

[Inscrever-se](#)[Entrar](#)

Meio de pesquisa



Escrever



Reduza o tempo de inicialização do banco de dados de 45 para 3 minutos no Oracle 19.3.0 dockerizado

Grzegorz Gajos · [Seguir](#)

4 minutos de leitura · 7 de janeiro de 2020



27



2





Quando criei a imagem docker do Oracle EE 19.3.0 usando o guia oficial, o tempo de inicialização foi de mais de 45 minutos. Eu queria usar isso durante o teste de integração, então o tempo não era aceitável. Consegui diminuir esse número para menos de 3 minutos. Abaixo, apresentarei um guia sobre como fazer isso. Além do uso do recurso de encaixe não tão popular. Fique atento!

Construindo e executando oracle/docker-image 19.3.0

1. Baixe o projeto oficial do GitHub <https://github.com/oracle/docker-images>
2. Baixe os arquivos oficiais de instalação do banco de dados
`LINUX.X64_193000_db_home.zip` do site da Oracle

<https://www.oracle.com/database/technologies/oracle19c-linux-downloads.html>

3. Coloque-o em um diretório apropriado no

projeto /OracleDatabase/SingleInstance/dockerfiles/19.3.0

4. **Importante!** Abra

/OracleDatabase/SingleInstance/dockerfiles/19.3.0/Dockerfile O

arquivo e remova a linha `VOLUME [“$ORACLE_BASE/oradata”]` - direi o porquê em um momento.

5. Executar `./buildDockerImage.sh -v 19.3.0`

6. Espere.

Quando o último comando terminar (e levará algum tempo). Você deve ter seu Oracle Database disponível no repositório docker local. Agora, você pode iniciar a imagem.

```
docker run -d -p 5500:5500 -p 1521:1521 -e ORACLE_PWD=GG_PASS  
oracle/database:19.3.0-ee
```

O comando deve levar cerca de 45 minutos e seu banco de dados deve estar funcionando. Você pode estar um pouco surpreso aqui, mas sim, iniciar um novo banco de dados Oracle não é muito rápido. O banco de dados executará algo como uma instalação inicial na primeira execução. Essa abordagem faz sentido, pois economiza muito espaço em disco.

Como configurar o banco de dados e evitar a espera de uma inicialização?

Quando você está iniciando e parando o container, a inicialização do banco de dados é bastante rápida (minutos). Esse benefício não funcionará para o processo de teste ou CI que está usando uma imagem nova. Durante os testes, você espera iniciar um banco de dados com alguma configuração mínima rapidamente. O mais rápido possível. Não em ridiculamente 45 minutos. Isto não é aceitável. Mas vamos voltar à configuração primeiro.

A forma “limpa” de configuração no mundo docker aponta para Dockerfile. Adicione alguns comandos lá e construa a imagem. Novamente, isso levará uma hora. Se você cometer um erro, algumas horas. Aqui, gostaria de focar no desempenho, e não nas melhores práticas. A maneira mais fácil de configuração do banco de dados é conectar-se ao banco de dados em execução e executar alguns comandos SQL. Vamos fazer isso via `sqlplus`.

1. Localize o id do banco de dados dockerizado: `docker ps`
2. Abra o bash dentro do contêiner do banco de dados: `docker exec -it 0f4acaaca803 /bin/bash`
3. Inicie o terminal do banco de dados: `sqlplus sys/GG_PASS@//localhost:1521/ORCLCDB as sysdba`

Neste ponto, podemos executar quaisquer comandos que desejarmos para configurar nosso banco de dados. Vamos criar um usuário e conceder alguns por.

```
CREATE USER GG_TEST identificado BY GG_TEST;  
CONCEDA CRIAR QUALQUER VISUALIZAÇÃO MATERIALIZADA, CRIAR TRABALHO,  
CONECTAR, RECURSOS, CRIAR QUALQUER VISUALIZAÇÃO PARA GG_TEST;  
GRANT TABLESPACE ILIMITADO PARA GG_TEST;
```

Parabéns! Configuração concluída. Mas espere. A configuração foi aplicada durante o tempo de execução do contêiner. Não podemos iniciar esse contêiner em outro lugar, certo? Errado! gostaria de apresentar `docker commit`.

Configuração de contêiner muito conveniente

Para salvar um instantâneo de nossa configuração atual, você pode executar o seguinte comando

```
docker commit --author "Autor <seu@email>" --message "Quick snapshot" 0f4acaaca803 ggajos/oracle-ee:19c-quick
```

Este comando vai pausar nosso contêiner de banco de dados e executar um instantâneo. O resultado será convertido em uma imagem e marcado como `ggajos/oracle-ee:19c-quick`. Agora podemos recriar o contêiner daquele momento específico em qualquer lugar. Também é muito útil se você quiser redefinir rapidamente o estado do banco de dados. Para executar nosso instantâneo, podemos usar nossa tag.

```
docker run -d -p 5500:5500 -p 1521:1521 ggajos/oracle-ee:19c-quick
```

Feito! O tempo de inicialização é muito curto (minutos), pois o banco de dados já está iniciado e configurado! Incluindo toda a personalização de tempo de execução aplicada antes de executar `docker commit` o comando.

Por que removemos o VOLUME do Dockerfile?

Nada funcionará se usarmos volumes externos. Essa é a razão pela qual o comando de volume foi removido do Dockerfile inicial. O comando de confirmação não é capaz de executar instantâneos de dados de volume externo (e não deveria). Normalmente, você deseja seus arquivos de dados fora do contêiner. Mas não neste caso. Temos que colocar todos os arquivos dentro do container para podermos gerenciar todo o estado.

Atenção! Não Exagere!

Embora tenhamos alcançado um desempenho matador (no mundo Oracle), existem algumas desvantagens.

1. Não podemos mais passar parâmetros na inicialização. A imagem oficial do docker Oracle permite passar senhas `-e ORACLE_PWD=GG_PASS` ou outros argumentos. O instantâneo de banco de dados contém um banco de dados já iniciado. Os argumentos já foram consumidos.
2. O tamanho da imagem é muito maior. A imagem do docker Oracle EE vanilla tem cerca de ~ 5 GB. A execução das operações acima pode produzir uma imagem maior que 10 GB. No meu caso foram 13GB.
3. A imagem não tem Dockerfile. Você não pode usá-lo na `FROM` cláusula. O ciclo de vida torna-se manual.

Você precisa considerar se todas as desvantagens estão ok no seu caso. Ser capaz de realizar testes de integração completos no banco de dados de destino é um grande benefício. O tempo de teste é importante. Executar o Oracle DB localmente em minutos ou no CI é definitivamente um grande recurso.

Docker

banco de dados Oracle

Dockerfiles

Integração contínua

Oráculo

**Escrito por Grzegorz Gajos**

41 seguidores


Engenheiro de software, co-fundador da opentangerine.com. Dono de uma experiência real, brutal e prática em diversas áreas de TI desde 2009.

Seguir



Mais de Grzegorz Gajos



 Grzegorz ... em FAUN — Comunidade de desenv...

Oi Docker! Por que você odeia tanto o Windows?






Você está usando o Windows. Você está usando o Docker. Sua vida está em perigo! Ir...


8 minutos de leitura · 17 de dezembro de 2018

 239  12



id12492.hprof

Aggregation:  Details:  Preview  Fields  References 	
	Count
piRequest	540,218
	1,316
	3,764
	3,656
	1,789
HashMap\$Node	1,740
	341

 Grzegorz Gajos

Exceção de falta de memória em Java


Neste artigo, mostrarei como depurar os amados erros java.lang.OutOfMemoryError...

2 minutos de leitura · 9 de julho de 2018

 22  2





 Grzegorz Gajos


Como o Hibernate quase arruinou...

Imagine que você é um desenvolvedor Java e está prestes a iniciar seu próximo grande...

10 minutos de leitura · 14 de abril de 2017

 15  1



 Grzegorz Gajos

Java 9. Partes de duração, truncamento e divisão.

O Java 9 está introduzindo alguns novos métodos na classe Duration. Antes de...

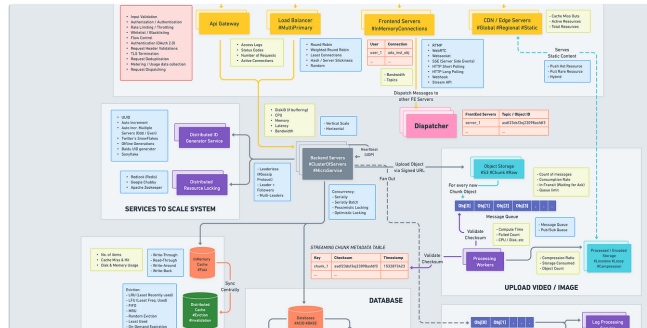
2 minutos de leitura · 19 de abril de 2017

 3 



See all from Grzegorz Gajos

Recommended from Medium



 Love Sharma in Dev Genius

System Design Blueprint: The Ultimate Guide

Developing a robust, scalable, and efficient system can be daunting. However,...

🌟 · 9 min read · Apr 20

 2.8K  25



```
commit ffc72c81b7ef612893529cef388cc1961ed64521 (HEAD -> master, origin/master, origin/bors/staging, origin/HEAD)
Merge: fc991bf81 5159211da
Author: lohk-bors[bot] <43231472@users.noreply.github.com>
Date: Tue Nov 8 17:44:34 2022 +0000

Merge #4553

4563: New p2p topology file format r=coot a=coot

Flxes #4559.

Co-authored-by: Marcin Szamotulski <coot@coot.me>
Co-authored-by: olgahryniuk <67585499@users.noreply.github.com>

commit fc991bf814891a9349f22cf7278632d39b84d4628
Merge: 5632d1c85 5cd94d372
Author: lohk-bors[bot] <43231472@users.noreply.github.com>
Date: Tue Nov 8 13:07:58 2022 +0000

Merge #4613

4613: Update building-the-node-using-nix.md r=CarlosLopezDeLara a=CarlosLopezDeLara

Build the cardano-node executable. No default configuration.

Co-authored-by: CarlosLopezDeLara <carlos.lopezdelara@iohk.io>

commit 5159211da7a644686a973e4fb316b64eb1aa34c
Author: olgahryniuk <67585499@users.noreply.github.com>
Date: Tue Nov 8 13:25:18 2022 +0000
```

 Jacob Bennett in Level Up Coding

Use Git like a senior engineer

Git is a powerful tool that feels great to use when you know how to use it.

🌟 · 4 min read · Nov 15, 2022

 5.7K  62



Lists



Staff Picks

342 stories · 102 saves



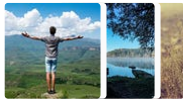
Stories to Help You Level-Up at Work

19 stories · 87 saves



Self-Improvement 101

20 stories · 146 saves



Productivity 101

20 stories · 162 saves



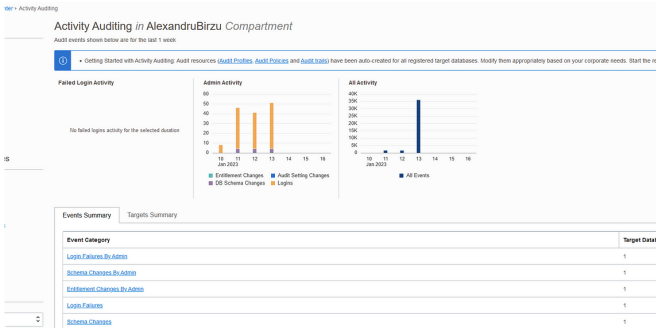
Tamir Suliman in Coinmonks

Deploy Elasticsearch & Kibana Environment with Docker

A pilha ELK, que inclui Elasticsearch, Kibana e Logstash, é considerada uma das...

🌟 · 7 minutos de leitura · 5 de março

👏 30 💬 1



Birzu Alexandru-Adrian em Aprenda OCI

Como ingerir eventos de auditoria de segurança de dados no log OCI


Uma das principais vantagens de rodar Bancos de Dados Oracle em OCI, além da...

🌟 · 7 minutos de leitura · 16 de janeiro

👏 💬





 Alexandre Nguyen em Codificação de nível


Por que continuo reprovando candidatos durante as entrevista...

Eles não atendem a barra.

★ · 4 minutos de leitura · 13 de abril

 4,5K  133



 Nick Fothergill em TechieLife

Como: criar um cluster de armazenamento de objetos de...

O armazenamento de objetos é o serviço de arquivo nativo da nuvem usado por...

★ · 6 minutos de leitura · 15 de fevereiro

 1 



Ver mais recomendações