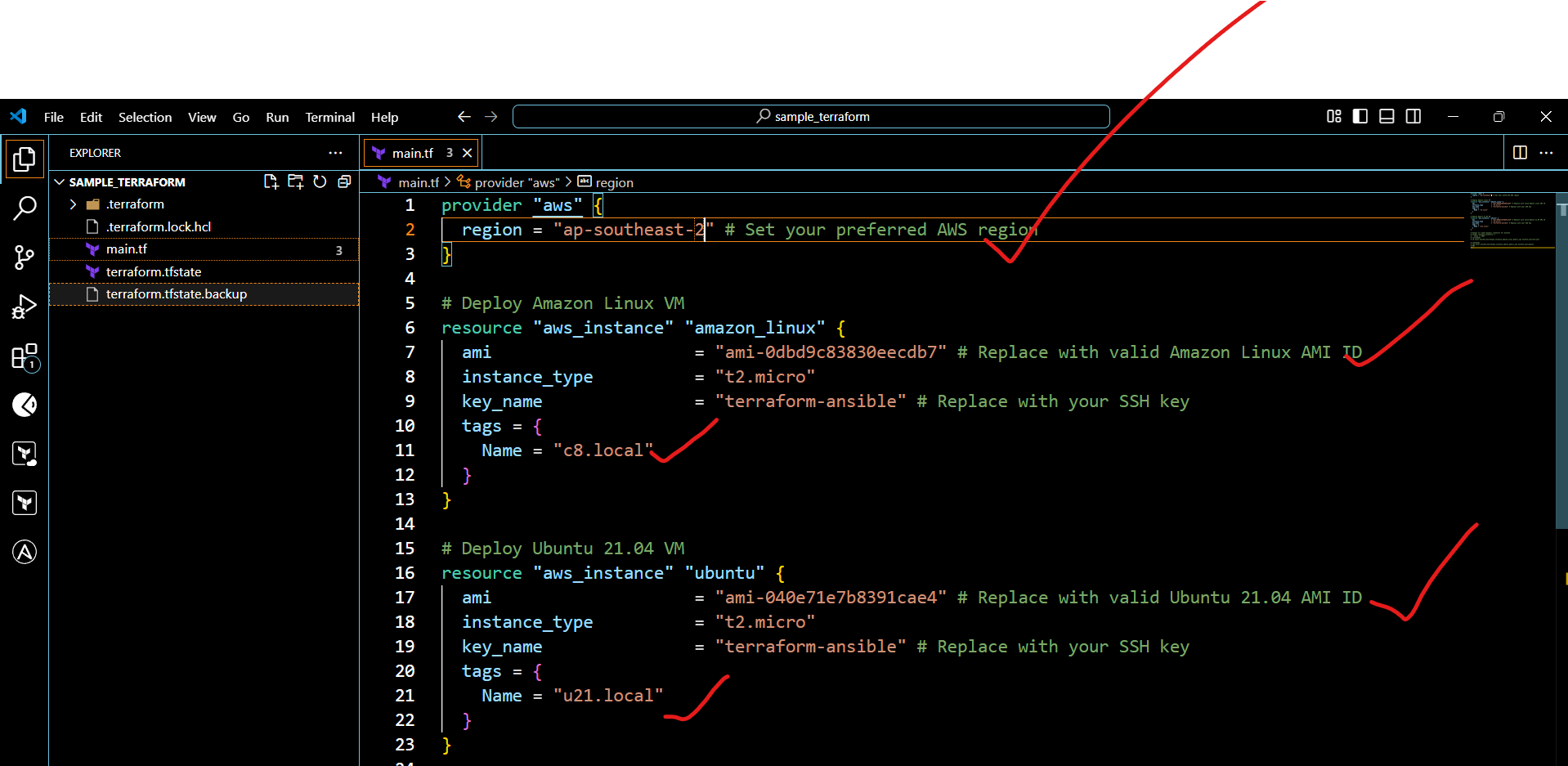
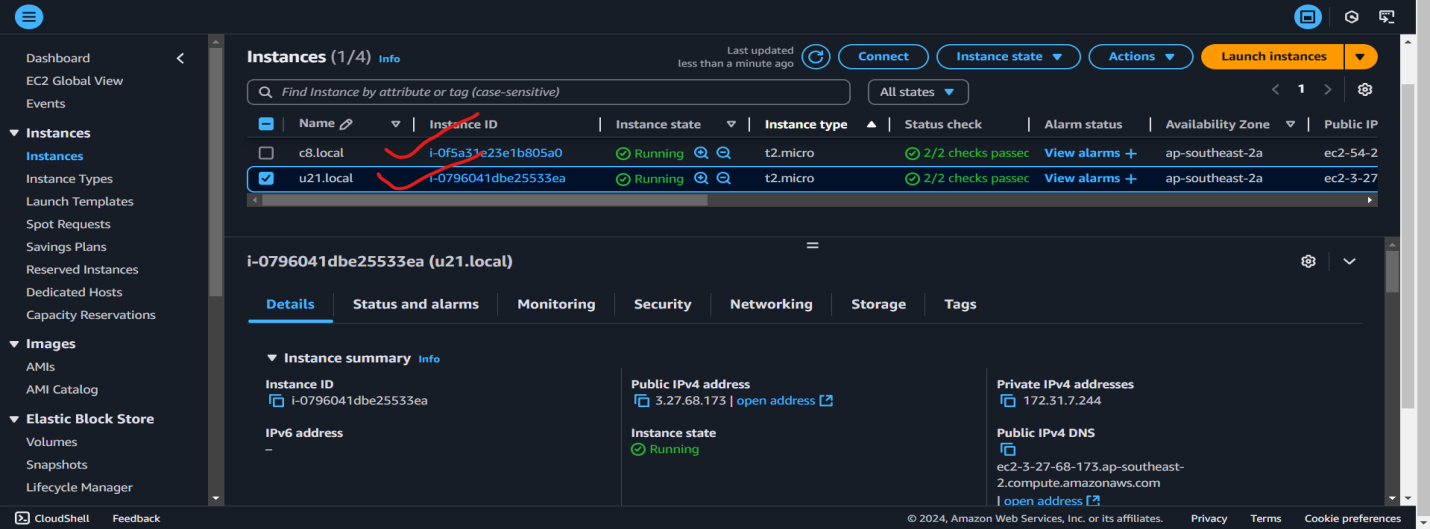
**Create ci pipeline that will deploy and configure these VMs using terraform and ansible according following requirements**

**1. Deploy**

**2 virtual machines using terraform.**

**first vm on Amazon linux, hostname: c8.local second vm on ubuntu 21.04, hostname: u21.local**

****

****

provider "aws" {

  region = "ap-southeast-2"

}

resource "aws\_instance" "amazon\_linux" {

  ami           = "ami-0d6560f3176dc9ec0"   # Amazon Linux AMI ID

  instance\_type = "t2.micro"

  key\_name      = "sample-code"

  tags = {

    Name = "aws-linux"

  }

}

resource "aws\_instance" "ubuntu" {

  ami           = "ami-003f5a76758516d1e"   # Ubuntu 21.04 AMI ID

  instance\_type = "t2.micro"

  key\_name      = "sample-code"

  tags = {

    Name = "ubuntu-linux"

  }

}

# Output VM IPs

output "amazon\_linux\_vm\_ip" {

  description = "Public IP of the Amazon Linux VM"

  value       = aws\_instance.amazon\_linux.public\_ip

}

output "ubuntu\_vm\_ip" {

  description = "Public IP of the Ubuntu VM"

  value       = aws\_instance.ubuntu.public\_ip

}

# Dynamically generate Ansible inventory file

resource "local\_file" "ansible\_inventory" {

  content = <<-EOF

[frontend]

c8.local ansible\_host=${aws\_instance.amazon\_linux.public\_ip}

[backend]

u21.local ansible\_host=${aws\_instance.ubuntu.public\_ip}

EOF

  filename = "${path.module}/ansible\_inventory.ini"

}

**ansible\_inventory.ini**

[frontend]

c8.local ansible\_host=52.63.192.129

[backend]

u21.local ansible\_host=3.27.192.212

****

**Playbook**

---

- 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

**we have to execute in this above playbook  
== > ansible-playbook playbook.yml --syntax-check**

**vim 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

****

**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

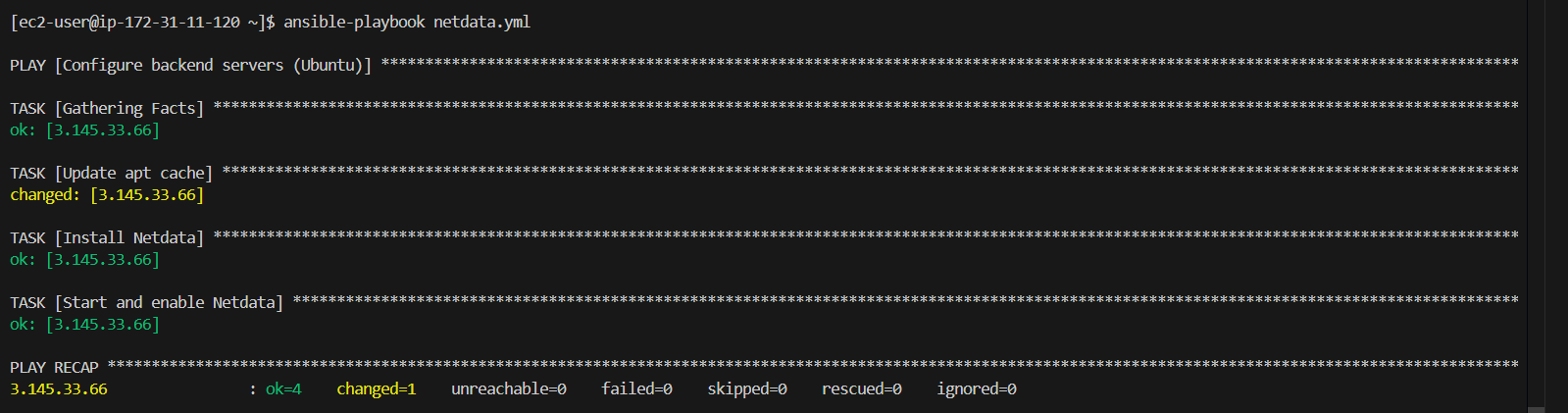
    - name: Start and enable Netdata

      service:

        name: netdata

        state: started

        enabled: yes

****