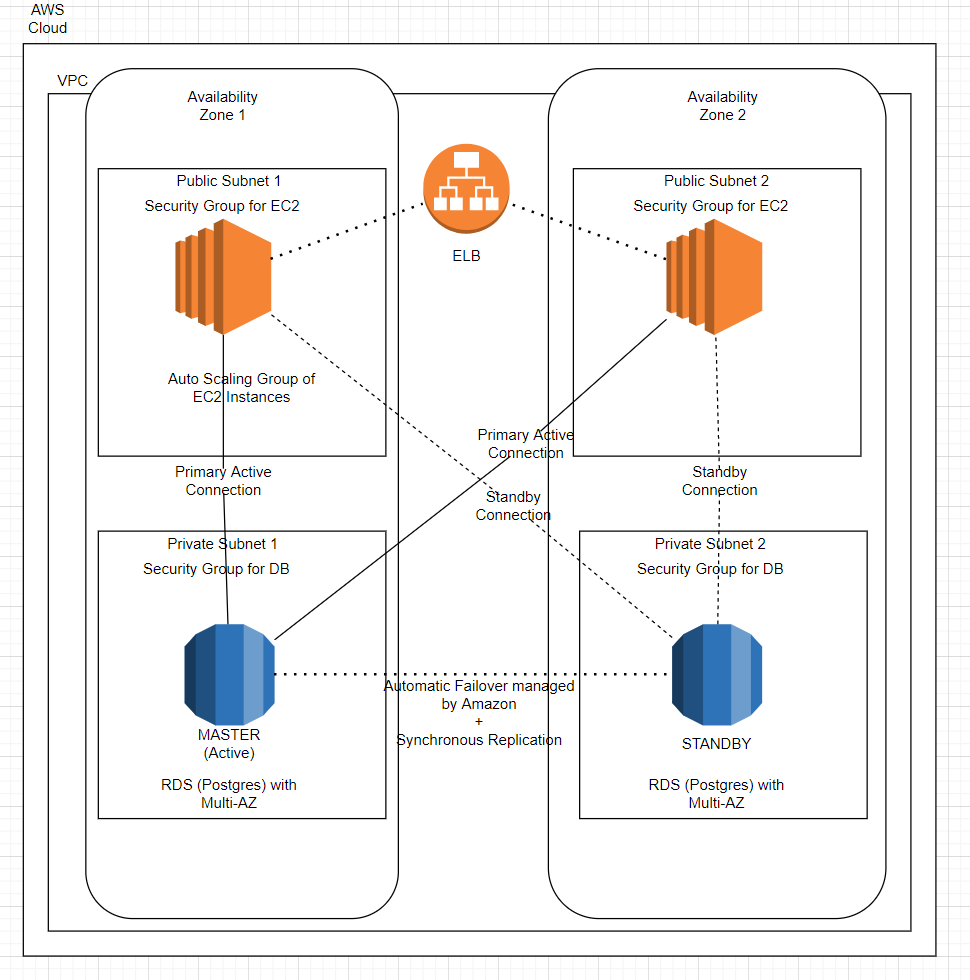
TECH APP Challenge

Prerequisites:

1. AWS Account
2. IAM user in AWS
3. Terraform Account and CLI

Infrastructure Architectural Diagram:



Solution Description (Infrastructure):

The entire solution will be deployed in a VPC.

The React front end and the Golang backend of the Tech App application will be run on an autoscaling group of EC2 instances. The subnets for these instances will be public. We will have 2 subnets which will be on 2 different AZs in the same region for redundancy. The minimum and maximum number of instances to be launched and running can be configured in the autoscaling group (for now I have set some random values in the Terraform code which can be changed as per the request). This will ensure that we will always have a minimum number of instances up and running fine to ensure that the application is fully redundant and scalable.

The Autoscaling group of EC2 instances will have a public facing Elastic Load Balancer which can be accessed using the URL (which will be obtained at the output of Terraform code). For now, I have made it a random one. We can use our own domain which we obtained using either Route 53 or other public domain name providers like GoDaddy. The ELB will allow us to load balance across the multiple running instances. The ELB is fully managed by Amazon and will be highly available, secure, and elastic.

For the Database, I have selected Amazon RDS (Postgres 9.6.22) with multi-AZ support enabled. This will ensure that we have redundancy in case of an AZ failure. The primary and standby RDS will be replicated synchronously and failover is done automatically by Amazon. In this way enhanced durability and increased availability of database is accomplished. The subnet for the RDS will be a private one. The inbound access to the RDS will only be possible from the security group assigned to the EC2 instances. Segmentation is achieved this way. This also makes the DB secure from other threats which is possible if the DB is exposed publicly.

AWS Code Deploy agents are installed automatically on all the launched EC2 instances by passing the installation script as User Data in the launch configuration.

Infrastructure Deployment (Terraform Code):

Terraform is used to automate the deployment of the necessary AWS infra components.

Terraform tf file 🡪 consists of the main code

Terraform tfvars file 🡪 consists of the variable definitions

CI/CD deployment using AWS Code Pipeline:

AWS Code Pipeline can be used to automatically create a pipeline to commit, build and deploy the code. This feature will enable us to automatically deploy the new versions once the code is committed to Git. We can either use Git or AWS Code Commit as the code repo.

The required installations (Go and other services) and the commands to start serving the app can be provided in the buildspec.yml file which should be provided in the Git repository.

AWS Code Deploy has the provision to deploy an app into an autoscaling group. This will ensure that all the EC2 instances which hosts the application will have the updated version whenever a new build is initiated.

AWS Secrets Manager:

All the credentials like DB logins and other important things can be stored in the AWS Secrets Manager and it can be retrieved from the AWS Secret Manager by running few commands in the EC2 instances. In this way, we can ensure that the credentials are not visible to the app developers. All they need to do is to reference the values as environment variables in their script.

Alerts:

We can use Amazon Cloud Watch to monitor the application and infrastructure. This will help us to send alarms if certain thresholds that we define are exceeded. We can also use it to track the API requests. These metrics provide a simple way to track the usage and outcomes of the API operations over time. This will give an understanding on how the application is performing and enables to identify and diagnose a variety of issues. We can also set alarms that watch for certain thresholds and send notifications or take specific actions when those thresholds are met.