

Relatório sobre a configuração do ambiente de desenvolvimento da plataforma *InterSCity*

Bruno Carneiro da Cunha
<brunocarneirodacunha@usp.br>

5 de setembro de 2020

1 Introdução

Este documento descreve as dificuldades encontradas para configurar um ambiente de desenvolvimento da plataforma InterSCity de acordo com as instruções do arquivo *HACKING.md*. O ambiente foi executado em uma máquina de 8 vCPU na plataforma *Google Cloud*, rodando *Debian Stretch*. O código-fonte da plataforma foi baixado diretamente do repositório oficial da plataforma no *GitLab*, e o último *commit* tinha o hash prefixo f965483d.

2 Dependências

- O ambiente de desenvolvimento da plataforma supõe que o *daemon* do *Docker* estará rodando, mas isso não é dito na seção *Services that are expected to be up and running*. Também é necessário ter instalado o *docker-compose*.
- O arquivo *HACKING.md* declara que é esperado que o serviço do RabbitMQ esteja rodando, porém o *kong-api-gateway* tenta subir outra instância desse mesmo servidor na máquina, levando à um conflito de portas.
- Nos scripts *./bin/setup* de cada microserviço, a gema **pg 1.1.4** falha na instalação. É necessário instalar o pacote **libpq-dev** para resolver o problema.

3 Microserviços

3.1 kong-api-gateway

- A documentação manda rodar o script *./bin/setup* de cada microserviço, porém o script falha se o serviço do **Kong** não estiver rodando. Logo, antes de rodar os scripts *./bin/setup*, é preciso rodar o script *./kong-api-gateway/scripts/setup* e *./kong-api-gateway/scripts/development start*.
- O contêiner *kong-database* não roda, é necessário incluir no *docker-compose.yml* a variável de ambiente **POSTGRES_HOST_AUTH_METHOD=trust**.
- Recomendo a inclusão dos outros serviços que precisam estar disponíveis na máquina *host*, como *MongoDB* e *PostgreSQL*, no *docker-compose.yml* desse serviço.

3.2 data-collector

- O script `./bin/setup` não roda, pois há um método não definido. É necessário incluir no arquivo `./config/application.rb` a linha **require "active_record/railtie"**.

3.3 resource-adaptor

- Há um *typo* na linha 4 do script `./bin/setup` que impede a execução.

4 Execução

Seguindo as instruções do *HACKING.md*, é esperado que após a execução dos scripts `./bin/setup` de cada microserviço, a plataforma estará rodando normalmente, mas na realidade, nenhum serviço estará respondendo aos requests. É também preciso subir os contêineres de cada serviço de acordo com os seus respectivos *scripts*. Não há nenhuma menção à isso na documentação.

Após chegar nesse ponto, não fica mais claro se os scripts `./bin/setup` devem ser executados na máquina *host* ou dentro dos contêineres. Os seguintes problemas precisam ser superados para a correta execução da plataforma:

- O serviço *data-collector* falha pois no seu arquivo `docker-compose.yml` está faltando o contêiner do **PostgreSQL**. Para consertar é necessário adicionar um novo contêiner *data-collector-pg*, e configurar o *data-collector* com as variáveis de ambiente necessárias para encontrar nova base de dados.
- O serviço *resource-adaptor-sidekiq* não consegue achar na rede o contêiner do *redis*. Para isso, é necessário alterar o arquivo `./config/initializers/sidekiq.rb` e adicionar o prefixo **redis://** à variável `config.redis.url`.