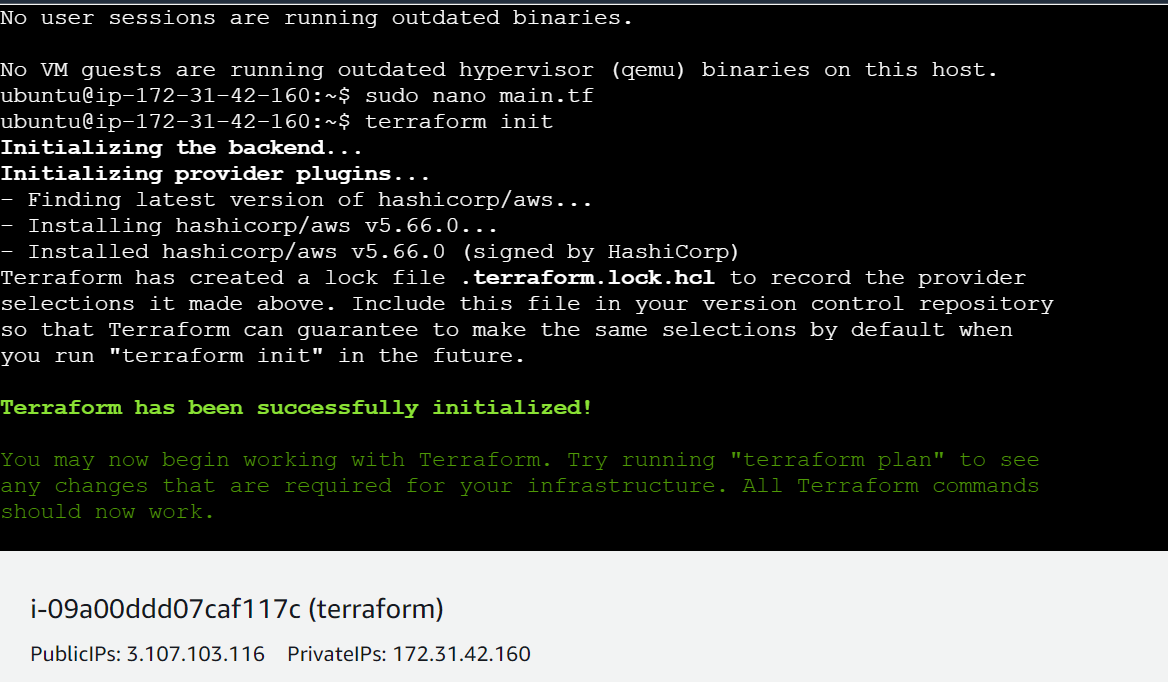
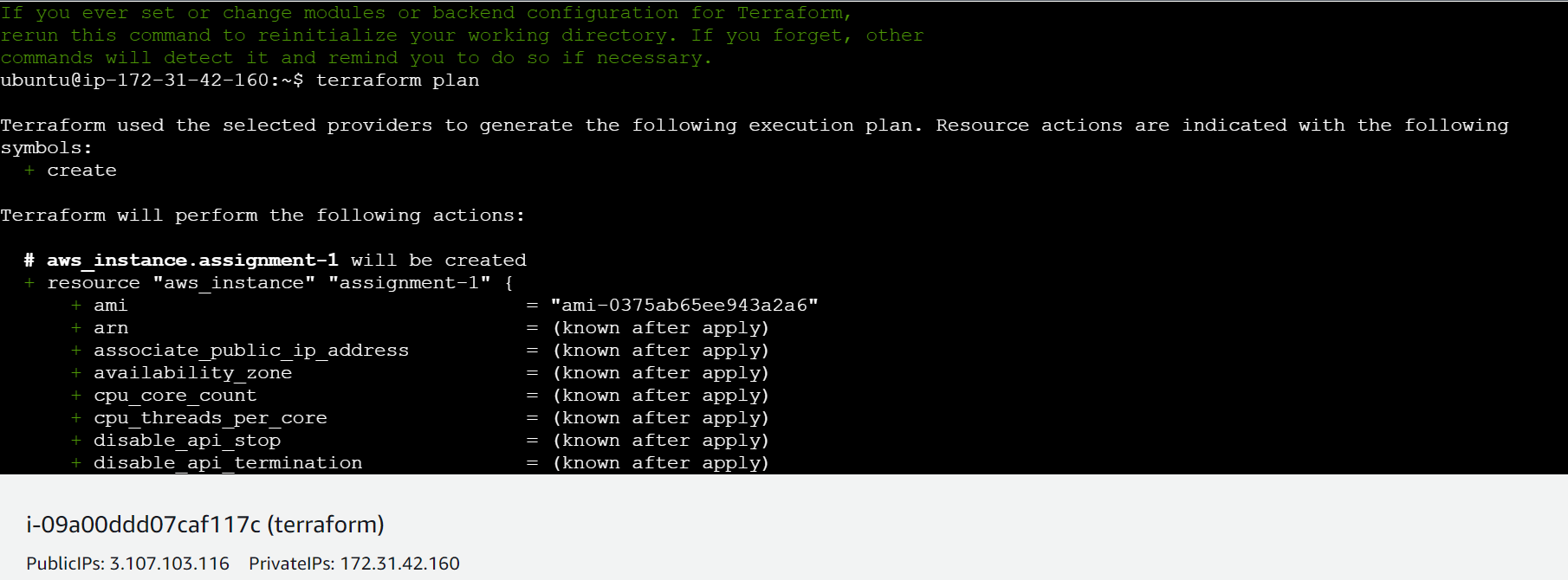
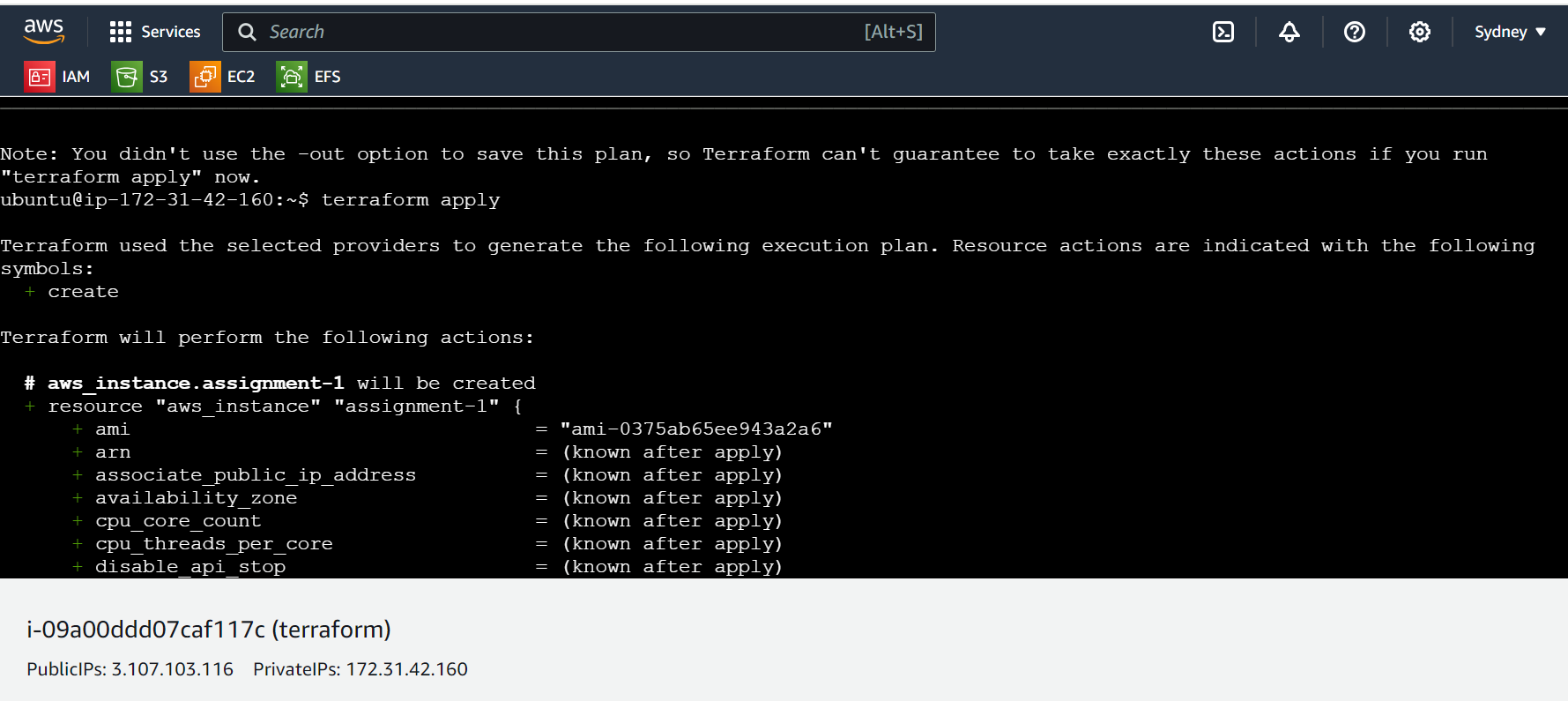
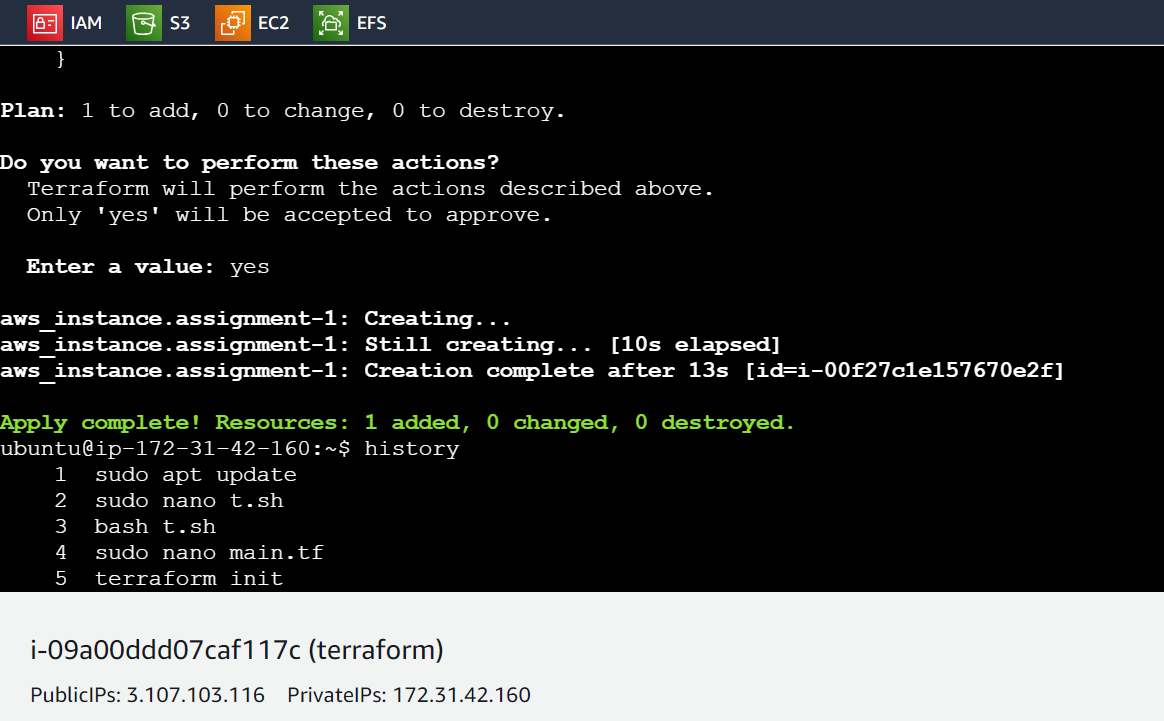
**Terraform Assignment – 1**

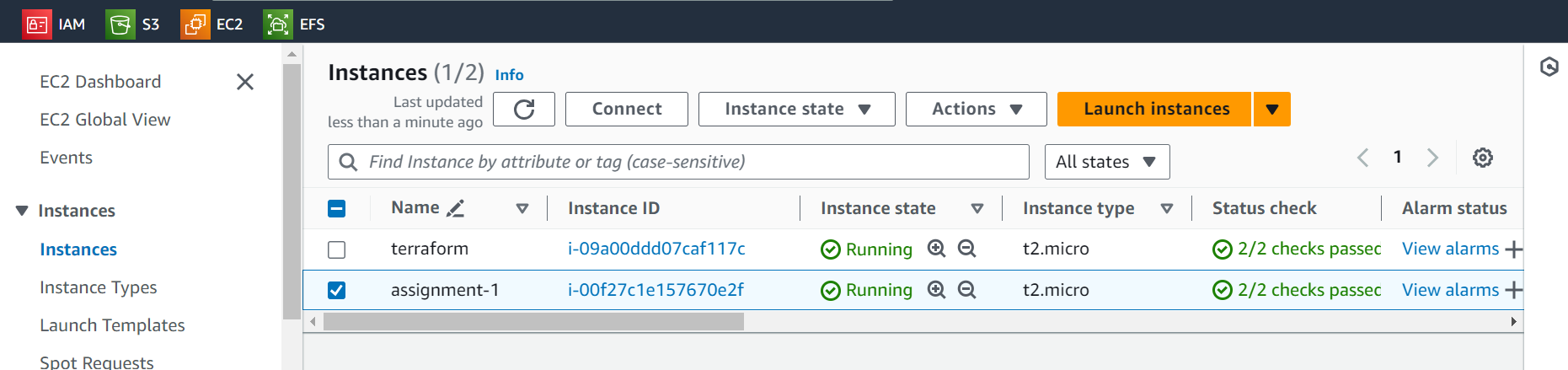
● Create an EC2 service in the default subnet in the ohio region











Main.tf –

provider "aws" {

region = "us-east-2"

access\_key = ""

secret\_key = ""

}

resource "aws\_instance" "assignment-1" {

ami = ""

instance\_type = "t2.micro"

key\_name = ""

tags = {

Name = "assignment-1"

}

}

t.sh –

wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

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 && sudo apt install terraform -y

History –

1 sudo apt update

2 sudo nano t.sh

3 bash t.sh

4 sudo nano main.tf

5 terraform init

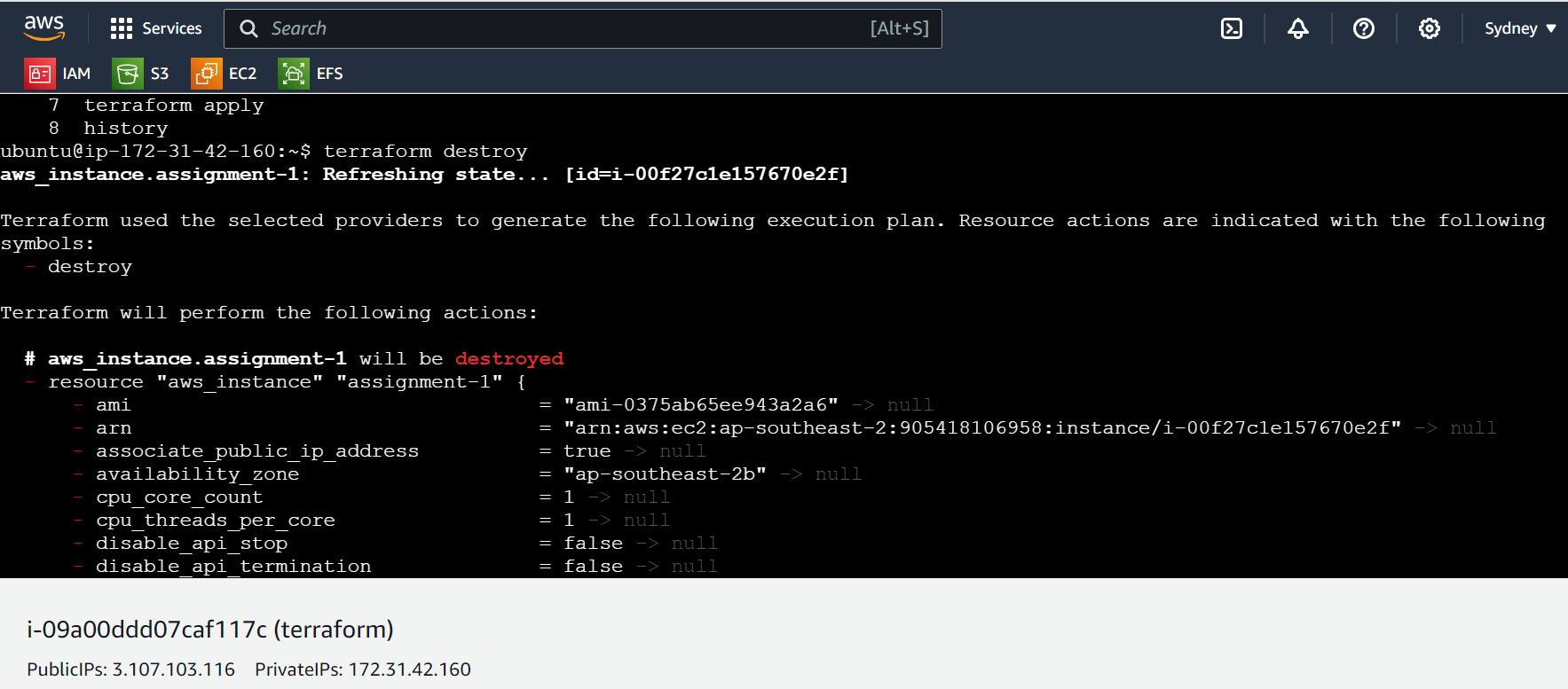
6 terraform plan

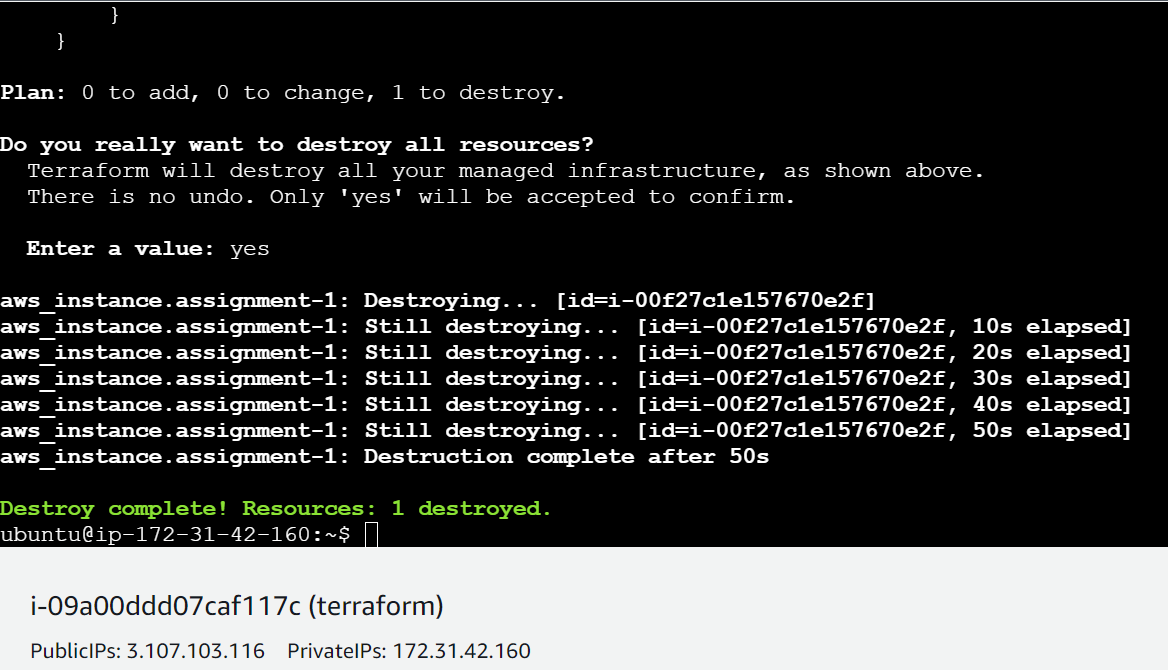
7 terraform apply

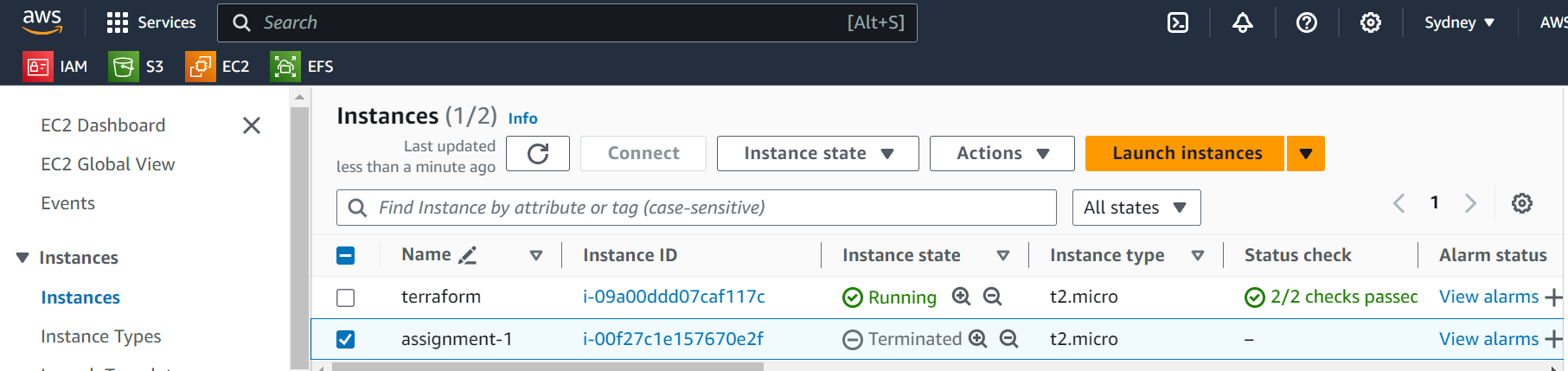
8 history

**Terraform Assignment – 2**

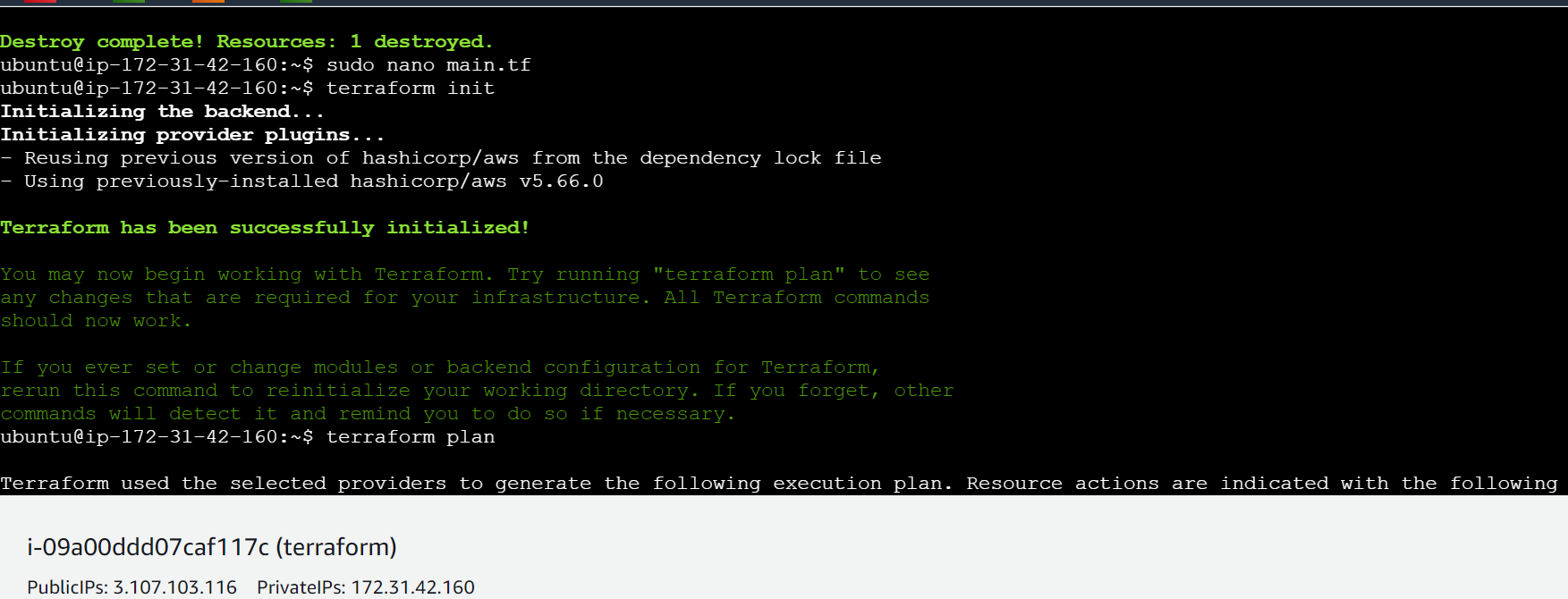
● Destroy the previous deployment

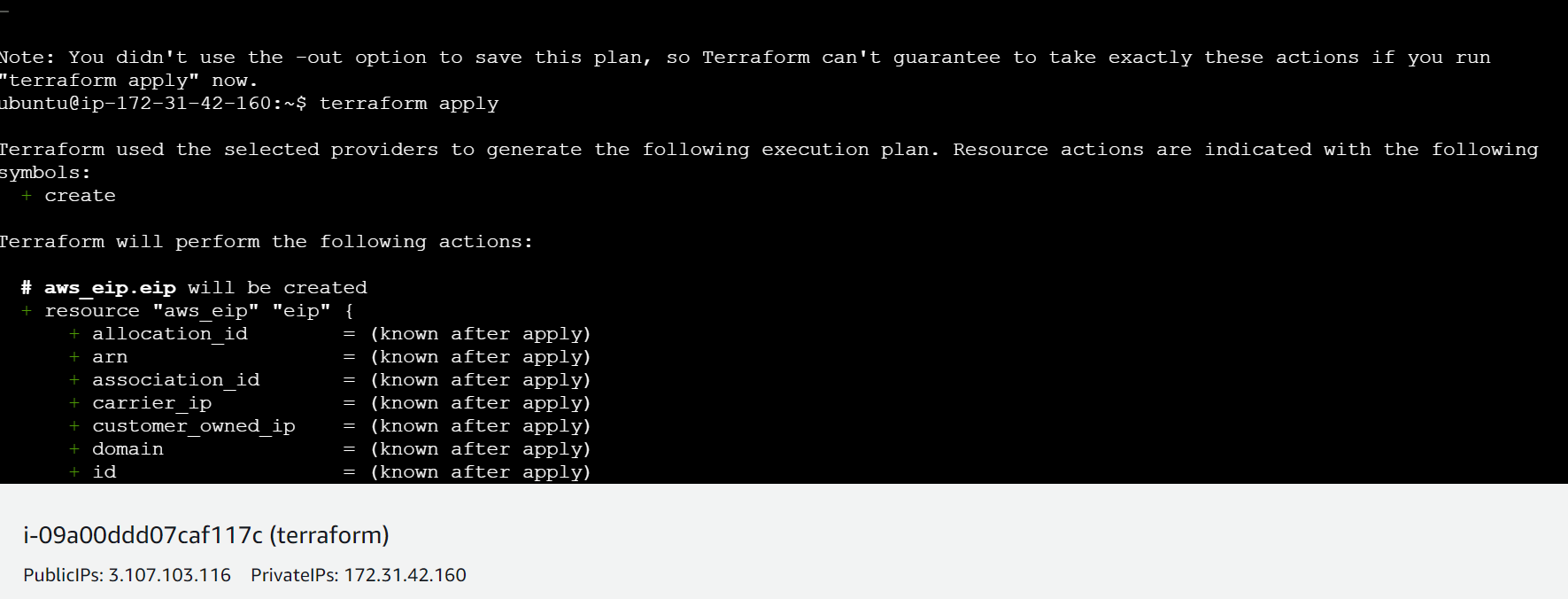


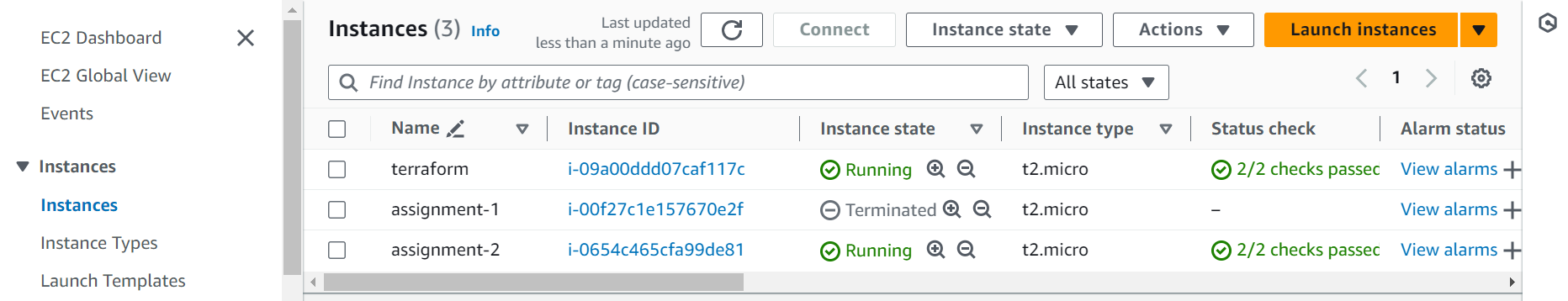




● Create a new EC2 instance with an Elastic IP







Main.tf –

provider "aws" {

region = "us-east-2"

access\_key = ""

secret\_key = ""

}

resource "aws\_instance" "assignment-2" {

ami = ""

instance\_type = "t2.micro"

key\_name = ""

tags = {

Name = "assignment-2"

}

}

resource "aws\_eip" "eip" {

vpc = true

}

resource "aws\_eip\_association" "eip\_assoc" {

instance\_id = aws\_instance.assignment-2.id

allocation\_id = aws\_eip.eip.id

}

History –

9 terraform destroy

10 sudo nano main.tf

11 terraform init

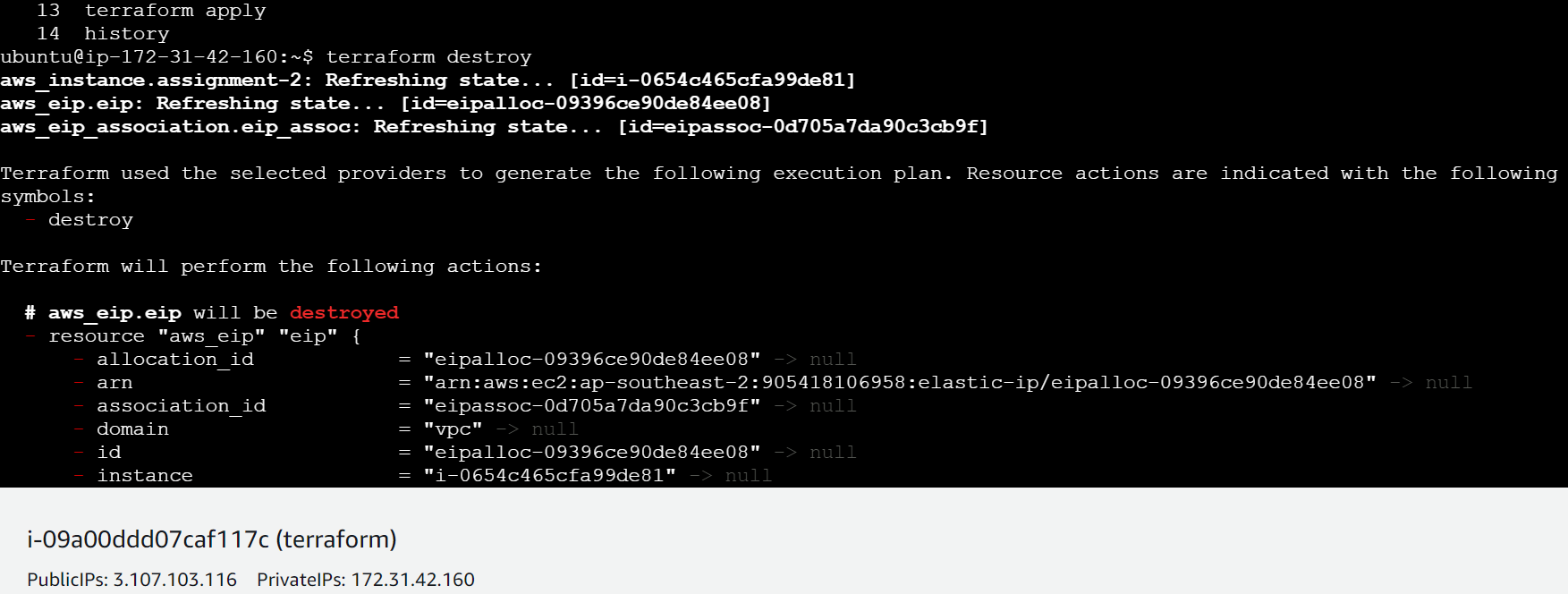
12 terraform plan

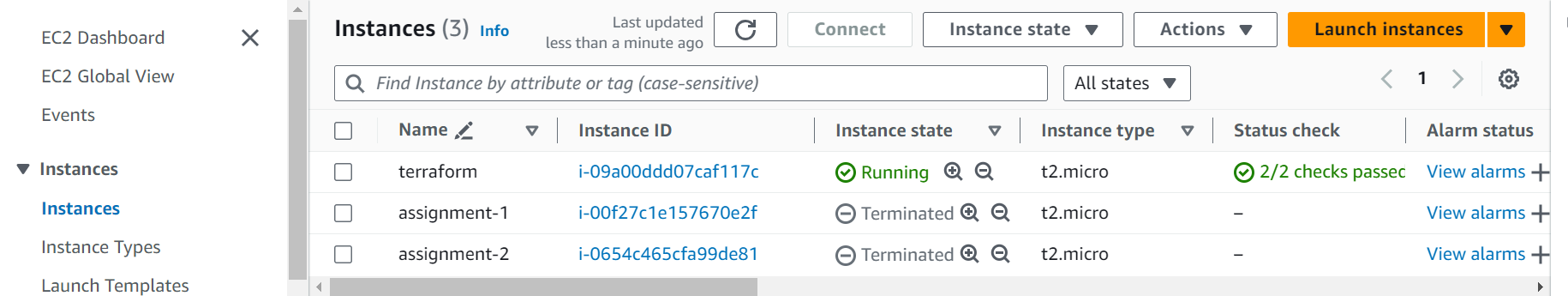
13 terraform apply

14 history

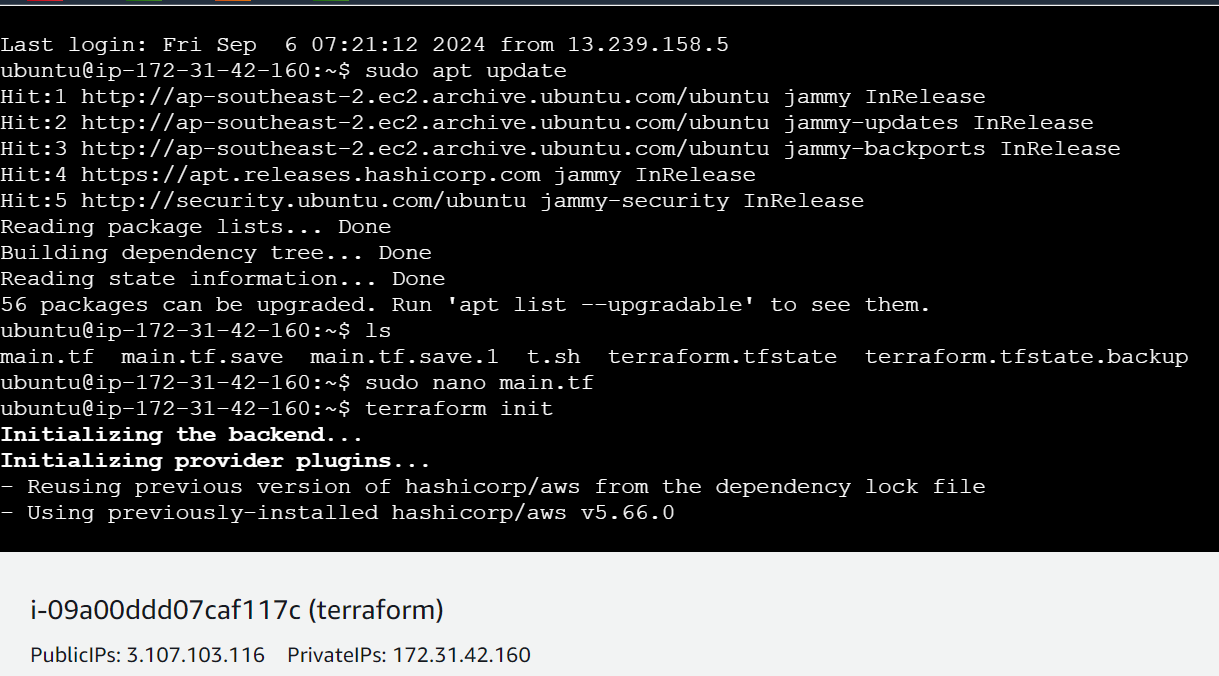
**Terraform Assignment – 3**

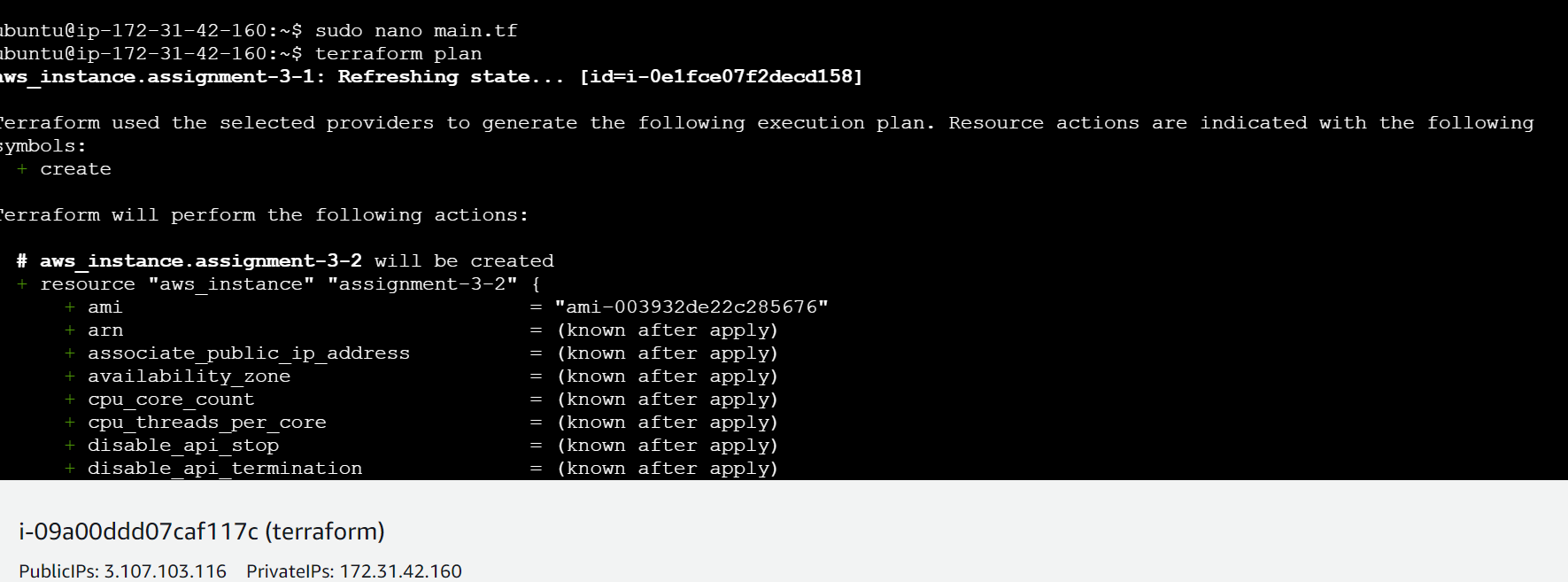
● Destroy the previous deployment

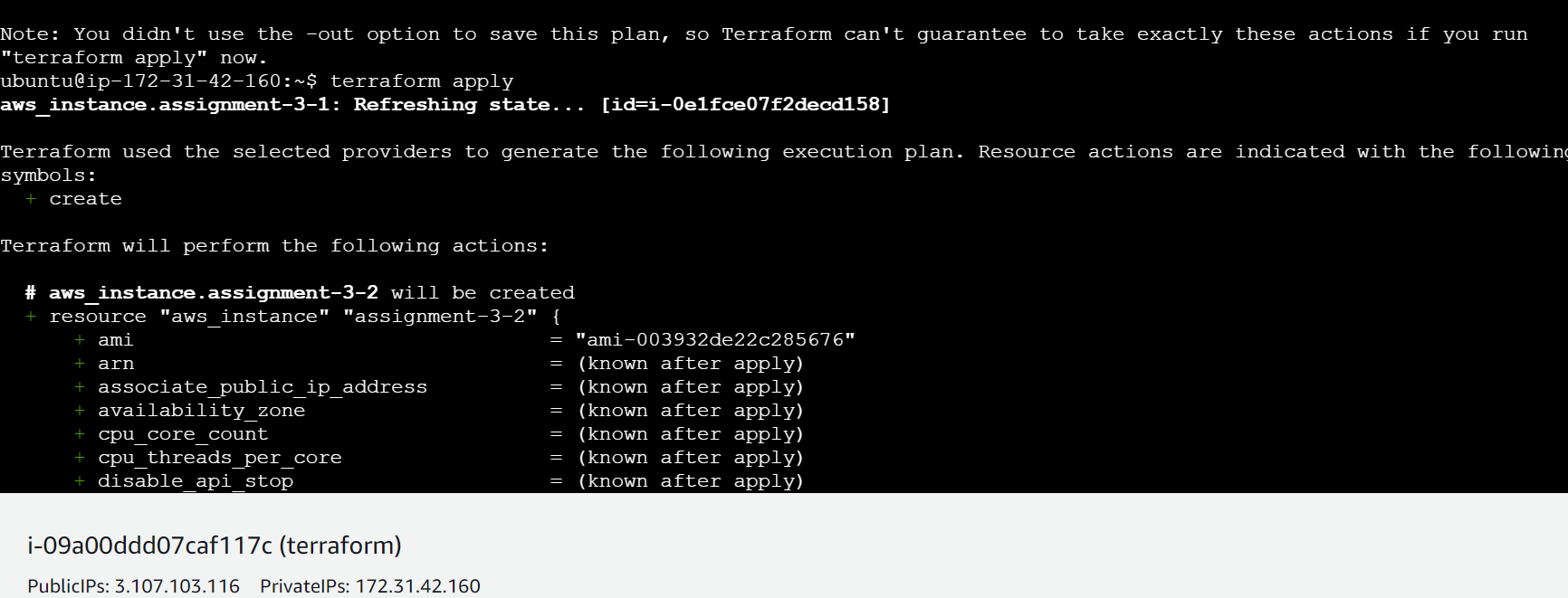


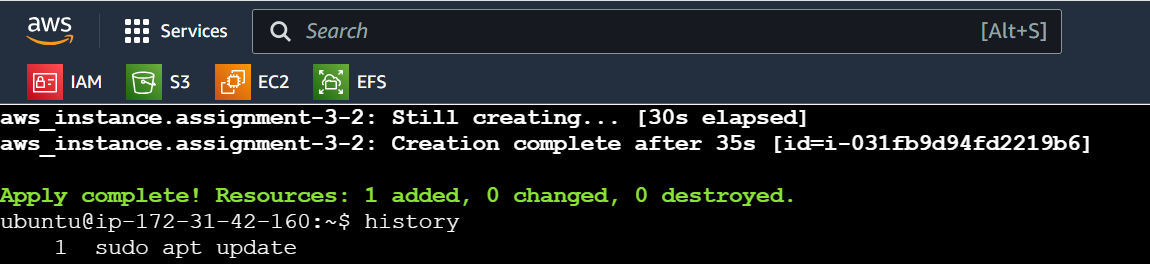


● Create 2 EC2 instances in sydeny and ohio respectively

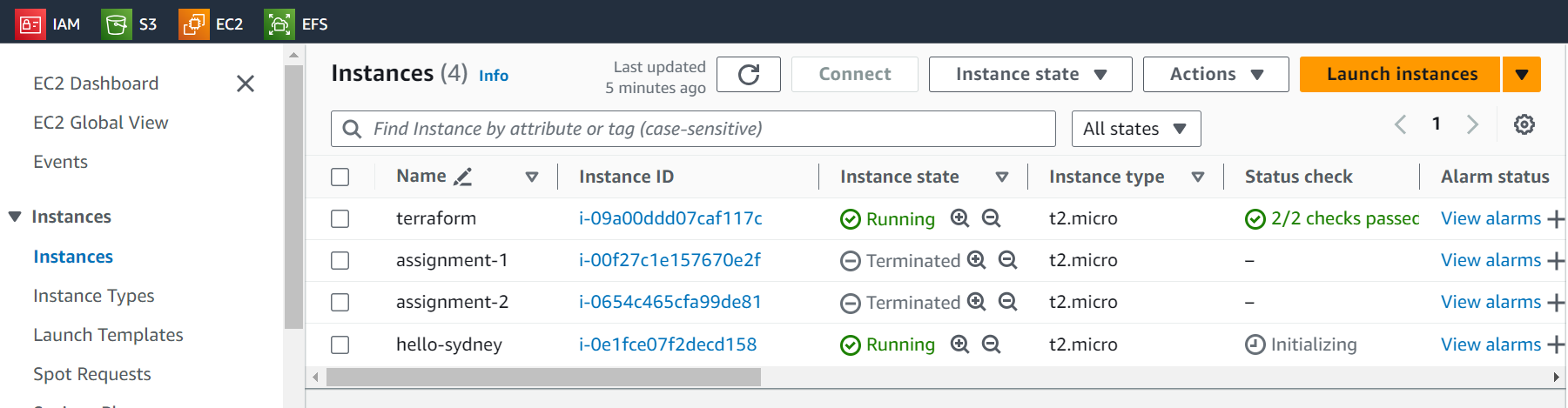


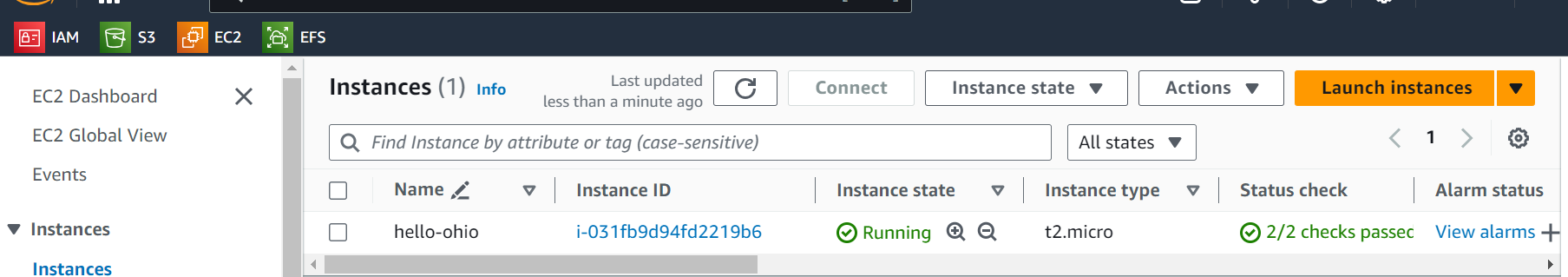






● Rename Ohio’s instance to ‘hello-ohio’ and Virginia’s instance to ‘hello-virginia’





Main.tf –

provider "aws" {

alias = "Sydney"

region = "ap-southeast-2"

access\_key = "AKIA5FTY75BHE3UFGNWD"

secret\_key = "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"

}

provider "aws" {

alias = "Ohio"

region = "us-east-2"

access\_key = "AKIA5FTY75BHE3UFGNWD"

secret\_key = "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"

}

resource "aws\_instance" "assignment-3-1" {

provider = aws.Sydney

ami = "ami-0375ab65ee943a2a6"

instance\_type = "t2.micro"

key\_name = "Git"

tags = {

Name = "hello-sydney"

}

}

resource "aws\_instance" "assignment-3-2" {

provider = aws.Ohio

ami = "ami-003932de22c285676"

instance\_type = "t2.micro"

key\_name = "Git"

tags = {

Name = "hello-ohio"

}

}

History –

15 terraform destroy

16 sudo nano main.tf

17 sudo apt update

18 ls

19 sudo nano main.tf

20 terraform init

21 terraform plan

22 terraform apply

23 sudo nano main.tf

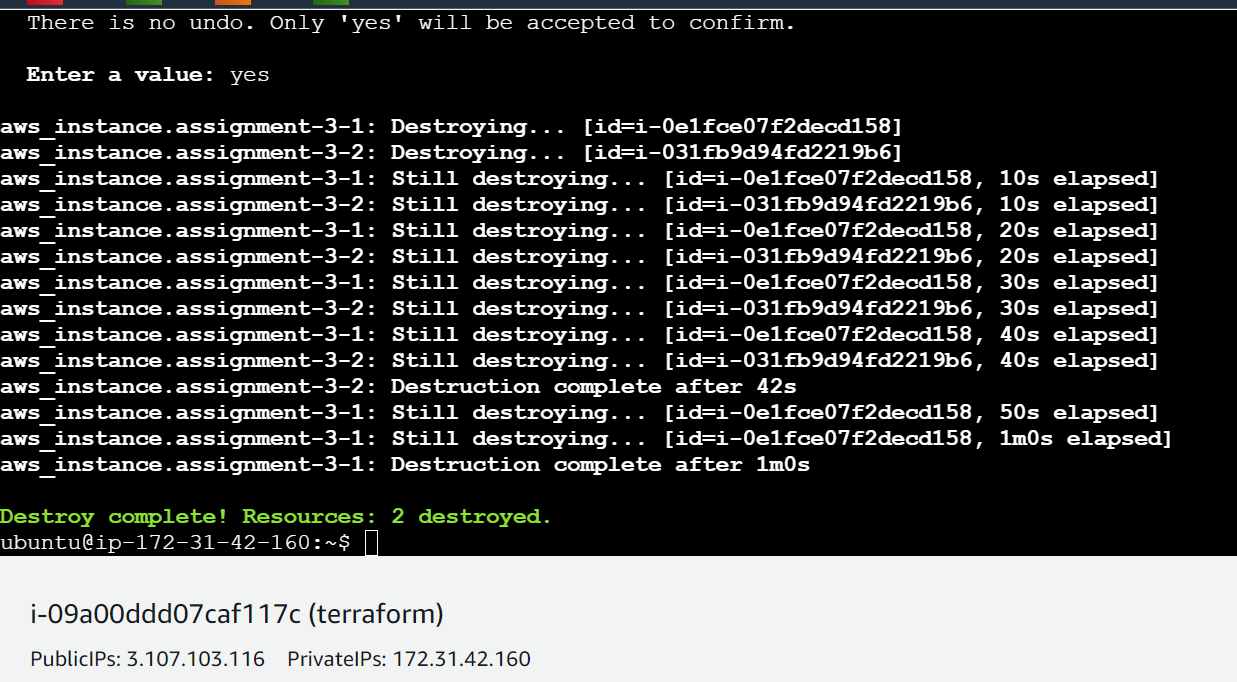
24 terraform plan

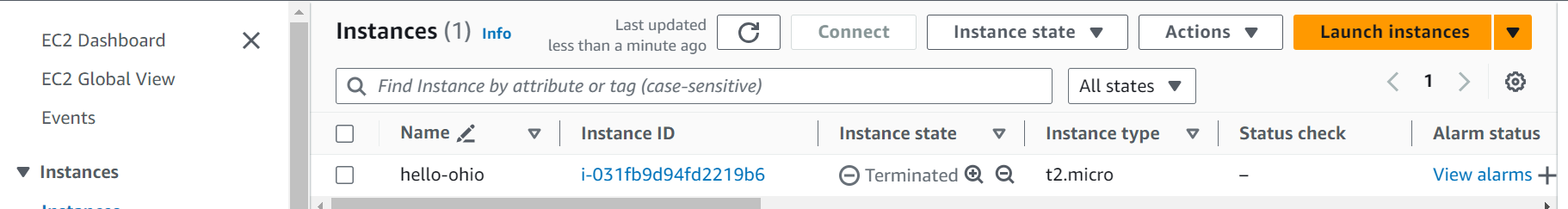
25 terraform apply

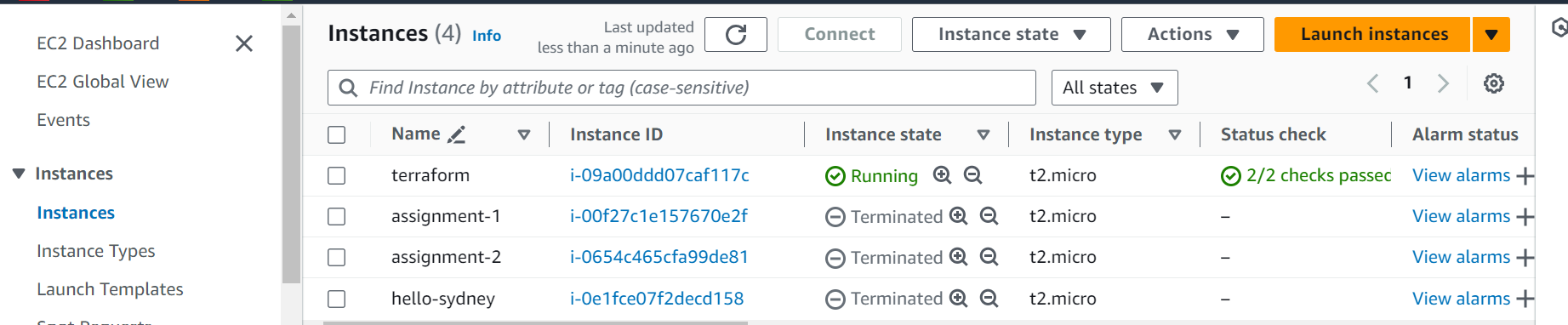
26 history

**Terraform Assignment – 4**

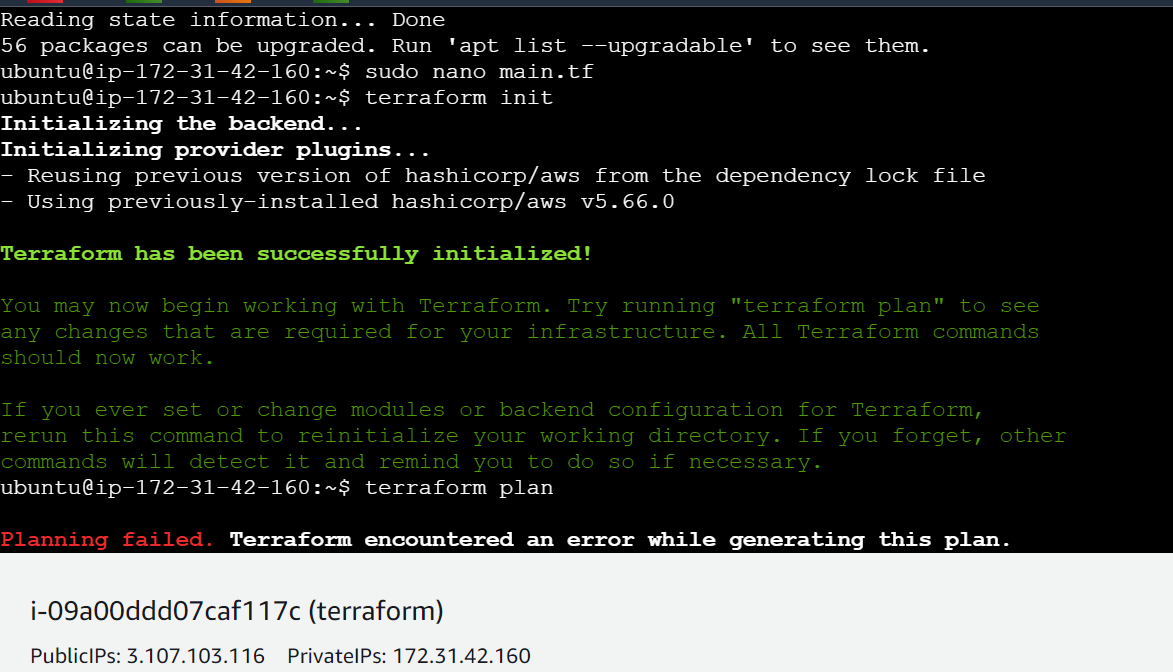
● Destroy the previous deployments

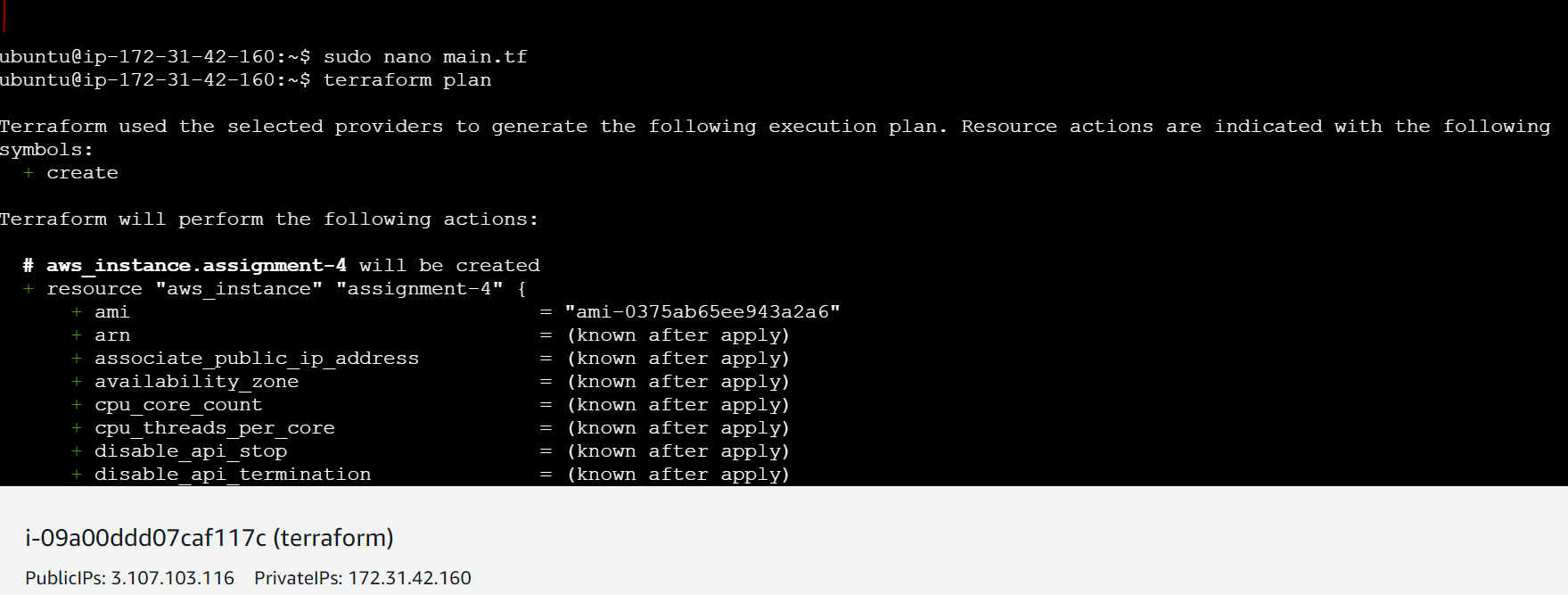


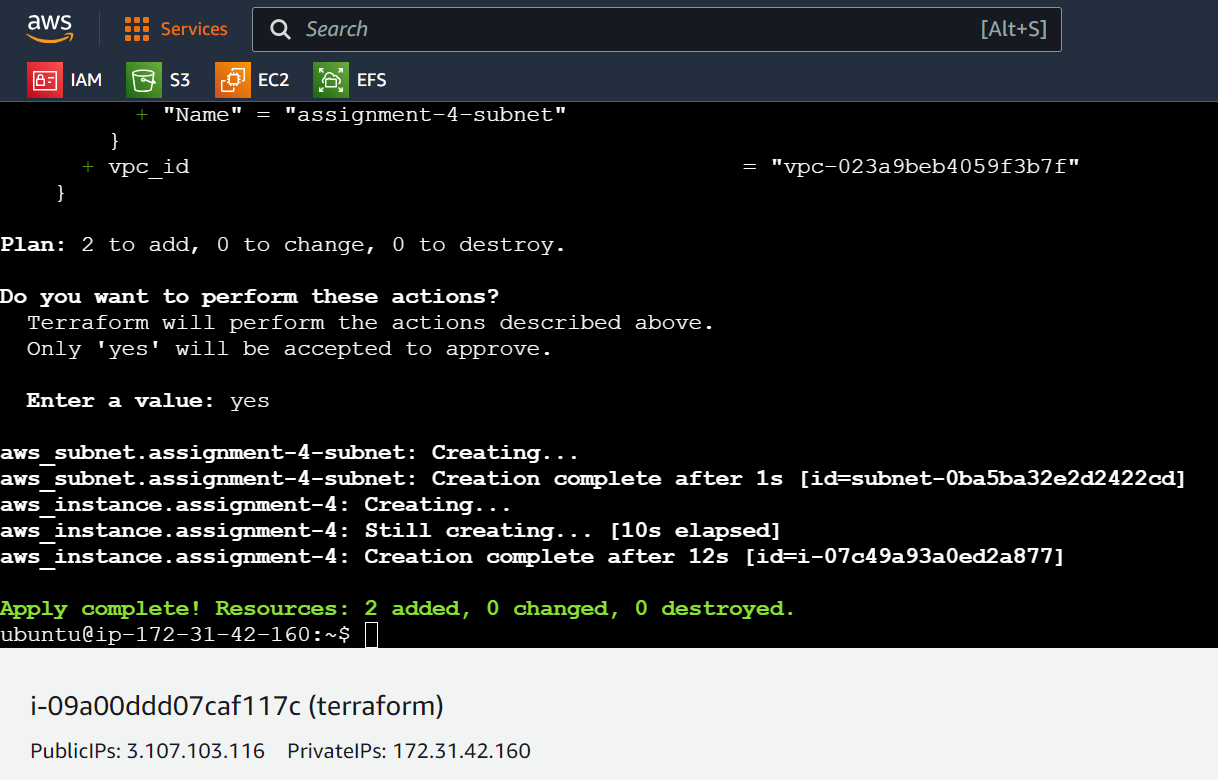




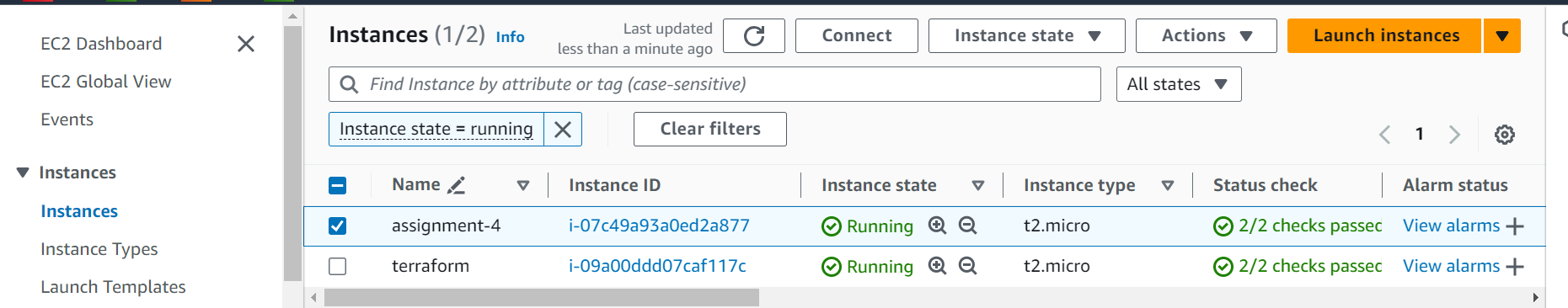
● Create a VPC with the required components using Terraform

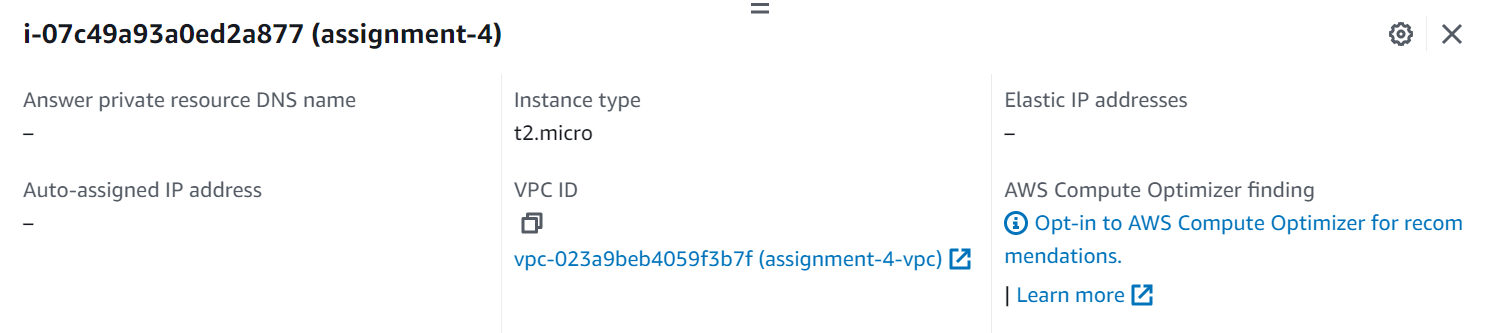


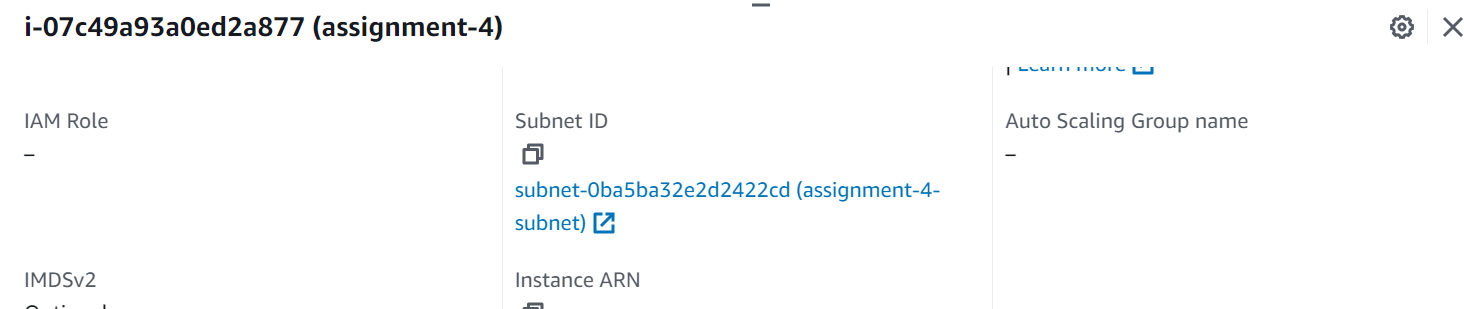




● Deploy an EC2 instance inside the VPC







Main.tf –

provider "aws" {

region = "us-east-1"

access\_key = ""

secret\_key = ""

}

resource "aws\_instance" "assignment-4" {

ami = "ami-0e001c9271cf7f3b9"

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.assignment-4-subnet.id

key\_name = "sagar-newnvkey"

tags = {

Name = "assignment-4"

}

}

resource "aws\_vpc" "assignment-4-vpc" {

cidr\_block = "10.0.0.0/16"

tags = {

Name = "assignment-4-vpc"

}

}

resource "aws\_subnet" "assignment-4-subnet" {

vpc\_id = aws\_vpc.assignment-4-vpc.id

cidr\_block = "10.0.0.0/23"

availability\_zone = "us-east-1a"

tags = {

Name = "assignment-4-subnet"

}

}

History –

30 terraform destroy

31 sudo apt update

32 sudo nano main.tf

33 terraform init

34 terraform plan

35 sudo nano main.tf

36 terraform plan

37 terraform apply

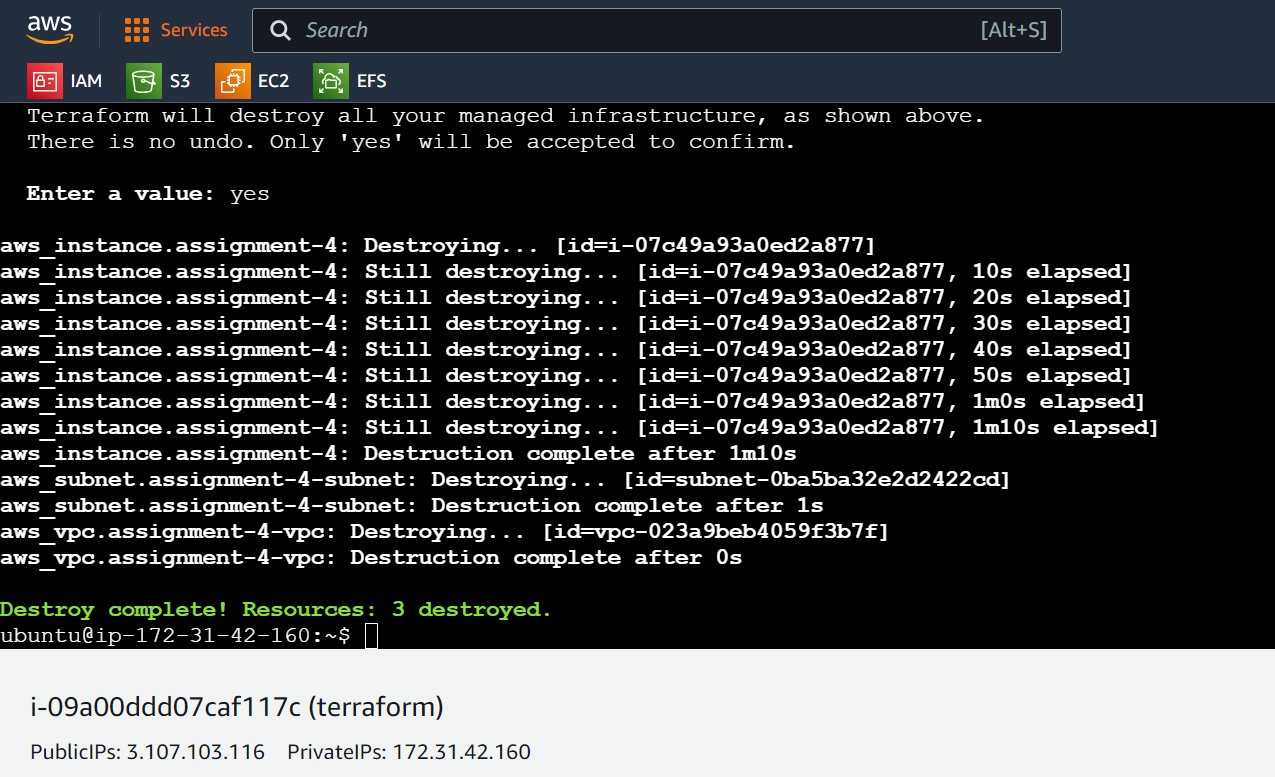
38 sudo nano main.tf

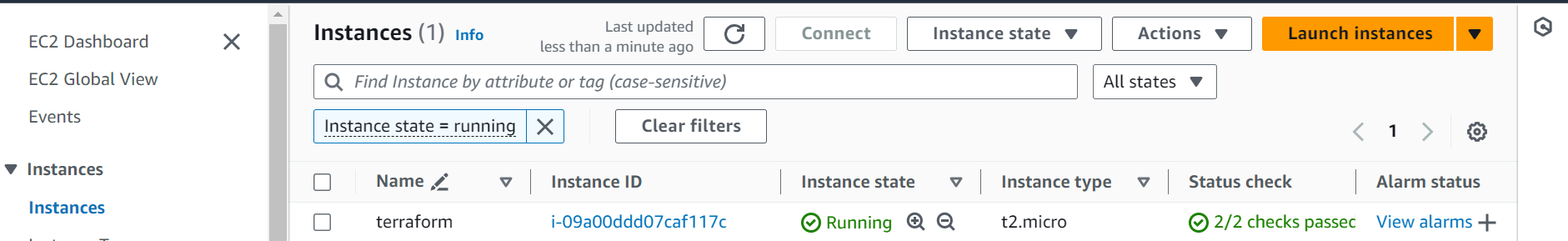
39 terraform apply

40 history

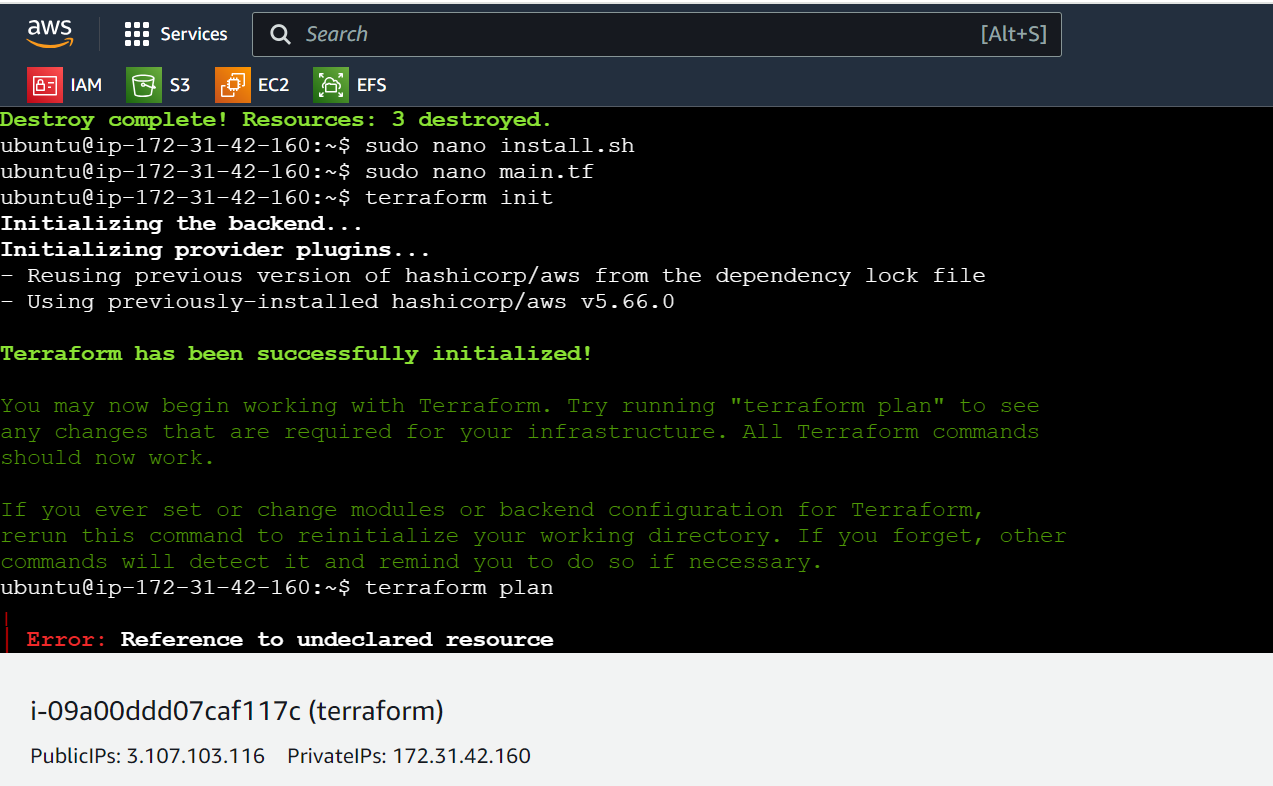
**Terraform Assignment – 5**

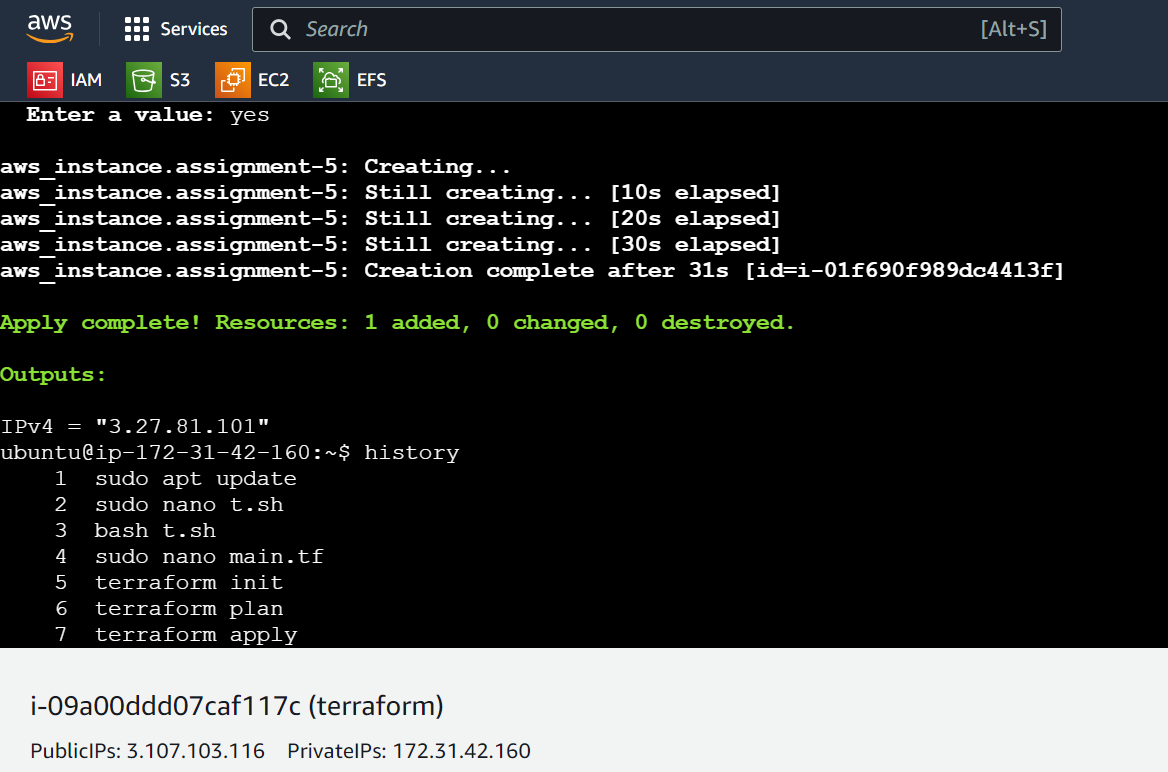
● Destroy the previous deployments





● Create a script to install apache2





● Run this script on a newly created EC2 instance

Install.sh (script)

#!/bin/bash

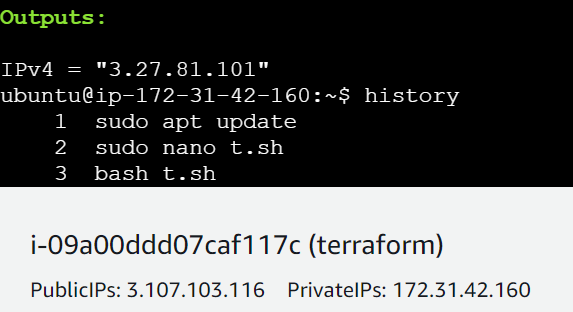
sudo apt update -y

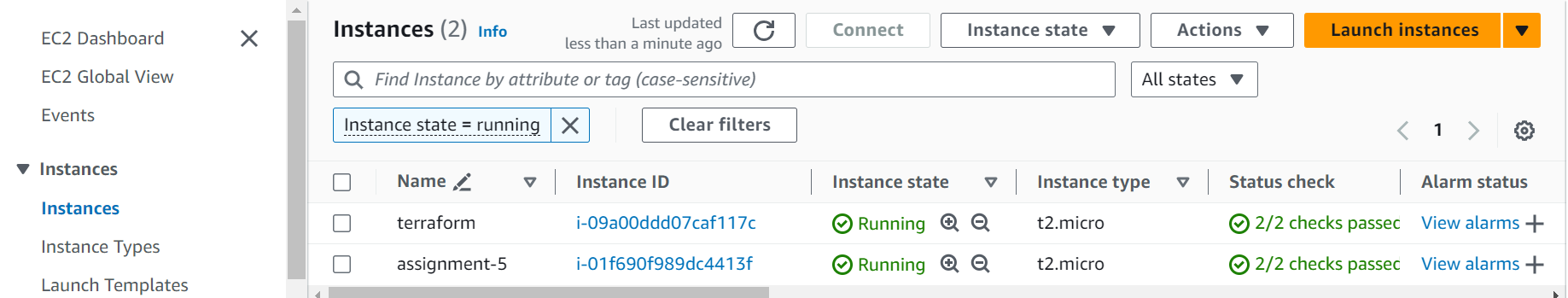
sudo apt install apache2 -y

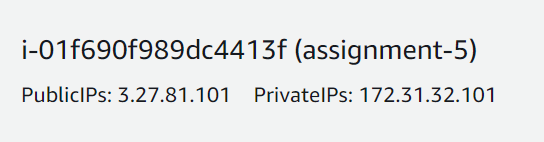
sudo su

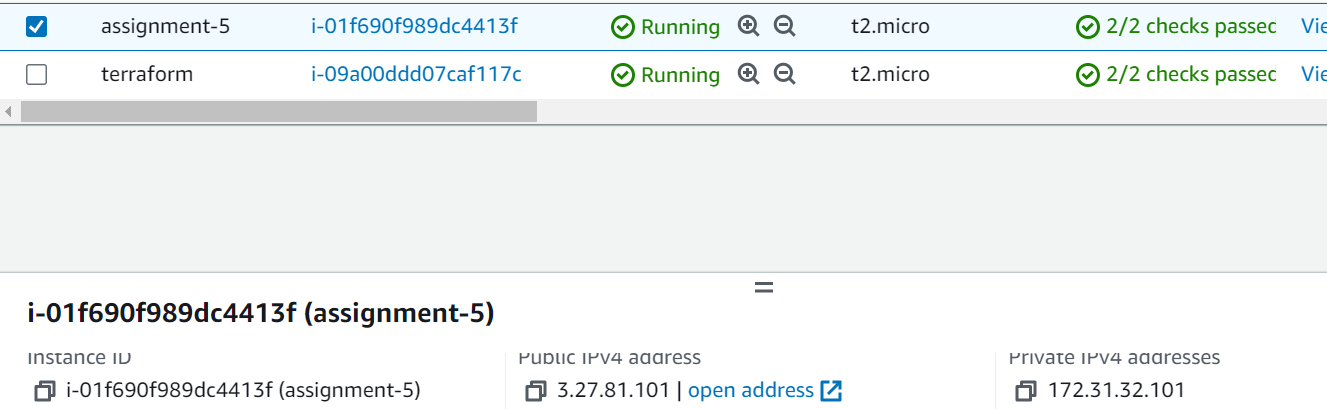
echo "Custom html page" > /var/www/html/index.html

● Print the IP address of the instance in a file on the local, once deployed









Main.tf –

provider "aws" {

access\_key = ""

secret\_key = ""

region = "us-east-1"

}

resource "aws\_instance" "a5-instance" {

ami = "ami-0e001c9271cf7f3b9"

instance\_type = "t2.micro"

key\_name = ""

user\_data = "${file("install.sh")}"

tags = {

Name = "a5-instance"

}

}

output "IPv4" {

value = aws\_instance.a5-instance.public\_ip

}

History –

41 terraform destroy

42 sudo nano install.sh

43 sudo nano main.tf

44 terraform init

45 terraform plan

46 sudo nano main.tf

47 terraform plan

48 sudo nano main.tf

49 terraform plan

50 terraform apply

51 history