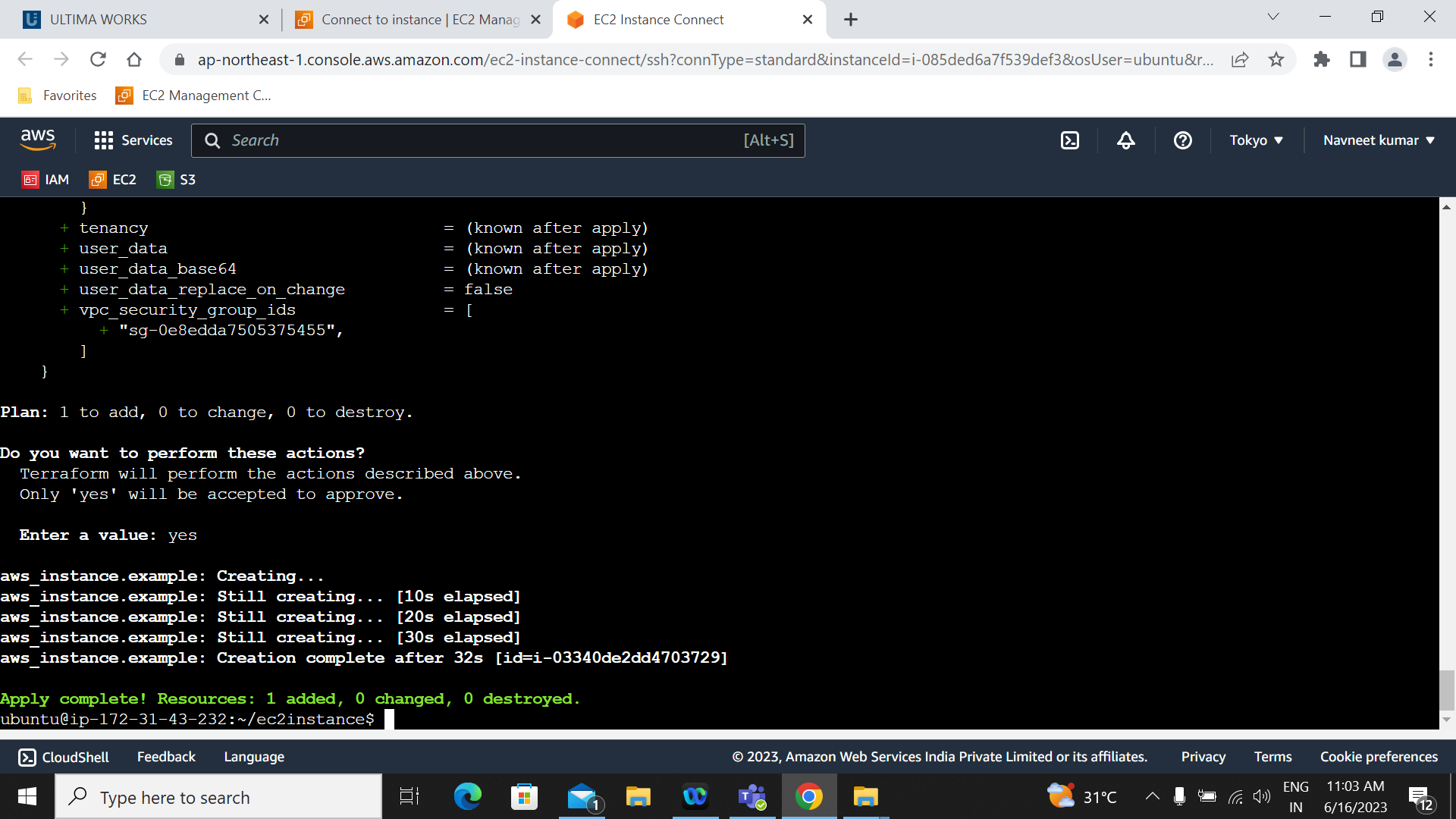
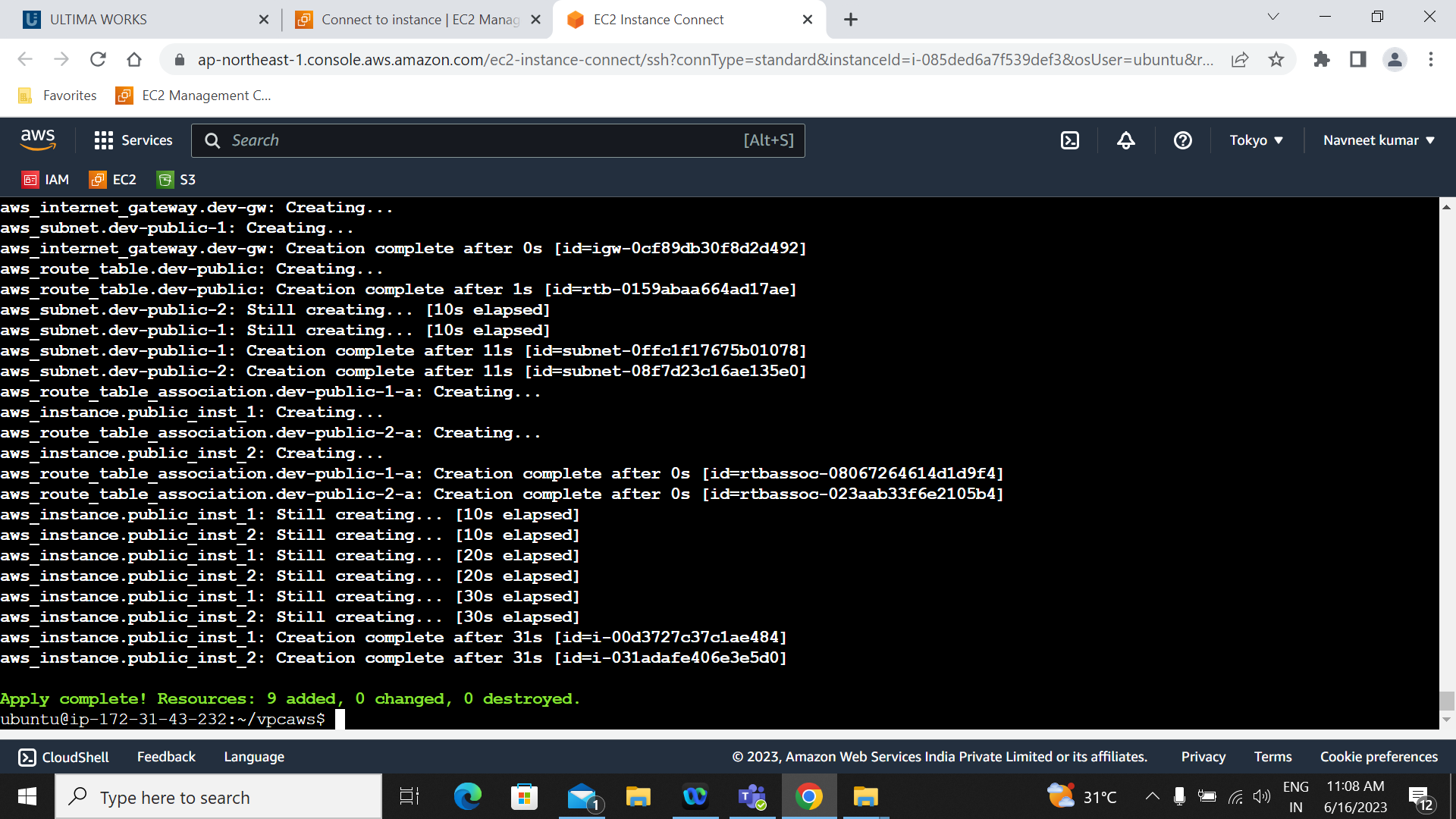
Terraform

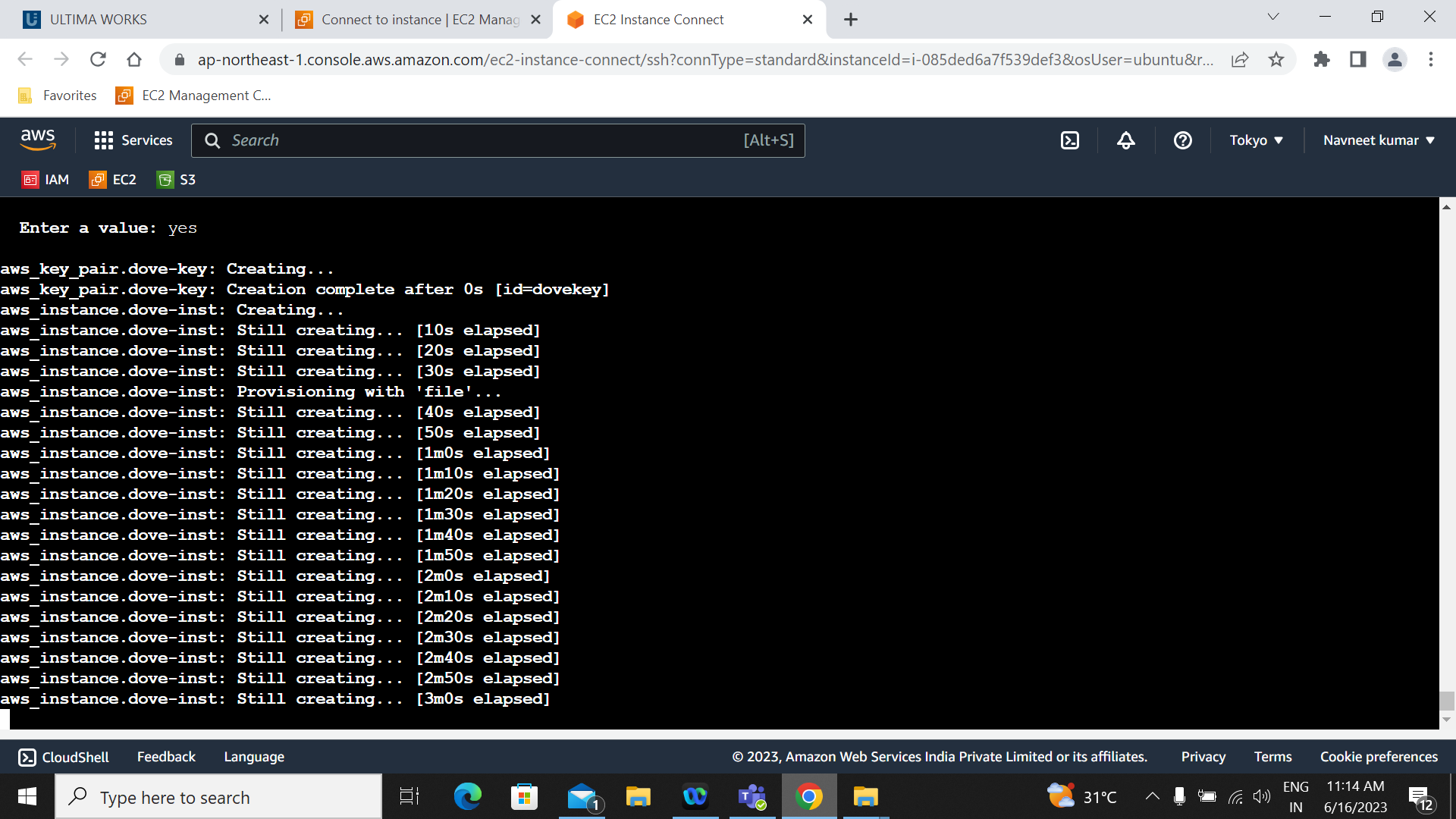
1. EC2 creation using terraform



2. VPC creation using terraform



3.Provisioning using terraform



4. Variables using terraform

A screenshot of a computer

Description automatically generated

########## Terraform ###############

step-1

create a EC2 instance i.e ubuntu

Step-2

connect EC2 using putty

step-3

Install terraform

1. Add HashiCorp GPG key

# sudo apt install curl

# curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -

2. Add Repository for Terraform

# sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb\_release -cs) main"

3.Run system update command

#sudo apt update

4.Install Terraform

#sudo apt-get install terraform

5. Check version

# terraform --version

step-4

Generate aws secret key and access key

#Open the IAM console.

#From the navigation menu, click Users.

#Select your IAM user name.

#Click User Actions, and then click Manage Access Keys.

#Click Create Access Key.

#Your keys will look something like this:

Access key ID example: AKIAIOSFODNN7EXAMPLE

Secret access key example: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

#Click Download Credentials, and store the keys in a secure location.

step-5

Attach a iam role with adminstrator full access ,ec2 and s3 with the created instance.

step-6

#create EC2 instance using terraform

\*\*ec2.tf file\*\*

provider "aws" {

region = "ap-northeast-1"

access\_key = "AKIA2GVOK3UKDLLJ673K"

secret\_key = "jJ8J3HpI2qU8cq3TBDfegUy6iIweN9PIQkBmZ2ZR"

}

resource "aws\_instance" "example" {

ami = "ami-0ed99df77a82560e6"

availability\_zone= "ap-northeast-1a"

key\_name = "terraform-1"

vpc\_security\_group\_ids= ["sg-0e8edda7505375455"]

instance\_type = "t2.micro"

tags={

Name="First-EC2"

}

}

- terraform init > terraform plan > terraform validate > terraform destroy

step-7

# create a vpc using terraform

\*\* aws\_vpc.tf \*\*

provider "aws" {

region = "ap-northeast-1"

access\_key = "AKIA2GVOK3UKDLLJ673K"

secret\_key = "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"

}

# Creating VPC,name, CIDR and Tags

resource "aws\_vpc" "dev" {

cidr\_block = "10.0.0.0/16"

instance\_tenancy= "default"

tags = {

Name = "dev"

}

}

# Creating Public Subnets in VPC

resource "aws\_subnet" "dev-public-1" {

vpc\_id = aws\_vpc.dev.id

cidr\_block = "10.0.1.0/24"

map\_public\_ip\_on\_launch = "true"

availability\_zone = "ap-northeast-1a"

tags = {

Name = "dev-public-1"

}

}

resource "aws\_subnet" "dev-public-2" {

vpc\_id = aws\_vpc.dev.id

cidr\_block = "10.0.2.0/24"

map\_public\_ip\_on\_launch = "true"

availability\_zone = "ap-northeast-1c"

tags = {

Name = "dev-public-2"

}

}

# Creating Internet Gateway in AWS VPC

resource "aws\_internet\_gateway" "dev-gw" {

vpc\_id = aws\_vpc.dev.id

tags = {

Name = "dev"

}

}

# Creating Route Tables for Internet gateway

resource "aws\_route\_table" "dev-public" {

vpc\_id = aws\_vpc.dev.id

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_internet\_gateway.dev-gw.id

}

tags = {

Name = "dev-public-1"

}

}

# Creating Route Associations public subnets

resource "aws\_route\_table\_association" "dev-public-1-a" {

subnet\_id = aws\_subnet.dev-public-1.id

route\_table\_id = aws\_route\_table.dev-public.id

}

resource "aws\_route\_table\_association" "dev-public-2-a" {

subnet\_id = aws\_subnet.dev-public-2.id

route\_table\_id = aws\_route\_table.dev-public.id

}

# Creating EC2 instances in public subnets

resource "aws\_instance" "public\_inst\_1" {

ami = "ami-0ed99df77a82560e6"

instance\_type = "t2.micro"

subnet\_id = "${aws\_subnet.dev-public-1.id}"

key\_name = "terraform-1"

tags = {

Name = "public\_inst\_1"

}

}

resource "aws\_instance" "public\_inst\_2" {

ami = "ami-0ed99df77a82560e6"

instance\_type = "t2.micro"

subnet\_id = "${aws\_subnet.dev-public-2.id}"

key\_name = "terraform-1"

tags = {

Name = "public\_inst\_2"

}

}

step-8

Handson on provisioning and variables using terraform