avito.tech

Тестовое задание на позицию стажёра в IT-Security: DevOps

Задача:

Разработать минималистичное приложение-интерфейс для работы с Redis'ом с проксированием трафика. Выполнить его сборку и развёртывание.

Требование к реализации:

1. Создать иерархию на файловой системе:

```
<root_project_dir>/
                docker-compose.yaml
                app/
                        <source_dir>
                        <dockerfile>
                redis/
                        <dockerfile>
                nginx/
                        <dockerfile>
```

- Каждый компонент содержит Dockerfile.
- Конфигурация компонентов "не запекается" внутри образа(-ов).
- Трафик к приложению проходит через web-сервер (nginx), порт 8089.

2. В приложении реализовать 3 endpoint'а (формат ответа произвольный):

   – Создать или перезаписать пару ключ-значение / значение по заданному ключу.

```
/set_key
request body: application/json
params:
     {
          "<key>": "<val>"
     }
```

   – Вернуть значение по ключу (если ключа нет, то возвращаем 404).

```
/get_key?key=<key>
```

   – Удалить пару ключ-значение.

```
/del_key
request body: application/json
params:
     {
          "key": "<key>"
     }
```

   – На все остальные uri отвечаем 403.

Технические требования:

1. В контейнерах, в качестве базового образа, используйте официальный образ Debian.
2. Развертывание и сборка должны выполняться по средствам Docker Compose (compose file version >= 3.3).
3. Приложение реализовано на Golang.
4. В Redis'е работаем только со строками. Набор команд.

Опциональные требования:

1. На Redis'е поддержана аутентификация.
2. Redis и приложение "общаются" по зашифрованному каналу (TLS-соединение).

Как оформить решение:

Решения на оба задания выложите в публичный Github-репозиторий, а ссылку на него пришлите нам через форму.