**Final Test**

**Name: ANGELA JONYL C REYES**

**Student Number: 10989583**

**Time: 3hrs**

**Submission Requirements:**

1. Create a Word File named with the following format: [***yourlastname\_Final#.docx***]
2. Number your responses correctly in your file.
3. Upload your solution file to the provided Final Test Dropbox when completed.
4. Upload your IaC script to the Final Test Dropbox folder.

**Total # 30 Marks**

**Note:** **Individual Open Book Assessment (Internet, labs, assignments, and previous notes).**

**Scenario:**

You have been hired on by a new startup and been tasked with deploying the company’s backend application in a repeatable, automatic, and scalable manner in the cloud.

**Note:** **DELETE ALL RESOURCES WHEN YOU HAVE COMPLETED YOUR TASKS!!!!!**

# **Deliverables:**

**Note:** Use your **lastname** as the appropriate naming context for your resources created.

Accurate, high quality and well formatted screenshots required as proof/evidence. Please do not submit screen shots of your tasks process as the deliverables are only concerned with the results.

1. Show evidence of a successfully built docker image for the application stored in a private registry. (**1 Mark**)
2. Show evidence of a successfully executed CI/CD Pipeline (successful build and deployment). (**3 Marks**)
3. Show a screenshot of your dockerfile. (**4 Marks**)
4. IaC infrastructure script created and submitted. (**5 Marks**)
5. Show evidence of application running behind an Elastic Load Balancer. (**2 Marks**)
6. Show evidence that each route is accessible for the API (main route, host and IP routes). (**3 Marks**)
7. Show evidence of at least two (2) containers running the API. (**2 Marks**)
8. Show evidence that the API application is running in private subnets. (**2 Marks**)
9. Show evidence of proper security policies in place (security group, network security group etc.). (**3 Marks**)
10. Show evidence of autoscaling configuration in place. (**4 Marks**)
11. Document structure and numbering. (**1 Mark**)

# **Tasks To be complete:**

**Note:** The source code for the backend API is in the assignment folder. Move the code to a GitHub Repo and containerize it.

* Modify the **main** route to say, “Welcome to ***yourlastname*** Final Test API Server”.

## **Phases**

**Phase 1 – Deploy the infrastructure.**

* Serverless (ECS, Cloud Run or Container App)
  + Containerized Python API backend microservice.
  + Use Proper Python version (Flask and Gunicorn)
* Any Cloud Platform of choice (Azure, GCP or AWS)
* A Load Balancer service to manage traffic to the API.
* Compute runs in private subnets (no public IPs)
* Appropriate security policies applied.

**Phase 2 – Deploy CI/CD Pipeline**

* Source phase.
* Build phase.
  + Save Container image in a container registry.
* Deployment phase
  + Deploy to a container service.

**Phase 3 - Deploy the Application**

* At least two (2) backend compute nodes.
* Create a scaling policy that can scale out nodes to five (5) if CPU peeks to 90% for 2 minutes).
  + Apply a scale in policy to remove instances when load is below 50%.
* Deploy behind an Elastic Load Balancer service/feature.

**Phase 4 – IaC script**

* Create an IaC script that the client can use to re-deploy core infrastructure.
  + VPC/VNet
  + Subnets
  + Security groups/rules
  + Load Balancer & Target groups/target pool.

**Note:** **DELETE ALL RESOURCES WHEN YOU HAVE COMPLETED YOUR TASKS!!!!!**