

DEVOPS Capstone Projects-02

Assignment Submitted By:-Hitesh Chauhan

Course Offered: -Advanced Cloud Computing and Devops

Assignment By: -Intellipaat

Trainer: -Kumar

Date Of Submission: -24/03/2025

CAPSTONE PROJECTS

You are hired as a DevOps engineer for Analytics Pvt Ltd. This company is a product-based organization which uses Docker for their containerization needs within the company. The final product received a lot of traction in the first few weeks of launch. Now with the increasing demand, the organization needs to have a platform for automating deployment, scaling, and operations of application containers across clusters of hosts, As a DevOps engineer, you need implement a DevOps life cycle, such that all the requirements are implemented without any change in the Docker containers in the testing environment.

Up until now, this organization used to follow a monolithic architecture with just 2 developers.

The product is present on

<https://github.com/hshar/website.git>

Following are the specifications of life-cycle:

1. Git workflow should be implemented. Since the company follows monolithic architecture of Development you need to take care of version control. The release should happen only on 25th of every month.
 2. Code build should be triggered once the commits are made in the master Branch.
 3. The code should be containerized with the help of the Docker file, The Dockerfile should be built every time if there is a push to Git-Hub. Create a custom Docker image using a Dockerfile.
 4. As per the requirement in the production server, you need to use the Kubernetes cluster and the containerized code from Docker hub should be deployed with 2 replicas. Create a NodePort service and configure the same for port 30008.
 5. Create a Jenkins pipeline script to accomplish the above task.
 6. For configuration management of the infrastructure, you need to deploy the configuration on the servers to install necessary software and configurations.
 7. Using Terraform accomplish the task of infrastructure creation in the AWS cloud provider.
-

Prerequisites

1. Infrastructure Setup

- Ensure that you have a server (EC2 instance) to install and run Jenkins.
- Install a configuration management tool (**Ansible**) to automate software installation.
- Ensure that Docker is installed on the machine running Jenkins.
- Ensure that Terraform is installed on the machine.

2. Version Control System

- Set up a **GitHub repository** for the source code.

3. Jenkins Installation and Configuration

- Install **Jenkins** on the server.
- Install necessary plugins:
 - **Pipeline Plugin**
- Configure **webhook integration** between GitHub and Jenkins to trigger builds on commits.

4. Docker Setup

- Ensure **Docker Engine** is installed on the Jenkins server.
- Verify that the user running Jenkins has permissions to execute Docker commands.

5. Web Application Code Placement

- Ensure that the application code is placed in `/var/www/html` inside the container as per the requirement.

6. Set Up Kubernetes Cluster

ARCHITECTURE OF CONFIGURATION MANAGEMENT TOOL.

Architectural Advice

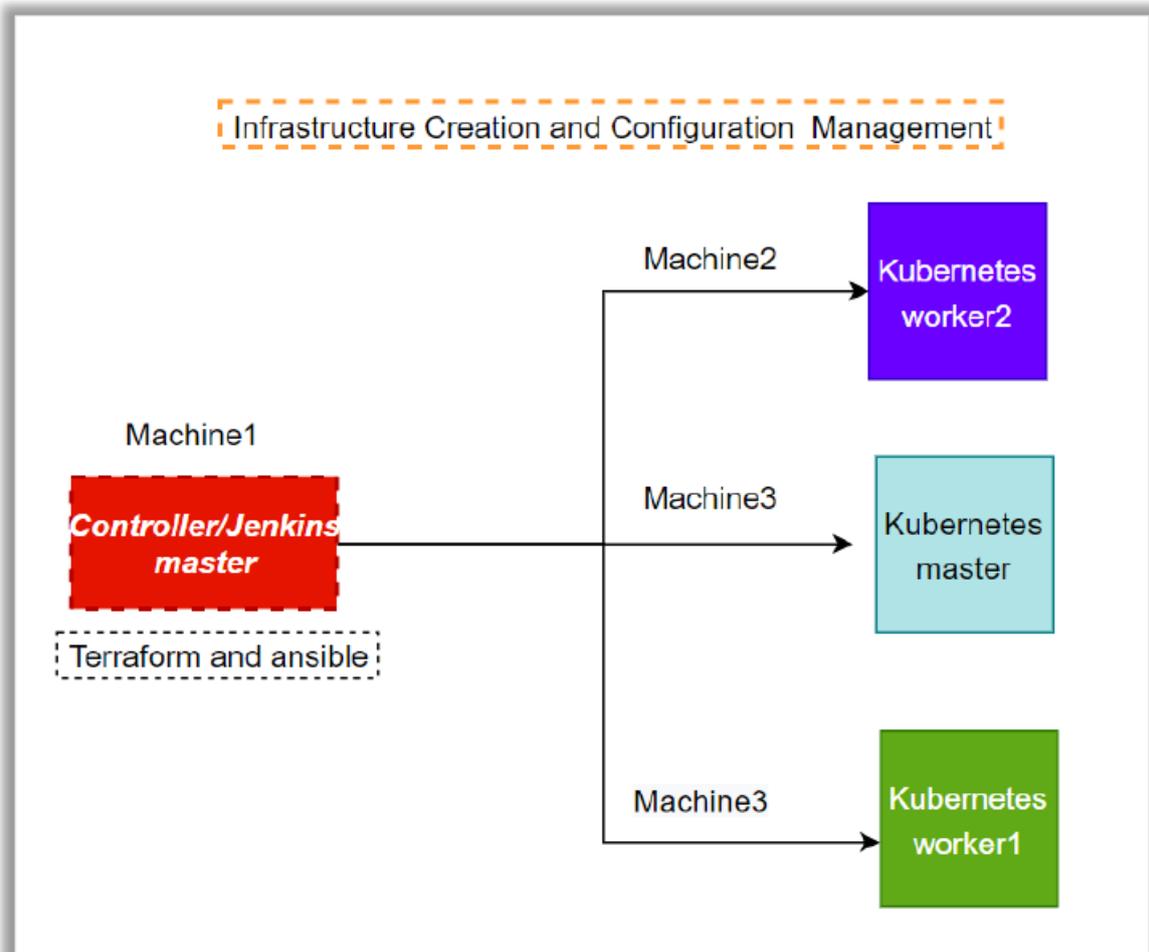
Software's to be installed on the respective machines using configuration management.

Worker1: Jenkins, Java.

Worker2: Docker, Kubernetes.

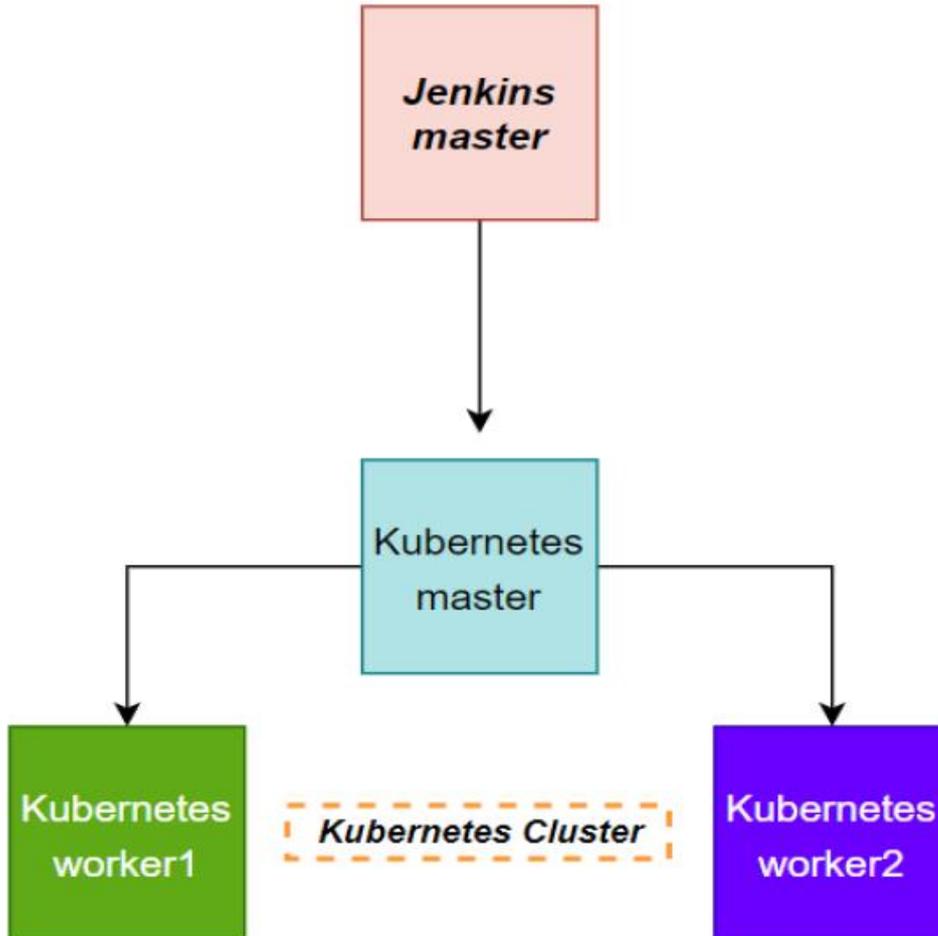
Worker3: Java, Docker, Kubernetes

Worker4: Docker, Kubernetes.



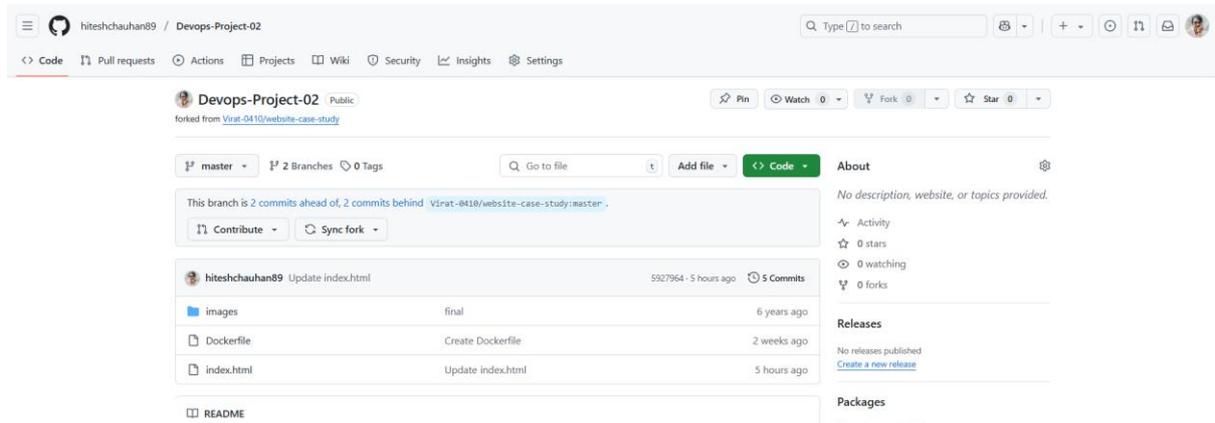
ARCHITECTURE OF JENKINS AND KUBERNETES

Servers for jenkins and kubernetes configuration



First Go to This Url And Rename The Repository.

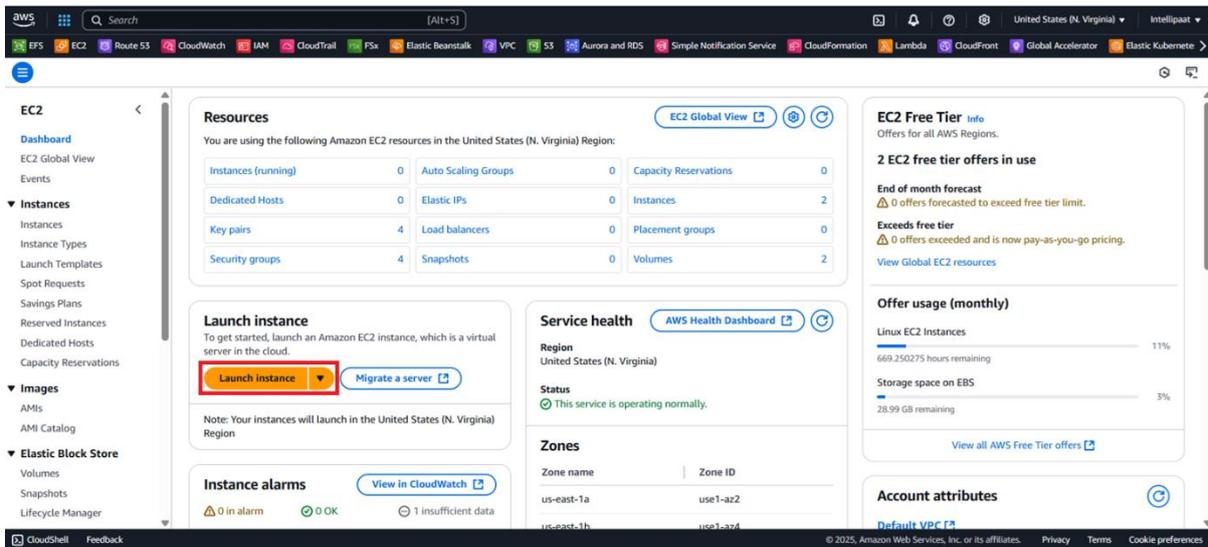
<https://github.com/hshar/website.git>



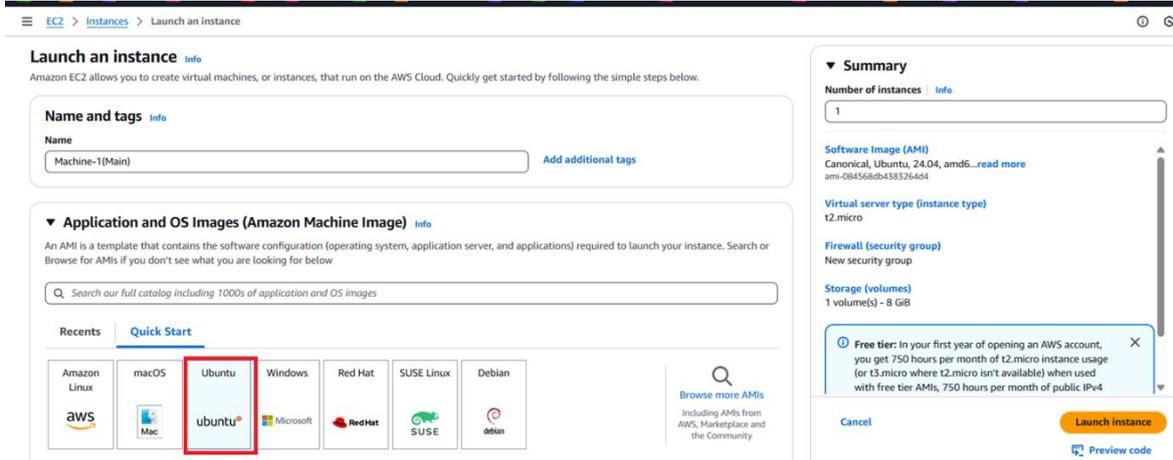
After Rename this repo.

<https://github.com/hiteshchauhan89/Devops-Project-02.git>

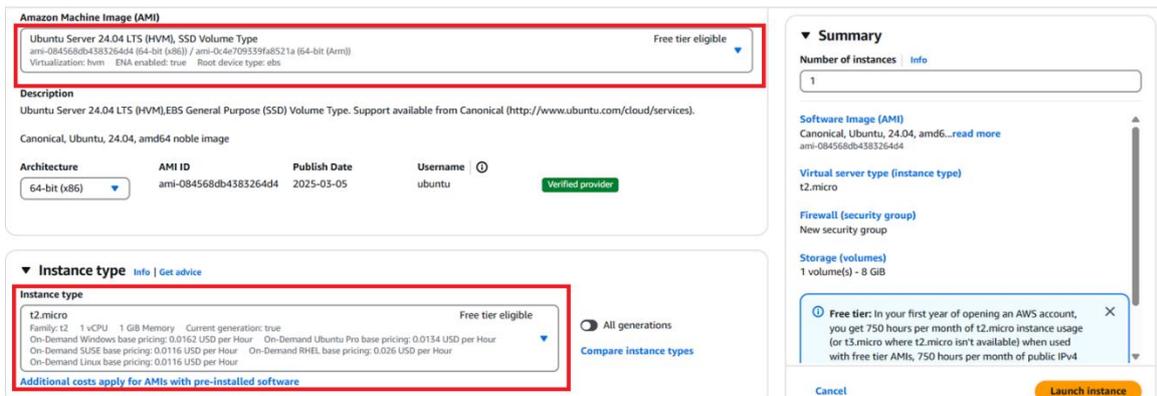
Step 1. Go to Aws Management Console and Click the Launch Instance.



Step 2: Choose “AMI” as “ubuntu”.



Step 3: Select Amazon Machine Images(AMI) and select Instance Types. Instance Type will be t2.micro



Step 4: Choose the “key pair (docker) — required”

Step 5: In the “Network Settings”, choose the “Select existing security group” in the “Firewall (security groups)” & While choosing the “Common security groups” as “default”.

EC2 > Instances > Launch an instance

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*

docker Create new key pair

Network settings Info Edit

Network Info vpc-056bf1d89dcf9098e

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

Common security groups Info

Select security groups

launch-wizard-1 sg-047bfe49f4b83fc4a Compare security group rules

Summary

Number of instances Info 1

Software Image (AMI) Canonical, Ubuntu, 24.04, amd64...read more ami-084568db4383264d4

Virtual server type (instance type) t2.micro

Firewall (security group) launch-wizard-1

Storage (volumes) 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4

Cancel Launch instance Preview code

Step 6: Choose the Select Storage.

Configure storage Info Advanced

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance.

Click refresh to view backup information

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

0 x File systems Edit

Advanced details Info

Software Image (AMI) Canonical, Ubuntu, 24.04, amd64...read more ami-084568db4383264d4

Virtual server type (instance type) t2.micro

Firewall (security group) launch-wizard-1

Storage (volumes) 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4

Cancel Launch instance Preview code

Now Launch The ec2 Instance.

Note: This EC2 Instance will use for terraform.

EC2 > Instances > Launch an instance

Launching Instance

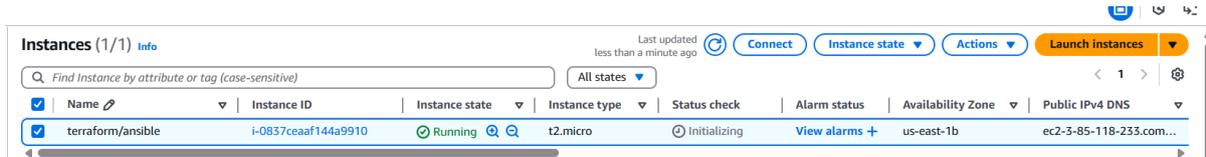
Launch initiation 77%

Details

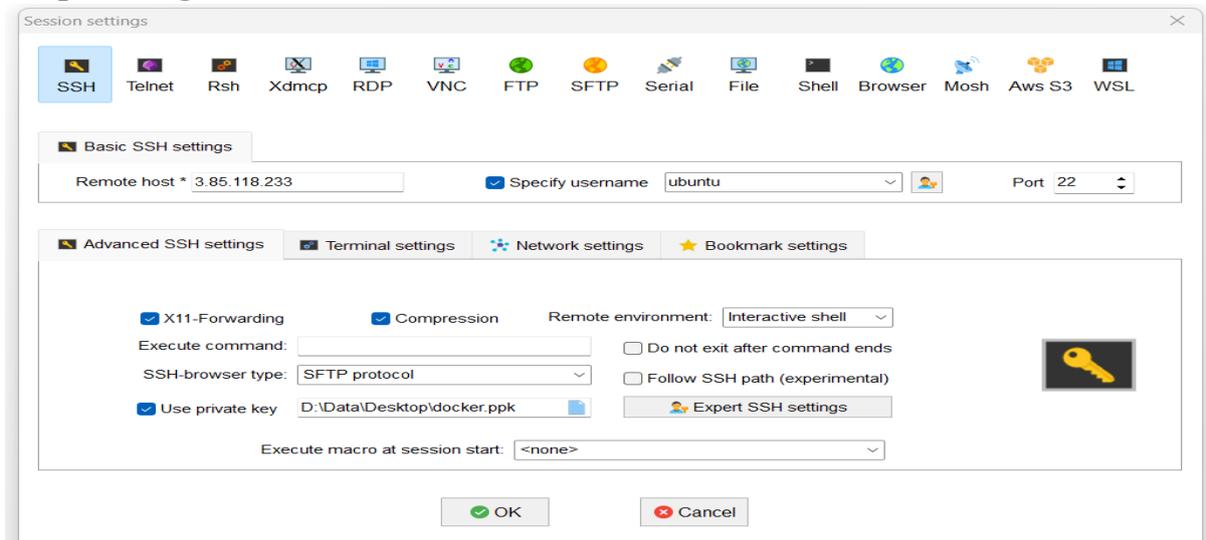
Step 7: The “Instance” will be successfully launched, click on “hyperlink (i-0837ceaaf144a9910)”.

This is EC2 instance Full Details.

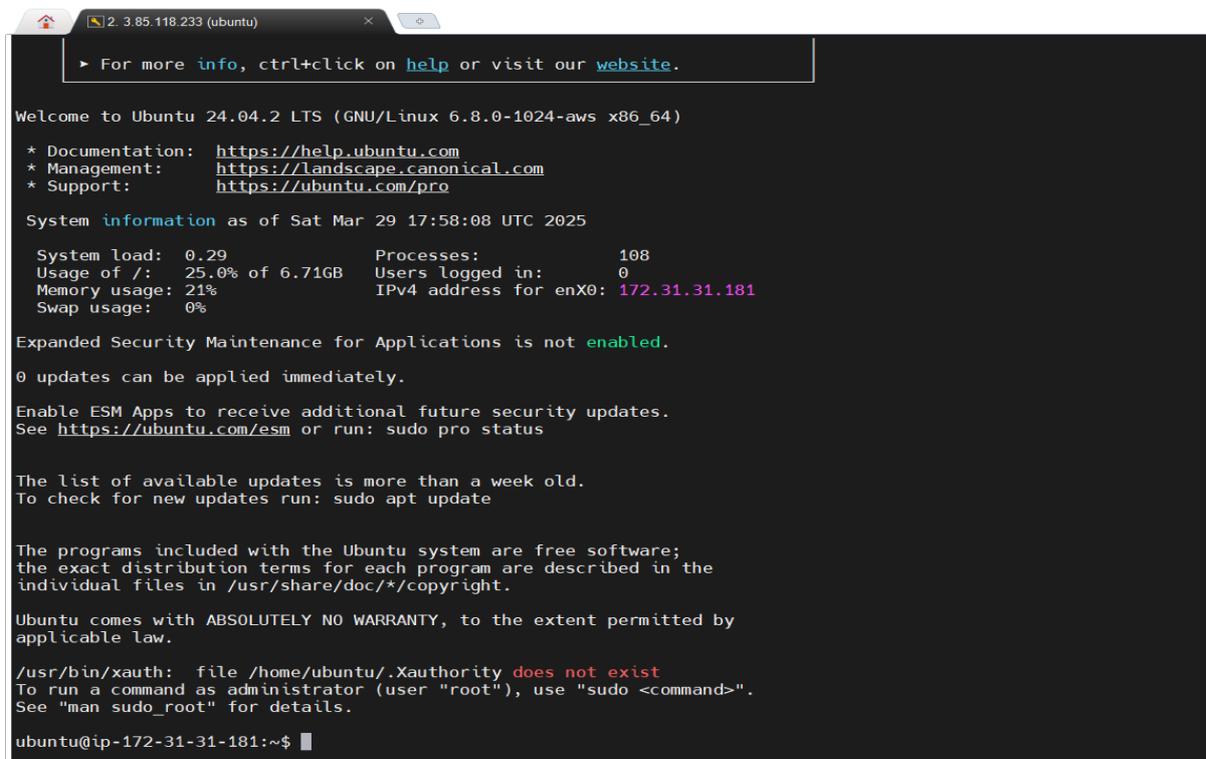
Step 9: The Instance “[Terraform/Ansible]” will be in the “Running” State.



Step 10: Again Connect to ec2 instance.



After Connected the Instance we need to update the machine.



Go to official Terraform documentation for install the terraform.

<https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli>

SCRIPTS FOR INSTALLATION OF TERRAFORM

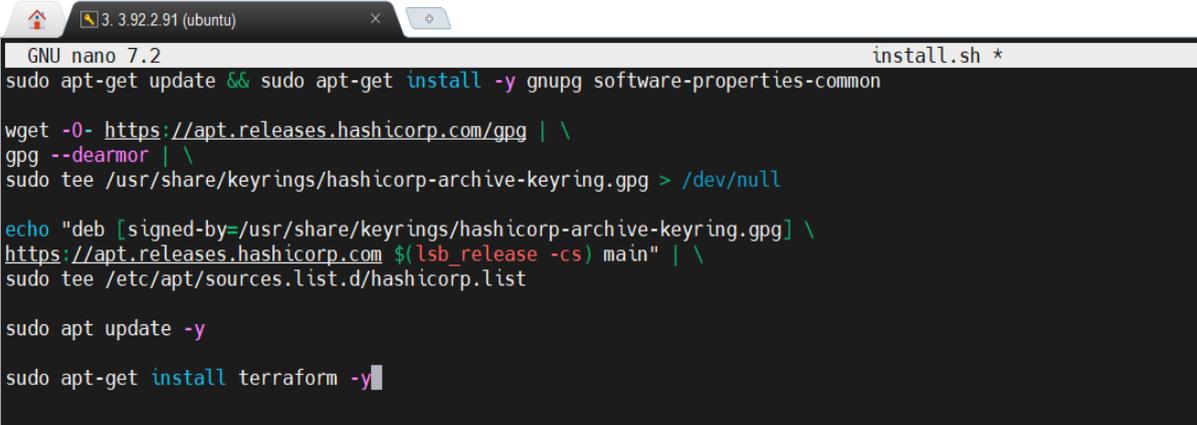
```
sudo apt-get update && sudo apt-get install -y gnupg software-properties-common  
wget -O- https://apt.releases.hashicorp.com/gpg | \  
gpg --dearmor | \  
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
```

```
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \  
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \  
sudo tee /etc/apt/sources.list.d/hashicorp.list
```

```
sudo apt update -y
```

```
sudo apt-get install terraform -y
```

Created The Script File for install the terraform.

A screenshot of a terminal window titled 'install.sh *' running GNU nano 7.2. The terminal shows the following commands:

```
sudo apt-get update && sudo apt-get install -y gnupg software-properties-common  
wget -O- https://apt.releases.hashicorp.com/gpg | \  
gpg --dearmor | \  
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null  
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \  
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \  
sudo tee /etc/apt/sources.list.d/hashicorp.list  
sudo apt update -y  
sudo apt-get install terraform -y
```

After Save this file to .sh format

```
root@ip-172-31-90-23:/home/ubuntu# sudo nano install.sh
root@ip-172-31-90-23:/home/ubuntu# sudo bash install.sh
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupp is already the newest version (2.4.4-2ubuntu17).
gnupp set to manually installed.
software-properties-common is already the newest version (0.99.49.1).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 29 not upgraded.
--2025-03-26 14:19:08-- https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 18.160.10.71, 18.160.10.45, 18.160.10.126, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|18.160.10.71|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3980 (3.9K) [binary/octet-stream]
Saving to: 'STDOUT'

-
100%[=====] 3.89K --.-KB/s in 0s

2025-03-26 14:19:08 (1004 MB/s) - written to stdout [3980/3980]

deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://apt.releases.hashicorp.com noble InRelease [12.9 kB]
Get:6 https://apt.releases.hashicorp.com noble/main amd64 Packages [174 kB]
Fetched 187 kB in 1s (284 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
29 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
2. 3.85.118.233 (ubuntu) x
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://apt.releases.hashicorp.com noble InRelease [12.9 kB]
Get:6 https://apt.releases.hashicorp.com noble/main amd64 Packages [174 kB]
Fetched 187 kB in 1s (273 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
34 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
 terraform
0 upgraded, 1 newly installed, 0 to remove and 34 not upgraded.
Need to get 27.6 MB of archives.
After this operation, 90.6 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com noble/main amd64 terraform amd64 1.11.3-1 [27.6 MB]
Fetched 27.6 MB in 0s (91.9 MB/s)
Selecting previously unselected package terraform.
(Reading database ... 70560 files and directories currently installed.)
Preparing to unpack ../terraform_1.11.3-1_amd64.deb ...
Unpacking terraform (1.11.3-1) ...
Setting up terraform (1.11.3-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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.
ubuntu@ip-172-31-31-181:~$
```

Need To check the terraform Version.

Run this command

terraform --version

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-31-181:~$ terraform --version
Terraform v1.11.3
on linux_amd64
ubuntu@ip-172-31-31-181:~$ █
```

Now run the terraform and check the output.

```
root@ip-172-31-90-23:/home/ubuntu# terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init           Prepare your working directory for other commands
  validate       Check whether the configuration is valid
  plan           Show changes required by the current configuration
  apply          Create or update infrastructure
  destroy        Destroy previously-created infrastructure

All other commands:
  console        Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  metadata      Metadata related commands
  modules       Show all declared modules in a working directory
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  state         Advanced state management
  taint         Mark a resource instance as not fully functional
  test          Execute integration tests for Terraform modules
  untaint       Remove the 'tainted' state from a resource instance
  version       Show the current Terraform version
  workspace     Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR    Switch to a different working directory before executing the
                given subcommand.
```

Create One Directory Name is **Terraform-projects-2** in under this folder create one main.tf file.

```
ubuntu@ip-172-31-95-142:~/terraform-projects2$ cat main.tf
provider "aws" {
  region = "us-east-1"
}

variable "ami_id" {
  default = "ami-084568db4383264d4"
}

variable "instance_type_master" {
  default = "t2.medium"
}

variable "instance_type_worker" {
  default = "t2.micro"
}

variable "subnet_id" {
  default = "subnet-083f3dea30f73b6b7"
}

variable "key_name" {
  default = "DockeR"
}

resource "aws_security_group" "k8s_sg" {
  name        = "kubernetes-sg"
  description = "Allow necessary Kubernetes ports"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 6443
    to_port     = 6443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

```

    ingress {
      from_port = 10250
      to_port   = 10250
      protocol  = "tcp"
      cidr_blocks = ["0.0.0.0/0"]
    }

    egress {
      from_port = 0
      to_port   = 0
      protocol  = "-1"
      cidr_blocks = ["0.0.0.0/0"]
    }
  }

  resource "aws_instance" "k8s_master" {
    ami             = var.ami_id
    instance_type   = var.instance_type_master
    subnet_id       = var.subnet_id
    key_name         = var.key_name
    security_groups = [aws_security_group.k8s_sg.name]
    associate_public_ip_address = true

    tags = {
      Name = "Kubernetes-Master"
    }
  }

  resource "aws_instance" "k8s_worker1" {
    ami             = var.ami_id
    instance_type   = var.instance_type_worker
    subnet_id       = var.subnet_id
    key_name         = var.key_name
    security_groups = [aws_security_group.k8s_sg.name]

    tags = {
      Name = "Kubernetes-Worker-1"
    }
  }
}

```

```

provider "aws" {
  region = "us-east-1"
  access_key = "AKIA4MI2JKFZJD4RZIJB"
  secret_key =
"PGOb1b6Dz5aZMwM8hjp28pKaITQH4tF
6MkE17TP7"
}

```

```

variable "ami_id" {
  default = "ami-084568db4383264d4"
}

```

```

variable "instance_type_master" {
  default = "t2.medium"
}

```

```

variable "instance_type_worker" {

```

```
    default = "t2.micro"
}

variable "subnet_id" {
    default = "subnet-083f3dea30f73b6b7"
}

variable "key_name" {
    default = "docker"
}

resource "aws_security_group" "k8s_sg" {
    name      = "kubernetes-sg"
    description = "Allow necessary
Kubernetes ports"

    ingress {
        from_port = 22
        to_port   = 22
        protocol  = "tcp"
        cidr_blocks = ["0.0.0.0/0"]
    }
    ingress {
        from_port = 6443
        to_port   = 6443
        protocol  = "tcp"
        cidr_blocks = ["0.0.0.0/0"]
    }

    ingress {
        from_port = 10250
        to_port   = 10250
        protocol  = "tcp"
        cidr_blocks = ["0.0.0.0/0"]
    }

    egress {
        from_port = 0
        to_port   = 0
```

```
    protocol = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_instance" "k8s_master" {
  ami          = var.ami_id
  instance_type =
var.instance_type_master
  subnet_id    = var.subnet_id
  key_name     = var.key_name
  vpc_security_group_ids =
[aws_security_group.k8s_sg.id]
  associate_public_ip_address = true

  tags = {
    Name = "Kubernetes-Master"
  }
}

resource "aws_instance" "k8s_worker1" {
  ami          = var.ami_id
  instance_type =
var.instance_type_worker
  subnet_id    = var.subnet_id
  key_name     = var.key_name
  vpc_security_group_ids =
[aws_security_group.k8s_sg.id]

  tags = {
    Name = "Kubernetes-Worker-1"
  }
}

resource "aws_instance" "k8s_worker2" {
  ami          = var.ami_id
  instance_type =
var.instance_type_worker
  subnet_id    = var.subnet_id
```

```

key_name      = var.key_name
vpc_security_group_ids =
[aws_security_group.k8s_sg.id]

tags = {
  Name = "Kubernetes-Worker-2"
}
}

resource "aws_instance" "k8s_worker3" {
  ami          = var.ami_id
  instance_type =
var.instance_type_worker
  subnet_id    = var.subnet_id
  key_name     = var.key_name
  vpc_security_group_ids =
[aws_security_group.k8s_sg.id]

tags = {
  Name = "Kubernetes-Worker-3"
}
}

```

```

ubuntu@ip-172-31-31-181:~/Terraform-projects-2$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.93.0...
- Installed hashicorp/aws v5.93.0 (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.
ubuntu@ip-172-31-31-181:~/Terraform-projects-2$ █

```

Go To Below links.

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

```
unzip awscliv2.zip
```

```
sudo ./aws/install
```

```
root@ip-172-31-95-142:/home/ubuntu/terraform-projects2# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total   Spent    Left     Speed
100 65.0M  100 65.0M    0     0  106M      0  --:--:-- --:--:-- --:--:--  106M
Command 'unzip' not found, but can be installed with:
apt install unzip
sudo: ./aws/install: command not found
root@ip-172-31-95-142:/home/ubuntu/terraform-projects2# apt install unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 23 not upgraded.
Need to get 174 kB of archives.
After this operation, 384 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 unzip amd64 6.0-28ubuntu4.1 [174 kB]
Fetched 174 kB in 0s (6615 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 70563 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-28ubuntu4.1_amd64.deb ...
Unpacking unzip (6.0-28ubuntu4.1) ...
Setting up unzip (6.0-28ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
```

```
root@ip-172-31-95-142:/home/ubuntu/terraform-projects2# apt install unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 23 not upgraded.
Need to get 174 kB of archives.
After this operation, 384 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 unzip amd64 6.0-28ubuntu4.1 [174 kB]
Fetched 174 kB in 0s (6615 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 70563 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-28ubuntu4.1_amd64.deb ...
Unpacking unzip (6.0-28ubuntu4.1) ...
Setting up unzip (6.0-28ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-90-23:/home/ubuntu# unzip awscliv2.zip
Archive:  awscliv2.zip
  creating:  aws/
  creating:  aws/dist/
 inflating:  aws/install
 inflating:  aws/README.md
 inflating:  aws/THIRD_PARTY_LICENSES
  creating:  aws/dist/awscli/
  creating:  aws/dist/cryptography/
  creating:  aws/dist/docutils/
  creating:  aws/dist/lib-dynload/
 inflating:  aws/dist/aws
 inflating:  aws/dist/aws_completer
 inflating:  aws/dist/libpython3.12.so.1.0
 inflating:  aws/dist/_awsCRT.abi3.so
 inflating:  aws/dist/_cffi_backend.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/_ruamel_yaml.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/libz.so.1
 inflating:  aws/dist/liblzma.so.5
 inflating:  aws/dist/libbz2.so.1
 inflating:  aws/dist/libffi.so.6
 inflating:  aws/dist/libuuid.so.1
 inflating:  aws/dist/libgcc_s.so.1
 inflating:  aws/dist/libsqlite3.so.0
 inflating:  aws/dist/base_library.zip
 inflating:  aws/dist/lib-dynload/_struct.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/zlib.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/grp.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/math.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/select.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/_posixsubprocess.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/fcntl.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/_datetime.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/unicodedata.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/_statistics.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/_contextvars.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/_decimal.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/_pickle.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/_hashlib.cpython-312-x86_64-linux-gnu.so
 inflating:  aws/dist/lib-dynload/_sha3.cpython-312-x86_64-linux-gnu.so
```

```
root@ip-172-31-90-23:/home/ubuntu# sudo ./aws/install
You can now run: /usr/local/bin/aws --version
root@ip-172-31-90-23:/home/ubuntu# █
```

```
ubuntu@ip-172-31-82-101:~/terraform-projects2$ sudo terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.92.0...
- Installed hashicorp/aws v5.92.0 (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.

```
ubuntu@ip-172-31-31-181:~/Terraform-projects-2$ terraform apply --auto-approve
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```
# aws_instance.k8s_master will be created
+ resource "aws_instance" "k8s_master" {
  + ami                    = "ami-084568db4383264d4"
  + arn                    = (known after apply)
  + associate_public_ip_address = true
  + availability_zone      = (known after apply)
  + cpu_core_count         = (known after apply)
  + cpu_threads_per_core   = (known after apply)
  + disable_api_stop       = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized          = (known after apply)
  + enable_primary_ipv6    = (known after apply)
  + get_password_data      = false
  + host_id                = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile   = (known after apply)
  + id                     = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle     = (known after apply)
  + instance_state         = (known after apply)
  + instance_type          = "t2.medium"
  + ipv6_address_count     = (known after apply)
  + ipv6_addresses        = (known after apply)
  + key_name               = "docker"
  + monitoring             = (known after apply)
  + outpost_arn            = (known after apply)
  + password_data          = (known after apply)
  + placement_group        = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns            = (known after apply)
  + private_ip             = (known after apply)
  + public_dns             = (known after apply)
  + public_ip              = (known after apply)
```

```

+ public_dns = (known after apply)
+ public_ip = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups = (known after apply)
+ source_dest_check = true
+ spot_instance_request_id = (known after apply)
+ subnet_id = "subnet-083f3dea30f73b6b7"
+ tags = {
+   "Name" = "Kubernetes-Master"
}
+ tags_all = {
+   "Name" = "Kubernetes-Master"
}
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)

+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}

```

```

# aws_instance.k8s_worker[0] will be created
+ resource "aws_instance" "k8s_worker" {
+   ami = "ami-084568db4383264d4"
+   arn = (known after apply)
+   associate_public_ip_address = true
+   availability_zone = (known after apply)
+   cpu_core_count = (known after apply)
+   cpu_threads_per_core = (known after apply)
+   disable_api_stop = (known after apply)
+   disable_api_termination = (known after apply)
+   ebs_optimized = (known after apply)
+   enable_primary_ipv6 = (known after apply)
+   get_password_data = false
+   host_id = (known after apply)
+   host_resource_group_arn = (known after apply)
+   iam_instance_profile = (known after apply)
+   id = (known after apply)
+   instance_initiated_shutdown_behavior = (known after apply)
+   instance_lifecycle = (known after apply)
+   instance_state = (known after apply)
+   instance_type = "t2.micro"
+   ipv6_address_count = (known after apply)
+   ipv6_addresses = (known after apply)
+   key_name = "docker"
+   monitoring = (known after apply)
+   outpost_arn = (known after apply)
+   password_data = (known after apply)
+   placement_group = (known after apply)
+   placement_partition_number = (known after apply)
+   primary_network_interface_id = (known after apply)
+   private_dns = (known after apply)
+   private_ip = (known after apply)
+   public_dns = (known after apply)
+   public_ip = (known after apply)
+   secondary_private_ips = (known after apply)
+   security_groups = (known after apply)
+   source_dest_check = true
+   spot_instance_request_id = (known after apply)
+   subnet_id = "subnet-083f3dea30f73b6b7"
+   tags = {
+     "Name" = "Kubernetes-Worker-1"

```

```

+ tags = {
+   + "Name" = "Kubernetes-Worker-1"
}
+ tags_all = {
+   + "Name" = "Kubernetes-Worker-1"
}
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)

+ capacity_reservation_specification (known after apply)

+ cpu_options (known after apply)

+ ebs_block_device (known after apply)

+ enclave_options (known after apply)

+ ephemeral_block_device (known after apply)

+ instance_market_options (known after apply)

+ maintenance_options (known after apply)

+ metadata_options (known after apply)

+ network_interface (known after apply)

+ private_dns_name_options (known after apply)

+ root_block_device (known after apply)
}

# aws_instance.k8s_worker[1] will be created
+ resource "aws_instance" "k8s_worker" {
+   ami = "ami-084568db4383264d4"
+   arn = (known after apply)
+   associate_public_ip_address = true
+   availability_zone = (known after apply)

```

```

+ "0.0.0.0/0",
]
+ from_port = 6443
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 6443
# (1 unchanged attribute hidden)
}],
+ name = "kubernetes-securitygroup1"
+ name_prefix = (known after apply)
+ owner_id = (known after apply)
+ revoke_rules_on_delete = false
+ tags_all = (known after apply)
+ vpc_id = (known after apply)
}

Plan: 6 to add, 0 to change, 0 to destroy.
aws_security_group.k8s_sg: Creating...
aws_security_group.k8s_sg: Creation complete after 2s [id=sg-000027647fe73b0e7]
aws_instance.k8s_master: Creating...
aws_instance.k8s_worker[0]: Creating...
aws_instance.k8s_worker[1]: Creating...
aws_instance.k8s_worker[3]: Creating...
aws_instance.k8s_worker[2]: Creating...
aws_instance.k8s_master: Still creating... [10s elapsed]
aws_instance.k8s_worker[0]: Still creating... [10s elapsed]
aws_instance.k8s_worker[1]: Still creating... [10s elapsed]
aws_instance.k8s_worker[3]: Still creating... [10s elapsed]
aws_instance.k8s_worker[2]: Still creating... [10s elapsed]
aws_instance.k8s_worker[3]: Creation complete after 13s [id=i-0d96fda4120baf7ee]
aws_instance.k8s_worker[0]: Creation complete after 13s [id=i-0cae158b8fd8c2e9c]
aws_instance.k8s_worker[2]: Creation complete after 13s [id=i-0b88c3f43dd09d443]
aws_instance.k8s_worker[1]: Creation complete after 13s [id=i-0da89b340394a7f0a]
aws_instance.k8s_master: Creation complete after 13s [id=i-0f3ab9d56942c9b69]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-31-181:~/Terraform-projects-2$ █

```

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Kubernetes-Worker-3	i-0b88c3f43dd09d443	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-107-21-29-175.co...
Kubernetes-Worker-2	i-0da89b340394a7f0a	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-3-93-145-163.com...
Kubernetes-Worker-1	i-0cae158b8fd8c2e9c	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-44-201-178-86.co...
Kubernetes-Master	i-0f3ab9d56942c9b69	Running	t2.medium	Initializing	View alarms +	us-east-1a	ec2-44-203-3-140.com...
Kubernetes-Worker-4	i-0d96fda4120baf7ee	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-3-84-16-226.comp...
terraform/ansible	i-0837ceaf144a9910	Running	t2.micro	2/2 checks pass	View alarms +	us-east-1b	ec2-3-85-118-233.com...

Now Go to Terraform/Ansible Server And this need to install the ansible.

Go to cd /home/ubuntu/ this path and create the install.sh script file.

sudo nano ansible.sh

```

GNU nano 7.2 ansible.sh *
sudo apt update -y
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible -y

```

Now we need to run this file.

Sudo bash ansible.sh

```

ubuntu@ip-172-31-31-181:~$ sudo bash ansible.sh
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://apt.releases.hashicorp.com noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
34 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.49.1).
0 upgraded, 0 newly installed, 0 to remove and 34 not upgraded.
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/
Suites: noble
Components: main
'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and
update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://apt.releases.hashicorp.com noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu noble InRelease [17.8 kB]
Get:7 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu noble/main amd64 Packages [776 B]
Get:8 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu noble/main Translation-en [472 B]
Fetched 19.1 kB in 1s (20.7 kB/s)
Reading package lists... Done
Reading package lists... Done

```

```
2. 3.85.118.233 (ubuntu) x
Unpacking python3-ntlm-auth (1.5.0-1) ...
Selecting previously unselected package python3-paramiko.
Preparing to unpack ../06-python3-paramiko_2.12.0-2ubuntu4.1_all.deb ...
Unpacking python3-paramiko (2.12.0-2ubuntu4.1) ...
Selecting previously unselected package python3-requests-ntlm.
Preparing to unpack ../07-python3-requests-ntlm_1.1.0-3_all.deb ...
Unpacking python3-requests-ntlm (1.1.0-3) ...
Selecting previously unselected package python3-xmltodict.
Preparing to unpack ../08-python3-xmltodict_0.13.0-1_all.deb ...
Unpacking python3-xmltodict (0.13.0-1) ...
Selecting previously unselected package python3-winrm.
Preparing to unpack ../09-python3-winrm_0.4.3-2_all.deb ...
Unpacking python3-winrm (0.4.3-2) ...
Selecting previously unselected package sshpass.
Preparing to unpack ../10-sshpass_1.09-1_amd64.deb ...
Unpacking sshpass (1.09-1) ...
Setting up python3-ntlm-auth (1.5.0-1) ...
Setting up python3-resolvlib (1.0.1-1) ...
Setting up python3-kerberos (1.1.14-3.1build9) ...
Setting up ansible-core (2.17.10-1ppa~noble) ...
Setting up sshpass (1.09-1) ...
Setting up python3-xmltodict (0.13.0-1) ...
Setting up ansible (10.7.0-1ppa~noble) ...
Setting up python3-nacl (1.5.0-4build1) ...
Setting up python3-requests-ntlm (1.1.0-3) ...
Setting up python3-winrm (0.4.3-2) ...
Setting up python3-paramiko (2.12.0-2ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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.
ubuntu@ip-172-31-31-181:~$
```

Now Ansible Setup has been successfully done.

Need to check the version.

```
ubuntu@ip-172-31-31-181:~$ ansible --version
ansible [core 2.17.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Feb  4 2025, 14:48:35) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-172-31-31-181:~$
```

Now we need to create the connection between 6 servers means we need to create the passwordless ssh.

So go to terraform/ansible server and run this command and create the ssh keygen.

ssh-keygen -t rsa

```

ubuntu@ip-172-31-31-181:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:IZHjJEPclgnB9Ffagb2VjDMY010zV+26UWpknAmeDvw ubuntu@ip-172-31-31-181
The key's randomart image is:
+---[RSA 3072]---+
|  ==+.+ +o+o. o|
|  =.0.=+B.=o ..|
|  *o.+=.B.+.+|
|  .o.+ o.* o|
|    S + o +|
|      E =|
|      . o|
|      .|
+-----[SHA256]-----+
ubuntu@ip-172-31-31-181:~$ █

```

Now Go to this path `cd /home/ubuntu/.ssh`

The keys has been created in .ssh folder. There is two files has been created.

1.id_rsa

2.id_rsa.pub

```

ubuntu@ip-172-31-31-181:~$ cd .ssh/
ubuntu@ip-172-31-31-181:~/.ssh$ ll
total 20
drwx----- 2 ubuntu ubuntu 4096 Mar 29 18:15 ./
drwxr-x--- 7 ubuntu ubuntu 4096 Mar 29 18:14 ../
-rw----- 1 ubuntu ubuntu 388 Mar 29 17:56 authorized_keys
-rw----- 1 ubuntu ubuntu 2610 Mar 29 18:15 id_rsa
-rw-r--r-- 1 ubuntu ubuntu 577 Mar 29 18:15 id_rsa.pub
ubuntu@ip-172-31-31-181:~/.ssh$ █

```

Run this command `cat id_rsa.pub`.you will show ssh key.this key need to be copy and more 5 instances need to paste in same location `cd /home/ubuntu/.ssh/authorized_keys`

```

ubuntu@ip-172-31-31-181:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDgELePvgCubDRrB97EaYvu5NAF0atT2L_i8KilCKg/60V1ruc6x0+T0X6n5Latn68ZF9av6TY0x0drT0maAIYTr1RYPH+dDL9fapTl526AmX6fYoTLUrs4M6g
gedMifxdzxl7Egq/krkFYH2Ulnf5zL0XDCu3cssrVDoynYIP0ddJVDvt47gGd7pF+om9zuxF7N2P94dn3rzGBecjvLuxtfyWUmlUieDn8BREWILy3dnvy+2w4RN0yC1LGoHob4XkXA6Wirc0hMYw1F+9nB+ksFc5
MTJxYDcZ0PkSnovFM7Ap1a8Sb7vS0g37H6ngz4yacqcG5Mt1AbMsRYZht+MM7Kb6AS0uNautgxLAYRD iqlWjJE0sjSG/EI1qt3xUhALFYey+QxnxgnEZw1N020kpBrqKA LxXIXseptLVmCYERZmXtAXB+lz i5no+2
YvC0mhSkqrnHVTcsu0XBB9s0Ujjs5ZuxmXq5VGguS9jfhjybj7mrGwKudLvyx2t8M= ubuntu@ip-172-31-31-181
ubuntu@ip-172-31-31-181:~/.ssh$ █

```

So first of all need to login the server one by one and paste this key into path `cd /home/ubuntu/.ssh`

First We take Kubernetes-Worker1 server.

Session settings ✕

SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL

Basic SSH settings

Remote host * Specify username Port

Advanced SSH settings Terminal settings Network settings Bookmark settings

X11-Forwarding Compression Remote environment:

Execute command: Do not exit after command ends

SSH-browser type: Follow SSH path (experimental)

Use private key 

Execute macro at session start:

```
2. 3.85.118.233 (ubuntu) 3. 44.203.3.140 (ubuntu)
> For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

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

System information as of Sat Mar 29 18:22:51 UTC 2025

System load: 0.0          Processes: 115
Usage of /: 25.0% of 6.71GB Users logged in: 0
Memory usage: 5%         IPv4 address for enX0: 172.31.88.162
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.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-88-162:~$
```

```
ubuntu@ip-172-31-88-162:~/ .ssh$ ll
total 12
drwx----- 2 ubuntu ubuntu 4096 Mar 29 18:23 ./
drwxr-x--- 4 ubuntu ubuntu 4096 Mar 29 18:26 ../
-rw----- 1 ubuntu ubuntu 965 Mar 29 18:23 authorized_keys
ubuntu@ip-172-31-88-162:~/ .ssh$
```

```
GNU nano 7.2 authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDwJDfZ6pZ2BLjV6ZoyJyc3mdbosVtTXUpug0mXGhKMyHLF5lKxik8PMTIxNwc1TXGAMM7027MVB4QaGZ9VmzL21HdwmrQA/rTaUqRwVnJfVgQ+3WK/ZCY6svhS
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQg0DYE0LcPvgCubDRrB97EaYvu5NAF0atT2Ll8Kl1GKg/6QV1ruc6x0+T0X6n5LaLn68Zf9av6TY0xodrTQmaAIYTcr1RYPHo+dDL9fapIL52GAmX6f0YoIUrs4MG
```

Save this key and exit this window.
Now go to ansible server and try to ssh of this server.

```
ubuntu@ip-172-31-31-181:~/.ssh$ pwd
/home/ubuntu/.ssh
ubuntu@ip-172-31-31-181:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDQYEOlcPvgCubDRrB97EaYvu5NAFOatT2Li8KilGKg/6QV1ruc6x0+T0X6n5LaLn68Zf9av6TY0x0drtQmaAIYTr1RYPHo+dL9fapIL52GAmX6f0YoILUrs4MGz
gedXHFxdzxl7Egd/krkFYH2UWnf5zLQXXDCu3cssrVDoyWYIP0ddJVdVt47gGd7pF+om9zxF7N2P94dn3rzGBecjvLuxtfYwDUmUieZDn8bREWILy3dnwy+2w4RN0yCN1GoHeb4XkXA6Vwrc0hMYW1F+9nB+ksFc9
MTJxYDcZ0PkSnovFM7ApIa0Sb7vS0g37H6ngz4yacqcG5MElAbMsRYZht+MM7Kb6AS0uNautgxLAYRDlqLWjJE0sjsG/EI1qt3xihALFYey+0xnXgnEzWlN020kpBrqKAixXIXseptLVmCYERZmXtAXB+lz15no+2t
YvC0mHskqrnHVTcsu0XBB0s0Ujs5ZuxmXq5Vggu59jfhjybj7mrGwKudLvyxX2t8M= ubuntu@ip-172-31-31-181
ubuntu@ip-172-31-31-181:~/.ssh$ ^C
ubuntu@ip-172-31-31-181:~/.ssh$ ssh 172.31.88.162
The authenticity of host '172.31.88.162 (172.31.88.162)' can't be established.
ED25519 key fingerprint is SHA256:ZyWd+R3Gd09zxH9IGUd4q0Ygh5/dRB/IL3NRuIi6DSI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.88.162' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

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

System information as of Sat Mar 29 18:24:54 UTC 2025

System load:  0.08      Processes:      121
Usage of /:   25.2% of 6.71GB  Users logged in:  1
Memory usage: 6%        IPv4 address for enX0: 172.31.88.162
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

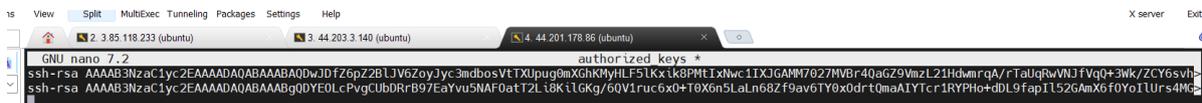
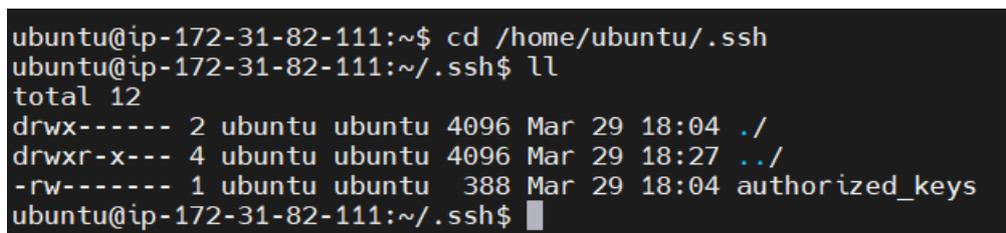
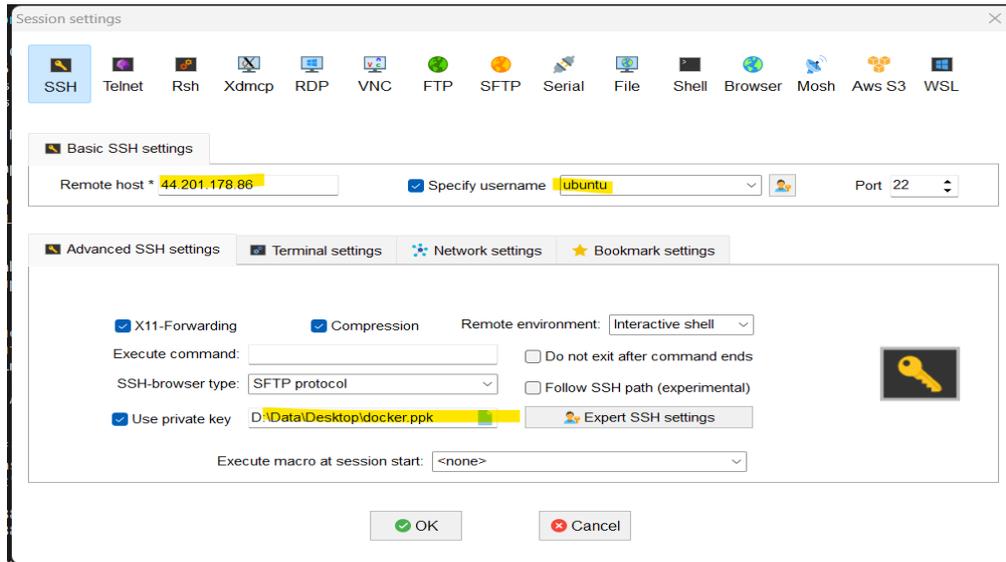
Last login: Sat Mar 29 18:22:52 2025 from 106.222.204.95
ubuntu@ip-172-31-88-162:~$
```

This Kubernetes connecting master server has been successfully connected to passwordless connection.

Now Take Kubernetes Worker1 Server we used same process.

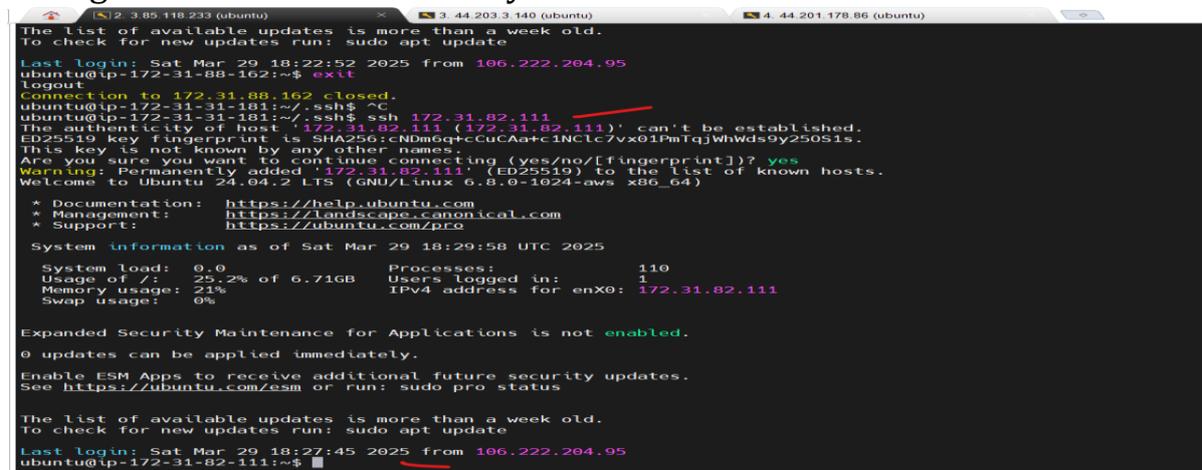
Run this command `cat id_rsa.pub`. you will show ssh key. this key need to be copy and more 5 instances need to paste in same location `cd /home/ubuntu/.ssh/authorized_keys`

Second Kubernetes Worker2



Save this key and exit this window.

Now go to ansible server and try to ssh of this server.

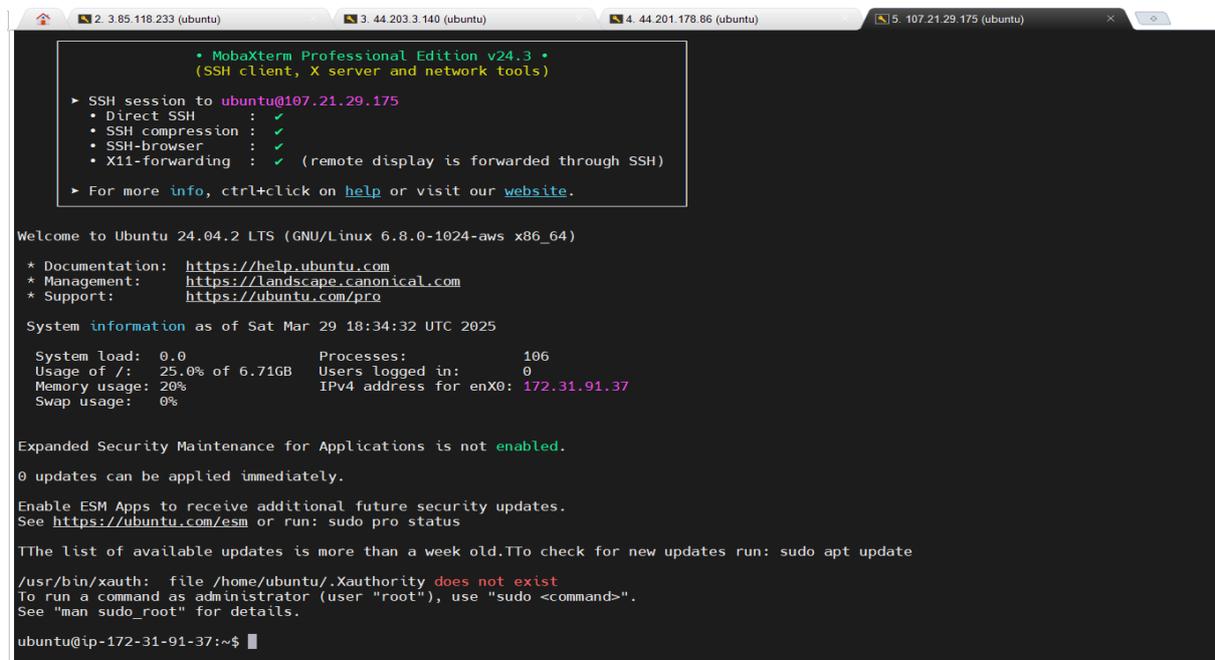
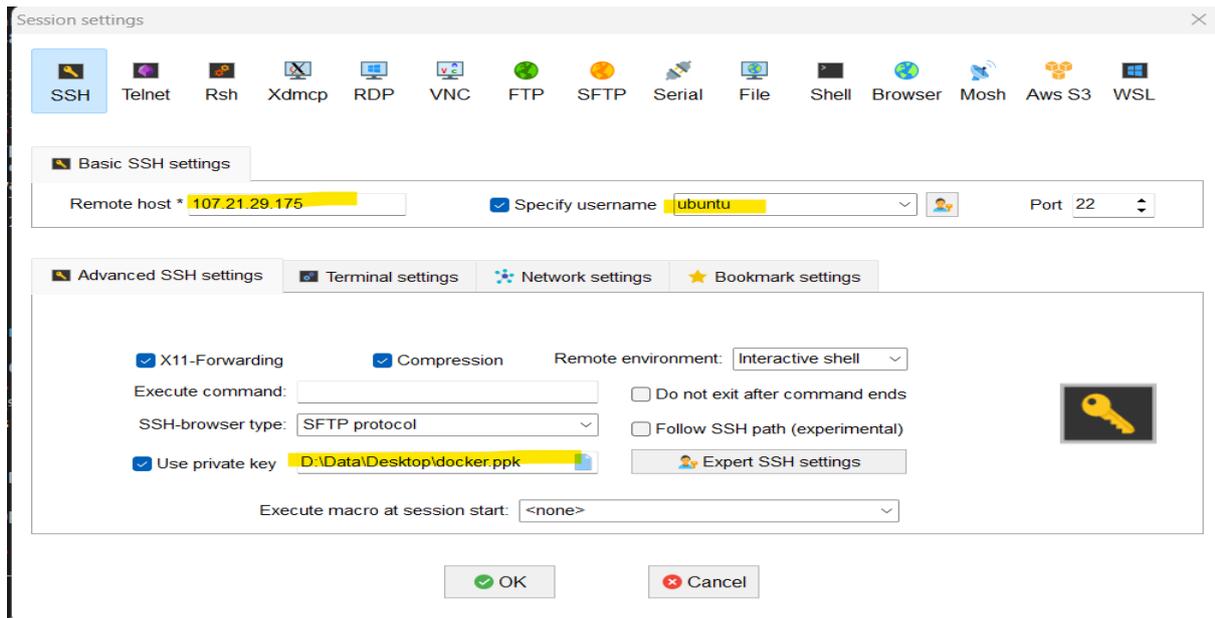


Kubernetes Worker2 has been done with passwordless ssh.

Third Take Kubernetes Worker3.

Now Take Kubernetes Worker3 Server we used same process.

Run this command `cat id_rsa.pub`. you will show ssh key. this key need to be copy and more 5 instances need to paste in same location `cd /home/ubuntu/.ssh/authorized_keys`



```
ubuntu@ip-172-31-91-37:~$ cd /home/ubuntu/.ssh/
ubuntu@ip-172-31-91-37:~/.ssh$ ll
total 12
drwx----- 2 ubuntu ubuntu 4096 Mar 29 18:04 ./
drwxr-x--- 4 ubuntu ubuntu 4096 Mar 29 18:34 ../
-rw----- 1 ubuntu ubuntu 388 Mar 29 18:04 authorized_keys
ubuntu@ip-172-31-91-37:~/.ssh$
```

```
GNU nano 7.2 authorized keys *
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQwJDfZ6pz2B1JV6ZoyJyc3mdbosVETXUpug8mXGhKMyHLF5LKxik8PMTIxNwc1IXJGAMM70Z7MVBf40aGZ9VmzL21HdwmrQA/fTaUqRwNjfvQq+3Wk/ZCY6svh
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQ0YEOlcPvgCUBDRrB9/EaYvu5NAF0at12L18k1LGkg/6QV1ruc6x0#T0X0n5LaLn68Zf9av6TY9xodrTQmaA1YIcr1RYPHo+dDL9FapIL52GAmX6f0Yo1UUs4M
```

```
ubuntu@ip-172-31-31-181:~/.ssh$ ssh 172.31.91.37
The authenticity of host '172.31.91.37 (172.31.91.37)' can't be established.
ED25519 key fingerprint is SHA256:/Eh4ogSQbFaM9kP6eTiaMoczW/Jgo1814IkM53Bji6A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.91.37' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

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

System information as of Sat Mar 29 18:36:11 UTC 2025

System load:  0.06          Processes:    110
Usage of /:   25.2% of 6.71GB  Users logged in: 1
Memory usage: 22%          IPv4 address for enX0: 172.31.91.37
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

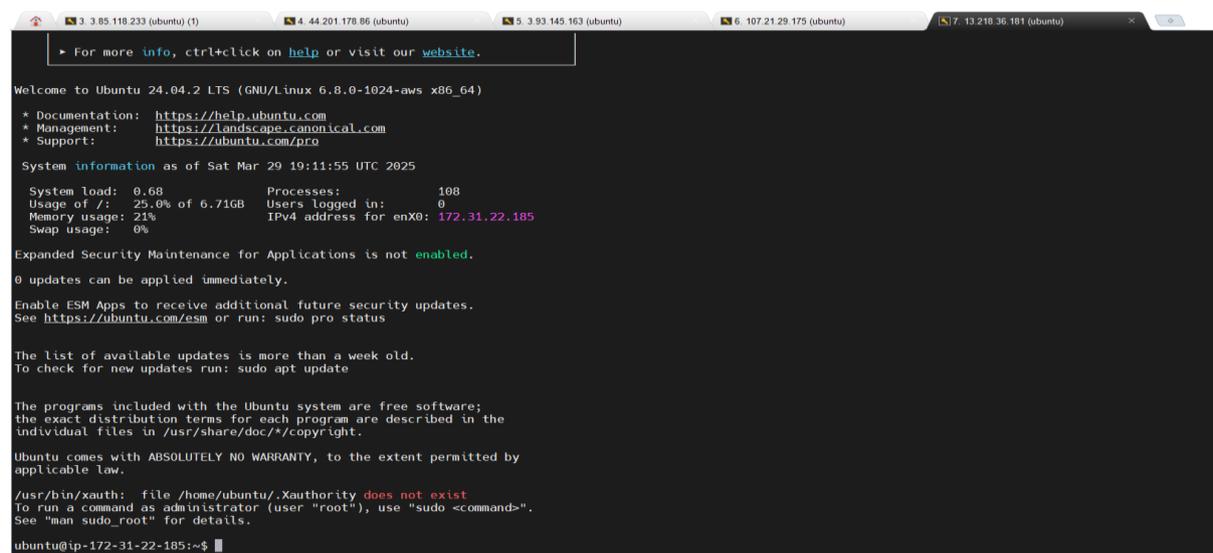
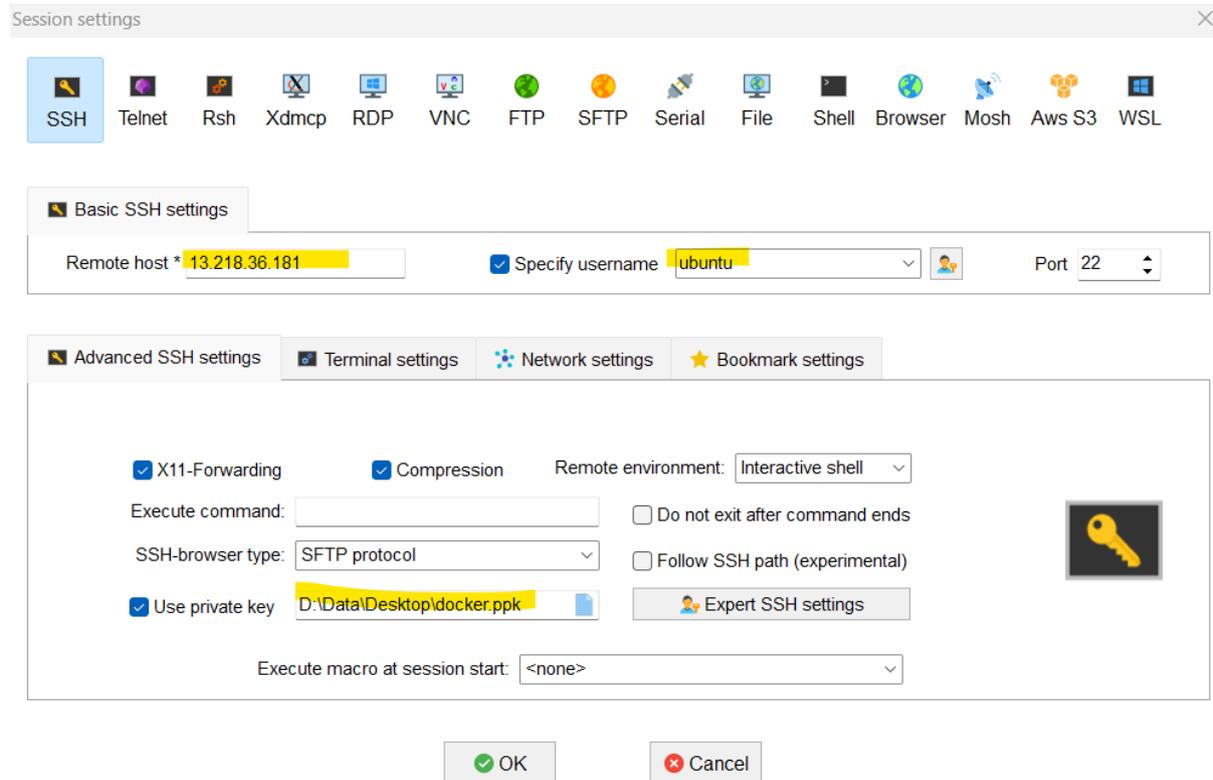
Last login: Sat Mar 29 18:34:35 2025 from 106.222.204.95
ubuntu@ip-172-31-91-37:~$
```

Kubernetes Worker3 has been done with passwordless ssh.

Fourth Take Kubernetes Worker4.

Now Take Kubernetes Worker4 Server we used same process.

Run this command `cat id_rsa.pub` you will show ssh key. this key need to be copy and more 5 instances need to paste in same location `cd /home/ubuntu/.ssh/authorized_keys`



```
ubuntu@ip-172-31-22-185:~/ssh$ ll
total 12
drwx----- 2 ubuntu ubuntu 4096 Mar 29 19:13 ./
drwxr-x--- 4 ubuntu ubuntu 4096 Mar 29 19:12 ../
-rw----- 1 ubuntu ubuntu 965 Mar 29 19:13 authorized_keys
ubuntu@ip-172-31-22-185:~/ssh$
```

```
GNU nano 7.2 authorized keys *
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDwJDFz6pZ281JV6ZoyJyc3mDbosVtIXUpug9mXGhKMyHLF51Kx1k8PMtLxNwc1TXJgAMM7027MVBr40aGZ9VmzL21HdwmrqA/rTaUgRwVNIffVgQ+3Wk/ZCYesvB
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDg0YEOLcPvgCubDRrB97EaYvu5NAF0atT2Li8K1lGkg/6QV1ruc6x0+T0X6n5LaLn68Zf9av6TY9x0drTQmaAIYTCr1RYPHo+dL9FapIL52GAmX6f0YoILUrS4Mg
```

```
ubuntu@ip-172-31-31-181:~/ssh$ ssh 172.31.22.185
The authenticity of host '172.31.22.185 (172.31.22.185)' can't be established.
ED25519 key fingerprint is SHA256:hxwE0MUSQIFhjcZHgyEiJm0fi05sy5dF3NthCCaT44A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.22.185' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

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

System information as of Sat Mar 29 19:13:44 UTC 2025

System load: 0.15 Processes: 113
Usage of /: 25.3% of 6.71GB Users logged in: 1
Memory usage: 22% IPv4 address for enX0: 172.31.22.185
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

Last login: Sat Mar 29 19:11:57 2025 from 106.222.204.95
ubuntu@ip-172-31-22-185:~$
```

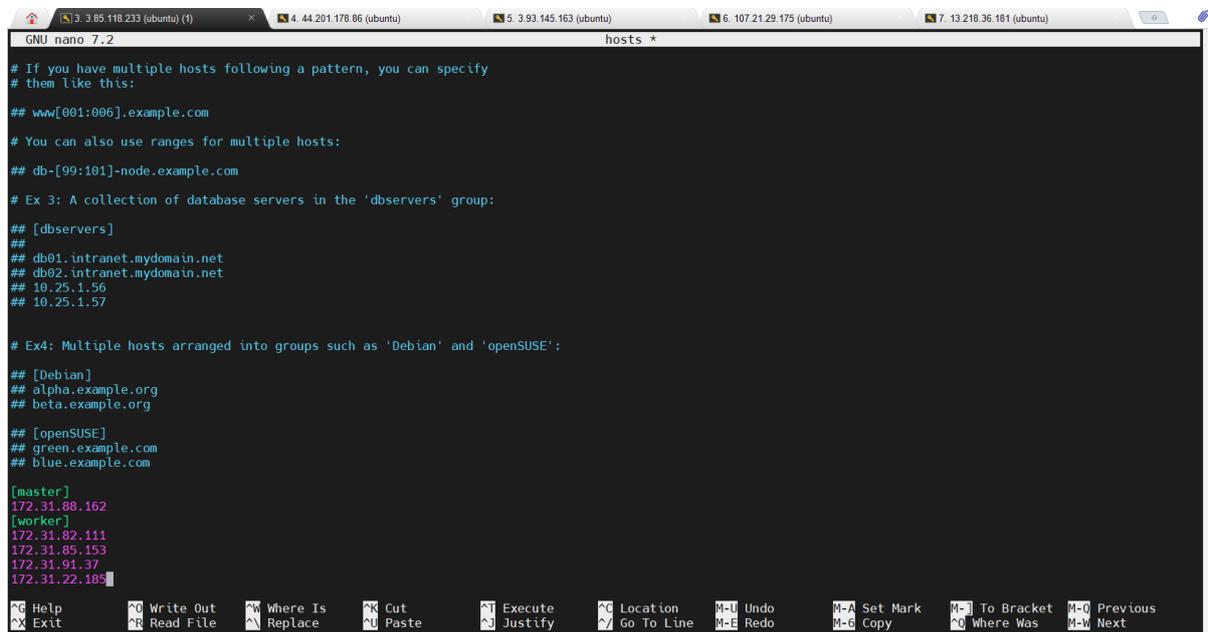
Kubernetes Worker4 has been done with passwordless ssh.

Now go to terraform/ansible server and go to this path `cd /etc/ansible` .

```
ubuntu@ip-172-31-31-181:~/ssh$ cd /etc/ansible/
ubuntu@ip-172-31-31-181:/etc/ansible$ ll
total 20
drwxr-xr-x  3 root root 4096 Mar 29 18:11 ./
drwxr-xr-x 109 root root 4096 Mar 29 18:11 ../
-rw-r--r--  1 root root  614 Mar 26 17:50 ansible.cfg
-rw-r--r--  1 root root 1175 Mar 26 17:50 hosts
drwxr-xr-x  2 root root 4096 Mar 26 17:50 roles/
ubuntu@ip-172-31-31-181:/etc/ansible$
```

Open hosts file

Sudo nano hosts



```
GNU nano 7.2 hosts *
# If you have multiple hosts following a pattern, you can specify
# them like this:
## www[001:006].example.com
# You can also use ranges for multiple hosts:
## db-[99:101]-node.example.com
# Ex 3: A collection of database servers in the 'dbbservers' group:
## [dbbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57
# Ex4: Multiple hosts arranged into groups such as 'Debian' and 'openSUSE':
## [Debian]
## alpha.example.org
## beta.example.org
## [openSUSE]
## green.example.com
## blue.example.com
[master]
172.31.88.162
[worker]
172.31.82.111
172.31.85.153
172.31.91.37
172.31.22.185
```

Now we need to add this 4 server in ansible hosts.

sudo nano script1.sh

Create the script1.sh file

#Step 2: Paste the below-given scripts here to Install the “Java” & “Jenkins”.

sudo apt-get update -y

sudo apt-get install openjdk-17-jre-headless -y

**sudo wget -O /usr/share/keyrings/jenkins-keyring.asc **

https://pkg.jenkins.io/debian/jenkins.io-2023.key

**echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] **

**https://pkg.jenkins.io/debian binary/ | sudo tee **

```
/etc/apt/sources.list.d/jenkins.list > /dev/null  
sudo apt-get update -y  
sudo apt-get install jenkins -y
```

```
sudo nano script2.sh  
sudo apt-get update -y  
sudo apt-get install openjdk-17-jre-headless -y
```

```
sudo apt-get update -y  
sudo apt install apt-transport-https curl -y
```

```
## Install containerd
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo  
gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

```
echo "deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs)  
stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update -y  
sudo apt-get install containerd.io -y
```

```
## Create containerd configuration
```

```
sudo mkdir -p /etc/containerd  
sudo containerd config default | sudo tee  
/etc/containerd/config.toml
```

```
sudo sed -i -e 's/SystemdCgroup = false/SystemdCgroup = true/g'  
/etc/containerd/config.toml
```

```
sudo systemctl restart containerd
```

```
## Install Kubernetes
```

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key |  
sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-  
keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-  
keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /' | sudo  
tee /etc/apt/sources.list.d/kubernetes.list  
sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
sudo systemctl enable --now kubelet
```

```
## Disable swap
```

```
sudo swapoff -a  
sudo modprobe br_netfilter  
sudo sysctl -w net.ipv4.ip_forward=1
```

```
sudo nano script3.sh  
sudo apt-get update -y  
sudo apt-get install openjdk-17-jre-headless -y
```

```
sudo apt-get update -y  
sudo apt install apt-transport-https curl -y
```

```
## Install containerd
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo  
gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

```
echo "deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs)  
stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update -y
```

```
sudo apt-get install containerd.io -y
```

```
## Create containerd configuration
```

```
sudo mkdir -p /etc/containerd  
sudo containerd config default | sudo tee  
/etc/containerd/config.toml
```

```
sudo sed -i -e 's/SystemdCgroup = false/SystemdCgroup = true/g'  
/etc/containerd/config.toml
```

```
sudo systemctl restart containerd
```

```
## Install Kubernetes
```

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key |  
sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-  
keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-  
keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /' | sudo  
tee /etc/apt/sources.list.d/kubernetes.list  
sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
sudo systemctl enable --now kubelet
```

```
## Disable swap
```

```
sudo swapoff -a  
sudo modprobe br_netfilter  
sudo sysctl -w net.ipv4.ip_forward=1
```

Now We need to create play1.yaml.

```
sudo nano play1.yaml
```

```
---
```

- name: install Jenkins & Java on Machine 1**
become: true
hosts: localhost
tasks:
 - name: running script1**
script: script1.sh

 - name: install Java, Docker & Kubernetes on Machine-1**
become: true
hosts: master
tasks:
 - name: running script2**
script: script2.sh

 - name: install Docker & Kubernetes on Machine-2&4**
become: true
hosts: worker
tasks:
 - name: running script3**
script: script3.sh
-

```

ubuntu@ip-172-31-31-181:/etc/ansible$ ansible -m ping all
[WARNING]: Platform linux on host 172.31.85.153 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.85.153 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.88.162 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.88.162 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.22.185 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.22.185 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.91.37 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.91.37 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}

```

```

[WARNING]: Platform linux on host 172.31.82.111 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.82.111 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-31-181:/etc/ansible$

```

```

ubuntu@ip-172-31-31-181:/etc/ansible$ ansible-playbook play1.yaml
PLAY [install Jenkins & Java on Machine 1] *****
TASK [Gathering Facts] *****
ok: [localhost]
TASK [running script1] *****
changed: [localhost]
PLAY [install Java, Docker & Kubernetes on Machine-1] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.88.162 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.88.162]
TASK [running script2] *****
changed: [172.31.88.162]
PLAY [install Docker & Kubernetes on Machine-2&4] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.82.111 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.82.111]
[WARNING]: Platform linux on host 172.31.22.185 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.22.185]
[WARNING]: Platform linux on host 172.31.85.153 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.85.153]
[WARNING]: Platform linux on host 172.31.91.37 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.91.37]

```

```

[WARNING]: Platform linux on host 172.31.88.162 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.88.162]
TASK [running script2] *****
changed: [172.31.88.162]
PLAY [install Docker & Kubernetes on Machine-2&4] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.82.111 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.82.111]
[WARNING]: Platform linux on host 172.31.22.185 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.22.185]
[WARNING]: Platform linux on host 172.31.85.153 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.85.153]
[WARNING]: Platform linux on host 172.31.91.37 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.91.37]
TASK [running script3] *****
changed: [172.31.85.153]
changed: [172.31.82.111]
changed: [172.31.91.37]
changed: [172.31.22.185]
PLAY RECAP *****
172.31.22.185 : ok=2  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
172.31.82.111 : ok=2  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
172.31.85.153 : ok=2  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
172.31.88.162 : ok=2  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
172.31.91.37 : ok=2  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
localhost   : ok=2  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
ubuntu@ip-172-31-31-181:/etc/ansible$

```

Go to first terraform/ansible server and check version

Jenkins --version

Java -version

```
ubuntu@ip-172-31-31-181:/etc/ansible$ jenkins --version
2.503
ubuntu@ip-172-31-31-181:/etc/ansible$ java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-124.04, mixed mode, sharing)
ubuntu@ip-172-31-31-181:/etc/ansible$ █
```

Now Go to master server and run

sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```
ubuntu@ip-172-31-88-162:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
I0329 19:39:09.311700 6934 version.go:256] remote version is much newer: v1.32.3; falling back to: stable-1.30
[tinit] Using Kubernetes version: v1.30.11
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0329 19:39:09.547479 6934 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
```

```
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
```

Your Kubernetes control-plane has initialized **successfully!**

To start using your cluster, you need to run the following as a regular user:

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Alternatively, if you are the root user, you can run:

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at: <https://kubernetes.io/docs/concepts/cluster-administration/addons/>

Then you can join any number of worker nodes by running the following on each as root:

```
kubeadm join 172.31.88.162:6443 --token hygvf7.rt4tlqihkbk8k191 \
--discovery-token-ca-cert-hash sha256:0f83294452b21d6c4394356db458b44d833e9629dd6e4d813b99f128fe4acf22
ubuntu@ip-172-31-88-162:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-88-162:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-88-162:~$ █
```

Now we need to run the health check command **all worker nodes**

sudo kubeadm reset pre-flight checks

```
System load: 0.0          Processes: 111
Usage of /: 25.2% of 6.71GB    Users logged in: 1
Memory usage: 21%           IPv4 address for enX0: 172.31.82.111
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

Last login: Sat Mar 29 18:29:59 2025 from 172.31.31.181
ubuntu@ip-172-31-82-111:~$ sudo kubeadm reset pre-flight checks
W0329 19:41:58.009231 6475 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0329 19:42:02.317507 6475 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d

The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must remove them manually.
Please, check the contents of the $HOME/.kube/config file.
ubuntu@ip-172-31-82-111:~$
```

```
ubuntu@ip-172-31-85-153:~$ ll
total 32
drwxr-x--- 4 ubuntu ubuntu 4096 Mar 29 19:28 ./
drwxr-xr-x 3 root root 4096 Mar 29 18:04 ../
-rw-r----- 1 ubuntu ubuntu 62 Mar 29 19:08 .Xauthority
-rw-r----- 1 ubuntu ubuntu 220 Mar 31 2024 .bash_logout
-rw-r----- 1 ubuntu ubuntu 3771 Mar 31 2024 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Mar 29 19:08 .cache/
-rw-r----- 1 ubuntu ubuntu 807 Mar 31 2024 .profile
drwx----- 2 ubuntu ubuntu 4096 Mar 29 18:04 .ssh/
-rw-r----- 1 ubuntu ubuntu 0 Mar 29 19:28 .sudo_as_admin_successful
ubuntu@ip-172-31-85-153:~$ cd .ssh/
ubuntu@ip-172-31-85-153:~/.ssh$ ll
total 12
drwx----- 2 ubuntu ubuntu 4096 Mar 29 18:04 ./
drwxr-x--- 4 ubuntu ubuntu 4096 Mar 29 19:28 ../
-rw-r----- 1 ubuntu ubuntu 388 Mar 29 18:04 authorized_keys
ubuntu@ip-172-31-85-153:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-85-153:~/.ssh$ sudo kubeadm reset pre-flight checks
W0329 19:43:37.596621 5788 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0329 19:43:40.424215 5788 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d

The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must remove them manually.
Please, check the contents of the $HOME/.kube/config file.
ubuntu@ip-172-31-85-153:~/.ssh$
```

```
System load: 0.0          Processes: 111
Usage of /: 25.2% of 6.71GB  Users logged in: 1
Memory usage: 21%        IPv4 address for enX0: 172.31.91.37
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

Last login: Sat Mar 29 18:48:45 2025 from 106.222.204.95
ubuntu@ip-172-31-91-37:~$ sudo kubeadm reset pre-flight checks
W0329 19:44:06.775179 8305 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0329 19:44:09.001289 8305 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d

The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must remove them manually.
Please, check the contents of the $HOME/.kube/config file.
ubuntu@ip-172-31-91-37:~$
```

```
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.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-22-185:~$ cd /home/ubuntu/.ssh/
ubuntu@ip-172-31-22-185:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-22-185:~/.ssh$ ll
total 12
drwxr-xr-x 2 ubuntu ubuntu 4096 Mar 29 19:13 ./
drwxr-xr-x 4 ubuntu ubuntu 4096 Mar 29 19:12 ../
-rw-r--r-- 1 ubuntu ubuntu 965 Mar 29 19:13 authorized_keys
ubuntu@ip-172-31-22-185:~/.ssh$ sudo kubeadm reset pre-flight checks
W0329 19:44:40.069256 5444 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0329 19:44:42.008655 5444 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d

The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must remove them manually.
Please, check the contents of the $HOME/.kube/config file.
ubuntu@ip-172-31-22-185:~/.ssh$
```

Now After sudo kubeadm reset pre-flight checks we need to join the cluster.this below command.

```
sudo kubeadm join 172.31.88.162:6443 --token  
hygvf7.rt4tlqihkbbk8k191 --discovery-token-ca-cert-hash  
sha256:0f83294452b21d6c4394356db458b44d833e9629dd6e4d81  
3b99f128fe4acf22 --v=5
```

Worker 1

```
ubuntu@ip-172-31-82-111:~$ sudo kubeadm join 172.31.88.162:6443 --token hygvf7.r4t1q1hkb8k191 --discovery-token-ca-cert-hash sha256:0f83294452b21d6c4394356db5b44d833e9e29d6e4d813b99f128f4ac722 --v=5
I0329 19:45:53.360968 7250 join.go:417] [preflight] found NodeName empty; using OS hostname as NodeName
I0329 19:45:53.362390 7250 initconfiguration.go:122] detected and using CRI socket: unix:///var/run/containerd/containerd.sock
[preflight] Running pre-flight checks
I0329 19:45:53.362890 7250 preflight.go:93] [preflight] Running general checks
I0329 19:45:53.363015 7250 checks.go:278] validating the existence of file /etc/kubernetes/kubelet.conf
I0329 19:45:53.363199 7250 checks.go:278] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I0329 19:45:53.363271 7250 checks.go:102] validating the container runtime
I0329 19:45:53.388913 7250 checks.go:637] validating whether swap is enabled or not
I0329 19:45:53.389061 7250 checks.go:368] validating the presence of executable crictl
I0329 19:45:53.389110 7250 checks.go:368] validating the presence of executable contrack
I0329 19:45:53.389270 7250 checks.go:368] validating the presence of executable ip
I0329 19:45:53.389299 7250 checks.go:368] validating the presence of executable iptables
I0329 19:45:53.389442 7250 checks.go:368] validating the presence of executable mount
I0329 19:45:53.389516 7250 checks.go:368] validating the presence of executable nsenter
I0329 19:45:53.389583 7250 checks.go:368] validating the presence of executable ethtool
I0329 19:45:53.389713 7250 checks.go:368] validating the presence of executable tc
I0329 19:45:53.389734 7250 checks.go:368] validating the presence of executable touch
I0329 19:45:53.389762 7250 checks.go:514] running all checks
I0329 19:45:53.406107 7250 checks.go:399] checking whether the given node name is valid and reachable using net.LookupHost
I0329 19:45:53.407975 7250 checks.go:603] validating kubelet version
I0329 19:45:53.451368 7250 checks.go:120] validating if the "kubelet" service is enabled and active
I0329 19:45:53.465872 7250 checks.go:201] validating availability of port 10250
I0329 19:45:53.466202 7250 checks.go:278] validating the existence of file /etc/kubernetes/pki/ca.crt
I0329 19:45:53.467214 7250 checks.go:428] validating if the connectivity type is via proxy or direct
I0329 19:45:53.467230 7250 checks.go:327] validating the contents of file /proc/sys/net/ipv4/tcp_forward
I0329 19:45:53.467260 7250 join.go:536] [preflight] discovering cluster-info
I0329 19:45:53.467283 7250 token.go:79] [discovery] Created cluster-info discovery client, requesting info from "172.31.88.162:6443"
I0329 19:45:53.488558 7250 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "hygvf7"
I0329 19:45:53.506720 7250 token.go:117] [discovery] Requesting info from "172.31.88.162:6443" again to validate TLS against the pinned public key
I0329 19:45:53.507136 7250 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "hygvf7"
I0329 19:45:53.515016 7250 token.go:134] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will use
API Server "172.31.88.162:6443"
I0329 19:45:53.515709 7250 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0329 19:45:53.515821 7250 join.go:550] [preflight] Fetching init configuration
I0329 19:45:53.515907 7250 join.go:596] [preflight] Retrieving KubeConfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
I0329 19:45:53.525410 7250 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0329 19:45:53.528724 7250 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
```

```
API Server "172.31.88.162:6443"
I0329 19:45:53.515709 7250 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0329 19:45:53.515821 7250 join.go:550] [preflight] Fetching init configuration
I0329 19:45:53.515907 7250 join.go:596] [preflight] Retrieving KubeConfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
I0329 19:45:53.525410 7250 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0329 19:45:53.528724 7250 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
I0329 19:45:53.531783 7250 initconfiguration.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
I0329 19:45:53.532021 7250 interface.go:432] Looking for default routes with IPv4 addresses
I0329 19:45:53.532965 7250 interface.go:437] Default route transits interface "enX0"
I0329 19:45:53.533134 7250 interface.go:209] Interface enX0 is up
I0329 19:45:53.533266 7250 interface.go:257] Interface "enX0" has 2 addresses : [172.31.82.111/20 fe80::108a:e2ff:feb7:2bfb/64].
I0329 19:45:53.533412 7250 interface.go:224] Checking addr 172.31.82.111/20.
I0329 19:45:53.533514 7250 interface.go:231] IP found 172.31.82.111
I0329 19:45:53.533577 7250 interface.go:263] Found valid IPv4 address 172.31.82.111 for interface "enX0".
I0329 19:45:53.533645 7250 interface.go:443] Found active IP 172.31.82.111
I0329 19:45:53.538948 7250 preflight.go:104] [preflight] Running configuration dependant checks
I0329 19:45:53.539059 7250 controlplaneprepare.go:225] [download-certs] Skipping certs download
I0329 19:45:53.539121 7250 kubelet.go:124] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
I0329 19:45:53.539681 7250 kubelet.go:139] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
I0329 19:45:53.540742 7250 kubelet.go:160] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-82-111" and status "Ready"
I0329 19:45:53.544224 7250 kubelet.go:175] [kubelet-start] Stopping the kubelet
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://172.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 533.483283ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
I0329 19:45:55.394688 7250 kubelet.go:242] [kubelet-start] preserving the crisocket information for the node
I0329 19:45:55.394709 7250 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/containerd/containerd.sock" to the Node API object "ip-172-31-82-111" as an annotation
I0329 19:45:55.394909 7250 cert_rotation.go:137] Starting client certificate rotation controller

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-82-111:~$ █
```

Worker2

```
3 3.85.118.233 (ubuntu) | 4 44.203.3.140 (ubuntu) | 4 44.201.178.86 (ubuntu) | 5 3.93.145.163 (ubuntu) | 6 107.21.29.175 (ubuntu) | 7 13.218.36.181 (ubuntu)

The reset process does not clean your kubeconfig files and you must remove them manually.
Please, check the contents of the $HOME/.kube/config file.
ubuntu@ip-172-31-85-153:~$ ssh$ sudo kubeadm join 172.31.88.162:6443 --token hygvf7.r4t1qlqhk8k191 --discovery-token-ca-cert-hash sha256:0f83294452b21d6c4394356
db458b44d833e9629dd0e4d813b99f128fe4acf22 --v=5
I0329 19:46:47.201388 6044 join.go:417] [preflight] found NodeName empty; using OS hostname as NodeName
I0329 19:46:47.201733 6044 initconfiguration.go:122] detected and using CRI socket: unix:///var/run/containerd/containerd.sock
[preflight] Running pre-flight checks
I0329 19:46:47.202096 6044 preflight.go:93] [preflight] Running general checks
I0329 19:46:47.202408 6044 checks.go:278] validating the existence of file /etc/kubernetes/kubelet.conf
I0329 19:46:47.202568 6044 checks.go:278] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I0329 19:46:47.202678 6044 checks.go:102] validating the container runtime
I0329 19:46:47.232282 6044 checks.go:637] validating whether swap is enabled or not
I0329 19:46:47.232703 6044 checks.go:368] validating the presence of executable crictl
I0329 19:46:47.232817 6044 checks.go:368] validating the presence of executable contrack
I0329 19:46:47.232841 6044 checks.go:368] validating the presence of executable ip
I0329 19:46:47.232918 6044 checks.go:368] validating the presence of executable iptables
I0329 19:46:47.234585 6044 checks.go:368] validating the presence of executable mount
I0329 19:46:47.234639 6044 checks.go:368] validating the presence of executable nsenter
I0329 19:46:47.235888 6044 checks.go:368] validating the presence of executable ethtool
I0329 19:46:47.236046 6044 checks.go:368] validating the presence of executable tc
I0329 19:46:47.236867 6044 checks.go:368] validating the presence of executable touch
I0329 19:46:47.236885 6044 checks.go:514] running all checks
I0329 19:46:47.253404 6044 checks.go:399] checking whether the given node name is valid and reachable using net.LookupHost
I0329 19:46:47.255491 6044 checks.go:603] validating kubelet version
I0329 19:46:47.301943 6044 checks.go:128] validating if the "kubelet" service is enabled and active
I0329 19:46:47.320630 6044 checks.go:201] validating availability of port 10250
I0329 19:46:47.320991 6044 checks.go:278] validating the existence of file /etc/kubernetes/pki/ca.crt
I0329 19:46:47.321113 6044 checks.go:428] validating if the connectivity type is via proxy or direct
I0329 19:46:47.321215 6044 checks.go:327] validating the contents of file /proc/sys/net/ipv4/ip_forward
I0329 19:46:47.321256 6044 join.go:536] [preflight] Discovering cluster-info
I0329 19:46:47.321334 6044 token.go:79] [discovery] Created cluster-info discovery client, requesting info from "172.31.88.162:6443"
I0329 19:46:47.343542 6044 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "hygvf7"
I0329 19:46:47.363555 6044 token.go:117] [discovery] Requesting info from "172.31.88.162:6443" again to validate TLS against the pinned public key
I0329 19:46:47.363904 6044 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "hygvf7"
I0329 19:46:47.374207 6044 token.go:134] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will use
API Server "172.31.88.162:6443"
I0329 19:46:47.374389 6044 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0329 19:46:47.374510 6044 join.go:550] [preflight] Fetching init configuration
I0329 19:46:47.374614 6044 join.go:596] [preflight] Retrieving KubeConfig objects
[preflight] Reading configuration from the cluster...
```

```
3 3.85.118.233 (ubuntu) | 4 44.203.3.140 (ubuntu) | 4 44.201.178.86 (ubuntu) | 5 3.93.145.163 (ubuntu) | 6 107.21.29.175 (ubuntu) | 7 13.218.36.181 (ubuntu)

API Server "172.31.88.162:6443"
I0329 19:46:47.374389 6044 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0329 19:46:47.374510 6044 join.go:550] [preflight] Fetching init configuration
I0329 19:46:47.374614 6044 join.go:596] [preflight] Retrieving KubeConfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
I0329 19:46:47.393199 6044 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0329 19:46:47.398861 6044 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
I0329 19:46:47.391874 6044 initconfiguration.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
I0329 19:46:47.392394 6044 interface.go:432] Looking for default routes with IPv4 addresses
I0329 19:46:47.392543 6044 interface.go:437] Default route transit interface "enX0"
I0329 19:46:47.392742 6044 interface.go:209] Interface enX0 is up
I0329 19:46:47.392875 6044 interface.go:257] Interface "enX0" has 2 addresses :[172.31.85.153/20 fe80::1033:79ff:fee8:443/64].
I0329 19:46:47.393010 6044 interface.go:224] Checking addr 172.31.85.153/20.
I0329 19:46:47.393082 6044 interface.go:231] IP found 172.31.85.153
I0329 19:46:47.393096 6044 interface.go:263] Found valid IPv4 address 172.31.85.153 for interface "enX0".
I0329 19:46:47.393102 6044 interface.go:443] Found active IP 172.31.85.153
I0329 19:46:47.399080 6044 preflight.go:104] [preflight] Running configuration dependant checks
I0329 19:46:47.399217 6044 controlplaneprepare.go:225] [download-certs] Skipping certs download
I0329 19:46:47.399325 6044 kubelet.go:124] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
I0329 19:46:47.399877 6044 kubelet.go:139] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
I0329 19:46:47.401372 6044 kubelet.go:160] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-85-153" and status "Ready"
I0329 19:46:47.405469 6044 kubelet.go:175] [kubelet-start] Stopping the kubelet
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://172.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.004696629s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
I0329 19:46:48.723213 6044 kubelet.go:242] [kubelet-start] preserving the crisocket information for the node
I0329 19:46:48.723439 6044 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/containerd/containerd.sock" to the Node API object "ip-172-31-85-153" as an annotation
I0329 19:46:48.723801 6044 cert_rotation.go:137] Starting client certificate rotation controller

This node has joined the cluster:
* Certificate signing request was sent to apserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-85-153:~$ ssh$
```

Worker3

```
ubuntu@ip-172-31-91-37:~$ sudo kubeadm join 172.31.88.162:6443 --token hygvf7.r4t4lqihk8k191 --discovery-token-ca-cert-hash sha256:0f83294452b21d6c434356db58b44d833e962dd0e4d813b99f128f4eac72 --v=5
I0329 19:47:29.840860 8534 join.go:417] [preflight] found NodeName empty; using OS hostname as NodeName
I0329 19:47:29.842311 8534 initconfiguration.go:122] detected and using CRI socket: unix:///var/run/containerd/containerd.sock
[preflight] Running pre-flight checks
I0329 19:47:29.842852 8534 preflight.go:93] [preflight] Running general checks
I0329 19:47:29.843070 8534 checks.go:278] validating the existence of file /etc/kubernetes/kubelet.conf
I0329 19:47:29.843211 8534 checks.go:278] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I0329 19:47:29.843477 8534 checks.go:102] validating the container runtime
I0329 19:47:29.873016 8534 checks.go:637] validating whether swap is enabled or not
I0329 19:47:29.873242 8534 checks.go:368] validating the presence of executable crictl
I0329 19:47:29.873367 8534 checks.go:368] validating the presence of executable contrack
I0329 19:47:29.873480 8534 checks.go:368] validating the presence of executable ip
I0329 19:47:29.873584 8534 checks.go:368] validating the presence of executable iptables
I0329 19:47:29.874215 8534 checks.go:368] validating the presence of executable mount
I0329 19:47:29.874352 8534 checks.go:368] validating the presence of executable nsetool
I0329 19:47:29.874474 8534 checks.go:368] validating the presence of executable nsenter
I0329 19:47:29.874583 8534 checks.go:368] validating the presence of executable ethtool
I0329 19:47:29.874668 8534 checks.go:368] validating the presence of executable tc
I0329 19:47:29.874762 8534 checks.go:514] running all checks
I0329 19:47:29.893347 8534 checks.go:399] checking whether the given node name is valid and reachable using net.LookupHost
I0329 19:47:29.894874 8534 checks.go:603] validating kubelet version
I0329 19:47:29.938510 8534 checks.go:128] validating if the "kubelet" service is enabled and active
I0329 19:47:29.953261 8534 checks.go:201] validating availability of port 10250
I0329 19:47:29.954858 8534 checks.go:278] validating the existence of file /etc/kubernetes/pki/ca.crt
I0329 19:47:29.954871 8534 checks.go:428] validating if the connectivity file is via proxy or direct
I0329 19:47:29.954895 8534 checks.go:327] validating the contents of file /proc/sys/net/ipv4/ip_forward
I0329 19:47:29.954127 8534 join.go:536] [preflight] Discovering cluster-info
I0329 19:47:29.954149 8534 token.go:79] [discovery] Created cluster-info discovery client, requesting info from "172.31.88.162:6443"
I0329 19:47:29.978510 8534 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "hygvf7"
I0329 19:47:29.996764 8534 token.go:117] [discovery] Requesting info from "172.31.88.162:6443" again to validate TLS against the pinned public key
I0329 19:47:29.997192 8534 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "hygvf7"
I0329 19:47:30.006029 8534 token.go:134] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will use API Server "172.31.88.162:6443"
I0329 19:47:30.006413 8534 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0329 19:47:30.006540 8534 join.go:550] [preflight] Fetching init configuration
I0329 19:47:30.006631 8534 join.go:596] [preflight] Retrieving kubeconfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
I0329 19:47:30.014319 8534 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0329 19:47:30.018049 8534 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
```

```
API Server "172.31.88.162:6443"
I0329 19:47:30.006413 8534 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0329 19:47:30.006540 8534 join.go:550] [preflight] Fetching init configuration
I0329 19:47:30.006631 8534 join.go:596] [preflight] Retrieving kubeconfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
I0329 19:47:30.014319 8534 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0329 19:47:30.018049 8534 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
I0329 19:47:30.021867 8534 initconfiguration.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
I0329 19:47:30.022154 8534 interface.go:432] Looking for default routes with IPv4 addresses
I0329 19:47:30.022291 8534 interface.go:437] Default route transits interface "enX0"
I0329 19:47:30.022875 8534 interface.go:209] Interface enX0 is up
I0329 19:47:30.023010 8534 interface.go:257] Interface "enX0" has 2 addresses :[172.31.91.37/20 fe80::10ae:edff:fe0e:55e3/64].
I0329 19:47:30.023125 8534 interface.go:224] Checking addr 172.31.91.37/20.
I0329 19:47:30.023236 8534 interface.go:231] IP found 172.31.91.37
I0329 19:47:30.023343 8534 interface.go:263] Found valid IPv4 address 172.31.91.37 for interface "enX0".
I0329 19:47:30.023441 8534 interface.go:443] Found active IP 172.31.91.37
I0329 19:47:30.029044 8534 preflight.go:104] [preflight] Running configuration dependant checks
I0329 19:47:30.029183 8534 controlplaneprepare.go:225] [download-certs] Skipping certs download
I0329 19:47:30.029281 8534 kubelet.go:124] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
I0329 19:47:30.029858 8534 kubelet.go:139] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
I0329 19:47:30.030488 8534 kubelet.go:160] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-91-37" and status "Ready"
I0329 19:47:30.033643 8534 kubelet.go:175] [kubelet-start] Stopping the kubelet
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.00217588s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
I0329 19:47:31.358947 8534 kubelet.go:242] [kubelet-start] preserving the crsocket information for the node
I0329 19:47:31.359220 8534 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/containerd/containerd.sock" to the Node API object "ip-172-31-91-37" as an annotation
I0329 19:47:31.359122 8534 cert_rotation.go:137] Starting client certificate rotation controller

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
ubuntu@ip-172-31-91-37:~$
```

Worker4

```
ubuntu@ip-172-31-22-185:~$ ssh -i /root/.ssh/id_rsa ubuntu@172.31.88.162:~$ sudo kubeadm join 172.31.88.162:6443 --token hygvf7.r4t4tlqihkbk8k191 --discovery-token-ca-cert-hash sha256:0f83294452b21d6c4394356db458b4d833e9629d6e44813b99f128fe4acf22 --v=5
10329 19:48:09.462579 5690 join.go:417] [preflight] Found nodeName empty; using OS hostname as nodeName
10329 19:48:09.465085 5690 initconfiguration.go:122] detected and using CRI socket: unix:///var/run/containerd/containerd.sock
[preflight] Running pre-flight checks
10329 19:48:09.465471 5690 preflight.go:93] [preflight] Running general checks
10329 19:48:09.466335 5690 checks.go:278] validating the existence of file /etc/kubernetes/kubelet.conf
10329 19:48:09.466452 5690 checks.go:278] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
10329 19:48:09.466591 5690 checks.go:278] validating the container runtime
10329 19:48:09.493268 5690 checks.go:367] validating whether swap is enabled or not
10329 19:48:09.493477 5690 checks.go:368] validating the presence of executable crictl
10329 19:48:09.493582 5690 checks.go:368] validating the presence of executable contrack
10329 19:48:09.493631 5690 checks.go:368] validating the presence of executable ip
10329 19:48:09.493699 5690 checks.go:368] validating the presence of executable iptables
10329 19:48:09.494954 5690 checks.go:368] validating the presence of executable mount
10329 19:48:09.495139 5690 checks.go:368] validating the presence of executable nsetool
10329 19:48:09.495303 5690 checks.go:368] validating the presence of executable nssm
10329 19:48:09.495438 5690 checks.go:368] validating the presence of executable tc
10329 19:48:09.495554 5690 checks.go:368] validating the presence of executable touch
10329 19:48:09.495601 5690 checks.go:351] running all checks
10329 19:48:09.549501 5690 checks.go:399] checking whether the given node name is valid and reachable using net.LookupHost
10329 19:48:09.558645 5690 checks.go:683] validating kubelet version
10329 19:48:09.596014 5690 checks.go:128] validating if the "kubelet" service is enabled and active
10329 19:48:09.610123 5690 checks.go:201] validating availability of port 10250
10329 19:48:09.610259 5690 checks.go:278] validating the existence of file /etc/kubernetes/pki/ca.crt
10329 19:48:09.610274 5690 checks.go:428] validating if the connectivity type is via proxy or direct
10329 19:48:09.610295 5690 checks.go:327] validating the contents of file /proc/sys/net/ipv4/ip_forward
10329 19:48:09.610314 5690 join.go:536] [preflight] Discovering cluster-info
10329 19:48:09.610466 5690 token.go:79] [discovery] Created cluster-info discovery client, requesting info from "172.31.88.162:6443"
10329 19:48:09.631040 5690 token.go:240] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "hygvf7"
10329 19:48:09.649013 5690 token.go:117] [discovery] Requesting info from "172.31.88.162:6443" again to validate TLS against the pinned public key
10329 19:48:09.658282 5690 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "hygvf7"
10329 19:48:09.657852 5690 token.go:134] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will use API Server "172.31.88.162:6443"
10329 19:48:09.658023 5690 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
10329 19:48:09.658150 5690 join.go:550] [preflight] Fetching init configuration
10329 19:48:09.658360 5690 join.go:596] [preflight] Retrieving KubeConfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
10329 19:48:09.666652 5690 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
10329 19:48:09.670862 5690 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
```

```
API Server "172.31.88.162:6443"
10329 19:48:09.658023 5690 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
10329 19:48:09.658150 5690 join.go:550] [preflight] Fetching init configuration
10329 19:48:09.658360 5690 join.go:596] [preflight] Retrieving KubeConfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
10329 19:48:09.666652 5690 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
10329 19:48:09.670862 5690 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
10329 19:48:09.675143 5690 initconfiguration.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
10329 19:48:09.675384 5690 interface.go:432] Looking for default routes with IPv4 addresses
10329 19:48:09.675497 5690 interface.go:437] Default route transits interface "enX0"
10329 19:48:09.675656 5690 interface.go:209] interface enX0 is up
10329 19:48:09.675792 5690 interface.go:257] interface "enX0" has 2 addresses: [172.31.22.185/20 fe80::8ff:eff:fec5:e069/64].
10329 19:48:09.676016 5690 interface.go:224] checking addr 172.31.22.185/20.
10329 19:48:09.676026 5690 interface.go:231] IP found 172.31.22.185
10329 19:48:09.676132 5690 interface.go:263] Found valid IPv4 address 172.31.22.185 for interface "enX0".
10329 19:48:09.676200 5690 interface.go:443] Found active IP 172.31.22.185
10329 19:48:09.681813 5690 preflight.go:104] [preflight] Running configuration dependant checks
10329 19:48:09.681833 5690 controlplaneprepare.go:225] [download-certs] Skipping certs download
10329 19:48:09.682078 5690 kubelet.go:124] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
10329 19:48:09.682597 5690 kubelet.go:139] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
10329 19:48:09.683207 5690 kubelet.go:160] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-22-185" and status "Ready"
10329 19:48:09.686329 5690 kubelet.go:175] [kubelet-start] Stopping the kubelet
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://172.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.000799335s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
10329 19:48:10.999546 5690 kubelet.go:242] [kubelet-start] preserving the cri socket information for the node
10329 19:48:10.999559 5690 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/containerd/containerd.sock" to the Node API object "ip-172-31-22-185" as an annotation
10329 19:48:10.999722 5690 cert_rotation.go:137] Starting client certificate rotation controller

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response was received.
* The Kubelet was informed of the new secure connection details.

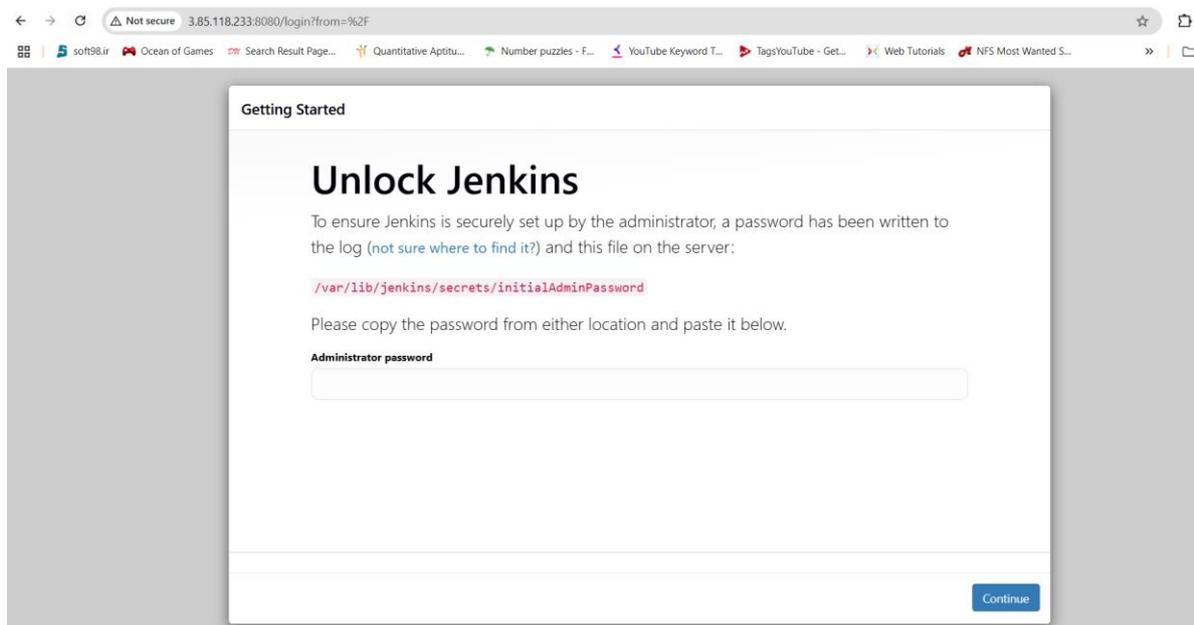
Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-22-185:~$
```

Then go to Kubernetes master node and run `kubectl get nodes`

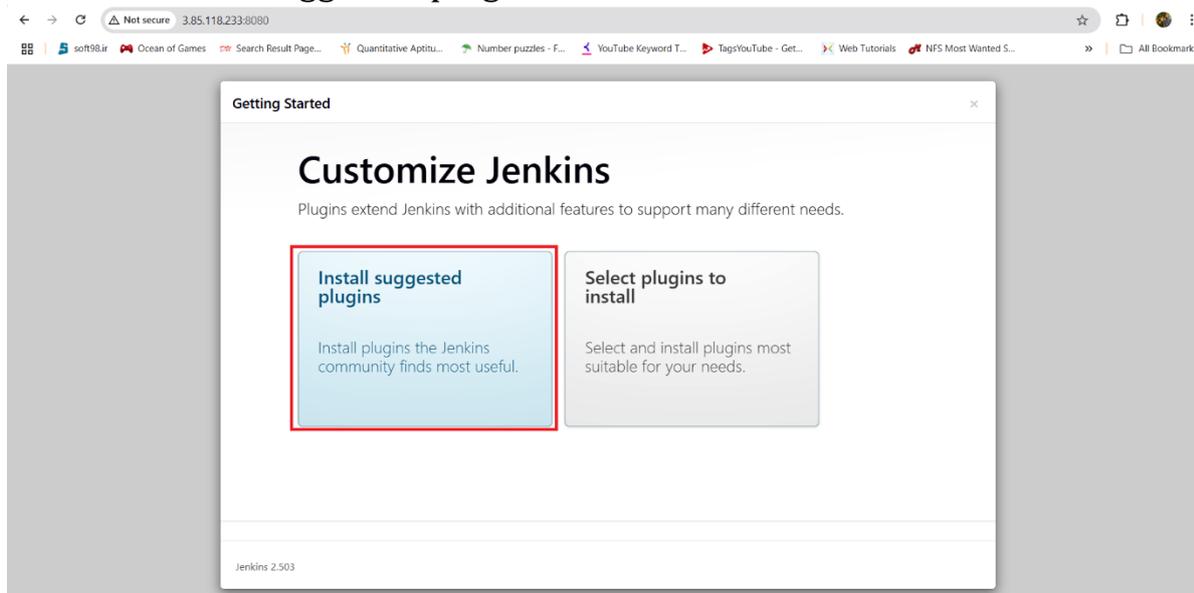
```
ubuntu@ip-172-31-88-162:~$ kubectl get nodes
NAME                 STATUS    ROLES    AGE     VERSION
ip-172-31-22-185    Ready    <none>   40s    v1.30.11
ip-172-31-82-111    Ready    <none>   2m56s  v1.30.11
ip-172-31-85-153    Ready    <none>   2m2s   v1.30.11
ip-172-31-88-162    Ready    control-plane 9m18s  v1.30.11
ip-172-31-91-37     Ready    <none>   79s    v1.30.11
ubuntu@ip-172-31-88-162:~$
```

All Nodes Joined Successfully Configure Jenkins Setup Properly Here on terraform/ansible machine.



```
ubuntu@ip-172-31-31-181:/etc/ansible$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
84cf273f744e40c8bb1f234dfb453  
ubuntu@ip-172-31-31-181:/etc/ansible$
```

Click the Install suggested plugins.



Getting Started

Getting Started

<input checked="" type="checkbox"/> Folders	<input checked="" type="checkbox"/> OWASP Markup Formatter	<input checked="" type="checkbox"/> Build Timeout	<input checked="" type="checkbox"/> Credentials Binding
<input checked="" type="checkbox"/> Timestamper	<input checked="" type="checkbox"/> Workspace Cleanup	<input checked="" type="checkbox"/> Ant	<input checked="" type="checkbox"/> Gradle
<input type="checkbox"/> Pipeline	<input type="checkbox"/> GitHub Branch Source	<input type="checkbox"/> Pipeline: GitHub Groovy Libraries	<input type="checkbox"/> Pipeline Graph View
<input type="checkbox"/> Git	<input type="checkbox"/> SSH Build Agents	<input type="checkbox"/> Matrix Authorization Strategy	<input type="checkbox"/> PAM Authentication
<input type="checkbox"/> LDAP	<input type="checkbox"/> Email Extension	<input checked="" type="checkbox"/> Mailer	<input type="checkbox"/> Dark Theme

Jenkins 2.503

- ** Pipeline: SCM Step
- ** Pipeline: Groovy
- ** Pipeline: Job
- ** Jakarta Activation API
- ** Jakarta Mail API
- ** Apache HttpComponents Client 4.x API
- ** Instance Identity
- Mailer**
- ** Pipeline: Basic Steps
- Gradle**
- ** Pipeline: Milestone Step
- ** Pipeline: Build Step
- ** Pipeline: Groovy Libraries
- ** Pipeline: Stage Step
- ** Sode File API
- ** Pipeline: Model API
- ** Pipeline: Declarative Extension Points API
- ** Branch API
- ** Pipeline: Multibranch
- ** Pipeline: Stage Tags Metadata
- ** - required dependency

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

Jenkins 2.503 [Skip and continue as admin](#) [Save and Continue](#)

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

Jenkins 2.503 [Skip and continue as admin](#) [Save and Continue](#)

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.503 Not now [Save and Finish](#)

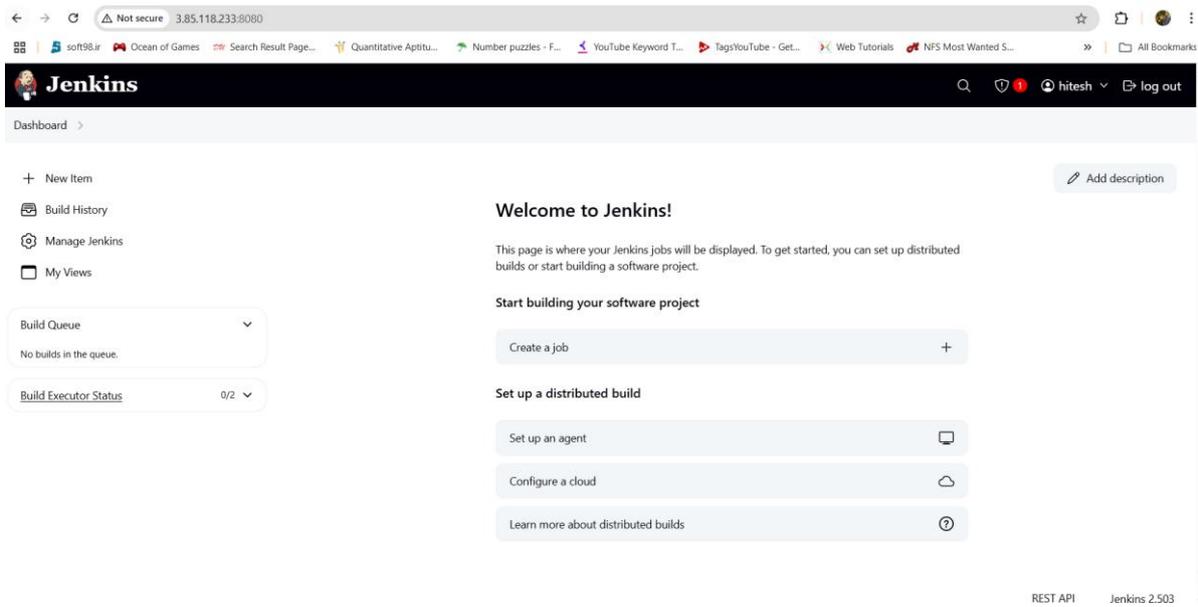
Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

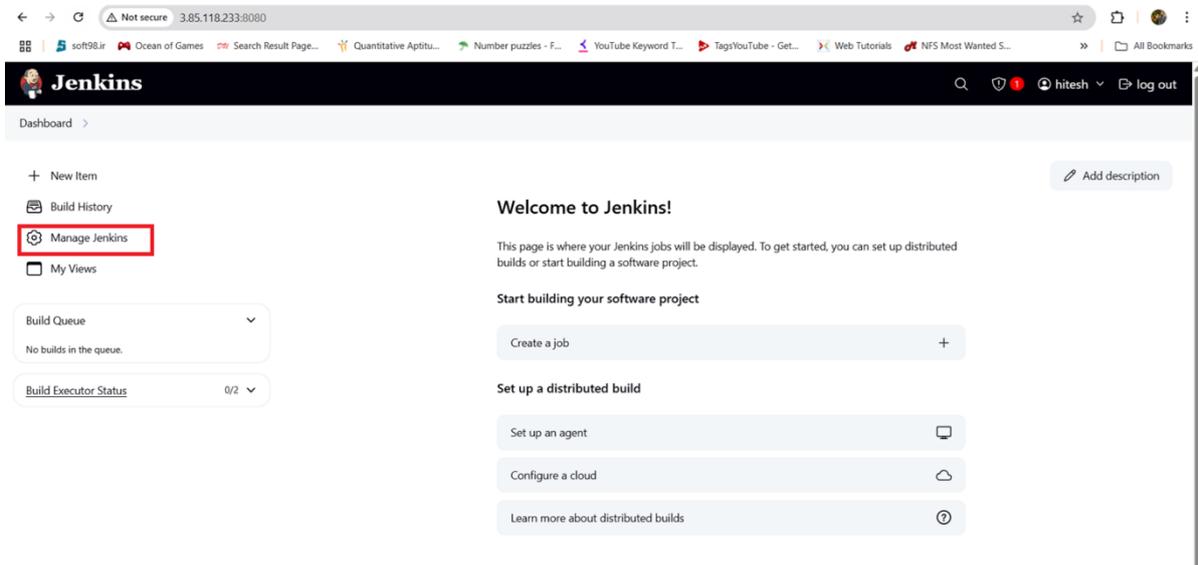
Jenkins 2.503



Jenkins Dashboard

10. Add “Kubernetes Master (Machine-3)” as a Node Here

Step 1: Click on “Set up an agent”.



← → ↻ Not secure 3.85.118.233:8080/manage/

Dashboard > Manage Jenkins

manage jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation. [Set up agent](#) [Set up cloud](#) [Dismiss](#)

System Configuration

- System: Configure global settings and paths.
- Tools: Configure tools, their locations and automatic installers.
- Plugins: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Clouds: Add, remove, and configure cloud instances to provision agents on-demand.
- Appearance: Configure the look and feel of Jenkins.

Nodes: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

- Security: Secure Jenkins; define who is allowed to access/use the system.
- Credentials: Configure credentials.
- Credential Providers: Configure the credential providers and types.
- Users: Create, delete, and modify users that can access Jenkins.

← → ↻ Not secure 3.85.118.233:8080/manage/computer/

Dashboard > Manage Jenkins > Nodes

Nodes

[+ New Node](#) [Configure Monitors](#) [Refresh](#)

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.05 GiB	0 B	3.05 GiB	0ms
	Data obtained	30 min	30 min	30 min	30 min	30 min	30 min

Icon: S M L Legend

← → ↻ Not secure 3.85.118.233:8080/manage/computer/new

Dashboard > Manage Jenkins > Nodes > New node

New node

Node name

Type

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

[Create](#)

Name ?
Kubernetes-Master

Description ?

[Plain text](#) [Preview](#)

Number of executors ?
1

Remote root directory ?
/home/ubuntu/jenkins

Kubernetes-Master

Usage ?
Use this node as much as possible

Launch method ?
Launch agents via SSH

Host ?
172.31.88.162

Credentials ?
- current -
[+ Add](#)

Host Key Verification Strategy ?
Non verifying Verification Strategy

[Save](#)

Advanced ▾

Availability ?

Keep this agent online as much as possible ▾ ?

Node Properties

- Disable deferred wipeout on this node ?
- Disk Space Monitoring Thresholds
- Environment variables
- Tool Locations

Save

← → ↻ ⚠ Not secure 3.85.118.233:8080/computer/

sof98.ir Ocean of Games Search Result Page... Quantitative Aptitu... Number puzzles - F... YouTube Keyword T... TagsYouTube - Get... Web Tutorials NFS Most Wanted S... All Bookmarks

Jenkins 🔍 🛡️ 📄 hitesh ▾ 🚪 log out

Dashboard > Nodes >

Nodes + New Node Configure Monitors ↻

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
🖥️	Built-In Node	Linux (amd64)	In sync	2.78 GiB	🔴 0 B	2.78 GiB	0ms
🖥️	Kubernetes-Master	Linux (amd64)	In sync	2.99 GiB	🔴 0 B	2.99 GiB	26ms
Data obtained		3 min 49 sec					

Build Queue: No builds in the queue.

Build Executor Status:

- Built-In Node 0/2
- Kubernetes-Master 0/1

Icon: S M L Legend

<https://github.com/hiteshchauhan89/Devops-Project-02.git>

← → ↻ github.com/hiteshchauhan89/Devops-Project-02/new/master

sof98.ir Ocean of Games Search Result Page... Quantitative Aptitu... Number puzzles - F... YouTube Keyword T... TagsYouTube - Get... Web Tutorials NFS Most Wanted S... All Bookmarks

hiteshchauhan89 / Devops-Project-02 🔍 Type [] to search

<> Code 🔄 Pull requests 🏠 Actions 📁 Projects 📖 Wiki 🛡️ Security 📊 Insights ⚙️ Settings

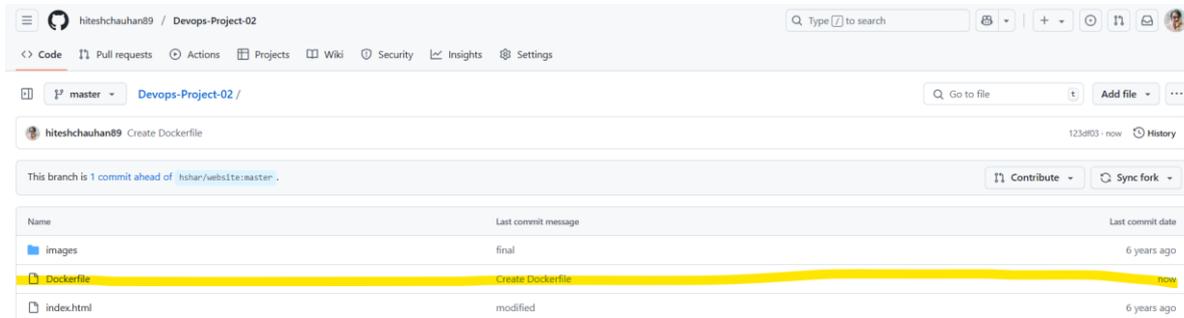
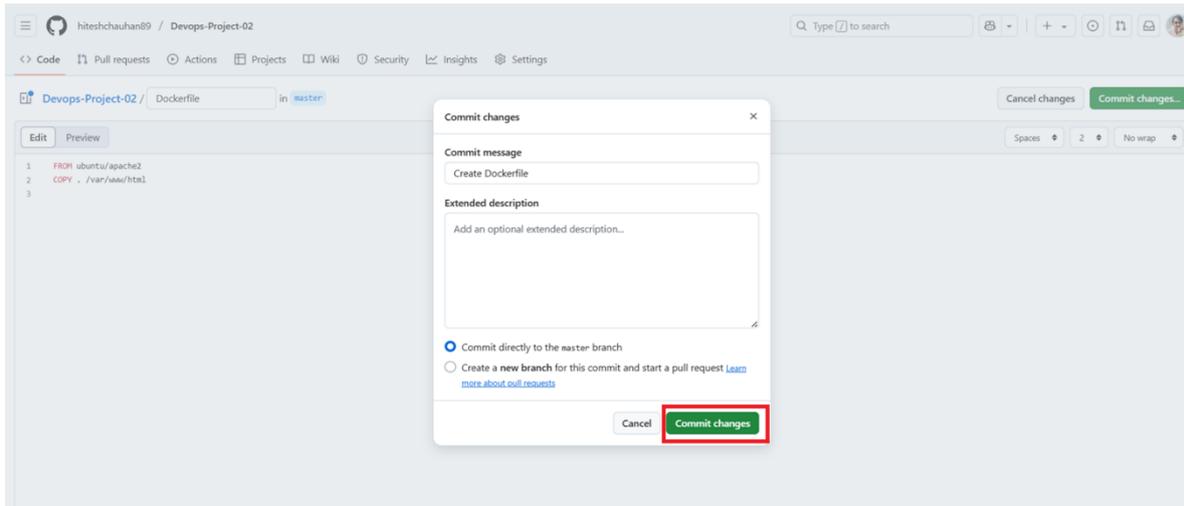
Devops-Project-02 / Dockerfile in master Cancel changes Commit changes...

Edit Preview Spaces 2 No wrap

```

1 FROM ubuntu/apache2
2 COPY ./var/www/html
3

```



Dockerfile Created Successfully

Create a Pipeline to Automate the Tasks

Note: Replace our DockerHub Credentials & Git Hub Repository URL with your DockerHub credentials & forked repository. Otherwise, build will not be successfully created.

Jenkins hitesh [log out](#)

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted)

New credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username:

Treat username as secret

Password:

ID:

Jenkins hitesh [log out](#)

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) [+ Add Credentials](#)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
3ba085a7-23b9-47d9-a4f2-4f5b56b88206	ubuntu	SSH Username with private key	
dbefe568-9e1e-4ab3-9d89-66aaaf7edb8c	hiteshtech003/*****	Username with password	

Icon: S M L

Step 1: Click on "Create a job".

Not secure 3.85.118.233:8080

Jenkins hitesh [log out](#)

Dashboard

- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views

Build Queue: No builds in the queue.

Build Executor Status:

- Built-in Node: 0/2
- Kubernetes-Master: 0/1

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

REST API Jenkins 2.503

Step 2:

Not secure 3.85.118.233:8080/view/all/newJob

soft98.ir Ocean of Games Search Result Page... Quantitative Aptit... Number puzzles - F... YouTube Keyword T... TagsYouTube - Get... Web Tutorials NFS Most Wanted S...

Jenkins

Dashboard > All > New Item

New Item

Enter an item name

Testpipeline

Select an item type

- Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**
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.
- Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder

OK

Create the New Pipeline

Step 3: Go to the “Pipeline” section & choose the “Hello World” script here.

Choose the “Hello World” Script Here

Step 4: Now, we will use the below-given script to check whether the pipeline script is working properly or not.

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent none
3   environment {
4     DOCKERHUB_CREDENTIALS=credentials("dbefe568-9e1e-4ab3-9d89-66aef7edb8c")
5   }
6   stages {
7     stage('Hello') {
8       steps {
9         echo 'Hello World'
10      }
11    }
12  }
13 }
14
```

Use Groovy Sandbox ?

Pipeline Syntax

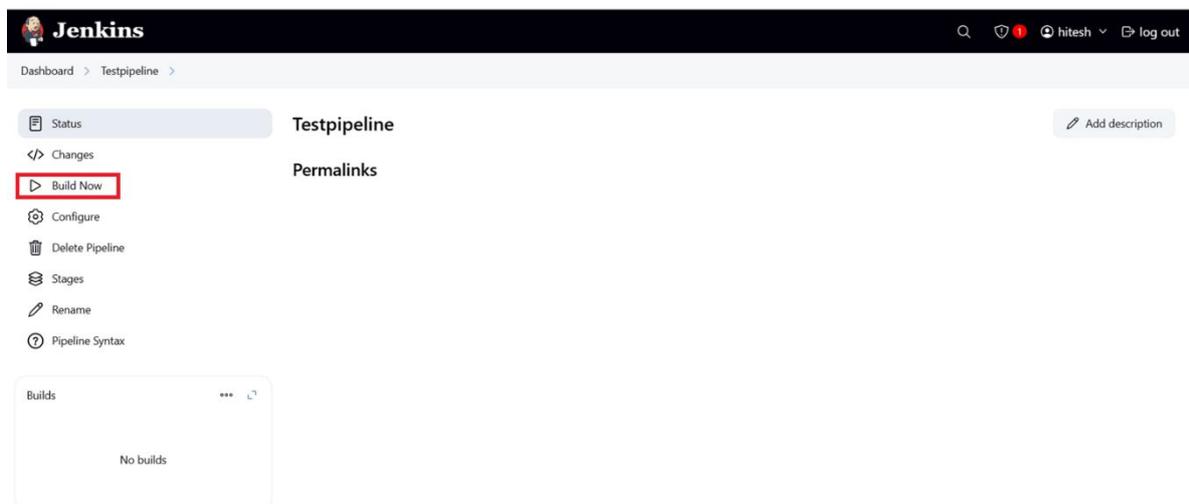
Save

Apply

Choose the “Hello World” Script Here

Step 4: Now, we will use the below-given script to check whether the pipeline script is working properly or not.

```
pipeline {
  agent none
  environment {
    DOCKERHUB_CREDENTIALS=credentials("dbefe568-9e1e-4ab3-9d89-66aeaf7edb8c")
  }
  stages {
    stage('Hello') {
      steps {
        echo 'Hello World'
      }
    }
  }
}
```



The screenshot shows the Jenkins interface for build #3. The top navigation bar includes the Jenkins logo, a search icon, a shield icon, a user profile for 'hitesh', and a 'log out' button. Below the navigation bar, the breadcrumb trail reads 'Dashboard > Testpipeline > #3'. The main content area is divided into two columns. The left column contains a sidebar with menu items: Status (selected), Changes, Console Output, Edit Build Information, Delete build '#3', Timings, Pipeline Overview, Pipeline Console, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The right column displays the build details for '#3 (Mar 30, 2025, 8:04:38 AM)'. It features a green checkmark icon, a 'Add description' button, and a 'Keep this build forever' button. The build status is 'Started by user hitesh', with a timestamp 'Started 38 sec ago' and a duration of 'Took 4.7 sec'. A 'This run spent:' section lists: '49 ms waiting', '4.7 sec build duration', and '4.7 sec total from scheduled to completion'. A 'No changes' message is shown at the bottom of the right column.

The screenshot shows the Jenkins console output for build #3. The top navigation bar is identical to the previous screenshot. The breadcrumb trail is 'Dashboard > Testpipeline > #3'. The main content area is divided into two columns. The left column contains the same sidebar as the previous screenshot. The right column displays the console output for 'Console Output'. It features a green checkmark icon, a 'Download' button, a 'Copy' button, and a 'View as plain text' button. The console output text is as follows: 'Started by user hitesh', '[Pipeline] Start of Pipeline', '[Pipeline] withCredentials', 'Masking supported pattern matches of \$DOCKERHUB_CREDENTIALS or \$DOCKERHUB_CREDENTIALS_PSW', '[Pipeline] {', '[Pipeline] stage', '[Pipeline] { (Hello)', '[Pipeline] echo', 'Hello World', '[Pipeline] }', '[Pipeline] // stage', '[Pipeline] }', '[Pipeline] // withCredentials', '[Pipeline] End of Pipeline', and 'Finished: SUCCESS'.

Build Successfully Created

Again, paste the below-given **script** in the **“Pipeline”** section.
Click on “Save”.

```
pipeline {
  agent none // No global agent, each stage should define its own agent

  environment {
    DOCKERHUB_CREDENTIALS = credentials("dbefe568-9e1e-4ab3-9d89-66aeaf7edb8c")
  }

  stages {
    stage('Hello') {
```


- Status
- Changes
- Console Output**
- Edit Build Information
- Delete build '#4'
- Timings
- Git Build Data
- Pipeline Overview
- Pipeline Console
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces
- Previous Build

Console Output

Download Copy View as plain text

```
Started by user hitesh
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] node
Running on Kubernetes-Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (git)
[Pipeline] node
Running on Kubernetes-Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
```

- Previous Build

```
Running on Kubernetes-Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/hiteshchauhan89/Devops-Project-02.git
> git init /home/ubuntu/jenkins/workspace/Testpipeline # timeout=10
Fetching upstream changes from https://github.com/hiteshchauhan89/Devops-Project-02.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/hiteshchauhan89/Devops-Project-02.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/hiteshchauhan89/Devops-Project-02.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 123df03262519d0ab9ca5c879356045a418addf3 (refs/remotes/origin/master)
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git config core.sparsecheckout # timeout=10
> git checkout -f 123df03262519d0ab9ca5c879356045a418addf3 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b master 123df03262519d0ab9ca5c879356045a418addf3 # timeout=10
Commit message: "Create Dockerfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
```

```
> git config core.sparsecheckout # timeout=10
> git checkout -f 123df03262519d0ab9ca5c879356045a418addf3 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b master 123df03262519d0ab9ca5c879356045a418addf3 # timeout=10
Commit message: "Create Dockerfile"
First time build. skipping changelog.
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] End of Pipeline
Finished: SUCCESS
```

```

Memory usage: 14%          IPv4 address for enX0: 172.31.88.162
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
37 updates can be applied immediately.
16 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 Mar 29 19:38:58 2025 from 106.222.204.95
ubuntu@ip-172-31-88-162:~$ cd /home/ubuntu/jenkins/
ubuntu@ip-172-31-88-162:~/jenkins$ ll
total 1384
drwx----- 4 ubuntu ubuntu      4096 Mar 30 08:13 ./
drwxr-x--- 8 ubuntu ubuntu      4096 Mar 30 08:13 ../
drwxrwxr-x 4 ubuntu ubuntu      4096 Mar 29 20:08 remoting/
-rw-rw-r-- 1 ubuntu ubuntu    1396936 Mar 29 20:08 remoting.jar
drwxrwxr-x 3 ubuntu ubuntu      4096 Mar 30 08:13 workspace/
ubuntu@ip-172-31-88-162:~/jenkins$ cd workspace/
ubuntu@ip-172-31-88-162:~/jenkins/workspace$ ll
total 12
drwxrwxr-x 3 ubuntu ubuntu      4096 Mar 30 08:13 ./
drwx----- 4 ubuntu ubuntu      4096 Mar 30 08:13 ../
drwxrwxr-x 4 ubuntu ubuntu      4096 Mar 30 08:13 Testpipeline/
ubuntu@ip-172-31-88-162:~/jenkins/workspace$ cd Testpipeline/
ubuntu@ip-172-31-88-162:~/jenkins/workspace/Testpipeline$ LL
LL: command not found
ubuntu@ip-172-31-88-162:~/jenkins/workspace/Testpipeline$ ll
total 24
drwxrwxr-x 4 ubuntu ubuntu      4096 Mar 30 08:13 ./
drwxrwxr-x 3 ubuntu ubuntu      4096 Mar 30 08:13 ../
drwxrwxr-x 8 ubuntu ubuntu      4096 Mar 30 08:13 .git/
-rw-rw-r-- 1 ubuntu ubuntu       41 Mar 30 08:13 Dockerfile
drwxrwxr-x 2 ubuntu ubuntu      4096 Mar 30 08:13 images/
-rw-rw-r-- 1 ubuntu ubuntu      193 Mar 30 08:13 index.html
ubuntu@ip-172-31-88-162:~/jenkins/workspace/Testpipeline$ █

```

Files Fetched from the Github Repository Automatically

```

pipeline {
  agent any
  environment {
    DOCKERHUB_CREDENTIALS = credentials('dbefe568-9e1e-4ab3-9d89-66aeaf7edb8c') // Replace with your credentials ID
    IMAGE_NAME = "hiteshtech003/project2"
  }
  stages {
    stage('Clone Repository') {
      steps {
        git 'https://github.com/hiteshchauhan89/devops-project-02.git'
      }
    }
    stage('Build Docker Image') {
      steps {
        sh 'docker build -t $IMAGE_NAME .'
      }
    }
    stage('Login to Docker Hub') {
      steps {
        sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
      }
    }
  }
}

```

```
}
}
stage('Push Docker Image') {
  steps {
    sh 'docker push $IMAGE_NAME'
  }
}
}
```

Dashboard > Testpipeline > Configuration

Configure

General

Triggers

Pipeline

Advanced

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

```
1- pipeline {
2-   agent any
3-   environment {
4-     DOCKERHUB_CREDENTIALS = credentials('dbefe568-9e1e-4ab3-9d89-66aaf7ed8bc') // Replace with your credentials ID
5-     IMAGE_NAME = "hiteshtech803/project2"
6-   }
7-   stages {
8-     stage('Clone Repository') {
9-       steps {
10-        git 'https://github.com/hiteshchauhan89/devops-project-02.git'
11-       }
12-     }
13-     stage('Build Docker Image') {
14-       steps {
15-        sh 'docker build -t $IMAGE_NAME .'
16-       }
17-     }
18-     stage('Login to Docker Hub') {
19-       steps {
20-        sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
21-       }
22-     }
23-     stage('Push Docker Image') {
24-       steps {
25-        sh 'docker push $IMAGE_NAME'
26-       }
27-     }
28-   }
29- }
30- }
```

Save

Apply

- 📄 Status
- <> Changes
- 📄 Console Output
- 📄 Edit Build Information
- 🗑️ Delete build '#4'
- 🕒 Timings
- 🔗 Git Build Data
- 🔗 Pipeline Overview
- 📄 Pipeline Console
- 🔄 Restart from Stage
- 🔄 Replay
- 📄 Pipeline Steps
- 📄 Workspaces
- ⬅️ Previous Build

Console Output

📄 Download 📄 Copy 📄 View as plain text

```
Started by user hitesh
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSM
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone Repository)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Testpipeline/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/hiteshchauhan89/devops-project-02.git # timeout=10
Fetching upstream changes from https://github.com/hiteshchauhan89/devops-project-02.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
```

⬅️ Previous Build

```
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/hiteshchauhan89/devops-project-02.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 123df03262519d0ab9ca5c879356045a418addf3 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 123df03262519d0ab9ca5c879356045a418addf3 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D master # timeout=10
> git checkout -b master 123df03262519d0ab9ca5c879356045a418addf3 # timeout=10
Commit message: "Create Dockerfile"
> git rev-list --no-walk 123df03262519d0ab9ca5c879356045a418addf3 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build Docker Image)
[Pipeline] sh
+ docker build -t hiteshtech003/project2 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 270.8kB

Step 1/2 : FROM ubuntu/apache2
latest: Pulling from ubuntu/apache2
2947c6b407f0: Pulling fs layer
2947c6b407f0: Pulling fs layer
02a81a111acb: Pulling fs layer
e9b5d2d6295b: Pulling fs layer
e9b5d2d6295b: Verifying Checksum
e9b5d2d6295b: Download complete
2947c6b407f0: Verifying Checksum
2947c6b407f0: Download complete
02a81a111acb: Verifying Checksum
02a81a111acb: Download complete
2947c6b407f0: Pull complete
02a81a111acb: Pull complete
e9b5d2d6295b: Pull complete
Digest: sha256:4f4289d63ca00ede07f8b878dc018e952f282815dfa7f928ab13184099080255
Status: Downloaded newer image for ubuntu/apache2:latest
--> bb0300c46637
Step 2/2 : COPY ./var/www/html
--> 223a23a13dba
Successfully built 223a23a13dba
Successfully tagged hiteshtech003/project2:latest
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
```

```
https://mads.madsen.com/gist/11111111/
Sending build context to Docker daemon 270.8kB

Step 1/2 : FROM ubuntu/apache2
latest: Pulling from ubuntu/apache2
2947c6b407f0: Pulling fs layer
2947c6b407f0: Pulling fs layer
02a81a111acb: Pulling fs layer
e9b5d2d6295b: Pulling fs layer
e9b5d2d6295b: Verifying Checksum
e9b5d2d6295b: Download complete
2947c6b407f0: Verifying Checksum
2947c6b407f0: Download complete
02a81a111acb: Verifying Checksum
02a81a111acb: Download complete
2947c6b407f0: Pull complete
02a81a111acb: Pull complete
e9b5d2d6295b: Pull complete
Digest: sha256:4f4289d63ca00ede07f8b878dc018e952f282815dfa7f928ab13184099080255
Status: Downloaded newer image for ubuntu/apache2:latest
--> bb0300c46637
Step 2/2 : COPY ./var/www/html
--> 223a23a13dba
Successfully built 223a23a13dba
Successfully tagged hiteshtech003/project2:latest
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
```

```
Dashboard > Testpipeline > #4

[Pipeline] stage
[Pipeline] { (Login to Docker Hub)
[Pipeline] sh
+ echo *****
+ docker login -u hiteshtech003 --password-stdin
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Push Docker Image)
[Pipeline] sh
+ docker push hiteshtech003/project2
Using default tag: latest
The push refers to repository [docker.io/hiteshtech003/project2]
6cfd90c99b3: Preparing
488aba8b8885: Preparing
bcc05c4e26b1: Preparing
9c8b3e5cca67: Preparing
bcc05c4e26b1: Mounted from ubuntu/apache2
9c8b3e5cca67: Mounted from ubuntu/apache2
488aba8b8885: Mounted from ubuntu/apache2
6cfd90c99b3: Pushed
latest: digest: sha256:272a95eed3c969da6202ed3f60acc3be3e735cd3a49cb8c1a0be8f77b701211 size: 1158
[Pipeline] }
```

```
Dashboard > Testpipeline > #4

+ docker push hiteshtech003/project2
Using default tag: latest
The push refers to repository [docker.io/hiteshtech003/project2]
6cfd90c99b3: Preparing
488aba8b8885: Preparing
bcc05c4e26b1: Preparing
9c8b3e5cca67: Preparing
bcc05c4e26b1: Mounted from ubuntu/apache2
9c8b3e5cca67: Mounted from ubuntu/apache2
488aba8b8885: Mounted from ubuntu/apache2
6cfd90c99b3: Pushed
latest: digest: sha256:272a95eed3c969da6202ed3f60acc3be3e735cd3a49cb8c1a0be8f77b701211 size: 1158
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Step 12: Login into your DockerHub Account & you will notice that “hiteshtech003/project2” will be successfully pushed to the DockerHub account.

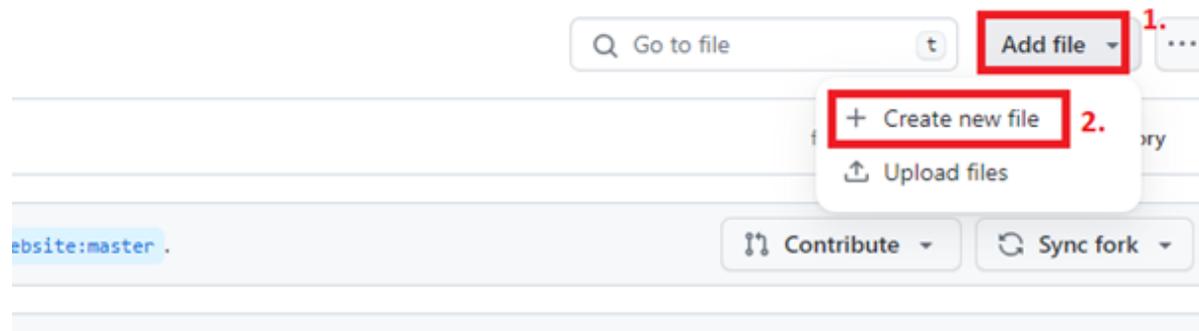
The screenshot shows the Docker Hub interface for the repository 'hiteshtech003/project2'. The repository name is highlighted in a red box. Below it, the 'Tags' section is also highlighted in a red box and contains the following table:

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	6 minutes

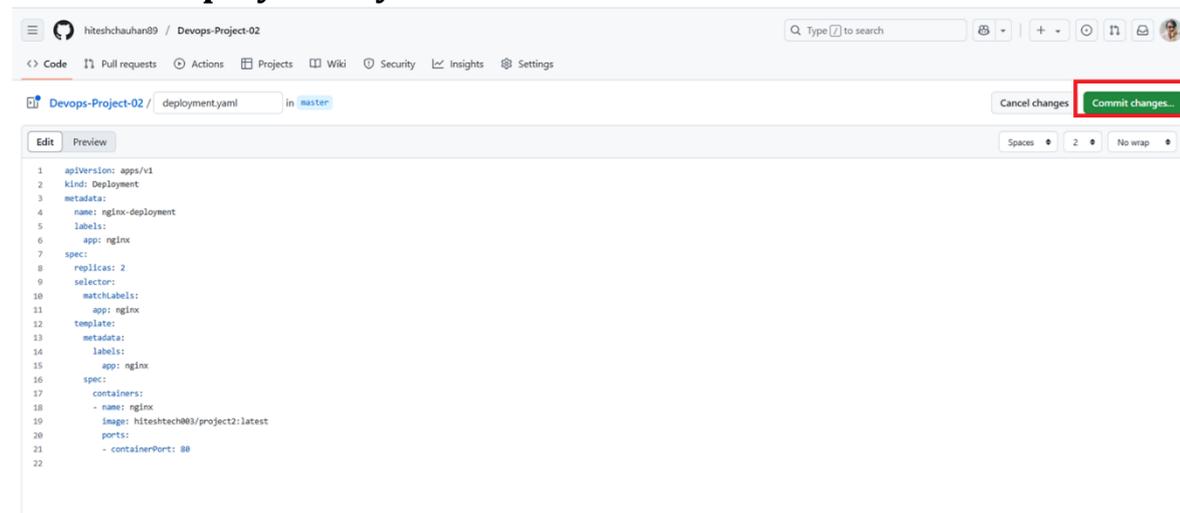
To the right of the 'Tags' table, the 'Docker commands' section is highlighted in a red box and shows the command: `docker push hiteshtech003/project2:tagname`. The interface also shows a sidebar with navigation options like 'Repositories', 'Settings', and 'Usage', and a top navigation bar with 'My Hub' and a search bar.

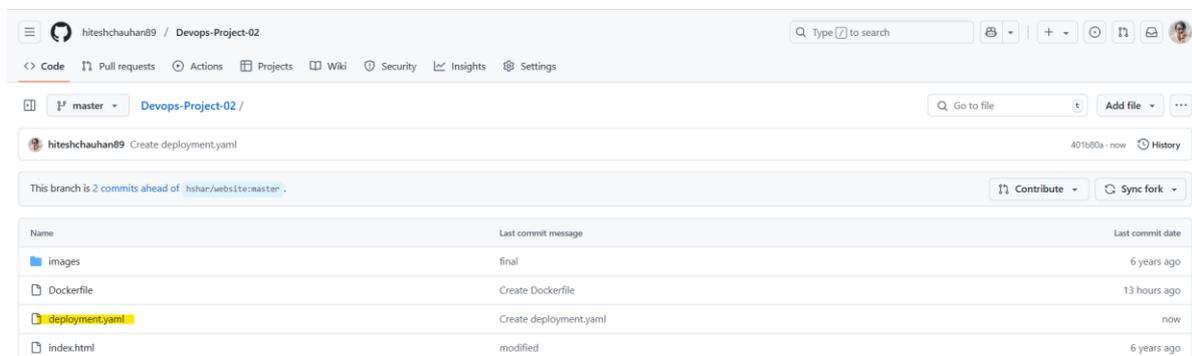
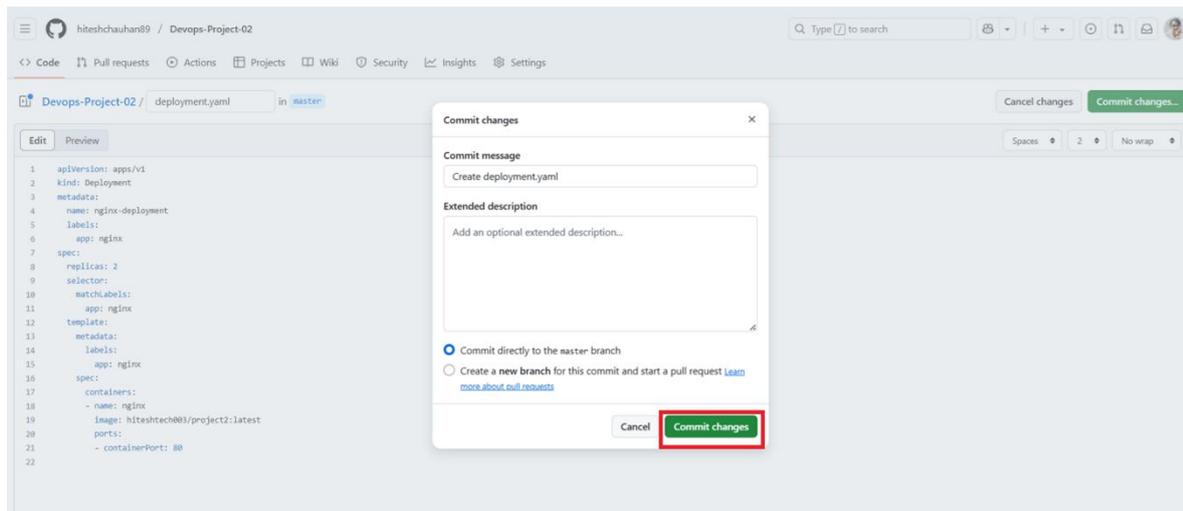
Step 13: Now, we will create the “deployment.yaml” & “service.yaml” file to deploy the website using the “Kubernetes” tool. Go to the “GitHub” account & click on “Add file”.

Click on “Create new file”.



Create a deployment.yaml file





Put the name as “deployment.yaml” & click on the “Commit changes” .

Create a “service.yaml” file for deploying the website over node port 30008.

apiVersion: v1

kind: Service

metadata:

name: my-service

spec:

type: NodePort

selector:

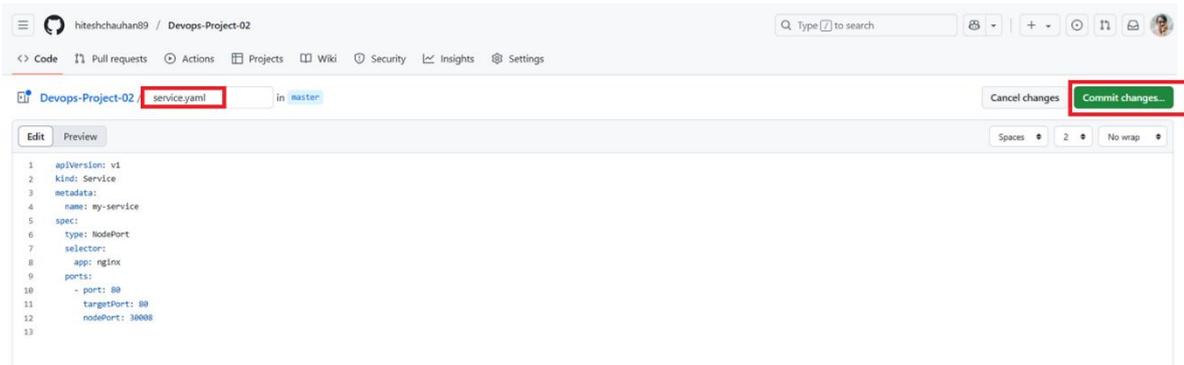
app: nginx

ports:

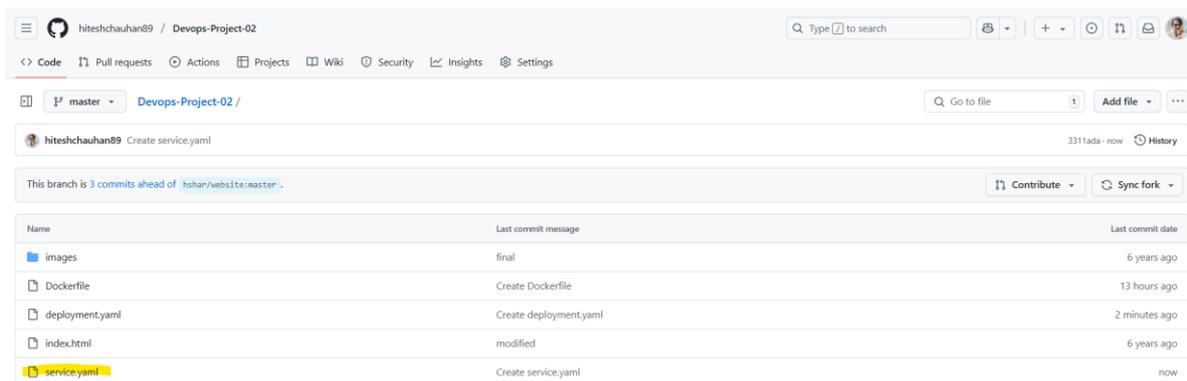
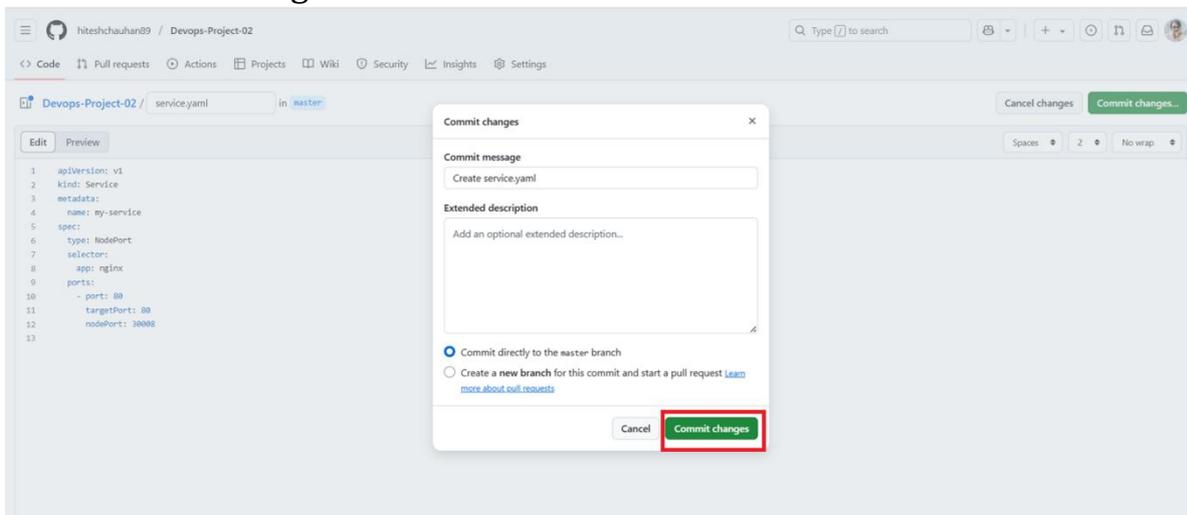
- port: 80

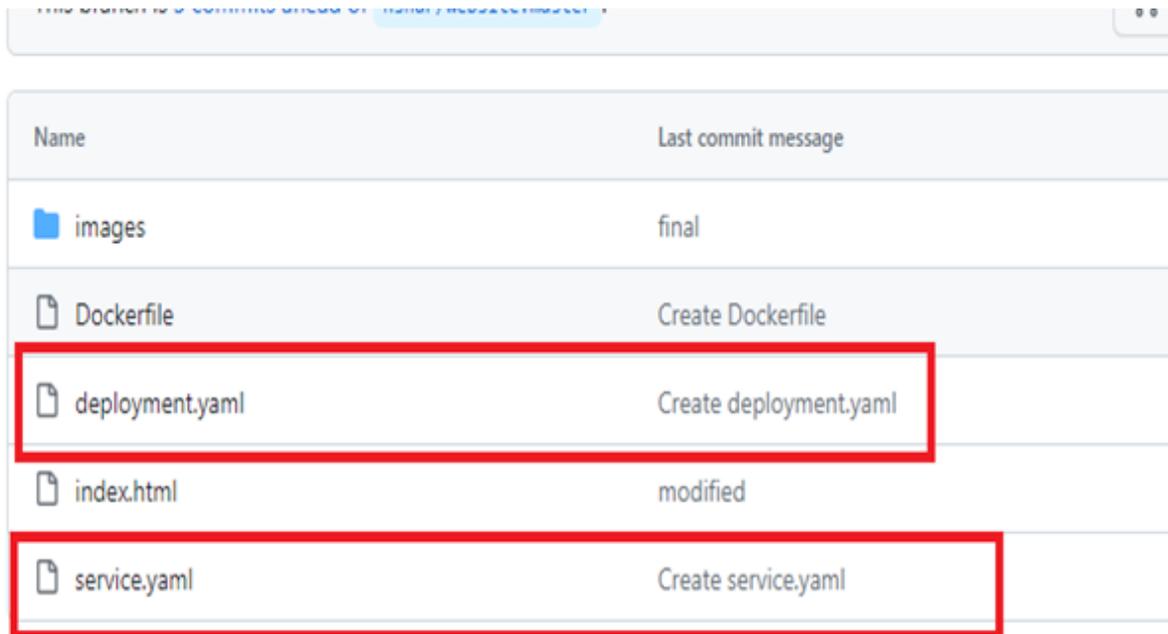
targetPort: 80

nodePort: 30008



Commit The Changes





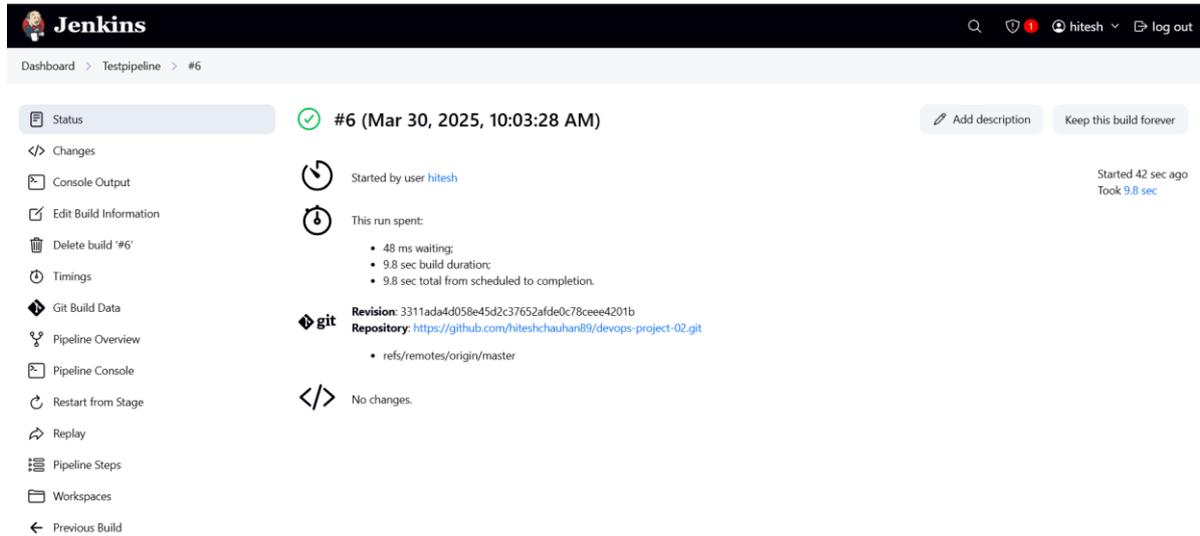
Name	Last commit message
images	final
Dockerfile	Create Dockerfile
deployment.yaml	Create deployment.yaml
index.html	modified
service.yaml	Create service.yaml

The deployment.yaml & service.yaml files in the GitHub Repository
Step 18: Go to the “Configure” in the “Jenkins” & paste the below-given code to create the Kubernetes Deployment for deploying the application:

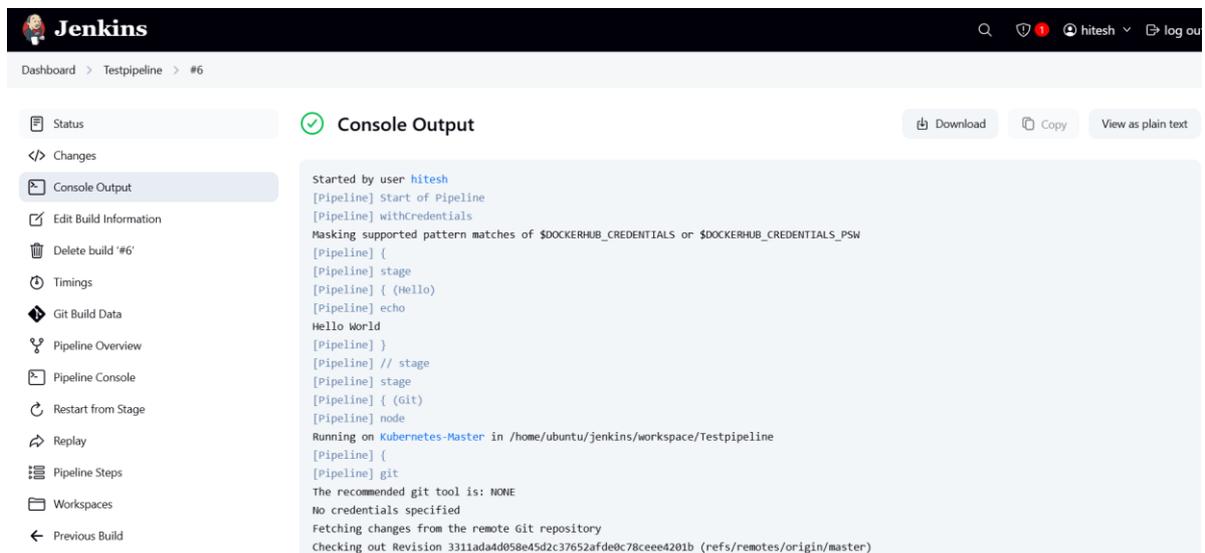
```
pipeline {
  agent none
  environment {
    DOCKERHUB_CREDENTIALS=credentials('dbefe568-9e1e-4ab3-9d89-66aeaf7edb8c')
  }
  stages {
    stage('Hello') {
      steps {
        echo 'Hello World'
      }
    }
    stage('Git') {
      agent {
        label 'Kubernetes-Master'
      }
      steps {
        git 'https://github.com/hiteshchauhan89/devops-project-02.gi
```


Save the “Code” of Kubernetes Deployment Creation

Step 19: Again, click on “Build Now”. Now, the build will be successfully created & the website has been successfully deployed over Slaves through Kubernetes Architecture.



The screenshot shows the Jenkins interface for build #6. The top navigation bar includes the Jenkins logo, search, and user information (hitesh). The breadcrumb trail is Dashboard > Testpipeline > #6. The main content area features a left sidebar with navigation options: Status, Changes, Console Output, Edit Build Information, Delete build '#6', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main panel displays the build status as a green checkmark and the title "#6 (Mar 30, 2025, 10:03:28 AM)". It includes buttons for "Add description" and "Keep this build forever". The build details show it was started by user hitesh, took 9.8 seconds, and lists the following metrics: 48 ms waiting, 9.8 sec build duration, and 9.8 sec total from scheduled to completion. The git information shows Revision: 3311ada4d058e45d2c37652afde0c78ceee4201b and Repository: https://github.com/hiteshchauhan89/devops-project-02.git. The changes section indicates "No changes."



The screenshot shows the Jenkins Console Output for build #6. The top navigation bar is identical to the previous screenshot. The breadcrumb trail is Dashboard > Testpipeline > #6. The main content area features a left sidebar with navigation options: Status, Changes, Console Output, Edit Build Information, Delete build '#6', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main panel displays the console output with a green checkmark and the title "Console Output". It includes buttons for "Download", "Copy", and "View as plain text". The console output text is as follows:
Started by user hitesh
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of \$DOCKERHUB_CREDENTIALS or \$DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (git)
[Pipeline] node
Running on Kubernetes-Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote git repository
Checking out Revision 3311ada4d058e45d2c37652afde0c78ceee4201b (refs/remotes/origin/master)

Dashboard > Testpipeline > #6

```
Commit message: "Create service.yaml"
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/Testpipeline/.git # timeout=10
> git config remote.origin.url https://github.com/hiteshchauhan89/devops-project-02.git # timeout=10
Fetching upstream changes from https://github.com/hiteshchauhan89/devops-project-02.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/hiteshchauhan89/devops-project-02.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git config core.sparsecheckout # timeout=10
> git checkout -f 3311ada4d058e45d2c37652afde0c78ceee4201b # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D master # timeout=10
> git checkout -b master 3311ada4d058e45d2c37652afde0c78ceee4201b # timeout=10
> git rev-list --no-walk 3311ada4d058e45d2c37652afde0c78ceee4201b # timeout=10
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Docker)
[Pipeline] node
Running on Kubernetes-Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] sh
+ sudo docker build /home/ubuntu/jenkins/workspace/Testpipeline -t hiteshtech003/project2
#0 building with "default" instance using docker driver
```

Dashboard > Testpipeline > #6

```
#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 788 done
#1 DONE 0.0s

#2 [auth] ubuntu/apache2:pull token for registry-1.docker.io
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/ubuntu/apache2:latest
#3 DONE 0.2s

#4 [internal] load .dockerignore
#4 transferring context: 2B done
#4 DONE 0.0s

#5 [internal] load build context
#5 transferring context: 10.30kB done
#5 DONE 0.0s

#6 [1/2] FROM docker.io/ubuntu/apache2:latest@sha256:4f4289d63ca00ede07f8b878dc018e952f282815d4fa7f928ab13184099080255
#6 CACHED

#7 [2/2] COPY . /var/www/html
#7 DONE 0.1s

#8 exporting image
#8 exporting layers 0.0s done
#8 writing image sha256:8593bb7c52230608d8346791b4a730fce8e3f46a3104b261960311f997f24044 done
#8 naming to docker.io/hiteshtech003/project2 done
```

Dashboard > Testpipeline > #6

```
[Pipeline] sh (hide)
+ + sudo docker login -u hiteshtech003 --password-stdin
sudo echo ****
Login Succeeded
[Pipeline] sh
+ sudo docker push hiteshtech003/project2
Using default tag: latest
The push refers to repository [docker.io/hiteshtech003/project2]
030db2156908: Preparing
488aba8b8885: Preparing
bcc05c4e26b1: Preparing
9c8b3e5cca67: Preparing
9c8b3e5cca67: Layer already exists
bcc05c4e26b1: Layer already exists
488aba8b8885: Layer already exists
030db2156908: Pushed
latest: digest: sha256:3b9138b220d2b34c74d1bb5685eb9df3e3e180b3ce5e4c9d3fe14685b6f5874a size: 1158
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (K8s)
[Pipeline] node
Running on Kubernetes-Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] sh
```

Dashboard > Testpipeline > #6

```
[Pipeline] }  
[Pipeline] // node  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (K8s)  
[Pipeline] node  
Running on Kubernetes-Master in /home/ubuntu/jenkins/workspace/Testpipeline  
[Pipeline] {  
[Pipeline] sh  
+ kubectl apply -f deployment.yaml  
deployment.apps/nginx-deployment unchanged  
[Pipeline] sh  
+ kubectl apply -f service.yaml  
service/my-service unchanged  
[Pipeline] }  
[Pipeline] // node  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] }  
[Pipeline] // withCredentials  
[Pipeline] End of Pipeline  
Finished: SUCCESS
```

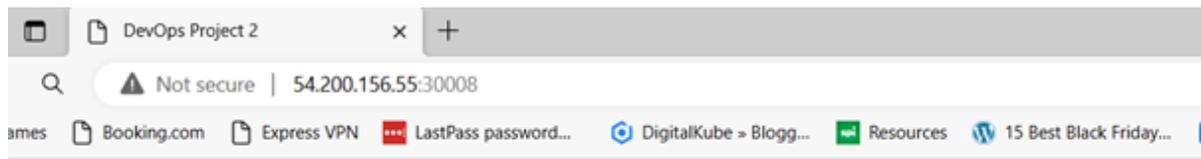
Kubernetes Master Server

Hello world!



GitHub

Kubernetes-Worker 1



Hello world!



Kubernetes Worker 2

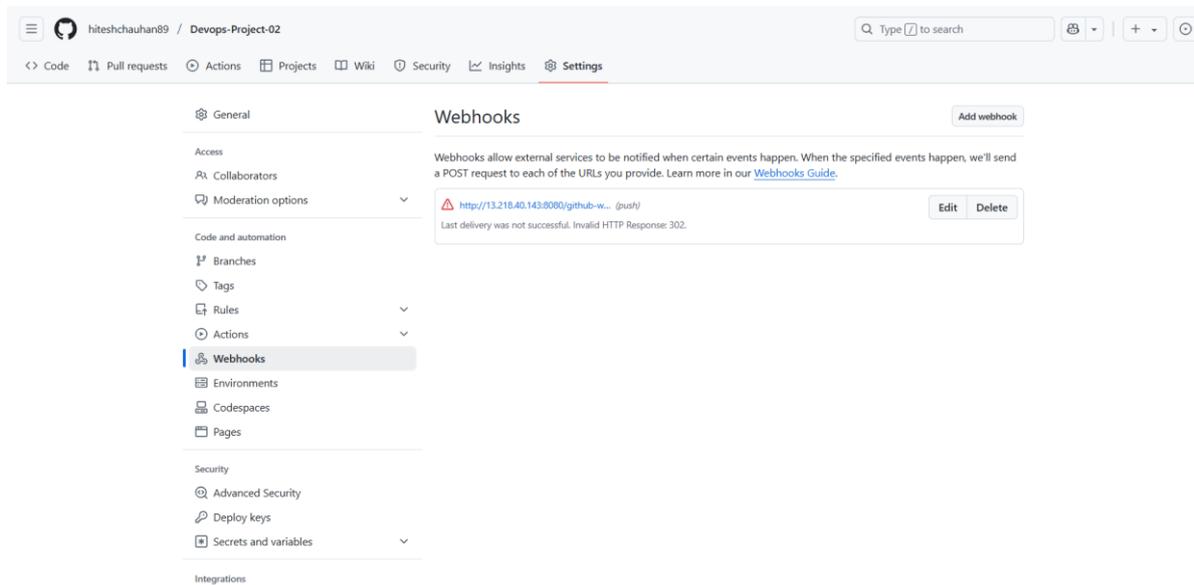


Hello world!

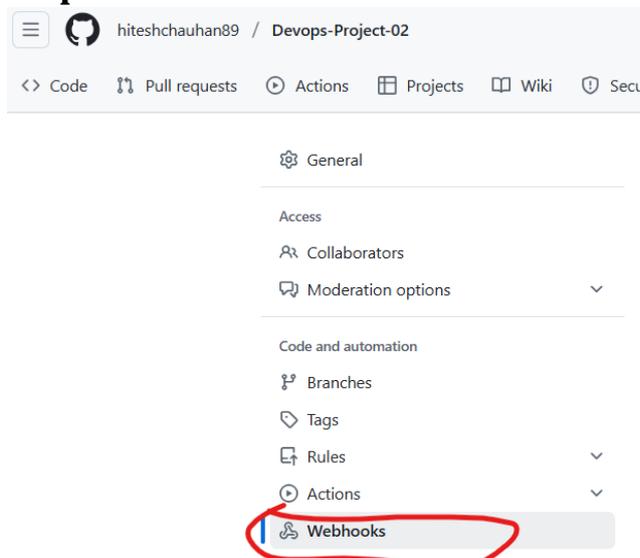


Automate the Pipeline using Github Webhooks.

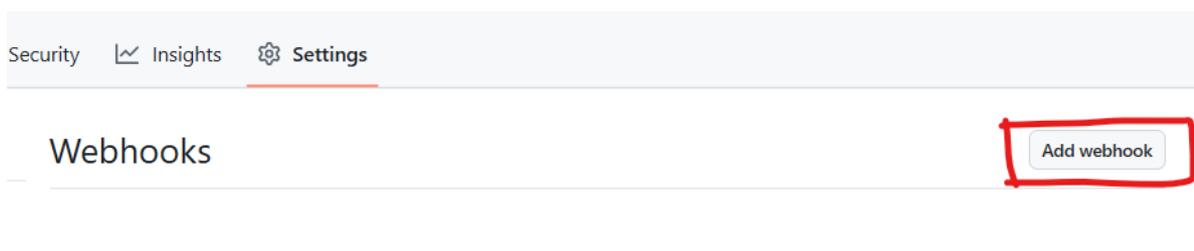
Step 1: Go to the “GitHub Repository (website)” & click on the “Settings”.

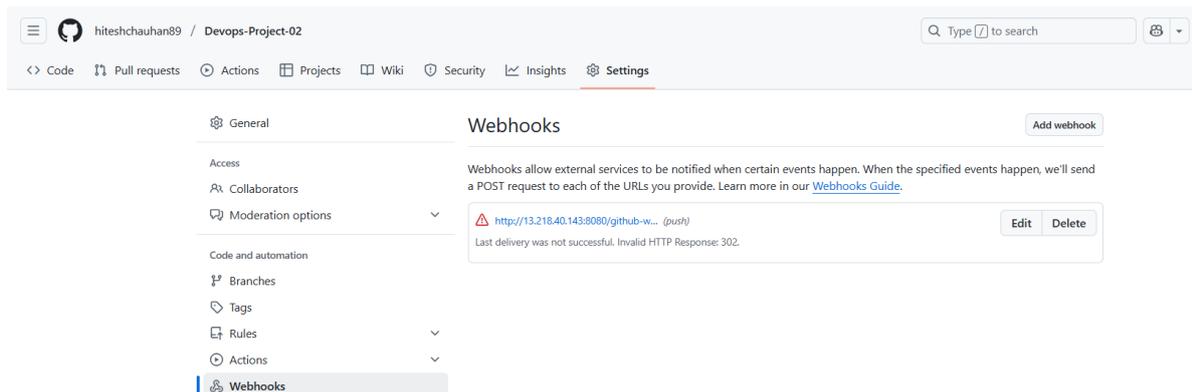


Step 2: Click on the “Webhooks”.



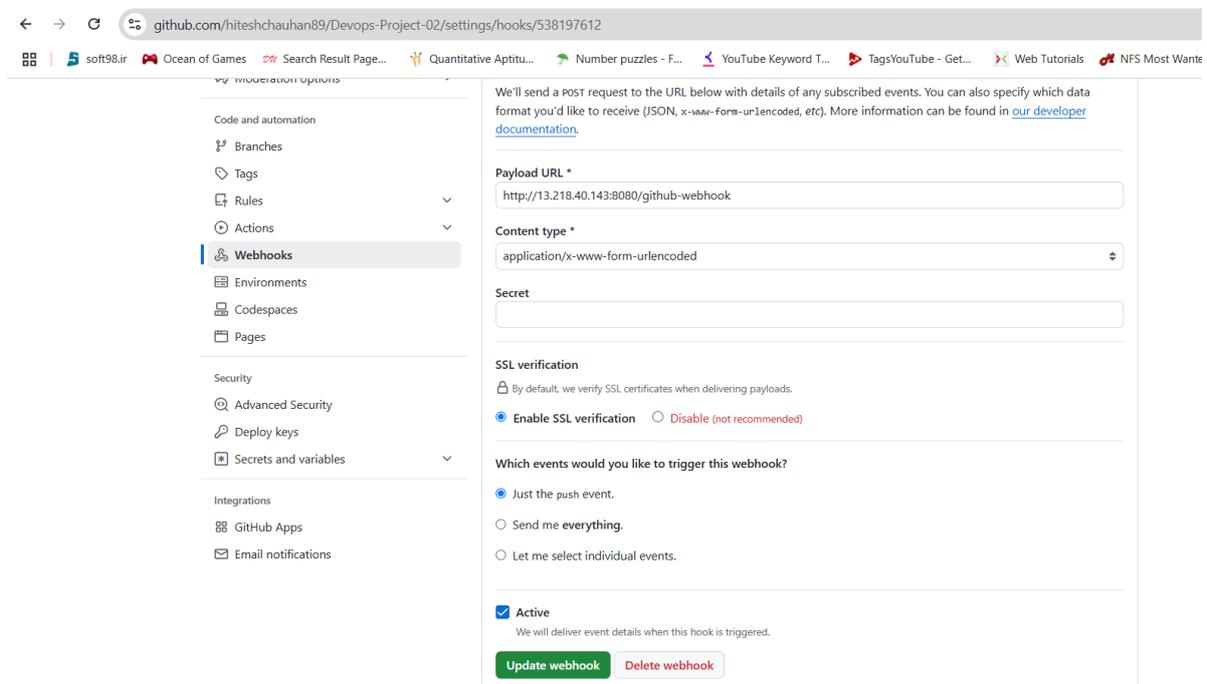
Go to the “Webhooks” section Step 3: Click on “Add webhook”.





Add the Webhook

Step 4: Choose “Payload URL” as <http://13.218.40.143:8080/github-webhook>



Then click Update Webhook.

Step 5: The Webhook has been successfully created.

Step 6: Go to the “Testpipeline” & choose “GitHub hook trigger for GITScm polling” in “Build Triggers”.

Configure

General

Triggers

Pipeline

Advanced

 Preserve stashes from completed builds ? This project is parameterized ? Throttle builds ?

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

 Build after other projects are built ? Build periodically ? GitHub hook trigger for GITScm polling ? Poll SCM ? Trigger builds remotely (e.g., from scripts) ?

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline code

Save

Apply

Choose the “GitHub hook trigger for GITScm polling” & Save the Pipeline
Click on “Save”.

16. Do the Changes & Test the Pipeline

Step 1: Go to the “Configure” & add this line to “pipeline code”.

```
sh 'kubectl delete deploy nginx-deployment'
```

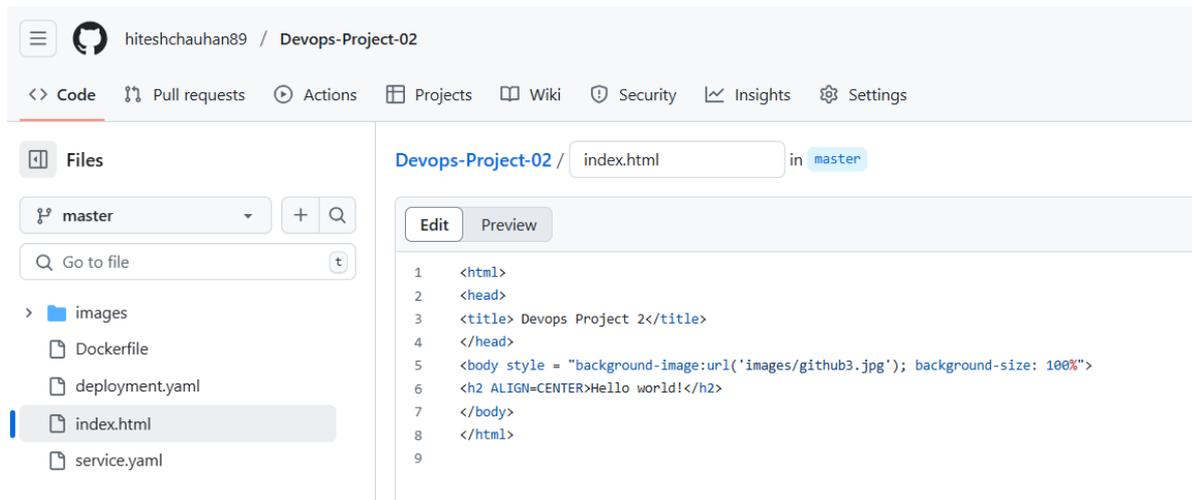
Save the **pipeline**.

Script ?

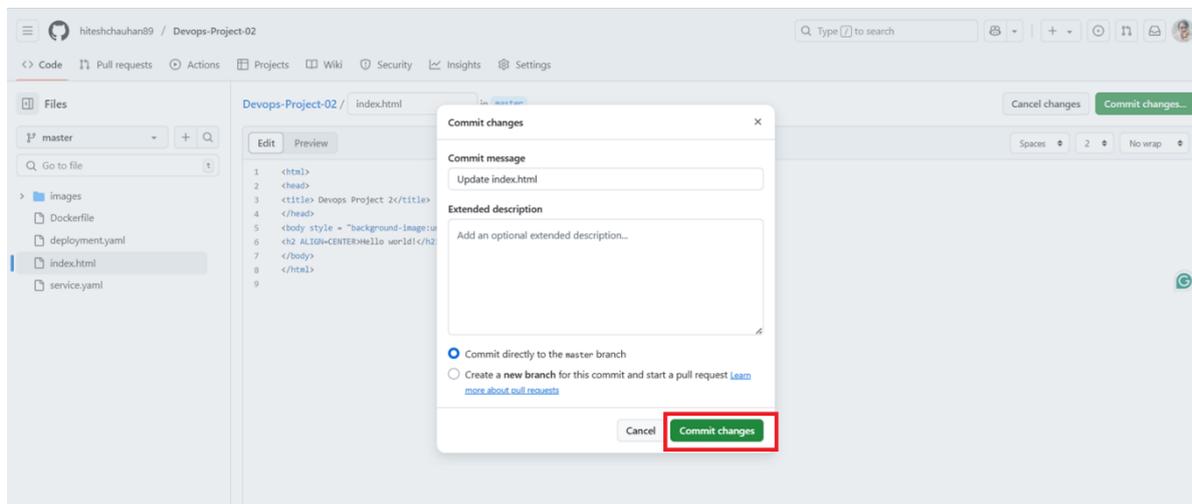
```
22     label 'kubernetes-master'
23   }
24   steps {
25     sh 'sudo docker build /home/ubuntu/jenkins/workspace/Testpipeline -t hiteshtech003/project2'
26     sh 'sudo echo $DOCKERHUB_CREDENTIALS_PSW | sudo docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
27     sh 'sudo docker push hiteshtech003/project2'
28   }
29 }
30 stage('K8s') {
31   agent {
32     label 'Kubernetes-Master'
33   }
34   steps {
35     sh 'kubectl delete deploy nginx-deployment'
36     sh 'kubectl apply -f deployment.yaml'
37     sh 'kubectl apply -f service.yaml'
38   }
39 }
40 }
41 }
```

Add the delete deployment command

Step 2: Now, go to the “index.html” file in the “GitHub Repository”.



Change the index.html file Title
Change “Title” from “intellipaat” to “DevOps Project 2”.



Write the New Title for index.html file
Click on “Commit Changes”.
Commit the index.html file
Step 3: Again, click on the “Commit Changes”.
Step 4: An automatic build will be created successfully. Click on “#13”.



- Status
- Changes
- Build Now
- Configure
- Delete Pipeline
- Stages
- Rename
- Pipeline Syntax
- GitHub Hook Log

Testpipeline

Permalinks

- Last build (#13), 54 min ago
- Last stable build (#13), 54 min ago
- Last successful build (#13), 54 min ago
- Last failed build (#3), 2 hr 57 min ago
- Last unsuccessful build (#3), 2 hr 57 min ago
- Last completed build (#13), 54 min ago

Builds

Filter

Today

#13 11:11 AM

Click on #13 Build

Step 5: Click on the “Console Output”. The pipeline will be shown to you.



- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#13'
- Timings
- Git Build Data
- Pipeline Overview
- Pipeline Console
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces
- Previous Build

Console Output

Download

```
Started by user hitesh
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Git)
[Pipeline] node
Running on Kubernetes-Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/Testpipeline/.git # timeout=10
Fetching changes from the remote Git repository
```

```
Dashboard > Testpipeline > #13
[Pipeline] stage
[Pipeline] { (K8s)
[Pipeline] node
Running on Kubernetes-Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] sh
+ kubectl delete deploy nginx-deployment
deployment.apps "nginx-deployment" deleted
[Pipeline] sh
+ kubectl apply -f deployment.yaml
deployment.apps/nginx-deployment created
[Pipeline] sh
+ kubectl apply -f service.yaml
service/my-service unchanged
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] End of Pipeline
Finished: SUCCESS
```

New Deployment Created Successfully

Step 6: Go to the “Browser” & again refresh the IP Address of “Machine-2” & “Machine-4”. You will notice that in “title”, now “DevOps Project 2” is showing



Hello world!

