Three Tier Environment Design Using Terraform in AWS Cloud

Contents

Objective

The objective is to create a three-tier environment structure in AWS using terraform. The expectation is to do the design following the modern cloud native architecture principles, industry best practices and considering reusability of the solution.

Tool selection

The choice of the cloud is AWS for this scenario. Since terraform is the most popular and cloud agnostic tool for automating the cloud infrastructure creation, we are going to use terraform as the tool of choice for this particular exercise.

Approach

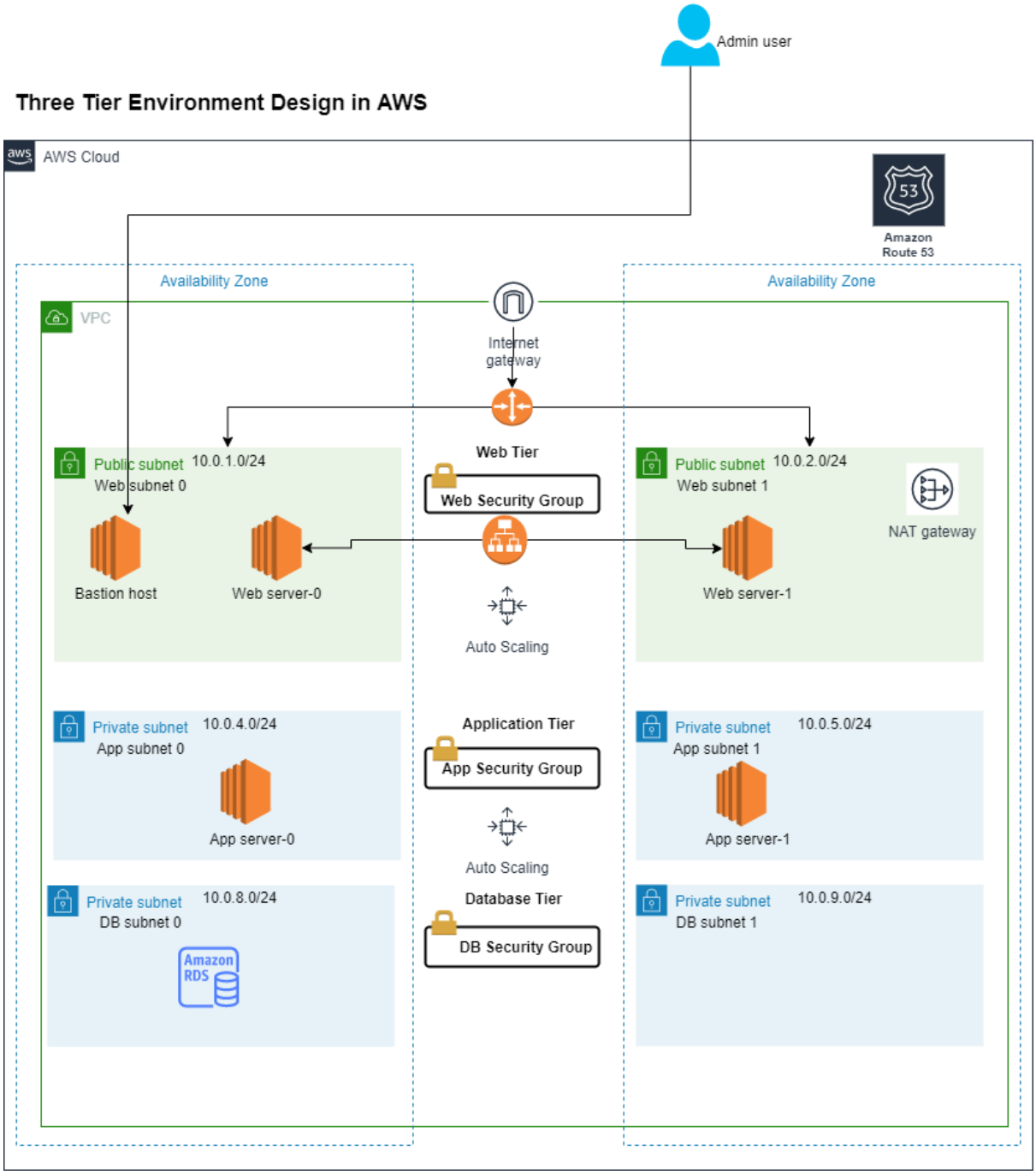
The solution is done be creating a high-level design diagram which depicts the various components that are getting created and how they are linked together. As a best practice and reusability terraform scripts are modularised. Terraform variables are used instead of hard cording the values within the main scripts. Descriptions are added as appropriate inside the code block. For this use

case the code is developed and tested in the local system but the same code can be re used in a pipeline for further automating the infrastructure deployment.

Pre requisites

- You need an AWS account with sufficient privileges for creating the infrastructure.
- AWS CLI installed and configured with your cloud account where the infrastructure is getting created.
- Terraform installed on your desktop/laptop.
- Your choice of IDE. I have used VSCODE.

High Level Design



Bill Of Materials

The terraform scripts are included to create the below components

1.VPC

2. Use the availability zones in the region

3. Internet gateway

4. Route tables

5. Two public subnets for two web servers in each availability zones

6. Four private subnets, two for app servers and two for db servers in each availability zones

7. Three security groups

8. Four EC2 instances (2 Web servers, 2 App servers)

9. Load balancer

Architecture Concerns

The design addresses the below architecture concerns

Repeatability

The code is written in such a way that it can be re used to create similar structure by modifying the variables to fit according to the requirements.

Scalability

The design deals with autoscaling the instances when there is an increase in the load on the instance.

Reliability

The infrastructure is created through pre-tested terraform scripts which benefit the systems to function less error prone due to human errors. The state of the infrastructure is saved in infrastructure state files to

Availability

The infrastructure is deployed to multiple availability zones to ensure high availability in case a data centre failure occurs.

Security

The architecture is a layered architecture with only the required sources and ports are enabled. The application servers and the DB servers are not exposed to internet. The scripts don't have passwords explicitly hard coded.

Terraform Execution

Navigate to the terraform root folder and the run the below commands

- terraform init
- terraform validate
- terraform plan
- terraform apply

Terraform Best Practices

A linting tool can be used for improving the quality of the terraform code. Use the modular approach. Use variables instead of hardcoding the values.