1. Create EC2 in AWS i.e. Master and slave
2. Connect the EC2 Instance using SSH Command
3. Install Ansible on Bastion Host

sudo apt update

sudo apt install ansible -y

1. We need to create directory structure for Ansible codebase

mkdir -p ansible-project/{inventory,playbooks,roles}

1. Now we must go to the folder

cd ansible-project

1. Now create ansible.cfg to configure Ansible

[defaults]

inventory = inventory

host\_key\_checking = False

1. Now we need to create inventory file for Jenkins and app

[jenkins]

44.211.201.97 ansible\_ssh\_user=ubuntu ansible\_ssh\_private\_key\_file="/home/ubuntu/task2TestInstance.pem"

[app]

44.201.76.3 ansible\_ssh\_user=ubuntu ansible\_ssh\_private\_key\_file="/home/ubuntu/task2TestInstance.pem"

8) Pinging the inventory

ansible all -m ping -i inventory

A screenshot of a computer

AI-generated content may be incorrect.

1. Now Create a directory File playbooks
2. Under playbooks directory create instal-docker.yaml
3. Now run the playbook

A screenshot of a computer screen

AI-generated content may be incorrect.

1. SSH into Jenkins instance

ssh -i /home/ubuntu/task2TestInstance.pem [ubuntu@44.211.201.97](mailto:ubuntu@44.211.201.97)

1. Install Jenkins manually

A screenshot of a computer

AI-generated content may be incorrect.

1. Find the password from the sudo cat /var/lib/jenkins/secrets/initialAdminPassword
2. Jenekins URL in Bastion

A screenshot of a computer

AI-generated content may be incorrect.

Now creating Load Balancer for Jenkins we can either do manually though using Terraform.

First Create IAM Role in AWS where we have to create Administrator Access for EC2 machine

A screenshot of a computer

AI-generated content may be incorrect.

Now we have to install Terraform

Create main.tf file

provider "aws" {

region = "us-east-1"

}

resource "aws\_lb" "jenkins\_alb" {

name = "jenkins-alb"

internal = false

load\_balancer\_type = "application"

security\_groups = ["sg-0ce0d725e38b35c47"]

subnets = ["subnet-073ebc5fb8366f5d3", "subnet-073ebc5fb8366f5d3"]

enable\_deletion\_protection = false

enable\_cross\_zone\_load\_balancing = true

access\_logs {

bucket = "my-log-bucket"

}

}

resource "aws\_lb\_target\_group" "jenkins\_target\_group" {

name = "jenkins-target-group"

port = 8080

protocol = "HTTP"

vpc\_id = "vpc-00d38621b687fad95"

}

resource "aws\_lb\_listener" "http" {

load\_balancer\_arn = aws\_lb.jenkins\_alb.arn

port = 80

protocol = "HTTP"

default\_action {

type = "fixed-response"

fixed\_response {

status\_code = 200

message\_body = "Jenkins is running"

}

}

action {

type = "forward"

target\_group\_arn = aws\_lb\_target\_group.jenkins\_target\_group.arn

}

}

resource "aws\_lb\_target\_group\_attachment" "jenkins\_target" {

target\_group\_arn = aws\_lb\_target\_group.jenkins\_target\_group.arn

target\_id = "54.196.147.164"

port = 8080

}

After that running the terraform init

And terraform apply

Now we have to create ECR Repository

resource "aws\_ecr\_repository" "node\_app" {

name = "node-app-repository"

}

Terraform apply

Now we can attach IAM role to ECR repository i.e. Full administrator Access.