Technical Challenge

The challenge is shaped around the day-to-day tasks that our DevOps engineers would likely need to work on.

This challenge set is designed for high positions, and as such, many of the questions are designed to be more open-ended. You are expected to leverage on your own experience to design a suitable solution and be able to justify your design decisions when asked.

Remember to approach the entire challenge set as a single solution. While some of the challenges seem unrelated, eventually they will converge. Read the questions carefully and consider your implementation decisions carefully.

Prerequisites

- An oracle cloud account. We recommend that you sign up for a trial account with the free $300 credits (and the free tier resources). We are not able to assist you in creating / provisioning a new account, or if you have issues accessing the service.
- A GIT repository that you can store your solution and share the link with us.

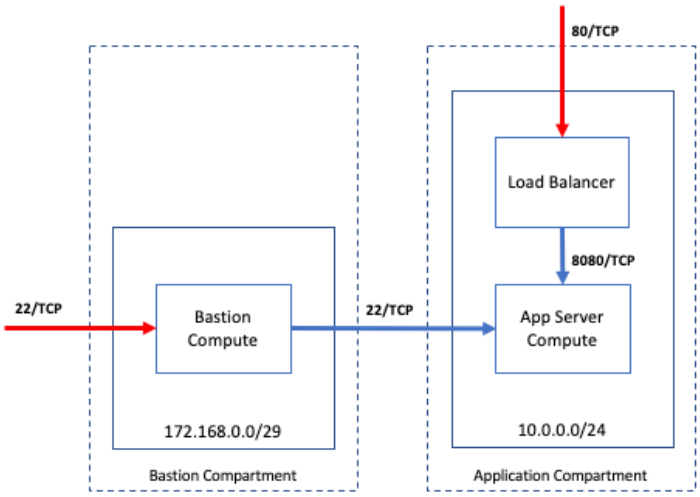
Challenge 1: DevOps Challenge 1

Write the terraform scripts that is able to provision an environment similar to the diagram below. Take note of the following:

1. Not all components are in the diagram, use your own experience / initiative to find out what else needs to be in the environment for it to work. E.g., Subnets are not in the diagram.
2. App Server is only accessible via port 8080 from the Load Balancer, and 22 from the bastion compute. Plan your network boundaries properly to ensure there is sufficient level of security.
3. The diagram leverages on the always free resources available on oracle cloud. (2x compute, LB).

There are tasks that need to be done as part of the provisioning

1. Install JDK8 on the App Server compute.
2. Install Docker engine on the App Server compute.



At the end of this challenge, push the terraform scripts to a git repository.

Challenge 2: Development Challenge

This challenge requires you to write some code.
Follow the step by step instructions from this link and build a dropwizard application. It's a simple Hello World application.

https://www.dropwizard.io/en/latest/getting-started.html

PDF version of the tutorial is available by request.

Challenge 3: Development Enhancements

Extending the dropwizard application in Challenge 2, add a new endpoint that exposes a fibonacci service. The service takes in a number between 1 to 100. It will then calculate the specified number of members in fibonacci sequence, and generates a json response with 2 properties: a list of numbers in the sequence, and the sum of numbers. A sample response for would look something like this:

```
{
    "member-count" : 6
    "sequence"     : [ 1, 2, 3, 5, 8, 13 ],
    "total"        : 32
}
```

Challenge 4: DevOps Challenge 2

Dockerize your application, publish it to a public docker repository (like dockerhub). Deploy the container onto the AppServer compute from challenge 1.

Focus areas: DockerFile, the deployment process, getting it to run on Oracle Cloud.

Challenge 5: Monitoring with Grafana

Set up Grafana on a new compute instance. Add it into your environment from Challenge 1. Show the following:

1. Heartbeat or health status of the App server
2. Number of requests being serviced.

Make sure that the Grafana dashboard is also accessible.

Submission

1. Upload all the scripts, source codes into a git repository. Document the basic steps on how to build and run it in a README.md file as part of the repository.
2. On the scheduled date, demonstrate the parts that you managed to complete.