PROJECT-1

Deploy Three-Tier architcure

AWS - using - Terraform

- What is Terraform?

- Terraform is an open-source infrastructure as a code (IAC) tool that allows to create, manage & deploy the production-ready environment. Terraform codifies cloud APIs into declarative configuration files. Terraform can manage both existing service providers and custom in-house solutions

.In this project, we will deploy a three-tier application in AWS

using Terraform.

- Prerequisites:

1. go to aws console and log into rootuser

2. go to IAMdashboard copy iam user id.

3.log out to root user.

4.log into IAM user phaste IAM id and give username and password and click on sign in.

5. go to ec2 instances create a instance.

6.copy ssh client.

7. cd downloads/ and phaste to terminal and connect.

8. goto hashicrop and clickon amazon linux installing terraform copy that commands .

9.phaste to terminal one by one.and install terraform.

10. then create a mkdir file.

11.create provider file and give access key&secret key

STEP - 1

- create a file for vpc .

- give terraform init,terraform validate,terraform
- apply

- go to vpcs and check created.

STEP - 2

- create a file for subnets.

1.create public subnets.

```
resource "aws_subnet" "public-3" {
  vpc_id                  = aws_vpc.vpc1.id
  map_public_ip_on_launch = true
  availability_zone       = "us-west-1b"
  cidr_block              = "10.0.208.0/20"

  tags = {
    name = "public-3"
  }
}

resource "aws_subnet" "public-2" {
  vpc_id                  = aws_vpc.vpc1.id
  map_public_ip_on_launch = true
  availability_zone       = "us-west-1b"
  cidr_block              = "10.0.224.0/19"

  tags = {
    name = "public-2"
  }
}
~
~
~
```

- give terraform init,terraform validate,terraform apply

- goto vpc subnets and to see created.

STEP - 3

- create a file for internet gateway.

- give terraform init,terraform validate,terraform apply

- go to internet gateways to see created

STEP - 4

- create a file for route table.

1. create a route table.

```
resource "aws_route_table" "myroute" {
  vpc_id = aws_vpc.vpc1.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }
  tags = {
    Name = "myroute"
  }
}
-- INSERT --                                                    1,1           All
```

2.create route table association file.

```
resource "aws_route_table_association" "route-a" {
  subnet_id      = aws_subnet.public-3.id
  route_table_id = aws_route_table.myroute.id
}

resource "aws_route_table_association" "route-b" {
  subnet_id      = aws_subnet.pvt-1.id
  route_table_id = aws_route_table.myroute.id
}
-- INSERT --                                                    7,46          All
```

- give terraform init,terraform validate,terraform apply

- goto vpc route tables and see created.

STEP - 5

- create a file for EC2 two instances

```
resource "aws_instance" "terraform2" {
  ami                          = "ami-00baaa26c63140022"
  instance_type                = "t2.micro"
  availability_zone            = "us-west-1b"
  key_name                     = "nani"
  vpc_security_group_ids       = [aws_security_group.vamsi.id]
  subnet_id                    = aws_subnet.public-3.id
  associate_public_ip_address = true
  user_data                   = "${file("userdata.sh")}"
  tags = {
    name = "terraform2"
  }
}

resource "aws_instance" "terraform3" {
  ami                          = "ami-00baaa26c63140022"
  instance_type                = "t2.micro"
  availability_zone            = "us-west-1b"
  key_name                     = "nani"
  vpc_security_group_ids       = [aws_security_group.vamsi.id]
  subnet_id                    = aws_subnet.public-3.id
  associate_public_ip_address = true
  user_data                   = "${file("userdata.sh")}"
  tags = {
    name = "terraform3"
  }
}
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    4,42          All
```

- give terraform init,terraform validate,terraform apply

- goto EC2 instances and see created.

STEP - 6

- create a file for security group.

- give terraform init,terraform validate,terraform apply

- goto security groups and see created

STEP - 7

- create a file for database security groups.

```
resource "aws_security_group" "vamsi2" {
  name        = "vamsi2"
  description = "Allow inbound traffic from application layer"
  vpc_id      = aws_vpc.vpc1.id

  ingress {
    description = "Allow inbound traffic from application layer"
    from_port   = 3306
    to_port     = 3306
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 32768
    to_port     = 65535
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    name = "vamsi2"
  }
}
-- INSERT --                                                    4,1           All
```

- give terraform init,terraform validate,terraform apply

- goto security groups and see created.

STEP - 8

- create a file for application load balancer

1.create a load balancer & target group.

```
resource "aws_lb" "my_lb" {
  internal           = false
  load_balancer_type = "application"
  subnets            = [aws_subnet.pvt-1.id, aws_subnet.public-3.id]

}

resource "aws_lb_target_group" "my-tg" {
  name     = "mytg"
  port     = 80
  protocol = "HTTP"
  vpc_id   = aws_vpc.vpc1.id
}

-- INSERT --                                                    12,20         All
```

2.create two attatchments with target group.

```
resource "aws_lb_target_group_attachment" "attatchment1" {
  target_group_arn = aws_lb_target_group.my-tg.arn
  target_id        = aws_instance.terraform2.id
  port             = 80
  depends_on       = [aws_instance.terraform2]
}

resource "aws_lb_target_group_attachment" "attatchment2" {
  target_group_arn = aws_lb_target_group.my-tg.arn
  target_id        = aws_instance.terraform3.id
  port             = 80
  depends_on       = [aws_instance.terraform3]
}

-- INSERT --                                                    15,1          All
```

3.create a listener file for load balancer.

- give terraform init,terraform validate,terraform apply

- goto target groups to see created.

- goto load balancer to see created.

STEP - 9

- create a file RDS database

1.create a rds file

2.create a rds subnet group file

- give terraform init,terraform validate,terraform apply

- goto RDS databases to see created.

- goto rds subnet groups to see created.

STEP - 10

- create a file for outputs

```
output "lb_dns_name" {
  description = "arn:aws:elasticloadbalancing:us-west-1:193633228050:loadbalancer/app/my-lb/dec0a768b90b4dee"
  value       = "aws_lb.my-lb.dns_my-lb-2062538271.us-west-1.elb.amazonaws.com"
}
~
~
~
```

STEP - 11

- create a file for variables

```
variable "vpc_cidr" {
  default = "10.0.0.0/16"
}

variable "subnet_cidr" {
  default = "10.0.1.0/24"
}

variable "subnet2_cidr" {
  default = "10.0.2.0/24"
}

variable "subnet3_cidr" {
  default = "10.0.3.0/24"
}
variable "subnet4_cidr" {
  default = "10.0.4.0/24"
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    1,1           All
```

STEP - 12

- create a file for userdata(create .sh file)

```
#!/bin/bash
sudo yum update -y
sudo yum -y install git
sudo yum -y install httpd
sudo systemctl start httpd
sudo systemctl enable httpd
echo "hello world from $(hostname -f)" > /var/www/html/index.html
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"userdata.sh" 7L, 203B                                            7,66          All
```

STEP - 13

- verify the resources
-
- After creation give (terraform apply)

```
aws_db_instance.rds: Still creating... [8m40s elapsed]
aws_db_instance.rds: Still creating... [8m50s elapsed]
aws_db_instance.rds: Still creating... [9m0s elapsed]
aws_db_instance.rds: Still creating... [9m10s elapsed]
aws_db_instance.rds: Still creating... [9m20s elapsed]
aws_db_instance.rds: Still creating... [9m30s elapsed]
aws_db_instance.rds: Still creating... [9m40s elapsed]
aws_db_instance.rds: Still creating... [9m50s elapsed]
aws_db_instance.rds: Still creating... [10m0s elapsed]
aws_db_instance.rds: Still creating... [10m10s elapsed]
aws_db_instance.rds: Still creating... [10m20s elapsed]
aws_db_instance.rds: Still creating... [10m30s elapsed]
aws_db_instance.rds: Still creating... [10m40s elapsed]
aws_db_instance.rds: Still creating... [10m50s elapsed]
aws_db_instance.rds: Still creating... [11m0s elapsed]
aws_db_instance.rds: Still creating... [11m10s elapsed]
aws_db_instance.rds: Still creating... [11m20s elapsed]
aws_db_instance.rds: Still creating... [11m30s elapsed]
aws_db_instance.rds: Still creating... [11m40s elapsed]
aws_db_instance.rds: Still creating... [11m50s elapsed]
aws_db_instance.rds: Still creating... [12m0s elapsed]
aws_db_instance.rds: Still creating... [12m10s elapsed]
aws_db_instance.rds: Still creating... [12m20s elapsed]
aws_db_instance.rds: Still creating... [12m30s elapsed]
aws_db_instance.rds: Still creating... [12m40s elapsed]
aws_db_instance.rds: Still creating... [12m50s elapsed]
aws_db_instance.rds: Still creating... [13m0s elapsed]
aws_db_instance.rds: Creation complete after 13m1s [id=db-A4KNJUIBF2CNJN7CVORS2MUVMU]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

lb_dns_name = "aws_lb.my-lb.dns_my-lb-2062538271.us-west-1.elb.amazonaws.com"
[ec2-user@ip-172-31-24-188 terraform-code]$ client_loop: send disconnect: Connection reset
PS C:\Users\gandh\Downloads>
```

- to see the dns output.

```
aws_db_instance.rds: Still creating... [13m0s elapsed]
aws_db_instance.rds: Creation complete after 13m1s [id=db-A4KNJUIBF2CNJN7CVORS2MUVMU]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

lb_dns_name = "aws_lb.my-lb.dns_my-lb-2062538271.us-west-1.elb.amazonaws.com"
[ec2-user@ip-172-31-24-188 terraform-code]$ client_loop: send disconnect: Connection reset
```

- browse it and to see web page.

Hello World from ip-10-0-2-189.eu-north-1.compute.internal

Then push all the code in git hub

GIT: git is a version comtrol system it stores coding in remote repositary.

1. create a git hub repositary in github

- clone the code

- add to terminal(git remote add origin "phaste")

```
[ec2-user@ip-172-31-24-188 terraform-code]$
[ec2-user@ip-172-31-24-188 terraform-code]$ git status
On branch terraform
nothing to commit, working tree clean
[ec2-user@ip-172-31-24-188 terraform-code]$
[ec2-user@ip-172-31-24-188 terraform-code]$
[ec2-user@ip-172-31-24-188 terraform-code]$ git remote add origin https://github.com/krishnanaidu99/terraform-codes.git
[ec2-user@ip-172-31-24-188 terraform-code]$ git remote -v
origin  https://github.com/krishnanaidu99/terraform-codes.git (fetch)
origin  https://github.com/krishnanaidu99/terraform-codes.git (push)
```

```
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$ git branch
* terraform
[ec2-user@ip-172-31-24-188 terraform]$ git remote -v
origin  https://github.com/krishnanaidu99/terraform-codes.git (fetch)
origin  https://github.com/krishnanaidu99/terraform-codes.git (push)
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$ ls
databasesg.tf  ec2.tf  id.tf  igw.tf  lb.tf  lbattat.tf  op.tf  provider.tf  pubsn.tf  pvtsn.tf  pvtsub.tf  rds.tf  rdssn.tf  rt.tf  rtass.tf  sg.tf  userdata.sh  variable.tf  vpc.tf
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$ cd
[ec2-user@ip-172-31-24-188 ~]$ lsd
-bash: lsd: command not found
[ec2-user@ip-172-31-24-188 ~]$ ld
ld: no input files
[ec2-user@ip-172-31-24-188 ~]$ ls
terraform  terraform2  vpc.tf
[ec2-user@ip-172-31-24-188 ~]$ cd terraform2/
[ec2-user@ip-172-31-24-188 terraform2]$ ls
[ec2-user@ip-172-31-24-188 terraform2]$ cd
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$ rm -rf terraform2/
[ec2-user@ip-172-31-24-188 ~]$ ls
terraform  vpc.tf
[ec2-user@ip-172-31-24-188 ~]$ rm -rf vpc.tf
[ec2-user@ip-172-31-24-188 ~]$ ls
terraform
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$ cd terraform/
[ec2-user@ip-172-31-24-188 terraform]$ ls
databasesg.tf  ec2.tf  id.tf  igw.tf  lb.tf  lbattat.tf  op.tf  provider.tf  pubsn.tf  pvtsn.tf  pvtsub.tf  rds.tf  rdssn.tf  rt.tf  rtass.tf  sg.tf  userdata.sh  variable.tf  vpc.tf
[ec2-user@ip-172-31-24-188 terraform]$ cd
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$ mkdir terraform-code
```

- check status .

```
ec2-user@ip-172-31-24-188:~/terraform-code
databasesg.tf  ec2.tf  id.tf  igw.tf  lb.tf  lbattat.tf  op.tf  provider.tf  pubsn.tf  pvtsn.tf  pvtsub.tf  rds.tf  rdssn.tf  rt.tf  rtass.tf  sg.tf  userdata.sh  variable.tf  vpc.tf
[ec2-user@ip-172-31-24-188 terraform]$ sudo mv databasesg.tf ec2.tf id.tf lb.tf lbattat.tf op.tf provider.tf pubsn.tf pvtsn.tf pvtsub.tf rdssn.tf rds.tf rtass.tf rt.tf sg.tf userdata.sh variable.tf vpc.tf /home
/ec2-user/terraform-code/
[ec2-user@ip-172-31-24-188 terraform]$ ls
igw.tf
[ec2-user@ip-172-31-24-188 terraform]$
[ec2-user@ip-172-31-24-188 terraform]$ cd
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$
[ec2-user@ip-172-31-24-188 ~]$ cd terraform-code/
[ec2-user@ip-172-31-24-188 terraform-code]$ ls
databasesg.tf  ec2.tf  id.tf  lb.tf  lbattat.tf  op.tf  provider.tf  pubsn.tf  pvtsn.tf  pvtsub.tf  rds.tf  rdssn.tf  rt.tf  rtass.tf  sg.tf  userdata.sh  variable.tf  vpc.tf
[ec2-user@ip-172-31-24-188 terraform-code]$
[ec2-user@ip-172-31-24-188 terraform-code]$ git init
Initialized empty Git repository in /home/ec2-user/terraform-code/.git/
[ec2-user@ip-172-31-24-188 terraform-code]$
[ec2-user@ip-172-31-24-188 terraform-code]$
[ec2-user@ip-172-31-24-188 terraform-code]$
[ec2-user@ip-172-31-24-188 terraform-code]$ git status
On branch terraform

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        databasesg.tf
        ec2.tf
        id.tf
        lb.tf
        lbattat.tf
        op.tf
        provider.tf
        pubsn.tf
        pvtsn.tf
        pvtsub.tf
        rds.tf
        rdssn.tf
        rt.tf
        rtass.tf
        sg.tf
        userdata.sh
        variable.tf
        vpc.tf

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-24-188 terraform-code]$ git add .
[ec2-user@ip-172-31-24-188 terraform-code]$ git commit -m "files"
[terraform (root-commit) fecec75] files
 Committer: EC2 Default User <ec2-user@ip-172-31-24-188.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:
```

- ADD files (git add .)
- commit changes (git commit -m"hii")
- push (git push -u origin "filename")

- goto github and open repo and see all files

- clone the code (copy client)

- clone the repo in terminal (sudo git clone "phaste")

- Then use comands 1.terraform init ,2.terraform validate

```
[ec2-user@ip-172-31-24-188 terraform-code]$ sudo git clone https://github.com/krishnanaidu99/terraform-codes.git
Cloning into 'terraform-codes'...
Username for 'https://github.com': krishnanaidu99
Password for 'https://krishnanaidu99@github.com':
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 22 (delta 3), reused 21 (delta 2), pack-reused 0
Receiving objects: 100% (22/22), done.
Resolving deltas: 100% (3/3), done.
[ec2-user@ip-172-31-24-188 terraform-code]$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.3.0

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-172-31-24-188 terraform-code]$ terraform validate
Success! The configuration is valid.

[ec2-user@ip-172-31-24-188 terraform-code]$ 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_db_instance.rds will be created
  + resource "aws_db_instance" "rds" {
```

3.terraform plan

```
[ec2-user@ip-172-31-24-188 terraform-code]$ 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_db_instance.rds will be created
  + resource "aws_db_instance" "rds" {
      + address                               = (known after apply)
      + allocated_storage                     = 8
      + 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)
      + character_set_name                    = (known after apply)
      + copy_tags_to_snapshot                 = false
      + db_name                               = "rds"
      + db_subnet_group_name                  = (known after apply)
      + delete_automated_backups              = true
      + endpoint                              = (known after apply)
      + engine                                = "mysql"
      + engine_version                        = "8.0.28"
      + engine_version_actual                 = (known after apply)
      + hosted_zone_id                        = (known after apply)
      + id                                    = (known after apply)
      + identifier                            = (known after apply)
      + identifier_prefix                     = (known after apply)
      + instance_class                        = "db.t2.micro"
      + iops                                  = (known after apply)
      + kms_key_id                            = (known after apply)
      + latest_restorable_time                = (known after apply)
      + license_model                         = (known after apply)
      + listener_endpoint                     = (known after apply)
      + maintenance_window                    = (known after apply)
```

4.terraform apply

```
[ec2-user@ip-172-31-24-188 terraform-code]$ terraform apply
aws_vpc.vpc1: Refreshing state... [id=vpc-00a414d08949887b7]
aws_subnet.pvt-2: Refreshing state... [id=subnet-0f3eccb10dd45702a]
aws_subnet.public-2: Refreshing state... [id=subnet-0a0e620cfaa22c706]
aws_lb_target_group.my-tg: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:193633228050:targetgroup/mytg/5facce73c6d0415e]
aws_internet_gateway.gw: Refreshing state... [id=igw-02bb92c42fd82c979]
aws_subnet.public-3: Refreshing state... [id=subnet-07efe2a09da6b05c5]
aws_security_group.vamsi: Refreshing state... [id=sg-0ba447f885575bb73]
aws_security_group.vamsi2: Refreshing state... [id=sg-043ed1f188ad99c7e]
aws_subnet.pvt-1: Refreshing state... [id=subnet-0d4049f7c785ea770]
aws_route_table.myroute: Refreshing state... [id=rtb-0ca185564812af6eb]
aws_lb.my_lb: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:193633228050:loadbalancer/app/tf-lb-20230615172047247900000002/95d3307d2480970b
]
aws_instance.terraform2: Refreshing state... [id=i-06bce27c1b4a7ac7c]
aws_instance.terraform3: Refreshing state... [id=i-032c04996eb7c29bc]
aws_route_table_association.route-a: Refreshing state... [id=rtbassoc-0bfbe55d1c516579a]
aws_lb_listener.my_lis: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:193633228050:listener/app/tf-lb-20230615172047247900000002/95d3307d24
80970b/79f650094c05ed82]
aws_lb_target_group_attachment.attatchment1: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:193633228050:targetgroup/mytg/5facce73c6d0415e-2
0230615172119363200000004]
aws_lb_target_group_attachment.attatchment2: Refreshing state... [id=arn:aws:elasticloadbalancing:us-west-1:193633228050:targetgroup/mytg/5facce73c6d0415e-2
0230615172119524000000005]

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.rds will be created
  + resource "aws_db_instance" "rds" {
      + address                               = (known after apply)
      + allocated_storage                     = 8
      + 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)
```

```
aws_db_instance.rds: Still creating... [8m10s elapsed]
aws_db_instance.rds: Still creating... [8m20s elapsed]
aws_db_instance.rds: Still creating... [8m30s elapsed]
aws_db_instance.rds: Still creating... [8m40s elapsed]
aws_db_instance.rds: Still creating... [8m50s elapsed]
aws_db_instance.rds: Still creating... [9m0s elapsed]
aws_db_instance.rds: Still creating... [9m10s elapsed]
aws_db_instance.rds: Still creating... [9m20s elapsed]
aws_db_instance.rds: Still creating... [9m30s elapsed]
aws_db_instance.rds: Still creating... [9m40s elapsed]
aws_db_instance.rds: Still creating... [9m50s elapsed]
aws_db_instance.rds: Still creating... [10m0s elapsed]
aws_db_instance.rds: Still creating... [10m10s elapsed]
aws_db_instance.rds: Still creating... [10m20s elapsed]
aws_db_instance.rds: Still creating... [10m30s elapsed]
aws_db_instance.rds: Still creating... [10m40s elapsed]
aws_db_instance.rds: Still creating... [10m50s elapsed]
aws_db_instance.rds: Still creating... [11m0s elapsed]
aws_db_instance.rds: Still creating... [11m10s elapsed]
aws_db_instance.rds: Still creating... [11m20s elapsed]
aws_db_instance.rds: Still creating... [11m30s elapsed]
aws_db_instance.rds: Still creating... [11m40s elapsed]
aws_db_instance.rds: Still creating... [11m50s elapsed]
aws_db_instance.rds: Still creating... [12m0s elapsed]
aws_db_instance.rds: Still creating... [12m10s elapsed]
aws_db_instance.rds: Still creating... [12m20s elapsed]
aws_db_instance.rds: Still creating... [12m30s elapsed]
aws_db_instance.rds: Still creating... [12m40s elapsed]
aws_db_instance.rds: Still creating... [12m50s elapsed]
aws_db_instance.rds: Still creating... [13m0s elapsed]
aws_db_instance.rds: Creation complete after 13m1s [id=db-A4KNJUIBF2CNJN7CVORS2MUVMU]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

lb_dns_name = "aws_lb.my-lb.dns_my-lb-2062538271.us-west-1.elb.amazonaws.com"
[ec2-user@ip-172-31-24-188 terraform-code]$ client_loop: send disconnect: Connection reset
PS C:\Users\gandh\Downloads> cd .\Downloads
```

- To installed all the files check your console
- browse the resulte dns output to see webpage

```
aws_db_instance.rds: Still creating... [13m0s elapsed]
aws_db_instance.rds: Creation complete after 13m1s [id=db-A4KNJUIBF2CNJN7CVORS2MUVMU]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

lb_dns_name = "aws_lb.my-lb.dns_my-lb-2062538271.us-west-1.elb.amazonaws.com"
[ec2-user@ip-172-31-24-188 terraform-code]$ client_loop: send disconnect: Connection reset
PS C:\Users\gandh\Downloads>
```

Hello World from ip-10-0-2-189.eu-north-1.compute.internal

Then deploy the terraform repository using Jenkins

jenkins: jenkins is a automation tool to deploy code and to install servers using "excute shell" in a job and build the code then browse the ipv4 adress tosee out put.

1. create a ec2 instance

2.connect to termainal

```
ubuntu@ip-172-31-20-175: ~
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\gandh> cd .\Downloads\
PS C:\Users\gandh\Downloads> ssh -i "nani.pem" ubuntu@ec2-204-236-197-255.compute-1.amazonaws.com
The authenticity of host 'ec2-204-236-197-255.compute-1.amazonaws.com (204.236.197.255)' can't be established.
ED25519 key fingerprint is SHA256:tOo3oKCXFGSNJFs6mRerS3DCNue27WrPMiFjXPyMohc.
This host key is known by the following other names/addresses:
    C:\Users\gandh/.ssh/known_hosts:163: ec2-54-204-96-22.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-204-236-197-255.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Jun 19 14:38:24 UTC 2023

  System load:  0.080078125       Processes:             98
  Usage of /:   44.2% of 7.57GB   Users logged in:       0
  Memory usage: 51%               IPv4 address for eth0: 172.31.20.175
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

65 updates can be applied immediately.
46 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

2 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Mon Jun 19 04:42:23 2023 from 103.213.202.30
ubuntu@ip-172-31-20-175:~$
```

3. install java and jenkins.

```
ubuntu@ip-172-31-20-175: ~   ec2-user@ip-172-31-27-166:~

/etc/yum.repos.d/jenkins.repo          100%[===================================================================>]      85  --.-KB/s    in 0s

2023-06-19 14:52:33 (5.57 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[ec2-user@ip-172-31-27-166 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[ec2-user@ip-172-31-27-166 ~]$ sudo yum upgrade
Jenkins-stable                                                                                    228 kB/s |  26 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-27-166 ~]$ sudo yum install java-11-openjdk
Last metadata expiration check: 0:00:16 ago on Mon Jun 19 14:53:09 2023.
No match for argument: java-11-openjdk
Error: Unable to find a match: java-11-openjdk
[ec2-user@ip-172-31-27-166 ~]$ java
-bash: java: command not found
[ec2-user@ip-172-31-27-166 ~]$ sudo yum install jenkins
Last metadata expiration check: 0:00:52 ago on Mon Jun 19 14:53:09 2023.
Dependencies resolved.
================================================================================================================================
 Package                    Architecture                Version                         Repository                         Size
================================================================================================================================
Installing:
 jenkins                    noarch                      2.401.1-1.1                     jenkins                            94 M

Transaction Summary
================================================================================================================================
Install  1 Package

Total download size: 94 M
Installed size: 94 M
Is this ok [y/N]: y
Downloading Packages:
jenkins-2.401.1-1.1.noarch.rpm                                                                     34 MB/s |  94 MB     00:02
--------------------------------------------------------------------------------------------------------------------------------
Total                                                                                              34 MB/s |  94 MB     00:02
Running transaction check
Transaction check succeeded.
Running transaction test
```

```
[ec2-user@ip-172-31-27-166 ~]$ sudo yum install java-11-openjdk
Last metadata expiration check: 0:01:50 ago on Mon Jun 19 14:53:09 2023.
No match for argument: java-11-openjdk
Error: Unable to find a match: java-11-openjdk
[ec2-user@ip-172-31-27-166 ~]$ sudo yum install java-11*jdk
Last metadata expiration check: 0:02:01 ago on Mon Jun 19 14:53:09 2023.
No match for argument: java-11*jdk
Error: Unable to find a match: java-11*jdk
[ec2-user@ip-172-31-27-166 ~]$ sudo yum install java-11*
Last metadata expiration check: 0:02:13 ago on Mon Jun 19 14:53:09 2023.
Dependencies resolved.
================================================================================================================================
 Package                          Architecture        Version                                  Repository              Size
================================================================================================================================
Installing:
 java-11-amazon-corretto          x86_64              1:11.0.19+7-1.amzn2023                   amazonlinux            198 k
 java-11-amazon-corretto-devel    x86_64              1:11.0.19+7-1.amzn2023                   amazonlinux            219 k
 java-11-amazon-corretto-javadoc  x86_64              1:11.0.19+7-1.amzn2023                   amazonlinux             13 M
 java-11-amazon-corretto-jmods    x86_64              1:11.0.19+7-1.amzn2023                   amazonlinux             71 M
Installing dependencies:
 alsa-lib                         x86_64              1.2.7.2-1.amzn2023.0.2                   amazonlinux            504 k
 cairo                            x86_64              1.17.4-3.amzn2023.0.2                    amazonlinux            674 k
 dejavu-sans-fonts                noarch              2.37-16.amzn2023.0.2                     amazonlinux            1.3 M
 dejavu-sans-mono-fonts           noarch              2.37-16.amzn2023.0.2                     amazonlinux            467 k
 dejavu-serif-fonts               noarch              2.37-16.amzn2023.0.2                     amazonlinux            1.0 M
 fontconfig                       x86_64              2.13.94-2.amzn2023.0.2                   amazonlinux            273 k
 fonts-filesystem                 noarch              1:2.0.5-5.amzn2023.0.2                   amazonlinux            8.7 k
 freetype                         x86_64              2.13.0-2.amzn2023.0.1                    amazonlinux            422 k
 giflib                           x86_64              5.2.1-9.amzn2023                         amazonlinux             49 k
 google-noto-fonts-common         noarch              20201206-2.amzn2023.0.2                  amazonlinux             15 k
 google-noto-sans-vf-fonts        noarch              20201206-2.amzn2023.0.2                  amazonlinux            492 k
 graphite2                        x86_64              1.3.14-7.amzn2023.0.2                    amazonlinux             97 k
 harfbuzz                         x86_64              7.0.0-2.amzn2023.0.1                     amazonlinux            868 k
 java-11-amazon-corretto-headless x86_64              1:11.0.19+7-1.amzn2023                   amazonlinux             91 M
 javapackages-filesystem          noarch              6.0.0-7.amzn2023.0.5                     amazonlinux             13 k
 langpacks-core-font-en           noarch              3.0-21.amzn2023.0.4                      amazonlinux             10 k
 libICE                           x86_64              1.0.10-6.amzn2023.0.2                    amazonlinux             71 k
 libSM                            x86_64              1.2.3-8.amzn2023.0.2                     amazonlinux             42 k
 libX11                           x86_64              1.7.2-3.amzn2023.0.2                     amazonlinux            657 k
 libX11-common                    noarch              1.7.2-3.amzn2023.0.2                     amazonlinux            152 k
```

4.goto ec2 sequirity groups and edit Enbound roules give port 8080 and start&enable jenkins

5.browse the ipv4 adress and assain port (:8080)to display the jenkins page

6. copy the path (sudo cat "phaste)and diaplay password and phaste.

7. install suggested pluggins

8. give a random username and password.

9. create a job using freestyle and give gitrepo url & branch&creaditionals.

10.clock on build steps & select excute shell & give comands to install terraform.

11. Save and apply them click on build now.

12 after success your job then browse the public ip address

Hello World from ip-10-0-2-189.eu-north-1.compute.internal