TERRAFORM-TEMPLATE-TASK

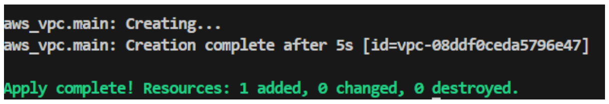
1) Create VPC

>>CREATE A DIRECTORY WITH NAME terraform-x **AND ADD A FILE** main.tf **THEN ADD THE TEMPLATE**

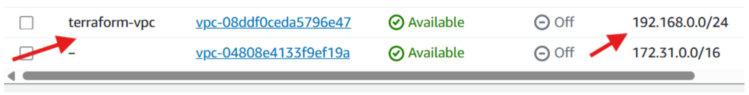
```
provider "aws" {
  region = "us-east-2"
}
resource "aws_vpc" "main" {
  cidr_block = "192.168.0.0/24"
  tags = {
    Name = "terraform-vpc"
  }
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply **<check on console if its created>**



```
aws_vpc.main: Creating...
aws_vpc.main: Creation complete after 5s [id=vpc-08ddf0ceda5796e47]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



☐	terraform-vpc	vpc-08ddf0ceda5796e47	Available	Off	192.168.0.0/24
☐	–	vpc-04808e4133f9ef19a	Available	Off	172.31.0.0/16

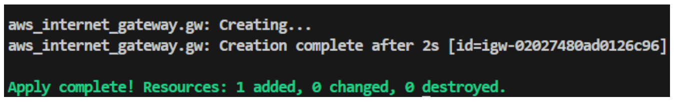
2) Create Internet Gateway

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

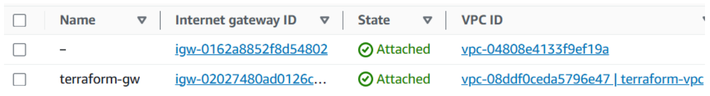
```
resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id
tags = {
    Name = "terraform-gw"
  }
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply **<check on console if its created>**



```
aws_internet_gateway.gw: Creating...
aws_internet_gateway.gw: Creation complete after 2s [id=igw-02027480ad0126c96]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



☐	Name	Internet gateway ID	State	VPC ID
☐	–	igw-0162a8852f8d54802	Attached	vpc-04808e4133f9ef19a
☐	terraform-gw	igw-02027480ad0126c…	Attached	vpc-08ddf0ceda5796e47 | terraform-vpc

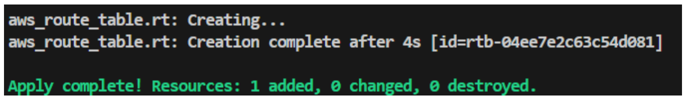
3) Create Custom Route Table

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

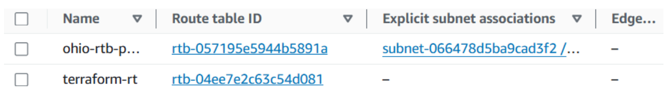
```
resource "aws_route_table" "rt" {
  vpc_id = aws_vpc.main.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }
tags = {
    Name = "terraform-rt"
  }
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply **<check on console if its created>**



```
aws_route_table.rt: Creating...
aws_route_table.rt: Creation complete after 4s [id=rtb-04ee7e2c63c54d081]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



	Name	Route table ID	Explicit subnet associations	Edge...
☐	ohio-rtb-p...	rtb-057195e5944b5891a	subnet-066478d5ba9cad3f2 /...	–
☐	terraform-rt	rtb-04ee7e2c63c54d081	–	–

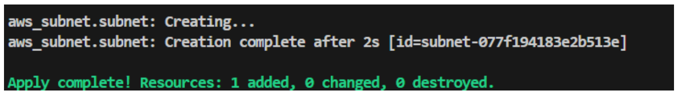
4) Create Subnet

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

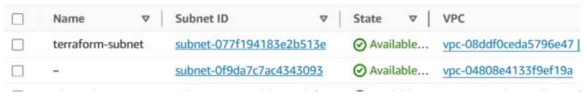
```
resource "aws_subnet" "subnet" {
     vpc_id    = aws_vpc.main.id
  cidr_block = "192.168.0.0/28"
  availability_zone = "us-east-2a"
  tags = {
    Name = "terraform-subnet"
  }
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply **<check on console if its created>**



```
aws_subnet.subnet: Creating...
aws_subnet.subnet: Creation complete after 2s [id=subnet-077f194183e2b513e]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



	Name	Subnet ID	State	VPC
☐	terraform-subnet	subnet-077f194183e2b513e	✓ Available...	vpc-08ddf0ceda5796e47 |
☐	–	subnet-0f9da7c7ac4343093	✓ Available...	vpc-04808e4133f9ef19a

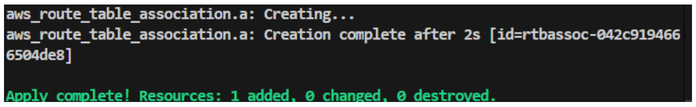
5) ASSOCIATE SUBNET WITH ROUTE TABLE

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

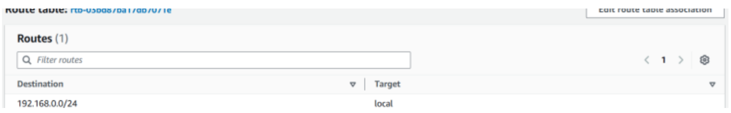
```
resource "aws_route_table_association" "a" {
  subnet_id      = aws_subnet.subnet.id
  route_table_id = aws_route_table.rt.id
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply **<check on console if its created>**



```
aws_route_table_association.a: Creating...
aws_route_table_association.a: Creation complete after 2s [id=rtbassoc-042c919466
6504de8]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



Route table: rtb-03bd87ba17db7071e

Routes (1)

Destination	Target
192.168.0.0/24	local

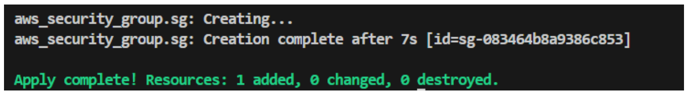
6) CREATE SECURITY GROUP TO ALLOW PORT 22,80,443

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

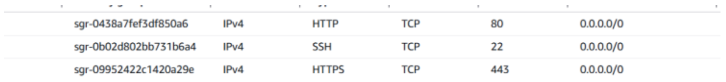
```
resource "aws_security_group" "sg" {
  vpc_id = aws_vpc.main.id
  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 443
    to_port     = 443
    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 = "main-sg"
  }
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply **<check on console if its created>**



```
aws_security_group.sg: Creating...
aws_security_group.sg: Creation complete after 7s [id=sg-083464b8a9386c853]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



sgr-0438a7fef3df850a6	IPv4	HTTP	TCP	80	0.0.0.0/0
sgr-0b02d802bb731b6a4	IPv4	SSH	TCP	22	0.0.0.0/0
sgr-09952422c1420a29e	IPv4	HTTPS	TCP	443	0.0.0.0/0

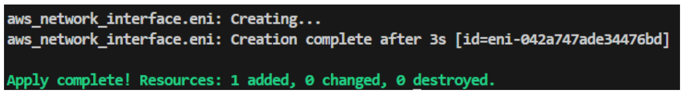
7) Create a network interface with an IP in the subnet that was created in step 4

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

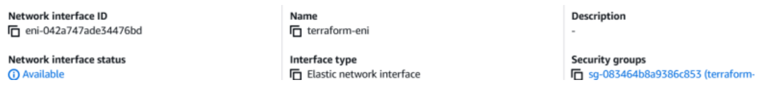
```
resource "aws_network_interface" "eni" {
  subnet_id       = aws_subnet.subnet.id
  private_ips     = ["192.168.0.12"]
  security_groups = [aws_security_group.sg.id]
    tags = {
    Name = "terraform-eni"
  }
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply **<check on console if its created>**



```
aws_network_interface.eni: Creating...
aws_network_interface.eni: Creation complete after 3s [id=eni-042a747ade34476bd]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



Network interface ID
eni-042a747ade34476bd

Network interface status
Available

Name
terraform-eni

Interface type
Elastic network interface

Description
-

Security groups
sg-083464b8a9386c853 (terraform-

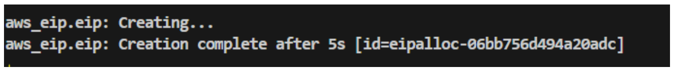
8) Assign an elastic IP to the network interface created in step 7

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

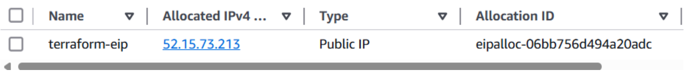
```
resource "aws_eip" "eip" {
  vpc             = true
  network_interface = aws_network_interface.eni.id
  tags = {
    Name = "terraform-eip"
  }
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply **<check on console if its created>**



```
aws_eip.eip: Creating...
aws_eip.eip: Creation complete after 5s [id=eipalloc-06bb756d494a20adc]
```



Name	Allocated IPv4 ...	Type	Allocation ID
terraform-eip	52.15.73.213	Public IP	eipalloc-06bb756d494a20adc

Note:

1) Create single main.tf which will be created the above resources and do not hardcode the ID's.

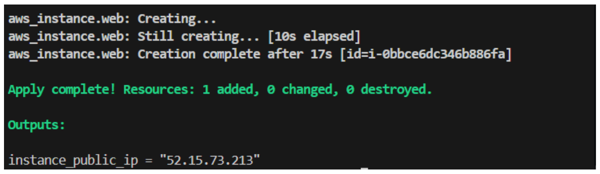
9) Create Ubuntu server and install/enable apache2

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

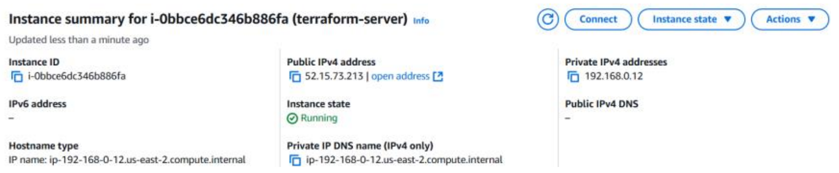
```
resource "aws_instance" "web" {
  ami           = "ami-036841078a4b68e14"
  instance_type = "t2.micro"
  network_interface {
    network_interface_id = aws_network_interface.eni.id
    device_index         = 0
  }

  user_data = <<-EOF
              #!/bin/bash
              sudo apt-get update -y
              sudo apt-get install -y apache2
              sudo systemctl start apache2
              sudo systemctl enable apache2
              EOF

  tags = {
    Name = "terraform-server"
  }
}

output "instance_public_ip" {
  value = aws_instance.web.public_ip
}
```

→ **CRL+S**

# terraform init > terraform plan > terraform apply **<check on console if its created>**



```
aws_instance.web: Creating...
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Creation complete after 17s [id=i-0bbce6dc346b886fa]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

instance_public_ip = "52.15.73.213"
```

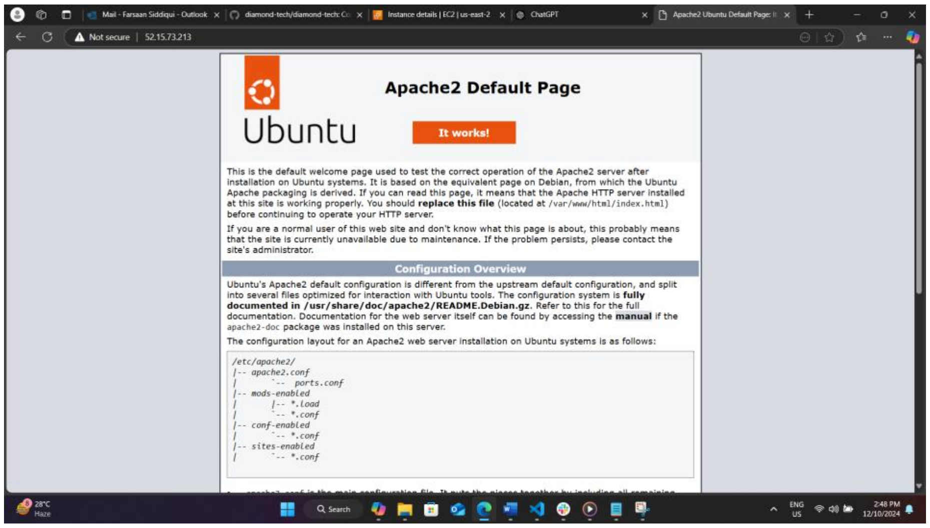


Instance summary for i-0bbce6dc346b886fa (terraform-server) Info

Updated less than a minute ago

Instance ID: i-0bbce6dc346b886fa
Public IPv4 address: 52.15.73.213 | open address
Private IPv4 addresses: 192.168.0.12
IPv6 address: –
Instance state: Running
Public IPv4 DNS: –
Hostname type: IP name: ip-192-168-0-12.us-east-2.compute.internal
Private IP DNS name (IPv4 only): ip-192-168-0-12.us-east-2.compute.internal

<CHECKING IF THE APACHE IS INSTALLED AND RUNNING BY USING PUBLIC IP *http://52.15.73.213:80* **>**



Apache2 Default Page

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

CONFIGURE S3 AS BACKEND AND DYNAMO DB LOCKING FOR MULTI USER EXECUTION.

→CREATING A S3 BUCKET AND DYNAMO DB

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

```
resource "aws_s3_bucket" "terraform_state" {
  bucket = "terraform-task-bucket"
  acl    = "private"

}

resource "aws_dynamodb_table" "terraform_state_lock" {
  name         = "terraform-task-dynamodb"
  hash_key     = "LockID"
  read_capacity  = 20
  write_capacity = 20

  attribute {
    name = "LockID"
    type = "S"
  }
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply

```
aws_dynamodb_table.terraform_state_lock: Creating...
aws_s3_bucket.terraform_state: Creating...
aws_s3_bucket.terraform_state: Creation complete after 7s [id=terraform-task-buck
et]
aws_dynamodb_table.terraform_state_lock: Still creating... [10s elapsed]
aws_dynamodb_table.terraform_state_lock: Creation complete after 11s [id=terrafor
m-task-dynamodb]
```

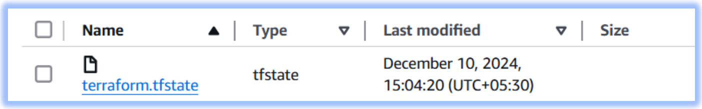
→CONFIGURING S3 AS BACKEND AND DYNAMODB LOCKING

>>ADD THIS TEMPLATE TO EXIXTING main.tf **FILE**

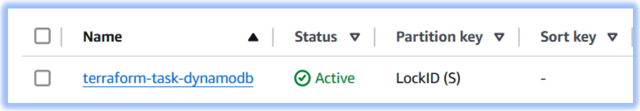
```
terraform {
  backend "s3" {
    bucket         = "terraform-task-bucket"
    key            = "terraform.tfstate"
    region         = "us-east-2"
    dynamodb_table = "terraform-task-dynamodb"
    encrypt        = true
  }
}
```

→ **CRL+S**

terraform init > terraform plan > terraform apply



Name	Type	Last modified	Size
terraform.tfstate	tfstate	December 10, 2024, 15:04:20 (UTC+05:30)	



Name	Status	Partition key	Sort key
terraform-task-dynamodb	Active	LockID (S)	-