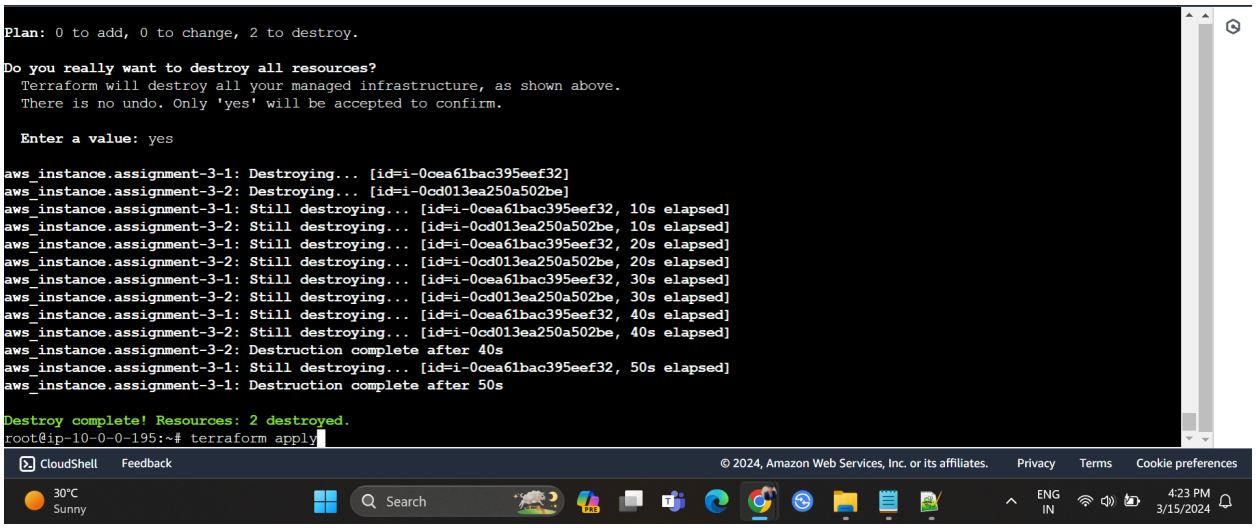
Terraform Assignment - 4

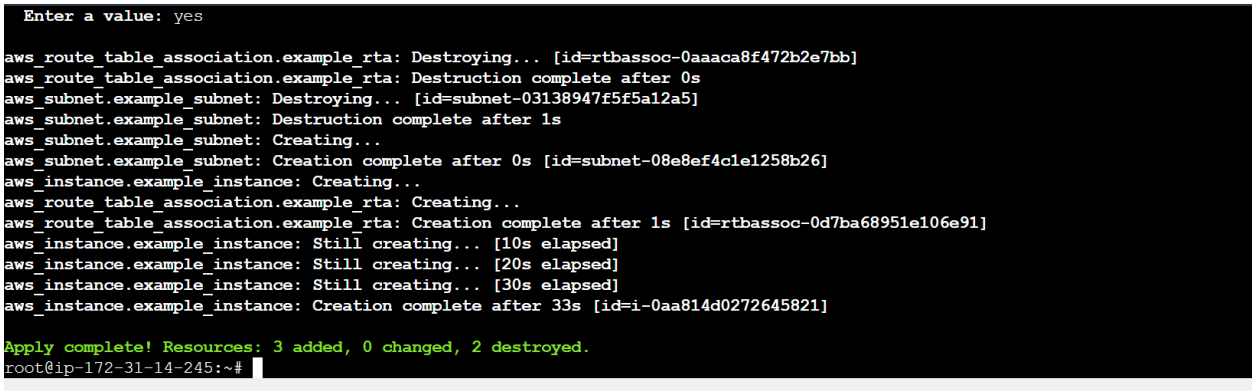
You have been asked to:

● Destroy the previous deployments



● Create a VPC with the required components using Terraform

● Deploy an EC2 instance inside the VPC



Code:

```
provider "aws" {
  region = "us-west-2"
  access_key = "AKIAVNPLXOGUAGQNP6LT"
  secret_key = "5J1CWJvqwfQh50sT5u789ndFvmWc+X2bIIf96neJ"
```

```
}

resource "aws_vpc" "example_vpc" {
  cidr_block = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true

  tags = {
    Name = "example-vpc"
  }
}

resource "aws_subnet" "example_subnet" {
  vpc_id     = aws_vpc.example_vpc.id
  cidr_block = "10.0.1.0/24"
availability_zone = "us-west-2a"
  tags = {
    Name = "example-subnet"
  }
}

resource "aws_internet_gateway" "example_igw" {
  vpc_id = aws_vpc.example_vpc.id

  tags = {
    Name = "example-igw"
  }
}

resource "aws_route_table" "example_rt" {
  vpc_id = aws_vpc.example_vpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.example_igw.id
  }

  tags = {
    Name = "example-rt"
  }
}

resource "aws_route_table_association" "example_rta" {
  subnet_id      = aws_subnet.example_subnet.id
```

```
  route_table_id = aws_route_table.example_rt.id
}

resource "aws_security_group" "example_sg" {
  name        = "example-sg"
  description = "Allow SSH inbound traffic"
  vpc_id      = aws_vpc.example_vpc.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "example-sg"
  }
}

resource "aws_instance" "example_instance" {
  ami           = "ami-08f7912c15ca96832"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.example_subnet.id
vpc_security_group_ids = [aws_security_group.example_sg.id]
  tags = {
    Name = "example-instance"
  }
}
```

Code 2: Shorter

```
provider "aws" {
  region = "us-west-2"
  access_key = "AKIAVNPLXOGUAGQNP6LT"
  secret_key = "5J1CWJvqwfQh50sT5u789ndFvmWc+X2bIIf96neJ"
}
```

```
resource "aws_vpc" "example_vpc" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "example-vpc"
  }
}
resource "aws_subnet" "example_subnet" {
  vpc_id     = aws_vpc.example_vpc.id
  cidr_block = "10.0.1.0/24"
availability_zone = "us-west-2a"
  tags = {
    Name = "example-subnet"
  }
}

resource "aws_instance" "example_instance" {
  ami           = "ami-08f7912c15ca96832"
  instance_type = "t2.micro"
  key_name = "WOW"
  subnet_id     = aws_subnet.example_subnet.id
  tags = {
    Name = "example-instance"
  }
}
```

Code 3: more shorter

```
provider "aws" {
  region = "us-west-2"
  access_key = "AKIAVNPLXOGUAGQNP6LT"
  secret_key = "5J1CWJvqwfQh50sT5u789ndFvmWc+X2bIIf96neJ"
}
resource "aws_vpc" "example_vpc" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "example-vpc"
  }
}

resource "aws_instance" "example_instance" {
  ami           = "ami-08f7912c15ca96832"
  instance_type = "t2.micro"
  key_name = "WOW"
```

```
  subnet_id     = aws_subnet.example_subnet.id
  tags = {
    Name = "example-instance"
  }
}
```