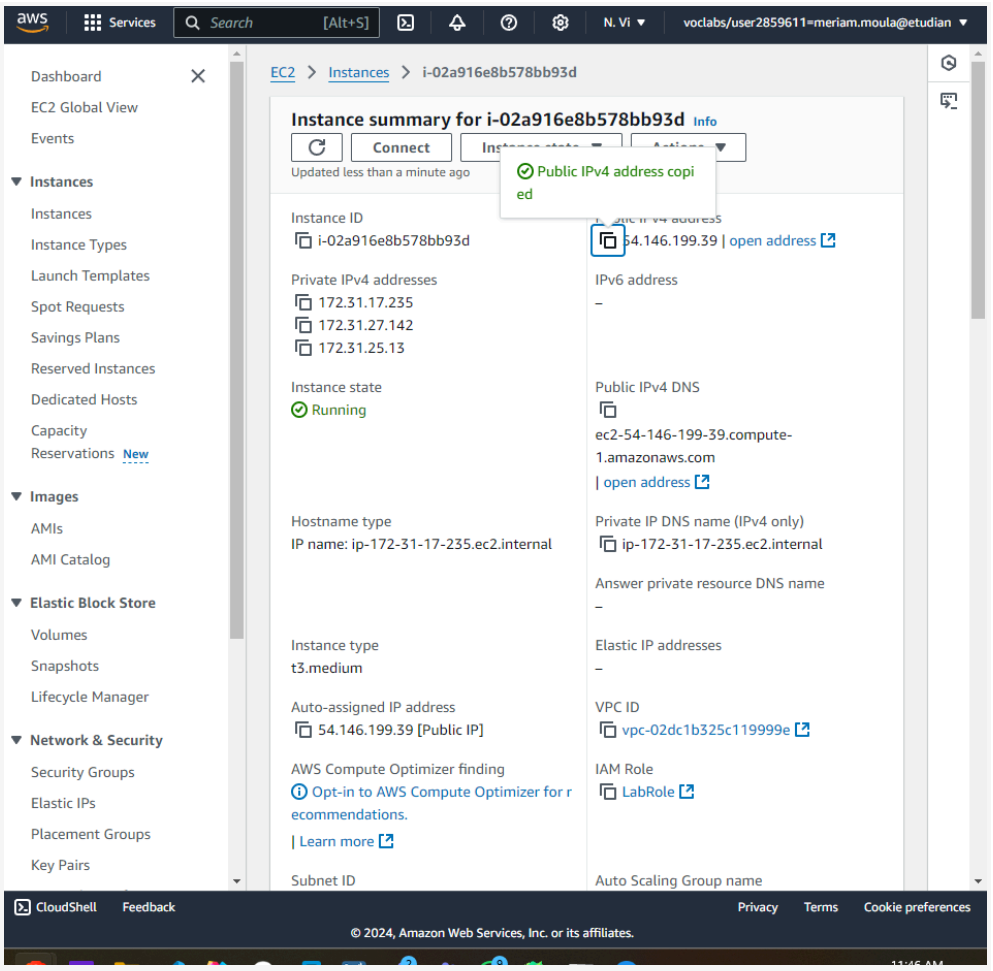




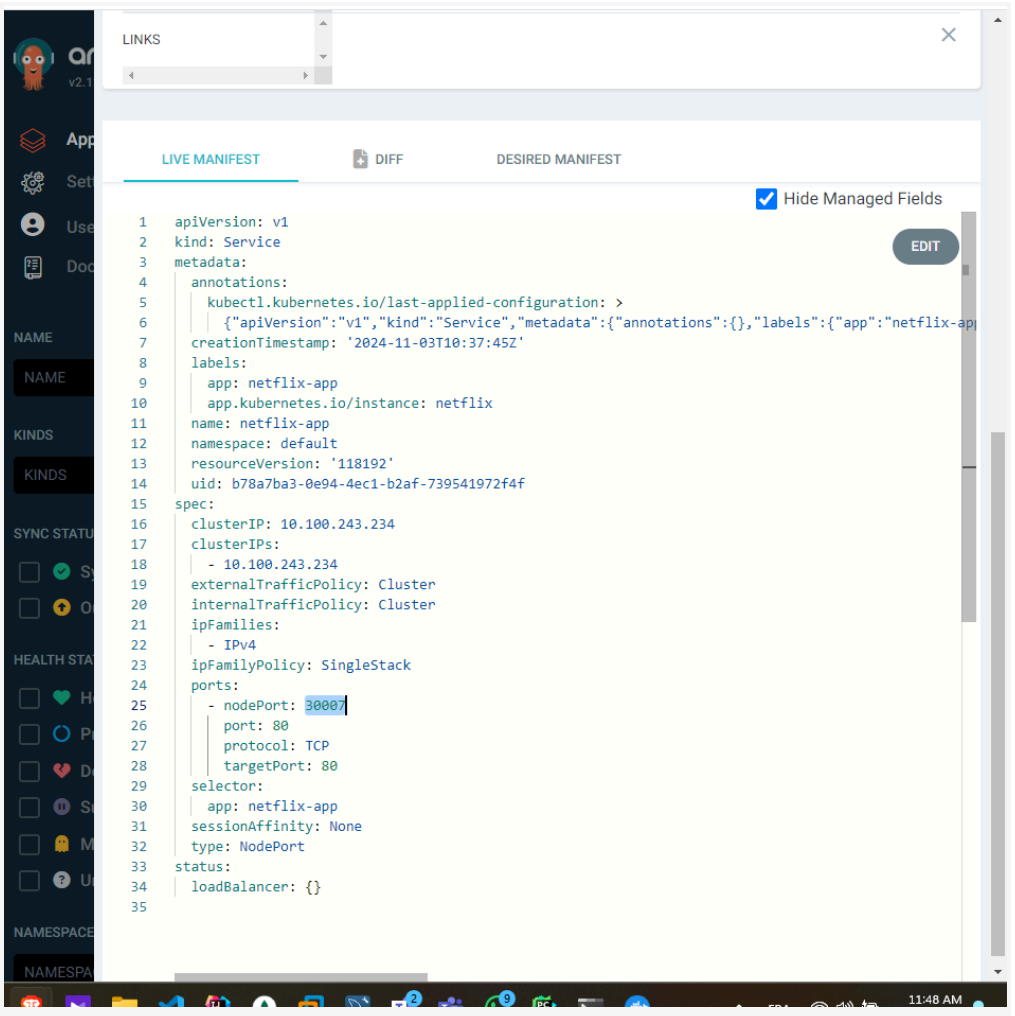


Copy the IP address of this node



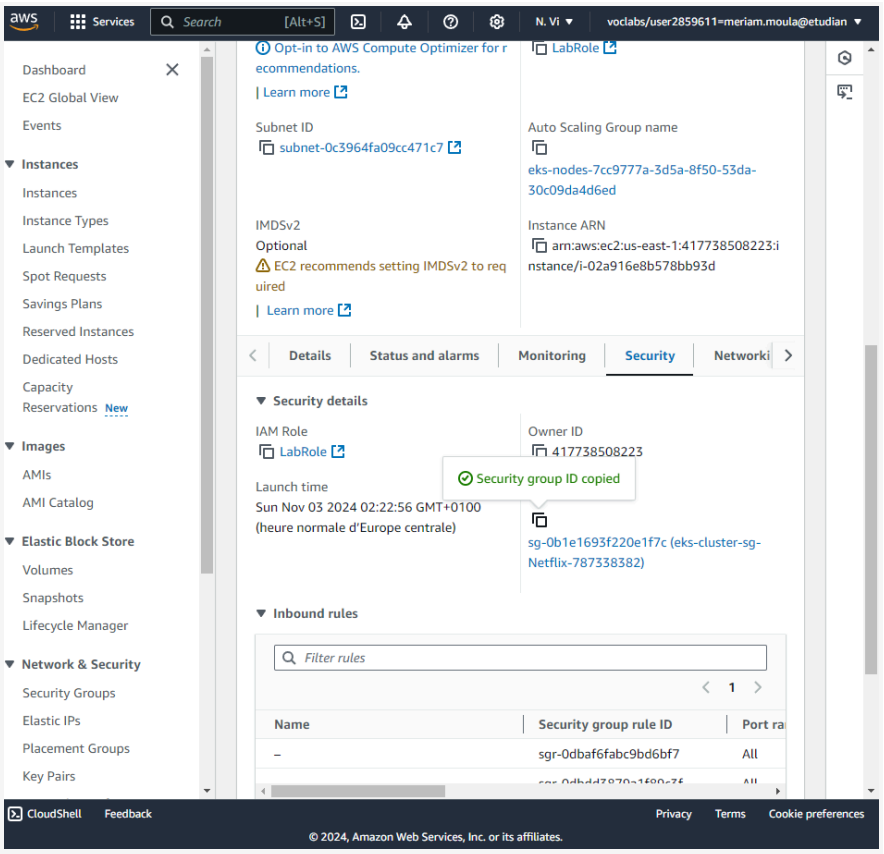


In this node , make sure that the port is open for 3007 or else we won't be able to access the application
We can see here that it is running on that port
which is mentionned in the service.yml file in the Kubernetes dir

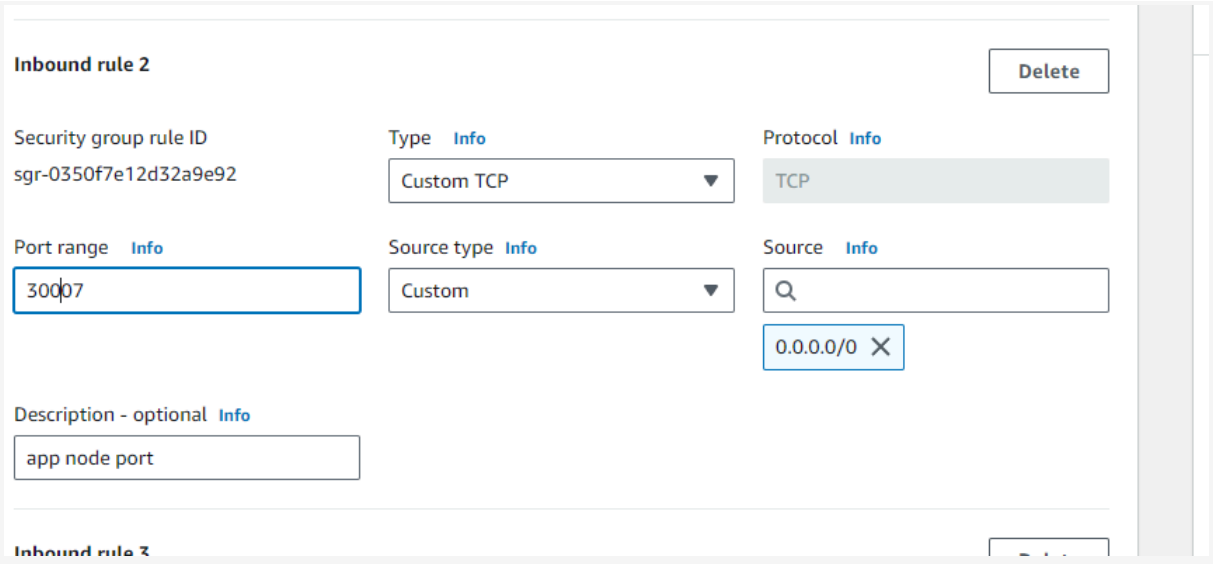


Modifying the security group:

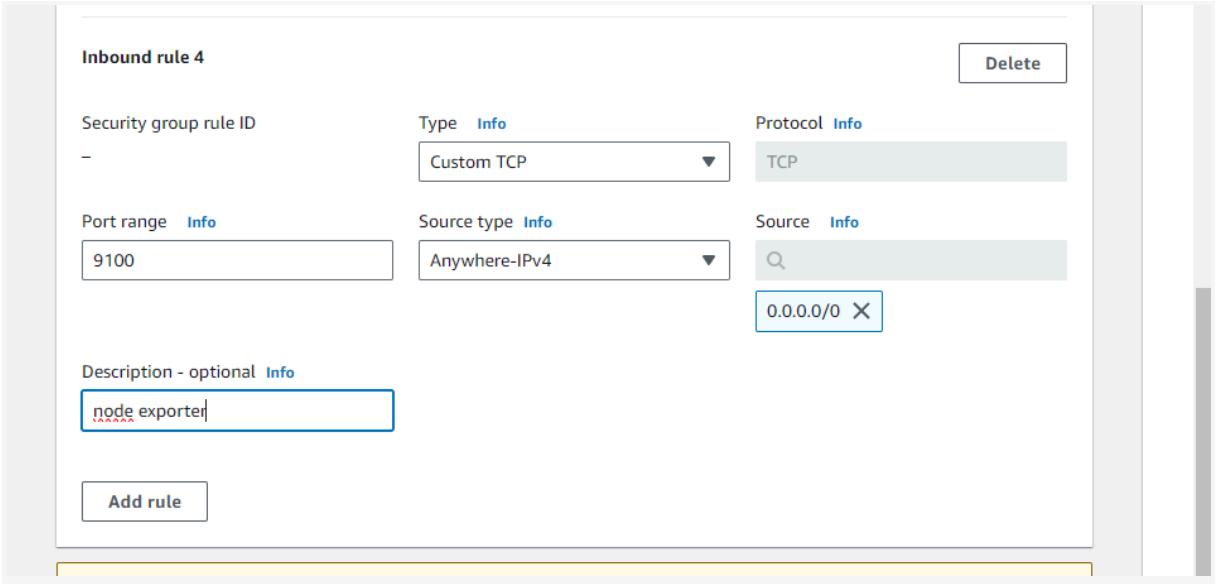
`Let 's access the node's security group to modify the rules :`


aws
Services
Search
[Alt+S]
N. Vi
voclabs/user2859611=meriam.moula@etudian
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Opt-in to AWS Compute Optimizer for recommendations.
Learn more
LabRole
Subnet ID
subnet-0c3964fa09cc471c7
Auto Scaling Group name
eks-nodes-7cc9777a-3d5a-8f50-53da-30c09da4d6ed
IMDSv2
Optional
EC2 recommends setting IMDSv2 to required
Learn more
Instance ARN
arn:aws:ec2:us-east-1:417738508223:instance/i-02a916e8b578bb93d
Details
Status and alarms
Monitoring
Security
Networki
Security details
IAM Role
LabRole
Owner ID
417738508223
Launch time
Sun Nov 03 2024 02:22:56 GMT+0100 (heure normale d'Europe centrale)
Security group ID copied
sg-0b1e1693f220e1f7c (eks-cluster-sg-Netflix-787338382)
Inbound rules
Filter rules
1
Name
Security group rule ID
Port ra
–
sgr-0dbaf6fabc9bd6bf7
All
CloudShell
Feedback
Privacy
Terms
Cookie preferences
© 2024, Amazon Web Services, Inc. or its affiliates.

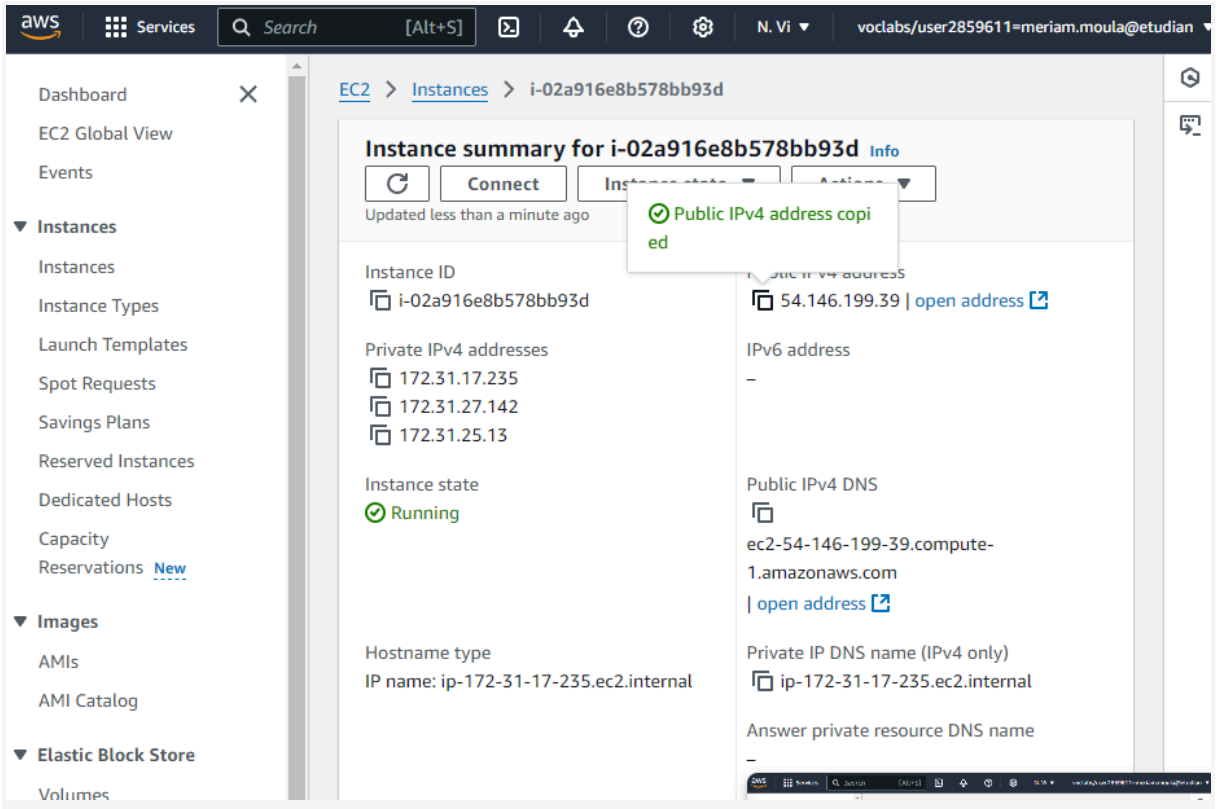
*


Inbound rule 2
Delete
Security group rule ID
sgr-0350f7e12d32a9e92
Type Info
Custom TCP
Protocol Info
TCP
Port range Info
30007
Source type Info
Custom
Source Info
0.0.0.0/0
Description - optional Info
app node port
Inbound rule 3

Add this rule as well for our node exporter

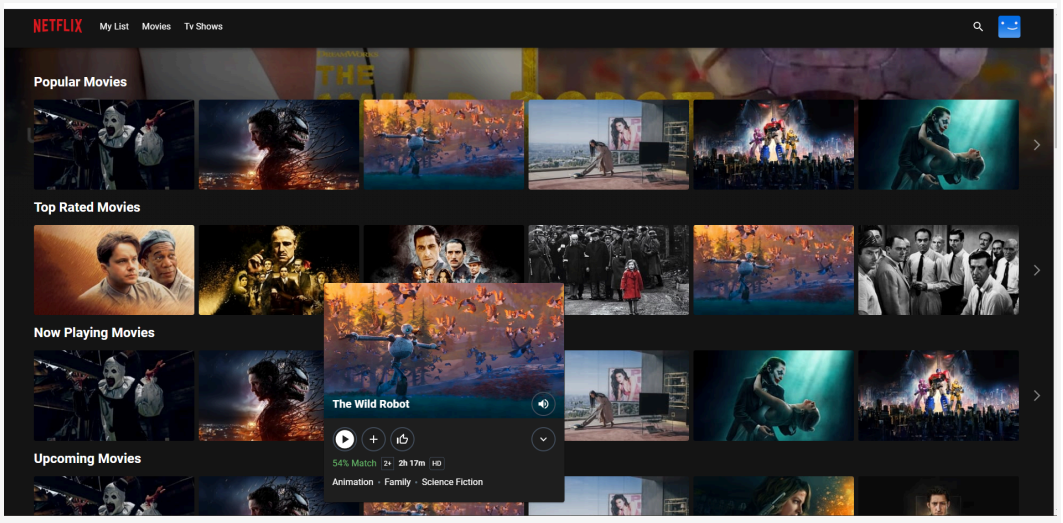


You can get the instance Ip address from here



and access the port 30007 , and it should look like this :

http://<ip-address>:30007



Install Node Exporter using Helm

Install helm on local machine

run powershell as admin
choco install kubernetes-helm
helm version

To begin monitoring your Kubernetes cluster, you'll install the Prometheus Node Exporter. This component allows you to collect system-level metrics from your cluster nodes. Here are the steps to install the Node Exporter using Helm:

Add the Prometheus Community Helm repository:

helm repo add prometheus-community
https://prometheus-community.github.io/helm-charts

Create a Kubernetes namespace for the Node Exporter:

kubectl create namespace prometheus-node-exporter

Install the Node Exporter using Helm:

helm install prometheus-node-exporter

prometheus-community/prometheus-node-exporter --namespace

prometheus-node-exporter

```
        cp
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> kubectl get ns
NAME                      STATUS   AGE
argocd                    Active   25m
default                   Active   53m
kube-node-lease           Active   53m
kube-public               Active   53m
kube-system               Active   53m
prometheus-node-exporter  Active   65s
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject>
```

```
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> kubectl get pods -n prometheus-node-exporter
NAME                                READY   STATUS    RESTARTS   AGE
prometheus-node-exporter-45prn      1/1     Running   0          3m58s
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject>
```

Updating thePrometheus configuration

To do the monitoring , we need to change the prometheus.yml file .
To that we need to run this command first on the Monitoring EC2
cd /etc/prometheus/
sudo nano prometheus.yml
In this file we need to add a job that is going to scrape metrics for our Kubernetes cluster and we need to addd the path as well :