

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 :**

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The first step, 'Name and tags', is completed with the name 'netflix-jenkins'. The second step, 'Application and OS Images (Amazon Machine Image)', is currently selected. It displays a search bar and a 'Quick Start' section with icons for various AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. A link to 'Browse more AMIs' is also present. The bottom of the screen shows the AWS navigation bar with CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

**Instance type** [Info](#) | [Get advice](#)

Instance type

**t2.large**  
 Family t2 - 2 vCPU - 8 GB 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

**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

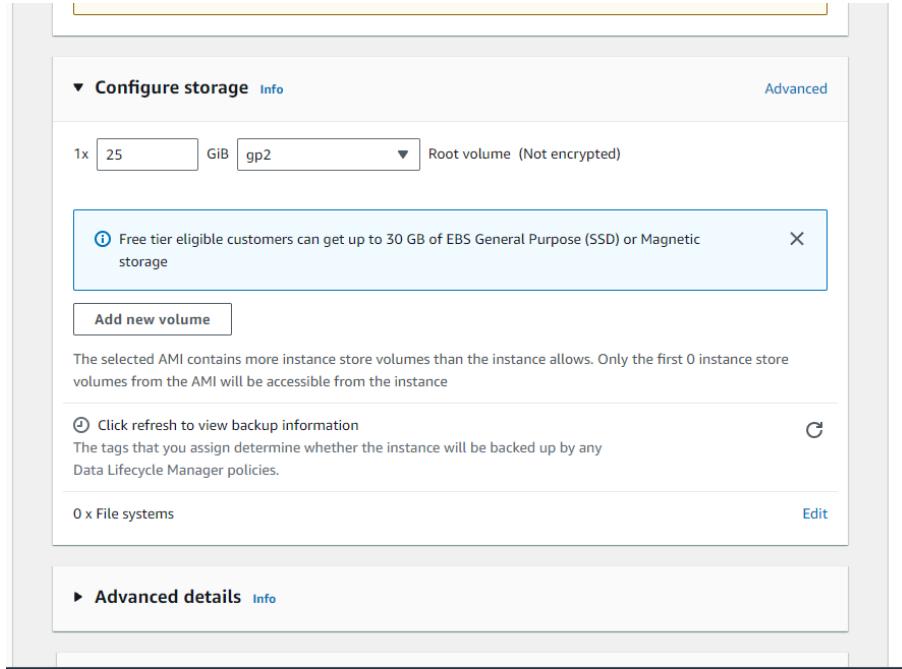
We'll create a new security group called 'launch-wizard-1' with the following rules:

<input checked="" type="checkbox"/> Allow SSH traffic from Helps you connect to your instance	Anywhere 0.0.0.0/0
<input checked="" type="checkbox"/> Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server	
<input checked="" type="checkbox"/> 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:

The screenshot shows the AWS EC2 console with the path: EC2 > Elastic IP addresses > Allocate Elastic IP address.

**Elastic IP address settings**

**Public IPv4 address pool**

- Amazon's pool of IPv4 addresses
- Public IPv4 address that you bring to your AWS account with BYOIP. (option disabled because no pools found) [Learn more](#)
- Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost. (option disabled because no customer owned pools found) [Learn more](#)
- Allocate using an IPv4 IPAM pool (option disabled because no public IPv4 IPAM pools with AWS service as EC2 were found)

**Network border group** [Info](#)

Q us-east-1 X

**Global static IP addresses**  
AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. [Learn more](#)

Create accelerator [\[ \]](#)

**Tags - optional**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

**Public IPv4 address pool**

- Amazon's pool of IPv4 addresses
- Public IPv4 address that you bring to your AWS account with BYOIP. (option disabled because no pools found) [Learn more](#)
- Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost. (option disabled because no customer owned pools found) [Learn more](#)
- Allocate using an IPv4 IPAM pool (option disabled because no public IPv4 IPAM pools with AWS service as EC2 were found)

**Network border group** [Info](#)

Q us-east-1 X

**Global static IP addresses**  
AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. [Learn more](#)

Create accelerator [\[ \]](#)

**Tags - optional**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag [\[ \]](#)  
You can add up to 50 more tag

Cancel **Allocate**

CloudShell Feedback Privacy Terms Cookie preferences  
© 2024, Amazon Web Services, Inc. or its affiliates.

## Renaming the EIP:

The screenshot shows the AWS Elastic IP Management interface. A success message at the top left states "Elastic IP address allocated successfully. Elastic IP address 34.192.55.188". Below it is a button labeled "Associate this Elastic IP address".

The main area displays "Elastic IP addresses (1/1)" with a single entry. The entry shows a Public IPv4 address of 34.192.55.188 and is associated with a name "Netflix-EIP". The status is "Allocated" and the type is "Public IP".

A tooltip message below the table says: "View IP address usage and recommendations to release unused IPs with [Public IP insights](#)".

The navigation sidebar on the left includes sections for Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, selected), Elastic IPs (Placement Groups, Key Pairs).

The bottom section shows a summary for the IP address 34.192.55.188, with tabs for Summary and Tags.

## Associating The IP address To the instance

The screenshot shows the AWS Management Console interface for associating an Elastic IP address. At the top, the navigation bar includes 'Services' and the user 'voclabs/user2859611=meriam.moula@etudiant'. Below the navigation is a breadcrumb trail: 'EC2 > Elastic IP addresses > Associate Elastic IP address'. The main title is 'Associate Elastic IP address' with a 'Info' link. A note below says 'Choose the instance or network interface to associate to this Elastic IP address (34.192.55.188)'. The 'Resource type' section has two options: 'Instance' (selected) and 'Network interface'. A warning message in a yellow box states: '⚠ 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.' It also notes that if no private IP address is specified, the Elastic IP address will be associated with the primary private IP address. The 'Instance' field contains the ID 'i-0bf97992eb88d1232'. The 'Private IP address' field is empty and labeled 'Choose a private IP address'. Under 'Reassociation', there is a checkbox 'Allow this Elastic IP address to be reassociated'. At the bottom right are 'Cancel' and 'Associate' buttons.

## 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

The screenshot shows the AWS EC2 Connect interface. At the top, there's a navigation bar with the AWS logo, 'Services' (selected), a search bar, and user information 'N. Vi'. Below the navigation is a breadcrumb trail: 'EC2 > Instances > i-0bf97992eb88d1232 > Connect to instance'. The main title is 'Connect to instance' with an 'Info' link. A message box states: 'Port 22 (SSH) is open to all IPv4 addresses. Port 22 (SSH) is currently open to all IPv4 addresses, indicated by 0.0.0.0/0 in the inbound rule in your security group. For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. [Learn more](#)'. Below this, the 'Instance ID' is listed as 'i-0bf97992eb88d1232 (netflix-jenkins)'. Under 'Connection Type', two options are shown: 'Connect using EC2 Instance Connect' (selected) and 'Connect using EC2 Instance Connect Endpoint'. The 'Public IPv4 address' field contains '34.192.55.188'. There's also an 'IPv6 address' field which is empty. In the 'Username' section, a search bar contains 'ubuntu'. A note at the bottom says: 'Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' At the bottom of the page are links for 'CloudShell', 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

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

```

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sat Nov 2 09:09:06 UTC 2024

System load: 0.0          Processes:           113
Usage of /: 6.6% of 23.17GB  Users logged in: 0
Memory usage: 2%           IPv4 address for enX0: 172.31.19.30
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-19-30:~$
```

i-0bf97992eb88d1232 (netflix-jenkins)

CloudShell Feedback Privacy Terms Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

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

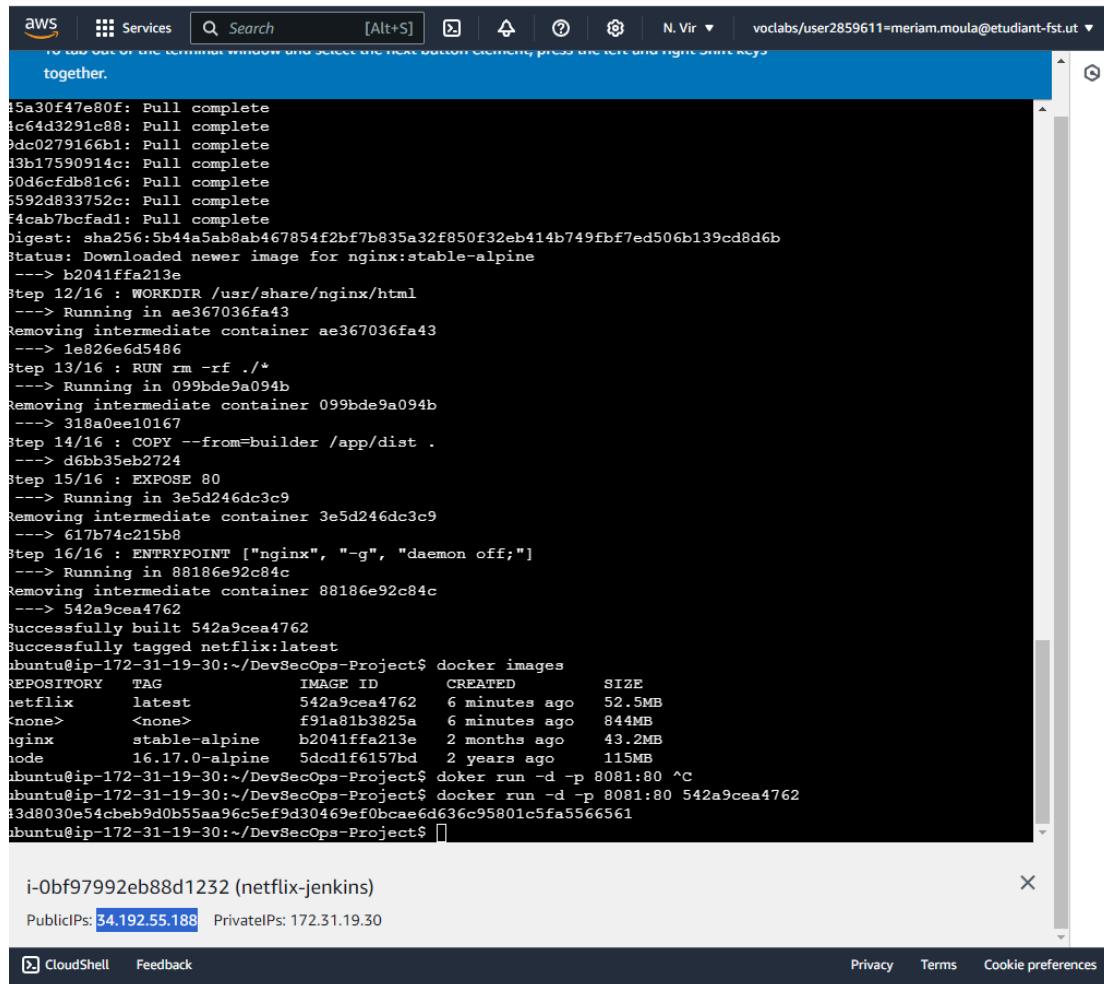
```
ubuntu@ip-172-31-19-30:~/DevSecOps-Project$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
netflix         latest       542a9cea4762   6 minutes ago  52.5MB
<none>          <none>      f91a81b3825a   6 minutes ago  844MB
nginx           stable-alpine b2041ffa213e   2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



The screenshot shows a CloudShell terminal window and a browser window. The terminal window displays the output of a Docker build command for a 'netflix' container. The browser window shows the 'Netflix Jenkins' dashboard, indicating a successful build with the Jenkins logo and the Jenkins URL '34.192.55.188:8081'.

```
5a50f47e80f: Pull complete
4c64d3291c88: Pull complete
9dc0279166b1: Pull complete
13b17590914c: Pull complete
50d6cfdb81c6: Pull complete
5592d833752c: Pull complete
f4cab7bcfad1: Pull complete
Digest: sha256:5b44a5ab8ab467854f2bf7b835a32f850f32eb414b749fb7ed506b139cd8d6b
Status: Downloaded newer image for nginx:stable-alpine
--> b2041ffa213e
Step 12/16 : WORKDIR /usr/share/nginx/html
--> Running in ae367036fa43
Removing intermediate container ae367036fa43
--> 1e826e6d5486
Step 13/16 : RUN rm -rf ./*
--> Running in 099bde9a094b
Removing intermediate container 099bde9a094b
--> 318a0ee10167
Step 14/16 : COPY --from=builder /app/dist .
--> d6bb35eb2724
Step 15/16 : EXPOSE 80
--> Running in 3e5d246dc3c9
Removing intermediate container 3e5d246dc3c9
--> 617b74c215b8
Step 16/16 : ENTRYPOINT ["nginx", "-g", "daemon off;"]
--> Running in 88186e92c84c
Removing intermediate container 88186e92c84c
--> 542a9cea4762
Successfully built 542a9cea4762
Successfully tagged netflix:latest
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  5dcdf6157bd   2 years ago   115MB
ubuntu@ip-172-31-19-30:~/DevSecOps-Project$ docker run -d -p 8081:80 ^C
ubuntu@ip-172-31-19-30:~/DevSecOps-Project$ docker run -d -p 8081:80 542a9cea4762
53d8030e54cbeb9d0b55aa96c5ef9d30469ef0bcae6d636c95801c5fa5566561
ubuntu@ip-172-31-19-30:~/DevSecOps-Project$ 
```

i-0bf97992eb88d1232 (netflix-jenkins)

PublicIPs: **34.192.55.188** PrivateIPs: 172.31.19.30

CloudShell Feedback Privacy Terms Cookie preferences

## Setting the security groups:

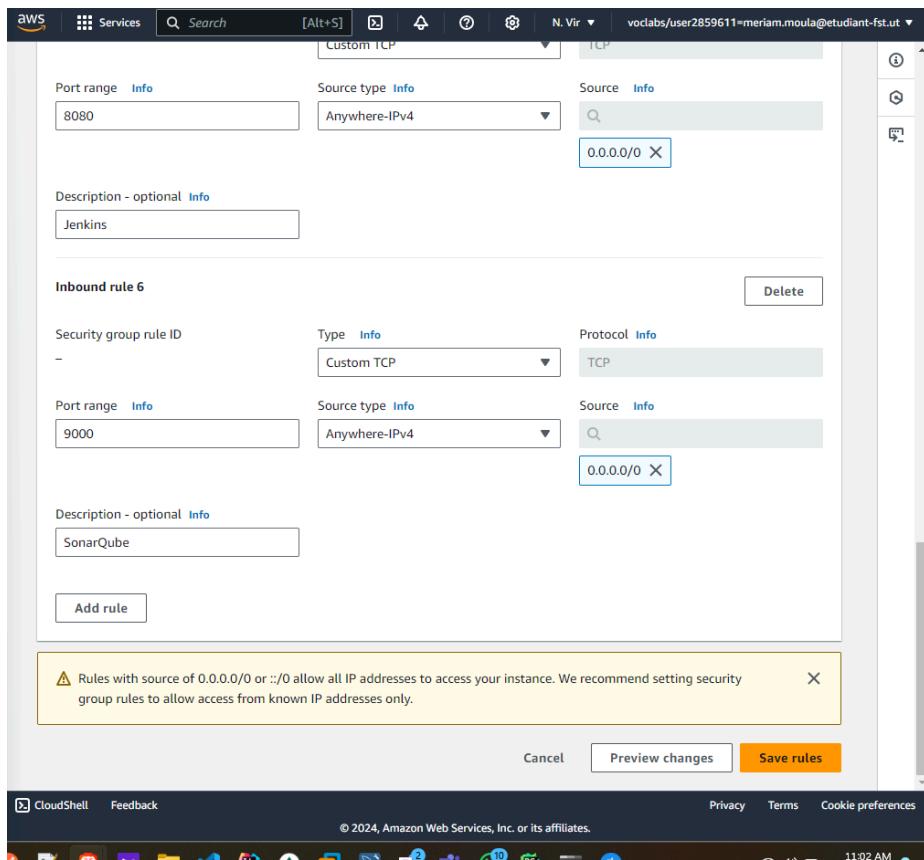
The access will only work if we have port 8081 in our security group . So in our EC2 instance , access the security group and add the rules

The screenshot shows the AWS Security Groups console for an Auto Scaling Group named 'Auto Scaling Group name'. The 'Security' tab is selected. Under 'Security details', it shows the IAM Role (empty), Owner ID (417738508223), and Launch time (Sat Nov 02 2024 09:59:19 GMT+0100). Under 'Security groups', it lists 'sg-0946447ead779d9ee (launch-wizard-1)'. Under 'Inbound rules', there are three rules: one for port 80 (TCP), one for port 22 (TCP), and one for port 443 (TCP). Under 'Outbound rules', there is a single rule for port 0 (TCP).

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

The screenshot shows the AWS Security Groups console with three new inbound rules being created:

- Inbound rule 4:** Security group rule ID: -; Type: Custom TCP; Protocol: TCP; Port range: 8081; Source type: Anywhere-IPv4; Source: 0.0.0.0/0. Description: App\_port.
- Inbound rule 5:** Security group rule ID: -; Type: Custom TCP; Protocol: TCP; Port range: 8080; Source type: Anywhere-IPv4; Source: 0.0.0.0/0. Description: Jenkins.
- Inbound rule 6:** Security group rule ID: -; Type: Custom TCP; Protocol: TCP; Port range: -; Source type: -; Source: -. Description: -.



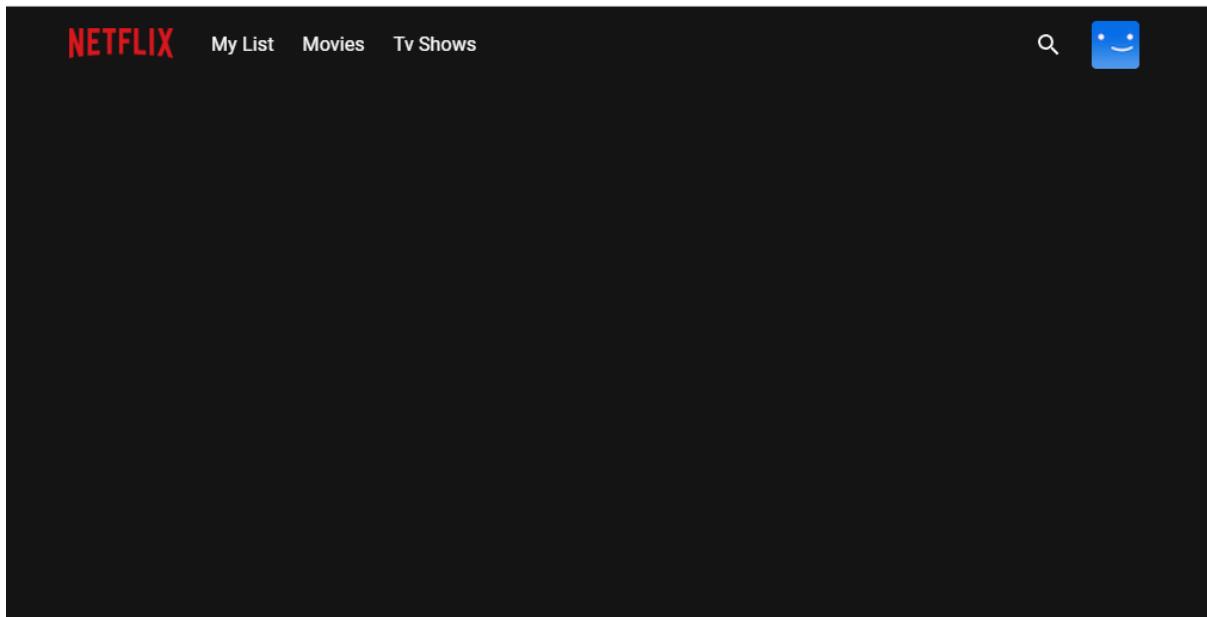
## Adding the necessary Keys and credentials :

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:16.7.0-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.

The screenshot shows the TMDB user interface. At the top, there's a navigation bar with links for 'Films', 'Émissions télévisées', 'Artistes', and 'Plus'. On the right side of the header are icons for adding a movie, switching to French ('FR'), a notification bell, a user profile ('M'), and a search icon. Below the header, the user's name 'MeriemeMoula' is displayed next to a red circular profile picture with a white letter 'M'.

The main content area has a sidebar on the left titled 'Paramètres' (Settings) with the following options:

- Modifier mon profil
- Mon compte
- Abonnements de streaming
- Notifications
- Utilisatrices et utilisateurs bloqué·e·s
- Importer une liste
- Partages
- Sessions
- API** (highlighted in red)
- Supprimer mon compte

The 'API' section contains three tabs: 'API' (selected), 'Vue d'ensemble' (Overview), and 'Créer' (Create). A note below the tabs states: "TMDB offers a powerful API service that is free to use as long as you properly attribute us as the source of the data and/or images you use. You can find the logos for attribution [here](#)."

**Documentation**: Our primary documentation is located at [developer.themoviedb.org](http://developer.themoviedb.org).

**Assistance**: If you have questions or comments about the information covered here, please create a post on our [support forums](#).

**Demandeur une clé d'API**: To generate a new API key, [click here](#).

The footer of the page features the TMDB logo and several legal links:

LES BASES	S'IMPLIQUER	COMMUNAUTÉ	MENTIONS LÉGALES
À propos de TMDB	Bible des contributio...	Règles	Conditions d'utilisation
Contactez-nous	Ajouter un film	Conversations	Conditions d'utilisation API
Forums d'assista...	Ajouter une émissio...	Top contributi...	Politique de confidentialité

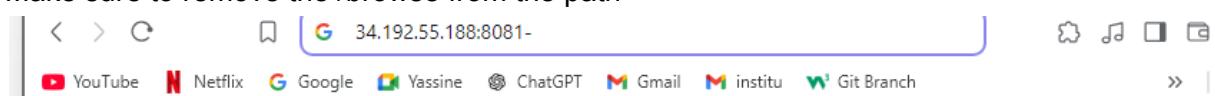
Now we run this command , If I have a server running am going to stop it and remove it , you can verify with this command :

```
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:its-community
```

Always make sure that you have the access to these ports with security groups

In case your EC2 instance is shut down you can always re-start it with this command :

```
docker start sonar
```

we can confirm that the containers are running with docker ps

```
ubuntu@ip-172-31-19-30:~/DevSecOps-Project$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5d08bd259947 sonarqube:its-community "/opt/sonarqube/dock..." About a minute ago Up About a minute 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp sonar
052109fd7a3a netflix "nginx -g 'daemon off;" 7 minutes ago Up 6 minutes 0.0.0.0:8081->80/tcp, :::8081->80/tcp pensive_engelbart
```

## 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

The screenshot shows the SonarQube interface with the following elements:

- Header: sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects...
- Main Content:
  - Text: How do you want to create your project?
  - Text: Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.
  - Text: First, you need to set up a DevOps platform configuration.
  - Buttons for creating projects from various platforms:
    - From Azure DevOps (Icon: Azure logo)
    - From Bitbucket Server (Icon: Bitbucket logo)
    - From Bitbucket Cloud (Icon: Bitbucket logo)
    - From GitHub (Icon: GitHub logo)
    - From GitLab (Icon: GitLab logo)
  - Text: Are you just testing or have an advanced use-case? Create a project manually.
  - Button: Manually (Icon: code editor icon)

A modal window at the bottom left provides information about using an embedded database:

  - Text: Embedded database should be used for...  
The embedded database will not scale, it will not support...
  - Icon: Information icon
  - Icon: Warning icon
  - Text: Get the most out of SonarQube!  
Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.
  - Buttons: Learn More, Dismiss

Footer: Community Edition - v9.0.7 (build 96285) - API v3 - Community - Documentation - Plugins - Web API

## **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 :

```
yarn.lock (yarn)
Total: 1 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 1, CRITICAL: 0)

| Library | Vulnerability | Severity | Status | Installed Version | Fixed Version | Title
| json5   | CVE-2022-46175 | HIGH     | fixed   | 2.2.1             | 2.2.2, 1.0.2 | json5: Prototype Pollution in JSON5 via Parse Method
| https://avd.aquasec.com/nvd/cve-2022-46175

ubuntu@ip-172-31-19-30:~/DevSecOps-Project$ trivy imagr netflix
Error: unknown command "imagr" for "trivy"
Usage:
  trivy [global flags] command [flags] target
  trivy [command]

Examples:
  # Scan a container image
  $ trivy image python:3.4-alpine

  # Scan a container image from a tar archive
  $ trivy image --input ruby-3.1.tar

  # Scan local filesystem
  $ trivy fs .

  # Run in server mode
  $ trivy server

Scanning Commands
config      Scan config files for misconfigurations
filesystem  Scan local filesystem
image       Scan a container image
kubernetes  [EXPERIMENTAL] Scan kubernetes cluster
repository  Scan a repository
rootfs      Scan rootfs
sbom        Scan SBOM for vulnerabilities and licenses
vm          [EXPERIMENTAL] Scan a virtual machine image

Management Commands
```

or we can use it to scan the image

```
trivy image <image-id>
```

```
Total: 4 (UNKNOWN: 0, LOW: 2, MEDIUM: 2, HIGH: 0, CRITICAL: 0)

| Library | Vulnerability | Severity | Status | Installed Version | Fixed Version | Title
|         |               |           |       |                 |             |           |
|         |               |           |       |                 |             |           |
| curl    | CVE-2024-8096 | MEDIUM   | fixed  | 8.9.1-r1        | 8.10.0-r0    | curl: OCSP
| stapling bypass with GnuTLS
| .aquasec.com/nvd/cve-2024-8096
|         |               |           |       |                 |             |           |
|         |               |           |       |                 |             |           |
| libcrypto3 | CVE-2024-9143 | LOW      |       | 3.3.2-r0        | 3.3.2-r1    | openssl: Lo
| w-level invalid GF(2^m) parameters lead to OOB
| .aquasec.com/nvd/cve-2024-9143
|         |               |           |       |                 |             |           |
|         |               |           |       |                 |             |           |
| libcurl   | CVE-2024-8096 | MEDIUM   |       | 8.9.1-r1        | 8.10.0-r0    | curl: OCSP
| stapling bypass with GnuTLS
| .aquasec.com/nvd/cve-2024-8096
|         |               |           |       |                 |             |           |
|         |               |           |       |                 |             |           |
| libss13   | CVE-2024-9143 | LOW      |       | 3.3.2-r0        | 3.3.2-r1    | openssl: Lo
| w-level invalid GF(2^m) parameters lead to OOB
| .aquasec.com/nvd/cve-2024-9143
|         |               |           |       |                 |             |           |

ubuntu@ip-172-31-19-30:~/DevSecOps-Project$
```

## **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 .

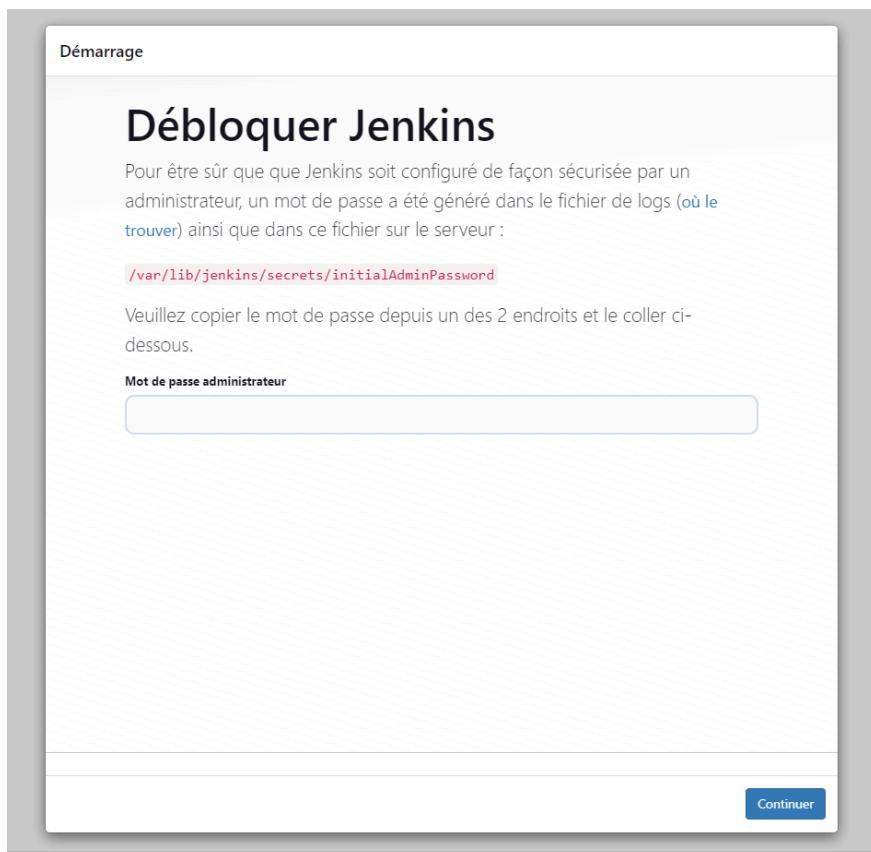
Here is the command that we will be using :

```
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/initialAdminPasswordd

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

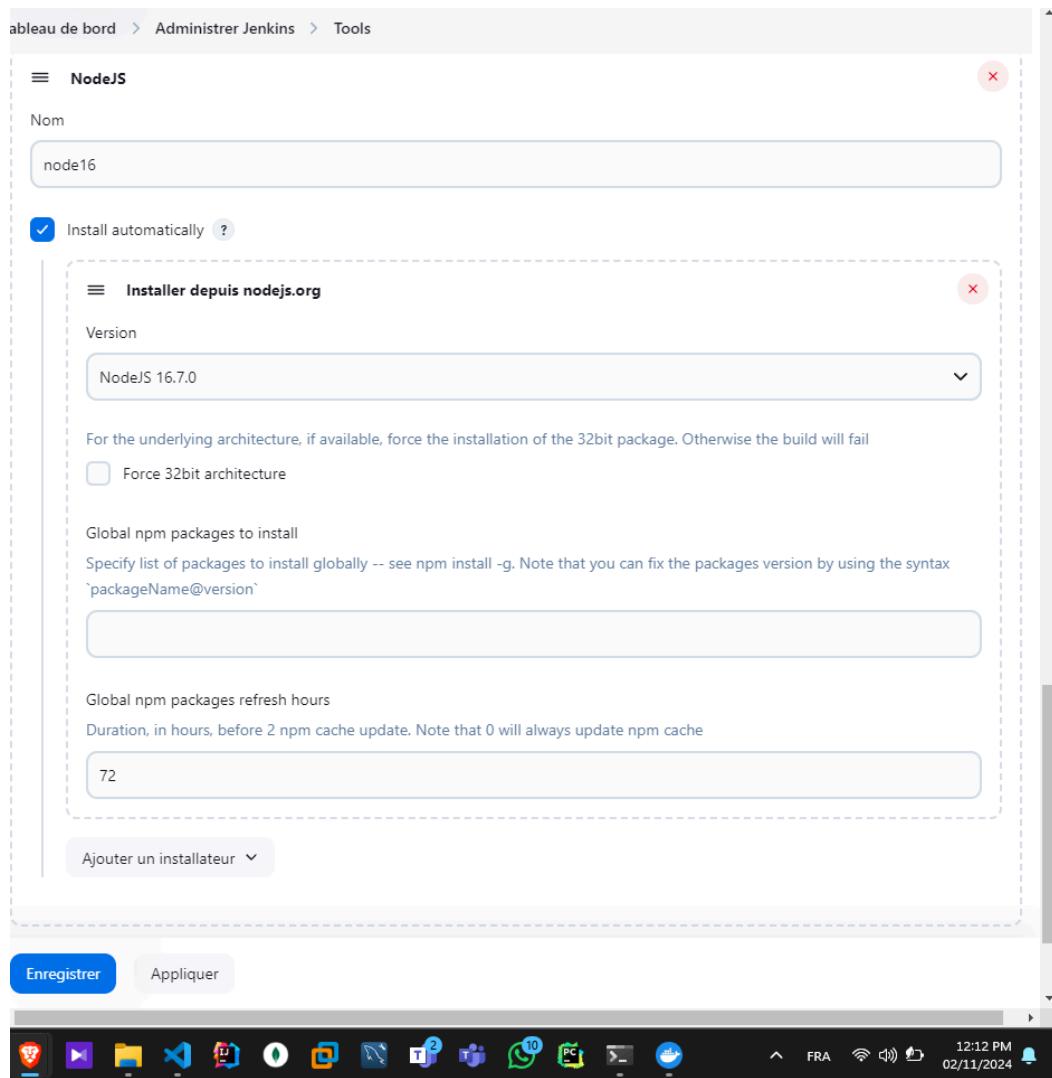
The screenshot shows the Jenkins Global Tool Configuration page. At the top, there is a breadcrumb navigation: "Tableau de bord > Administrer Jenkins > Tools". Below this, a dropdown menu says "Utiliser les réglages globaux Maven par défaut".

**Installations JDK**

A "Nom" field contains "jdk17". A checkbox "Install automatically" is checked. Under "Install from adoptium.net", a "Version" dropdown is set to "jdk-17.0.8.1+1".

**Git installations**

At the bottom, there are "Enregistrer" and "Appliquer" buttons.



## SonarQube:

### Creating and saving the token:

Create the token

GO to SonarQube

Administration->Security -> Users->Update tokens -> 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.

The screenshot shows the Jenkins 'Configuration' page for the 'netflix' project. The left sidebar has tabs for 'Général', 'Advanced Project Options' (which is selected), and 'Pipeline'. The main area is titled 'Advanced Project Options' with a 'Pipeline' tab selected under 'Definition'. A dropdown menu shows 'Pipeline script'. Below it is a code editor with Groovy syntax highlighting, containing a Jenkins Pipeline script. The script includes stages for cloning a repository, running SonarQube analysis, and performing a quality gate check. A checkbox for 'Use Groovy Sandbox' is checked. At the bottom are 'Sauvegarder' and 'Appliquer' buttons, and footer links for REST API and Jenkins version 2.479.1.

```
18 git branch: 'main', url: 'https://try.samplePipeline...'
19
20
21 }
22 }
23 stage("Sonarqube Analysis") {
24     steps {
25         withSonarQubeEnv('sonar-server') {
26             sh '''$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectKey=Netflix'''
27         }
28     }
29 }
30 stage("quality gate") {
31     steps {
32         script {
33             waitForQualityGate abortPipeline: false, credential...
34 }
```

Sauvegarder      Appliquer

REST API      Jenkins 2.479.1

## 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 -> Manually

## Create a project

All fields marked with \* are required

**Project display name \***

Up to 255 characters. Some scanners might override the value you provide.

**Project key \***

The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '\_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

**Main branch name \***

The name of your project's default branch  [Learn More](#)

**Set Up**

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 screenshot shows the SonarQube interface for the 'Netflix' project. At the top, there are navigation links: Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. Below the header, the project name 'Netflix' is displayed with a star icon and a 'main' branch indicator.

The main content area has tabs: Overview, Issues, Security Hotspots, Measures, Code, and Activity. The 'Overview' tab is selected. A message says: 'We initialized your project on SonarQube, now it's up to you to launch analyses!' Below this, step 1 'Provide a token' is completed with a green checkmark and the text 'Analyze "Netflix":sqp\_6bfcf6f0a4eb8dee656dbc51b2d09a5ecaecfdb3'. Step 2 'Run analysis on your project' is listed. Under 'What option best describes your build?', 'Maven' is selected. Under 'What is your OS?', 'Linux' is selected. A note says: 'Download and unzip the Scanner for Linux' and provides a link to the official documentation. Another note says: 'Visit the official documentation of the Scanner to download the latest version, and add the bin directory to the PATH environment variable'. A section titled 'Execute the Scanner' contains a command line example:

```
sonar-scanner \
-Dsonar.projectKey=Netflix \
-Dsonar.sources= \
-Dsonar.host.url=http://34.192.55.188:9000 \
-Dsonar.login=sqp_6bfcf6f0a4eb8dee656dbc51b2d09a5ecaecfdb3
```

A note at the bottom says: 'Please visit the official documentation of the Scanner for more details.' A modal window titled 'Get the most out of SonarQube!' is open, encouraging users to use SonarLint, a free IDE plugin. It includes a 'Learn More' button and a 'Dismiss' button.

## The page will change the build is correct

The screenshot shows the SonarQube interface for the 'Netflix' project after the analysis has been run. The 'Overview' tab is selected. The 'MEASURES' section displays various quality gate status and metrics:

- QUALITY GATE STATUS**: Passed (All conditions passed).
- MEASURES**:
  - New Code: 0 Bugs, Reliability: A
  - Overall Code: 0 Vulnerabilities, Security: A
  - Security Hotspots: 4, 0.0% Reviewed, Security Review: E
  - Debt: 1h 43min, Code Smells: 18, Maintainability: A
  - Coverage: 0.0% (Coverage on 421 Lines to cover), Unit Tests: -
  - Duplications: 0.0% (Duplications on 3.2k Lines), Duplicated Blocks: 0
- ACTIVITY**: Choose graph type (Recent, Overall, Trend, etc.).

At the bottom right, the date is November 2, 2024 at 12:53 PM.

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

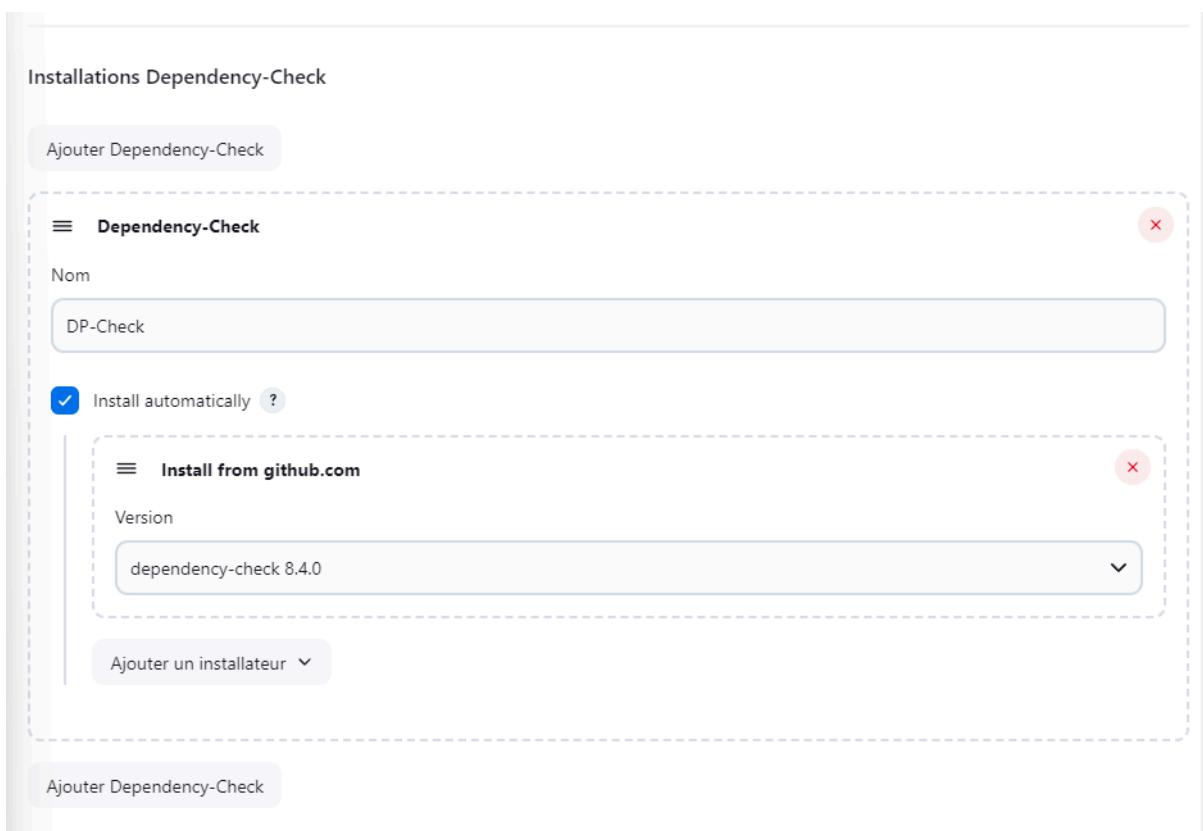
Install Dependency-Check and Docker Tools in Jenkins

### 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:

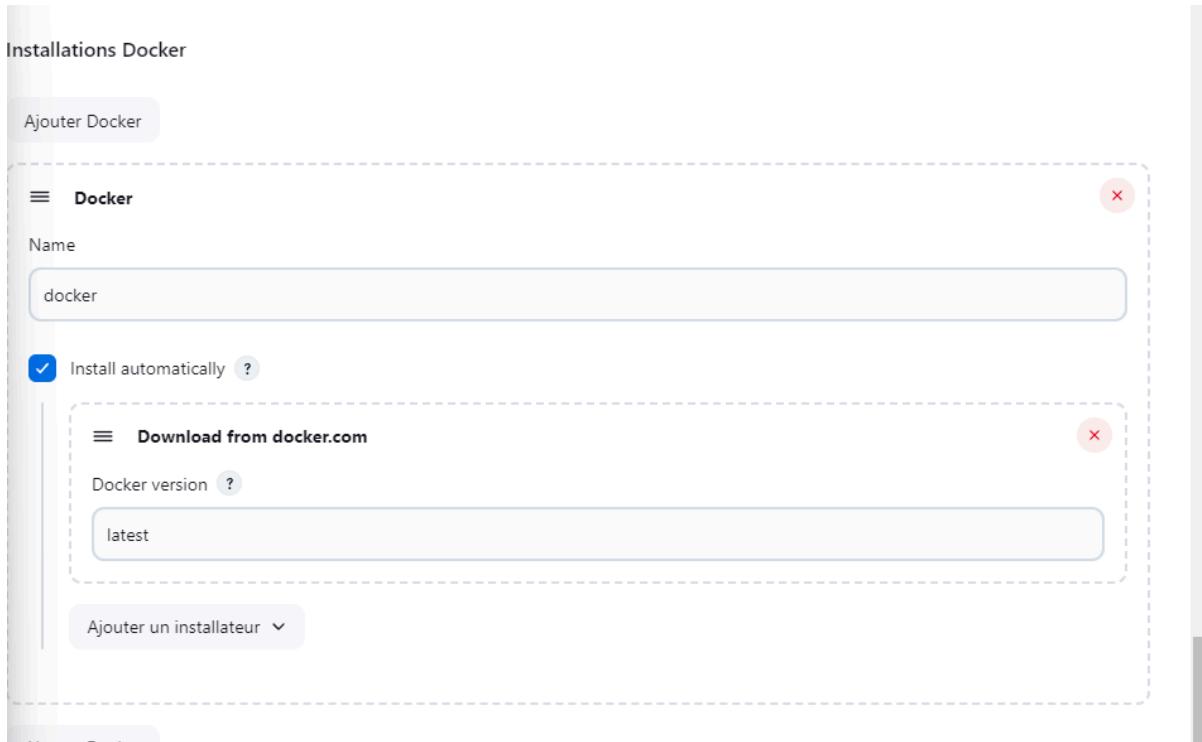
- After installing the Dependency-Check plugin, you need to configure the tool.
- Go to "Dashboard" → "Manage Jenkins" → "Global Tool Configuration."
- Find the section for "OWASP Dependency-Check."
- Add the tool's name, e.g., "DP-Check."
- Save your settings.



### 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
```

New credentials

Type

Nom d'utilisateur et mot de passe

Portée ?

Global (Jenkins, agents, items, etc...)

Nom d'utilisateur ?

mimi019

Treat username as secret ?

Mot de passe ?

\*\*\*\*\*

ID ?

docker

Description ?

docker

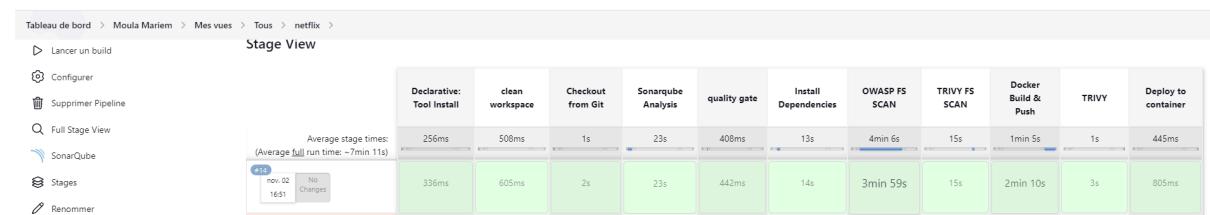
Create

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

The screenshot shows a Docker Hub profile for the user 'mimi019'. The profile page includes a blue circular icon with a white 'M', the user's name 'mimi019' with a link to 'Edit profile', a 'Community User' badge, and the date 'Joined October 6, 2024'. Below the profile, there are tabs for 'Repositories' (which is underlined), 'Starred', and 'Contributed'. A search bar shows 'Search by repository name' and displays 'Displaying 1 to 5 of 5 repositories'. The list of repositories is as follows:

- mimi019/netflix - By mimi019 • Updated 17 minutes ago
- mimi019/incomplete-cicd-01 - By mimi019 • Updated 3 days ago
- mimi019/docker-spring-boot - By mimi019 • Updated 6 days ago
- mimi019/testing-node - By mimi019 • Updated 24 days ago

## 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

Servicios | Search [Alt+S] N. Vi vocabs/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 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. X

**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 X

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 you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies. C

0 x File systems Edit

**Advanced details** Info

CloudShell Feedback Privacy Terms Cookie preferences

Screenshot of the AWS EC2 Elastic IP Addresses page showing a successful allocation and renaming of an Elastic IP address.

The main pane displays the "Elastic IP addresses (1/2)" table with one entry:

Name	Allocated IPv4 addr...	Type
Netflix-EIP	34.192.55.188	Public IP

A modal window titled "Edit Name" is open, showing the current name "220.15.226" and a new name "Monitoring-EIP". A "Save" button is visible at the bottom of the modal.

A tooltip message at the bottom of the main pane says: "View IP address usage and recommendations to release unused IPs with [Public IP insights](#)".

Screenshot of the "Associate Elastic IP address" wizard page.

**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

**Warning:** 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 reassigned with a different resource if it already associated with a resource.  
 Allow this Elastic IP address to be reassigned

Cancel Associate

## **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/

```
ubuntu@ip-172-31-27-82:~$ cd /etc/prometheus/
ubuntu@ip-172-31-27-82:/etc/prometheus$ ls
console_libraries  consoles  prometheus.yml
ubuntu@ip-172-31-27-82:/etc/prometheus$ cd /usr/local/
ubuntu@ip-172-31-27-82:/usr/local/bin$ ls
prometheus  promtool
```

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

The screenshot shows a terminal window with the following content:

```
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Sat Nov 2 16:01:14 UTC 2024

System load: 0.08      Processes:          114
Usage of /: 11.3% of 18.33GB  Users logged in:    0
Memory usage: 5%        IPv4 address for enX0: 172.31.27.82
Swap usage: 0%          IPv6 address for enX0: fe80::5e0:9ff:fe31:2782%enX0

Expanded Security Maintenance for Applications is not enabled.

41 updates can be applied immediately.
22 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Nov 2 15:51:45 2024 from 18.206.107.27
ubuntu@ip-172-31-27-82:~$ sudo nano /etc/systemd/system/prometheus.service
ubuntu@ip-172-31-27-82:~$
```

The terminal window has a blue header bar with the AWS logo, Services, Search, [Alt+S], N. Virgi, and vocabs/user2859611=meriam.moula@etudiant-fst.utn.tr. A tooltip for the keyboard shortcut "ctrl tab" is visible.

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/prometheus.service.
ubuntu@ip-172-31-27-82:~$
```

## Verify Prometheus's status:

Run : sudo systemctl status prometheus

We can see that the server is UP and running

```
[service]
ubuntu@ip-172-31-27-82:~$ sudo systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-11-02 16:04:12 UTC; 32s ago
       Main PID: 3325 (prometheus)
          Tasks: 7 (limit: 4676)
        Memory: 18.7M (peak: 19.0M)
         CPU: 76ms
      CGroup: /system.slice/prometheus.service
              └─3325 /usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=
```

Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.539Z caller=head.go:681 level=info c>
Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.539Z caller=head.go:689 level=info c>
Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.540Z caller=head.go:760 level=info c>
Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.540Z caller=head.go:797 level=info c>
Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.541Z caller=main.go:1045 level=info >
Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.541Z caller=main.go:1048 level=info >
Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.541Z caller=main.go:1229 level=info >
Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.545Z caller=main.go:1266 level=info >
Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.545Z caller=main.go:1009 level=info >
Nov 02 16:04:12 ip-172-31-27-82 prometheus[3325]: ts=2024-11-02T16:04:12.545Z caller=manager.go:1009 level=info >

[lines 1-20/20 (END)]

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

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

The screenshot shows the Prometheus configuration interface. At the top, there's a navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below the navigation, there are several configuration checkboxes: "Use local time" (unchecked), "Enable query history" (unchecked), "Enable autocomplete" (checked), "Enable highlighting" (checked), and "Enable linter" (checked). A search bar labeled "Expression (press Shift+Enter for newlines)" is present, along with a "Execute" button. Below the search bar, there are tabs for "Table" and "Graph", with "Graph" currently selected. A dropdown menu for "Evaluation time" is open, showing arrows to navigate between time points. A message "No data queried yet" is displayed in the main panel area. In the bottom right corner of the panel, there's a "Remove Panel" link. On the far left, there's a sidebar with a "Add Panel" button.

## 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

```
ubuntu@ip-172-31-27-82:~$ ls  
node_exporter-1.6.1.linux-amd64.tar.gz  prometheus-2.47.1.linux-amd64  prometheus-2.47.1.linux-amd64.tar.gz  
ubuntu@ip-172-31-27-82:~$
```

### 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*
```

```
ubuntu@ip-172-31-27-82:~$ ls /usr/local/bin  
node_exporter  prometheus  promtool
```

### 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

```
[root@jenkins ~]# cat /etc/prometheus/prometheus.yml  
ubuntu@ip-172-31-27-82:~$ cd /etc/prometheus/  
ubuntu@ip-172-31-27-82:/etc/prometheus$ cat 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"]  
ubuntu@ip-172-31-27-82:/etc/prometheus$ ^C
```

```

GNU nano 1.2                                     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"]


^G Help      ^C Write Out   ^W Where Is     ^B Cut        ^T Execute      ^C Location    M-U Undo
^X Exit      ^R Read File   ^V Replace      ^U Paste       ^J Justify      ^/ Go To Line  M-E Redo

```

i-0f03aea51c016837a (Monitoring) X

DiskUsage: 44.220.15.226 Drives: /dev/172.31.27.02

## Check the validity of the configuration file:

`promtool check config /etc/prometheus/prometheus.yml`

```

ubuntu@ip-172-31-27-82:/etc/prometheus$ promtool check config /etc/prometheus/prometheus.yml
Checking /etc/prometheus/prometheus.yml
  SUCCESS: /etc/prometheus/prometheus.yml is valid prometheus config file syntax

ubuntu@ip-172-31-27-82:/etc/prometheus$ 
```

## 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

The screenshot shows the AWS CloudFormation console interface. A new security group rule is being configured:

- Security group rule ID:** -
- Type:** Info (Custom TCP)
- Protocol:** TCP
- Port range:** Info (9100)
- Source type:** Info (Anywhere-IPv4)
- Source:** Info (0.0.0.0/0)
- Description - optional:** node-exporter

A warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules" with a close button.

Below the configuration, the Prometheus dashboard is shown with the "Targets" section. It lists one target, "node\_exporter (1/1 up)", which is healthy. The table details the endpoint, state, labels, last scrape time, scrape duration, and error status.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://44.220.15.226:9100/metrics	UP	instance="44.220.15.226:9100" job="node_exporter"	2.171s ago	12.508ms	

## 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

Inbound rule 6

Security group rule ID	Type <a href="#">Info</a> Custom TCP	Protocol <a href="#">Info</a> TCP
Port range <a href="#">Info</a>	Source type <a href="#">Info</a> Anywhere-IPv4	Source <a href="#">Info</a> 0.0.0.0/0 <a href="#">X</a>
Description - optional <a href="#">Info</a> <input type="text" value="Grafana"/>		
<a href="#">Add rule</a>		

**⚠** Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only [X](#)

## Add the GPG Key:

### Add the GPG key for Grafana:

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

```
ubuntu@ip-172-31-27-82:/etc/prometheus$ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

### 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

The screenshot shows the 'Add data source' page in Grafana. At the top, there's a search bar and a 'ctrl+k' keyboard shortcut. Below the search bar, the breadcrumb navigation shows: Home > Connections > Data sources > Add data source. On the left, there's a sidebar with a 'Time series databases' section containing options for Prometheus, Graphite, InfluxDB, and OpenTSDB. Under 'Logging & document databases', there's an option for Loki. Each data source entry includes a small icon, a name, a brief description, and a 'Core' button. The 'Prometheus' entry is highlighted with a blue background.

The screenshot shows the 'Data sources' configuration screen in Grafana. At the top, there is a search bar and a URL input field containing 'http://44.220.15.226:9090/'. Below this, the 'Connection' section is visible, followed by the 'Authentication' section. Under 'Authentication methods', it says 'Choose an authentication method to access the data source' and shows a dropdown menu set to 'No Authentication'. The 'TLS settings' section contains three checkboxes: 'Add self-signed certificate', 'TLS Client Authentication', and 'Skip TLS certificate validation'. The 'HTTP headers' section is collapsed. At the bottom, the 'Advanced settings' section is collapsed. A green success message box at the bottom right states: '✓ Successfully queried the Prometheus API. Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#)'. There are 'Delete' and 'Save & test' buttons at the bottom.

The screenshot shows the 'Dashboards' management screen in Grafana. At the top, there is a search bar and a 'New' button. A context menu is open over the 'New' button, showing options: 'New dashboard', 'Import dashboard', 'New alert rule', and 'New'. Below the search bar, there is a 'Create and manage dashboards to visualize your data' section. It includes a search input, a 'Filter by tag' dropdown, a 'Starred' checkbox, and sorting controls. The main area displays a list of dashboards, though no specific details are visible.



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.

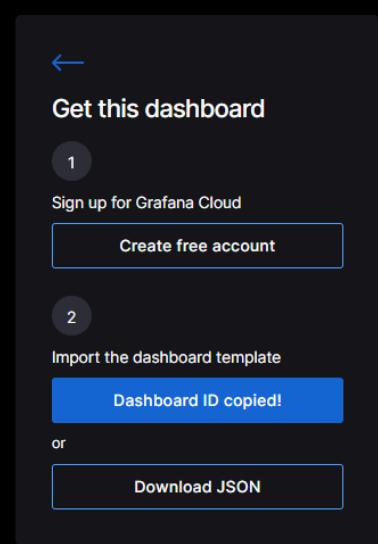
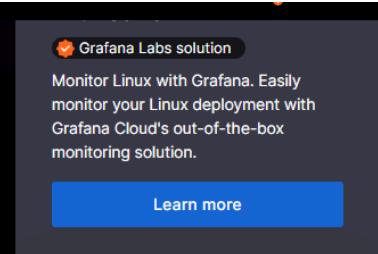


Grafana

<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.

Resources

Docs: Importing dashboards

Webinar: Getting started with Grafana

The screenshot shows the 'Import dashboard' dialog in Grafana. At the top, there's a search bar with the placeholder 'Search or jump to...' and a keyboard shortcut 'ctrl+k'. Below the search bar, the navigation path is 'Home > Dashboards > Import dashboard'. The main title 'Import dashboard' is displayed, followed by the sub-instruction 'Import dashboard from file or Grafana.com'. A dashed rectangular area contains an 'Upload dashboard JSON file' section with an upward arrow icon, a 'Drag and drop here or click to browse' instruction, and a note 'Accepted file types: .json, .txt'. Below this is a link 'Find and import dashboards for common applications at [grafana.com/dashboards](#)'. A text input field contains the ID '1860' and a blue 'Load' button. Underneath, there's a code editor for 'Import via dashboard JSON model' containing a partial JSON object:

```
{  
  "title": "Example - Repeating Dictionary variables",  
  "uid": "_0HnEoN4z",  
  "panels": [...]  
  ...  
}
```

At the bottom of the dialog are two buttons: a blue 'Load' button and a grey 'Cancel' button.

Home > Dashboards > Import dashboard

## Import dashboard

Import dashboard from file or Grafana.com

### Importing dashboard from Grafana.com

Published by	rfmoz
Updated on	2024-05-22 17:07:35

#### Options

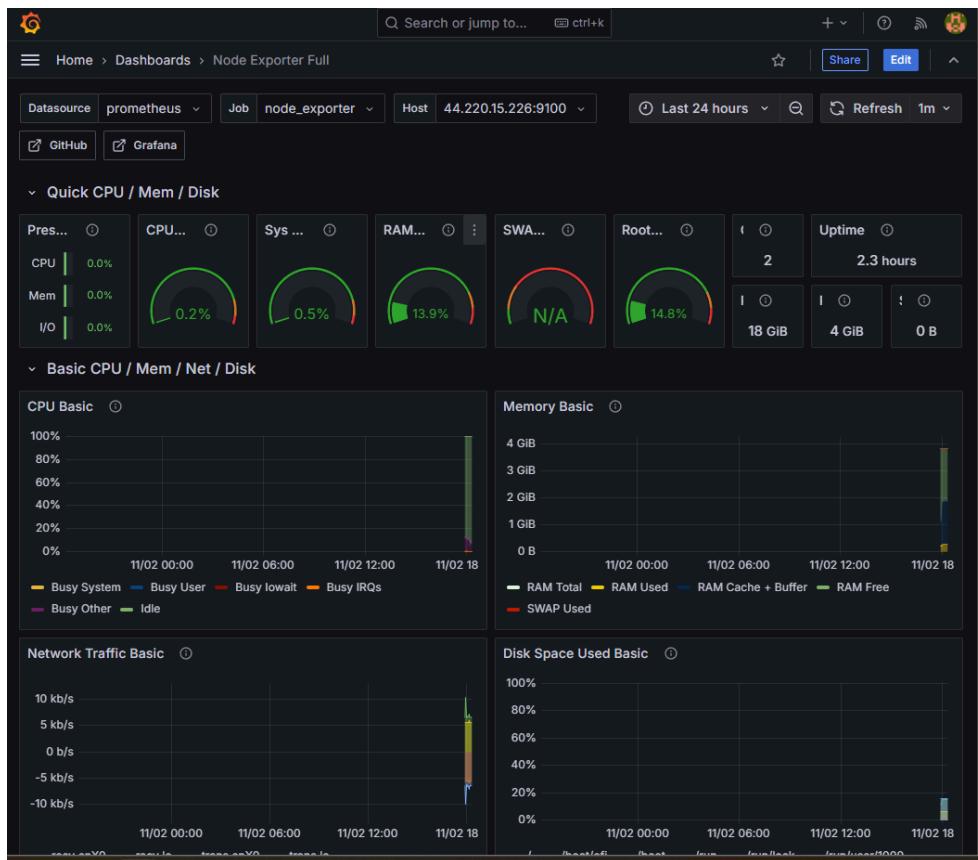
Name: Node Exporter Full

Folder: Dashboards

Unique Identifier (UID): rYddlPWk Change uid

Prometheus: prometheus

Import Cancel



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

The screenshot shows the Jenkins Plugins management interface. The left sidebar has tabs for 'Mises à jour', 'Plugins disponibles' (which is selected), 'Plugins installés', and 'Paramètres avancés'. The main area has a search bar with 'Prometheus' and a 'Filtrer' button, followed by a large 'Installer' button with a download icon. A table lists three available plugins:

Installer	Nom	Publié
<input checked="" type="checkbox"/>	Prometheus metrics 795.v995762102f28 monitoring Divers	Il y a 1 j 8 h
<input type="checkbox"/>	Cortex Metrics 1.0.1 Adds the ability to publish run results to Cortex directly using the Prometheus push endpoint.	Il y a 3 an. 7 mo.
<input type="checkbox"/>	Otel agent host metrics monitoring 1.3.0 monitoring observability	Il y a 8 h 43 mn

```
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**

The screenshot shows the Prometheus Targets page. At the top, there are navigation links: Prometheus, Alerts, Graph, Status, Help, and three icons. Below the header, there's a search bar and a filter section with checkboxes for Unknown (checked), Unhealthy (checked), and Healthy (unchecked). A table lists the target 'jenkins':

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://34.192.55.188:8080/prometheus	UP	instance="34.192.55.188:8080", job="jenkins"	5.436s ago	9.962ms	



jenkins grafana dashboard



Tous

Images

Vidéos

Actualités

Web

Livres

Finance

Outils



Grafana

https://grafana.com › dashboards

Traduire cette page



## Jenkins: Performance and Health Overview

Jobs queue speeds and rates, Executors availability, Nodes status, Jenkins and JVM resource usage, Jenkins Job Status, and lot more.



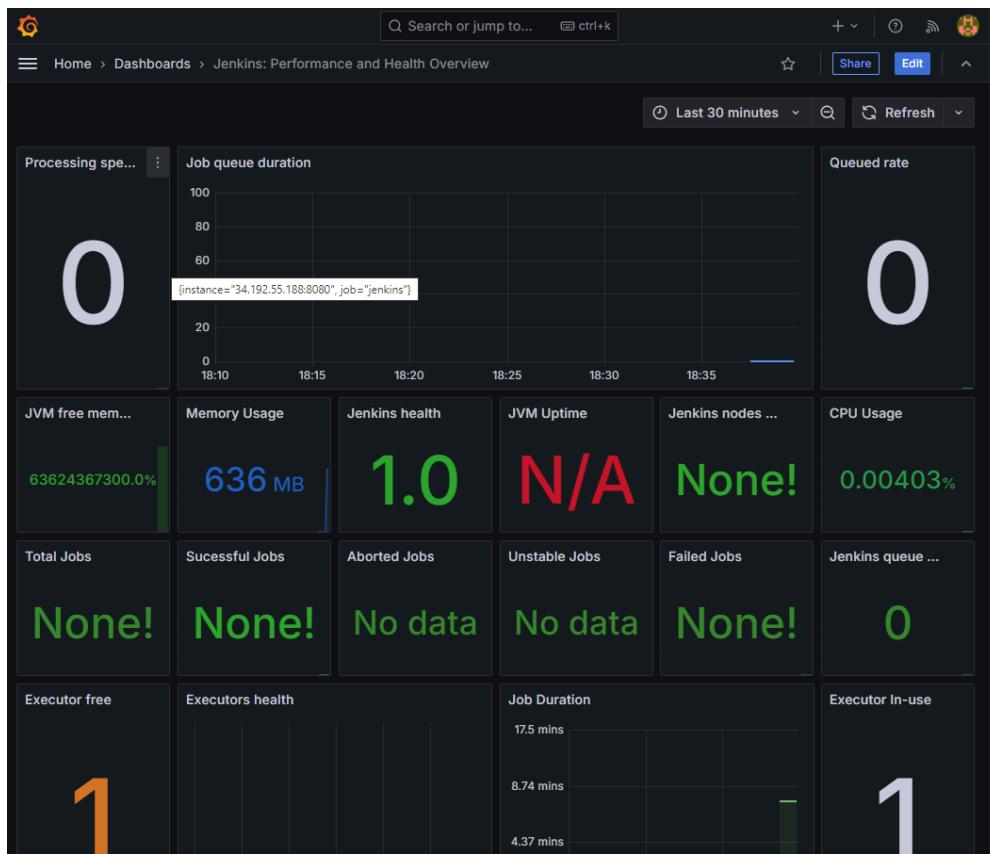
Grafana

https://grafana.com › dashboards

Traduire cette page



[Jenkins Dashboard | Grafana Labs](#)



## Enabling notifications:

You need to have a gmail account

Google Compte

Accueil Informations personnelles Données et confidentialité Sécurité Contacts et partage Paiements et abonnements À propos

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

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.meriane@gmail.com

! This field should be '@' followed by a domain name.

Avancé ▾ / Edited

Use SMTP Authentication ?

Nom d'utilisateur

moula.meriane@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

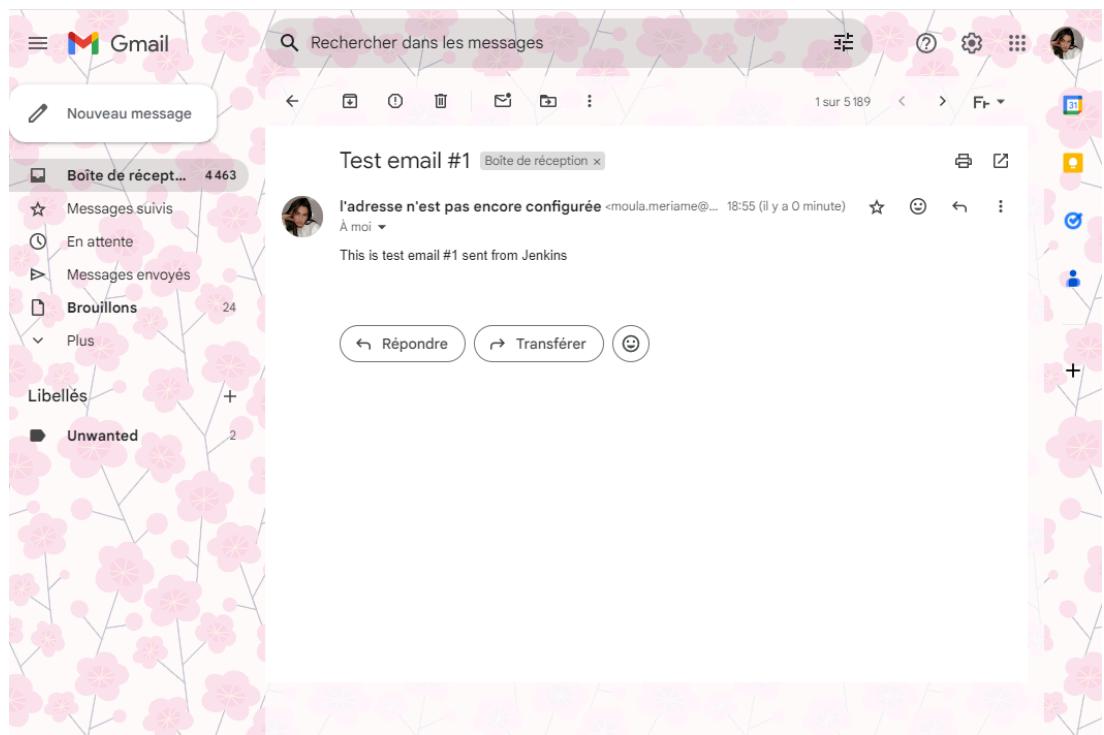
Tester la configuration en envoyant un e-mail de test

Destinataire du courriel de test

moula.meriane@gmail.com

Tester la configuration

Enregistrer Appliquer



**Make sure to add the credentials in the jenkins :**

The screenshot shows the Jenkins Global Credentials configuration page. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'Moula Mariem'. The breadcrumb path indicates the current location: Tableau de bord > Administrer Jenkins > Identifiants > System > Identifiants globaux (illimité) > New credentials.

**New credentials**

Type: Nom d'utilisateur et mot de passe

Portée: Global (Jenkins, agents, items, etc...)

Nom d'utilisateur: moula.merriame@gmail.com

Treat username as secret

Mot de passe: ..... (redacted)

ID: app-password

Description: app-password (highlighted with a blue border)

**Create**

### 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 ?

The screenshot shows the Jenkins Global Configuration page. At the top, there are four checkboxes under the heading 'Additional Behaviors': 'Enable Debug Mode', 'Require Administrator for Template Testing', 'Enable watching for jobs', and 'Allow sending to unregistered users'. Below this is a section titled 'Default Triggers' with a collapse arrow. Under 'Default Triggers', there is a list of triggers with checkboxes: 'Aborted', 'Always' (which is checked), 'Before Build', 'Failure - 1st', 'Failure - 2nd', 'Failure - Any' (which is checked), 'Failure - Still', 'Failure - X', 'Failure -> Unstable (Test Failures)', 'Fixed', 'Not Built', 'Script - After Build', 'Script - Before Build', 'Status Changed', 'Success', 'Test Improvement', and 'Test Regression'. At the bottom of the configuration area are two buttons: 'Enregistrer' (Save) and 'Appliquer' (Apply).

## 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.

Specify networking

**Networking** [Info](#)  
IP address family and service IP address range cannot be changed after cluster creation.

**VPC** [Info](#)  
Select a VPC to use for your EKS cluster resources. To create a new VPC, go to the [VPC console](#).  
vpc-02dc1b325c119999e | Default

**Subnets** [Info](#)  
Choose the subnets in your VPC where the control plane may place elastic network interfaces (ENIs) to facilitate communication with your cluster. To create a new subnet, go to the corresponding page in the [VPC console](#).

Select subnets  
 subnet-07067e6979cdedb74 X us-east-1a 172.31.32.0/20  
 subnet-09d65867ffa1dd35d X us-east-1c 172.31.80.0/20  
 subnet-016c1217cf3c58e68 X us-east-1e 172.31.48.0/20  
 subnet-0f8fd835294ad8fc9 X us-east-1f 172.31.64.0/20  
 subnet-0c3964fa09cc471c7 X us-east-1d 172.31.16.0/20  
 subnet-0d685727969827a81 X us-east-1b 172.31.0.0/20

[C](#) [Clear selected subnets](#)

**Security groups** [Info](#)  
Choose the security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your control plane subnets. To create a new security group, go to the corresponding page in the [VPC console](#).

Select security groups  
 sg-0763f371acda8894f | default X default VPC security group

[C](#) [Clear selected security groups](#)

**Choose cluster IP address family** [Info](#)  
Specify the IP address type for pods and services in your cluster.  
 IPv4  
 IPv6

**Configure Kubernetes service IP address block** [Info](#)  
 Specify the range from which cluster services will receive IP addresses.

**Cluster endpoint access** [Info](#)  
Configure access to the Kubernetes API server endpoint.

**Public**  
The cluster endpoint is accessible from outside of your VPC. Worker node traffic will leave your VPC to connect to the endpoint.

**Public and private**  
The cluster endpoint is accessible from outside of your VPC. Worker node traffic to the endpoint will stay within your VPC.

**Private**  
The cluster endpoint is only accessible through your VPC. Worker node traffic to the endpoint will stay within your VPC.

**Advanced settings**

[Cancel](#) [Previous](#) [Next](#)

[CloudShell](#) [Feedback](#) [Privacy](#) [Terms](#) [Cookie preferences](#)

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

Screenshot of the AWS EKS console showing the Compute tab for a cluster.

**Cluster Overview:**

- Status: Active
- Kubernetes version: 1.31
- Support period: Standard support until November 26, 2025
- Provider: EKS

**Compute Tab:**

- Overview
- Resources
- Compute** (selected)
- Networking
- Add-ons
- Access

**Nodes (0) Info:**

No Nodes  
This cluster does not have any Nodes, or you don't have permission to view them.

**Node groups (0) Info:**

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.

Success message: 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: **Amazon Linux 2 (AL2\_x86\_64)**

Capacity type: **On-Demand**

Instance types: **t3.medium**  
vCPU: 2 vCPUs | Memory: 4 GiB | Network: Up to 5 Gigabit | Max ENI: 3 | Max IPs: 18

Disk size: **20 GiB**

#### Node group scaling configuration

**Node group scaling configuration**

**Desired size**  
Set the desired number of nodes that the group should launch with initially.  
 nodes  
Desired node size must be greater than or equal to 0

**Minimum size**  
Set the minimum number of nodes that the group can scale in to.  
 nodes  
Minimum node size must be greater than or equal to 0

**Maximum size**  
Set the maximum number of nodes that the group can scale out to.  
 nodes  
Maximum node size must be greater than or equal to 1 and cannot be lower than the minimum size

**Node group update configuration** Info

**Maximum unavailable**  
Set the maximum number or percentage of unavailable nodes to be tolerated during the node group version update.

Number  
Enter a number

Percentage  
Specify a percentage

**Value**  
 node  
Node count must be greater than 0

## 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`

```
unknown options: --region,us-east-1
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/](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
```

PS C:\Users\myria\OneDrive\Desktop\IGL5\NetFlixDevOpsProject> kubectl get all -n argocd				
NAME	READY	STATUS	RESTARTS	AGE
pod/argocd-application-controller-0	1/1	Running	0	11m
pod/argocd-applicationset-controller-d4df5c8f8-4mlld	1/1	Running	0	11m
pod/argocd-dex-server-5449fdc856-m9mvs	1/1	Running	2 (11m ago)	11m
pod/argocd-notifications-controller-c75bf9cb6-ppkr4	1/1	Running	0	11m
pod/argocd-redis-6d5ddc7cb-tftw6	1/1	Running	0	11m
pod/argocd-repo-server-cd9d97db-78vtm	1/1	Running	0	11m
pod/argocd-server-7755999557-pxnfr	1/1	Running	0	11m
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
service/argocd-applicationset-controller	ClusterIP	10.100.93.204	<none>	7000/TCP, 8080/TCP
service/argocd-dex-server	ClusterIP	10.100.244.87	<none>	5556/TCP, 5555/TCP
service/argocd-metrics	ClusterIP	10.100.78.47	<none>	8082/TCP
service/argocd-notifications-controller-metrics	ClusterIP	10.100.80.72	<none>	9001/TCP
service/argocd-redis	ClusterIP	10.100.152.134	<none>	6379/TCP
service/argocd-repo-server	ClusterIP	10.100.16.84	<none>	8081/TCP, 8080/TCP
service/argocd-server	ClusterIP	10.100.208.190	<none>	80/TCP, 443/TCP
service/argocd-server-metrics	ClusterIP	10.100.89.77	<none>	8083/TCP

## 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
```

PS C:\Users\myria\OneDrive\Desktop\IGL5\NetFlixDevOpsProject> kubectl patch svc argocd-server -n argocd --type='json' -p '[{"op": "replace", "path": "/spec/type", "value": "LoadBalancer"}]'				
service/argocd-server patched				
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetFlixDevOpsProject> kubectl get svc argocd-server -n argocd				
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
argocd-server	LoadBalancer	10.100.208.190	a0dc74d3da31f4c588a2368dd0899ba5-1661715903.us-east-1.elb.amazonaws.com	80:31525/TCP, 443:30656/TCP

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.

The screenshot shows the AWS EC2 Load Balancers console. On the left, there's a navigation sidebar with various services like Instance Types, Launch Templates, and Capacity Reservations. The main area is titled 'Load balancers' and shows a table with one item. The table columns are Name, DNS name, State, and VPC ID. The single row shows 'a0dc74d3da31f4c588a2368dd0899ba5' as the Name, with a corresponding DNS name 'a0dc74d3da31f4c588a2368dd0899ba5.elb.amazonaws.com'. The State is listed as '-' and the VPC ID is 'vpc-02dc1b325c119999e'. Below the table, there's a detailed view for the selected load balancer, showing its type (Classic), scheme (Internet-facing), and VPC information.

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

This command does the following:

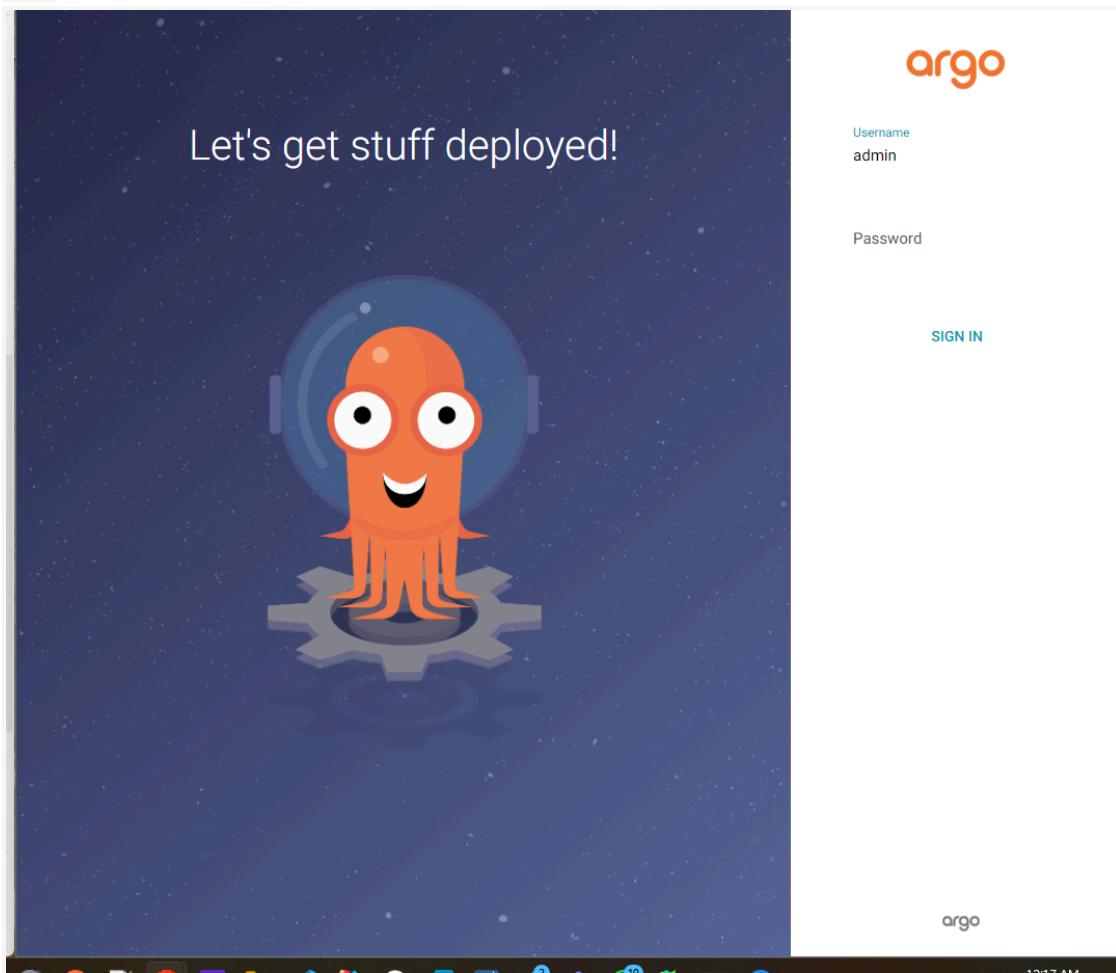
1. Runs `kubectl get svc argoctl-server -n argoctl -o json` to retrieve the service information in JSON format.
2. Pipes the JSON output to `ConvertFrom-Json`, which allows PowerShell to work with the JSON data as an object.
3. Accesses the nested property `.status.loadBalancer.ingress[0].hostname` and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

```
PS C:\Users\myria\OneDrive\Desktop\IGL5\NetflixDevOpsProject> $ARGOCD_SERVER = (kubectl get svc argoctl-server -n argoctl -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]::FromBase64S  
tring((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.

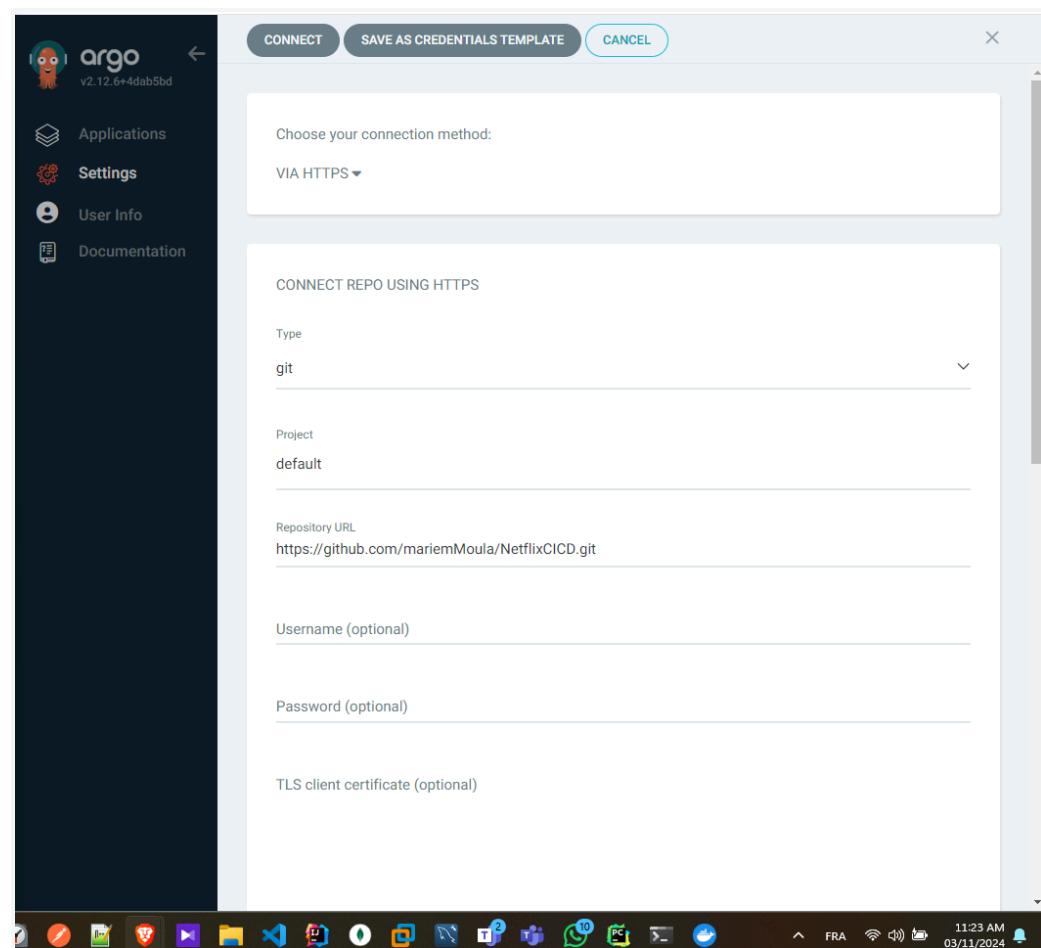
2. `[System.Convert]::FromBase64String(...)` decodes the base64 string.
3. `[System.Text.Encoding]::UTF8.GetString(...)` converts the decoded bytes to a UTF-8 string, making it human-readable.

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 argo cd get secret argo cd-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

The screenshot shows the Argo UI interface. On the left is a sidebar with icons for Applications, Settings (selected), User Info, and Documentation. The main area is titled "Settings / Repositories" and shows a table of connected repositories. The table has columns: TYPE, NAME, PROJECT, REPOSITORY, and CONNECTION STATUS. There is one entry: "git" under "TYPE", "git" under "NAME", "default" under "PROJECT", "https://github.com/mariemMou..." under "REPOSITORY", and "Successful" with a green checkmark under "CONNECTION STATUS". A "Log out" button is in the top right corner.

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

The screenshot shows the Argo UI interface with the "CREATE" button highlighted. The "GENERAL" tab is selected. In the "Application Name" field, "netflix" is entered. In the "Project Name" field, "default" is selected. Under "SYNC POLICY", "Automatic" is chosen. Under "SYNC OPTIONS", several checkboxes are available: "PRUNE RESOURCES", "SELF HEAL", "SET DELETION FINALIZER", "SKIP SCHEMA VALIDATION", "PRUNE LAST", "RESPECT IGNORE DIFFERENCES", "AUTO-CREATE NAMESPACE", "APPLY OUT OF SYNC ONLY", and "SERVER-SIDE APPLY". Under "PRUNE PROPAGATION POLICY", "foreground" is selected. At the bottom, there are checkboxes for "REPLACE" and "RETRY".

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
prometheus-node-exporter Active  11h
```

The screenshot shows the Argo UI interface. On the left, there's a sidebar with navigation links: Applications (selected), Settings, User Info, and Documentation. The main area is titled 'CREATE' and contains fields for Repository URL (https://github.com/mariemMoula/NetflixCICD.git), Revision (HEAD), Path (Kubernetes), and Destination (Cluster URL: https://kubernetes.default.svc, Namespace: default). Below these, there's a 'DIRECTORY' section with a 'DIRECTORY RECURSE' checkbox. At the bottom right of the main area, there are 'SYNC', 'REFRESH', and 'DELETE' buttons.

**DESTINATION**

Cluster URL: <https://kubernetes.default.svc> URL ▾

Namespace: default

Directory ▾

**DIRECTORY**

DIRECTORY RECURSE

**Applications**

+ NEW APP ⚡ SYNC APPS ⚡ REFRESH APPS ⚡ Search applications... ⚡

Sort: name ▾ Items per page: 10 ▾

**APPLICATIONS TILES**

Project	Status	Synced	Last Sync
netflix	default	Healthy	2 minutes ago

**Favorites Only**

**SYNC STATUS**

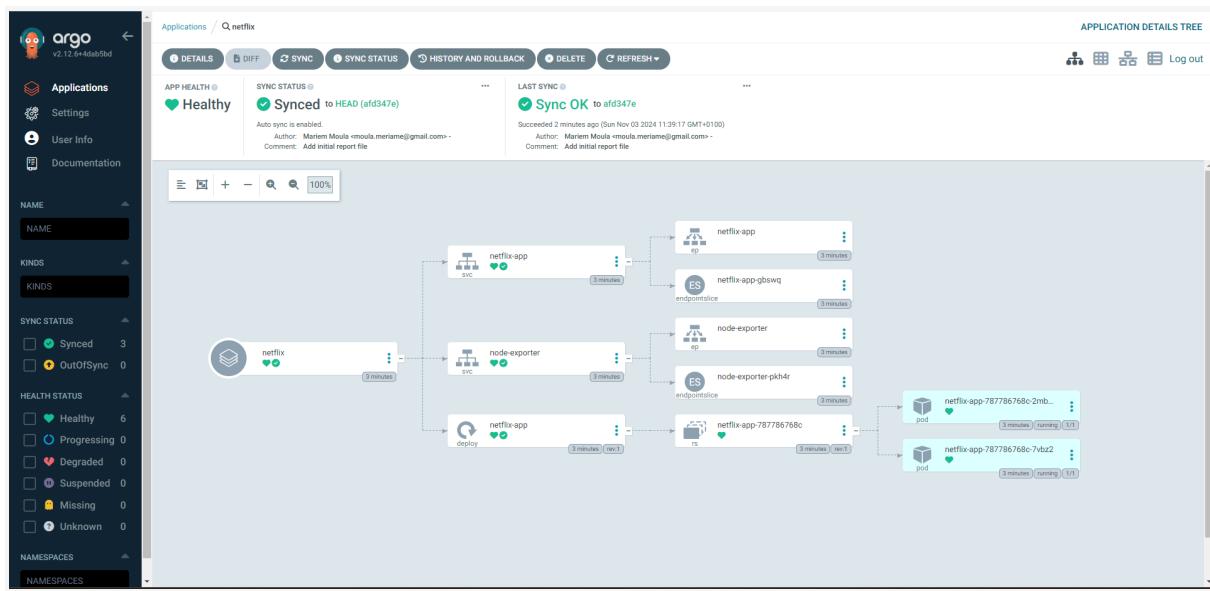
- Unknown: 0
- Synced: 1
- OutOfSync: 0

**HEALTH STATUS**

- Unknown: 0
- Progressing: 0
- Suspended: 0
- Healthy: 1
- Degraded: 0
- Missing: 0

**LABELS**

**PROJECTS**



Copy the IP address of this node

**ip-172-31-17-235.ec2.internal**

**Structured view** | **Raw view**

Details	
Status	Kernel version
Ready	5.10.226-214.880.amzn2.x86_64
Last transition time	Created
9 hours ago	9 hours ago
Node group	Container runtime
nodes	containerd://1.7.22
OS (Architecture)	Kubelet version
linux (amd64)	v1.31.0-eks-a737599
OS image	Instance
Amazon Linux 2	i-02a916e8b578bb93d
	Instance type
	t3.medium

Capacity allocation	
<b>Cores</b>	<b>Memory</b>
2 Cores	3.76 Memory
System reserved 70 m, 4%	System reserved 0.53 GiB, 14%
Workload... 350 m, 1...	Workload... 0.14 Gi...

**CloudShell** **Feedback** **Privacy** **Terms** **Cookie preferences**

© 2024, Amazon Web Services, Inc. or its affiliates.

The screenshot shows the AWS EC2 Instances page. A context menu is open over the instance summary for 'i-02a916e8b578bb93d'. The menu item 'Public IPv4 address copied' is highlighted in green. The main content area displays the following details:

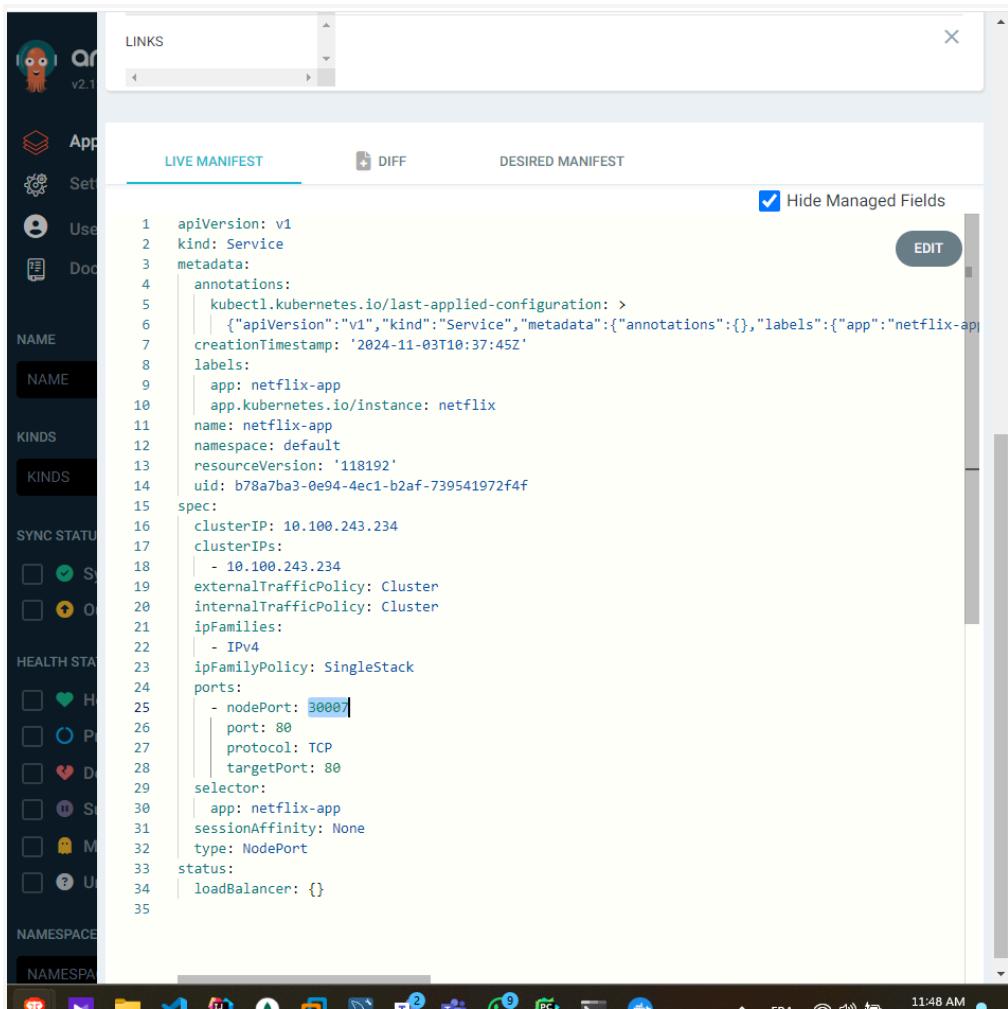
- Instance ID:** i-02a916e8b578bb93d
- Private IPv4 addresses:** 172.31.17.235, 172.31.27.142, 172.31.25.13
- Instance state:** Running
- Hostname type:** IP name: ip-172-31-17-235.ec2.internal
- Instance type:** t3.medium
- Auto-assigned IP address:** 54.146.199.39 [Public IP]
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations.
- Subnet ID:**
- Auto Scaling Group name:**
- Public IPv4 DNS:** ec2-54-146-199-39.compute-1.amazonaws.com
- Private IP DNS name (IPv4 only):** ip-172-31-17-235.ec2.internal
- Answer private resource DNS name:**
- Elastic IP addresses:**
- VPC ID:** vpc-02dc1b325c119999e
- IAM Role:** LabRole

At the bottom of the page, there are links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences. The footer indicates © 2024, Amazon Web Services, Inc. or its affiliates.

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



```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   annotations:
5     kubectl.kubernetes.io/last-applied-configuration: >
6       {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"netflix-app","name":"netflix-app"}}, "creationTimestamp": "2024-11-03T10:37:45Z"
7   creationTimestamp: '2024-11-03T10:37:45Z'
8   labels:
9     app: netflix-app
10    app.kubernetes.io/instance: netflix
11   name: netflix-app
12   namespace: default
13   resourceVersion: '118192'
14   uid: b78a7ba3-0e94-4ec1-b2af-739541972f4f
15 spec:
16   clusterIP: 10.100.243.234
17   clusterIPs:
18     - 10.100.243.234
19   externalTrafficPolicy: Cluster
20   internalTrafficPolicy: Cluster
21   ipFamilies:
22     - IPv4
23   ipFamilyPolicy: SingleStack
24   ports:
25     - nodePort: 30007
26       port: 80
27       protocol: TCP
28       targetPort: 80
29   selector:
30     app: netflix-app
31   sessionAffinity: None
32   type: NodePort
33   status:
34     loadBalancer: {}
35
```

## Modifying the security group:

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

Screenshot of the AWS Management Console showing the EC2 Instances page. A tooltip indicates that the Security group ID has been copied.

Subnet ID: subnet-0c3964fa09cc471c7

Auto Scaling Group name: eks-nodes-7cc9777a-3d5a-8f50-53da-30c09da4d6ed

IMDSv2: Optional. EC2 recommends setting IMDSv2 to required.

Instance ARN: arm:aws:ec2:us-east-1:417738508223:instance/i-02a916e8b578bb93d

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 rule 2

Delete

Security group rule ID: sgr-0350f7e12d32a9e92

Type: Custom TCP

Protocol: TCP

Port range: 30007

Source type: Custom

Source: 0.0.0.0/0

Description - optional: app node port

Inbound rule 3

Add this rule as well for our node exporter

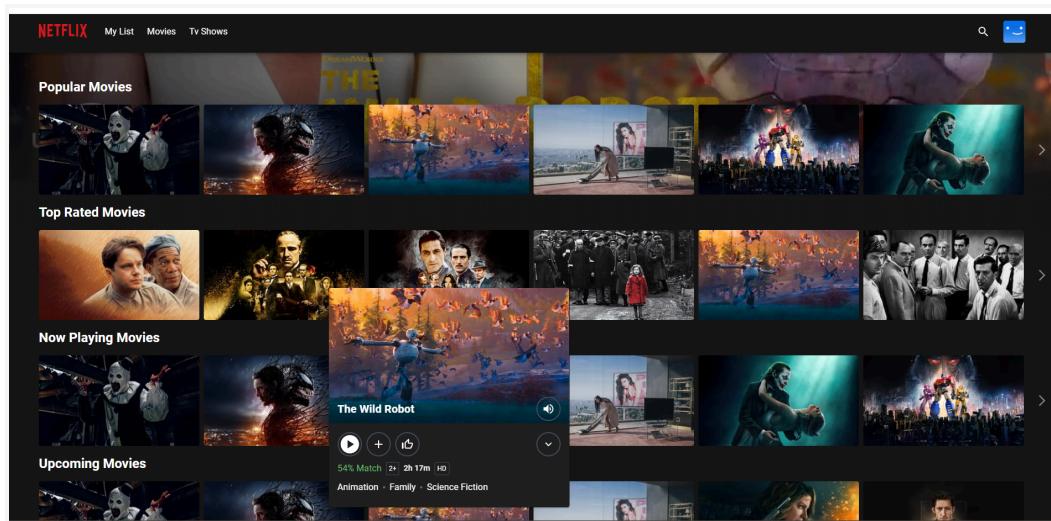
The screenshot shows the 'Inbound rule 4' configuration in the AWS Security Groups interface. It includes fields for Security group rule ID, Type (Custom TCP), Port range (9100), Source type (Anywhere-IPv4), Source (0.0.0.0/0), and a description field containing 'node exporter'. A blue box highlights the 'node exporter' text.

You can get the instance Ip address from here

The screenshot shows the AWS EC2 Instances page for instance i-02a916e8b578bb93d. The instance is listed as 'Running' with a public IPv4 address of 54.146.199.39 and a private IPv4 address of 172.31.17.235. The sidebar shows navigation links for Dashboard, Services, and various EC2-related options like Instances, Instance Types, and Launch Templates.

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
```

```
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 the Prometheus 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 add the path as well :