Software Engineer - Take-home exercise

Tags	
Main stakeholder	 Richa Vohra

Take home exercise: nomad

As part of our hiring process, we would like to invite you to complete a take-home exercise that will assess your skills in building a REST API using Golang.

Exercise Description:

Your task is to create a REST API that provides an endpoint, `PUT /services/<name>`, which allows creating or updating a service with the given name. It accepts the following parameters:

- `script` (boolean)
- `url` (string)

Upon receiving a request, the API should start a Nomad job (refer to https://www.nomadproject.io/) and return the URL at which the service is available.

The service is an Nginx server that serves the page downloaded from the specified URL. The behavior of the URL content depends on the value of the `script` flag:

- If `script` is set to `false` (default), the content is downloaded and served as the index page.

- If `script` is set to `true`, the content of the URL should be executed to generate the page.

Please follow these guidelines for the exercise:

1. Create a GitHub repository for your project.
2. Include a clear README file that explains how to run the project locally.
3. Write clean and well-structured code in Golang.
4. Ensure that your code is automatically tested after every push.
5. Handle errors gracefully.
6. Note that scalability is not a requirement for this exercise.
7. Consider yourself as the owner of the project and make necessary changes if required while adhering to the core requirements.
8. While not expected, if you find the exercise interesting and would like to add any bonus features or enhancements, feel free to do so!

After completing the exercise, we would like to organize a ~1 hour discussion to get your feedback about the exercise, but also talk about scaling considerations, including how we could support thousands of clients, how we could improve security, code instrumentation, and other relevant topics.

To set up Nomad locally, please refer to the official documentation at https://developer.hashicorp.com/nomad/tutorials/get-started/gs-start-a-cluster. You can start Nomad using the command `nomad agent -dev -bind 0.0.0.0 -network-interface='{{ GetDefaultInterfaces | attr "name" }}'`.

Example with script set to false:

- Create a new entry on pastebin.com with the content "Hello world, this is the content of my webpage"
- Create a new service using your API: `curl localhost:3000/services/mypage -X POST -d '{"url": "https://pastebin.com/raw/hEFbnx33", "script": false}'`
- The API returns:

```
{
    "url": "http://192.168.1.104:20676"
}
```

- The content of the pastebin is downloaded and served with Nginx. A request to http://192.168.1.104:20676 returns "Hello world, this is the content of my webpage"

Example with script set to true:

- Create a new entry on pastebin.com with the content:

```
#!/bin/sh

echo "Hello world!"
```

- Create a new service using your API: `curl localhost:3000/services/mypage -X PUT -d '{"url": "https://pastebin.com/raw/abcde123", "script": true}'`
- The API returns:

```
{
    "url": "http://192.168.1.104:22733"
}
```

- The content of the pastebin is executed as a script, and accessing the URL at http://192.168.1.104:22733 returns "Hello world!"

We look forward to reviewing your submission and discussing your exercise with you. If you have any questions or require further clarification, please let us know.