iPAY™

DevOps Engineer

Congratulations on making it to the next round! In order for us to better understand your technical skills, we'd like you to complete some tasks for us.

Overview:

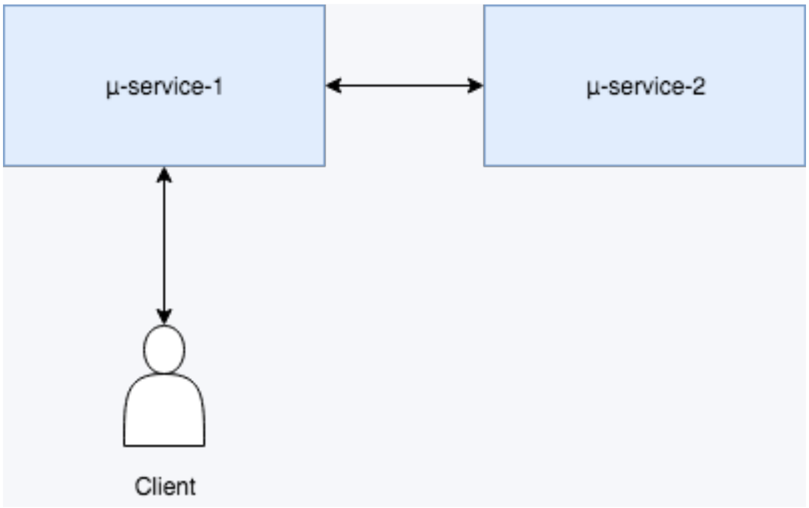
- Implement two basic, dependent microservices.
- Deploy the solution on a local container orchestrator.
- Create a diagram and briefly explain how this would be deployed using Continuous Integration / Deployment on a cloud provider.

Test Instructions:

- You will be evaluated both on your implementation choices and the quality of your code; so please ensure code is well structured, extensible, testable, readable, commented etc.
- You should submit your solution as a git bundle file, or provide a GitHub link.
- You should provide step by step documentation on how to execute your submission.

Part 1

1. We would like you to create two dockerized microservices (emphasis on the *micro*) that provide a JSON API over HTTP.



- µ-service-1 should provide a single endpoint at POST /api. When sent data in the format:

```
{ "message":"abcdefg" }
```

It should respond with message reversed and a random number, for good measure:

```
{ "message":"gfedcba", "rand": 0.12345678 }
```

- Since µ-service-1 is breaking such a sweat generating all those random numbers, it outsources the string reversal to µ-service-2 who also provides a single endpoint. The endpoint POST /reverse implements the following:

```
{ "message":"abcdefg" } → { "message":"gfedcba" }
```

Part 2

1. Write configuration files to deploy the microservices on one container orchestrator. Example orchestrators being:
    - Kubernetes (Minikube)
    - Docker Swarm
    - Nomad
    - ECS

Note: we **do not** expect you to provide configuration for the orchestrator itself, a brief description of the deployment environment is fine.

Part 3

1. Produce a system diagram and a brief description of how you would deploy the solution on a cloud provider such as AWS or GCP.

2. Explain how you would integrate CI/CD to perform deployments, and how you would ensure downtime was avoided?
3. Add integration test script to confirm that the whole system works as expected.