LAUNCHPAD ASSESSMENT SOLUTION

Assessment Name	Assignment for the Assessment of Launchpad Padawans
Version	1.2
Date	11 October 2022
Author	Guney Ozkaya
Contact Info	guney.ozkaya@nordcloud.com
GitHub Repository	https://github.com/gozkaya/aws_infra

1. Scope

This repository provides solution response to the problem definition given by AWS Launchpad Team in September 2022.

The repository contains IaC files written in Terraform and relevant folders along with this Launchpad-Assessment-Guney-Ozkaya.pdf project report file.

2. Solution Description, Constraints

The repository also includes GitHub actions yaml file stored in .github/workflows folder which initiates a run on GHA on every push to the Main branch.

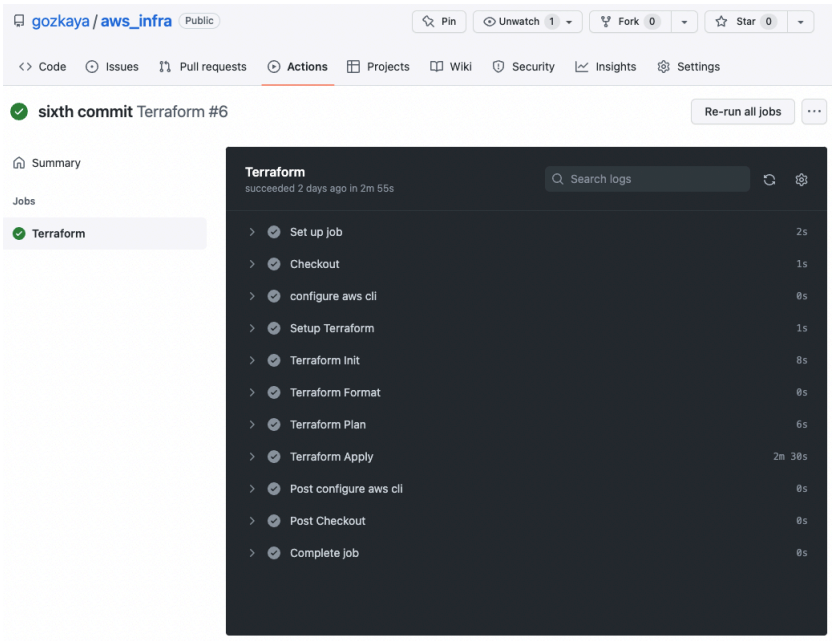

Figure 1 CI/CD Pipeline triggers by every push on the repository

There are no branches other than the Main branch. Dev and Test branches are deleted after successful operation.

a. Solution Architecture

The IaC code provisions resources according to the following Architecture.

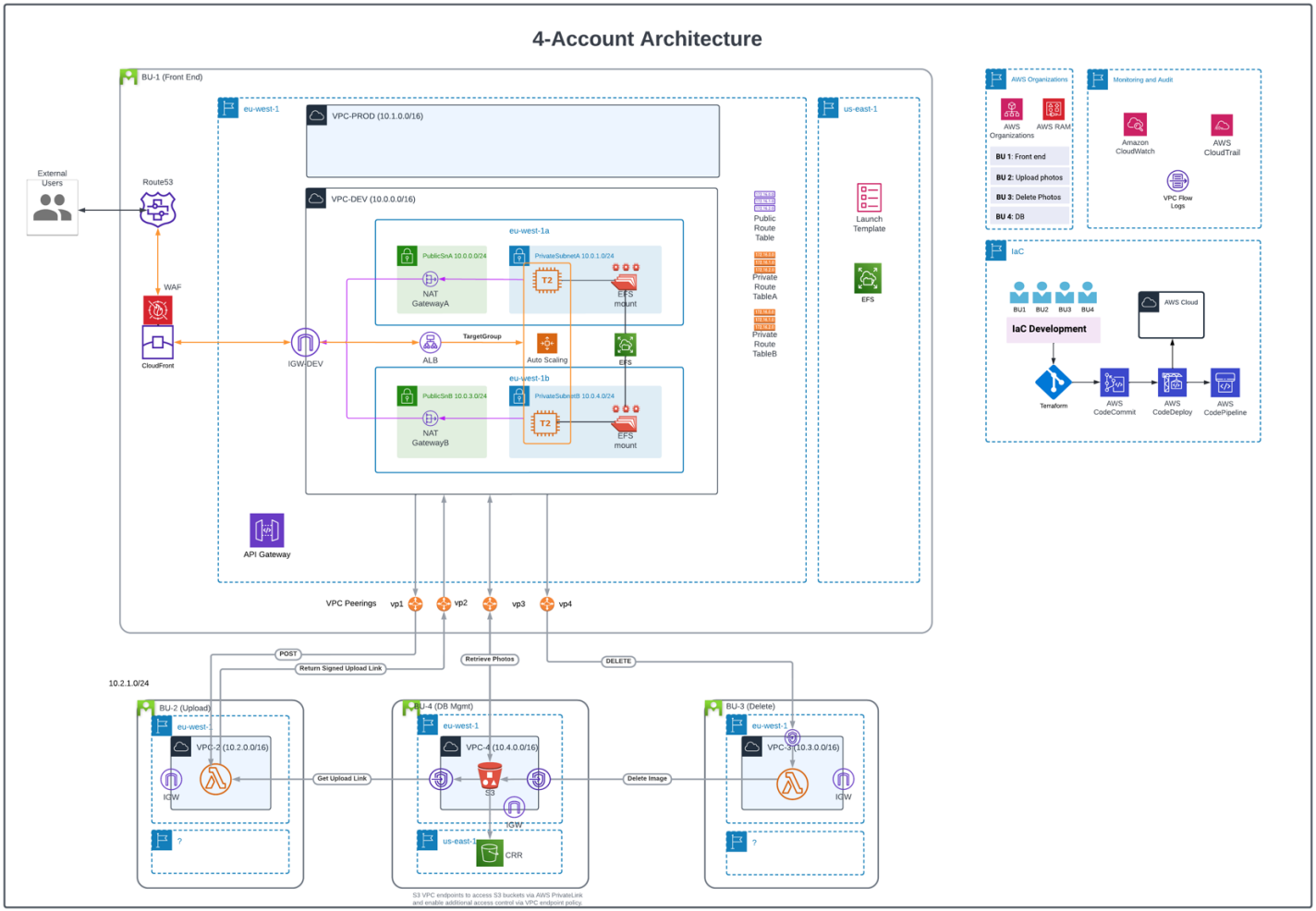

Figure 2 The proposed solution architecture

The solution code provided with the github repository is covering only account-1 part of the problem description. Moreover, the IaC solution does not only provision 100% of the resources depicted in Figure 2. Please refer to section b and section c for further details.

b. Notes on the AWS Account Structure

There are 4 accounts created for the purpose of this project:

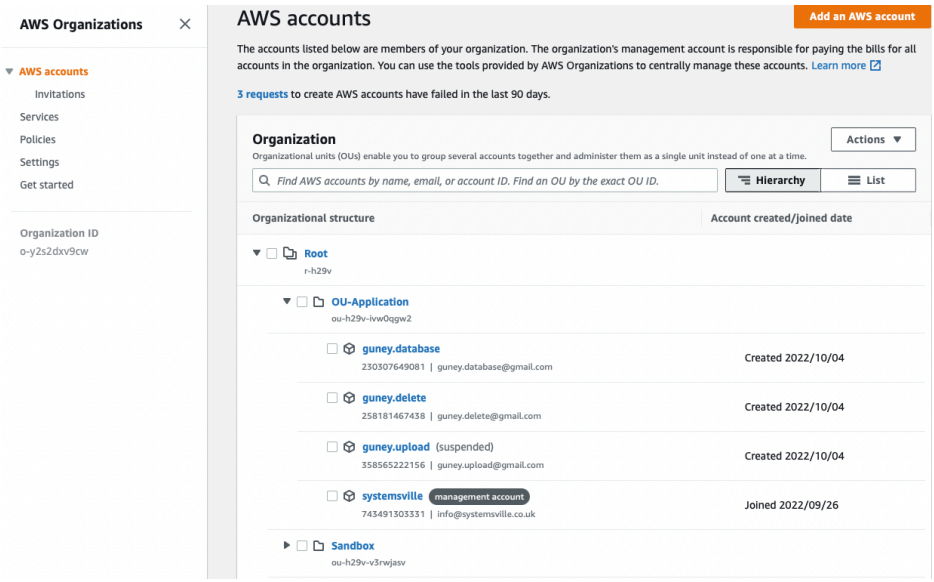

Figure 3 Account structure as developed in AWS Organisation on the main account.

The following information exchange is proposed between the accounts:

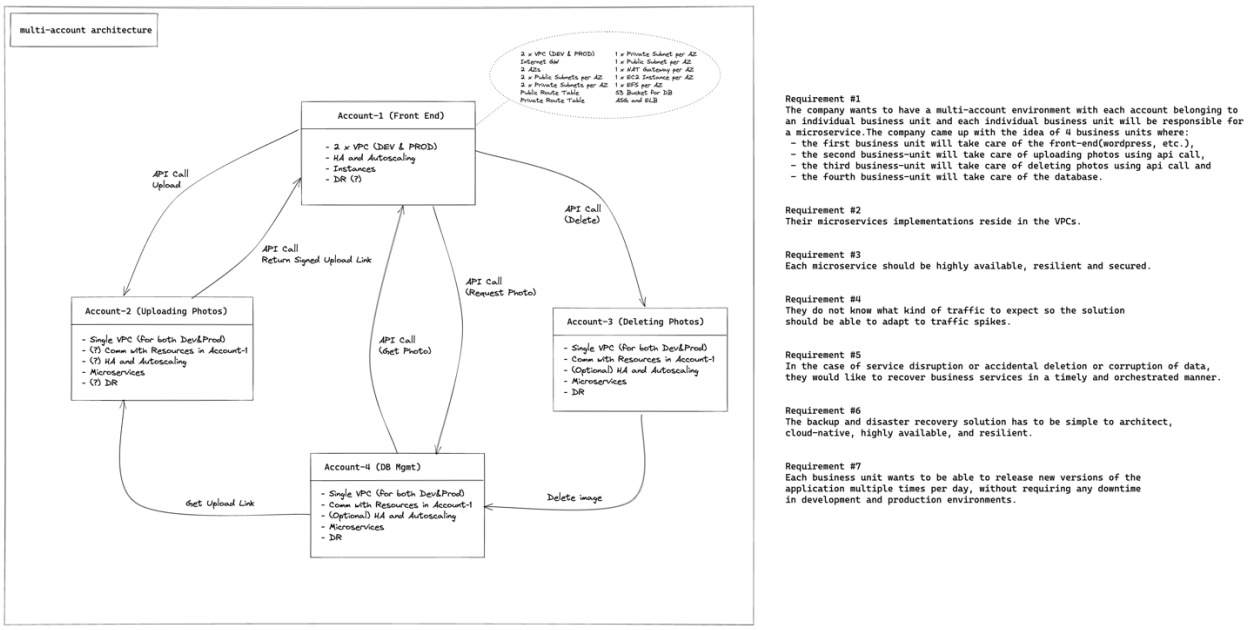

Figure 4 Information Exchange between accounts

The current version of the code is provisioning resources only on the main account. The overall consolidated account billing monthly estimation seemed to be hitting 100 USD this month. As I didn't have access to company's Sandbox account, I decided to keep the solution simple and confined into a single account in this version.

The solution only focuses on the VPC- DEV as shown in Figure 3. The current version does not provision *Upload*, *Database* and/or *Delete* VPCs. Therefore, VPC peering between VPCs are not created by the IaC code even if they have been tested manually in previous versions.

c. IaC vs Manual Provisioning

There are two VPCs created in the main account for DEV and PROD. However, the IaC code provided provisions resources only for the VPC-DEV. No resources are being created I other VPCs.

Some services need to be provisioned manually. These are:

- Route53
- CloudFront
- WAF
- CloudWatch
- API Gateway

The terraform code runs only in default workspace. No other workspace is created or used.

d. Terraform Project Code Breakdown Structure

This section summarizes which parts of the

/Module	folders include reusable resource descriptions to be instantiated by the .tf files residing in the project root folder which is /aws_infrastructure.
backend.tf	• Sets AWS provider region to eu-west-1 (Ireland) • Requires manual creation on an S3 bucket to store state file.
provider.tf	Sets AWS provider region for Disaster recovery Scenario to us-east-1 (Virginia)
network.tf	Provisions the following resources in 2 Availability Zones (eu-west-1a and eu-west-1b). • DEV and PROD VPCs and Internet Gas • Nat GWs • Route Tables and their associations • Subnets
secgrp.tf	Security Group creation for WebApp Instance and EFSMountTarget.
web_app.tf	Provisions the following resources: • EF Drive • Launch template

	• Load Balancer • Auto Scaling Group • Target Group
cloud_front.tf	*INACTIVE* Commented out as the code receives several errors. Aimed at creating CloudFront and WAF.
resize_app.tf	*INACTIVE* Commented out as the code receives several errors. Aimed at creating API Gateway and Lambda Functions.