

CASE STUDY -CREATING AN ARCHITECTURE USING TERRAFORM ON AWS

You work as a DevOps Engineer in leading Software Company. You have been asked to build an infrastructure safely and efficiently.

The company Requirements:

1. Use AWS cloud Provider and the software to be installed is Apache2
2. Use Ubuntu AMI

The company wants the Architecture to have the following services:

1. Create a template with a VPC, 2 subnets and 1 instance in each subnet
 2. Attach Security groups, internet gateway and network interface to the instance
-

Terraform Script:

```
provider "aws" {  
  region = "us-east-2"  
}
```

Create VPC

```
resource "aws_vpc" "my_vpc" {  
  cidr_block = "10.0.0.0/16"  
  enable_dns_support = true  
  enable_dns_hostnames = true  
  tags = {  
    Name = "casestudy_VPC"  
  }  
}
```

Create first Subnet

```
resource "aws_subnet" "subnet_a" {  
  vpc_id      = aws_vpc.my_vpc.id  
  cidr_block   = "10.0.1.0/24"  
  availability_zone = "us-east-2a" #Set your desired availability zone  
  tags = {  
    Name = "assignment_subnet_a"
```

```
}  
}
```

Create second Subnet

```
resource "aws_subnet" "subnet_b" {  
  vpc_id      = aws_vpc.my_vpc.id  
  cidr_block  = "10.0.2.0/24"  
  availability_zone = "us-east-2b" #Set your desired availability zone  
  tags = {  
    Name = "assignment_subnet_b"  
  }  
}
```

Create Internet Gateway

```
resource "aws_internet_gateway" "my_igw" {  
  vpc_id = aws_vpc.my_vpc.id  
  tags = {  
    Name = "casestudy_IGW"  
  }  
}
```

Create Security Group

```
resource "aws_security_group" "instance_security_group" {  
  vpc_id = aws_vpc.my_vpc.id  
  tags = {  
    Name = "casestudy_SG"  
  }  
}
```

Create Network Interface

```
resource "aws_network_interface" "instance_nic" {
```

```
subnet_id = aws_subnet.subnet_a.id
private_ips = ["10.0.1.10"]
}

# Create First Instance inside Subnet_a
resource "aws_instance" "my_instance_a" {
  ami          = "ami-09040d770ffe2224f" # Set your desired AMI ID
  instance_type = "t2.micro"             # Set your desired instance type
  subnet_id    = aws_subnet.subnet_a.id
  security_groups = [aws_security_group.instance_security_group.id]
  tags = {
    Name = "casestudy_instance_1"
  }
}
```

```
# Create second Instance inside Subnet_b
resource "aws_instance" "my_instance_b" {
  ami          = "ami-09040d770ffe2224f" # Set your desired AMI ID
  instance_type = "t2.micro"             # Set your desired instance type
  subnet_id    = aws_subnet.subnet_b.id
  security_groups = [aws_security_group.instance_security_group.id]
  tags = {
    Name = "casestudy_instance_2"
  }
}
```

```

ubuntu@ip-172-31-34-196:~$ mkdir casestudy
ubuntu@ip-172-31-34-196:~$ cd casestudy
ubuntu@ip-172-31-34-196:~/casestudy$ vi main.tf
ubuntu@ip-172-31-34-196:~/casestudy$ cat main.tf
provider "aws" {
  region = "us-east-2"
}

# Create VPC
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/16"
  enable_dns_support = true
  enable_dns_hostnames = true
  tags = {
    Name = "casestudy_VPC"
  }
}

```

i-0d81c556e43b0246d (Terraform-Assignment)

PublicIPs: 13.126.139.48 PrivateIPs: 172.31.34.196

```

# Create first Subnet
resource "aws_subnet" "subnet_a" {
  vpc_id      = aws_vpc.my_vpc.id
  cidr_block  = "10.0.1.0/24"
  availability_zone = "us-east-2a" #Set your desired availability zone
  tags = {
    Name = "assignment_subnet_a"
  }
}

# Create second Subnet
resource "aws_subnet" "subnet_b" {
  vpc_id      = aws_vpc.my_vpc.id
  cidr_block  = "10.0.2.0/24"
  availability_zone = "us-east-2b" #Set your desired availability zone
  tags = {
    Name = "assignment_subnet_b"
  }
}

```

i-0d81c556e43b0246d (Terraform-Assignment)

PublicIPs: 13.126.139.48 PrivateIPs: 172.31.34.196

```
# Create Internet Gateway
resource "aws_internet_gateway" "my_igw" {
  vpc_id = aws_vpc.my_vpc.id
  tags = {
    Name = "casestudy_IGW"
  }
}

# Create Security Group
resource "aws_security_group" "instance_security_group" {
  vpc_id = aws_vpc.my_vpc.id
  tags = {
    Name = "casestudy_SG"
  }
}
```

```
# Create Network Interface
resource "aws_network_interface" "instance_nic" {
  subnet_id = aws_subnet.subnet_a.id
  private_ips = ["10.0.1.10"]
}

# Create First Instance inside Subnet_a
resource "aws_instance" "my_instance_a" {
  ami           = "ami-09040d770ffe2224f" # Set your desired AMI ID
  instance_type = "t2.micro"               # Set your desired instance type
  subnet_id     = aws_subnet.subnet_a.id
  security_groups = [aws_security_group.instance_security_group.id]
  tags = {
    Name = "casestudy_instance_1"
  }
}
```

```
# Create second Instance inside Subnet_b
resource "aws_instance" "my_instance_b" {
  ami           = "ami-09040d770ffe2224f" # Set your desired AMI ID
  instance_type = "t2.micro"               # Set your desired instance type
  subnet_id     = aws_subnet.subnet_b.id
  security_groups = [aws_security_group.instance_security_group.id]
  tags = {
    Name = "casestudy_instance_2"
  }
}
```

ubuntu@ip-172-31-34-196:~/casestudy\$

i-0d81c556e43b0246d (Terraform-Assignment)

PublicIPs: 13.126.139.48 PrivateIPs: 172.31.34.196

```
ubuntu@ip-172-31-34-196:~/casestudy$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.52.0...
- Installed hashicorp/aws v5.52.0 (signed by HashiCorp)

Terraform has created a lock file **.terraform.lock.hcl** to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

```
ubuntu@ip-172-31-34-196:~/casestudy$
```

i-0d81c556e43b0246d (Terraform-Assignment)

PublicIPs: 13.126.139.48 PrivateIPs: 172.31.34.196

```
ubuntu@ip-172-31-34-196:~/casestudy$ terraform plan
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

aws_instance.my_instance_a will be created
+ resource "aws_instance" "my_instance_a" {

i-0d81c556e43b0246d (Terraform-Assignment)

PublicIPs: 13.126.139.48 PrivateIPs: 172.31.34.196

```
ubuntu@ip-172-31-34-196:~/casestudy$ terraform apply
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

aws_instance.my_instance_a will be created
+ resource "aws_instance" "my_instance_a" {
+ ami = "ami-09040d770ffe2224f"

i-0d81c556e43b0246d (Terraform-Assignment)

PublicIPs: 13.126.139.48 PrivateIPs: 172.31.34.196

```
Enter a value: yes

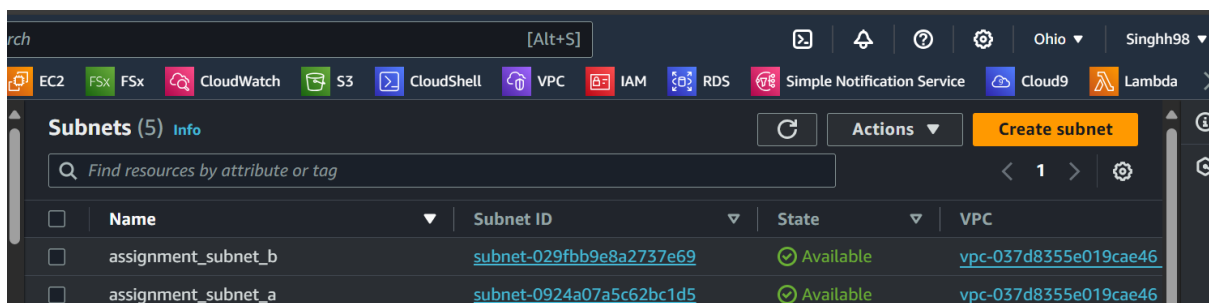
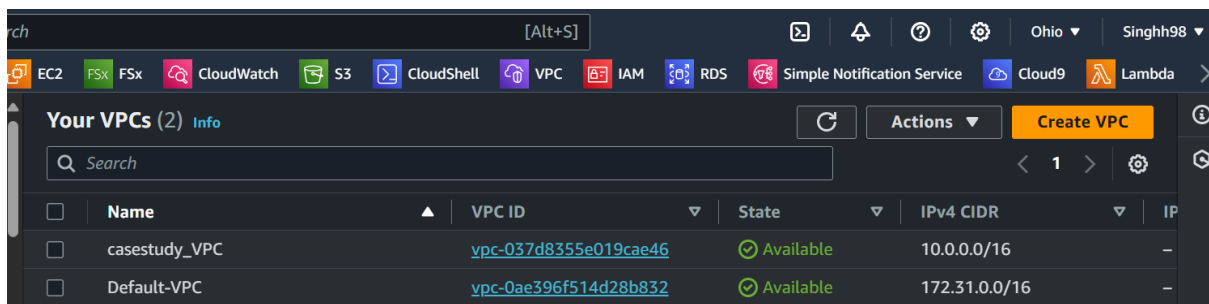
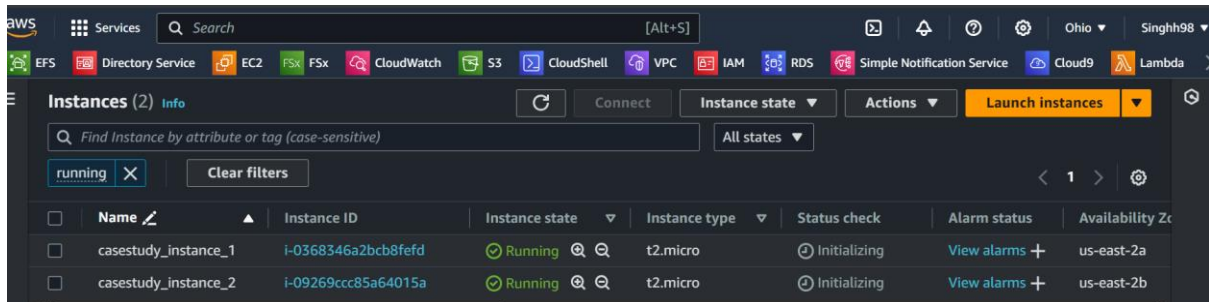
aws_vpc.my_vpc: Creating...
aws_vpc.my_vpc: Still creating... [10s elapsed]
aws_vpc.my_vpc: Creation complete after 14s [id=vpc-037d8355e019cae46]
aws_subnet.subnet_a: Creating...
aws_internet_gateway.my_igw: Creating...
aws_subnet.subnet_b: Creating...
aws_security_group.instance_security_group: Creating...
aws_subnet.subnet_b: Creation complete after 2s [id=subnet-029fbb9e8a2737e69]
aws_subnet.subnet_a: Creation complete after 2s [id=subnet-0924a07a5c62bc1d5]
aws_network_interface.instance_nic: Creating...
aws_internet_gateway.my_igw: Creation complete after 2s [id=igw-09f4884be36b0b81b]
aws_network_interface.instance_nic: Creation complete after 1s [id=eni-07f5145bd183021ff]
aws_security_group.instance_security_group: Creation complete after 4s [id=sg-0071948a8cf241a98]
aws_instance.my_instance_a: Creating...
aws_instance.my_instance_b: Creating...
aws_instance.my_instance_a: Still creating... [10s elapsed]
aws_instance.my_instance_b: Still creating... [10s elapsed]
aws_instance.my_instance_a: Still creating... [20s elapsed]
aws_instance.my_instance_b: Still creating... [20s elapsed]
aws_instance.my_instance_a: Still creating... [30s elapsed]
aws_instance.my_instance_b: Still creating... [30s elapsed]
aws_instance.my_instance_a: Creation complete after 34s [id=i-0368346a2bcb8fefdf]
aws_instance.my_instance_b: Creation complete after 34s [id=i-09269ccc85a64015a]

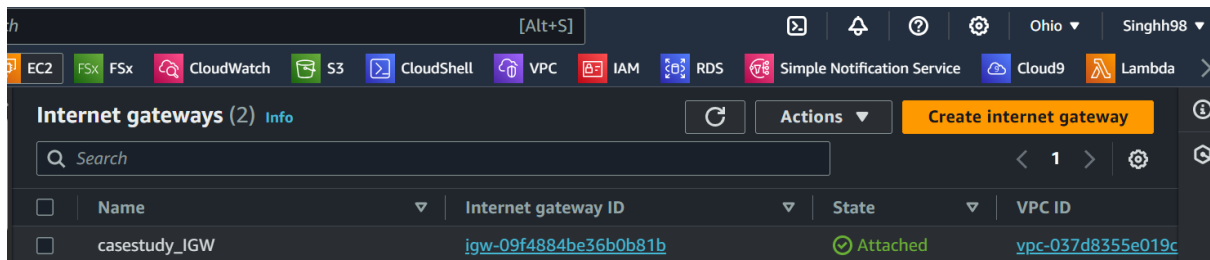
Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-34-196:~/casestudy$
```

i-0d81c556e43b0246d (Terraform-Assignment)

PublicIPs: 13.126.139.48 PrivateIPs: 172.31.34.196

We could see in the below snip that all the resources have been launched.





***** THE END *****