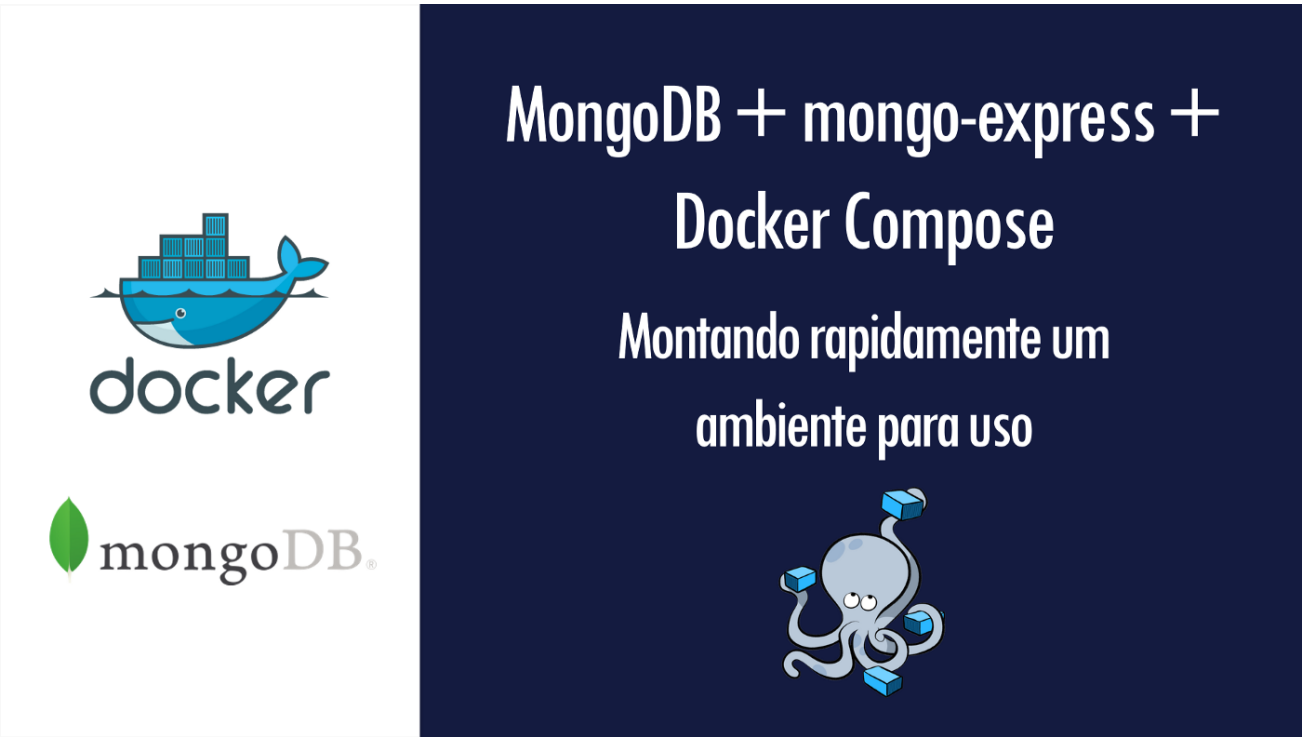
MongoDB + mongo-express + Docker Compose: montando rapidamente um ambiente para uso

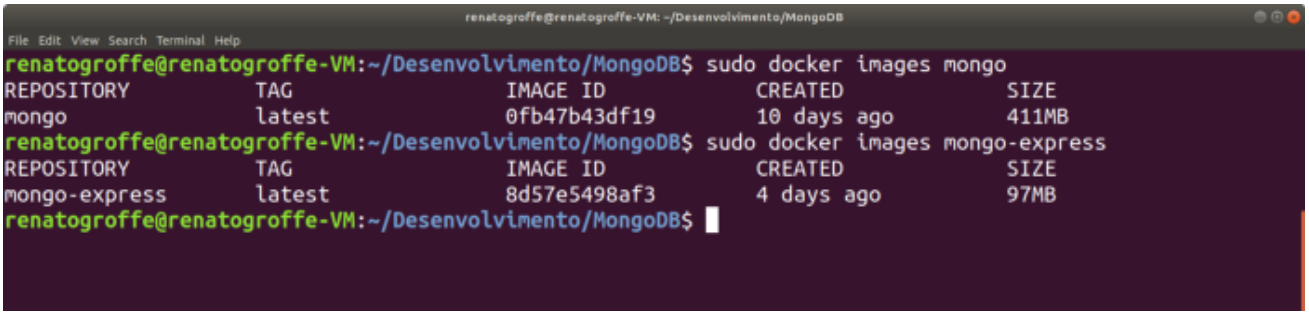
Renato Groffe Jun 9, 2019 · 4 min read



Neste artigo demonstro como criar rapidamente um ambiente para Desenvolvimento/Testes e baseado em **containers Docker** com o **MongoDB**, empregando para isto o **mongo-express** (interface de adminstração do **MongoDB** via Web) e o **Docker Compose**.

> *E aproveito este espaço para deixar aqui também um convite.*
>
> *Dia **13/06/2019 (quinta-feira) às 21:30 — horário de Brasília —** teremos mais um **evento online gratuito** no **Canal .NET**. Desta vez farei uma apresentação online cobrindo as **principais novidades do .NET Core 3.0 e do ASP.NET Core 3.0.***
>
> *Para efetuar a sua inscrição acesse a **página do evento no Meetup**. A transmissão acontecerá via **YouTube**, em um link a ser divulgado em breve.*

Para este exemplo serão utilizadas as imagens **mongo** e **mongo-express**:

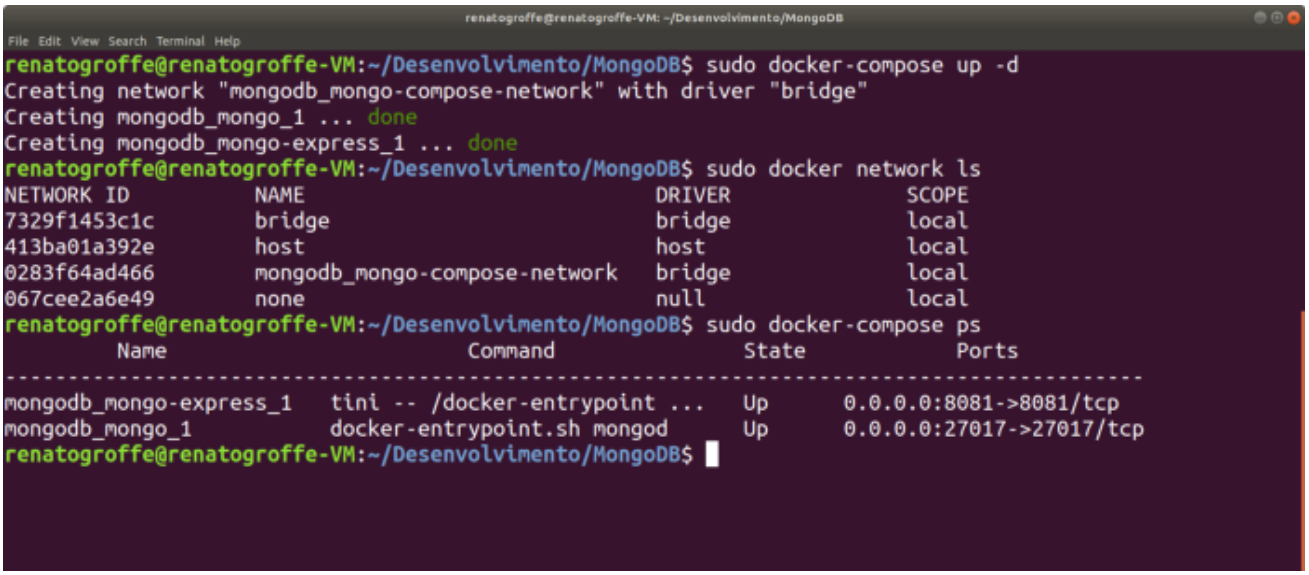


Na listagem a seguir está o conteúdo do arquivo **docker-compose.yml** que permitirá a criação do ambiente citado (**MongoDB** + **mongo-express**). Os testes descritos neste artigo foram realizados no **Ubuntu Desktop 18.04**:

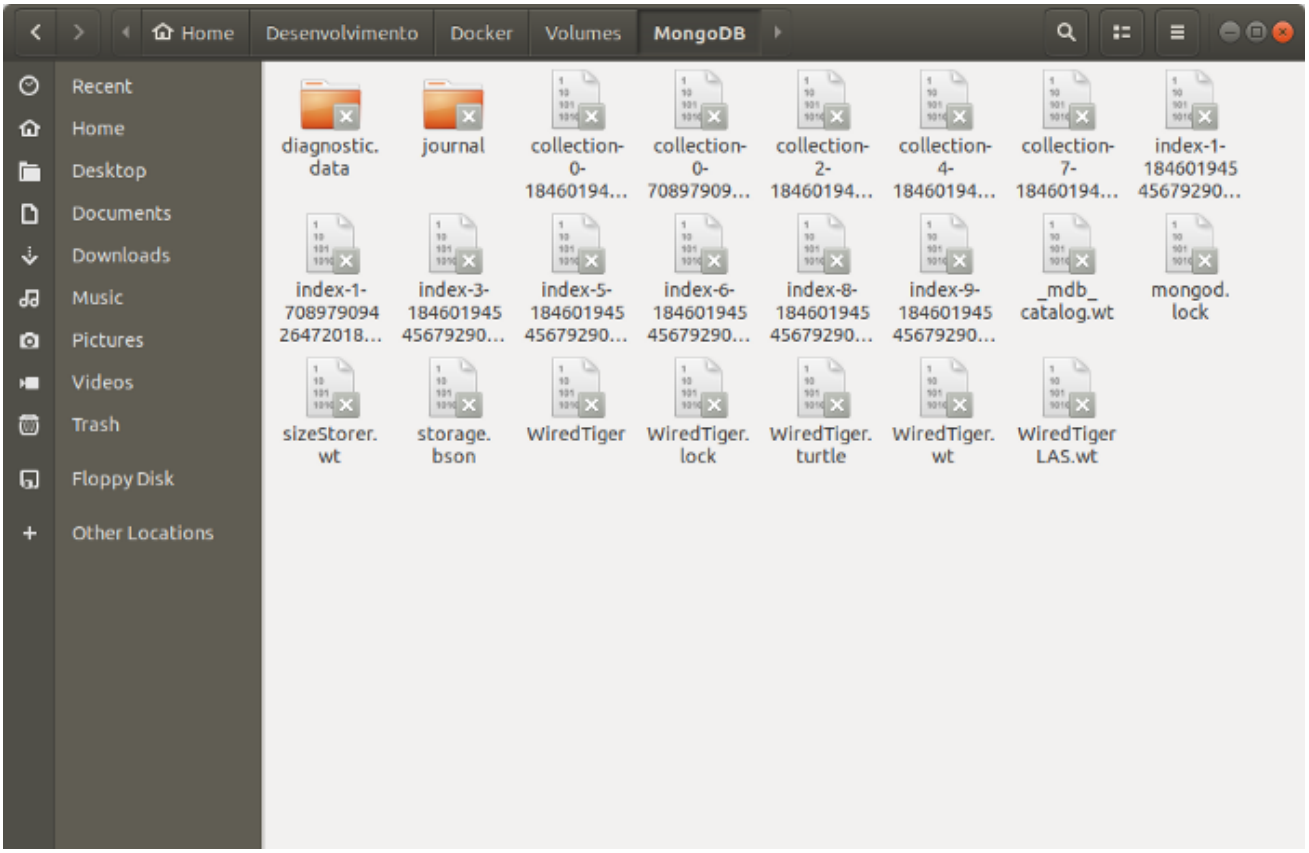
- O serviço **mongo** se refere à instância do **MongoDB** a ser criada para acesso na porta **27017**;
- Já o serviço **mongo-express** corresponde ao container que permitirá a execução da aplicação Web para administração da instância do **MongoDB** (imagem **mongo-express**) a partir da **porta 8081**;
- Nas seções **environment** dos serviços **mongo** e **mongo-express** foram definidas configurações (variáveis de ambiente) necessárias para a geração dos 2 containers;
- Nas variáveis **MONGO_INITDB_ROOT_USERNAME** e **MONGO_INITDB_ROOT_PASSWORD** do serviço **mongo** foram especificados um login de usuário (**root**) e sua respectiva senha de acesso à instância do **MongoDB**;
- Em **mongo-express** a seção **links** indica a ligação deste container com o serviço **mongo**. Nas variáveis **ME_CONFIG_BASICAUTH_USERNAME** e **ME_CONFIG_BASICAUTH_PASSWORD** estão o usuário (**renatogroffe**) e a senha de acesso ao **mongo-express**, respectivamente. Já em **ME_CONFIG_MONGODB_PORT**, **ME_CONFIG_MONGODB_ADMINUSERNAME e ME_CONFIG_MONGODB_ADMINPASSWORD** foram especificadas as configurações de acesso (porta, usuário e senha) para que o **mongo-express** acesse a instância do **MongoDB**;

- As imagens referenciadas serão baixadas caso ainda não existam no ambiente a partir do qual o **Docker Compose** foi executado;
- Foi especificado ainda um volume para **mongo**, indicando com isto o diretório no **Ubuntu Desktop** em que serão gravados os arquivos de dados (**/home /renatogroffe/Desenvolvimento/Docker/Volumes/MongoDB**);
- Por meio da network **mongo-compose-network** acontecerá a comunicação entre os containers **mongo** e **mongo-express**.

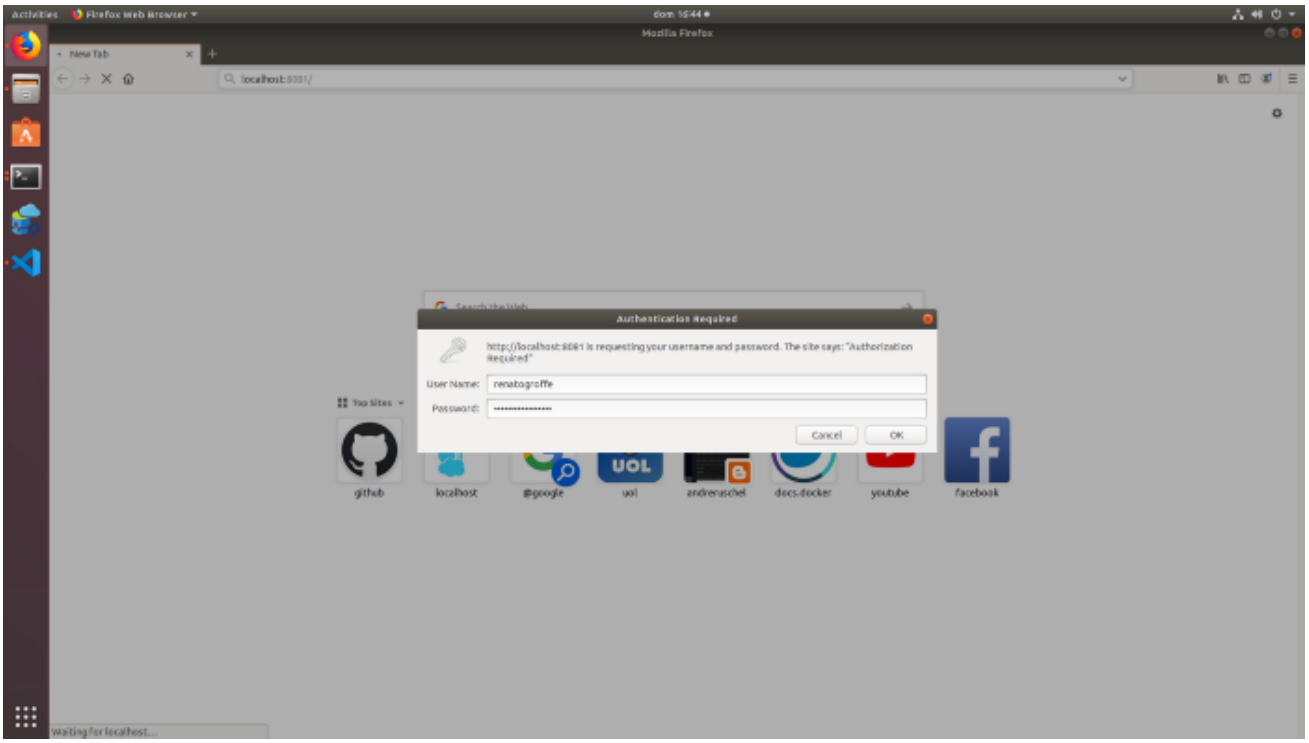
O comando **docker-compose up -d** efetuará a criação da network e dos containers indicados no arquivo **docker-compose.yml**. A instrução docker **network ls** mostrará que a rede **mongo-compose-network** foi gerada, ao passo que o comando **docker-compose** listará os containers criados:



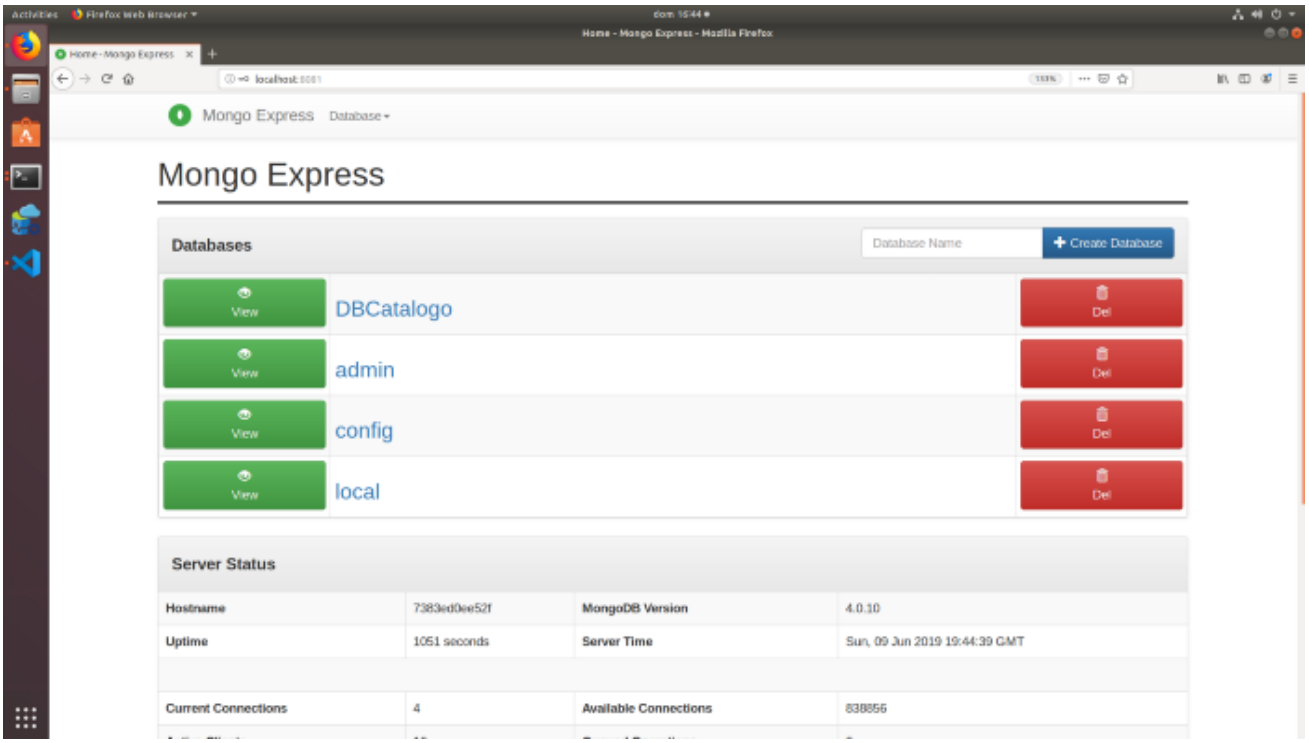
Já a próxima imagem traz os arquivos e diretórios criados para o volume definido em **docker-compose.yml**:

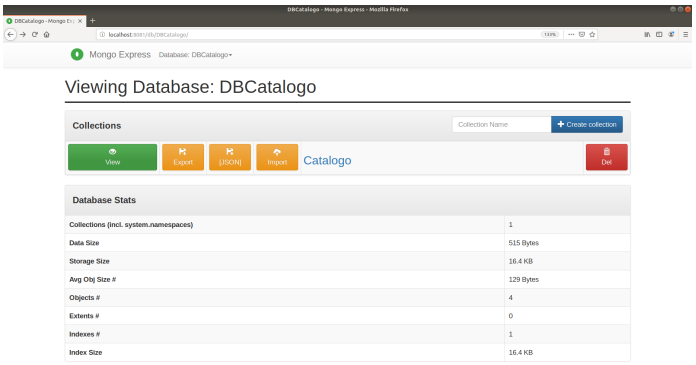


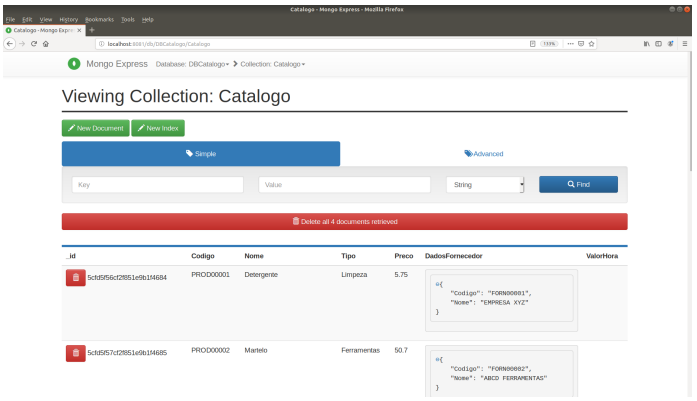
Ao acessar o endereço **http://localhost:8081** via browser aparecerá inicialmente uma janela solicitando as credenciais para uso do **mongo-express**:



Nas próximas imagens é possível observar o **mongo-express** conectado à instância do **MongoDB**, bem como a existência de um banco criado para testes (**DBCatalogo**) e uma coleção vinculada ao mesmo (**Catalogo**):







Referências

Docker para Desenvolvedores .NET - Guia de Referência

mongo - Docker Hub

mongo-express - Docker Hub