ASSIGNMENT DOCUMENTATION ON TERRAFORM ECS MICROSERVICES

Dated 17-11-2022 (RITESH)

**Overview :**

In this documentation we have to **make 6 microservices in a single cluster using ECS, it will have 6 pipeline for 6 microservices.** Where we have to create services on single cluster where we access vpc, load balancer, ECS ECR , IAM role and policies and autoscalling.

**Steps**

* Application configuration
* VPC
* Required IAM roles
* Required security groups
* Load balancers
* Route53 record for the internal ALB
* ECR for docker images
* ECS cluster
* ECS task definitions
* ECS services
* ECS service autoscaling

## **Terraform**

Terraform is an open source infrastructure as code software tool created by HashiCorp. Users define and provide data center infrastructure using a declarative configuration language.

## **AWS CodePipeline**

AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates.

## **Let's start to build the AWS infrastructure**.

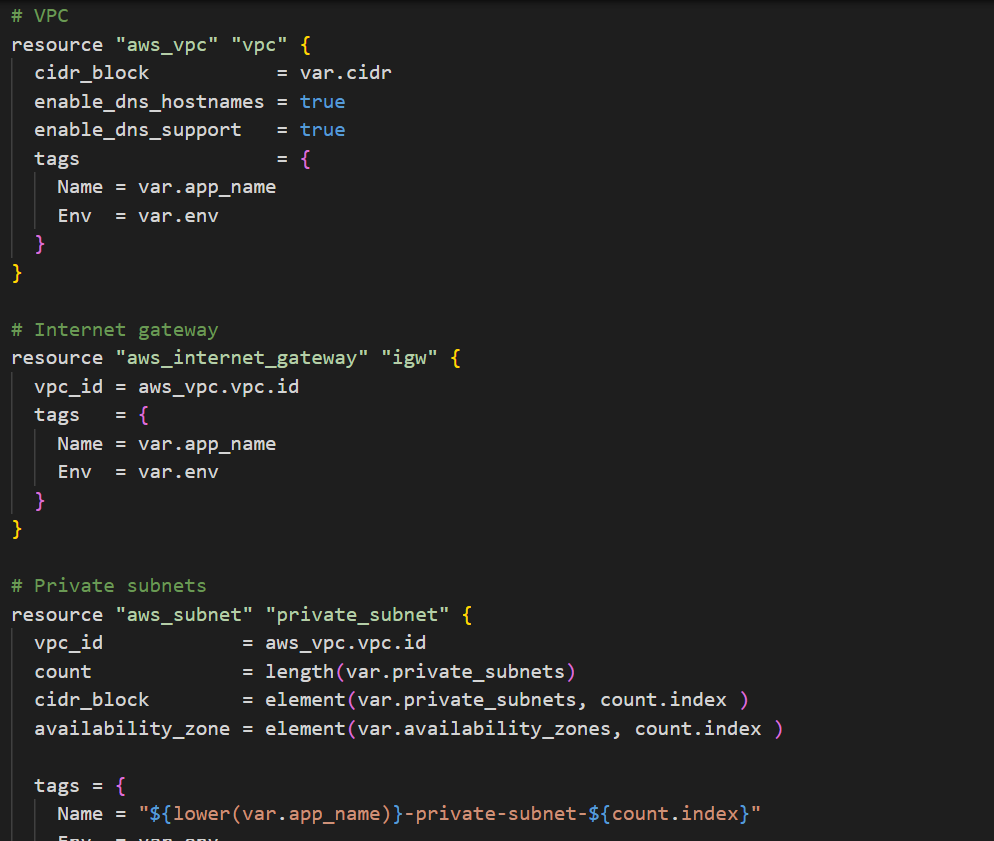
* Setting up the infrastructure is as simple and running the included terraform scripts, which uses Terraform to deploy several services into the AWS account.
* Setting up an IAM roles and policies.
* Install Terraform, I have chosen the us-east-1 default AWS region.

## **Build Infrastructure using Terraform**

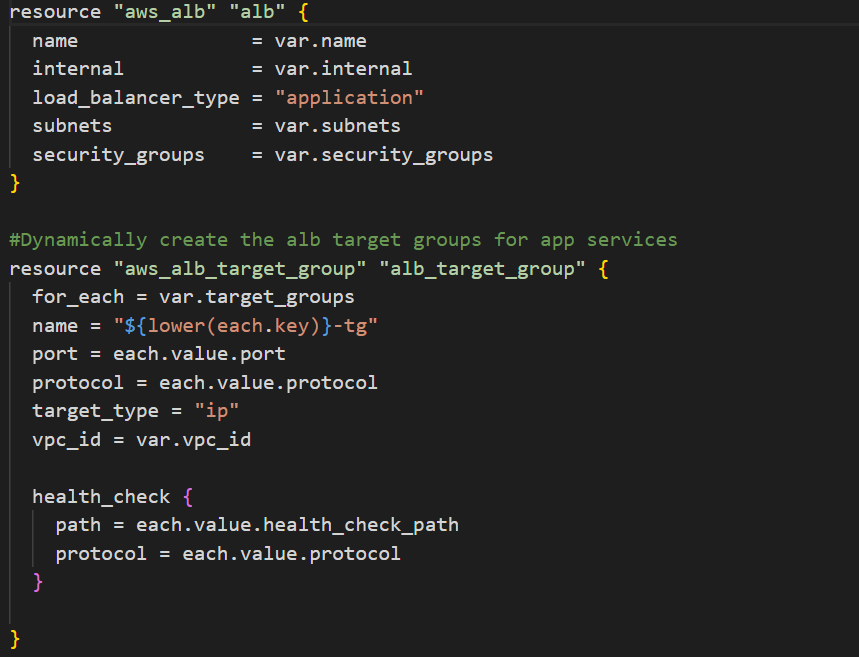
* The **terraform init** command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control. It is safe to run this command multiple times.
* **Terraform plan** command is to test your terraform scripts. ‘terraform plan’ alone will not actually carry out the proposed changes, and so you can use this command to check whether the proposed changes match what you expected.
* Use ‘terraform apply’ to actually build the AWS infrastructure and the CI/CD pipeline
* Compare the Terraform scripts to the AWS environment using AWS console. Let's have a look into vpc.tf, ecr.tf, ecs.tf and rds.tf

**Creating a VPC ,Application Load Balancer**

* vpc.tf creates a vpc

****

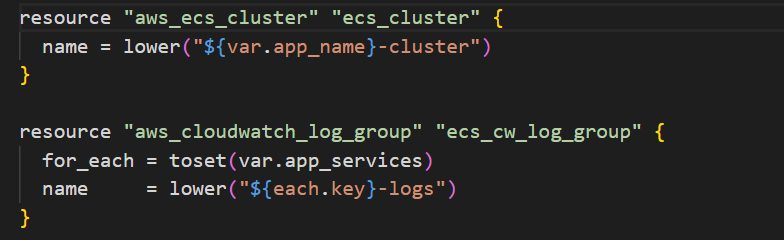
* vpc.tf creates an internet gateway to connect the VPC to the internet.
* Creating a ALB, where we have to process private and public ALB and Listener rules.

****

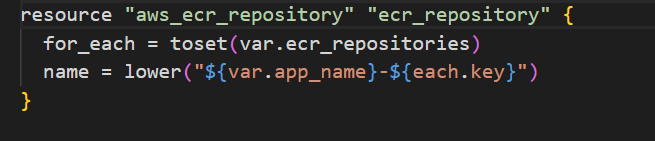
****

**ECS AND ECR**

* ecs.tf creates fargate cluster to run the docker image on ECR repositories.

****

* ecr.tf creates a container image in the ECR Repositories.

****