# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 14

Date Of Performance: 10/10/2025

Aim: To understand Infrastructure as Code (IaC) concepts and implement AWS EC2 instance provisioning using Terraform's lifecycle management.

1. What is Terraform?

2. What is Infrastructure as a Code (IaC) ?

3. Perform an experiment, to understand Terraform lifecycle, core concepts/terminologies and install it on a Linux Machine.

4. Using Terraform , create an EC2 instance on AWS cloud

5. Explain following Terraform commands in one line

* terraform init:
* terraform validate:
* terraform plan:
* terraform apply:
* terraform destroy.

Ans.1:

An open-source Infrastructure as Code (IaC) tool by HashiCorp that uses declarative configuration files to define, provision, and manage cloud infrastructure safely and efficiently.

Ans.2:

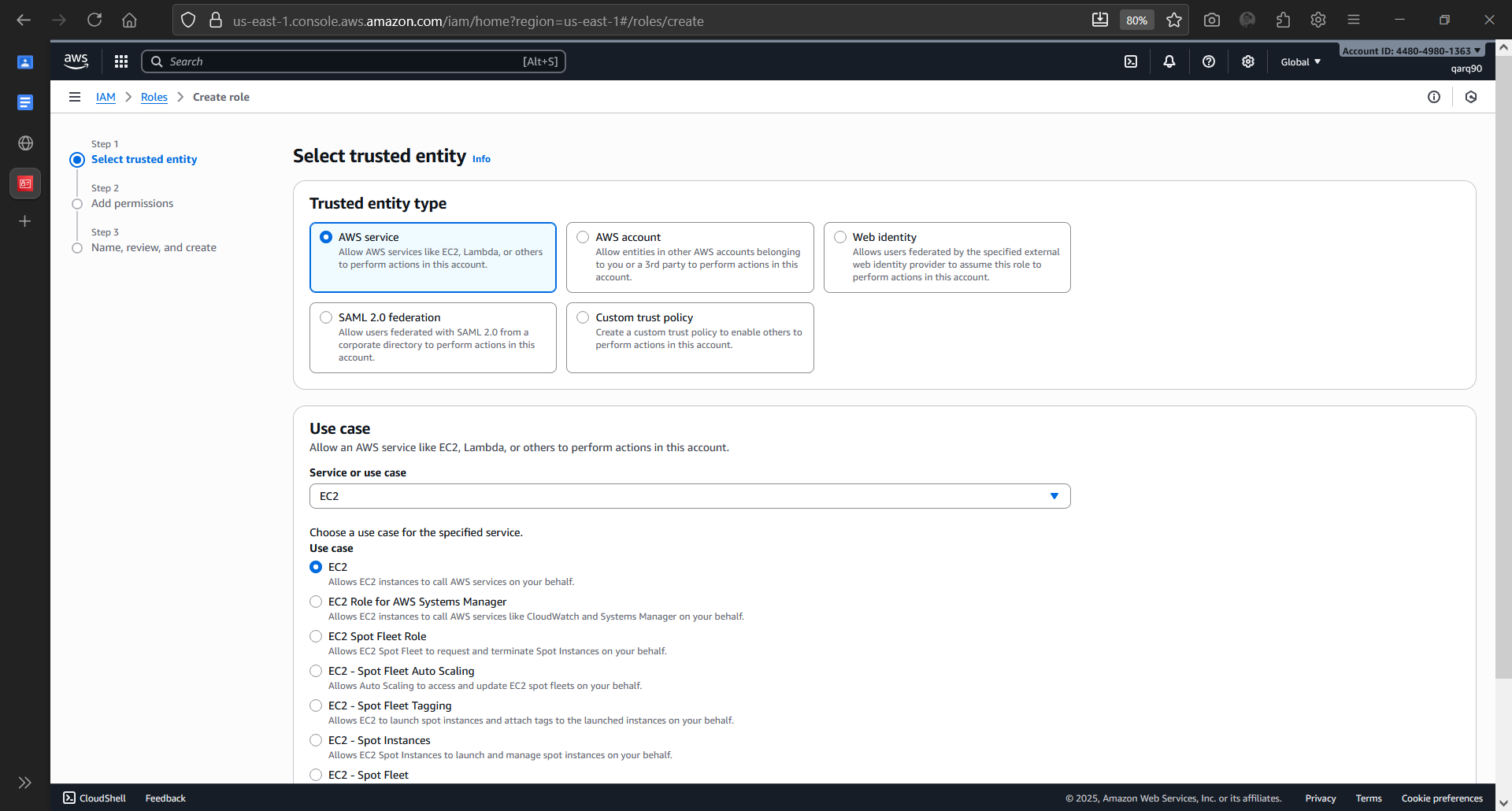
Infrastructure as Code (IaC) is a key DevOps practice that manages and provisions computing infrastructure through machine-readable definition files, instead of manual processes. It treats servers, networks, and databases as software, defined in structured code.

This enables automated, consistent, and repeatable environment setups, version control for infrastructure changes, and significantly reduces human error, leading to faster, more reliable deployments across development, staging, and production.

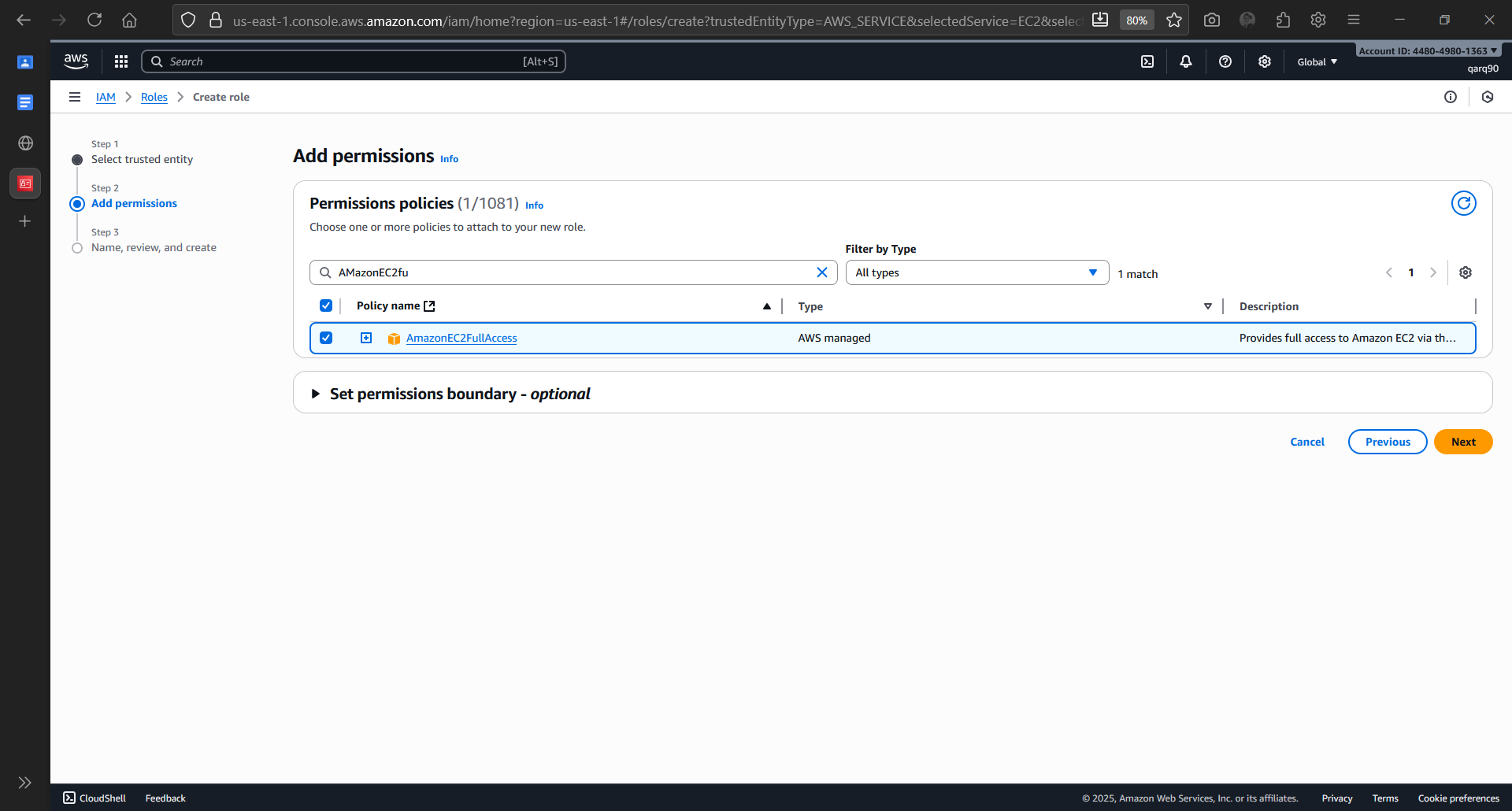
Ans.5:

* terraform init: Initializes a working directory, downloading provider plugins.
* terraform validate: Checks the configuration files for syntax errors.
* terraform plan: Creates an execution plan showing what infrastructure will be created.
* terraform apply: Builds or changes the infrastructure as defined in the plan.
* terraform destroy: Destroys all Terraform-managed infrastructure.

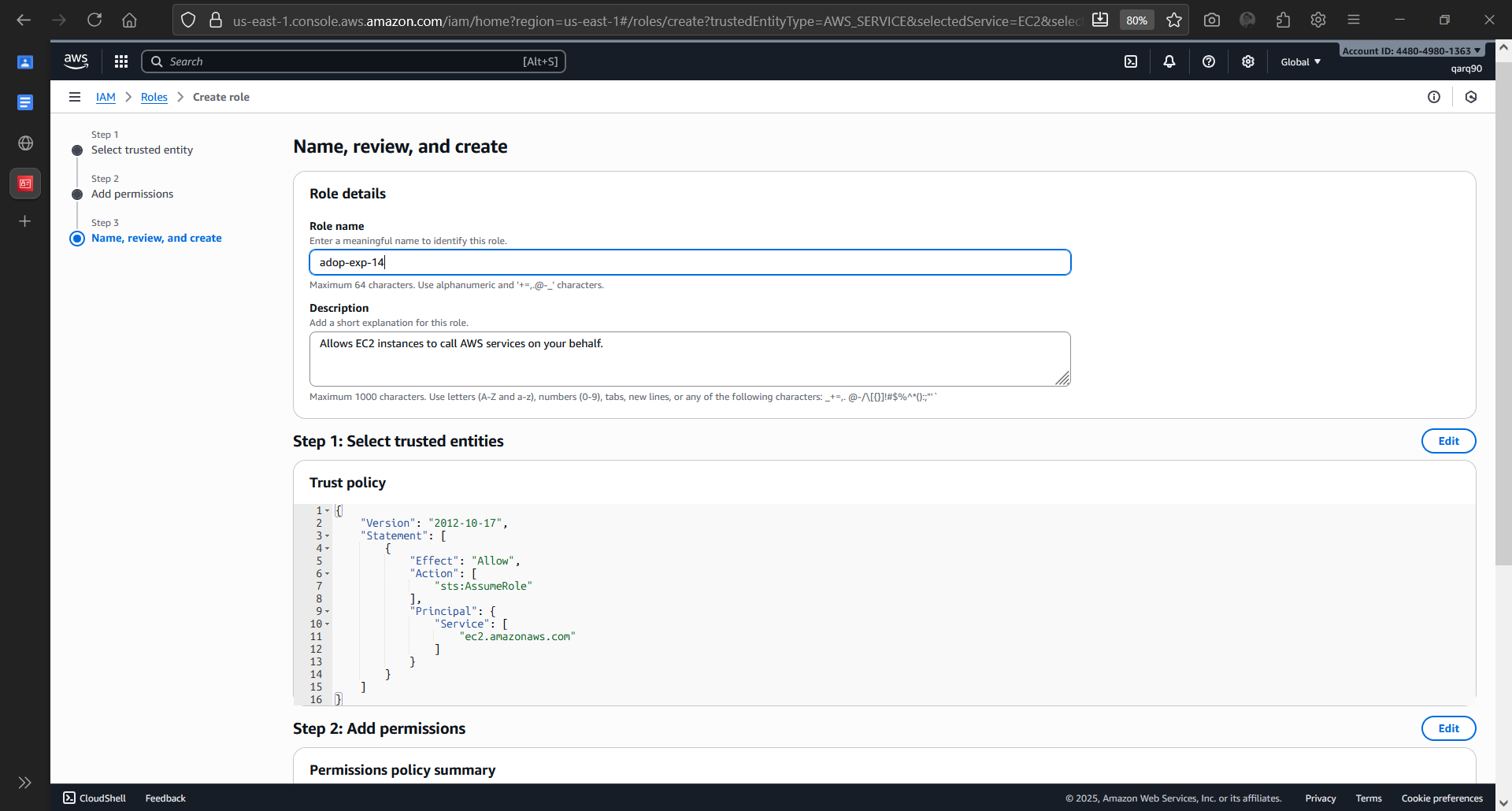
Ans.3 & 4:



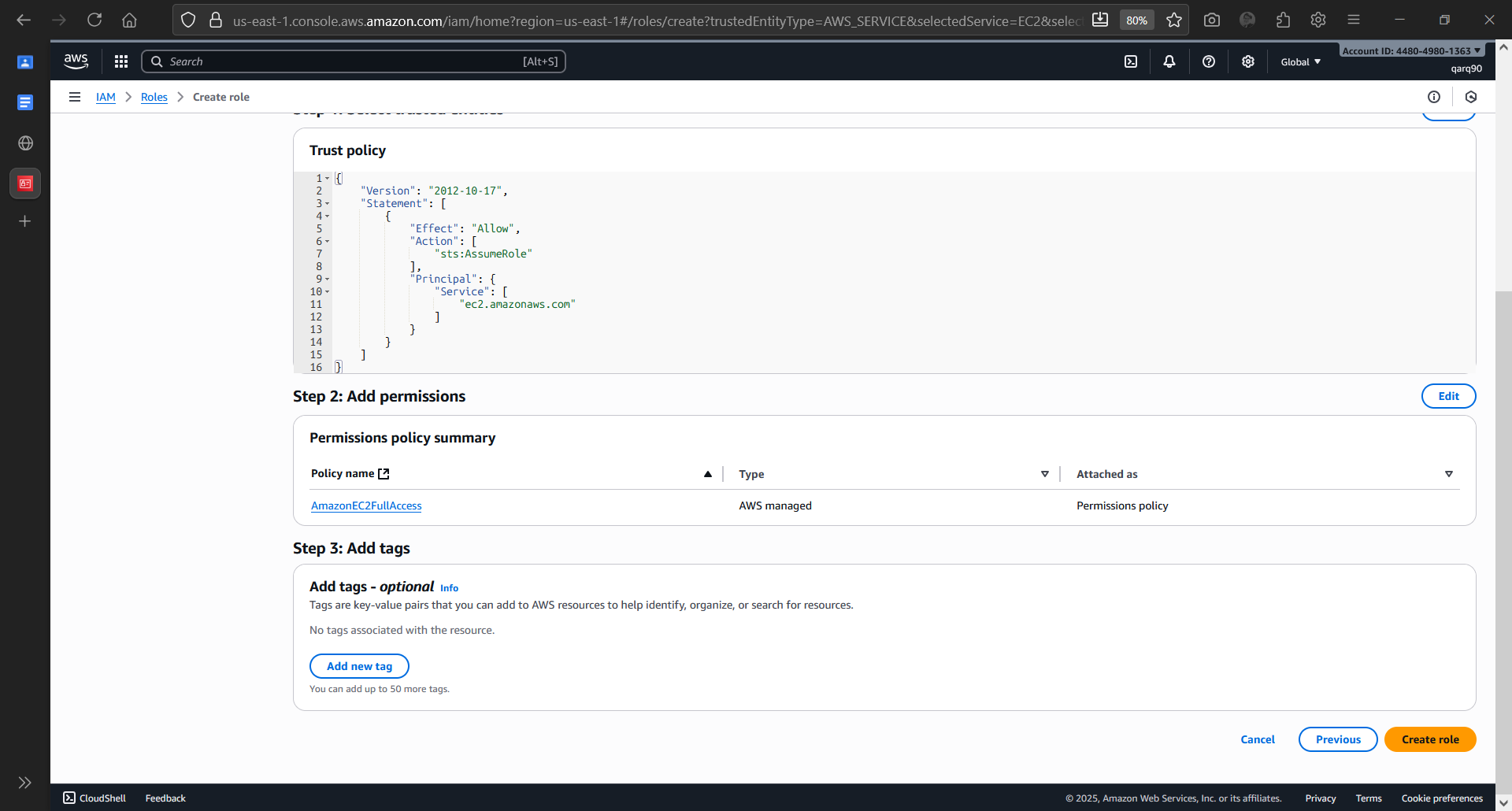
Creating IAM Role



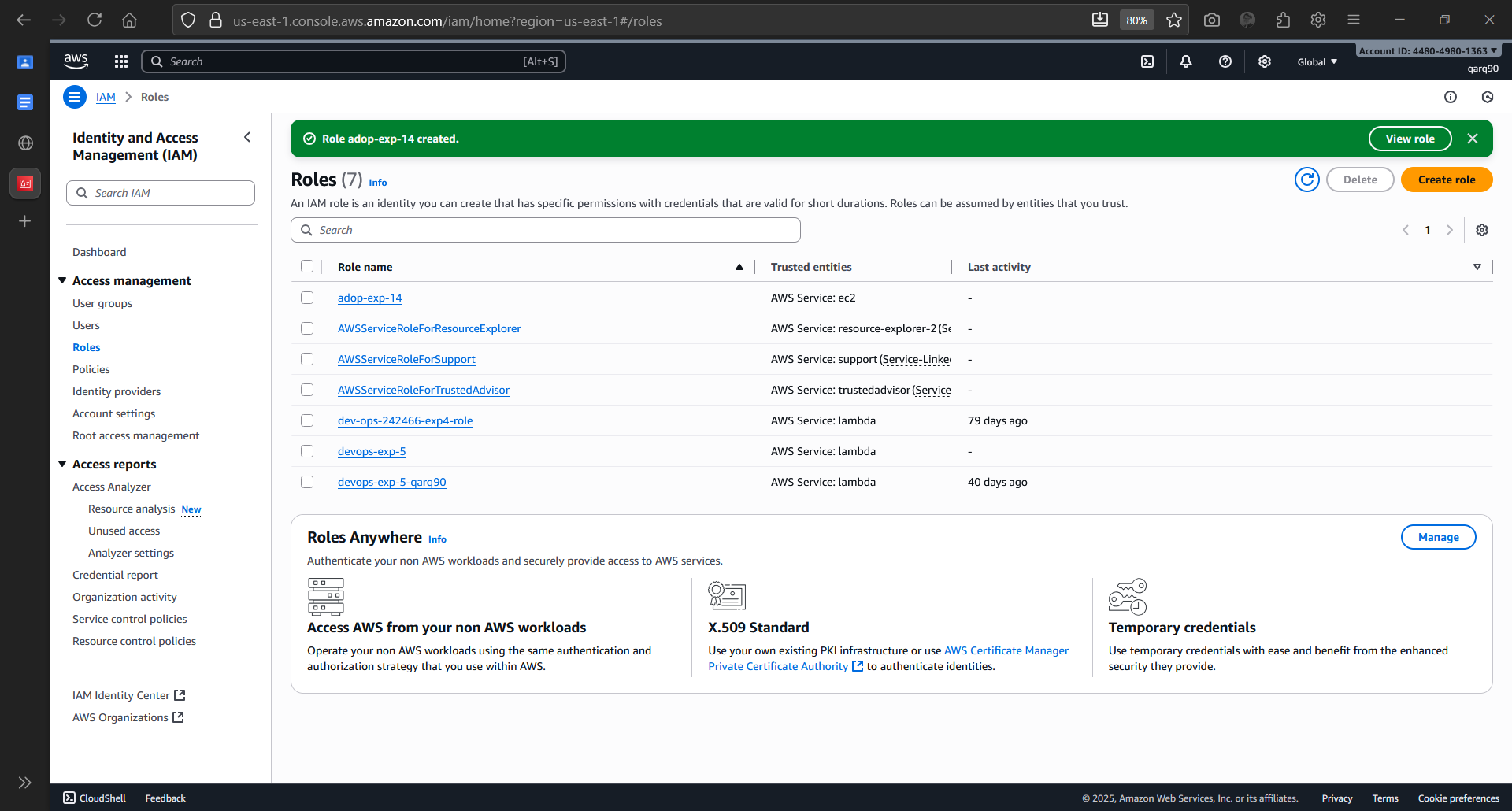
Adding Permission



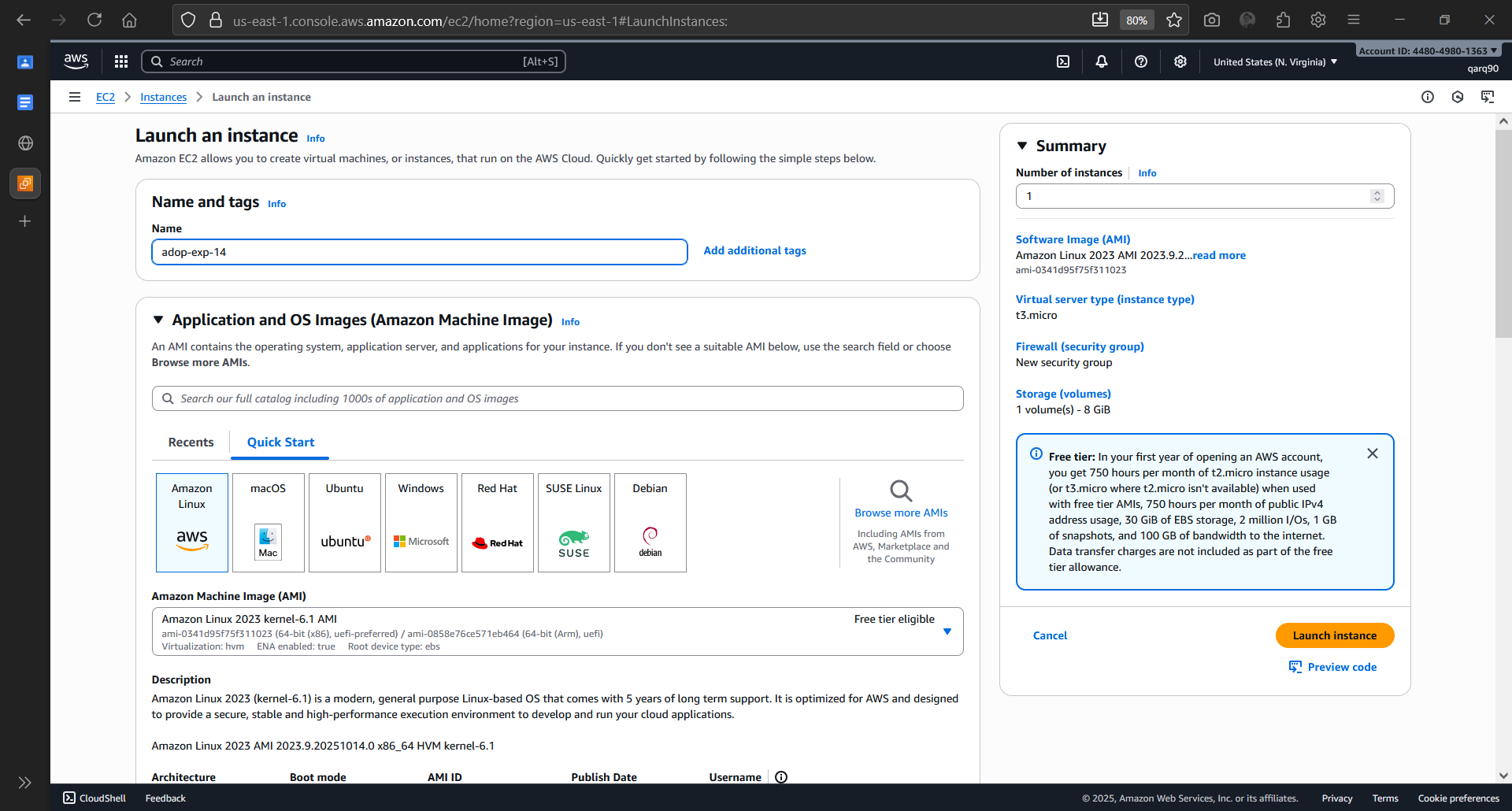
Setting Role Name



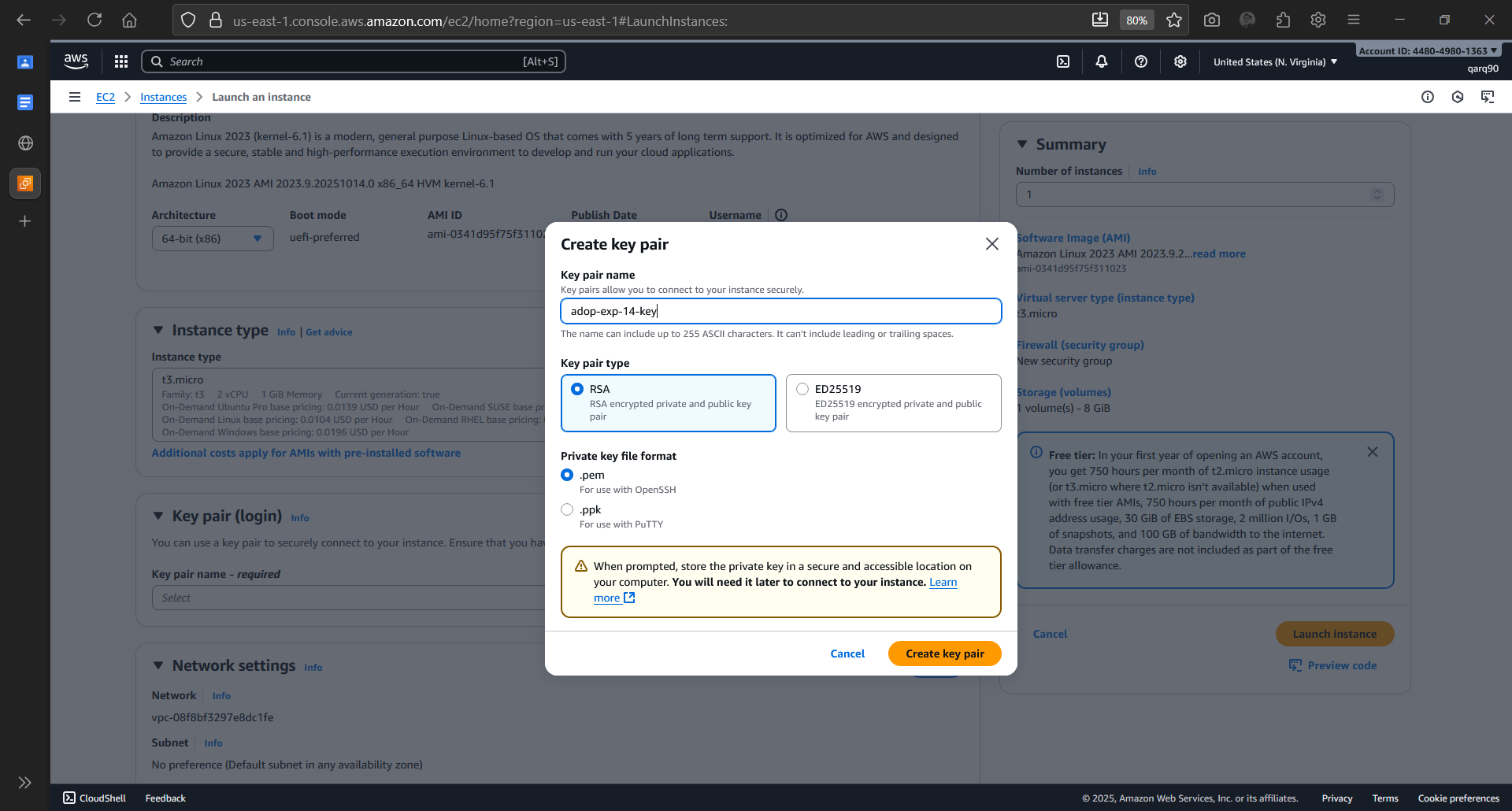
Role Configuration Complete



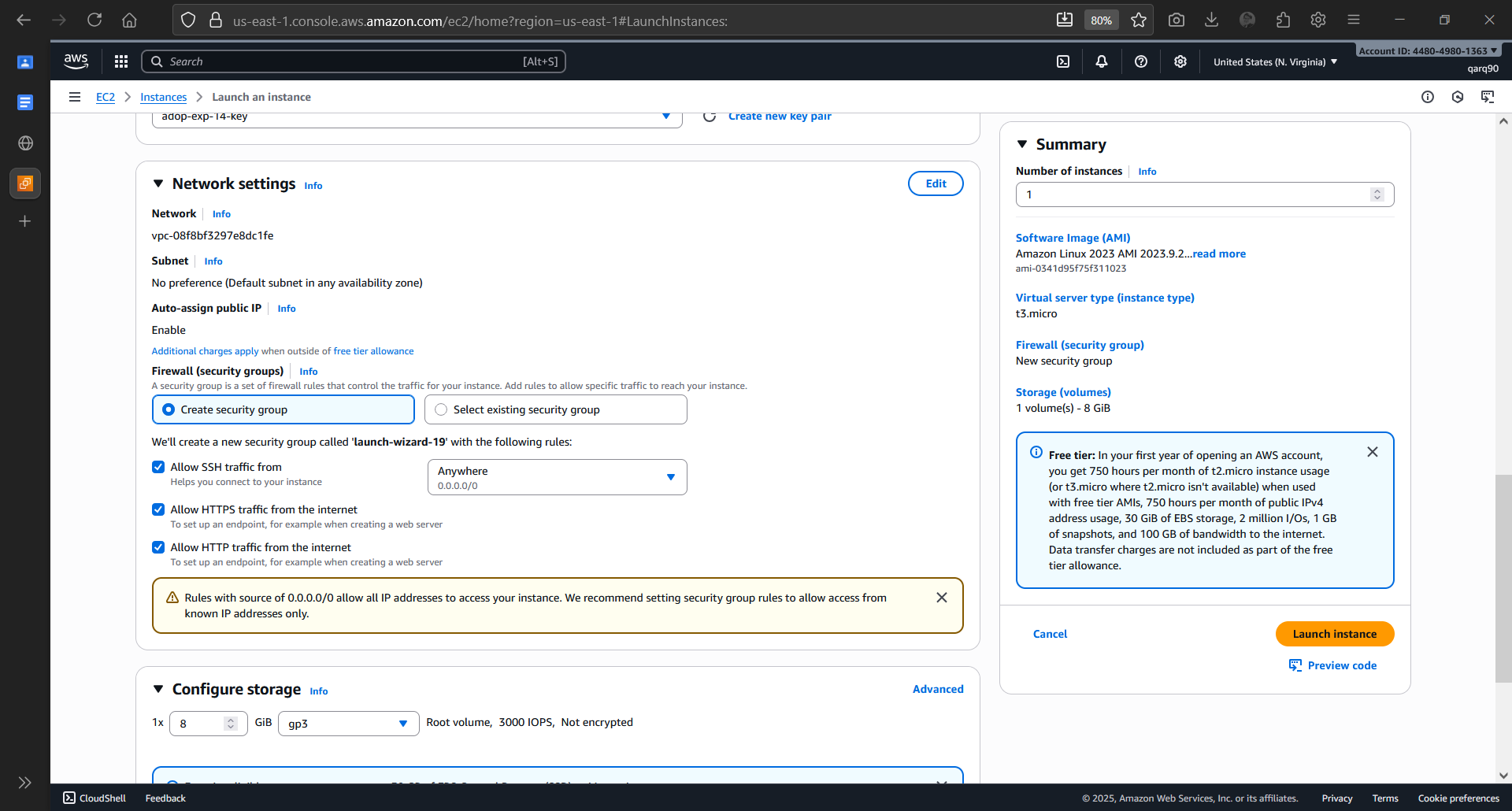
Role Created Successfully



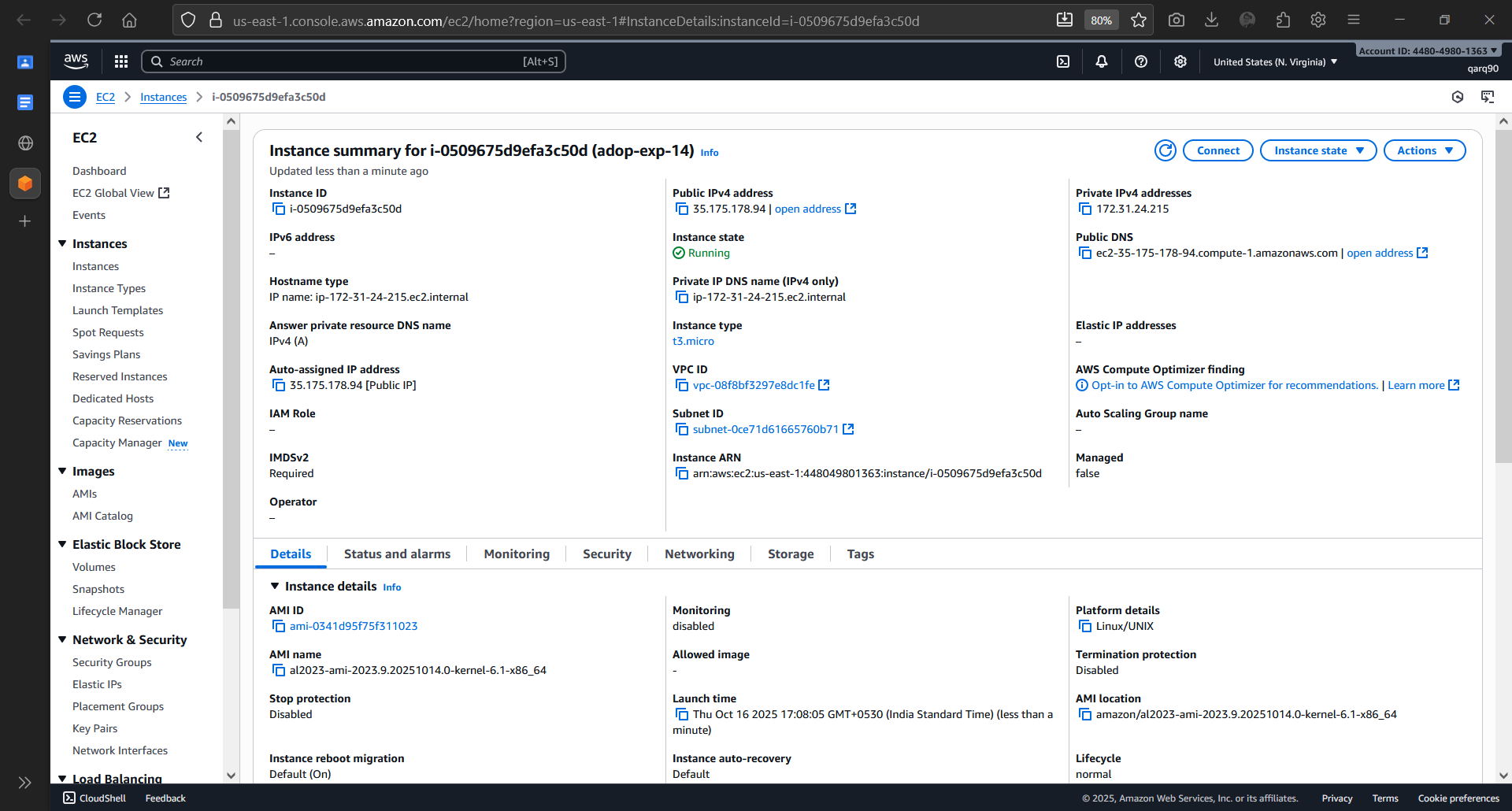
Creating an EC2 Instance



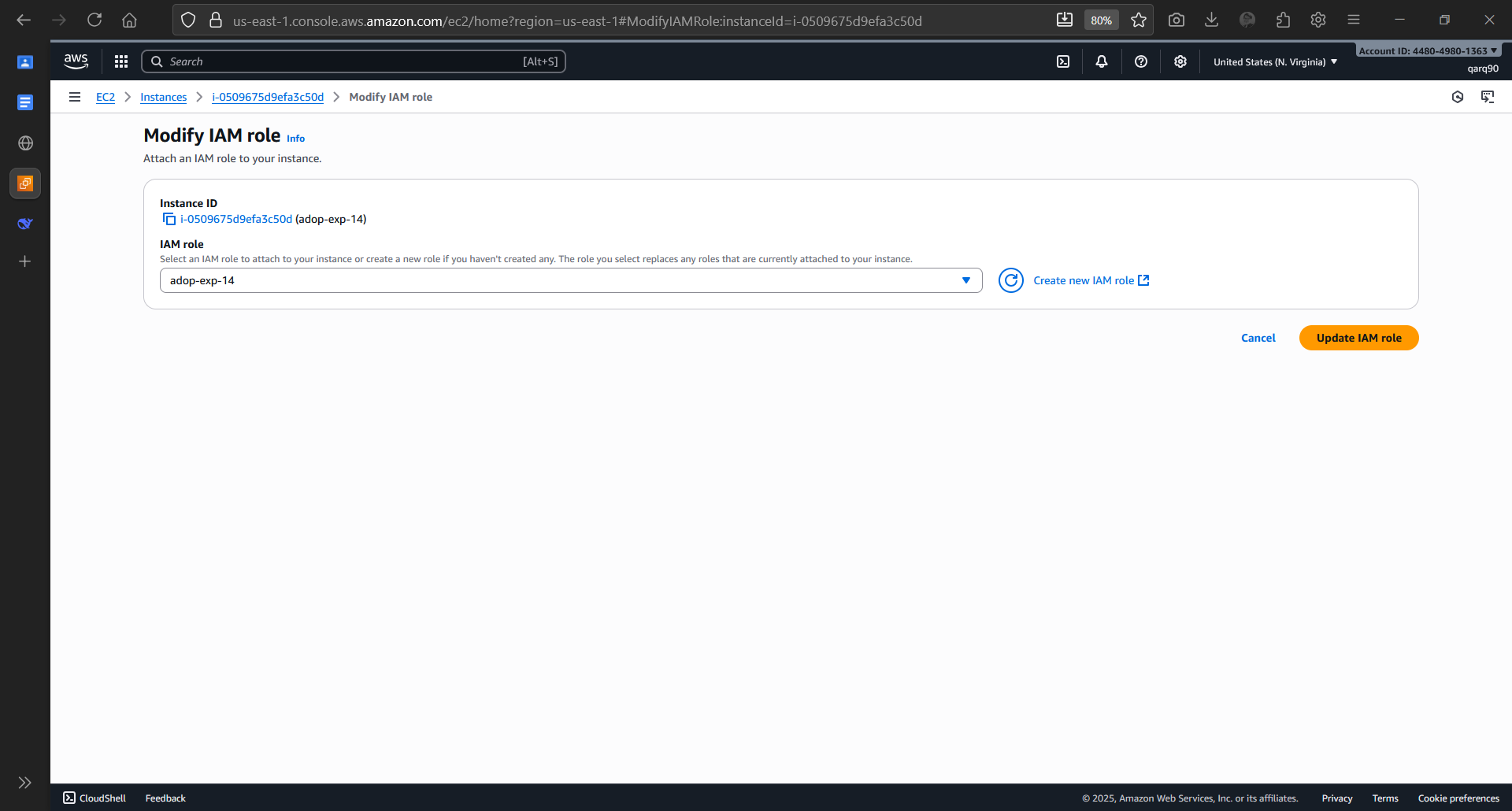
Creating Instance Key



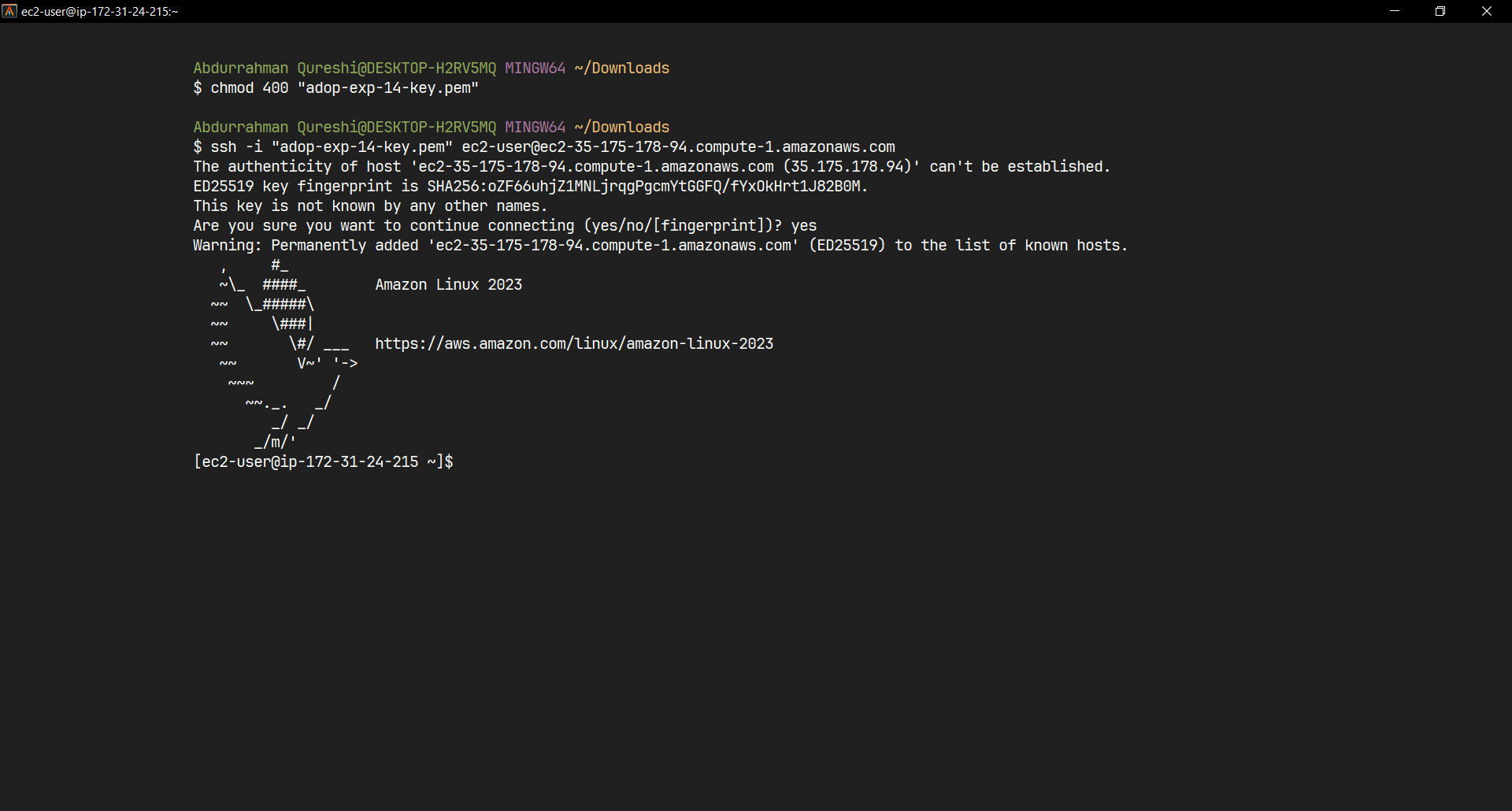
Configuring Instance Network Settings



Instance Details



Adding the IAM Role to the Instance



Connected to the Instance Successfully

variables.tf:

sudo vi variables.tf

variable "aws\_region" {

description = "The AWS region to create things in."

default = "us-east-1”

}

variable "key\_name" {

description = " SSH keys to connect to ec2 instance"

default = "[ Your PEM File Name without extension ]"

}

variable "instance\_type" {

description = "instance type for ec2"

default = "t2.micro"

}

variable "security\_group" {

description = "Name of security group"

default = "my-jenkins-security-group"

}

variable "tag\_name" {

description = "Tag Name of for Ec2 instance"

default = "my-ec2-instance"

}

variable "ami\_id" {

description = "AMI for Ubuntu Ec2 instance"

default = "[ Select instance from portal ] "

}

main.tf:

sudo vi main.tf

provider "aws" {

region = var.aws\_region

}

#Create security group with firewall rules

resource "aws\_security\_group" "security\_jenkins\_grp" {

name = var.security\_group

description = "security group for jenkins"

ingress {

from\_port = 8080

to\_port = 8080

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

# outbound from jenkis server

egress {

from\_port = 0

to\_port = 65535

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

tags= {

Name = var.security\_group

}

}

resource "aws\_instance" "myFirstInstance" {

ami = var.ami\_id

key\_name = var.key\_name

instance\_type = var.instance\_type

vpc\_security\_group\_ids = [aws\_security\_group.security\_jenkins\_grp.id]

tags= {

Name = var.tag\_name

}

}

# Create Elastic IP address

resource "aws\_eip" "myFirstInstance" {

domain = "vpc"

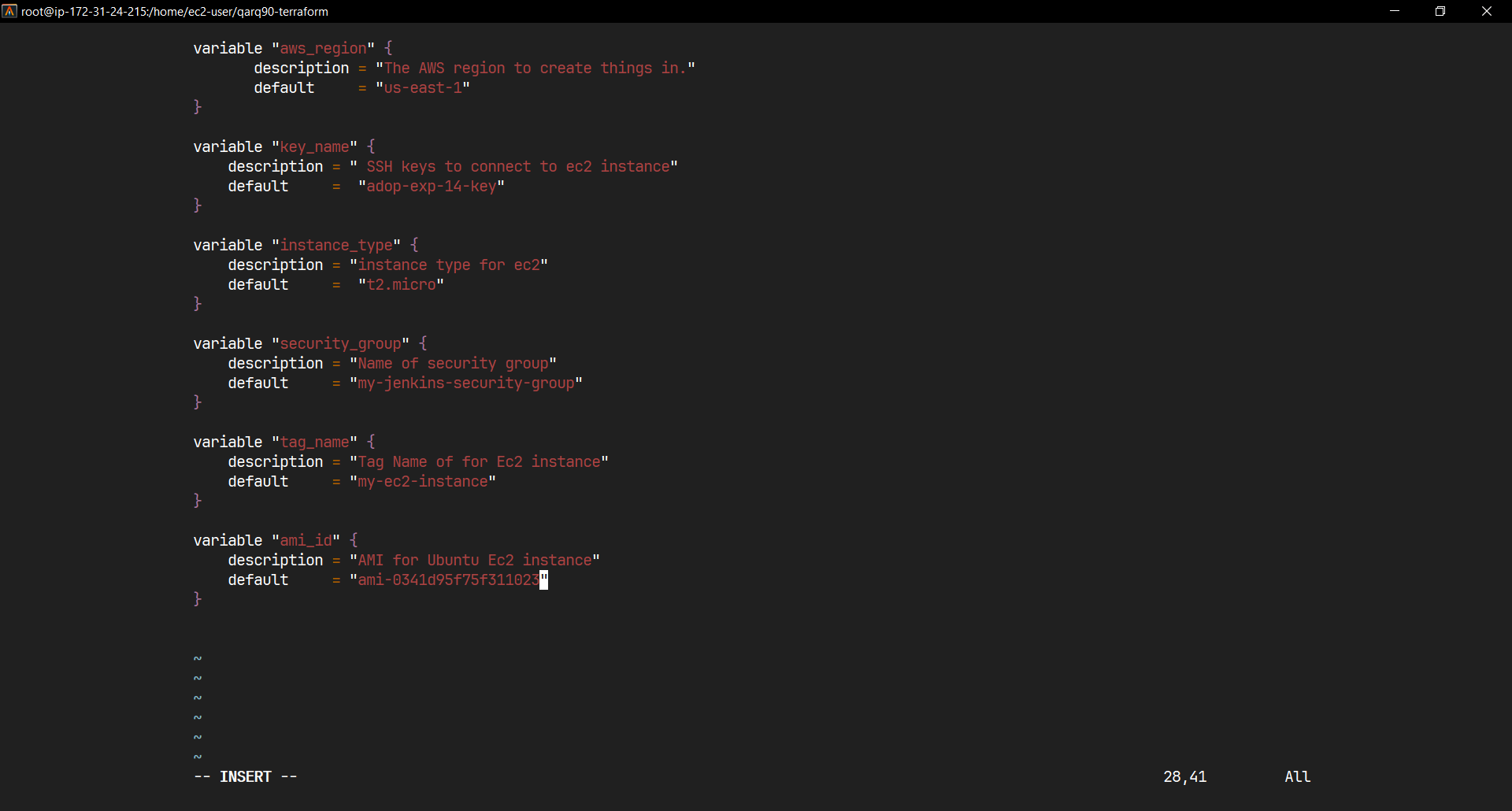
instance = aws\_instance.myFirstInstance.id

tags= {

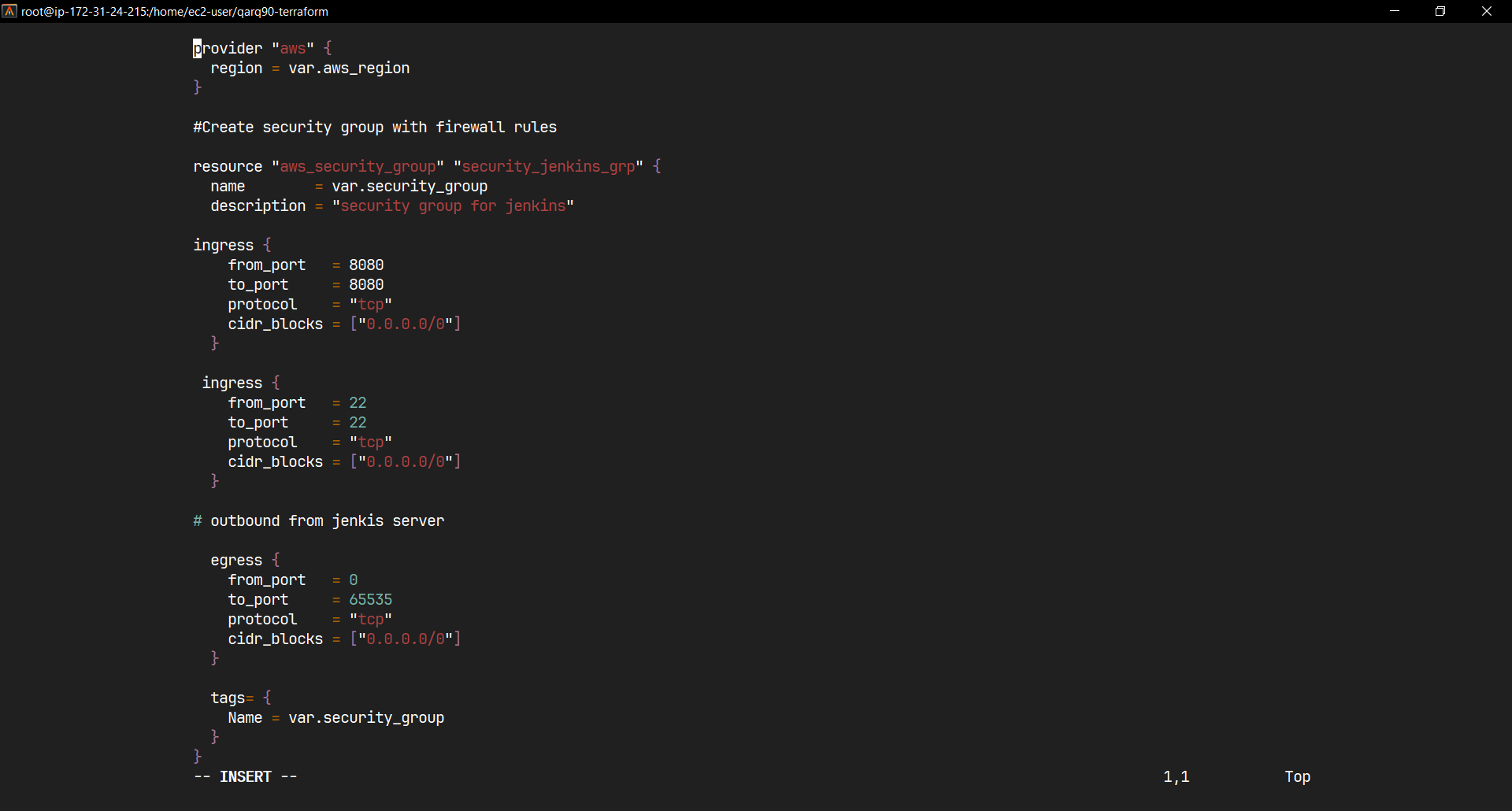
Name = "jenkins\_elastic\_ip"

}

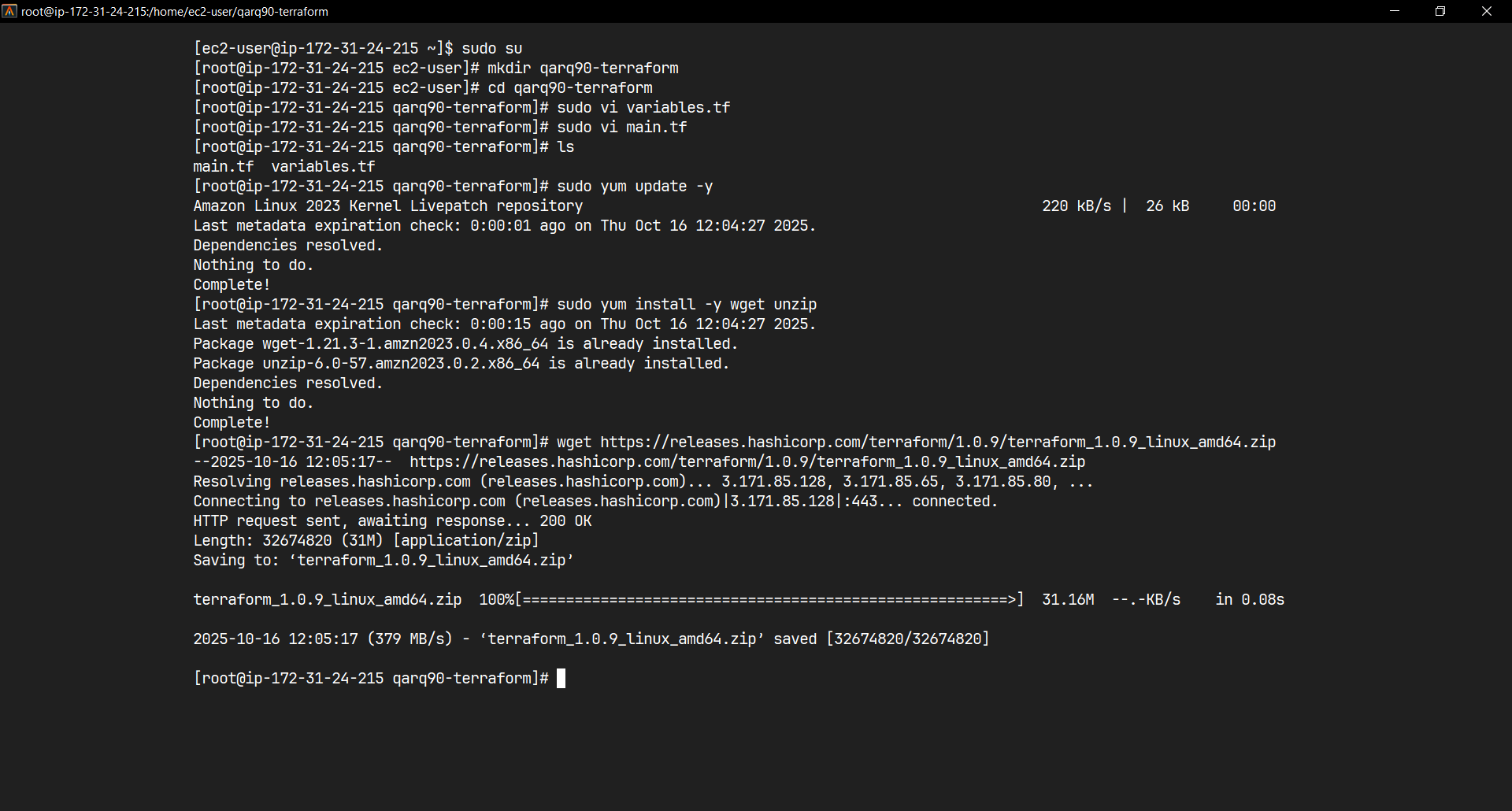
}



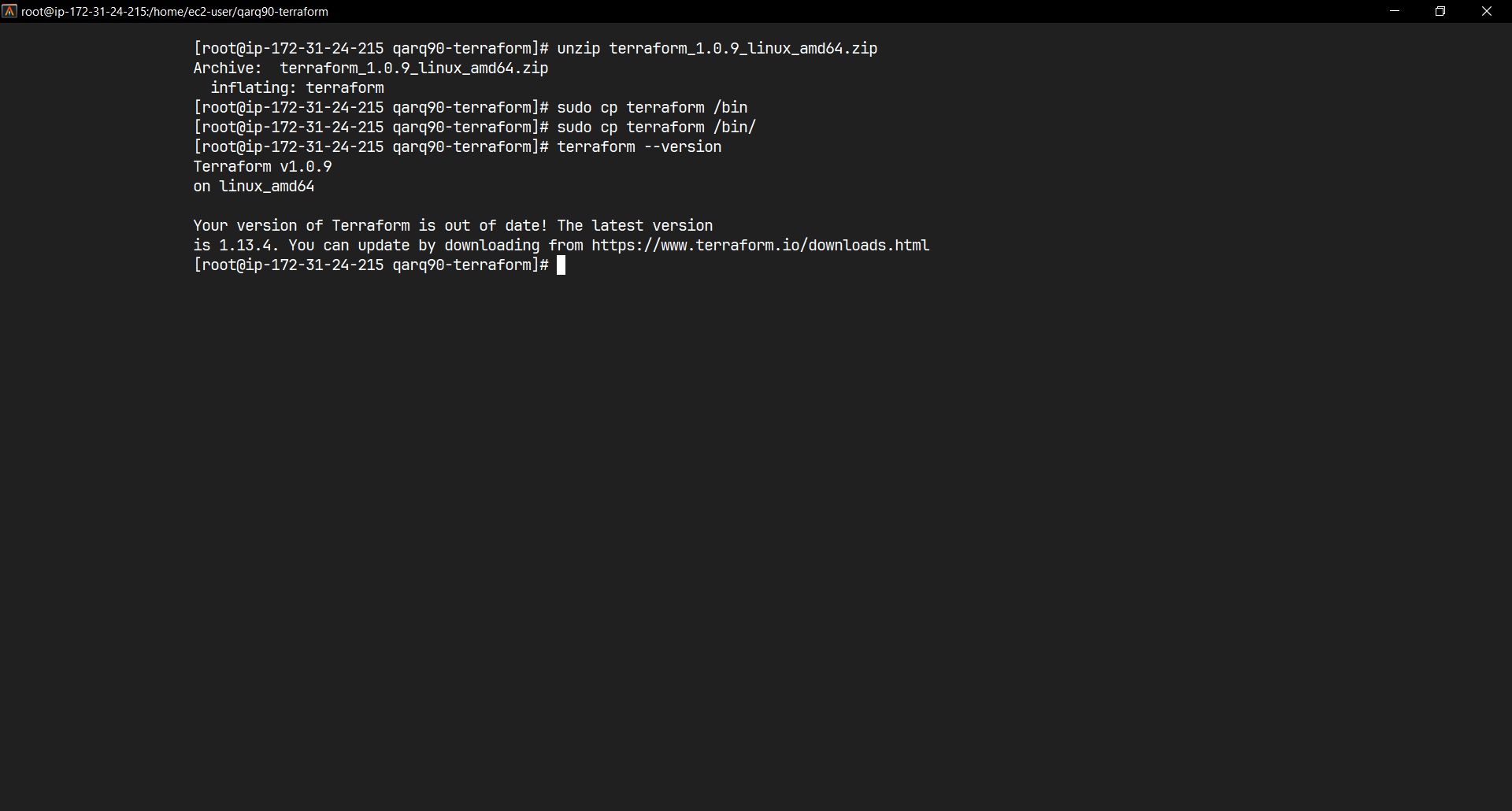
Variable.tf File



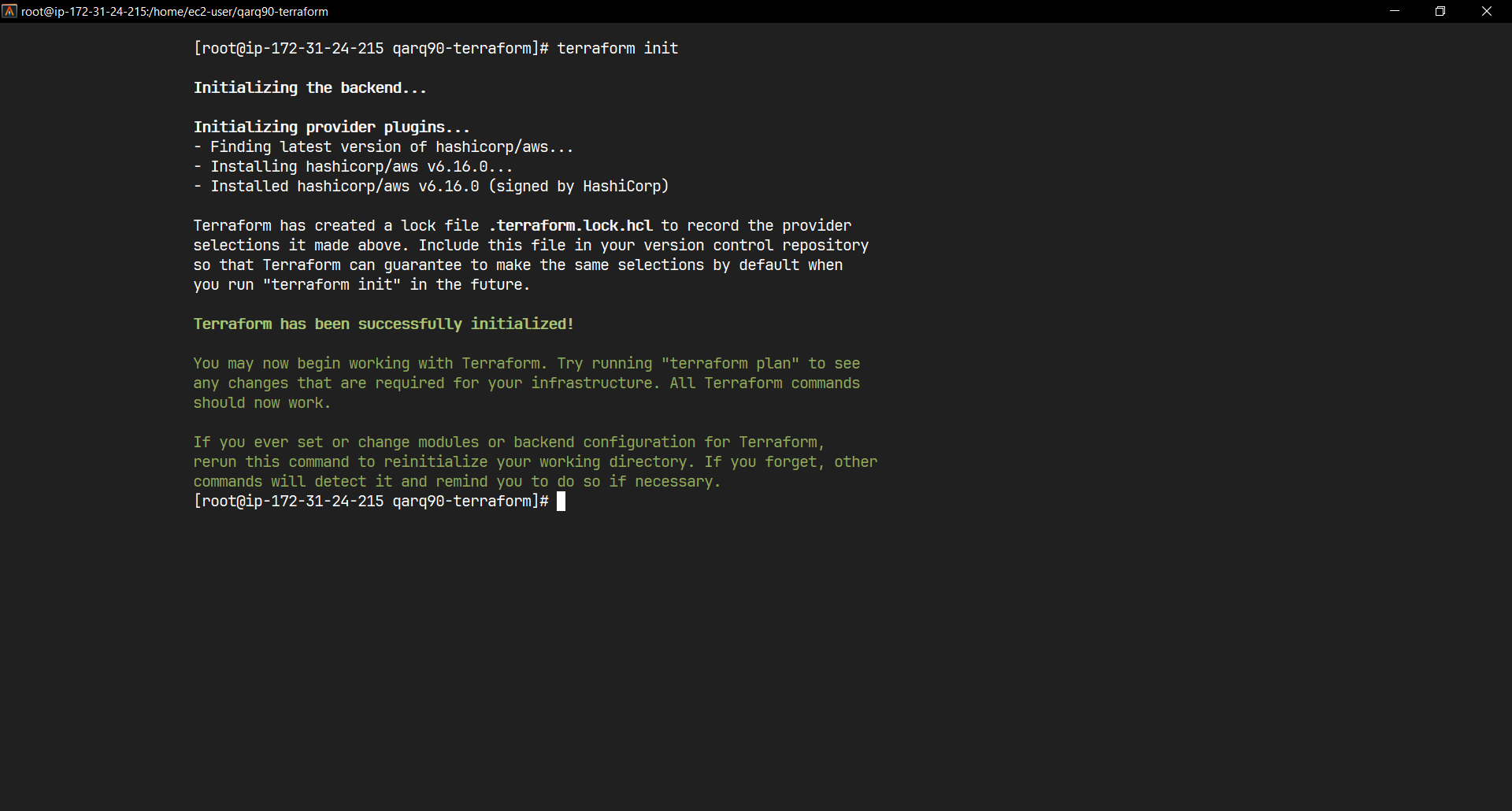
Main.tf



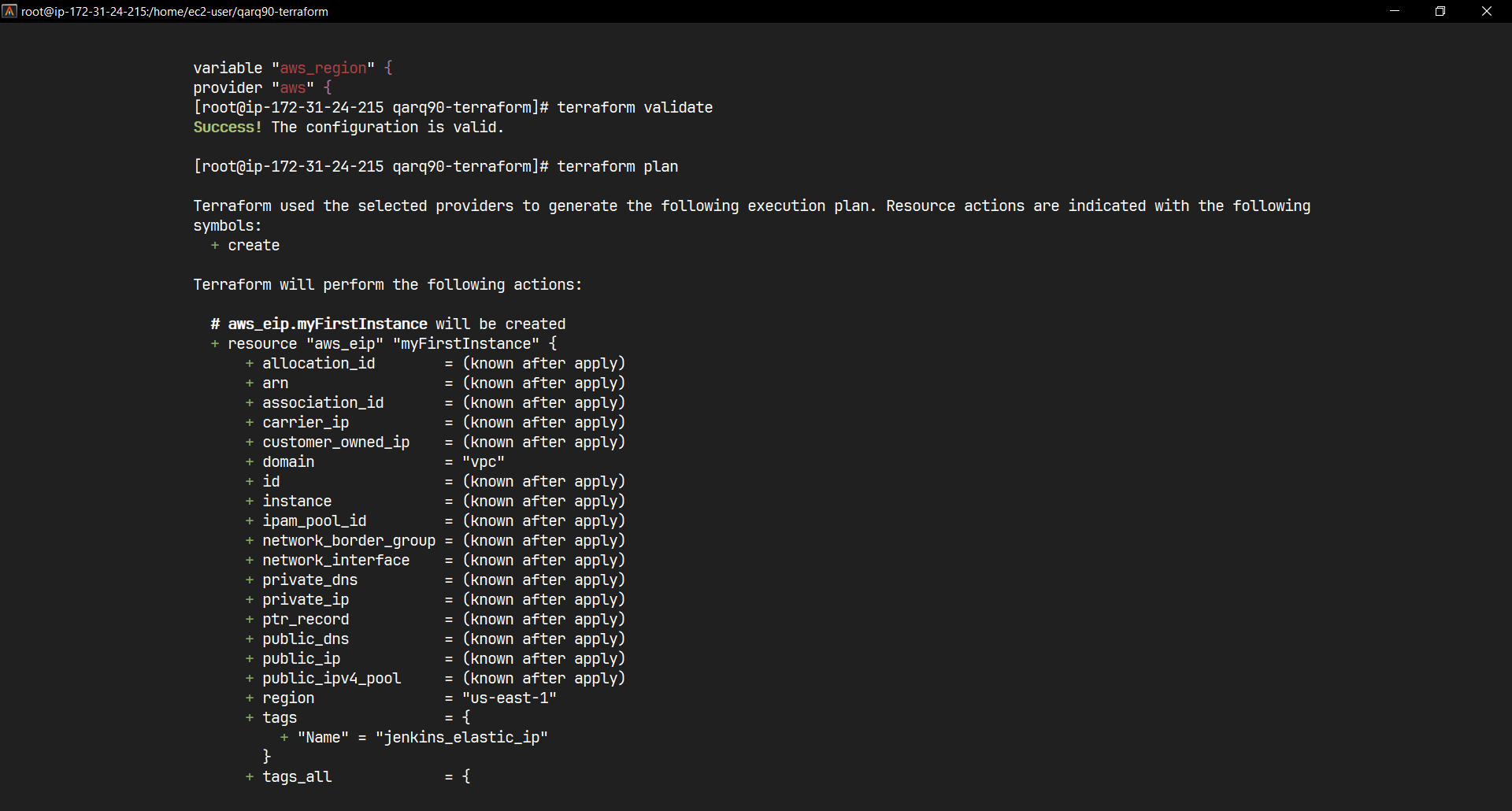
Installing Terrform

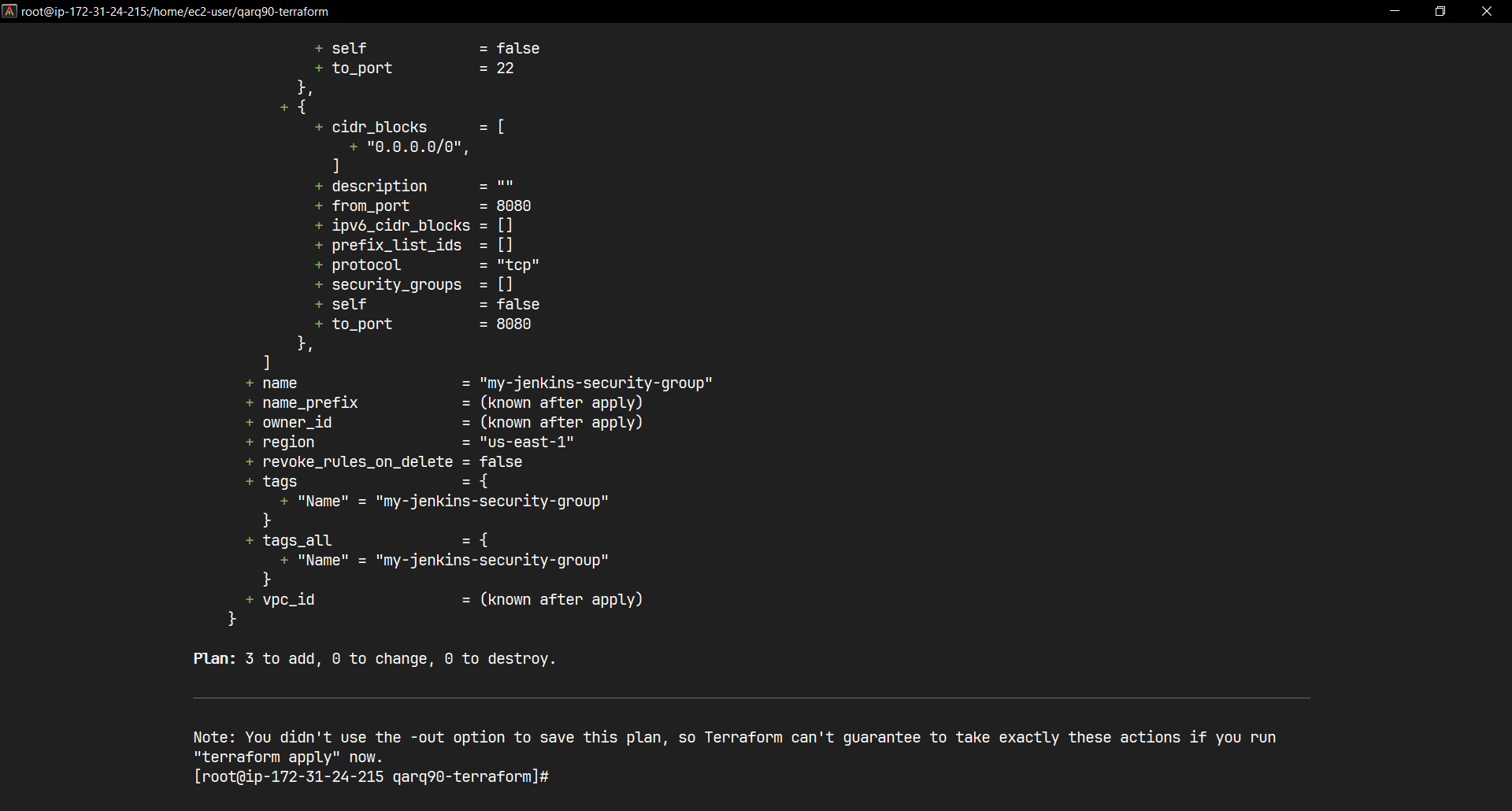


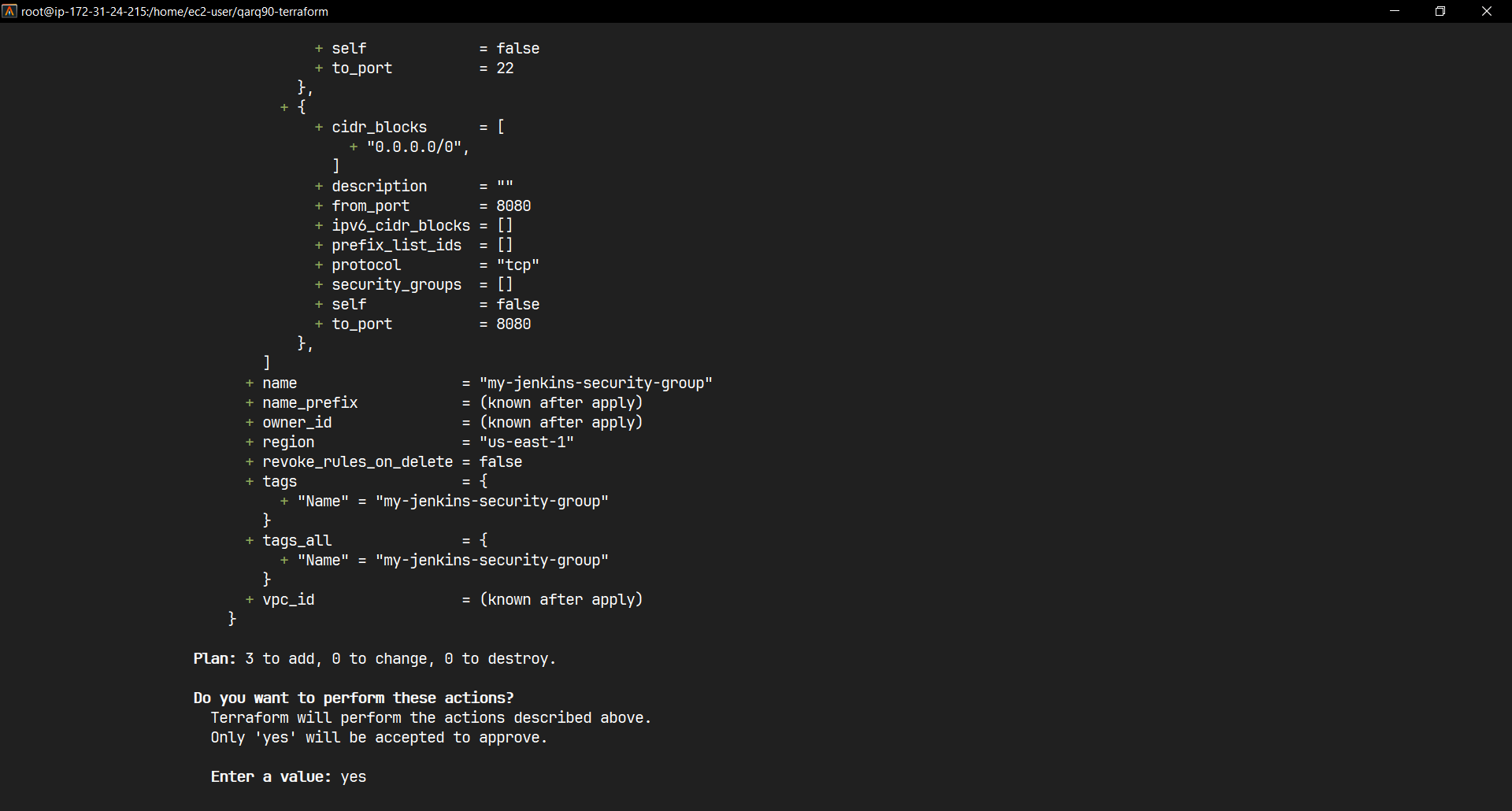
Installing Dependencies



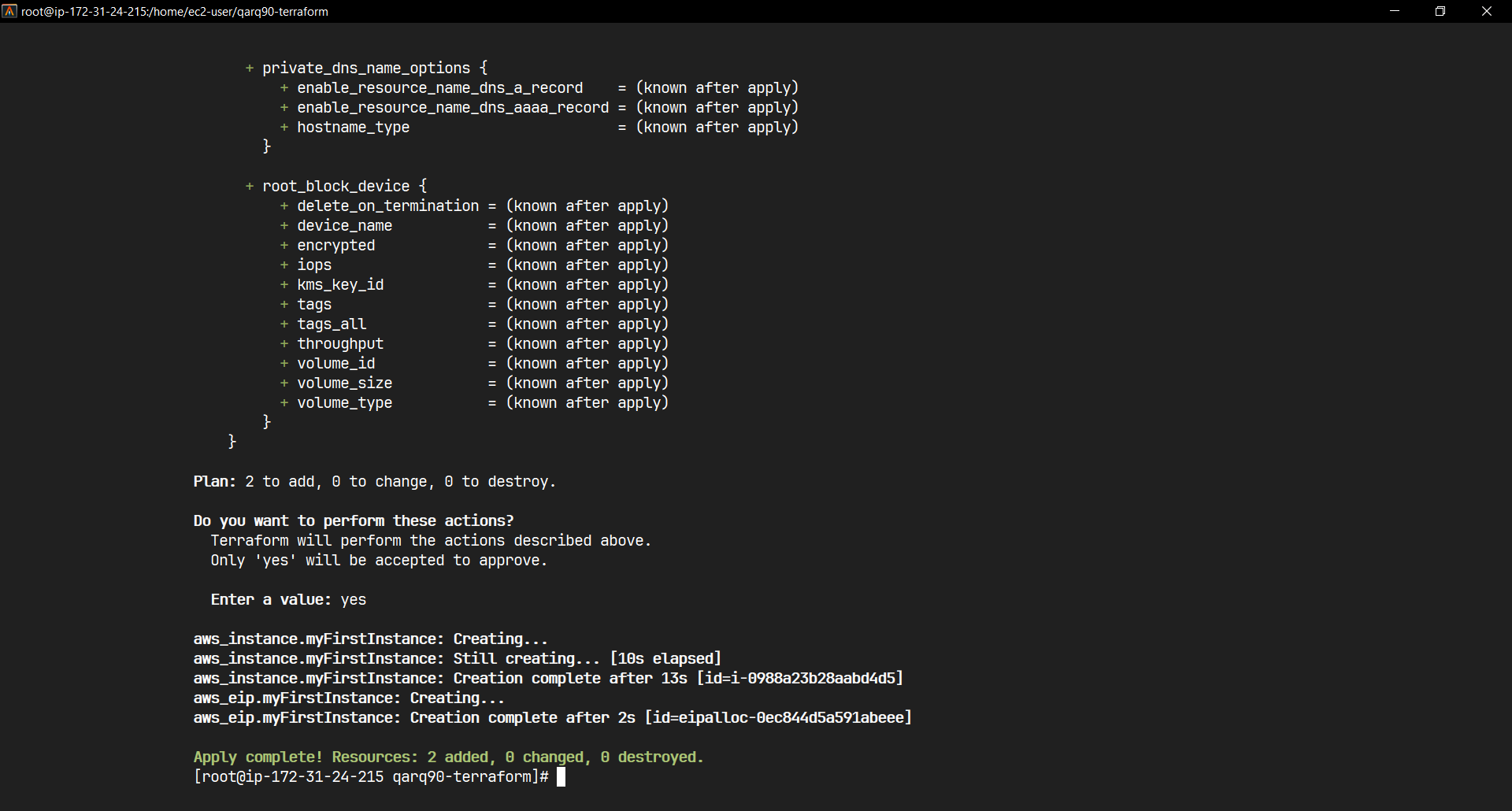
“terraform init” Ouput



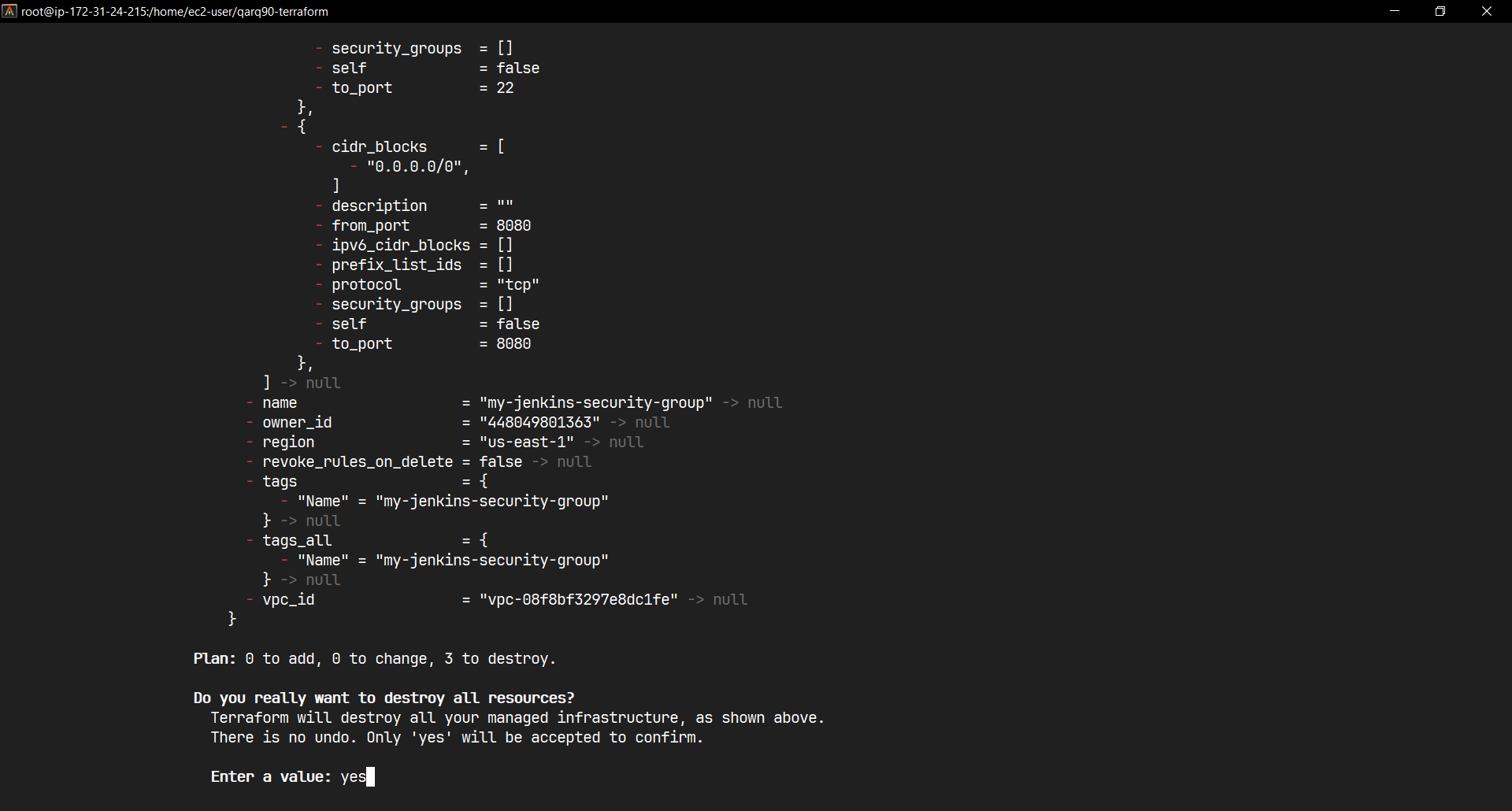


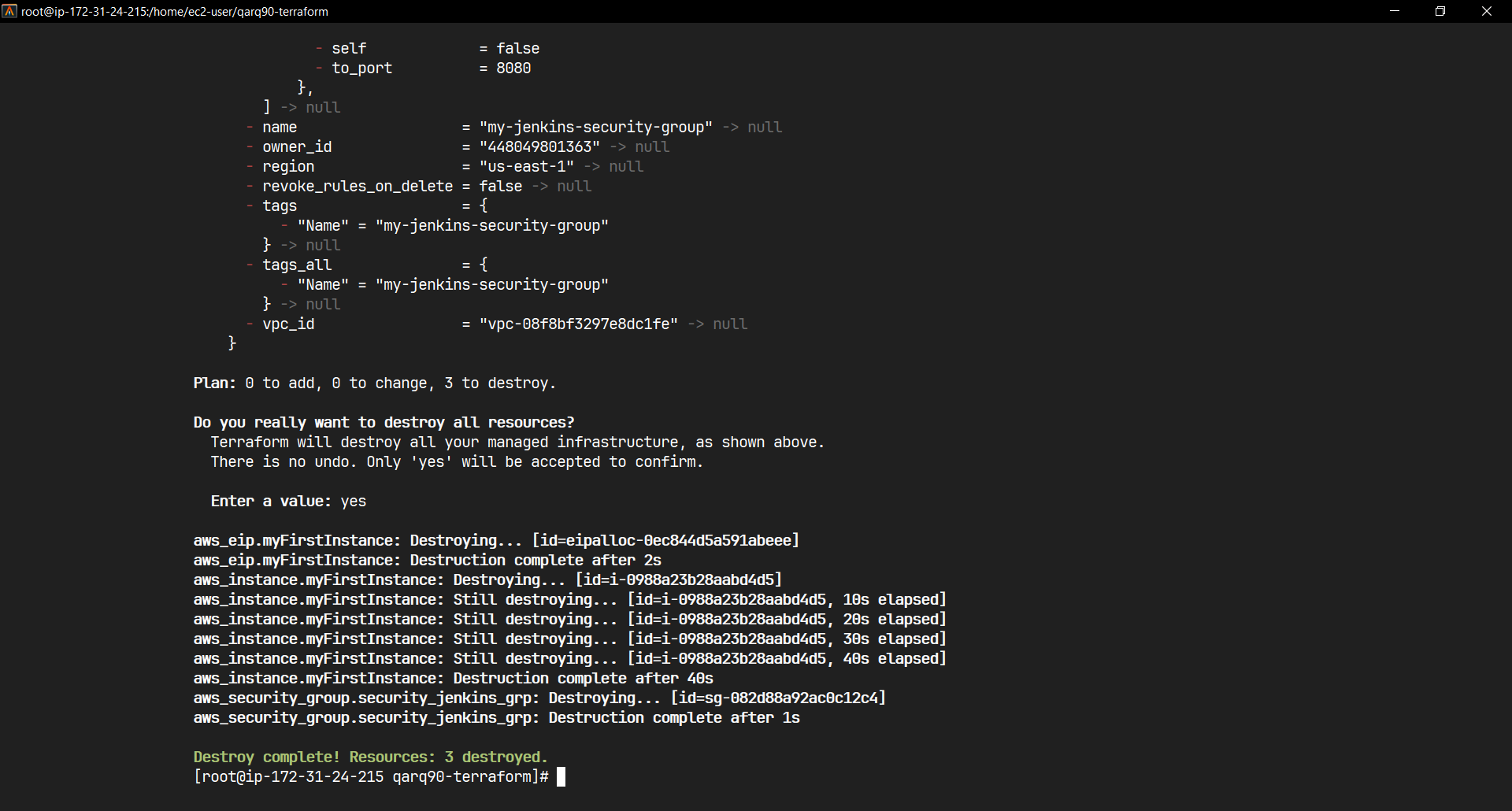


“terraform validate” and “terraform plan” Output

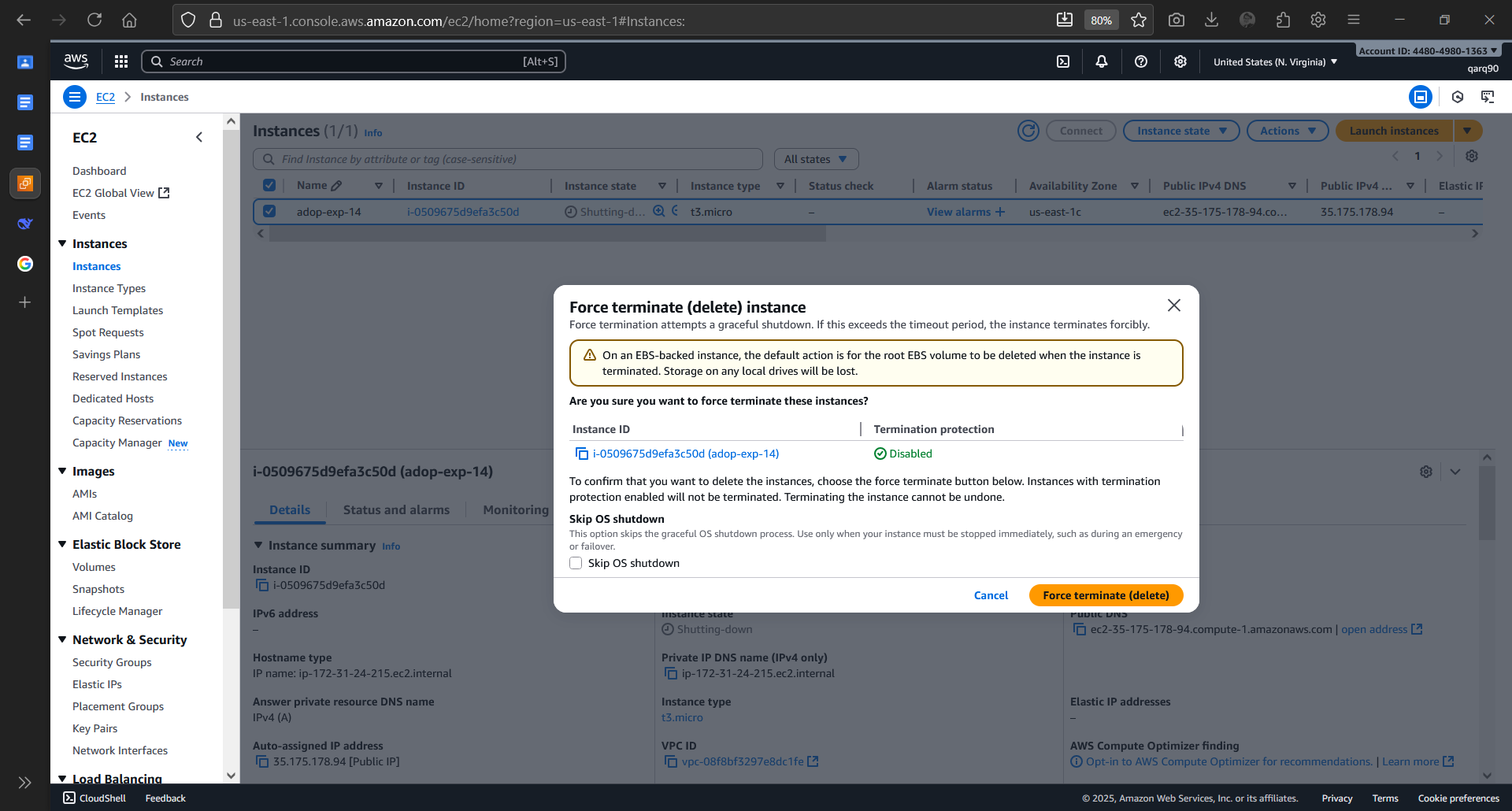


“terraform apply” Output

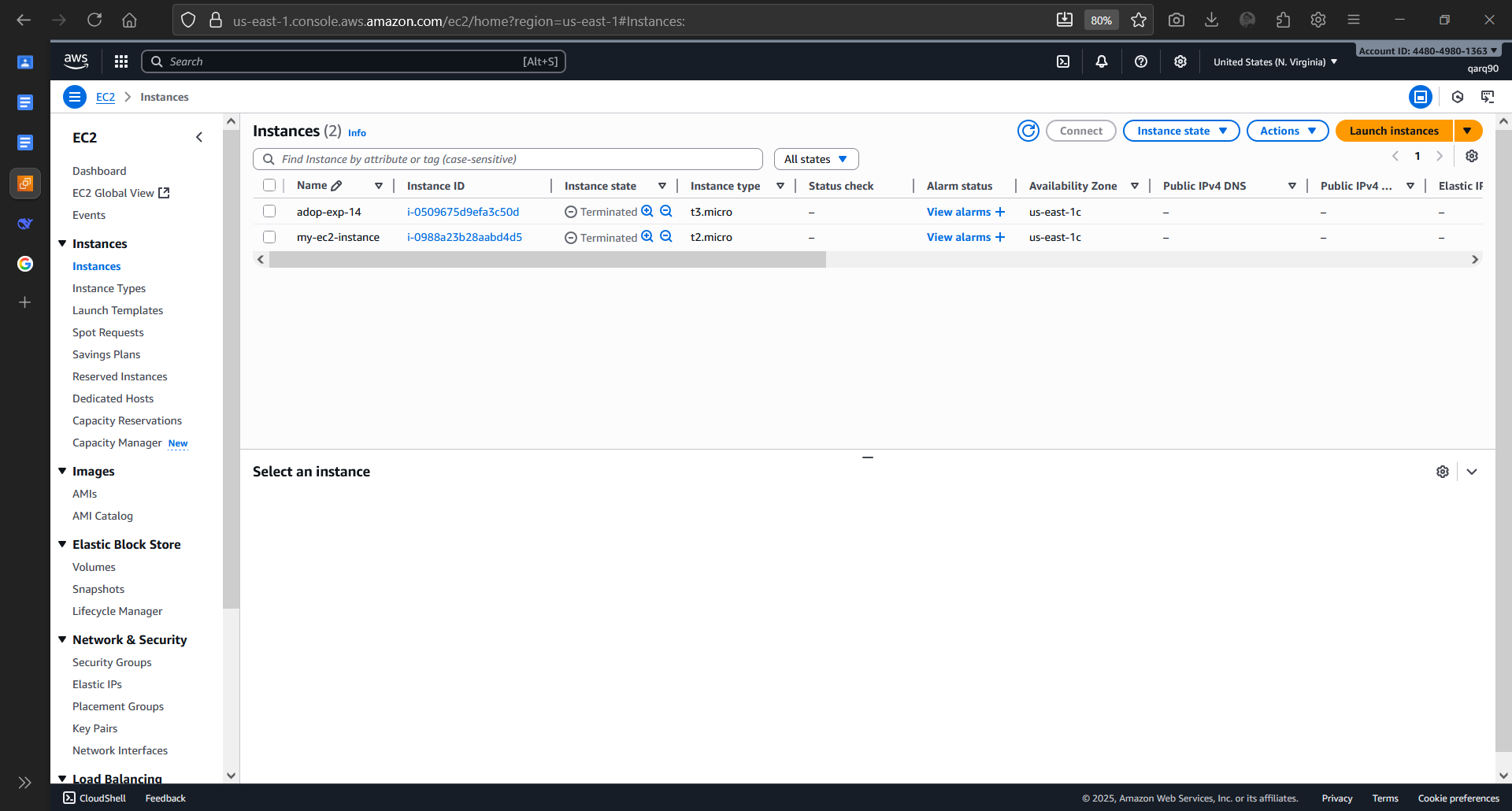




“terraform destroy” Output



Deleting Instance



Terminated the Instance