Тестовое задание

После очередного архитектурного комитета, перед вами поручили решить задачу реализации сервиса, предоставляющего асинхронный интерфейс инициирования задач на активацию оборудования.

Вам необходимо реализовать два сервиса:

* Существует синхронный **сервис конфигурации A**, который закрыт для изменения.

**Необходимо реализовать сервис-заглушку**, предоставляющий один единственный HTTP/s endpoint для конфигурации конкретного устройства.

Endpoint имеет следующий вид:

**POST /api/v1/equipment/cpe/{id}**

Где {id} – это серийные номер устройства вида ^[a-zA-Z0-9]{6,}$, а параметры типового запроса представлены ниже:

{

“timeoutInSeconds”: 14, // A maximum amount of time to await provision response

“parameters”: [

“username”: “admin”,

“password”: “admin”,

“vlan”: 534, // optional

“interfaces”: [

1,2,3,4

]

]

}

Пример ответа:

{

code: 200,

message: “success”

}

Возможные варианты ошибок:

{

code: 500,

message: “Internal provisioning exception”

},

{

code: 404,

message: “The requested equipment is not found”

}

Отправка ответа должна осуществляться по истечению 60 секунд.

* Реализовать Fronted **сервис B** решающий задачу асинхронного запуска процесса конфигурации. Для создания задач конфигурации, сервис дублирует HTTP интерфейс выше и предоставляет дополнительный endpoint для проверки статуса задачи конфигурации.

Endpoint получения статуса имеет следующий вид:

**GET /api/v1/equipment/cpe/{id}/task/{task}**

где {id} – это серийные номер устройства вида ^[a-zA-Z0-9]{6,}$, {task} – уникальный id задачи.

Примеры ответов:

{

code: 200,

message: “Completed”

},

{

code: 204,

message: “Task is still running” – задача все еще выполняется, не пришло уведомление о завершении

}

Возможные варианты ошибок:

{

code: 500,

message: “Internal provisioning exception”

},

{

code: 404,

message: “The requested equipment is not found”

},

{

code: 404,

message: “The requested task is not found”

}

При получении запроса на активацию, сервис должен создать задачу, сохранив входные параметры, id устройства и сгенерировать уникальный id задачи и отправить сообщение в очередь RabbitMQ, и незамедлительно вернуть результат с номером задачи:

{

“code”: 200,

“taskId”: ????

}

Фактический статус завершения конфигурации, **сервисом A**, определяется путем подписки на соответствующую очередь в RabbitMQ, по уникальному id задачи.

* На заключительном этапе необходимо реализовать скрипт, который фактически выполняет задачи конфигурации, путем вызова внешнего **сервиса конфигурации A**.

Бизнес логика работы скрипта:

* + Сервис должен прочитать задачу из соответствующей очереди RabbitMQ;
  + Вызвать **сервис A**;
  + Отправить результат вызова в соответствующую очередь RabbitMQ;

Общие требования

* Python 2.7+;
* Любые сторонние библиотеки;
* Способ хранения задач timestamp, id, equipment id, parameter, status – на ваше усмотрение;
* Параллельная обработка очереди;
* Быть надежным. Учитывать корнер кейсы н.п. недоступность внешнего сервиса и т.п.
* Поддержка HTTPS;
* Рез тат опубликовать в проект на github с описанием шагов для сборки и запуска;
* Можно использовать любой другой брокер сообщений;

Будет плюсом

* Диаграмма взаимодействия сервисов;
* Автоматизация процесса развертывания Docker + ansible;
* Написанные интеграционные тесты;