

Terraform

Creating VPC & EC2 instance

- ✧ Create directory sample *mkdir vpc* > go to directory sample *cd vpc*.
- ✧ Creating VPC infrastructure file.

Main.tf

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

resource "aws_vpc" "myvpc" {
  cidr_block      = "10.0.0.0/16"
  instance_tenancy = "default"
  tags = {
    Name = "main"
  }
}

resource "aws_subnet" "pubsub" {
  vpc_id     = aws_vpc.myvpc.id
  cidr_block = "10.0.1.0/24"

  tags = {
    Name = "publicsubnet"
  }
}

resource "aws_internet_gateway" "tigw" {
  vpc_id = aws_vpc.myvpc.id

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

resource "aws_route_table" "pubrt" {
  vpc_id = aws_vpc.myvpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.tigw.id
  }
  tags = {
    Name = "myvpc_pubrt"
  }
}

resource "aws_route_table_association" "pubass" {
  subnet_id     = aws_subnet.pubsub.id
  route_table_id = aws_route_table.pubrt.id
}
```

```

resource "aws_security_group" "allow_all" {
  name      = "allow_all"
  description = "Allow TLS inbound traffic"
  vpc_id    = aws_vpc.myvpc.id

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

  ingress {
    description = "TLS from VPC"
    from_port   = 80
    to_port     = 80
    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"]
    ipv6_cidr_blocks = [ "::/0" ]
  }

  tags = {
    Name = "allow_all_myvpc"
  }
}

resource "aws_instance" "terra" {
  ami = "ami-002068ed284fb165b"
  instance_type = "t2.micro"
  subnet_id = aws_subnet.pubsub.id
  key_name = "webapp"
  vpc_security_group_ids = ["${aws_security_group.allow_all.id}"]
  associate_public_ip_address = true
  tags = {
    Name = "ec2_instance_terra"
  }
}

```

- ✧ Now build the infrastructure.
- ✧ Execute ***terraform init*** > initialize a working directory that contains a Terraform configuration.

```
[root@ip-172-31-30-219 ~]# terraform --version
Terraform v1.1.2
on linux_amd64
[root@ip-172-31-30-219 ~]# clear
[root@ip-172-31-30-219 ~]# cd
[root@ip-172-31-30-219 ~]# mkdir myvpc
[root@ip-172-31-30-219 ~]# cd myvpc
[root@ip-172-31-30-219 myvpc]# vi myvpc.tf
myvpc.tf
[root@ip-172-31-30-219 myvpc]# terraform init
Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.70.0...
- Installed hashicorp/aws v3.70.0 (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.
```

- ✧ Next, execute ***terraform plan*** > creates an execution plan.

```
[root@ip-172-31-30-219 myvpc]# 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_vpc.myvpc will be created
+ resource "aws_vpc" "myvpc" {
  + arn                = (known after apply)
  + cidr_block          = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id     = (known after apply)
  + enable_classiclink   = (known after apply)
  + enable_classiclink_dns_support = (known after apply)
  + enable_dns_hostnames = (known after apply)
  + enable_dns_support   = true
  + id                  = (known after apply)
  + instance_tenancy    = "default"
  + ipv6_association_id = (known after apply)
  + ipv6_cidr_block      = (known after apply)
  + main_route_table_id = (known after apply)
  + owner_id            = (known after apply)
  + tags                = {
    + "Name" = "main"
  }
  + tags_all            = {
    + "Name" = "main"
  }
}

Plan: 1 to add, 0 to change, 0 to destroy.
```

- ✧ Execute ***terraform apply*** > executes the actions proposed in a Terraform plan.

```
+ resource "aws_vpc" "myvpc" {
  + arn                = (known after apply)
  + cidr_block          = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id     = (known after apply)
  + enable_classiclink   = (known after apply)
  + enable_classiclink_dns_support = (known after apply)
  + enable_dns_hostnames = (known after apply)
  + enable_dns_support   = true
  + id                  = (known after apply)
  + instance_tenancy    = "default"
  + ipv6_association_id = (known after apply)
  + ipv6_cidr_block      = (known after apply)
  + main_route_table_id = (known after apply)
  + owner_id            = (known after apply)
  + tags                = {
    + "Name" = "main"
  }
  + tags_all            = {
    + "Name" = "main"
  }
}

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.myvpc: Creating...
aws_vpc.myvpc: Creation complete after 1s [id=vpc-00bbc246302ee7d52]

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

✧ Go to aws VPC console and VPC is created.

New VPC Experience
Tell us what you think

VPC Dashboard
EC2 Global View **New**
Filter by VPC:
Select a VPC
VIRTUAL PRIVATE CLOUD
Your VPCs
Subnets

Your VPCs (2) Info

Filter VPCs

Actions Create VPC

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	vpc-52553639	Available	172.31.0.0/16	-
<input type="checkbox"/>	main	vpc-0402ddafa55ef10c8	Available	10.0.0.0/16	-

New EC2 Experience
Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances **New**
Instance Types
Launch Templates
Spot Requests
Savings Plans

Instances (4) Info

Search

Connect Instance state Actions Launch Instances

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
<input type="checkbox"/>	app	i-041540f90beff5cf2	Stopped	t2.micro	-	No alarms	us-e
<input type="checkbox"/>	terraform	i-0829c9ce957db4ae2	Running	t2.micro	2/2 checks passed	No alarms	us-e
<input type="checkbox"/>	tom	i-0f0e394407a1b8013	Stopped	t2.micro	-	No alarms	us-e
<input type="checkbox"/>	ec2_instance_t...	i-0e3a3908d25a7c4b2	Running	t2.micro	2/2 checks passed	No alarms	us-e

Select an instance