**Create ci pipeline that will deploy and configure these VMs using terraform and ansible**

🡪 **according following requirements**

* Deploy 2 virtual machines using terraform.

first vm on Amazon linux, hostname: c8.local

second vm on ubuntu 21.04, hostname: u21.local

* 2. As a result of terraform execution, dynamically create inventory for ansible

c8.local should be in the frontend group

u21.local should be in the backend group

* 3. Create ansible playbook for c8.local and u21.local

for linux OS playbook should apply the following changes

selinux: disable

firewalld: disable

* 4.for frontend playbook group should install and configure nginx

nginx configuration should do proxying from port 80 on port 19999 to the

backend group

* 5.for the backend group, the playbook must install the Netdata application from the

official repositories and run it on port 19999.

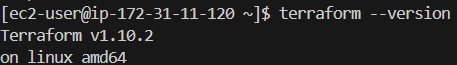
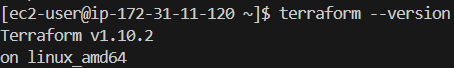
🡪create a ansible server on aws 

🡪login to server through ssh and install ansible through visual studio 

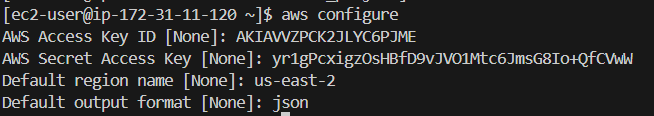
🡪Install Ansible and Terraform on Your EC2 Instance

|  |
| --- |
| sudo yum update -y  sudo amazon-linux-extras install ansible2 -y  ansible --version  sudo yum install -y yum-utils shadow-utils  sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo  sudo yum -y install terraform |

🡪**check for version**

** **

🡪download aws cli and configure it

****

🡪 **Generate an SSH Key Pair**

We'll generate a new SSH key pair to use for your Terraform-managed instances.

# ssh-keygen -t rsa -b 2048 -f ~/.ssh/pipeline\_key -N ""

🡪create a terraform directory and main.tf file

|  |
| --- |
| mkdir terraform\_project  cd terraform\_project  vi main.tf |

**🡪add the following template which will create two instance one with linux image and one with ubuntu image and give the public ip as output**

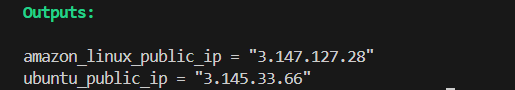
|  |
| --- |
| provider "aws" {  region = "us-east-2"  }  resource "aws\_key\_pair" "pipeline" {  key\_name = "pipeline"  public\_key = file("~/.ssh/pipeline\_key.pub")  }  resource "aws\_instance" "c8\_linux" {  ami = "ami-088d38b423bff245f"  instance\_type = "t2.micro"  key\_name = aws\_key\_pair.pipeline.key\_name  associate\_public\_ip\_address = true  tags = {  Name = "c8\_linux"  }  user\_data = <<-EOF  #!/bin/bash  hostnamectl set-hostname c8.local  EOF  }  resource "aws\_instance" "u21\_ubuntu" {  ami = "ami-00eb69d236edcfaf8"  instance\_type = "t2.micro"  key\_name = aws\_key\_pair.pipeline.key\_name  associate\_public\_ip\_address = true  tags = {  Name = "u21\_ubuntu"  }  user\_data = <<-EOF  #!/bin/bash  hostnamectl set-hostname u21.local  EOF  }  output "amazon\_linux\_public\_ip" {  value = aws\_instance.c8\_linux.public\_ip  }  output "ubuntu\_public\_ip" {  value = aws\_instance.u21\_ubuntu.public\_ip  } |

**🡪**run the following commands

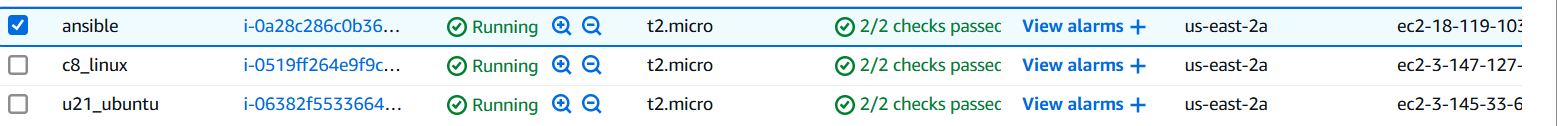
**# terraform init**

**# terraform plan**

**# terraform apply**

****

**🡪checking on aws console if instance is ok!**

****

**🡪** now get the output which are public ip’s of the created server and add to ansible inventory

# terraform output <**copy the ip’s>**

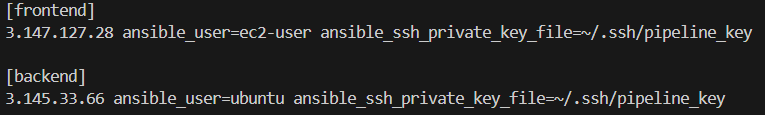
# cd /etc

# sudo chown -R ec2-user:ec2-user ansible

# vi /etc/ansible/hosts

|  |
| --- |
| [frontend]  Public\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=~/.ssh/pipeline\_key  [backend]  Public\_ip ansible\_user=ubuntu ansible\_ssh\_private\_key\_file=~/.ssh/pipeline\_key |

# cat /etc/ansible/hosts



🡪**Ensure SELinux and Firewalld are Disabled**

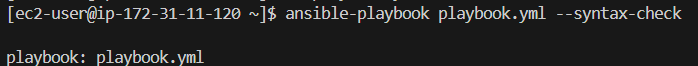
🡪make sure connectivity is established



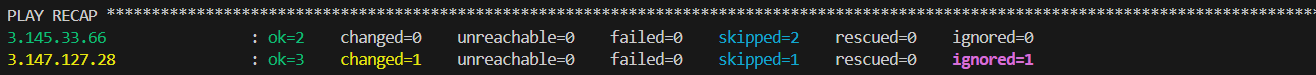
# vi playbook.yml

|  |
| --- |
| ---  - name: Configure Amazon Linux and Ubuntu servers  hosts: all  become: yes  tasks:  - name: Disable SELinux on Amazon Linux  when: ansible\_distribution == "Amazon"  command: setenforce 0  ignore\_errors: yes  - name: Disable firewalld on Amazon Linux  when: ansible\_distribution == "Amazon"  service:  name: firewalld  state: stopped  enabled: no  - name: Disable firewalld on Ubuntu  when: ansible\_distribution == "Ubuntu"  service:  name: ufw  state: stopped  enabled: no |

**#** ansible-playbook playbook.yml –syntax-check



# ansible-playbook playbook.yml



🡪**Install and Configure Nginx on the Frontend (Amazon Linux)**

🡪create template that ensures Nginx proxies requests from port 80 to port 19999 on the backend server.

# mkdir template

# cd template

# vi nginx.conf.j2

|  |
| --- |
| worker\_processes 1;  events {  worker\_connections 1024;  }  http {  upstream backend\_group {  {% for host in groups['backend'] %}  server {{ host }}:19999;  {% endfor %}  }  server {  listen 19999;  location / {  proxy\_pass http://backend\_group;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  }  }  } |

🡪create a playbook to install nginx and proxy its port

# vi nginx.yml

|  |
| --- |
| ---  - name: Configure frontend servers (Amazon Linux)  hosts: frontend  become: yes  tasks:  - name: Add the official Nginx repository  get\_url:  url: http://nginx.org/packages/centos/7/x86\_64/RPMS/nginx-1.22.1-1.el7.ngx.x86\_64.rpm  dest: /opt/nginx.rpm  - name: Install Nginx from the downloaded RPM  yum:  name: /opt/nginx.rpm  state: present  - name: Configure nginx proxy  template:  src: templates/nginx.conf.j2  dest: /etc/nginx/nginx.conf  - name: Start and enable nginx  service:  name: nginx  state: started  enabled: yes |

# ansible-playbook nginx.yml

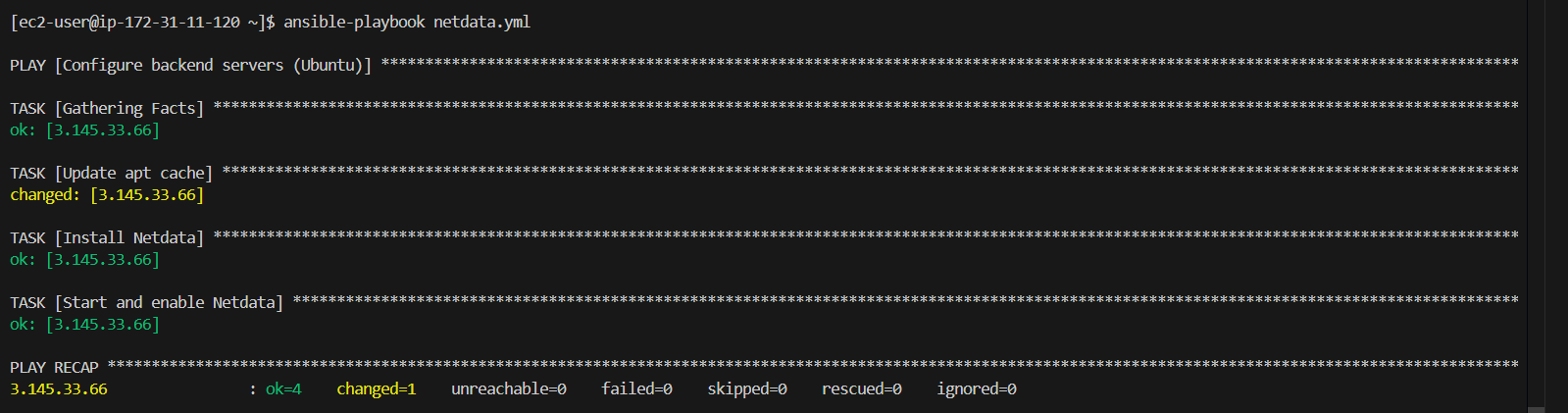


🡪**Install and Run Netdata on the Backend (Ubuntu)**

# vi netdata.yml

|  |
| --- |
| ---  - name: Configure backend servers (Ubuntu)  hosts: backend  become: yes  tasks:  - name: Update apt cache  apt:  update\_cache: yes  - name: Install Netdata  apt:  name: netdata  state: present  - name: Start and enable Netdata  service:  name: netdata  state: started  enabled: yes |

# ansible-playbook netdata.yml



\***Amazon Linux (Frontend Server)**: Nginx is installed, configured, and running to proxy requests from port 80 to port 19999 on the backend server.

**Ubuntu (Backend Server)**: Netdata is installed and running on port 19999.\*

**-------------------------------setup-completed---------------------------------**

**<INSTANCE RESTARTED SO THE PUBLIC IP GOT CHANGED>**

**🡪checking on browser get the public ip of nginx and search by port 19999🡨**

****

< <http://3.149.237.241:19999> >

