Kyosk Backend Test

Hello and thanks for taking the time to try this out.

The goal of this test is to assert (to some degree) your coding, testing, automation and documentation skills. You're given a simple problem, so you can focus on showcasing your techniques.

Problem definition

The aim of test is to create a simple HTTP service that stores and returns configurations that satisfy certain conditions. Since we love automating things, the service should be automatically deployed to kubernetes.

You can use either Java/Spring or Scala/Play/Akka

Note: While we love open source here at Kyosk, please do not create a public repo with your test in! This challenge is only shared with people interviewing, and for obvious reasons we'd like it to remain this way.

Instructions

1. Create a private repository(Github Preferred).
2. Add us to the repository.
3. Create a new `dev` branch.
4. Solve the task and commit your code. Commit often, we like to see small commits that build up to the end result of your test, instead of one final commit with all the code.
5. Do a pull request from the `dev` branch to the `master` branch. More on that right below.
6. Reply to the thread you are having with our HR department so we can start reviewing your code.

In your pull request, make sure to write about your approach in the description. One or more of our engineers will then perform a code review. We will ask questions which we expect you to be able to answer. Code review is an important part of our process; this gives you as well as us a better understanding of what working together might be like.

We believe it will take 4 to 8 hours to develop this task, however, feel free to invest as much time as you want.

Endpoints

Your application **MUST** conform to the following endpoint structure and return the HTTP status codes appropriate to each operation.

Following are the endpoints that should be implemented:

Name	Method	URL
List	`GET`	`/configs`
Create	`POST`	`/configs`
Get	`GET`	`/configs/{name}`
Update	`PUT/PATCH`	`/configs/{name}`
Delete	`DELETE`	`/configs/{name}`
Query	`GET`	`/search?metadata.key=value`

Query

The query endpoint **MUST** return all configs that satisfy the query argument.

Query example-1:

```
curl http://config-service/search?metadata.monitoring.enabled=true
```

Response example:

```
[
  {
    "name": "datacenter-1",
    "metadata": {
      "monitoring": {
        "enabled": "true"
      },
      "limits": {
        "cpu": {
          "enabled": "false",
          "value": "300m"
        }
      }
    }
  },
  {
    "name": "datacenter-2",
    "metadata": {
      "monitoring": {
        "enabled": "true"
      },
      "limits": {
        "cpu": {
          "enabled": "true",
          "value": "250m"
        }
      }
    }
  },
]
```

Query example-2:

```
curl http://config-service/search?metadata.limits.cpu.enabled=true
```

Response example-2:

```
[
  {
    "name": "datacenter-2",
    "metadata": {
      "monitoring": {
        "enabled": "true"
      },
      "limits": {
        "cpu": {
          "enabled": "true",
          "value": "250m"
        }
      }
    }
  }
]
```

Schema

- **Config**
  - Name (string)
  - Metadata (nested key:value pairs where both key and value are strings of arbitrary length)

Configuration

Your application **MUST** serve the API on the port defined by the environment variable `SERVE_PORT`. The application **MUST** fail if the environment variable is not defined.

Deployment

The application **MUST** be deployable on a kubernetes cluster. Please provide manifest files and a script that deploys the application on a minikube cluster. The application **MUST** be accessible from outside the minikube cluster.

Rules

- You can use either Java/Spring or Scala/Play/Akka
- The API **MUST** return valid JSON and **MUST** follow the endpoints set out above.
- You **SHOULD** write testable code and demonstrate unit testing it.
- You can use any testing, mocking libraries provided that you state the reasoning and it's simple to install and run.
- You **SHOULD** document your code and scripts.
- Check in your code to bitbucket and share the link