

CI/CD Pipeline for Automating Website Deployment

Server 1 (EC2):

- Purpose: Jenkins server for CI/CD pipeline.
- Role: Automates build and deployment triggers.

Server 2 (EC2):

- Purpose: Ansible server for configuration management.
- Role: Executes playbooks to manage deployments (deployment.yaml, service.yaml).

Server 3 (EC2):

- Purpose: Terraform for infrastructure as code (IaC).
- Role: Manages VPC creation for Server 1 and Server 2.

The screenshot shows the AWS Management Console interface for the EC2 service. In the left sidebar, under the 'Instances' section, there are links for 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', and 'Dedicated Hosts'. The main content area is titled 'Instances (3) Info' and displays a table of three running instances. The columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IP. The instances are: 'jenkins server' (i-0150c5c06bab55fe2), 'terraform server' (i-04326746e81c90482), and 'ansible server' (i-0d4f78c5c69705e1e). All three instances are listed as 'Running'.

This terraform installation process:

Switch the normal user to the root user then update my server because I use Ubuntu OS, so OS update requires installing some packages. after installing the terraform and step-by-step type all terraform commands.

The `wget` command is used to copy the terraform package link.

The screenshot shows a terminal window with the following command and output:

```
root@ip-172-31-45-227:~# wget https://releases.hashicorp.com/terraform/1.4.7/terraform_1.4.7_linux_amd64.zip
--2024-12-14 08:17:57-- https://releases.hashicorp.com/terraform/1.4.7/terraform_1.4.7_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 3.171.05.80, 3.171.05.65, 3.171.05.128, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|3.171.05.80|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20779728 (20MiB) [application/zip]
Saving to: 'terraform_1_4_7_linux_amd64.zip'

terraform_1.4.7_linux_amd64.zip          100%[=====] 19.82M   114MB/s    in 0.2s
2024-12-14 08:17:57 (114 MB/s) - `terraform_1.4.7_linux_amd64.zip' saved [20779728/20779728]
root@ip-172-31-45-227:~#
```

Then unzip the terraform zip file.

The screenshot shows a terminal window with the following command and output:

```
root@ip-172-31-45-227:~# ll
total 20324
drwx----- 4 root root 4096 Dec 14 08:17 .
drwxr-xr-x 19 root root 4096 Dec 14 08:15 ..
-rw-r--r-- 1 root root 3106 Oct 15 2021 .bashrc
-rw-r--r-- 1 root root 161 Jul 9 2019 .profile
drwx----- 2 root root 4096 Dec 14 08:15 .ssh/
-rw-r--r-- 1 root root 177 Dec 14 08:17 .wget-hsts
drwx----- 4 root root 4096 Dec 14 08:16 snap/
-rw-r--r-- 1 root root 20779728 Sep 13 2023 terraform_1.4.7_linux_amd64.zip
root@ip-172-31-45-227:~#
```

After that unzip that file then run Terraform used to unzip the Terraform file.

```
drwx----- 4 root root 4096 Dec 14 08:18 .
drwxr-xr-x 19 root root 4096 Dec 14 08:15 ..
-rw-r--r-- 1 root root 3106 Oct 15 2021 .bashrc
-rw-r--r-- 1 root root 161 Jul 9 2019 .profile
drwx----- 2 root root 4096 Dec 14 08:15 .ssh/
-rw-r--r-- 1 root root 177 Dec 14 08:17 .wget-hsts
drwx----- 4 root root 4096 Dec 14 08:16 snap/
-rwxr-xr-x 1 root root 64626688 Sep 13 2023 terraform*
-rw-r--r-- 1 root root 20779728 Sep 13 2023 terraform_1.4.7_linux_amd64.zip
```

The next step is to create a file using with Terraform extension. for example (main.tf)

```
route {
  cidr_block = "0.0.0.0/0"
  gateway_id = aws_internet_gateway.igwl.id
}

tags = [
  Name = "terraform_pub_rt"
  Managed_by = "terraform"
]

#7. Private route table
resource "aws_route_table" "private_RT" {
  vpc_id = aws_vpc.vpcl.id

  tags = [
    Name = "terraform_pri_rt"
    Managed_by = "terraform"
  ]
}

#8.public subnet association
resource "aws_route_table_association" "pubsubl_pubrt" {
  subnet_id   = aws_subnet.public_subnet_1.id
  route_table_id = aws_route_table.public_RT.id
}

#9.private subnet association
resource "aws_route_table_association" "prisubl_prixt" {
  subnet_id   = aws_subnet.private_subnet_1.id
  route_table_id = aws_route_table.private_RT.id
}
```

"main.tf" 95L, 1767B

95,1

Bot

(Terraform init) This command is used to create a terraform repo and also check the terraform code

```
AWS Search [Alt+S] N. Virginia ass @ 1243-5566-1575

root@ip-172-31-45-227:~# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.0"...
- Installing hashicorp/aws v5.81.0...
- Installed hashicorp/aws v5.81.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ip-172-31-45-227:~#
```

Step by step type the terraform command and eventually create a vpc for two servers. The second command is used for planning I mean terraform tell a server. what will it create using that code? Last command is terraform apply it command using for run code and then create a vpc.

```
Enter a value: yes

aws_vpc.vpcl: Creating...
aws_vpc.vpcl: Creation complete after 1s [id=vpc-066d046814e3857b5]
aws_route_table.private_RT: Creating...
aws_internet_gateway.igwl: Creating...
aws_subnet.public_subnet_1: Creating...
aws_subnet.private_subnet_1: Creating...
aws_internet_gateway.igwl: Creation complete after 1s [id=igw-01cac0266685fc13a]
aws_route_table.public_RT: Creating...
aws_route_table.private_RT: Creation complete after 1s [id=rtb-0db02d216ff5723df]
aws_subnet.public_subnet_1: Creation complete after 1s [id=subnet-056c7a149b7882c45]
aws_subnet.private_subnet_1: Creation complete after 1s [id=subnet-05ff94081b02b5085]
aws_route_table_association.prisubl_prixt: Creating...
aws_route_table_association.prisubl_prixt: Creation complete after 0s [id=rtaassoc-043bb3708e403dc03]
aws_route_table_association.public_RT: Creation complete after 0s [id=rtb-012e096b0635990db]
aws_route_table_association.pubsubl_pubrt: Creating...
aws_route_table_association.pubsubl_pubrt: Creation complete after 1s [id=rtaassoc-0f48ded7999ff1604]

Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
root@ip-172-31-45-227:~#
```

i-04326746e81c90482 (terraform server)

Finally, successfully create a VPC:

The screenshot shows the AWS VPC dashboard with the title "Your VPCs (1/3) Info". It lists three VPCs in a table:

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option
-	vpc-031ae5e788f6f45ee	Available	Off	172.31.0.0/16	-	dopt-0134ce
main_vpc	vpc-0dd47c3e6015c8245	Available	Off	10.0.0.0/16	-	dopt-0134ce
terraform_vpc	vpc-066cd046814e3857b5	Available	Off	10.0.0.0/16	-	dopt-0134ce

JENKINS SERVER:

(Server 1 Jenkins server)

Sudo apt update:

```
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

 System information as of Sat Dec 14 09:55:29 UTC 2024

ubuntu@ip-172-31-17-2:~$ sudo apt update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
```

Then install the Jenkins Java package because Jenkins build Java so that Java package requires.

```
ubuntu@ip-172-31-17-2:~$ 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 install package:

```
Adding debian:Trustwave_Global_ECC_P256_Certification_Authority.pem
Adding debian:Trustwave_Global_ECC_P384_Certification_Authority.pem
Adding debian:TunTrust_Root_CA.pem
Adding debian:UCA_Extended_Validation_Root.pem
Adding debian:UCA_Global_G2_Root.pem
Adding debian:USERTrust_ECC_Certification_Authority.pem
Adding debian:USERTrust_RSA_Certification_Authority.pem
Adding debian:Xamp_Global_CA_Root.pem
Adding debian:certsIGN_ROOT_CA.pem
Adding debian:certsIGN_Root_CA_G2.pem
ubuntu@ip-172-31-17-2:~$ 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/" \
> /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
```

Then go to the Ansible server then install the Ansible package in (server 2 ansible server)

```
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

 System information as of Sat Dec 14 10:04:21 UTC 2024

ubuntu@ip-172-31-31-118:~$ sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

Then install the DOCKER in server 2.



```
ubuntu@ip-172-31-31-118:~$ sudo apt update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu noble InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-31-118:~$ # Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
# Add the repository to Apt sources:
echo \"
deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Then check docker is successfully installed in server 2

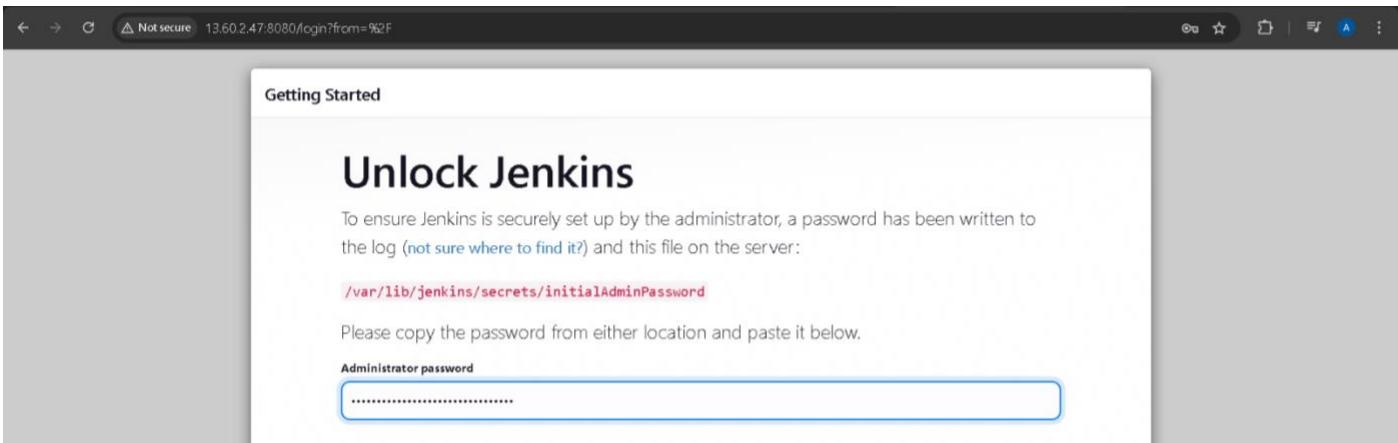
Docker images



```
ubuntu@ip-172-31-31-118:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu@ip-172-31-31-118:~$
```

Next step: copy the Jenkins server public IP address number and then paste it within the browser. Then open the Jenkins page. Copy the path above the admin password.

1)



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

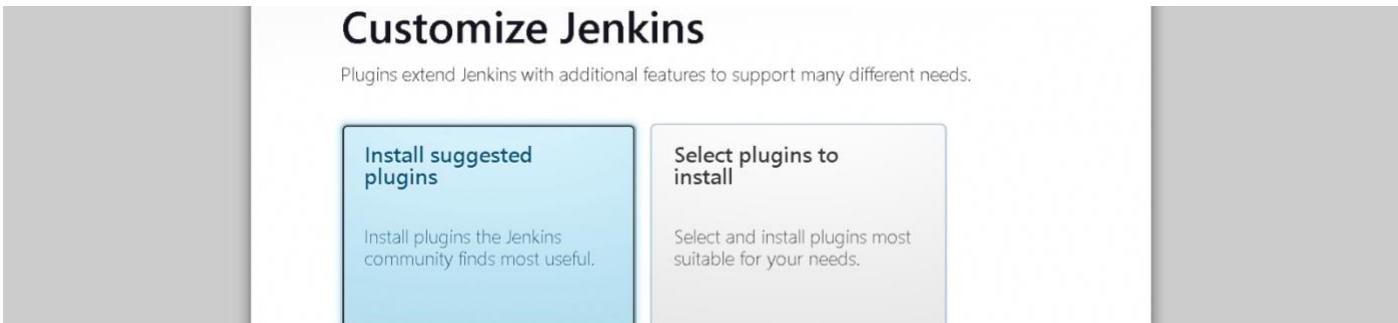
`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

.....

2)



Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

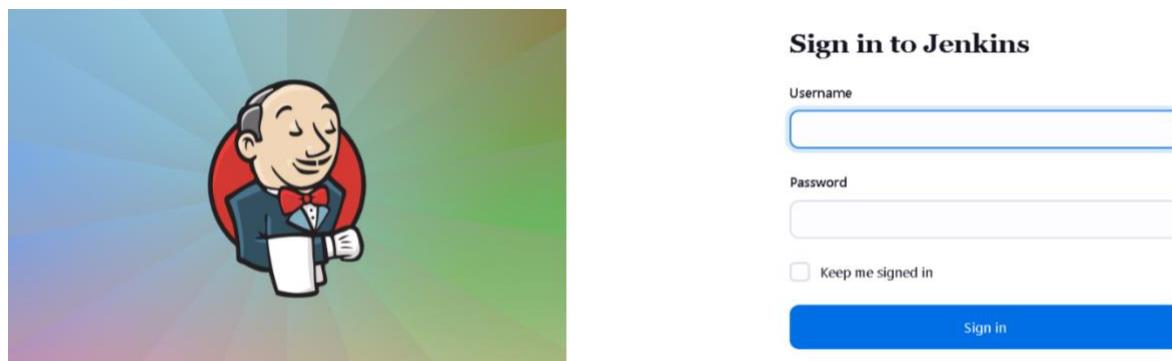
Then create a username and password after install the default plugins.

Install this plugin (**ssh-agent**, **pipeline stage view**, **docker pipeline**)

The screenshot shows the Jenkins plugin manager interface. On the left, there's a sidebar with options like 'Updates', 'Available plugins' (which is selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area has a search bar at the top right. Below it, a table lists three plugins with checkboxes next to them:

Install	Name	Released
<input checked="" type="checkbox"/>	SSH Agent 376.v8933585c69d3 This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.	4 mo 16 days ago
<input checked="" type="checkbox"/>	Pipeline: Stage View 2.34 User Interface Pipeline Stage View Plugin.	1 yr 1 mo ago
<input checked="" type="checkbox"/>	Docker Pipeline 580.vc0c340686b_54 pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	6 mo 26 days ago

Install important plugins then restart Jenkins and type the username and password.



And then already created a docker file and index.html file then pushed those files to GitHub

Name	Date modified	Type	Size
Dockerfile	10-12-2024 20:10	File	1 KB
index	13-12-2024 13:14	Microsoft Edge HT...	4 KB

GIT:

Git init. This command is used for create a git repo because I didn't create a git repo then I never pushed those files to GitHub.

```
MINGW64:/d/devops pipeline project
rishi@arisisivasankar MINGW64 /d/devops pipeline project
git init
initialized empty Git repository in D:/devops pipeline project/.git/
rishi@arisisivasankar MINGW64 /d/devops pipeline project (main)
```

Type the step by step git command.

Git remote add origin: This command is used to connect the remote repo.

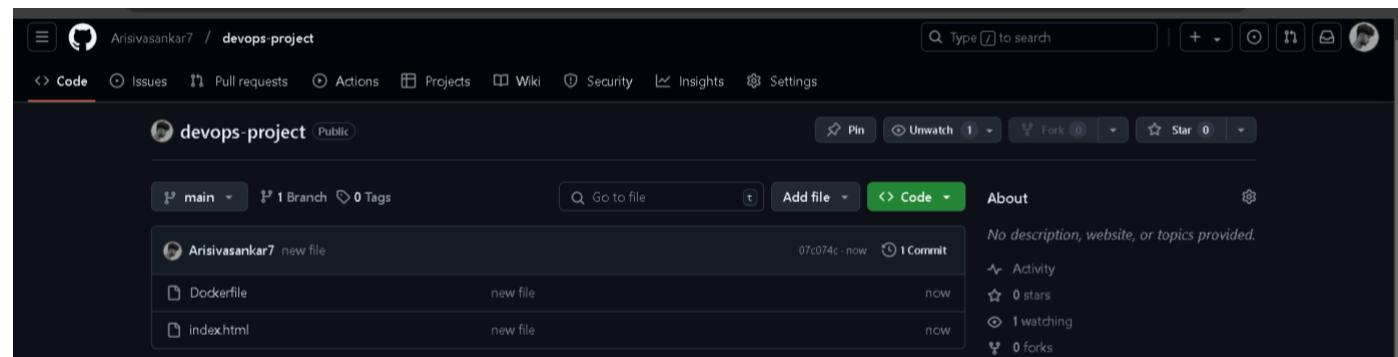
Git add: used to push those files in the staging area which means temporary storage.

Git commit: command used to push stage area to the local repo.

```
MINGW64:/d/devops pipeline project
arisi@arisivasankar MINGW64 /d/devops pipeline project (main)
$ git remote add origin https://github.com/Arisivasankar7/devops-project.git
arisi@arisivasankar MINGW64 /d/devops pipeline project (main)
$ git add .
arisi@arisivasankar MINGW64 /d/devops pipeline project (main)
$ git commit -m "new file"
[main (root-commit) 07c074c] new file
 2 files changed, 180 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 index.html
arisi@arisivasankar MINGW64 /d/devops pipeline project (main)
$ |
```

Git push origin main: these commands are used to push those files in the remote repo.

```
arisi@arisivasankar MINGW64 /d/devops pipeline project (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 2.12 KiB | 433.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Arisivasankar7/devops-project.git
 * [new branch]      main -> main
arisi@arisivasankar MINGW64 /d/devops pipeline project (main)
$
```



Then go back Jenkins page. Create a new job.

The screenshot shows the Jenkins 'New Item' creation interface. The item name is set to 'devops-project'. Under 'Select an item type', the 'Pipeline' option is selected, with a description: 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.' Other options like 'Freestyle project' and 'Workflow' are also listed.

Select the pipeline then click (ok) then choose build trigger.

The screenshot shows the Jenkins 'Configuration' page for the 'devops-project'. The 'Build Triggers' section is active, showing several trigger options: 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling' (which is checked), 'Poll SCM', 'Quiet period', and 'Trigger builds remotely (e.g., from scripts)'.

Then click pipeline syntax and choose the **git: Git** and copy the GitHub repo link. After that paste it into the repo URL and select the branch finally click the generate pipeline script and automatically create a script then copy that script and go back script page and paste it.

The screenshot shows the Jenkins 'Pipeline Syntax' page. The 'Snippet Generator' for 'git: Git' is selected. The 'Repository URL' is set to 'https://github.com/Arisivasankar7/devops-project.git' and the 'Branch' is set to 'main'. A sample step 'git' is shown in the dropdown.

Generate Pipeline Script

```
git branch: 'main', url: 'https://github.com/Arisivasankar7/devops-project.git'
```

This script is used to pull the files from a GitHub repo

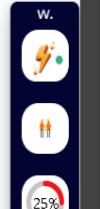
Definition

Pipeline script

Script ?

```
1 ~ node{  
2 ~   stage('pull the file from github repo'){  
3 |     git branch: 'main', url: 'https://github.com/Arisivasankar7/devops-project.git'  
4 }  
5 }
```

try sample Pipeline... ▾



Then click apply and save and click build now then automatically pulled the files from GitHub repo.

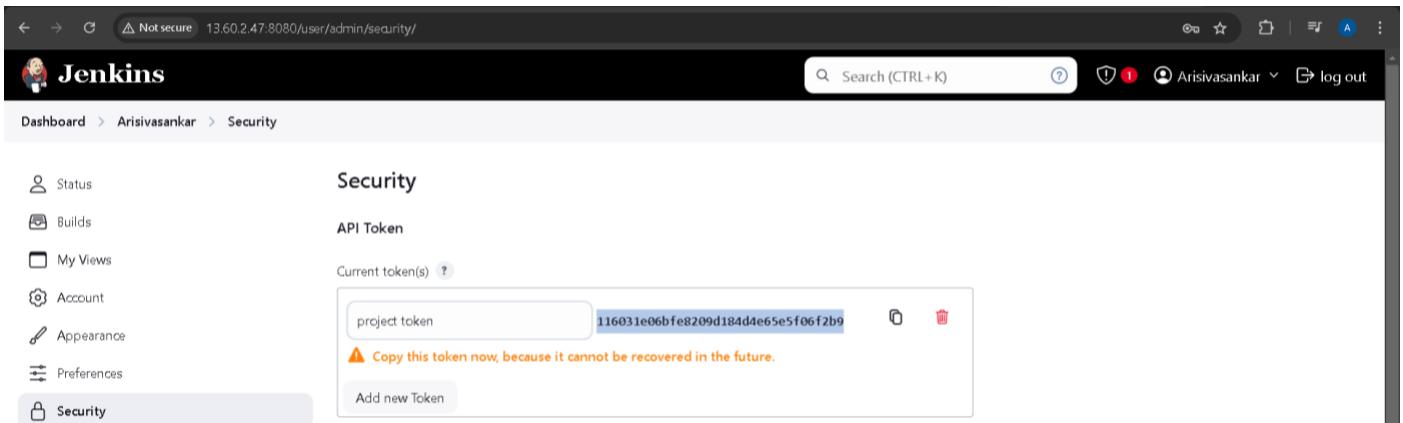
The screenshot shows the Jenkins Pipeline Stage View for the 'devops-project'. On the left, there's a sidebar with options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The main area is titled 'Stage View' and shows a single stage named 'pull the file from github repo'. This stage is represented by a light blue box with a green bar indicating it took 4 seconds. Below the box, it says 'Dec 14 16:00' and 'No Changes'. A tooltip above the stage says 'Average stage times: (Average full run time: ~5s)'. To the right of the stage view, there are three small circular icons with letters W, J, and H, and a progress bar at 25%.

Go back to the Jenkins server and change the Jenkins path.

```
ubuntu@ip-172-31-17-2:~$ cd /var/lib/jenkins/workspace/devops-project  
ubuntu@ip-172-31-17-2:~/var/lib/jenkins/workspace/devops-project$ ls  
Dockerfile index.html  
ubuntu@ip-172-31-17-2:~/var/lib/jenkins/workspace/devops-project$
```

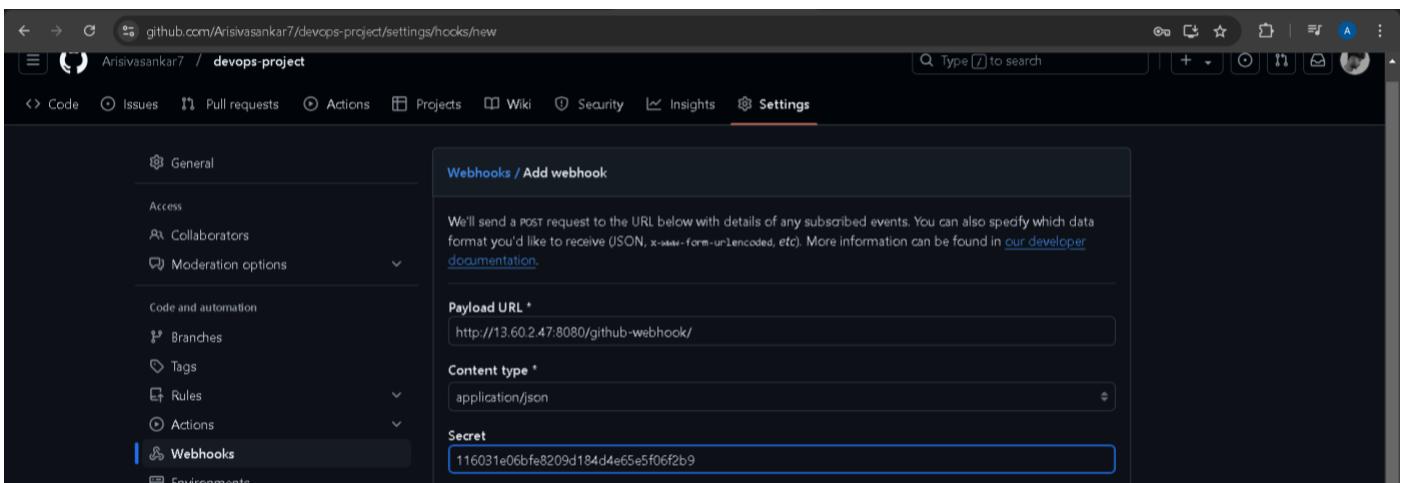
Successfully run first stage script

The next step create a GitHub webhook because if I don't create a GitHub webhook then never automatically triggers the stage script so first create an API token. this token is just for security purposes and to create a GitHub webhook. What it takes to make it create.. the first Jenkins page URL and next is API token.

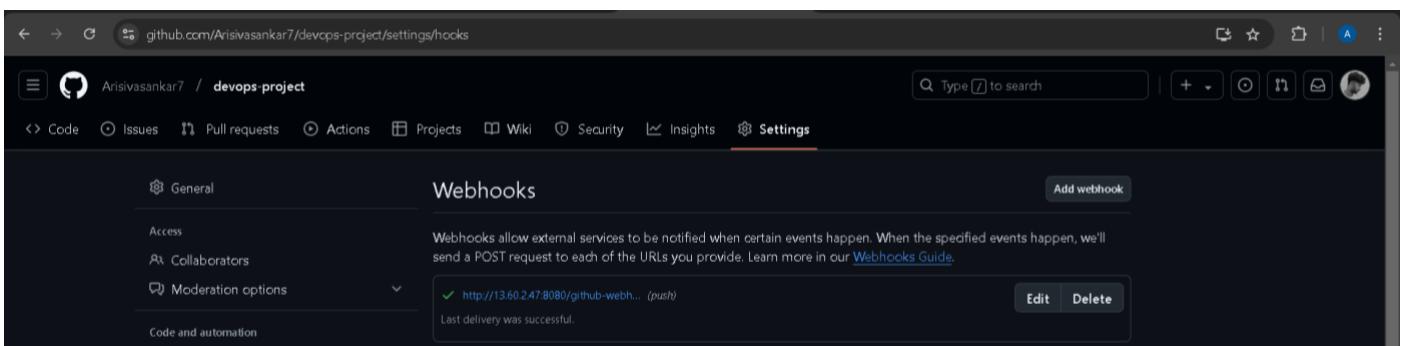


The screenshot shows the Jenkins Security page at the URL 13.60.2.47:8080/user/admin/security/. The left sidebar has links for Status, Builds, My Views, Account, Appearance, Preferences, and Security (which is selected). The main area is titled "Security" and "API Token". It shows a current token named "project token" with the value "116031e06bfe8209d184d4e65e5f06f2b9". A warning message says "⚠ Copy this token now, because it cannot be recovered in the future." Below the token is a button to "Add new Token".

GitHub webhook page:



The screenshot shows the GitHub Settings page for a repository named "devops-project" at the URL github.com/Arisivasankar7/devops-project/settings/hooks/new. The left sidebar has sections for General, Access, Collaborators, Moderation options, Code and automation (Branches, Tags, Rules, Actions), and Webhooks (which is selected). The main area is titled "Webhooks / Add webhook". It explains that GitHub will send a POST request to the URL with event details. The "Payload URL" field contains "http://13.60.2.47:8080/github-webhook/", the "Content type" is set to "application/json", and the "Secret" field contains "116031e06bfe8209d184d4e65e5f06f2b9".



The screenshot shows the GitHub Settings page for the same repository, now displaying the list of webhooks. The left sidebar is identical to the previous screenshot. The main area is titled "Webhooks" and shows one entry: "http://13.60.2.47:8080/github-webhook... (push)". Below the URL, it says "Last delivery was successful." There are "Edit" and "Delete" buttons next to the URL.

The next concept is to create a master machine and a slave machine:

Instances	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Instances	jenkins	i-0e1c19b3e924587bf	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-51-20-
Instance Types	ansible	i-06e2323e2e26e48f4	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-16-17-
Launch Templates	master	i-09983b809d672575f	Running	t3.medium	3/3 checks passed	View alarms +	eu-north-1a	ec2-51-20-
Spot Requests	slave	i-0b063bed401a7a442	Running	t3.medium	3/3 checks passed	View alarms +	eu-north-1a	ec2-13-49-

Kubernetes:

Master machine:

```
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease:/main/deb InRelease  
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.29/deb InRelease  
Reading package lists... Done  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
jq is already the newest version (1.7.1-3build1).  
jq set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.  
ubuntu@ip-172-31-19-214:~$ sudo kubeadm config images pull  
sudo kubeadm init  
mkdir -p "$HOME"/.kube  
sudo cp -i /etc/kubernetes/admin.conf "$HOME"/.kube/config  
sudo chown "$(id -u)":$(id -g) "$HOME"/.kube/config  
# Network Plugin = calico  
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml  
i-00cdbb293abbd2541 (master )  
PublicIP: 16.171.55.133 PrivateIP: 172.31.10.214
```

Slave machine:

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"  
echo "$(cat kubectl.sha256)  kubectl" | sha256sum --check  
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl  
chmod +x kubectl  
mkdir -p ~/.local/bin  
mv ./kubectl ~/.local/bin/kubectl  
# and then append (or prepend) ~/.local/bin to $PATH  
kubectl version --client  
# disable swap  
sudo swapoff -a  
# Create the .conf file to load the modules at bootup  
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf  
overlay  
br_netfilter  
EOF  
sudo modprobe overlay  
sudo modprobe br_netfilter  
# sysctl params required by setup, params persist across reboots  
sudo systemctl start kubeletkubelet29.0-* kubectl="1.29.0-*" kubeadm="1.29.0-*"k8s.io/core:/stable:/v1.29/deb/ /* | sudo tee /etc/apt/sources.list.d/kubernetes.list>>  
i-0ada6fd785aef040a (slave)  
PublicIPs: 16.170.108.195 PrivateIPs: 172.31.5.188
```

Then copy and paste kubeadm token:

```
aws | ⚡ | Search [Alt+S] | 🌐 | 🔍 | ⓘ | 🛡️ | Stockholm | ass@123-5566-1379 | ↻
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgpfilters.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/irpreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
ubuntu@ip-172-31-19-214:~$ kubeadm token create --print-join-command
```

i-00cd8b293abbd2541 (master)
PublicIPs: 16.171.55.133 PrivateIPs: 172.31.19.214

Slave:

```
Building dependency tree... Done
Reading state information... Done
jq is already the newest version (1.7.1-3build1).
jq set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded,
ubuntu@ip-172-31-5-188:~$ sudo kubeadm reset pre-flight checks
```

i-0ada6fd785aef040a (slave)
PublicIPs: 16.170.108.195 PrivateIPs: 172.31.5.188

Then copy this line and those line paste it slave server.

```
aws | ⚡ | Search [Alt+S] | 🌐 | 🔍 | ⓘ | 🛡️ | Stockholm | ass@123-5566-1379 | ↻
customresourcedefinition.apiextensions.k8s.io/bgpfilters.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/irpreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
ubuntu@ip-172-31-19-214:~$ kubeadm token create --print-join-command
kubeadm join 172.31.19.214:6443 --token ftsospa.u728ou0x9v2g6ln2 --discovery-token-ca-cert-hash sha256:a42321dd8bf2elaa267a479b43d28432758c5ca81cd7755af1f5662fa064c54a
ubuntu@ip-172-31-19-214:~$
```

i-00cd8b293abbd2541 (master)
PublicIPs: 16.171.55.133 PrivateIPs: 172.31.19.214

Successfully configured kubeadm cluster.

Type this command after successfully conigigure kubeadm cluster.

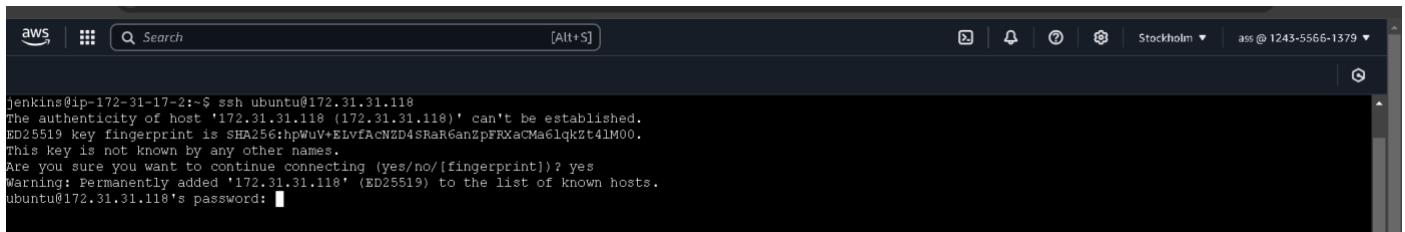


```
ubuntu@ip-172-31-19-214:~$ kubectl get nodes
NAME      STATUS   ROLES      AGE     VERSION
ip-172-31-19-214   Ready    control-plane   8m40s   v1.29.0
ip-172-31-5-188   Ready    <none>     19s    v1.29.0
ubuntu@ip-172-31-19-214:~$
```

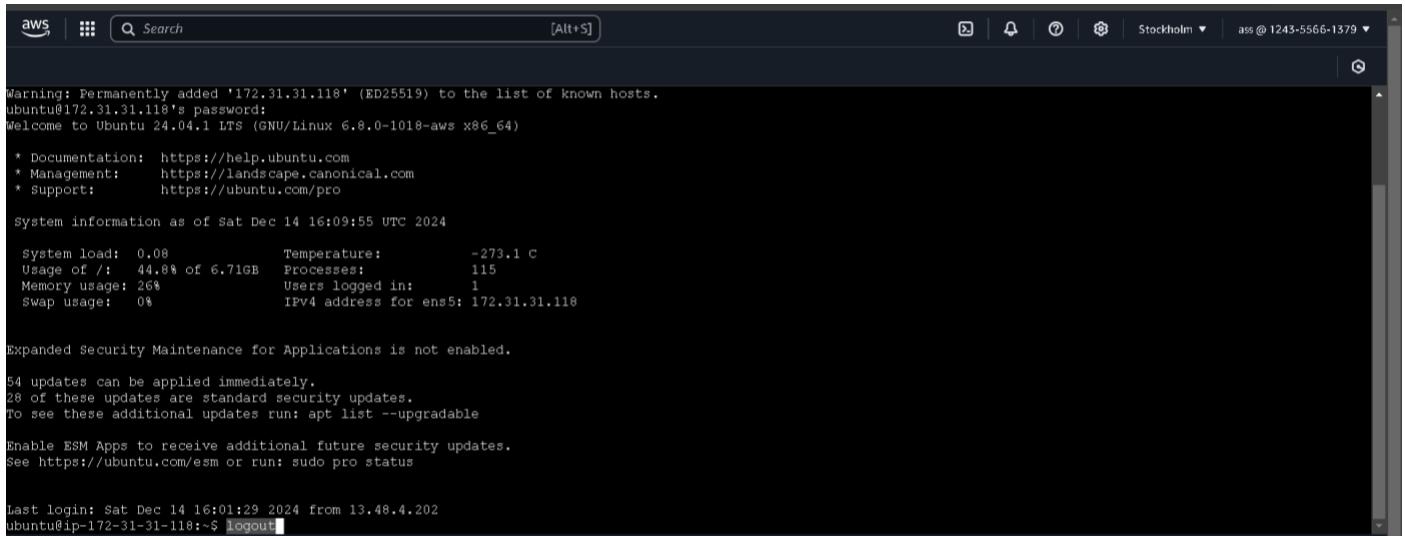
Next step is passwordless authentication:

First switch normal user to Jenkins user then create a passwordless key. After that go to the ansible serverThen modify some files and successfully modify those files then create a password for security purpose and that password helpful to connect server to server so it requires and also same process in master server.

ssh ubuntu@123.34.445.6 : this command is using to connect server to server and it ask password then type it password then connected that server in Jenkins server



```
Jenkins@ip-172-31-17-2:~$ ssh ubuntu@172.31.31.118
The authenticity of host '172.31.31.118 (172.31.31.118)' can't be established.
ED25519 key fingerprint is SHA256:hpWuV+ElvfAcNZD4SRaR6anZpFRXaCMa6lqKzt4lM00.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.31.118' (ED25519) to the list of known hosts.
ubuntu@172.31.31.118's password: [REDACTED]
```



```
Warning: Permanently added '172.31.31.118' (ED25519) to the list of known hosts.
ubuntu@172.31.31.118's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

System information as of Sat Dec 14 16:09:55 UTC 2024

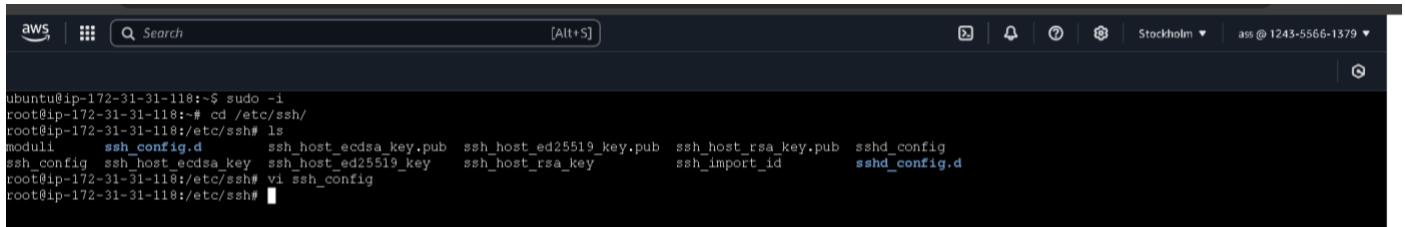
System load: 0.08      Temperature:        -273.1 C
Usage of /: 44.8% of 6.71GB  Processes:          115
Memory usage: 26%        Users logged in:      1
Swap usage:  0%          IPv4 address for ens5: 172.31.31.118

Expanded Security Maintenance for Applications is not enabled.

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

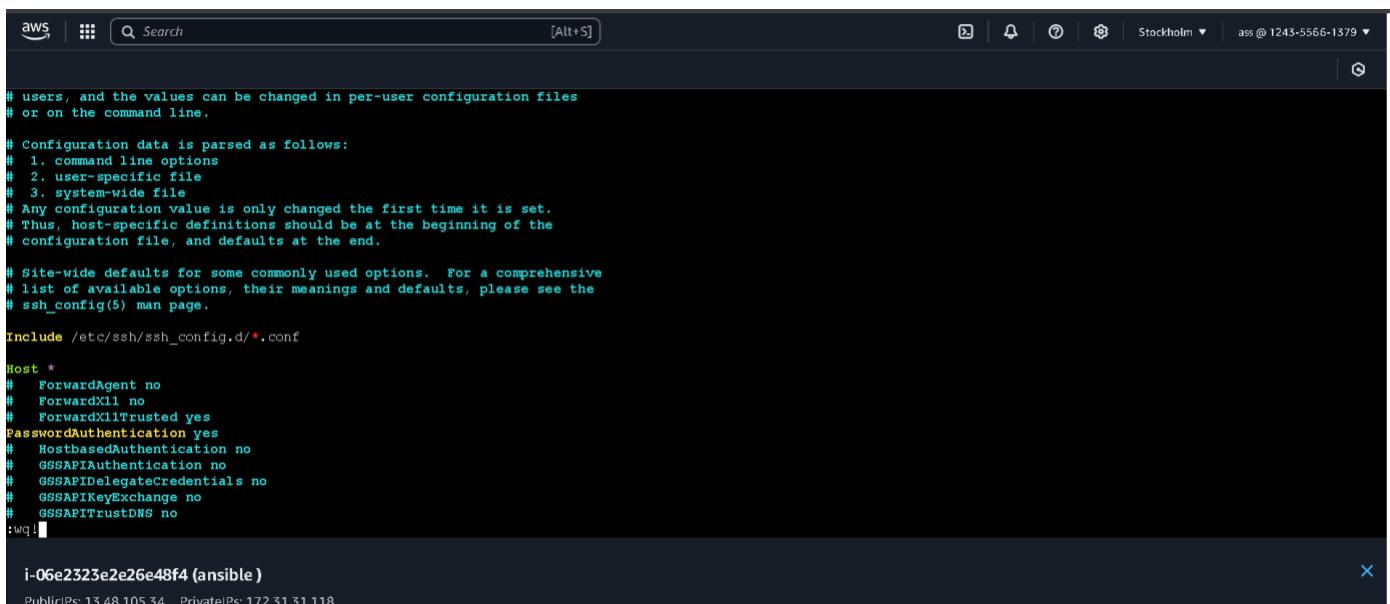
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Dec 14 16:01:29 2024 from 13.48.4.202
ubuntu@ip-172-31-31-118:~$ logout [REDACTED]
```



```
ubuntu@ip-172-31-31-118:~$ sudo -i
root@ip-172-31-31-118:~# cd /etc/ssh/
root@ip-172-31-31-118:/etc/ssh# ls
moduli      ssh_config.d      ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub  sshd_config
ssh_config  ssh_host_ecdsa_key  ssh_host_ed25519_key  ssh_host_rsa_key      ssh_import_id      sshd_config.d
root@ip-172-31-31-118:/etc/ssh# vi ssh_config
root@ip-172-31-31-118:/etc/ssh# [REDACTED]
```

First file



```
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
# 1. command line options
# 2. user-specific file
# 3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options. For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Include /etc/ssh/sshd_config.d/*.conf

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
PasswordAuthentication yes
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegatecredentials no
# GSSAPIKeyExchange no
# GSSAPITrustDNS no

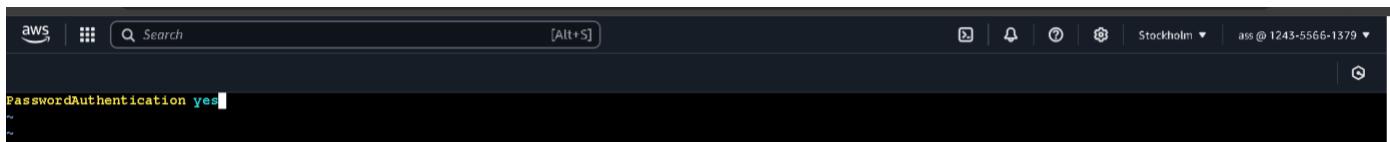
:Wq

i-06e2323e2e26e48f4 (ansible)
PublicIPs: 13.48.105.34 PrivateIPs: 172.31.31.118
```



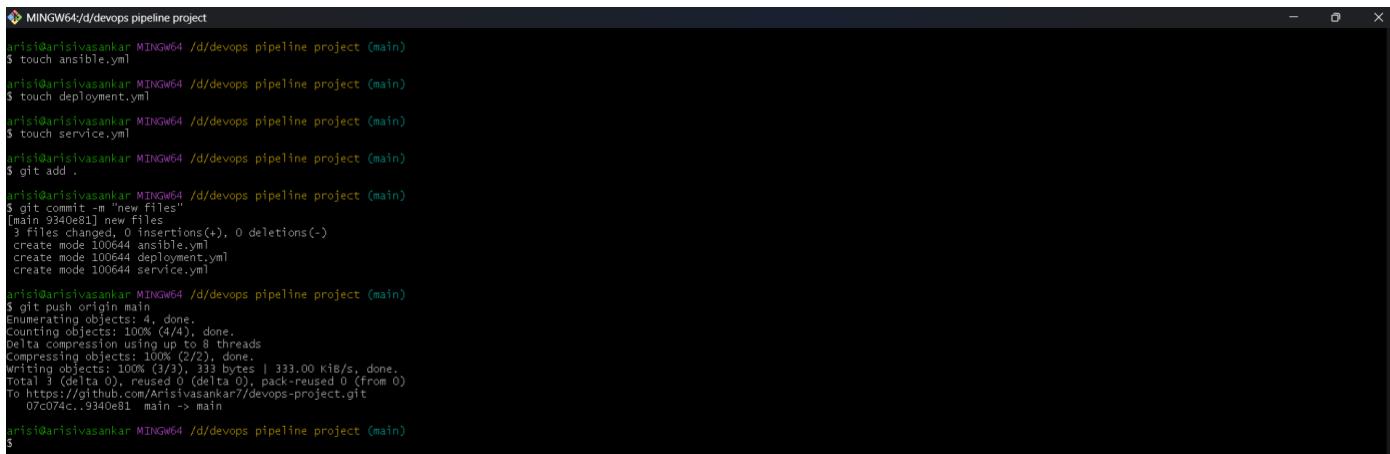
```
ubuntu@ip-172-31-31-118:~$ sudo -i
root@ip-172-31-31-118:~# cd /etc/ssh/
root@ip-172-31-31-118:/etc/ssh# ls
moduli      ssh_config.d      ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub  sshd_config
ssh_config  ssh_host_ecdsa_key  ssh_host_ed25519_key   ssh_host_rsa_key       ssh_import_id      sshd_config.d
root@ip-172-31-31-118:/etc/ssh# vi ssh_config
root@ip-172-31-31-118:/etc/ssh# cd sshd_config.d/
root@ip-172-31-31-118:/etc/ssh/sshd_config.d# ls
60-cloudimg-settings.conf
root@ip-172-31-31-118:/etc/ssh/sshd_config.d# vi 60-cloudimg-settings.conf
```

Second file.



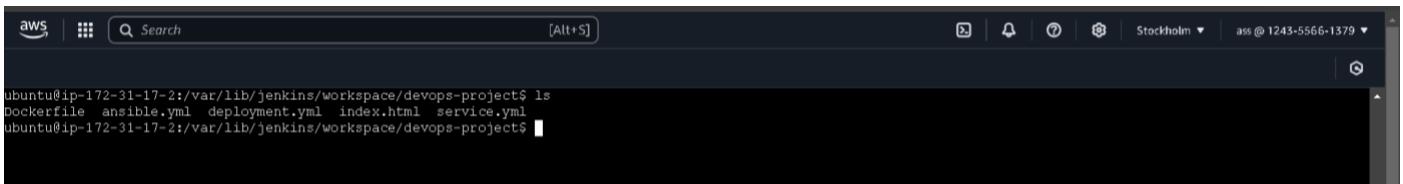
```
PasswordAuthentication yes
```

Next one is files creation: **ansible.yml, deployment.yml, service.yml**



```
arisi@arisisivasankar MINGW64 ~/d/devops pipeline project
$ touch ansible.yml
arisi@arisisivasankar MINGW64 ~/d/devops pipeline project (main)
$ touch deployment.yml
arisi@arisisivasankar MINGW64 ~/d/devops pipeline project (main)
$ touch service.yml
arisi@arisisivasankar MINGW64 ~/d/devops pipeline project (main)
$ git add .
arisi@arisisivasankar MINGW64 ~/d/devops pipeline project (main)
$ git commit -m "New files"
[main 9340e81] new files
 3 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 ansible.yml
  create mode 100644 deployment.yml
  create mode 100644 service.yml
arisi@arisisivasankar MINGW64 ~/d/devops pipeline project (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 333 bytes | 333.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Arisisivasankar7/devops-project.git
 07c074c..9340e81 main -> main
arisi@arisisivasankar MINGW64 ~/d/devops pipeline project (main)
$
```

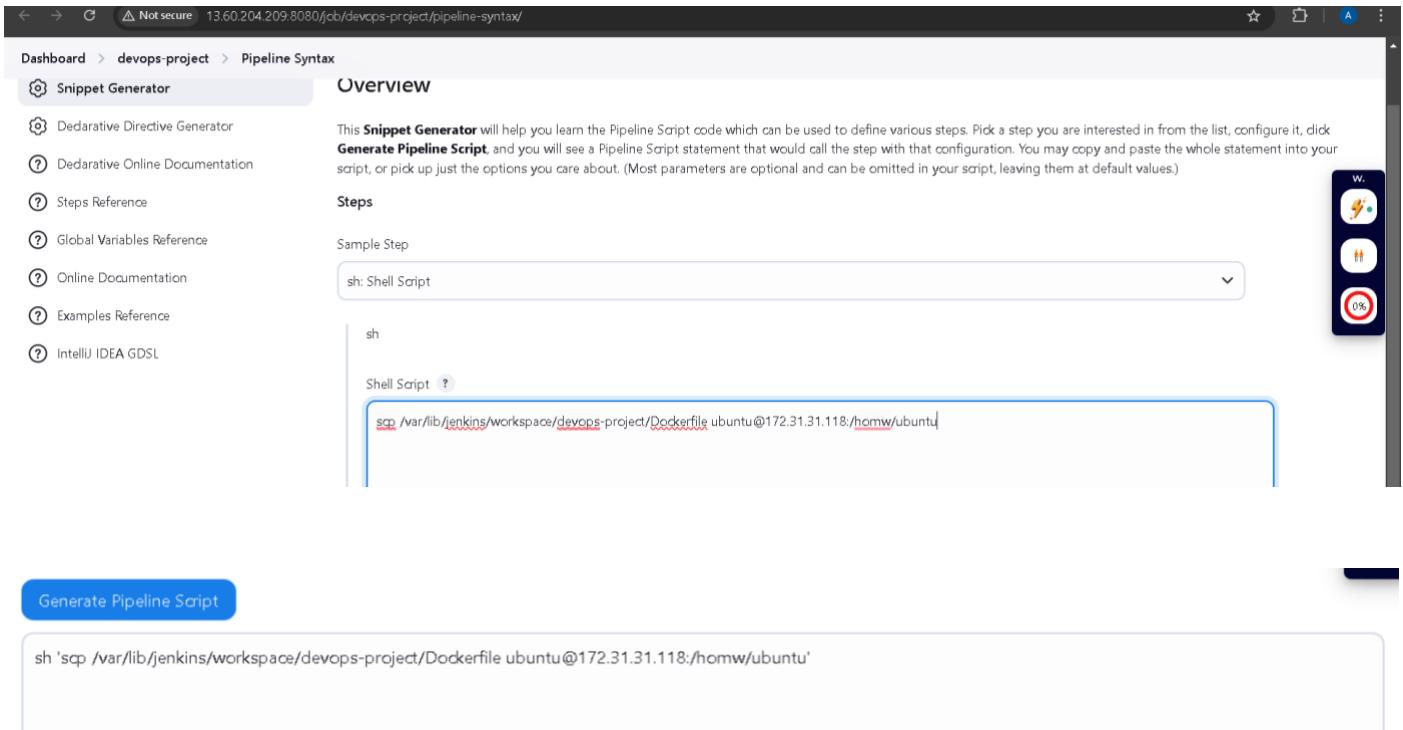
Then automatically transfer files in server because already created a github webhook so push that files to github then Jenkins job automatically trigger.



```
aws | ⚡ Search [Alt+S]
ubuntu@ip-172-31-17-2:/var/lib/jenkins/workspace/devops-project$ ls
Dockerfile ansible.yml deployment.yml index.html service.yml
ubuntu@ip-172-31-17-2:/var/lib/jenkins/workspace/devops-project$
```

A screenshot of a terminal window titled 'aws'. It shows the command 'ls' being run in the directory '/var/lib/jenkins/workspace/devops-project'. The output lists several files: Dockerfile, ansible.yml, deployment.yml, index.html, and service.yml.

Then go back Jenkins page and click pipeline syntax and choose **sh: shell script**

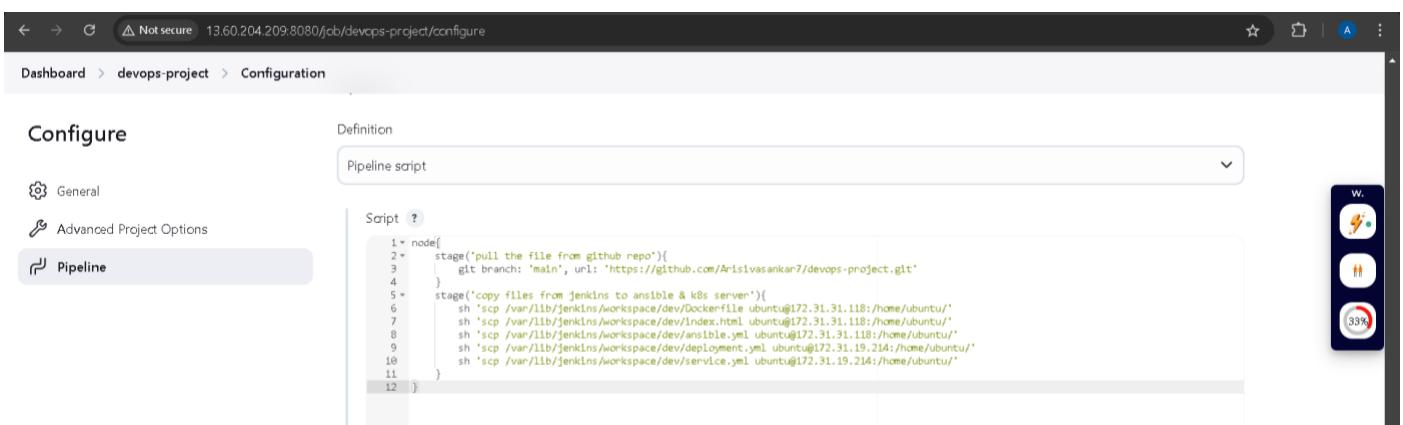


The screenshot shows the Jenkins Pipeline Syntax Generator interface. On the left, there's a sidebar with links like Snippet Generator, Declarative Directive Generator, etc. The main area has tabs for Overview and Steps. Under Steps, there's a dropdown set to 'sh: Shell Script'. Below it, there's a code editor with the following Groovy code:

```
sh 'scp /var/lib/jenkins/workspace/devops-project/Dockerfile ubuntu@172.31.31.118:/homw/ubuntu'
```

At the bottom, there's a 'Generate Pipeline Script' button and a preview box containing the same Groovy code.

Then copy this command and paste it stage script:



The screenshot shows the Jenkins Configuration page for the 'devops-project'. On the left, there are tabs for General, Advanced Project Options, and Pipeline. The Pipeline tab is selected. In the center, there's a 'Definition' section with a dropdown set to 'Pipeline script'. Below it is a code editor with the following Groovy code:

```
node{
  stage('pull the file from github repo'){
    git branch: 'main', url: 'https://github.com/Arisivasankar7/devops-project.git'
  }
  stage('copy files from Jenkins to ansible & k8s server'){
    sh 'scp /var/lib/jenkins/workspace/dev/Dockerfile ubuntu@172.31.31.118:/home/ubuntu/'
    sh 'scp /var/lib/jenkins/workspace/dev/index.html ubuntu@172.31.31.118:/home/ubuntu/'
    sh 'scp /var/lib/jenkins/workspace/dev/ansible.yml ubuntu@172.31.31.118:/home/ubuntu/'
    sh 'scp /var/lib/jenkins/workspace/dev/deployment.yml ubuntu@172.31.19.214:/home/ubuntu/'
    sh 'scp /var/lib/jenkins/workspace/dev/service.yml ubuntu@172.31.19.214:/home/ubuntu/'
  }
}
```

Then I was face one error because type the wrong job name that's why error then change my job name

The screenshot shows the Jenkins interface with the 'Console Output' tab selected. The output window displays the following error message:

```
+ scp /var/lib/jenkins/workspace/dev/Dockerfile ubuntu@172.31.31.118:/home/ubuntu/
scp: stat local "/var/lib/jenkins/workspace/dev/Dockerfile": No such file or directory
```

Pipeline stage view:

The screenshot shows the Jenkins interface with the 'Stage View' tab selected. The stage 'copy files from jenkins to ansible & k8s server' failed with a duration of 402ms. The stage view table includes the following data:

Stage	Average stage times: (Average full run time: ~4s)	Duration	Status
pull the file from github repo	2s	992ms	Passed
copy files from jenkins to ansible & k8s server	402ms	402ms	Failed

Again recreate script then click apply and save next run job

The screenshot shows the Jenkins interface with the 'Configuration' tab selected. The 'Pipeline' section is active, showing the following Groovy script:

```
node{
    stage('pull the file from github repo'){
        git branch: 'main', url: 'https://github.com/Arisivasankar7/devops-project.git'
    }
    stage('copy files from jenkins to ansible & k8s server'){
        sh 'scp /var/lib/jenkins/workspace/dev/Dockerfile ubuntu@172.31.31.118:/home/ubuntu/'
        sh 'scp /var/lib/jenkins/workspace/devops-project/index.html ubuntu@172.31.31.118:/home/ubuntu/'
        sh 'scp /var/lib/jenkins/workspace/devops-project/ansible.yml ubuntu@172.31.31.118:/home/ubuntu/'
        sh 'scp /var/lib/jenkins/workspace/devops-project/deployment.yml ubuntu@172.31.19.214:/home/ubuntu/'
        sh 'scp /var/lib/jenkins/workspace/devops-project/service.yml ubuntu@172.31.19.214:/home/ubuntu/'
    }
}
```

The screenshot shows the Jenkins Stage View interface. On the left, there's a sidebar with options like Dashboard, Configure, Delete Pipeline, Full Stage View, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. Below that is a 'Builds' section with a filter and a list of recent builds: #4 (4:39 pm), #3 (4:34 pm), and #2 (4:22 pm). The main area is titled 'Stage view' and displays a grid of stages. The stages are: 'pull the file from github repo' (1s), 'copy files from jenkins to ansible & k8s server' (1s), 'Dec 14 22:09 No Changes' (594ms), 'Dec 14 22:04 No Changes' (992ms failed), 'Dec 14 21:52 1 commit' (1s), and 'Dec 14 16:00 No Changes' (4s). A progress bar at the top indicates an average stage time of 1s and an average full run time of ~4s.

Then check my Ansible server for those files that are transferred.

Ansible server:

```
ubuntu@ip-172-31-31-118:~$ ls
Dockerfile  ansible.yml  index.html
ubuntu@ip-172-31-31-118:~$
```

Master server:

```
ubuntu@ip-172-31-19-214:~$ ls
deployment.yml  kubectl.sha256  service.yml
ubuntu@ip-172-31-19-214:~$
```

Create docker image with tag script:

The screenshot shows the Jenkins Configuration screen for a project named 'devops-project'. On the left, there are tabs for General, Advanced Project Options, and Pipeline. The Pipeline tab is selected. The right side shows the 'Definition' section with a dropdown set to 'Pipeline script'. Below it is a code editor containing a Groovy script:

```

3   git branch: 'main', url: 'https://github.com/Arisilvasankar7/devops-project.git'
4 }
5 +
6 stage('copy files from jenkins to ansible & k8s server'){
7     sh 'scp /var/lib/jenkins/workspace/devops-project/Dockerfile ubuntu@172.31.31.118:/home/ubuntu/'
8     sh 'scp /var/lib/jenkins/workspace/devops-project/index.html ubuntu@172.31.31.118:/home/ubuntu/'
9     sh 'scp /var/lib/jenkins/workspace/devops-project/ansible.yml ubuntu@172.31.31.118:/home/ubuntu/'
10    sh 'scp /var/lib/jenkins/workspace/devops-project/deployment.yml ubuntu@172.31.19.214:/home/ubuntu/'
11 }
12 +
13 stage("create docker Image with tag"){
14     sh 'ssh ubuntu@172.31.31.118 cd /home/ubuntu'
15     sh 'ssh ubuntu@172.31.31.118 docker image build -t sivasankarr/$JOB_NAME:v1.$BUILD_ID .'
16     sh 'ssh ubuntu@172.31.31.118 docker image build -t sivasankarr/$JOB_NAME:latest .'
17 }
18 }
19 }
```

Successfully run those scripts.

The screenshot shows the Jenkins Pipeline dashboard for the 'devops-project'. On the left, there's a sidebar with options like 'Configure', 'Delete Pipeline', 'Full Stage View', 'Stages', 'Rename', 'Pipeline Syntax', and 'GitHub Hook Log'. Below that is a 'Builds' section listing recent builds: #5 (4:48 pm), #4 (4:39 pm), #3 (4:34 pm), #2 (4:22 pm), and #1 (10:30 am). The main area displays a pipeline stage grid with three stages: 'pull the file from github repo', 'copy files from jenkins to ansible & k8s server', and 'create docker image with tag'. The first two stages have average times of 1s and 2s respectively. The third stage has an average time of 26s. A specific build row for build #3 is highlighted in red, showing a duration of 992ms for the first stage and 402ms for the second stage, both of which failed. The third stage for this build is not yet completed.

The screenshot shows a terminal window on an AWS instance. The user is listing Docker images with the command 'ls' and then checking the details of the 'devops-project' image with 'docker images'. The output shows two images: one tagged 'latest' and another tagged 'v1.5', both created about a minute ago and sized at 186MB.

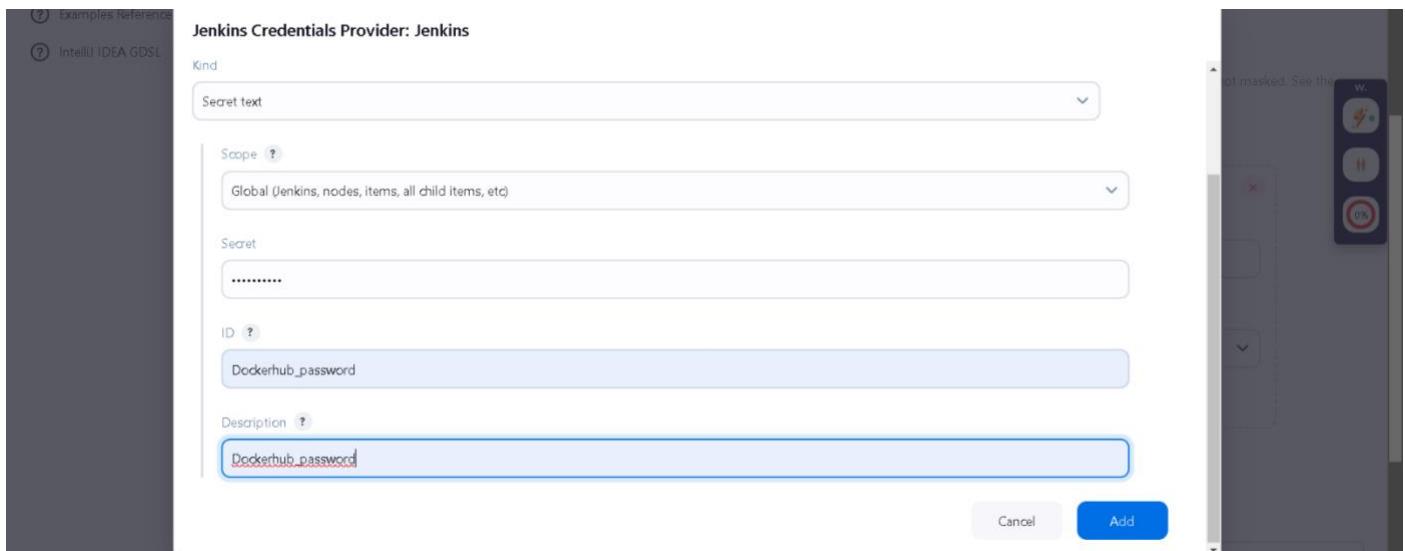
```
ubuntu@ip-172-31-31-118:~$ ls
Dockerfile ansible.yml index.html
ubuntu@ip-172-31-31-118:~$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
sivasankarrr/devops-project    latest   798fd6ace579  About a minute ago  186MB
sivasankarrr/devops-project    v1.5    798fd6ace579  About a minute ago  186MB
ubuntu@ip-172-31-31-118:~$
```

The next step is to push image to docker hub.

Some configura do that dockerhub password before push image to dockerhub. just because update raw dockerhub password in script it not safe that's why I do that some credentials for dockerhub

The screenshot shows the 'Pipeline Syntax' configuration page for the 'devops-project'. It includes sections for 'Global Variables Reference', 'Online Documentation', 'Examples Reference', and 'IntelliJ IDEA GDSL'. The main area shows a 'Sample Step' with the code 'withCredentials: Bind credentials to variables'. Below this is a 'withCredentials' block with a note about secret masking. Under 'Bindings', there's a 'Secret text' field and a 'Variable' dropdown containing 'Dockerhub_password'.

Next step is give the dockerhub password then create a id name and then click the apply button.



Go back stage script page. Create a script for push image to dockerhub and dockerhub login in ansible server before click apply.

```
sh 'scp /var/lib/jenkins/workspace/devops-project/deployment.yml ubuntu@172.31.19.214:/home/ubuntu/'  
sh 'scp /var/lib/jenkins/workspace/devops-project/service.yml ubuntu@172.31.19.214:/home/ubuntu/'  
stage('create docker image with tag'){  
    sh 'ssh ubuntu@172.31.31.118 cd /home/ubuntu'  
    sh 'ssh ubuntu@172.31.31.118 docker image build -t sivasankarrr/$JOB_NAME:v1.$BUILD_ID .'  
    sh 'ssh ubuntu@172.31.31.118 docker image build -t sivasankarrr/$JOB_NAME:latest .'  
}  
stage('push image to the Dockerhub'){  
    withCredentials([string(credentialsId: 'Dockerhub_password', variable: 'Dockerhub_password')]) {  
        sh 'ssh ubuntu@172.31.31.118 docker login -u sivasankarrr -p ${Dockerhub_password}'  
        sh 'ssh ubuntu@172.31.31.118 docker image push sivasankarrr/$JOB_NAME:v1.$BUILD_ID'  
        sh 'ssh ubuntu@172.31.31.118 docker image push sivasankarrr/$JOB_NAME:latest'  
    }  
}
```

Then click the build now.

Stage	Average stage time
pull the file from github repo	1s
copy files from jenkins to ansible & k8s server	2s
create docker image with tag	14s
push image to the Dockerhub	18s

This is my dockerhub account and it's successfully push image to my dockerhub.

The screenshot shows the Docker Hub interface. At the top, there are tabs for Explore, Repositories (which is selected), Organizations, and Usage. A search bar says "Search Docker Hub" with a "ctrl+K" keyboard shortcut. Below the search bar, there's a dropdown for "sivasankarrr" and a "Create a repository" button. The main area displays a table of repositories:

Name	Last Pushed	Contains	Visibility	Scout
sivasankarrr/devops-project	4 minutes ago	IMAGE	Public	Inactive
sivasankarrr/dev	1 day ago		Public	Inactive
sivasankarrr/devops	6 days ago		Public	Inactive

On the right side of the screen, there's a decorative graphic of three interconnected icons: a red circle with a lock, a blue hexagon with a lock, and a green triangle with a lock.

The screenshot shows the Docker Hub repository page for "sivasankarrr/devops-project". The top navigation bar includes links for "General", "Code", "Builds", "Dockerfile", "Logs", "Actions", and "Tags". The main content area has sections for "Description" (with fields for "Add a description" and "Add a category"), "Docker commands" (with a command line input field containing "docker push sivasankarrr/devops-project:tagname"), and "Automated builds". The "Tags" section lists two tags: "latest" and "v1.6", both pushed 5 minutes ago. The "Automated builds" section explains how to connect GitHub or Bitbucket for automatic builds and provides an "Upgrade" button.

The next step is write the yaml script for three yaml file. The first yaml file use for run both yaml file I mean the first yaml file work by ansible playbook.

The screenshot shows a terminal window titled "MINGW64:/d/devops pipeline project". It contains an Ansible YAML configuration file with the following content:

```
name: Apply k8s configuration
hosts: node
become: yes
tasks:
  - name: Create new deployment
    command: kubectl apply -f deployment.yml
    register: result
    environment:
      KUBECONFIG: /home/ubuntu/.kube/config
    args:
      chdir: /home/ubuntu

  - name: Create new service
    command: kubectl apply -f service.yml
    register: result
    environment:
      KUBECONFIG: /home/ubuntu/.kube/config
    args:
      chdir: /home/ubuntu

  - name: restart deployment to pull the latest image
    command: kubectl rollout restart deployment devops-deploy
    register: result
    environment:
      KUBECONFIG: /home/ubuntu/.kube/config
    args:
      chdir: /home/ubuntu
```

Deployment.yaml:

```
MINGW64:/d/devops pipeline project
kind: Deployment
apiVersion: apps/v1
metadata:
  name: devops
spec:
  replicas: 3
  selector:
    matchLabels:
      app: devops
  template:
    metadata:
      labels:
        app: devops
    spec:
      containers:
        - name: devops
          image: sivasankarrr/dev
          imagePullPolicy: Always
          ports:
            - containerPort: 80
```

Service.yaml:

```
MINGW64:/d/devops pipeline project
kind: Service
apiVersion: v1
metadata:
  name: devops
  labels:
    app: devops
spec:
  ports:
    - port: 8080
      targetPort: 80
      nodePort: 31200
  selector:
    app: devops
  type: LoadBalancer
```

Push those all yaml file to server after it change some configura within ansible server.

```
ubuntu@ip-172-31-31-118:~$ ls
Dockerfile  ansible.yml  index.html
ubuntu@ip-172-31-31-118:~$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
sivasankarrr/devops-project latest   798fd6ace579  About a minute ago  186MB
sivasankarrr/devops-project v1.5    798fd6ace579  About a minute ago  186MB
ubuntu@ip-172-31-31-118:~$ cd /etc/ansible/
ubuntu@ip-172-31-31-118:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-31-118:/etc/ansible$ vi hosts
```

```
aws | ■ | Q Search [Alt+S]
[  ] | [  ] | [  ] | [  ] | Stockholm | ass@123-5566-1379 ▾

#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
#
# Ex 1: Ungrouped hosts, specify before any group headers:
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
#
# Ex 2: A collection of hosts belonging to the 'webservers' group:
[node]
172.31.19.214
## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110
```

The configuration finish those file and then it's create ssh key and copy master server private ip address within ansible server because if it don't connect to each server then it's never properly run that yaml file that's why it is must be copy those private ip .in ansible server.

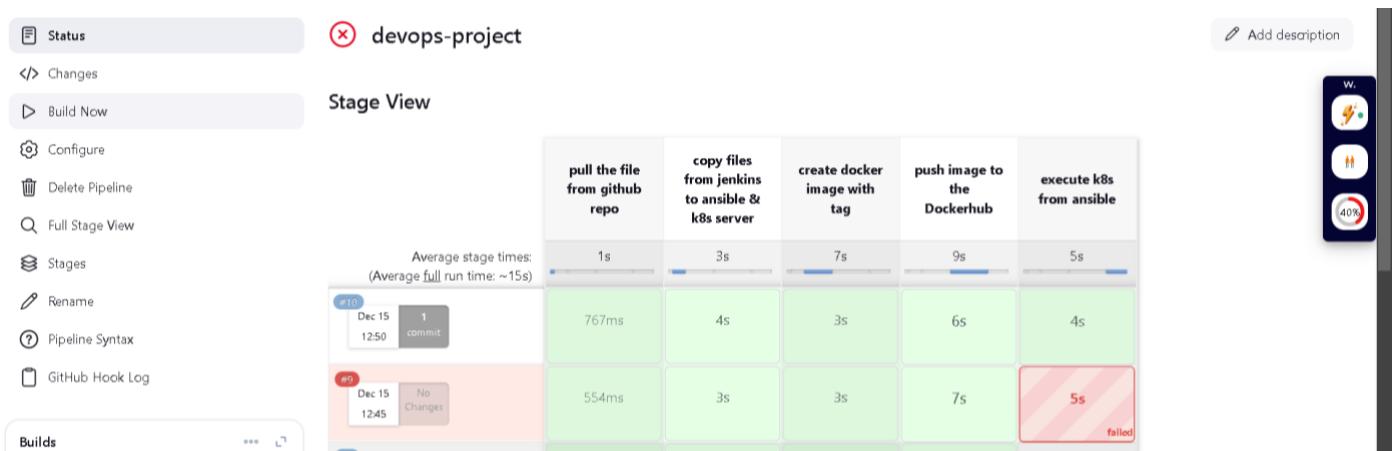
Push all three yaml file to github and then automatically trigger Jenkins script. After those edited three file store in ansible server and master machine.

```
arisiv@arisivasankar MINGW64 ~/d/devops pipeline project (main)
$ vi ansible.yml
arisiv@arisivasankar MINGW64 ~/d/devops pipeline project (main)
$ vi ansible.yml
arisiv@arisivasankar MINGW64 ~/d/devops pipeline project (main)
$ vi deployment.yml
arisiv@arisivasankar MINGW64 ~/d/devops pipeline project (main)
$ vi service.yml
arisiv@arisivasankar MINGW64 ~/d/devops pipeline project (main)
$ vi service.yml
arisiv@arisivasankar MINGW64 ~/d/devops pipeline project (main)
$ git add .
warning: In the working copy of 'ansible.yml', LF will be replaced by CRLF the next time Git touches it
warning: In the working copy of 'deployment.yml', LF will be replaced by CRLF the next time Git touches it
warning: In the working copy of 'service.yml', LF will be replaced by CRLF the next time Git touches it
arisiv@arisivasankar MINGW64 ~/d/devops pipeline project (main)
$ git commit -m "update files"
[main 26ca1c] update files
 3 files changed, 67 insertions(+)
arisiv@arisivasankar MINGW64 ~/d/devops pipeline project (main)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 957 bytes | 319.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Arisivasankar7/devops-project.git
  9340e81..26ca1c main -> main
arisiv@arisivasankar MINGW64 ~/d/devops pipeline project (main)
$
```

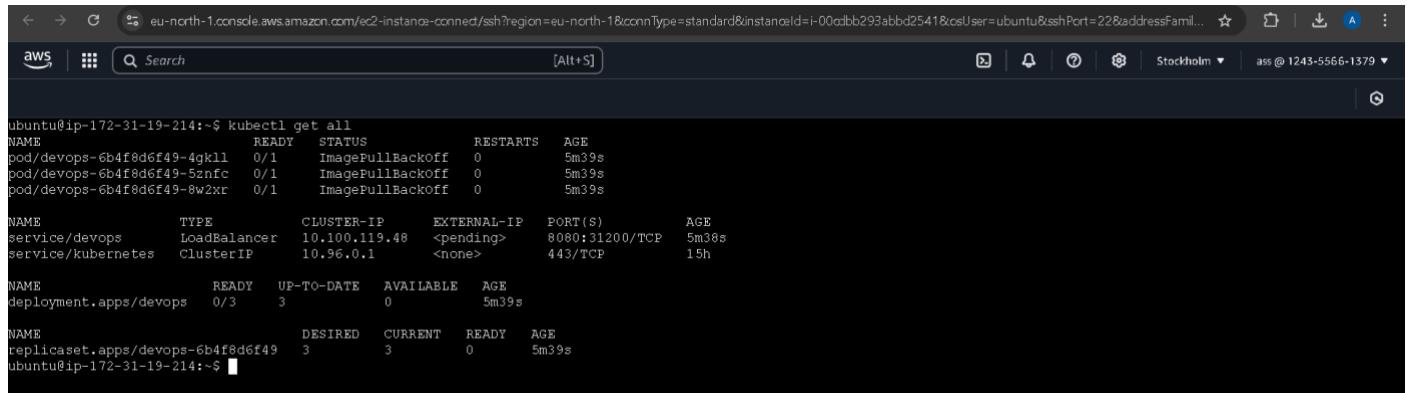
Successfully push all files.

Finally run the last stage script. This script do run the ansible playbook ansible.yml file

```
stage('execute k8s from ansible'){
    sh 'ssh ubuntu@172.31.28.53 cd /home/ubuntu'
    sh 'ssh ubuntu@172.31.28.53 ansible-playbook ansible.yml'
}
```



Finally job is successfully run then go back the master server. after type this command (**kubectl get all**)



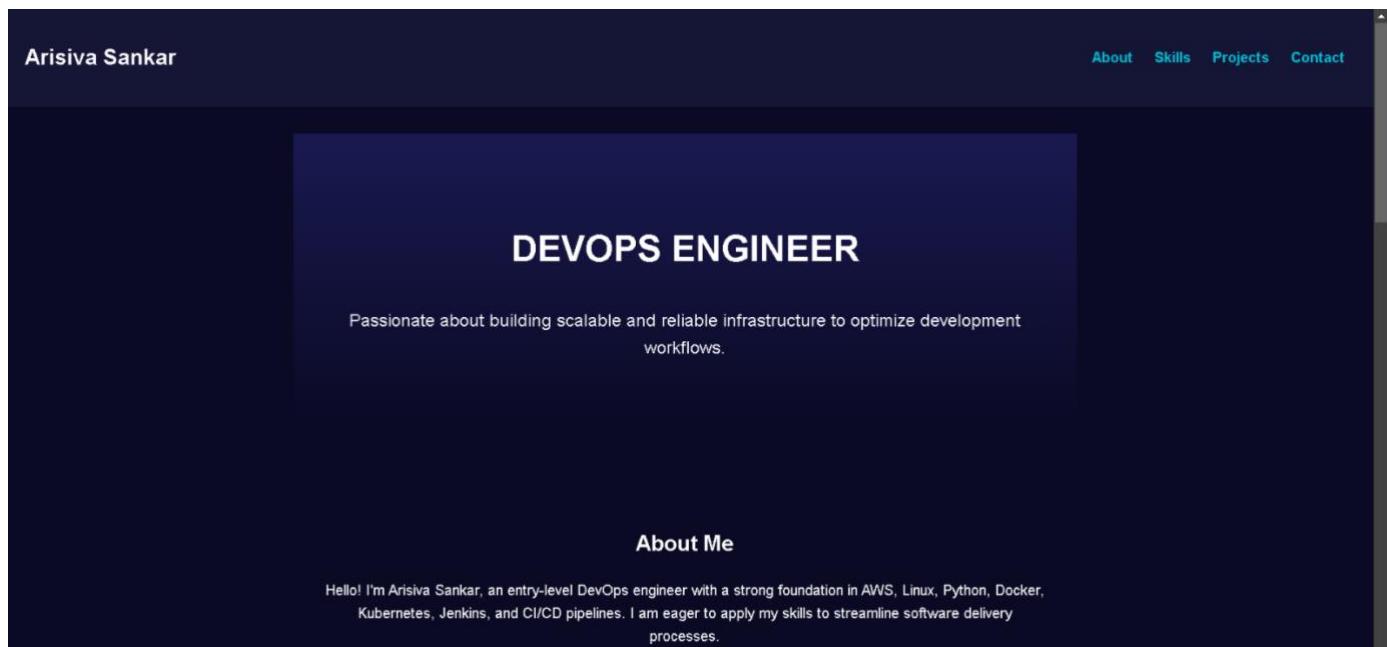
```
ubuntu@ip-172-31-19-214:~$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/devops-6b4f8d6f49-4gkll        0/1     ImagePullBackOff  0          5m39s
pod/devops-6b4f8d6f49-5znfc        0/1     ImagePullBackOff  0          5m39s
pod/devops-6b4f8d6f49-8w2xr        0/1     ImagePullBackOff  0          5m39s

NAME                  TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/devops       LoadBalancer  10.100.119.48   <pending>      8080:31200/TCP  5m38s
service/kubernetes  ClusterIP   10.96.0.1      <none>         443/TCP     15h

NAME                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/devops  0/3      3           0          5m39s

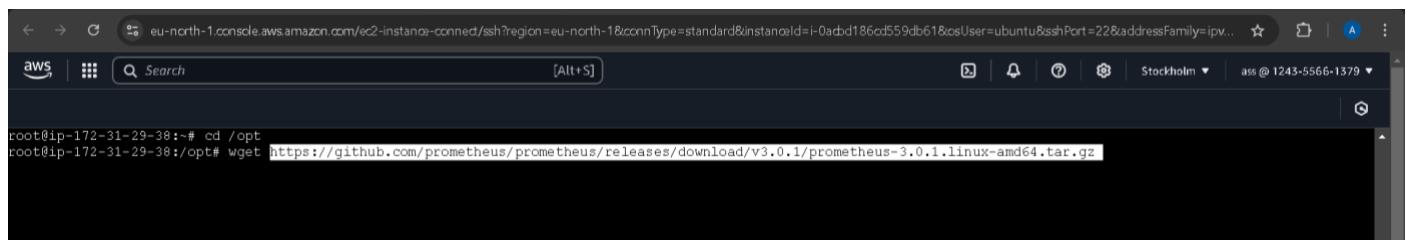
NAME                DESIRED  CURRENT  READY   AGE
replicaset.apps/devops-6b4f8d6f49  3        3           0          5m39s
ubuntu@ip-172-31-19-214:~$
```

Then finally check the project output through browser.



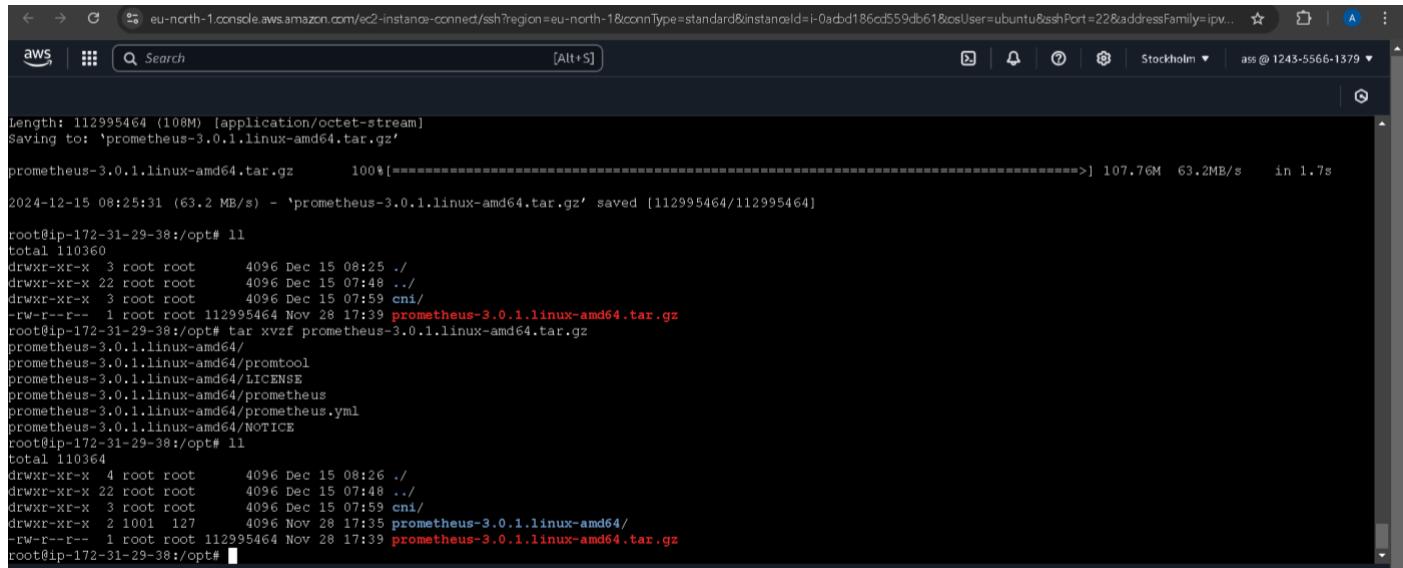
A screenshot of a resume website for Arisiva Sankar. The top navigation bar includes links for About, Skills, Projects, and Contact. The main heading is "DEVOPS ENGINEER". Below it is a brief description: "Passionate about building scalable and reliable infrastructure to optimize development workflows." A large dark blue rectangular area contains the heading "About Me". Underneath, there is a bio: "Hello! I'm Arisiva Sankar, an entry-level DevOps engineer with a strong foundation in AWS, Linux, Python, Docker, Kubernetes, Jenkins, and CI/CD pipelines. I am eager to apply my skills to streamline software delivery processes." The overall design is clean and professional.

The next one is Prometheus installation. The Prometheus use for metrics monitoring purpose and it helpful tool because sometimes we don't know how many memory we are using for server, network traffic, and mostly use this tool for cpu utilization because we don't down cpu contition then it come biggest problem because we don't know which time server down that's why it's important.



```
root@ip-172-31-29-38:~# cd /opt
root@ip-172-31-29-38:/opt# wget https://github.com/prometheus/prometheus/releases/download/v3.0.1/prometheus-3.0.1.linux-amd64.tar.gz
```

Successfully install that Prometheus tool in server then xtrack Prometheus file next run the Prometheus.



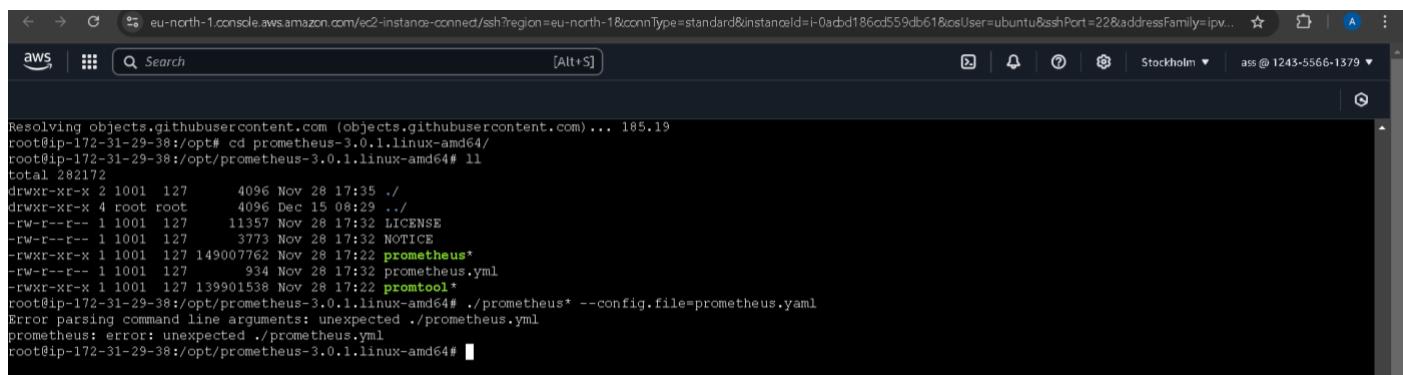
```
Length: 112995464 (108M) [application/octet-stream]
Saving to: 'prometheus-3.0.1.linux-amd64.tar.gz'

prometheus-3.0.1.linux-amd64.tar.gz      100%[=====] 107.7GM 63.2MB/s    in 1.7s

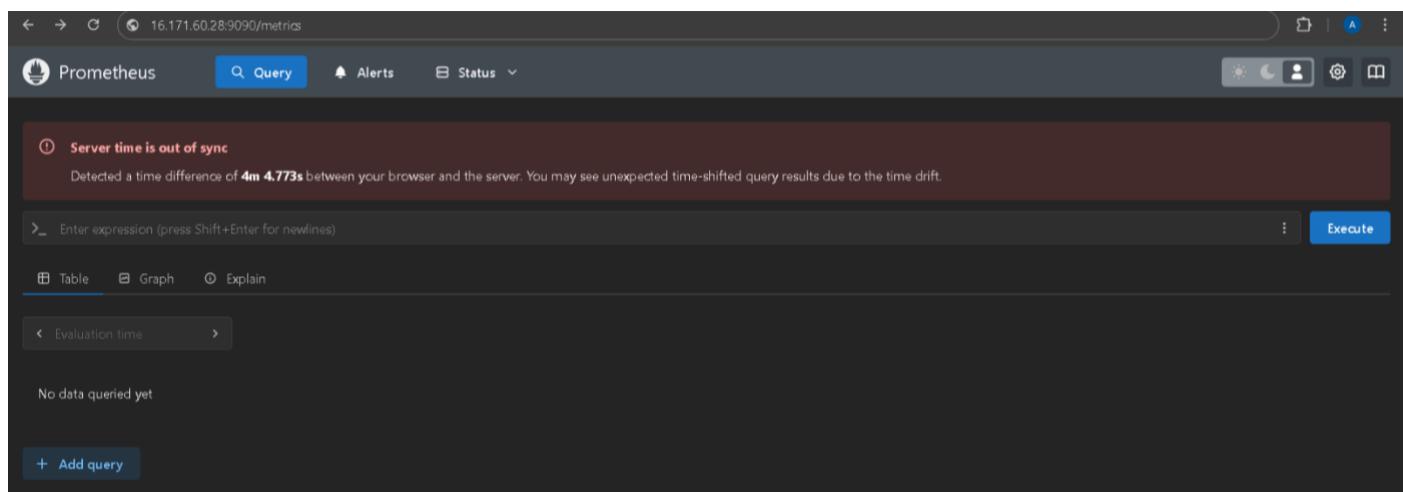
2024-12-15 08:25:31 (63.2 MB/s) - 'prometheus-3.0.1.linux-amd64.tar.gz' saved [112995464/112995464]

root@ip-172-31-29-38:/opt# ll
total 110360
drwxr-xr-x  3 root root   4096 Dec 15 08:25 .
drwxr-xr-x 22 root root   4096 Dec 15 07:48 ..
drwxr-xr-x  3 root root   4096 Dec 15 07:59 cni/
-rw-r--r--  1 root root 112995464 Nov 28 17:39 prometheus-3.0.1.linux-amd64.tar.gz
root@ip-172-31-29-38:/opt# tar xvzf prometheus-3.0.1.linux-amd64.tar.gz
prometheus-3.0.1.linux-amd64/
prometheus-3.0.1.linux-amd64/promtool
prometheus-3.0.1.linux-amd64/LICENSE
prometheus-3.0.1.linux-amd64/prometheus
prometheus-3.0.1.linux-amd64/prometheus.yml
prometheus-3.0.1.linux-amd64/NOTICE
root@ip-172-31-29-38:/opt# ll
total 110364
drwxr-xr-x  4 root root   4096 Dec 15 08:26 .
drwxr-xr-x 22 root root   4096 Dec 15 07:48 ..
drwxr-xr-x  3 root root   4096 Dec 15 07:59 cni/
drwxr-xr-x  2 1001 127   4096 Nov 28 17:35 prometheus-3.0.1.linux-amd64/
-rw-r--r--  1 root root 112995464 Nov 28 17:39 prometheus-3.0.1.linux-amd64.tar.gz
root@ip-172-31-29-38:/opt#
```

Run the Prometheus.



```
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.19
root@ip-172-31-29-38:/opt# cd prometheus-3.0.1.linux-amd64/
root@ip-172-31-29-38:/opt/prometheus-3.0.1.linux-amd64# ll
total 282172
drwxr-xr-x  2 1001 127   4096 Nov 28 17:35 .
drwxr-xr-x  4 root root   4096 Dec 15 08:29 ..
-rw-r--r--  1 1001 127   11357 Nov 28 17:32 LICENSE
-rw-r--r--  1 1001 127   3773 Nov 28 17:32 NOTICE
-rwxr-xr-x  1 1001 127 149007762 Nov 28 17:22 prometheus*
-rw-r--r--  1 1001 127   934 Nov 28 17:32 prometheus.yml
-rwxr-xr-x  1 1001 127 139901538 Nov 28 17:22 promtool*
root@ip-172-31-29-38:/opt/prometheus-3.0.1.linux-amd64# ./prometheus --config.file=prometheus.yml
error parsing command line arguments: unexpected ./prometheus.yml
prometheus: error: unexpected ./prometheus.yml
root@ip-172-31-29-38:/opt/prometheus-3.0.1.linux-amd64#
```



⌚ Server time is out of sync
Detected a time difference of **4m 4.773s** between your browser and the server. You may see unexpected time-shifted query results due to the time drift.

Enter expression (press Shift+Enter for newlines)

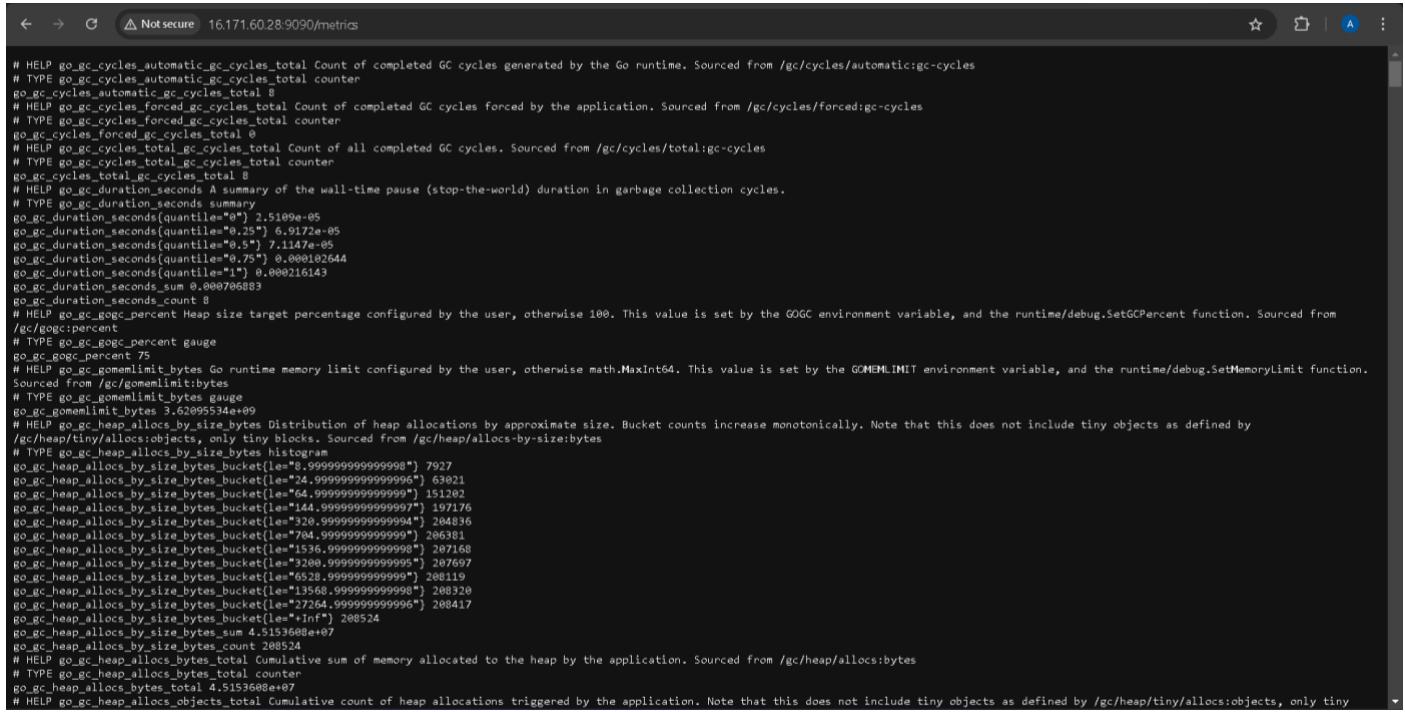
Table Graph Explain

Evaluation time < Evaluation time >

No data queried yet

+ Add query

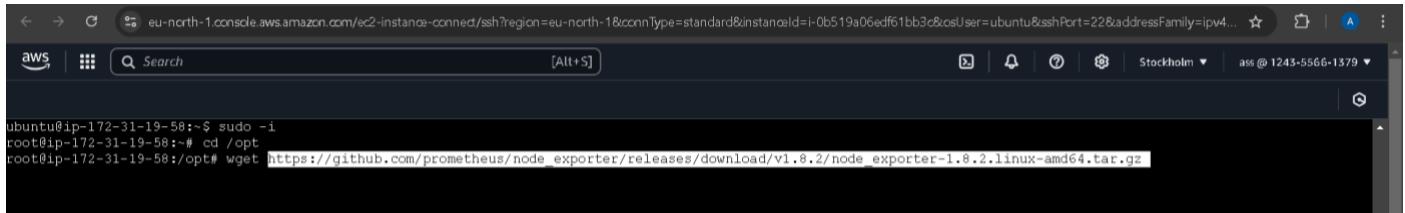
This is the metrics.



The screenshot shows a browser window displaying a list of Go runtime metrics. The URL is 16.171.60.28:9090/metrics. The metrics listed include:

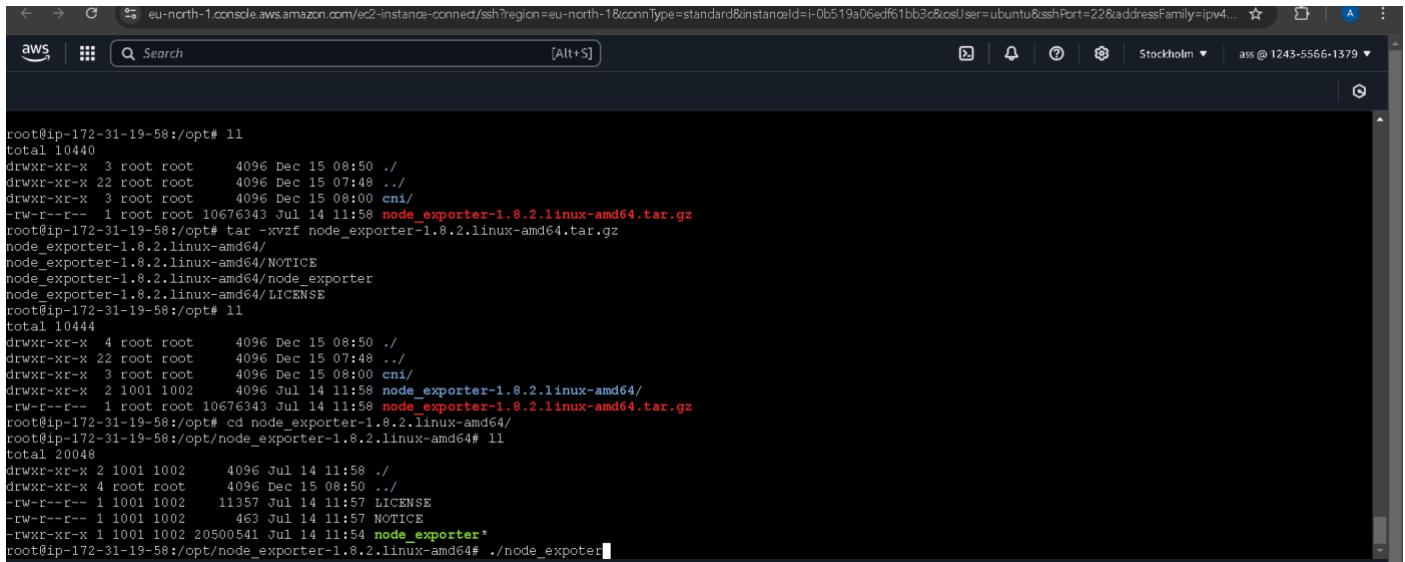
- # HELP go_gc_cycles_automatic_gc_cycles_total Count of completed GC cycles generated by the Go runtime. Sourced from /gc/cycles/automatic:gc-cycles
- # TYPE go_gc_cycles_automatic_gc_cycles_total counter
- go_gc_cycles_automatic_gc_cycles_total 8
- # HELP go_gc_cycles_forced_gc_cycles_total Count of completed GC cycles forced by the application. Sourced from /gc/cycles/forced:gc-cycles
- # TYPE go_gc_cycles_forced_gc_cycles_total counter
- go_gc_cycles_forced_gc_cycles_total 0
- # HELP go_gc_cycles_total_gc_cycles_total Count of all completed GC cycles. Sourced from /gc/cycles/total:gc-cycles
- # TYPE go_gc_cycles_total_gc_cycles_total counter
- go_gc_cycles_total_gc_cycles_total 8
- # HELP go_gc_duration_seconds A summary of the wall-time pause (stop-the-world) duration in garbage collection cycles.
- # TYPE go_gc_duration_seconds summary
- go_gc_duration_seconds{quantile="0"} 2.5189e-05
- go_gc_duration_seconds{quantile="0.25"} 6.9172e-05
- go_gc_duration_seconds{quantile="0.5"} 7.1147e-05
- go_gc_duration_seconds{quantile="0.75"} 0.000102644
- go_gc_duration_seconds{quantile="1"} 0.000216143
- go_gc_duration_seconds_sum 0.000796683
- go_gc_duration_seconds_count 8
- # HELP go_gc_gogg_percent Heap size target percentage configured by the user, otherwise 100. This value is set by the GOGC environment variable, and the runtime/debug.SetGCPerc function. Sourced from /gc/gogg:percent
- # TYPE go_gc_gogg_percent gauge
- go_gc_gogg_percent 75
- # HELP go_gc_gomemlimit_bytes Go runtime memory limit configured by the user, otherwise math.MaxInt64. This value is set by the GOMEMLIMIT environment variable, and the runtime/debug.SetMemoryLimit function. Sourced from /gc/gomemlimit:bytes
- # TYPE go_gc_gomemlimit_bytes gauge
- go_gc_gomemlimit_bytes 3.62095534e+99
- # HELP go_gc_heap_allocs_by_size_bytes Distribution of heap allocations by approximate size. Bucket counts increase monotonically. Note that this does not include tiny objects as defined by /gc/heap/tiny/allocs:objects, only tiny blocks. Sourced from /gc/heap/allocs:bytes
- # TYPE go_gc_heap_allocs_by_size_bytes histogram
- go_gc_heap_allocs_by_size_bytes_bucket{[le="8.000000000000000e-09"]} 7937
- go_gc_heap_allocs_by_size_bytes_bucket{[le="24.99999999999999e-09"]} 63021
- go_gc_heap_allocs_by_size_bytes_bucket{[le="64.99999999999999e-09"]} 11282
- go_gc_heap_allocs_by_size_bytes_bucket{[le="144.99999999999997e-09"]} 197176
- go_gc_heap_allocs_by_size_bytes_bucket{[le="320.99999999999994e-09"]} 248436
- go_gc_heap_allocs_by_size_bytes_bucket{[le="704.9999999999999e-09"]} 206381
- go_gc_heap_allocs_by_size_bytes_bucket{[le="1536.9999999999998e-09"]} 287168
- go_gc_heap_allocs_by_size_bytes_bucket{[le="3200.999999999995e-09"]} 287697
- go_gc_heap_allocs_by_size_bytes_bucket{[le="6528.999999999999e-09"]} 288119
- go_gc_heap_allocs_by_size_bytes_bucket{[le="13568.999999999999e-09"]} 288328
- go_gc_heap_allocs_by_size_bytes_bucket{[le="27264.999999999996e-09"]} 288417
- go_gc_heap_allocs_by_size_bytes_bucket{[le="Inf"]} 208524
- go_gc_heap_allocs_by_size_bytes_sum 4.5153608e+07
- go_gc_heap_allocs_by_size_bytes_count 208524
- # HELP go_gc_heap_allocs_bytes_total Cumulative sum of memory allocated to the heap by the application. Sourced from /gc/heap/allocs:bytes
- # TYPE go_gc_heap_allocs_bytes_total counter
- go_gc_heap_allocs_bytes_total 4.5153608e+07
- # HELP go_gc_heap_allocs_objects_total Cumulative count of heap allocations triggered by the application. Note that this does not include tiny objects as defined by /gc/heap/tiny/allocs:objects, only tiny

The next installation is the node_exporter. the node exporter work is collating the slave server metrics then push those slave metrics to master server. that is node exporter concept.



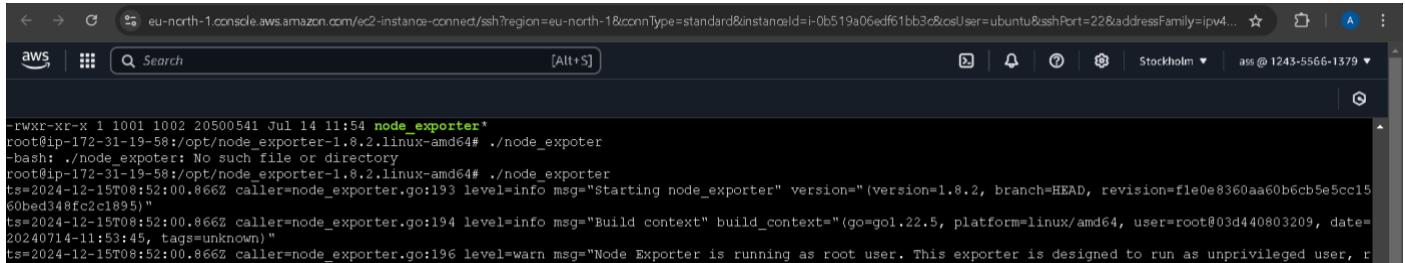
```
ubuntu@ip-172-31-19-58:~$ sudo -i
root@ip-172-31-19-58:~# cd /opt
root@ip-172-31-19-58:/opt# wget https://github.com/prometheus/node_exporter/releases/download/v1.8.2/node_exporter-1.8.2.linux-amd64.tar.gz
```

Then extract the node exporter file



```
root@ip-172-31-19-58:/opt# ll
total 10440
drwxr-xr-x  3 root root    4096 Dec 15 08:50 .
drwxr-xr-x 22 root root    4096 Dec 15 07:48 ..
drwxr-xr-x  3 root root    4096 Dec 15 08:00 cni/
-rw-r--r--  1 root root 10676343 Jul 14 11:58 node_exporter-1.8.2.linux-amd64.tar.gz
root@ip-172-31-19-58:/opt# tar -xvf node_exporter-1.8.2.linux-amd64.tar.gz
node_exporter-1.8.2.linux-amd64/
node_exporter-1.8.2.linux-amd64/NOTICE
node_exporter-1.8.2.linux-amd64/node_exporter
node_exporter-1.8.2.linux-amd64/LICENSE
root@ip-172-31-19-58:/opt# ll
total 10444
drwxr-xr-x  4 root root    4096 Dec 15 08:50 .
drwxr-xr-x 22 root root    4096 Dec 15 07:48 ..
drwxr-xr-x  3 root root    4096 Dec 15 08:00 cni/
drwxr-xr-x  2 1001 1002   4096 Jul 14 11:58 node_exporter-1.8.2.linux-amd64/
-rw-r--r--  1 root root 10676343 Jul 14 11:58 node_exporter-1.8.2.linux-amd64.tar.gz
root@ip-172-31-19-58:/opt# cd node_exporter-1.8.2.linux-amd64/
root@ip-172-31-19-58:/opt/node_exporter-1.8.2.linux-amd64# ll
total 20048
drwxr-xr-x  2 1001 1002   4096 Jul 14 11:58 .
drwxr-xr-x  4 root root   4096 Dec 15 08:50 ..
-rw-r--r--  1 1001 1002   11357 Jul 14 11:57 LICENSE
-rw-r--r--  1 1001 1002     463 Jul 14 11:57 NOTICE
-rwxr-xr-x  1 1001 1002 20500541 Jul 14 11:54 node_exporter*
root@ip-172-31-19-58:/opt/node_exporter-1.8.2.linux-amd64# ./node_exporter*
```

Finally run the node exporter.

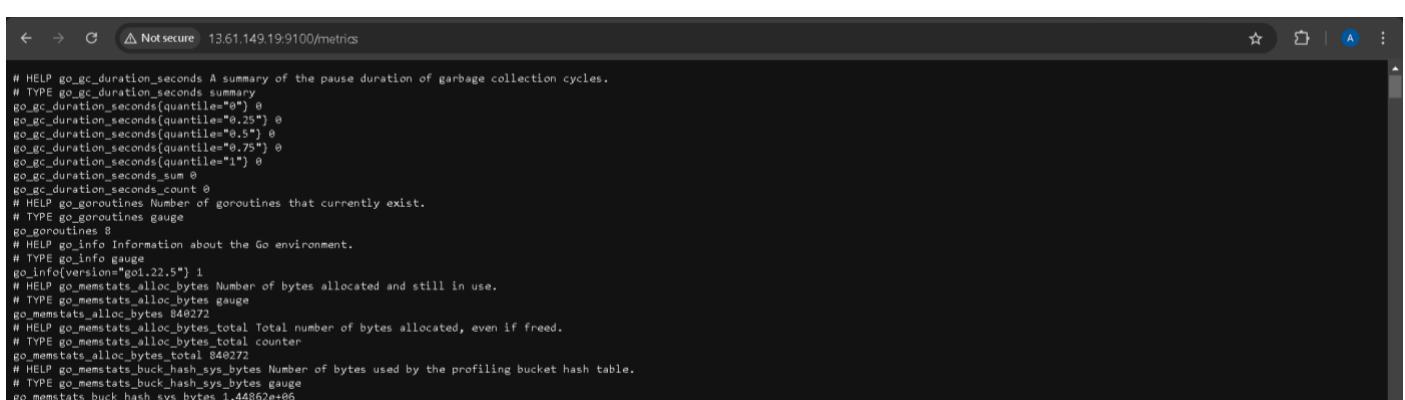


```
-rwxr-xr-x 1 1001 1002 20500541 Jul 14 11:54 node_exporter*
root@ip-172-31-19-58:/opt/node_exporter-1.8.2.linux-amd64# ./node_exporter
-bash: ./node_exporter: No such file or directory
root@ip-172-31-19-58:/opt/node_exporter-1.8.2.linux-amd64# ./node_exporter
ts=2024-12-15T08:52:00.866Z caller=node_exporter.go:193 level=info msg="Starting node_exporter" version="(version=1.8.2, branch=HEAD, revision=f1e0e8360aa60b6cb5e5cc1560bed348fc2c1895)"
ts=2024-12-15T08:52:00.866Z caller=node_exporter.go:194 level=info msg="Build context" build_context="(go=golang 22.5, platform=linux/amd64, user=root@03d440803209, date=2024-07-14 11:53:45, tags=unknown)"
ts=2024-12-15T08:52:00.866Z caller=node_exporter.go:196 level=warn msg="Node Exporter is running as root user. This exporter is designed to run as unprivileged user, r
```

Node exporter home page.

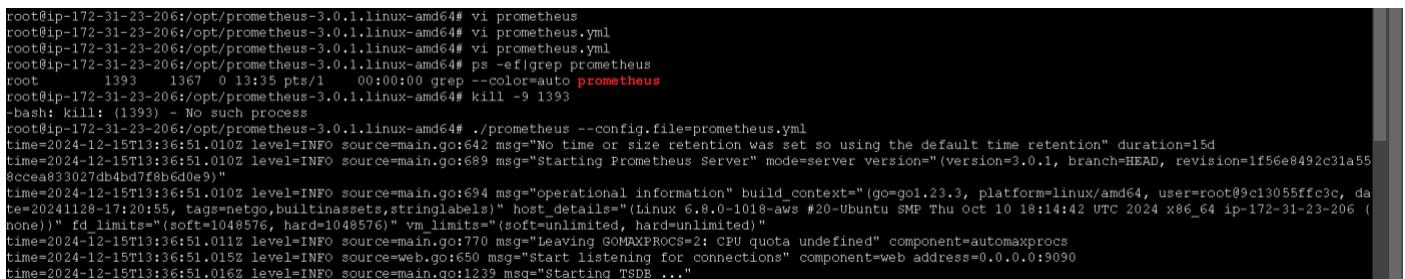


Slave server metrics.



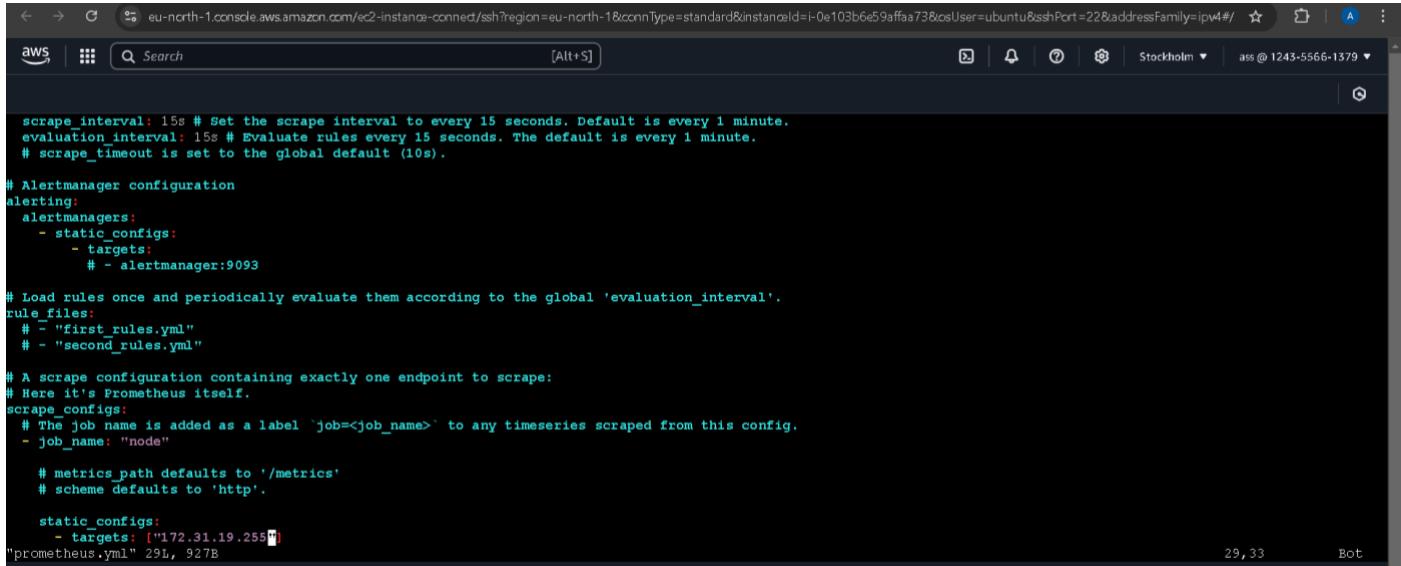
```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 8
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="golang 22.5"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 848272
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 848272
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.44862e+06
```

The next process is the connected the master server Prometheus to slave server node exporter.



```
root@ip-172-31-23-206:/opt/prometheus-3.0.1.linux-amd64# vi prometheus
root@ip-172-31-23-206:/opt/prometheus-3.0.1.linux-amd64# vi prometheus.yml
root@ip-172-31-23-206:/opt/prometheus-3.0.1.linux-amd64# vi prometheus.yml
root@ip-172-31-23-206:/opt/prometheus-3.0.1.linux-amd64# ps -ef|grep prometheus
root 1393 1367 0 13:35 pts/1 00:00:00 grep --color=auto prometheus
root@ip-172-31-23-206:/opt/prometheus-3.0.1.linux-amd64# kill -9 1393
-bash: kill: (1393) - No such process
root@ip-172-31-23-206:/opt/prometheus-3.0.1.linux-amd64# ./prometheus --config.file=prometheus.yml
time=2024-12-15T13:36:51.010Z level=INFO source=main.go:642 msg="No time or size retention was set so using the default time retention" duration=15d
time=2024-12-15T13:36:51.010Z level=INFO source=main.go:689 msg="Starting Prometheus Server" mode=server version="(version=3.0.1, branch=HEAD, revision=f1f56e8492c31a558cce833027db4bd7f86d0e9)"
time=2024-12-15T13:36:51.010Z level=INFO source=main.go:694 msg="operational information" build_context="(go=golang 22.3, platform=linux/amd64, user=root@9c13055ffcc3c, date=2024-11-28 17:20:55, tags=netgo,builtinaassets,stringlabels)" host_details="(Linux 6.8.0-1018-aws #20-Ubuntu SMP Thu Oct 10 18:14:42 UTC 2024 x86_64 ip-172-31-23-206 (none))" fd_limits="(soft=1048576, hard=1048576)" vm_limits="(soft=unlimited, hard=unlimited)"
time=2024-12-15T13:36:51.011Z level=INFO source=main.go:770 msg="Leaving GOMAXPROCS=2: CPU quota undefined" component=automaxprocs
time=2024-12-15T13:36:51.015Z level=INFO source=web.go:650 msg="Start listening for connections" component=web address=0.0.0.0:9090
time=2024-12-15T13:36:51.016Z level=INFO source=main.go:1239 msg="starting TSDB ..."
```

Set job name and targets:



```
eu-north-1.console.aws.amazon.com/ec2-instances-connected/ssh?region=eu-north-1&connType=standard&instanceId=i-0e103b6e59affaa73&osUser=ubuntu&sshPort=22&addressFamily=ipv4#/
```

```
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: "node"

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

    static_configs:
      - targets: ["172.31.19.255"]
```

29, 33 Bot

:wq!

Next type this command (ps -ef| grep Prometheus)

Kill command use for process kill

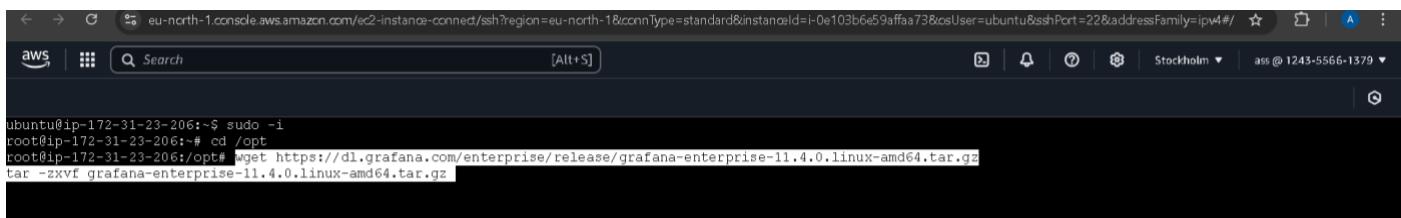
Again one more time run Prometheus.

./promrtheus --config.file=Prometheus.yml

Finally successfully connected Prometheus and node exporter together.

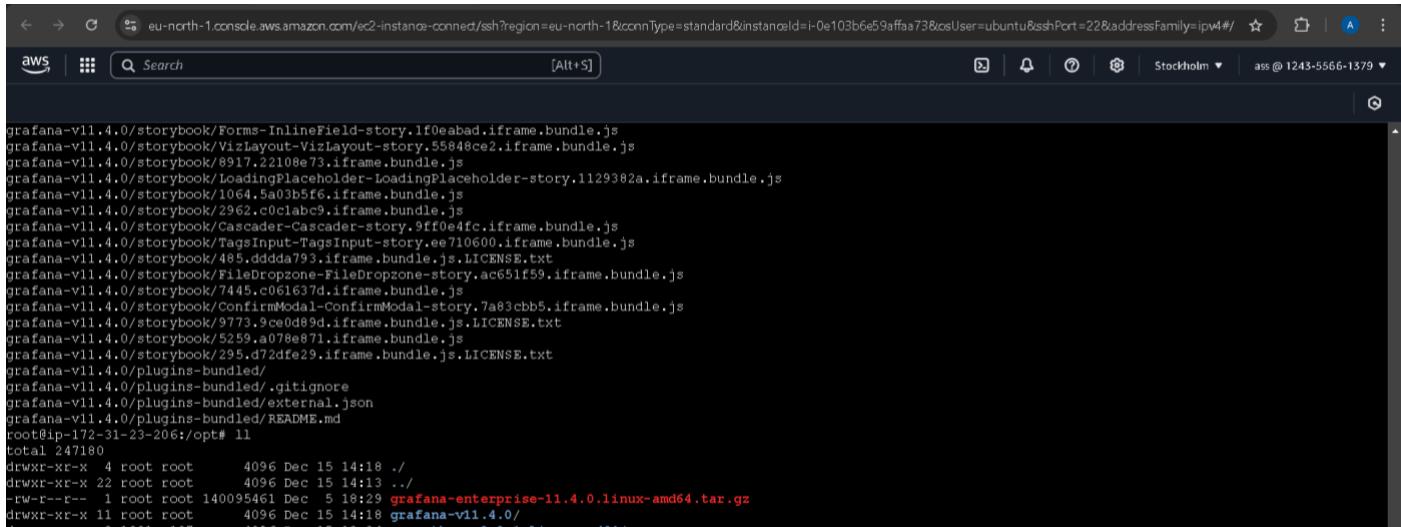
Then check the slave server metrics by master server because it already connected the booth tools

Finally install is Grafana tool.



```
eu-north-1.console.aws.amazon.com/ec2-instances-connected/ssh?region=eu-north-1&connType=standard&instanceId=i-0e103b6e59affaa73&osUser=ubuntu&sshPort=22&addressFamily=ipv4#/
```

```
ubuntu@ip-172-31-23-206:~$ sudo -i
root@ip-172-31-23-206:~# cd /opt
root@ip-172-31-23-206:/opt# wget https://dl.grafana.com/enterprise/release/grafana-enterprise-11.4.0.linux-amd64.tar.gz
tar -zxvf grafana-enterprise-11.4.0.linux-amd64.tar.gz
```



```
eu-north-1.console.aws.amazon.com/ec2-instances-connected/ssh?region=eu-north-1&connType=standard&instanceId=i-0e103b6e59affaa73&osUser=ubuntu&sshPort=22&addressFamily=ipv4#/
```

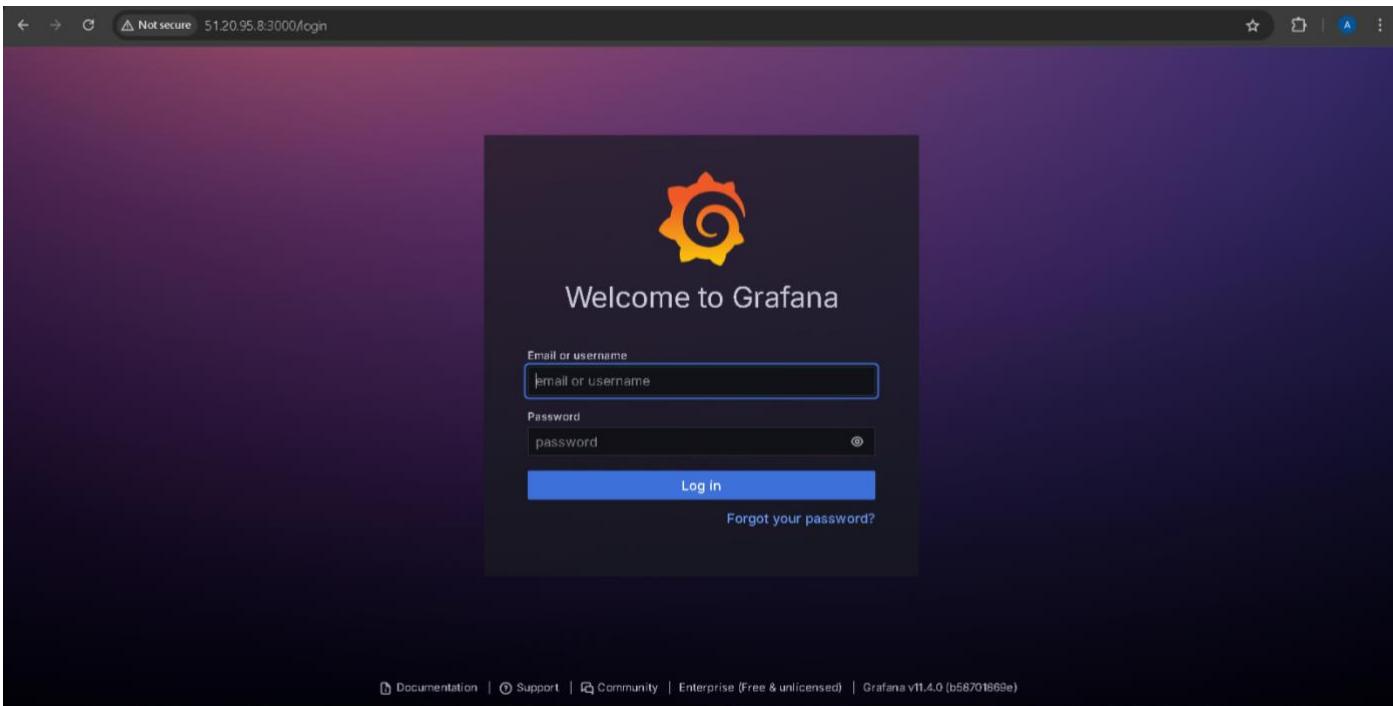
```
grafana-v11.4.0/storybook/Forms-InlineField-story.1f0eabed.inline.bundle.js
grafana-v11.4.0/storybook/VizLayout-story.55848ce2.inline.bundle.js
grafana-v11.4.0/storybook/8917.22108e73.inline.bundle.js
grafana-v11.4.0/storybook>LoadingPlaceholder-loadingPlaceholder-story.1129382a.inline.bundle.js
grafana-v11.4.0/storybook/1064.5a03b5f6.inline.bundle.js
grafana-v11.4.0/storybook/2962.c0c1abc9.inline.bundle.js
grafana-v11.4.0/storybook/Cascader-Cascader-story.9ff0e4fc.inline.bundle.js
grafana-v11.4.0/storybook/TagsInput-TagsInput-story.ee710600.inline.bundle.js
grafana-v11.4.0/storybook/485.dddd4793.inline.bundle.js.LICENSE.txt
grafana-v11.4.0/storybook/FileDropzone-FileDropzone-story.ac651f59.inline.bundle.js
grafana-v11.4.0/storybook/7445.c061637d.inline.bundle.js
grafana-v11.4.0/storybook/ConfirmModal-ConfirmModal-story.7a83cbb5.inline.bundle.js
grafana-v11.4.0/storybook/9773.9ce0d89d.inline.bundle.js.LICENSE.txt
grafana-v11.4.0/storybook/5259.a078e871.inline.bundle.js
grafana-v11.4.0/storybook/295.d72dfe29.inline.bundle.js.LICENSE.txt
grafana-v11.4.0/plugins-bundled/
grafana-v11.4.0/plugins-bundled/.gitignore
grafana-v11.4.0/plugins-bundled/external.json
grafana-v11.4.0/plugins-bundled/README.md
root@ip-172-31-23-206:/opt# ls
total 247180
drwxr-xr-x  4 root root   4096 Dec 15 14:18 .
drwxr-xr-x 22 root root   4096 Dec 15 14:13 ..
-rw-r--r--  1 root root 140095461 Dec  5 18:29 grafana-enterprise-11.4.0.linux-amd64.tar.gz
drwxr-xr-x 11 root root   4096 Dec 15 14:18 grafana-v11.4.0/
```

Xtrack Grafana file then gone bin path

```
eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh?regi... Stockholm ass @ 1243-5566-1379

drwxr-xr-x  4 root root      4096 Dec 15 14:18 ./
drwxr-xr-x 22 root root      4096 Dec 15 14:13 ../
-rw-r--r--  1 root root 140095461 Dec  5 18:29 grafana-enterprise-11.4.0.linux-amd64.tar.gz
drwxr-xr-x 11 root root      4096 Dec 15 14:21 grafana-v11.4.0/
drwxr-xr-x  3 1001 127      4096 Dec 15 13:34 prometheus-3.0.1.linux-amd64/
-rw-r--r--  1 root root 112995464 Nov 28 17:39 prometheus-3.0.1.linux-amd64.tar.gz
z
root@ip-172-31-23-206:/opt# cd grafana-v11.4.0/
root@ip-172-31-23-206:/opt/grafana-v11.4.0# ll
total 108
drwxr-xr-x 11 root root 4096 Dec 15 14:21 ./
drwxr-xr-x  4 root root 4096 Dec 15 14:18 ../
-rw-r--r--  1 root root 6188 Dec  4 21:35 Dockerfile
-rw-r--r--  1 root root 12155 Dec  4 21:36 LICENSE
-rw-r--r--  1 root root 105 Dec  4 21:35 NOTICE.md
-rw-r--r--  1 root root 3261 Dec  4 21:35 README.md
-rw-r--r--  1 root root 8 Dec  4 21:36 VERSION
drwxr-xr-x  2 root root 4096 Dec  4 21:43 bin/
drwxr-xr-x  3 root root 4096 Dec  4 21:35 conf/
drwxr-xr-x  3 root root 4096 Dec 15 14:18 docs/
drwxr-xr-x  2 root root 4096 Dec  4 21:39 npm-artifacts/
drwxr-xr-x  6 root root 4096 Dec 15 14:18 packaging/
drwxr-xr-x  2 root root 4096 Dec  4 21:35 plugins-bundled/
drwxr-xr-x 17 root root 4096 Dec  4 21:41 public/
drwxr-xr-x  8 root root 32768 Dec  4 21:39 storybook/
drwxr-xr-x  2 root root 4096 Dec 15 14:21 tools/
root@ip-172-31-23-206:/opt/grafana-v11.4.0# 
```

```
root@ip-172-31-23-206:/opt/grafana-v11.4.0/bin# ll
total 239996
drwxr-xr-x  2 root root 4096 Dec  4 21:43 ./
drwxr-xr-x 11 root root 4096 Dec 15 14:21 ../
-rw-r--r--  1 root root 241729232 Dec  4 21:43 grafana*
-rw-r--r--  1 root root 2004288 Dec  4 21:43 grafana-cli*
-rw-r--r--  1 root root 2004320 Dec  4 21:43 grafana-server*
root@ip-172-31-23-206:/opt/grafana-v11.4.0/bin# 
```



Checking the Prometheus Datasource:

The screenshot shows the Grafana interface. On the left, there's a sidebar with icons for General / Home, Dashboards, Browse, and other settings. The main area displays the 'Welcome to Grafana' panel. Below it, under 'Data sources', there's a section titled 'TUTORIAL DATA SOURCE AND DASHBOARDS Grafana fundamentals'. It includes a brief description: 'Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.' To the right of this panel are three cards: 'COMPLETE Add your first data source' with a database icon, 'COMPLETE Create your dashboard' with a dashboard icon, and 'Learn how in the docs' with a link icon.

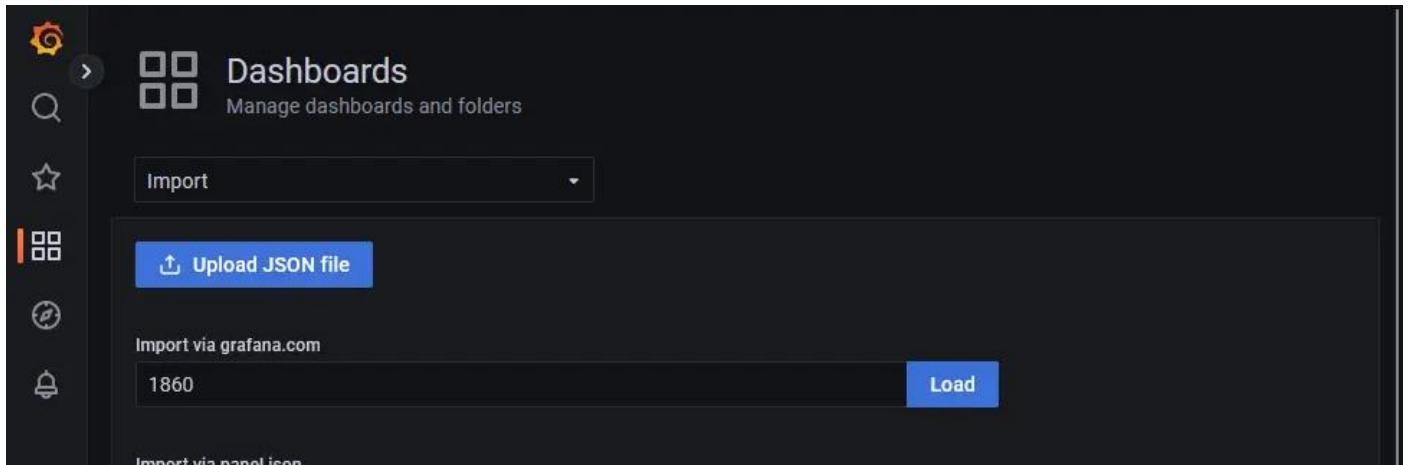
Select the datasources.

This screenshot shows the 'Data Sources / Prometheus-1' configuration page. At the top, it says 'Type: Prometheus'. Below that is a 'Settings' dropdown menu. Under 'HTTP', there are fields for 'URL' (set to 'http://localhost:9090'), 'Allowed cookies' (with a placeholder 'New tag (enter key to add)'), and 'Timeout' (with a placeholder 'Timeout in seconds').

The screenshot shows the Grafana General / Home page again. The sidebar has 'General / Home' selected. The main area displays the 'Welcome to Grafana' panel. Below it, under 'Dashboards', there's a card with the text 'Remove this panel'.

Upload the dashboard file. Then finish all configuration and then create a Grafana dashboard for the master server

metrics



Finally, create a dashboard.



THE END