Cloud Provider:

Amazon Web Services are used to host the infrastructure needed for this project.

Infrastructure As A Code:

Terraform is used to describe the infrastructure and to dynamically create and configure resources on AWS.

Infrastructure Overview:

AWS VPC: It is used to create a virtual private cloud infrastructure for this project.
AWS Internet Gateway: It is used to connect the resources inside the virtual private cloud to the outside internet.
Security Groups: Security Groups and subnets are used to differentiate public and private accessible resources.
Route Table: Is used to map the routes from the internet gateway to the resources accessible to the public internet.
EC2 instances: 2 EC2 compute instances are used for the project

1. For hosting the web application
2. For running the scrapping and the email alert service.

   Both EC2 instance are in public subnet

RDS MySQL: RDS MySQL instance is used to create and maintain the database for this project.

It is in the private subnet and can only be accessible through the EC2 instances created for this project.

Other supporting resources are also used in deriving and maintaining the infrastructure.

Files:

Modules/vpc-double:

main.tf: This file will create the instances of the following resources.

1. AWS VPC
2. AWS Internet Gateway
3. AWS route
4. AWS subnet (1 public, 2 private)
5. Elastic IP
6. AWS NAT gateway
7. AWS Route Table
8. AWS Route Table association for all the 3 subnets.

outputs.tf: This file stores the information about output variables needed for the module.

1. Information about AWS VPC
2. Information about Internet Gateway
3. Information about Subnets (1 public, 2 private)
4. Information about Route Table

variables.tf: This file will have the values for variables used in the main script.

1. Variables for CIDR block
2. Variables for tenancy

3. Variables for vpc name
4. Variables for availability zones.

SRIJAS_AWS:

ec2.tf: This file will create 2 EC2 instances

1. Web server
   The type of instance will be t2.micro.
   The instance will be inside the public subnet created earlier.
   After initialization the shell script will run to configure the server for web hosting.
2. Scrapping server
   The type of instance will be t2.micro.
   The instance will be inside the public subnet created earlier.
   After initialization the shell script will run to configure the server scraping and email service.

rds.tf: This file will create a RDS DB instance and other required resources for RDS.

1. AWS_DB_parameters_group
2. AWS_DB_subnet_group
3. AWS_security_group
4. AWS_DB_instance
   This instance will have following properties
   a. Allocated storage: 10 GB
   b. Storage type: gp2
   c. Engine: MySQL
   d. Engine_version: 8.0
   e. Instance_class: db.t2.micro
   f. Public_accessibility: False

main.tf: This file will define the provider we want to use for this project.
It will also create the backbone infrastructure using the vpc-double module.
AWS will be selected as the provider and region will be us-east-1
The access key and secret key can be fetched from the terraform.tfvars file.

output.tf: This file stores the information about output variables needed.

1. Information about the ec2-web server
2. Information about the ec2-scrapping server
3. Information about the RDS DB instance

variables.tf: This file will have the values for variables used in the main script.

1. Variable for selecting the AWS_region
2. Variable for the ami being used for ec2 instance(region specific)
3. Variable for type of instance
4. The variables needed to be read from terraform.tfvars file

terraform.tfvars (Not committed on github needs to be created by developer):

This file contains the sensitive information about credentials and should be created by the user.
User can follow the format described below for their terraform.tfvars file

aws_access_key = ""

aws_secret_key = ""
db_user_name = ""
db_password = ""

To generate the access key secret key pair, user can follow the link:
https://aws.amazon.com/premiumsupport/knowledge-center/create-access-key/

Installation and Running:

In order to use the infrastructure defined as a code, Terraform must be installed.
Users can follow https://learn.hashicorp.com/tutorials/terraform/install-cli.
Set the path of installed Terraform to environment variables.
Then follow the commands to create the infrastructure on your AWS account.

1. Locate to the Infrastructure/SRIJAS_AWS directory of this project in terminal (cmd).
2. Run “terraform init” (will install required provider for your infrastructure)
3. Run “terraform plan” (will show the changes that will be made)
4. Run “terraform apply” (will ask for confirmation and apply the changes to AWS)
5. In case you want to remove all the resources created by terraform on your AWS account Run “terraform destroy” (will ask for confirmation and destroy infrastructure).

Verification:

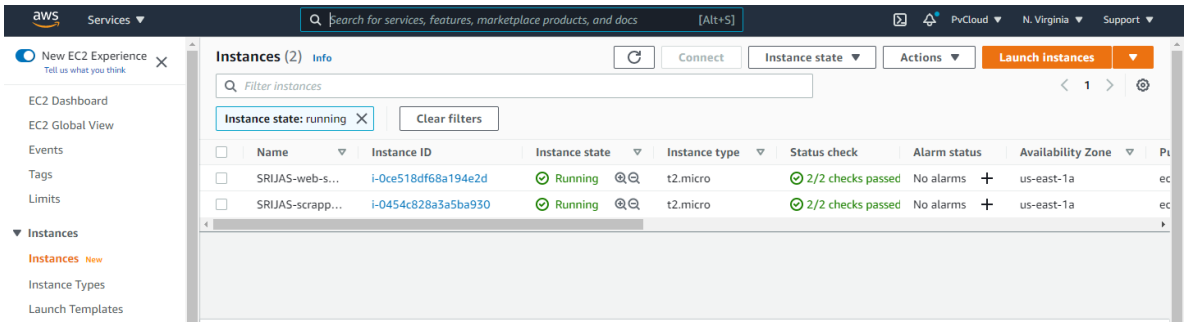
CLI outputs: After running the “terraform apply” command user can see the output variables. It will list all the information required for ec2 instances and the rds db instance. The output should look like this

```
aws_instance.ec2-webserver: Still creating... [30s elapsed]
aws_instance.ec2-scrapper: Still creating... [30s elapsed]
aws_instance.ec2-webserver: Still creating... [40s elapsed]
aws_instance.ec2-scrapper: Still creating... [40s elapsed]
aws_instance.ec2-webserver: Creation complete after 44s [id=i-0ce518df68a194e2d]
aws_instance.ec2-scrapper: Still creating... [50s elapsed]
aws_instance.ec2-scrapper: Still creating... [1m0s elapsed]
aws_instance.ec2-scrapper: Creation complete after 1m4s [id=i-0454c828a3a5ba930]

Apply complete! Resources: 19 added, 0 changed, 0 destroyed.

Outputs:

db-address = "srijas.cgdwdytucgi0.us-east-1.rds.amazonaws.com"
db-arn = "arn:aws:rds:us-east-1:765246388793:db:srijas"
db-endpoint = "srijas.cgdwdytucgi0.us-east-1.rds.amazonaws.com:3306"
db-hosted_zone_id = "Z2R2ITUGPM61AM"
db-id = "srijas"
db-name = "srijas"
db-port = 3306
ec2-scrapper-dns = "ec2-54-234-51-122.compute-1.amazonaws.com"
ec2-scrapper-ip = "54.234.51.122"
ec2-scrapper-private-dns = "ip-192-168-1-230.ec2.internal"
ec2-scrapper-private-ip = "192.168.1.230"
ec2-webserver-dns = "ec2-54-83-124-38.compute-1.amazonaws.com"
ec2-webserver-ip = "54.83.124.38"
ec2-webserver-private-dns = "ip-192-168-1-99.ec2.internal"
ec2-webserver-private-ip = "192.168.1.99"
```

Verification on AWS: Log in to your AWS management console.

1. Verify EC2 instances: Go to the EC2 Dashboard and verify the servers we created.

It should look like this



You can also verify other information about the instances by clicking on it.

2. Verify RDS instance: Go to the RDS Dashboard and click on DB Instances

It should look like this