Fern Tech

Developer Test

Document Metadata

Identifier	Details
Title	Cloud/DevOps Engineer Test
Date issued	01 March, 2020
Author	Ehfaz Rezwan
Reference Number	
Approved By	Ehfaz Rezwan

Document History

Version	Date	Author	Approved By
1.0.0	01 March, 2020	Ehfaz Rezwan	Ehfaz Rezwan

Disclaimer

Table of Contents

Requirements

- 1+ year server management experience (could simply be experience in dealing with servers in general)
- 1+ years experience in DevOps (Docker, Containers, Docker-Compose, Kubernetes)
- 1+ years of experience in full-stack development (NodeJS, ReactJS, Redux, Django, WebSockets)
- Familiarity with microservice architecture
- Experience in developing or working with CI/CD pipelines, specifically in Jenkins
- AWS experience/certification is a plus

The Test

Part I:

Create an excel file using any of your preferred python library. Using the attached “index.html” file populate the excel file with the same table structure, do not pick any row that contains "an" as a substring in any of the names in the "Name" column.

Once the Excel file is created, construct a “Google Forms” form with the same fields and use Python and Selenium to enter the data from the excel file onto the Google form.

Part II:

Dockerize the Selenium script. Do so by making use of the selenium images available on Docker Hub. We expect you to use the concept of remote web drivers to accomplish this task. The final deliverable here will be a combination of Dockerfiles and a docker-compose.yml file that ties everything together.

Part III:

Use Kubernetes to turn your docker-compose file into a scalable cluster. You may use “minkube” for this.

Part IV:

Deploy the Kubernetes master node onto a VPS. Feel free to make use of the free tiers of AWS, GCP or any other cloud platform for this task.

Bonus round:

Do one or more of the below tasks to establish your credibility even further. These tasks are not mandatory, however, they will help you with your application:

1. Use django-channels to create a websocket channel with which your Selenium script may communicate to tell the server what it is doing.
    a. For example, sending the message “Extracting data from Excel file” to the django-channels server
2. Use React.js and Redux to create a frontend interface that communicates with any websocket data feed to show data in real-time
    a. Look into https://www.npmjs.com/package/@giantmachines/redux-websocket
3. Create a CI/CD pipeline using Jenkins to accomplish the following tasks:

a. Package the selenium script into a docker container
b. Push the image to docker hub

Deliverables

- Clean, readable code that is also annotated
- Code must follow developing conventions
- All work must be uploaded to Github, with regular commits made i.e. don't just upload the entire code to Github AFTER finishing development - commit at relevant checkpoints
- Share Git link once done