Dynamic Web Implementation & Management by Terraform

Flow: This project is processed via Four Steps!

1. Organising Variables & Logging the Resource State

* variables.tf // outputs.tf
* These code files are updated every step.
* Variables are organised for the convenience of resource management.
* The administrator[me] can check and track the crucial resource states selected in the outputs.tf file.

1. Setting up Essential Network sys & Resource

* main.tf // security.tf // database.tf
* Through the main.tf file, the basic network infrastructure is implemented.
* The security.tf file includes the security groups for each resource management.
* The RDS instance is built up by the snapshot created manually. This database.tf file has codes associated with the database.

1. Advanced Network System

* loadbalancer.tf // monitoring.tf
* The loadbalancer.tf file enables the admin to configure the Application Load Balancer with enhanced security
* The monitoring.tf file includes the SNS Topic to notify the change to the admin.

1. High Availability & Guarantee Usability [automatic try~while computer on:UUsing Python Lambda]

* compute.tf // dns.tf
* The compute.tf file contains implementing the auto-scaling group and associates with the target group of ELB & SNS Topic.
* The dns.tf file configures the simple network setting[record] to connect with the load balancer.

1. [Configuration AWS to interact with Terraform]

* IAM > Add User[admin\_dwa] to Group[admin\_team]>
* Security Credentials > -> Issue Access Keys
* Install the AWS CLI on Windows
* Locates in the directory[aws\_project\_3Terraform]
* Authentication: **aws configure** with verify Access Keys

1. [main.tf]

* resource “aws\_vpc”
* resource “aws\_internet\_gateway” => attaching IGW to VPC
* resource “aws\_subnet” => creating 2 Public Subnets
* Added attributes:
* availability\_zone: Public subnets are placed in two Availability Zones.
* map\_public\_ip\_on\_launch: The Public subnet needs A Public IPv4 address.
* resource “aws\_route\_table” => Routing Traffic to IGW
* The public route table should route to EVERYWHERE[0.0.0.0/0] through IGW.
* resource “aws\_route\_table\_association”

=> Place the Route table in each subnet

* resource “aws\_subnet” => creating 4 Private Subnets
* To manage the DB instance and the Web Instance separately
* resource “aws\_eip” => creating the 2 Public IPs for NATs
* resource “aws\_nat\_gateway” => creating 2 NAT gateway
* The NAT Gateway is placed in the public subnet[1,2] and has each EIP.
* Plus, it is dependent on the IGW gateway to communicate with WWW.
* resource “aws\_route\_table” => creating Route table in each NAT
* Thus, each route table routes to the [NAT/IGW] Gateway. And the NAT Gateway is attached to the IGW gateway, which is attached to VPC. [NAT->IGW->VPC]
* resource “aws\_route\_table\_association”

=> Place the 4 Route tables in 4 Subnets

* AZ1 Route table is placed on the AZ1 subnets[DB1, Web1].
* AZ2 Route table is placed on the AZ2 subnets[DB2, Web2].
* **EVERY AZ**, 1 Route Table Routing to-> 1 NAT Gateway

1. variables.tf
2. outputs.tf

* To manage the manifest information of the Resources, this outputs.tf includes the essential information to monitor and check.

1. Organise the Current Workload: output “current\_env” / output “current\_vpc”
2. Divide the Availability Zones & Check each Subnet CIDR & EIP

* output “public\_subnets”
* output “private\_az1\_subnets” // output “eip1\_az1”
* output “private\_az2\_subnets” // output “eip2\_az2”

1. Terraform APPLY

* terraform init -> terraform plan -> terraform apply
* Error 1: [Wrong address CIDR]
* The subnet CIDR has an error due to assigning an Invalid Network address.

Subnet Mask /27

🡪 (32-27) => 2^5(host bits) = every 32 addresses per subnet

Subnet Mask /28

🡪 (32-28) => 2^4(host bits) = every 16 addresses per subnet

=> 192.168.56.0, 16, 32, 64 … // = > 192.168.56.0, 32, 64, 96, …

* Error 2: [IAM User Authorisation]

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* Add the Policies to IAM User[admin\_dwa]:

AmazonEC2FullAccess / AmazonVPCFullAccess

* Success > The output contents are identified the manifest information.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명