

### *Figure 1: Job Seeking App Architecture*

### Introduction

As shown in Figure 1, the architecture shows the high-end level of the job seeking application. The services run on the AWS Cloud using different AWS Services.

### AWS Application Load Balancer (ALB)

We are using AWS ALB to direct the traffic directly to the AWS Fargate cluster. AWS ALB can also connect to the ec2 instances that run on private subnets as long as allowed to the security groups. This can also help to increase security so that traffic will not directly connect to the ECS cluster.

### AWS ECS

To orchestrate the docker containers to AWS and so that to deploy, manage, and scale containerized applications would be easier. We are also using 2 Availability zones for the high availability purposes.

### AWS Fargate

We are using fargate launch type to launch containers without worrying about provisioning of ec2 instances. Fargate launch is serverless and AWS managed. We can also set the desired utilization for cost-saving purposes.

### AWS NAT Gateway

All machines that run in a private subnet have no capabilities to connect to the internet. The solution is to provision a NAT Gateway in a public subnet.

### AWS Internet Gateway

This serves as our main door to allow communication between the VPC and the internet.

### AWS IAM Role.

Some tasks need to connect to the other AWS Services e.g. S3 and PostgreSQL, we are using IAM role to give permission so that both AWS services can connect with each other.

### AWS S3

As mentioned in a business requirement, the Introduction service feature delivers static web content. One of the features of AWS S3 is providing web static content.

### AWS RDS PostgreSQL

This is an AWS managed database so that stakeholders don’t have to worry about provisioning infrastructure. Also, Postgresql is preferred since it is open source.

### AWS CodePipeline

To create a pipeline with source, build, and deploy stages. The pipeline is invoked when a developer commits a code change to the CodeCommit repository.

### AWS CodeCommit

To host the application source code in a Git repository.

### AWS CodeBuild

To serve as a CodePipeline stage for performing the build process for a container image and push it to the AWS ECR

### AWS ECR

Each container app should have a container image that will live in its own repository in a container image registry. Our solution uses a private Amazon ECR repository as the container image registry.

### AWS Cloudwatch

We can use this service to monitor the containers or nodes and also for auto-scaling.

### AWS Auto-scaling

If the utilizations of containers or nodes have reached the threshold we set in cloudwatch, we can use this service to auto-scale.