

PROJECT-1

Deploy Three-Tier Architecture in AWS using Terraform

Step1: Create a file for Provider

```
1 terraform {
2     required_providers {
3         aws = {
4             source = "hashicorp/aws"
5             version = "~> 4.65.0"
6         }
7     }
8 }
9
10
11 provider "aws" {
12     profile     = "default"
13     region      = "ap-south-1"
14     access_key  = "AKIAVMQJROITQWSD727D"
15     secret_key  = "B02gg+udlnUwwm8isWlz+btMViPmP/A/RprFSpf/"
16 }
17 }
```

```
ec2-user@ip-172-31-92-53:~/ terraform
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!

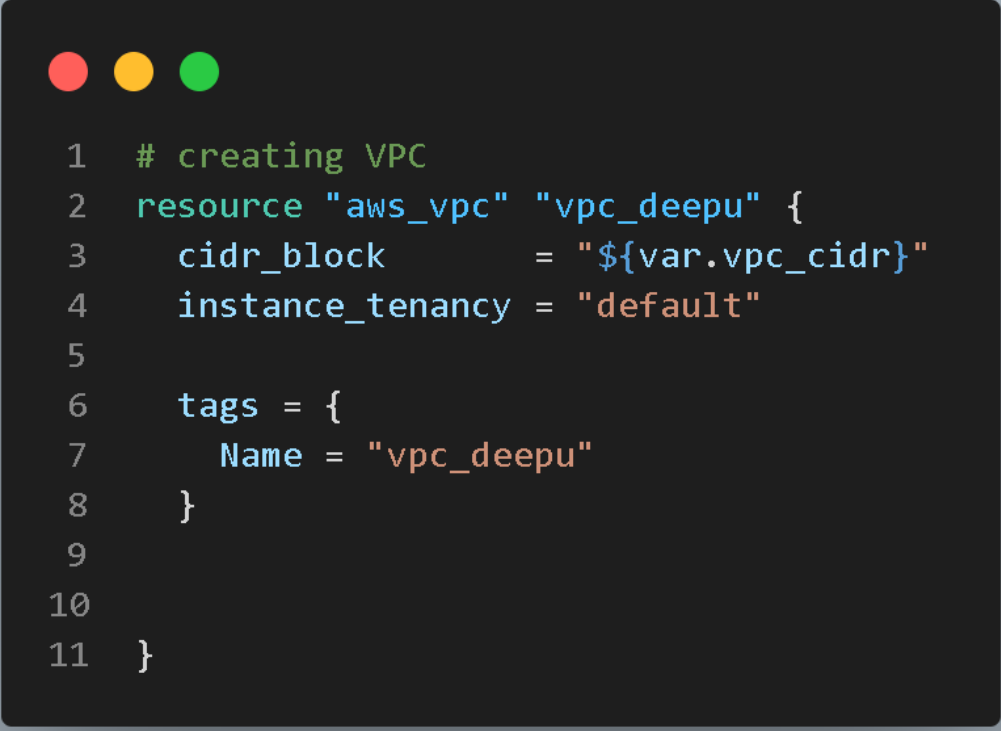
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

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.
[ec2-user@ip-172-31-92-53 terraform]$ ls
provider.tf
[ec2-user@ip-172-31-92-53 terraform]$ vi provider.tf
[ec2-user@ip-172-31-92-53 terraform]$ terraform fmt
[ec2-user@ip-172-31-92-53 terraform]$ terraform validate
Success! The configuration is valid.
[ec2-user@ip-172-31-92-53 terraform]$ terraform plan
No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
[ec2-user@ip-172-31-92-53 terraform]$ terraform apply
No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

Step2: Create a file for the vpc



```
1  # creating VPC
2  resource "aws_vpc" "vpc_deepu" {
3      cidr_block      = "${var.vpc_cidr}"
4      instance_tenancy = "default"
5
6      tags = {
7          Name = "vpc_deepu"
8      }
9
10
11 }
```

```

+ 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
+ enable_network_address_usage_metrics = (known after apply)
+ id = (known after apply)
+ instance_tenancy = "default"
+ ipv6_association_id = (known after apply)
+ ipv6_cidr_block = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id = (known after apply)
+ owner_id = (known after apply)
+ tags = {
+   "Name" = "vpc_pjt"
+ }
+ tags_all = {
+   "Name" = "vpc_pjt"
+ }
}

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

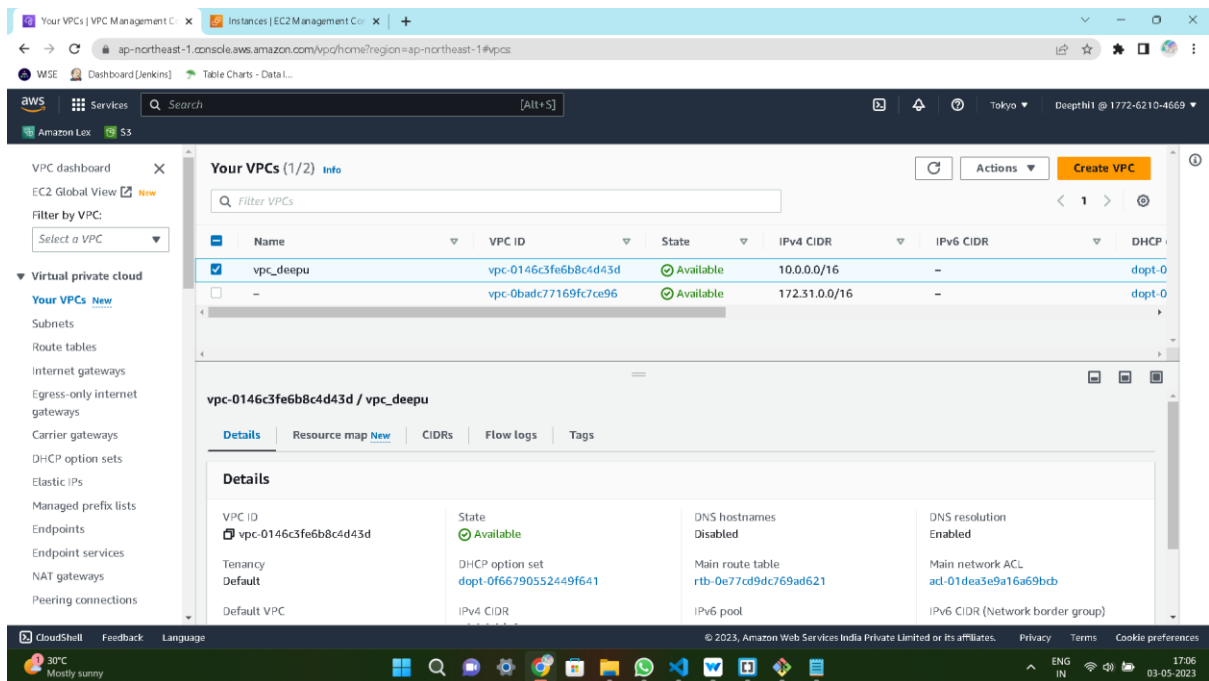
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_vpc.vpc_deepu: Creating...
aws_vpc.vpc_deepu: Creation complete after 1s [id=vpc-0146c3fe6b8c4d43d]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$

```



Step3: Create a file for the Subnet

```
1  #creating 1st web subnet
2  resource "aws_subnet" "pub-sub-1" {
3      vpc_id            = aws_vpc.vpc_deepu.id
4      cidr_block         = "${var.websub_1}"
5      availability_zone  = "ap-northeast-1a"
6      map_public_ip_on_launch = true
7
8      tags = {
9          Name = "pub-sub-1"
10     }
11 }
12
13
14 #creating 2nd web subnet
15 resource "aws_subnet" "pub-sub-2" {
16     vpc_id            = aws_vpc.vpc_deepu.id
17     cidr_block         = "${var.websub_2}"
18     availability_zone  = "ap-northeast-1c"
19     map_public_ip_on_launch = true
20
21     tags = {
22         Name = "pub-sub-2"
23     }
24 }
25
26
```

```
1  #creating 1st application subnet
2  resource "aws_subnet" "app-subnet-1" {
3      vpc_id            = aws_vpc.vpc_deepu.id
4      cidr_block         = "${var.appsub_1}"
5      availability_zone   = "ap-northeast-1a"
6      map_public_ip_on_launch = false
7
8      tags = {
9          Name = "app-subnet-1"
10     }
11 }
12
13
14 #creating 2nd application subnet
15 resource "aws_subnet" "app-subnet-2" {
16     vpc_id            = aws_vpc.vpc_deepu.id
17     cidr_block         = "${var.appsub_2}"
18     availability_zone   = "ap-northeast-1c"
19     map_public_ip_on_launch = false
20
21     tags = {
22         Name = "app-subnet-2"
23     }
24 }
25
26
```

```
1  #creating 1st data base subnet
2  resource "aws_subnet" "dbs-1" {
3      vpc_id            = aws_vpc.vpc_deepu.id
4      cidr_block         = "${var.dbsub_1}"
5      availability_zone   = "ap-northeast-1a"
6      map_public_ip_on_launch = false
7
8      tags = {
9          Name = "dbs-1"
10     }
11 }
12
13
14 #creating 2nd data base subnet
15 resource "aws_subnet" "dbs-2" {
16     vpc_id            = aws_vpc.vpc_deepu.id
17     cidr_block         = "${var.dbsub_2}"
18     availability_zone   = "ap-northeast-1c"
19     map_public_ip_on_launch = false
20
21     tags = {
22         Name = "dbs-2"
23     }
24 }
25
26
```

```

ec2-user@ip-172-31-6-110:~/terraform
+ owner_id = (known af
ter apply)
+ private_dns_hostname_type_on_launch = (known af
ter apply)
+ tags = {
  "Name" = "pub-sub-2"
}
+ tags_all = {
  "Name" = "pub-sub-2"
}
+ vpc_id = "vpc-0146
c3fe6b8c4d43d"
}

Plan: 6 to add, 0 to change, 0 to destroy.

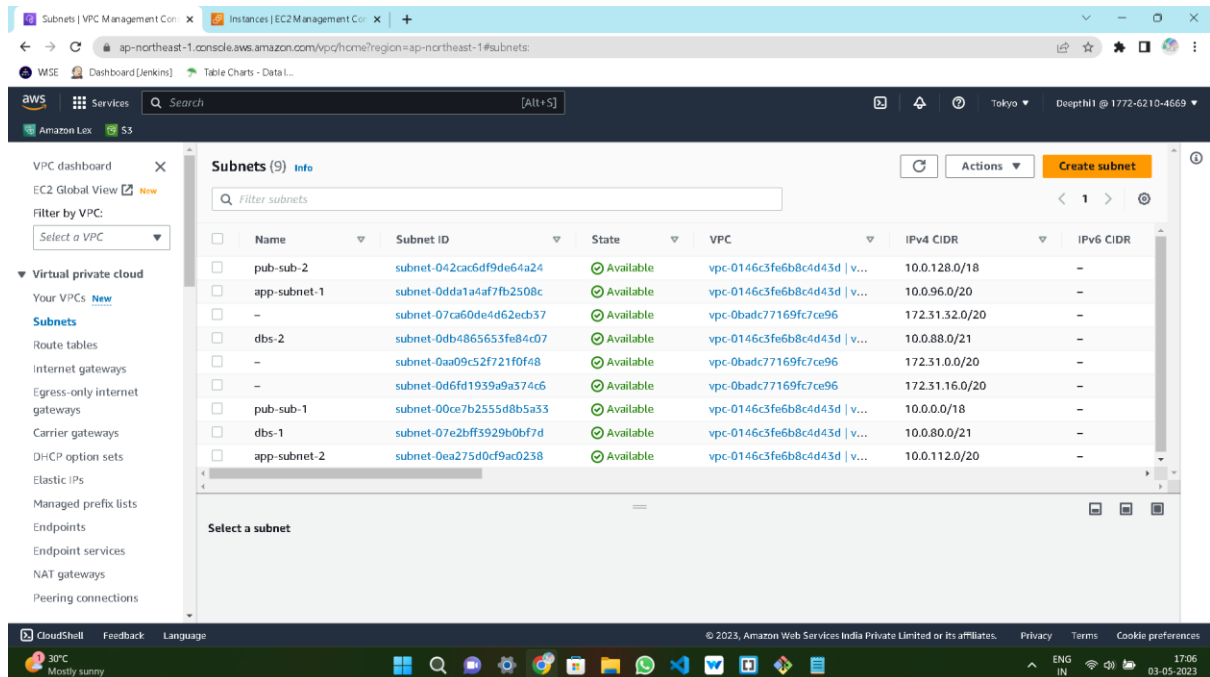
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_subnet.pub-sub-2: Creating...
aws_subnet.pub-sub-1: Creating...
aws_subnet.app-subnet-2: Creating...
aws_subnet.dbs-1: Creating...
aws_subnet.dbs-2: Creating...
aws_subnet.app-subnet-1: Creating...
aws_subnet.app-subnet-2: Creation complete after 1s [id=subnet-0ea275d0cf9ac0238]
aws_subnet.app-subnet-1: Creation complete after 1s [id=subnet-0dda1a4af7fb2508c]
aws_subnet.dbs-1: Creation complete after 1s [id=subnet-07e2bfff3929b0bf7d]
aws_subnet.dbs-2: Creation complete after 1s [id=subnet-0db4865653fe84c07]
aws_subnet.pub-sub-2: Still creating... [10s elapsed]
aws_subnet.pub-sub-1: Still creating... [10s elapsed]
aws_subnet.pub-sub-1: Creation complete after 11s [id=subnet-00ce7b2555d8b5a33]
aws_subnet.pub-sub-2: Creation complete after 11s [id=subnet-042cac6df9de64a24]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$

```



Step4: Create a file for the Internet Gateway

```
1  # creating internet gateway
2  resource "aws_internet_gateway" "igw_deepu" {
3      vpc_id = aws_vpc.vpc_deepu.id
4
5      tags = {
6          Name = "igw_deepu"
7      }
8
9  }
```

```
ec2-user@ip-172-31-6-110:~/terraform
5a33]
aws_subnet.app-subnet-1: Refreshing state... [id=subnet-0dd1a4af7fb2508c]

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_internet_gateway.igw_deepu will be created
+ resource "aws_internet_gateway" "igw_deepu" {
+   arn           = (known after apply)
+   id            = (known after apply)
+   owner_id      = (known after apply)
+   tags          = {
+     + "Name" = "igw_deepu"
+   }
+   tags_all      = {
+     + "Name" = "igw_deepu"
+   }
+   vpc_id        = "vpc-0146c3fe6b8c4d43d"
}

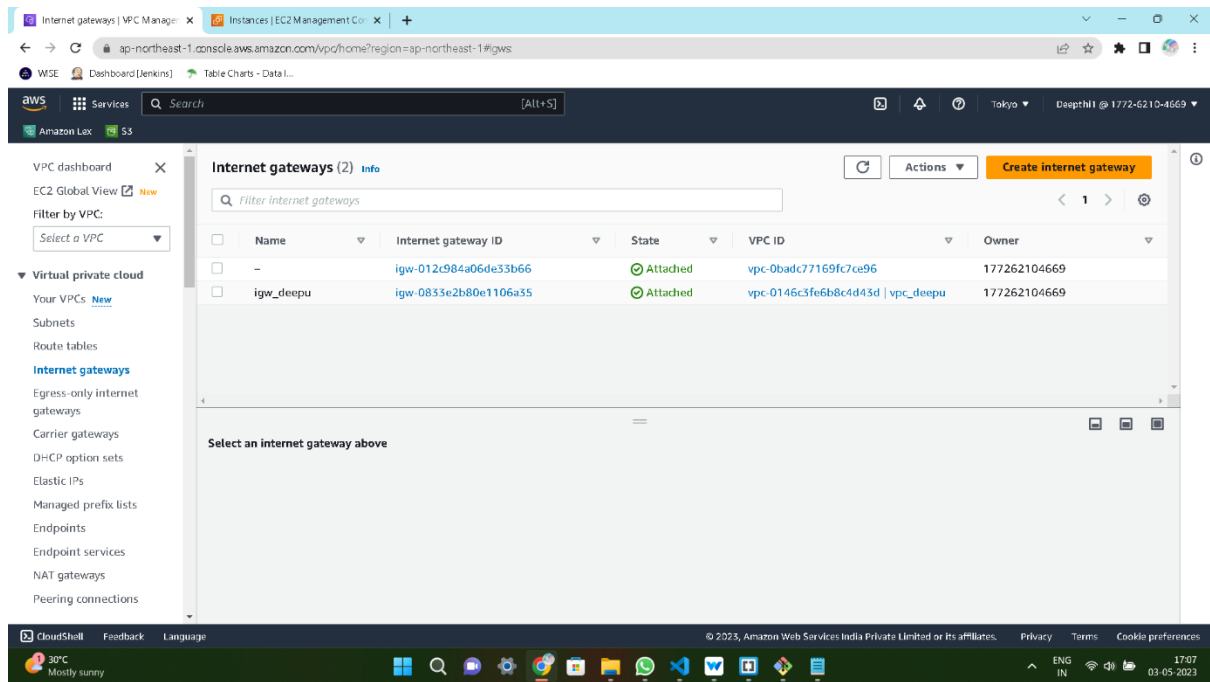
Plan: 1 to add, 0 to change, 0 to destroy.

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_internet_gateway.igw_deepu: Creating...
aws_internet_gateway.igw_deepu: Creation complete after 1s [id=igw-0833e2b80e1106a35]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$
```

Step5: Create a file for the Route Table

```
1  #creating route table
2  resource "aws_route_table" "public_route_table" {
3      vpc_id = aws_vpc.vpc_deepu.id
4
5      tags = {
6          Name = "public_route_table"
7      }
8
9  }
10 }
11
12 resource "aws_route" "route" {
13     route_table_id = aws_route_table.public_route_table.id
14     destination_cidr_block = "0.0.0.0/0"
15     gateway_id = aws_internet_gateway.igw_deepu.id
16
17 }
18 }
19
20 #subnet associations
21
22 resource "aws_route_table_association" "route1" {
23     subnet_id = aws_subnet.pub-sub-1.id
24     route_table_id = aws_route_table.public_route_table.id
25
26 }
27
28 resource "aws_route_table_association" "route2" {
29     subnet_id = aws_subnet.pub-sub-2.id
30     route_table_id = aws_route_table.public_route_table.id
31
32 }
```

```

ec2-user@ip-172-31-6-110:~/terraform
+ tags_all = {
+   + "Name" = "public_route_table"
+ }
+ vpc_id = "vpc-0146c3fe6b8c4d43d"
}

# aws_route_table_association.route1 will be created
+ resource "aws_route_table_association" "route1" {
+   id = (known after apply)
+   route_table_id = (known after apply)
+   subnet_id = "subnet-00ce7b255d8b5a33"
+ }

# aws_route_table_association.route2 will be created
+ resource "aws_route_table_association" "route2" {
+   id = (known after apply)
+   route_table_id = (known after apply)
+   subnet_id = "subnet-042cac6df9de64a24"
+ }

Plan: 4 to add, 0 to change, 0 to destroy.

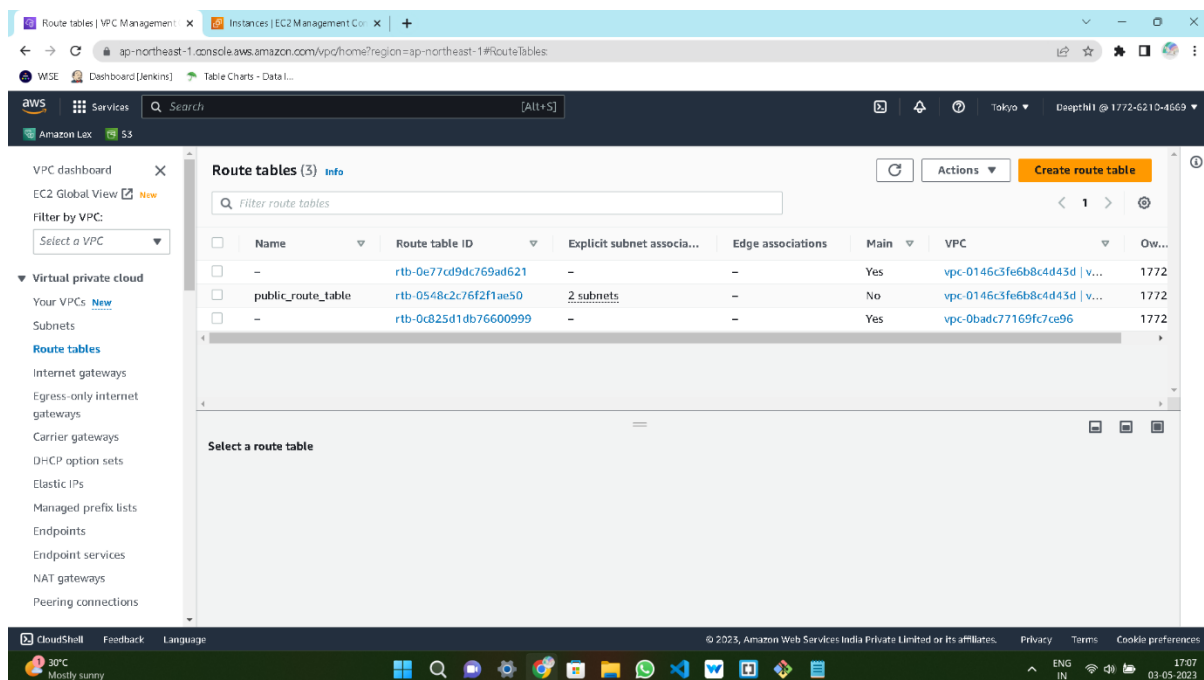
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_route_table.public_route_table: Creating...
aws_route_table.public_route_table: Creation complete after 0s [id=rtb-0548c2c76f2f1ae50]
aws_route_table_association.route2: Creating...
aws_route_table_association.route1: Creating...
aws_route.route: Creating...
aws_route_table_association.route1: Creation complete after 1s [id=rtbassoc-05f28fe6400f6b34a]
aws_route_table_association.route2: Creation complete after 1s [id=rtbassoc-08f0dfec8fbb742ab]
aws_route.route: Creation complete after 1s [id=r-rtb-0548c2c76f2f1ae501080289494]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$

```



Step6: Create a file for the EC2 Instance



```
1  # creating ec2 instance in public subnet 1
2  resource "aws_instance" "web_1" {
3      ami = "ami-0df2ca8a354185e1e"
4      instance_type = "t2.micro"
5      key_name = "deepu"
6      vpc_security_group_ids = [aws_security_group.wsg.id]
7      subnet_id = aws_subnet.pub-sub-1.id
8      associate_public_ip_address = true
9      user_data = "${file("userdata.sh")}"
10
11      tags = {
12          Name = "web_1"
13      }
14
15  }
16
17  #creating ec2 instance in 2nd public subnet
18  resource "aws_instance" "web_2" {
19      ami = "ami-0df2ca8a354185e1e"
20      instance_type = "t2.micro"
21      key_name = "deepu"
22      vpc_security_group_ids = [aws_security_group.wsg.id]
23      subnet_id = aws_subnet.pub-sub-2.id
24      associate_public_ip_address = true
25      user_data = "${file("userdata.sh")}"
26
27      tags = {
28          Name = "web_2"
29      }
30
31
32  }
```

```

ec2-user@ip-172-31-6-110:~/terraform
+ subnet_id = "subnet-042cac6df9d
e64a24"
+ tags = {
+   "Name" = "web_2"
+ }
+ tags_all = {
+   "Name" = "web_2"
+ }
+ tenancy = (known after apply)
+ user_data = "I631df8346d219bee3
de9910ca307eb8d013cf98"
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = [
+   "sg-0ce4afe0573e1c54e",
+ ]
}

Plan: 2 to add, 0 to change, 0 to destroy.

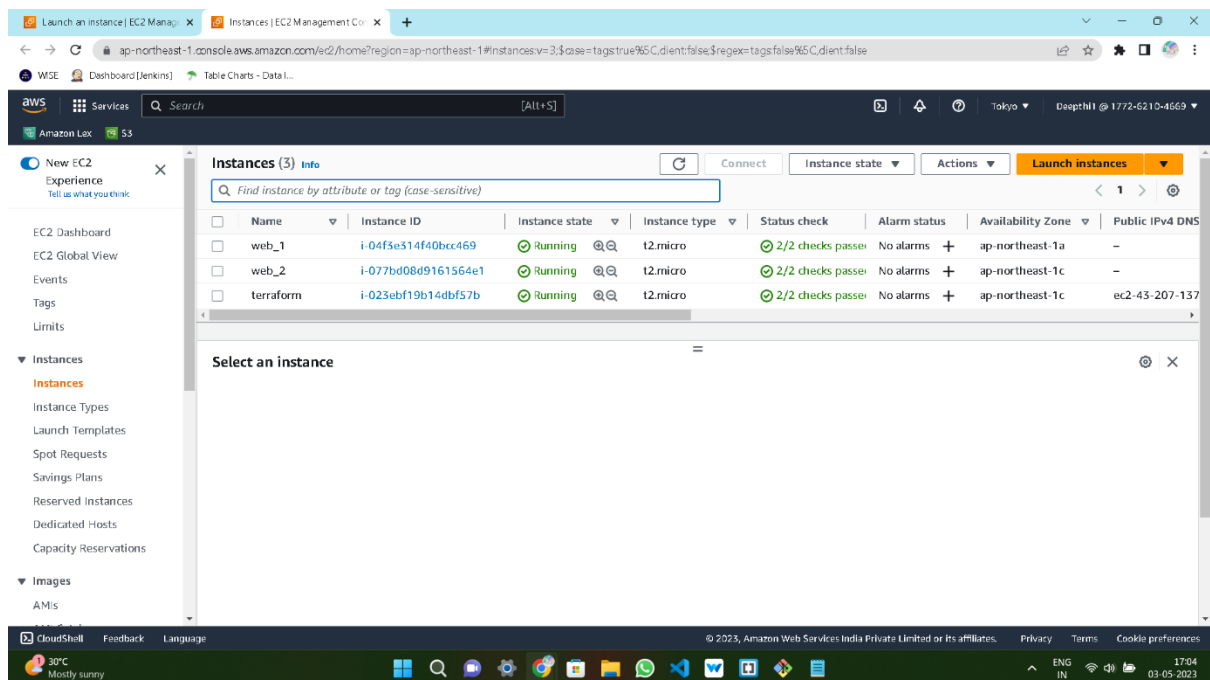
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.web_2: Creating...
aws_instance.web_1: Creating...
aws_instance.web_2: Still creating... [10s elapsed]
aws_instance.web_1: Still creating... [10s elapsed]
aws_instance.web_2: Still creating... [20s elapsed]
aws_instance.web_1: Still creating... [20s elapsed]
aws_instance.web_2: Still creating... [30s elapsed]
aws_instance.web_1: Still creating... [30s elapsed]
aws_instance.web_2: Creation complete after 32s [id=i-077bd08d9161564e1]
aws_instance.web_1: Creation complete after 32s [id=i-04f3e314f40bcc469]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$

```



Step7: Create a file for Security Group for the Front-end tier

```
1  # creating security group for frontend tier
2  resource "aws_security_group" "wsg" {
3    vpc_id = aws_vpc.vpc_deepu.id
4
5    #inbound rules
6    ingress {
7      from_port    = 80
8      to_port      = 80
9      protocol     = "tcp"
10     cidr_blocks = ["0.0.0.0/0"]
11   }
12
13   #HTTPS access from anywhere
14
15   ingress {
16     from_port    = 443
17     to_port      = 443
18     protocol     = "tcp"
19     cidr_blocks = ["0.0.0.0/0"]
20
21   }
22
23   #ssh access from anywhere
24
25   ingress {
26     from_port    = 22
27     to_port      = 22
28     protocol     = "tcp"
29     cidr_blocks = ["0.0.0.0/0"]
30   }
31
32   #outbound rules
33   egress {
34     from_port    = 0
35     to_port      = 0
36     protocol     = "-1"
37     cidr_blocks = ["0.0.0.0/0"]
38
39   }
40 }
41 tags = {
42   Name = "wsg"
43 }
44
45
46
47 }
```

```

ec2-user@ip-172-31-6-110:~/terraform
+ cidr_blocks = [
+   + "0.0.0.0/0",
+ ]
+ description = ""
+ from_port = 80
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 80
+ },
+ name = (known after apply)
+ name_prefix = (known after apply)
+ owner_id = (known after apply)
+ revoke_rules_on_delete = false
+ tags = {
+   + "Name" = "wsg"
+ }
+ tags_all = {
+   + "Name" = "wsg"
+ }
+ vpc_id = "vpc-0146c3fe6b8c4d43d"
}

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

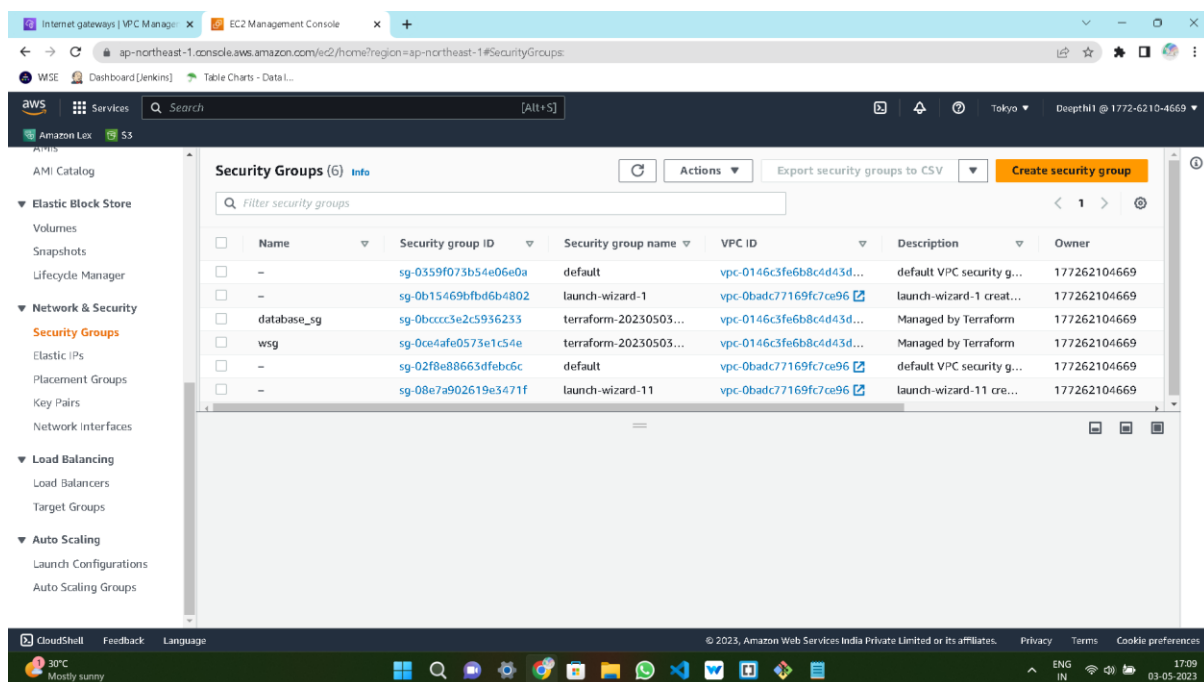
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_security_group.wsg: Creating...
aws_security_group.wsg: Creation complete after 2s [id=sg-0ce4afe0573e1c54e]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$

```



Step8: Create a file for Security Group for the Database tier

```

1  #creating database security group
2  resource "aws_security_group" "database_sg" {
3      vpc_id = aws_vpc.vpc_deepu.id
4
5      ingress {
6          description = "allow inbound traffic from application layer"
7          from_port = 3306
8          to_port = 3306
9          protocol = "tcp"
10         security_groups = [aws_security_group.wsg.id]
11     }
12
13     egress {
14         from_port = 32768
15         to_port = 65535
16         protocol = "tcp"
17         cidr_blocks = [ "0.0.0.0/0" ]
18     }
19
20     tags = {
21         Name = "database_sg"
22     }
23
24
25 }

```

```

ec2-user@ip-172-31-6-110:~/terraform
+ cidr_blocks      = []
+ description      = "allow inbound traffic from application layer"
+ from_port        = 3306
+ ipv6_cidr_blocks = []
+ prefix_list_ids  = []
+ protocol         = "tcp"
+ security_groups  = [
+   "sg-0ce4afe0573e1c54e",
+ ]
+ self             = false
+ to_port          = 3306
+ },
+ name             = (known after apply)
+ name_prefix      = (known after apply)
+ owner_id         = (known after apply)
+ revoke_rules_on_delete = false
+ tags             = {
+   "Name" = "database_sg"
+ }
+ tags_all         = {
+   "Name" = "database_sg"
+ }
+ vpc_id           = "vpc-0146c3fe6b8c4d43d"
}

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

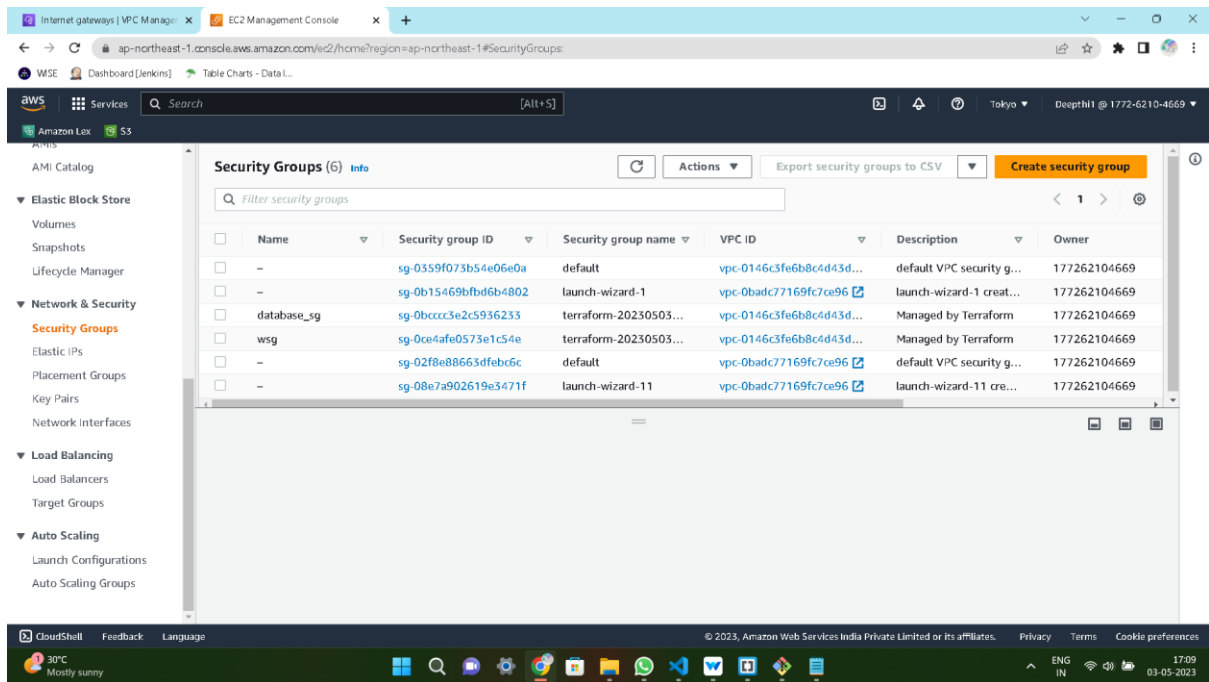
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_security_group.database_sg: Creating...
aws_security_group.database_sg: Creation complete after 2s [id=sg-0bcccc3e2c5936233]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$

```

Step9: Create a file for Application Load Balancer

```
1 resource "aws_lb" "external_lb" {
2   name           = "external-lb"
3   internal       = false
4   load_balancer_type = "application"
5   security_groups = [aws_security_group.wsg.id]
6   subnets       = [aws_subnet.pub-sub-1.id,aws_subnet.pub-sub-2.id]
7 }
8
9 resource "aws_lb_target_group" "target_alb" {
10  name      = "target-alb"
11  port      = 80
12  protocol  = "HTTP"
13  vpc_id    = aws_vpc.vpc_deepu.id
14 }
15
16
17 resource "aws_lb_target_group_attachment" "attahment" {
18   target_group_arn = aws_lb_target_group.target_alb.arn
19   target_id        = aws_instance.web_1.id
20   port             = 80
21   depends_on       = [aws_instance.web_1]
22 }
23
24
25 resource "aws_lb_target_group_attachment" "attachment2" {
26   target_group_arn = aws_lb_target_group.target_alb.id
27   target_id        = aws_instance.web_2.id
28   port             = 80
29   depends_on       = [aws_instance.web_1]
30 }
31
32
33
34
35 resource "aws_lb_listener" "frontend" {
36   load_balancer_arn = aws_lb.external_lb.arn
37   port              = 80
38   protocol          = "HTTP"
39   default_action {
40     type = "forward"
41     target_group_arn = aws_lb_target_group.target_alb.arn
42   }
43 }
44
```

```

ec2-user@ip-172-31-6-110:~/terraform
Plan: 5 to add, 0 to change, 0 to destroy.

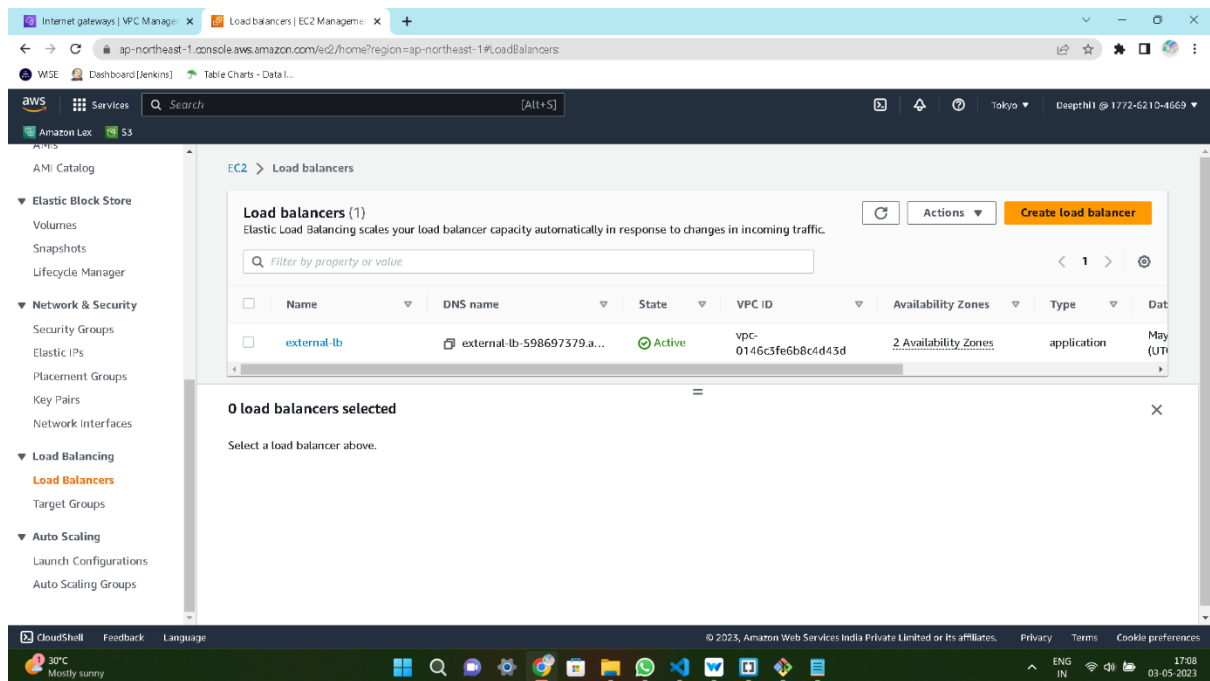
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_lb_target_group.target_alb: Creating...
aws_lb_external_lb: Creating...
aws_lb_target_group.target_alb: Creation complete after 1s [id=arn:aws:elasticloadbalancing:ap-northeast-1:177262104669:targetgroup/target-alb/04953d56abe9688b]
aws_lb_target_group_attachment.attachment2: Creating...
aws_lb_target_group_attachment.attachment: Creation complete after 0s [id=arn:aws:elasticloadbalancing:ap-northeast-1:177262104669:targetgroup/target-alb/04953d56abe9688b-20230503105350879100000001]
aws_lb_target_group_attachment.attachment2: Creation complete after 0s [id=arn:aws:elasticloadbalancing:ap-northeast-1:177262104669:targetgroup/target-alb/04953d56abe9688b-20230503105350922600000002]
aws_lb_external_lb: Still creating... [10s elapsed]
aws_lb_external_lb: Still creating... [20s elapsed]
aws_lb_external_lb: Still creating... [30s elapsed]
aws_lb_external_lb: Still creating... [40s elapsed]
aws_lb_external_lb: Still creating... [50s elapsed]
aws_lb_external_lb: Still creating... [1m0s elapsed]
aws_lb_external_lb: Still creating... [1m10s elapsed]
aws_lb_external_lb: Still creating... [1m20s elapsed]
aws_lb_external_lb: Still creating... [1m30s elapsed]
aws_lb_external_lb: Still creating... [1m40s elapsed]
aws_lb_external_lb: Still creating... [1m50s elapsed]
aws_lb_external_lb: Still creating... [2m0s elapsed]
aws_lb_external_lb: Creation complete after 2m2s [id=arn:aws:elasticloadbalancing:ap-northeast-1:177262104669:loadbalancer/app/external-lb/3ad0f10be1e655af]
aws_lb_listener.frontend: Creating...
aws_lb_listener.frontend: Creation complete after 0s [id=arn:aws:elasticloadbalancing:ap-northeast-1:177262104669:listener/app/external-lb/3ad0f10be1e655af/9fa36cd1d836f033]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$

```



Step10: Create a file for RDS Instance

```

1  #creating rds instance
2  resource "aws_db_subnet_group" "rds" {
3      name             = "main"
4      subnet_ids       = [aws_subnet.dbs-1.id,aws_subnet.dbs-2.id]
5      tags              = {
6          Name          = "my DB subnet group"
7      }
8  }
9
10 resource "aws_db_instance" "db_server" {
11     allocated_storage = 10
12     db_subnet_group_name = aws_db_subnet_group.rds.id
13     engine             = "mysql"
14     engine_version     = "5.7"
15     instance_class     = "db.t2.micro"
16     multi_az           = true
17     username           = "username"
18     password           = "password"
19     skip_final_snapshot = true
20     vpc_security_group_ids = [aws_security_group.wsg.id]
21 }

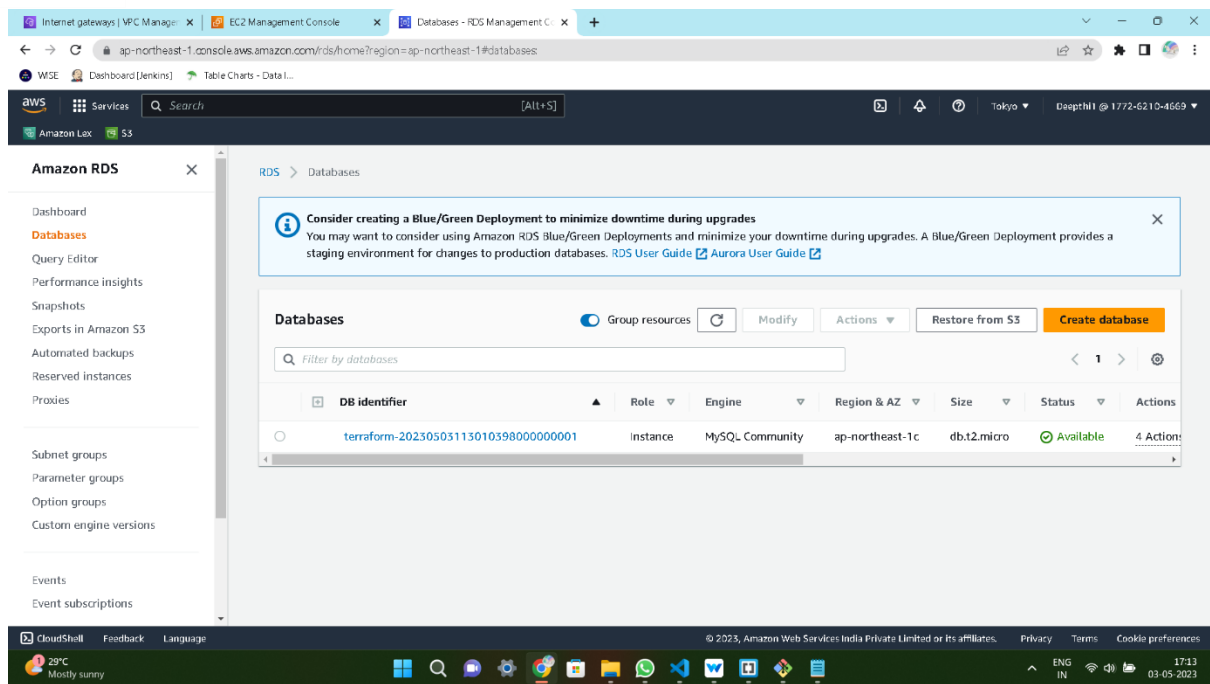
```

```

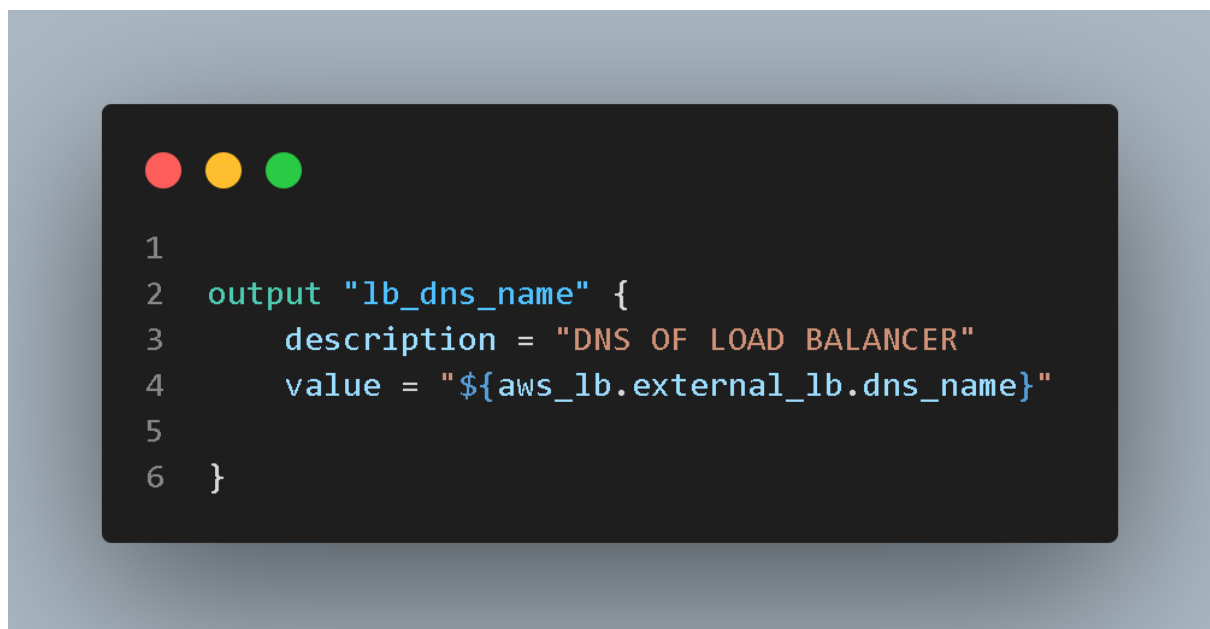
ec2-user@ip-172-31-6-110:~/terraform
aws_db_instance.db_server: Still creating... [5m40s elapsed]
aws_db_instance.db_server: Still creating... [5m50s elapsed]
aws_db_instance.db_server: Still creating... [6m0s elapsed]
aws_db_instance.db_server: Still creating... [6m10s elapsed]
aws_db_instance.db_server: Still creating... [6m20s elapsed]
aws_db_instance.db_server: Still creating... [6m30s elapsed]
aws_db_instance.db_server: Still creating... [6m40s elapsed]
aws_db_instance.db_server: Still creating... [6m50s elapsed]
aws_db_instance.db_server: Still creating... [7m0s elapsed]
aws_db_instance.db_server: Still creating... [7m10s elapsed]
aws_db_instance.db_server: Still creating... [7m20s elapsed]
aws_db_instance.db_server: Still creating... [7m30s elapsed]
aws_db_instance.db_server: Still creating... [7m40s elapsed]
aws_db_instance.db_server: Still creating... [7m50s elapsed]
aws_db_instance.db_server: Still creating... [8m0s elapsed]
aws_db_instance.db_server: Still creating... [8m10s elapsed]
aws_db_instance.db_server: Still creating... [8m20s elapsed]
aws_db_instance.db_server: Still creating... [8m30s elapsed]
aws_db_instance.db_server: Still creating... [8m40s elapsed]
aws_db_instance.db_server: Still creating... [8m50s elapsed]
aws_db_instance.db_server: Still creating... [9m0s elapsed]
aws_db_instance.db_server: Still creating... [9m10s elapsed]
aws_db_instance.db_server: Still creating... [9m20s elapsed]
aws_db_instance.db_server: Still creating... [9m30s elapsed]
aws_db_instance.db_server: Still creating... [9m40s elapsed]
aws_db_instance.db_server: Still creating... [9m50s elapsed]
aws_db_instance.db_server: Still creating... [10m0s elapsed]
aws_db_instance.db_server: Still creating... [10m10s elapsed]
aws_db_instance.db_server: Still creating... [10m20s elapsed]
aws_db_instance.db_server: Still creating... [10m30s elapsed]
aws_db_instance.db_server: Still creating... [10m40s elapsed]
aws_db_instance.db_server: Still creating... [10m50s elapsed]
aws_db_instance.db_server: Still creating... [11m0s elapsed]
aws_db_instance.db_server: Still creating... [11m10s elapsed]
aws_db_instance.db_server: Still creating... [11m20s elapsed]
aws_db_instance.db_server: Creation complete after 11m30s [id=terraform-20230503113010398000000001]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$

```



Step11: Create a file for Outputs



Step12: Create a file for Variable



```
1  # defining cidr block for vpc
2
3  variable "vpc_cidr" {
4      default = "10.0.0.0/16"
5
6  }
7
8
9  variable "websub_1" {
10     default = "10.0.0.0/18"
11
12 }
13
14
15 variable "websub_2" {
16     default = "10.0.128.0/18"
17
18 }
19
20
21 variable "appsub_1" {
22     default = "10.0.96.0/20"
23
24 }
25
26
27 variable "appsub_2" {
28     default = "10.0.112.0/20"
29
30 }
31
32 variable "dbsub_1" {
33     default = "10.0.80.0/21"
34
35 }
36
37 variable "dbsub_2" {
38     default = "10.0.88.0/21"
39
40 }
41
42
```

Step13: Create a file for User Data

```

1  #!/bin/bash
2  sudo yum update -y
3  sudo yum install httpd -y
4  sudo yum install git -y
5  sudo systemctl start httpd
6  sudo systemctl enable httpd
7  echo "hello world from $(hostname -f)" > /var/www/html/index.html

```

Step14: Verify the Resources

```

apply complete! Resources: 17 added, 0 changed, 0 destroyed.

Outputs:

```

```

ec2-user@ip-172-31-6-110:~$ terraform
aws_db_instance.db_server: Still destroying... [id=terraform-20230503113010398000000001, 5m40s elapsed]
aws_db_instance.db_server: Still destroying... [id=terraform-20230503113010398000000001, 5m50s elapsed]
aws_db_instance.db_server: Still destroying... [id=terraform-20230503113010398000000001, 6m0s elapsed]
aws_db_instance.db_server: Still destroying... [id=terraform-20230503113010398000000001, 6m10s elapsed]
aws_db_instance.db_server: Still destroying... [id=terraform-20230503113010398000000001, 6m20s elapsed]
aws_db_instance.db_server: Still destroying... [id=terraform-20230503113010398000000001, 6m30s elapsed]
aws_db_instance.db_server: Still destroying... [id=terraform-20230503113010398000000001, 6m40s elapsed]
aws_db_instance.db_server: Destruction complete after 6m45s
aws_security_group.wsg: Destroying... [id=sg-0ce4afe0573elc54e]
aws_db_subnet_group.rds: Destroying... [id=main]
aws_db_subnet_group.rds: Destruction complete after 0s
aws_subnet.dbs-1: Destroying... [id=subnet-07e2bff3929b0bf7d]
aws_subnet.dbs-2: Destroying... [id=subnet-0db4865653fe84c07]
aws_subnet.dbs-2: Destruction complete after 0s
aws_subnet.dbs-1: Destruction complete after 0s
aws_security_group.wsg: Destruction complete after 1s
aws_vpc.vpc_deepu: Destroying... [id=vpc-0146c3fe6b8c4d43d]
aws_vpc.vpc_deepu: Destruction complete after 0s

Destroy complete! Resources: 23 destroyed.
[ec2-user@ip-172-31-6-110 terraform]$ 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_db_instance.db_server will be created
+ resource "aws_db_instance" "db_server" {
  + address                        = (known after apply)
  + allocated_storage             = 10
  + apply_immediately            = false
  + arn                          = (known after apply)
  + auto_minor_version_upgrade   = true
  + availability_zone            = (known after apply)
  + backup_retention_period       = (known after apply)
  + backup_window                = (known after apply)
  + ca_cert_identifier           = (known after apply)

```

- All the resources like vpc, subnets, internet gateway, route table, ec2 instances, rds, alb, security groups have been applied by terraform and verified as shown in above screenshot.