



Prof. Jessen Vidal

**SHAM VINICIUS FIORIN**

**SERVIDORES EM UM REPOSITORIO  
DOCKER**

**SÃO JOSÉ DOS CAMPOS**

**2017**

**SHAM VINICIUS FIORIN**

**SERVIDORES EM UM REPOSITORIO DOCKER**

**Relatorio final de sistemas operacionais 2**

Trabalho de Sistemas Operacionais  
ao Curso de Análise e  
Desenvolvimento de Sistemas, da  
Faculdade de Tecnologia de São  
José dos Campos

Orientador: Prof. Jean Carlos  
Lourenco Costa

**SÃO JOSÉ DOS CAMPOS**

**2017**

## INDICE

Introdução.....	4
Secure Shell (SSH).....	5
File Transfer Protocol (FTP).....	8
MySQL.....	10
PHPMyAdmin.....	12
Recursos do PHPMyAdmin.....	14
MongoDB.....	17
Docker.....	19
Conclusão.....	23
Referências.....	24

## INTRODUÇÃO

Esse trabalho é sobre servidores em um ambiente docker, com objetivo de usar um ambiente mais leve e seguro para rodar servidores linux e ter acesso posteriormente a repositórios remotos. O trabalho está dividido em 7 capítulos com um servidor para cada, que são SSH, FTP, MySQL, PHPMyAdmin, Recusos do PHPMyAdmin, MongoDB e Docker. Em um sistema Ubuntu 16.04 rodando um Docker com container de uma imagem Debian 8 “Jessie”, conectado a uma rede wifi com acesso a internet.

## Secure Shell (SSH)

Instalando servidor Secure Shell em um container Debian

```
root@lab: /home/fatec/Documentos
root@relatorio:/# apt install openssh-server
```

Ativando serviço emulação do servidor SSH

```
root@relatorio:/home/sham# service ssh start
[ ok ] Starting OpenBSD Secure Shell server: sshd.
```

Fazendo conexão ao servidor SSH

```
sham@relatorio: ~
root@lab:/home/fatec# ssh sham@172.17.0.2
sham@172.17.0.2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jun  8 11:49:39 2017 from 172.17.0.1
sham@relatorio:~$

root@lab: /home/fatec/Documentos
root@relatorio:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe11:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:67 errors:0 dropped:0 overruns:0 frame:0
          TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:10126 (9.8 KiB)  TX bytes:6390 (6.2 KiB)
```

Acessando configurações do servidor SSH

```
root@relatorio:/home/sham# nano /etc/ssh/sshd_config
```

Alterando a porta padrão 22 do servidor SSH para porta 2222

```
GNU nano 2.2.6      File: /etc/ssh/sshd_config      Modified
# Package generated configuration file
# See the sshd_config(5) manpage for details
# What ports, IPs and protocols we listen for
Port 2222
```

Limitar o acesso ao SSH, liberando somente para usuario específicos no caso “sham”

```
root@relatorio:/home/sham# nano /etc/ssh/sshd_config
```

```
AllowUsers sham
```

Fazendo download do servidor Secure Shell

Repositório no Docker

```
root@relatorio:/home/sham# ls
download.txt
```

Fazendo download do servidor SSH

```
root@lab:/home/fatec# scp -P 2222 sham@172.17.0.2:/home/sham/download.txt /home/fatec/Documentos/download.txt
sham@172.17.0.2's password:
download.txt                                100%   9   0.0KB/s   00:00
root@lab:/home/fatec#
```

Fazendo upload para o servidor Secure Shell

Fazendo upload do servidor SSH

```
root@lab:/home/fatec# scp -P 2222 /home/fatec/Documentos/upload.txt sham@172.17.0.2:/home/sham/upload.txt
sham@172.17.0.2's password:
upload.txt                                100%   8   0.0KB/s   00:00
root@lab:/home/fatec#
```

## Repositório no Docker

```
root@relatorio:/home/sham# ls
download.txt  upload.txt
root@relatorio:/home/sham#
```

# File Transfer Protocol (FTP)

Instalando o proftpd

```
sham@sham: ~  
root@relatorio:/# apt-get install proftpd
```

Configurando o proftpd

```
root@relatorio:/# nano /etc/proftpd/proftpd.conf
```

Acessando o servidor FTP

Com o Comando “ftp <ip servidor>” acessa um servidor ftp

```
root@relatorio:/# ifconfig  
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02  
          inet addr:172.17.0.2  Bcast:0.0.0.0  Mask:255.255.0.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:3873  errors:0  dropped:0  overruns:0  frame:0  
  
root@sham: /home/sham  
root@sham:/home/sham# ftp 172.17.0.2  
Connected to 172.17.0.2.  
220 ProFTPD 1.3.5 Server (Debian) [::ffff:172.17.0.2]  
Name (172.17.0.2:root): sham  
331 Password required for sham  
Password:  
230 User sham logged in  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp>
```

Fazendo download e upload do servidor FTP

Download



```
ftp> get arquivoDoServidor
local: arquivoDoServidor remote: arquivoDoServidor
200 PORT command successful
150 Opening BINARY mode data connection for arquivoDoServidor (9 bytes)
226 Transfer complete
9 bytes received in 0.00 secs (54.9316 kB/s)
```

### Upload

```
ftp> put vazio.txt
local: vazio.txt remote: vazio.txt
200 PORT command successful
150 Opening BINARY mode data connection for vazio.txt
226 Transfer complete
```

Limitando o acesso ao servidor FTP

### Prendendo o usuario a seu home

```
# Use this to jail all users in their homes
DefaultRoot ~
```

### Desativando login com root

```
RootLogin off
```

# MySQL

Instalando Servidor MySQL

```
root@relatorio:/home/sham# apt install mysql-server
```

Iniciando serviço do servidor MySQL

```
root@relatorio:/home/sham# service mysql start
```

Conectando ao servidor MySQL via localhost

```
root@relatorio:/home/sham# mysql -h localhost -u root -p
```

Conectando via Cliente

Instalando cliente MySQL

```
root@sham:/home/sham# apt install mysql-client-core-5.7
```

Acesse /etc/mysql/my.cnf

```
root@relatorio:/home/sham# nano /etc/mysql/my.cnf
```

Comente a Linha Bind-Address com '#' dessa forma

```
#bind-address = 127.0.0.1
```

Conect pelo ip do servidor mysql

```
root@sham:/home/sham# mysql -h 172.17.0.2 -P 3306 -u fatec -p
```

Aplicação usando servidor MySQL com ajuda modulo MySQLDB de python

Instalando modulo python mysqldb

```
#sudo apt install python-mysqldb
```

## Script acesso a tabela USUARIO no banco REGISTRO

### Via Cliente

```
import MySQLdb

db = MySQLdb.connect(host="172.17.0.2", user="fatec", passwd="aluno", db="REGISTRO")
cursor = db.cursor()

cursor.execute("SELECT * FROM USUARIO")
print(cursor.fetchall())
```

### Via Servidor

```
GNU nano 2.2.6 File: mostrarCadastros.py

import MySQLdb

db = MySQLdb.connect(host="localhost", user="fatec", passwd="aluno", db="REGISTRO")
cursor = db.cursor()

cursor.execute("SELECT * FROM USUARIO")
print(cursor.fetchall())
```

Executando Script ele retorna os valores nome, rg da tabela USUARIO

```
root@sham:/home/sham/Documentos# python mostrarCadastros.py
(('SHAM', 1234L), ('PEDRO', 1234L), ('ARUA', 1234L), ('MARIA', 1234L), ('BRUNA', 1234L), ('ISA', 1234L))
```

# PHPMyAdmin

Instalando PHPMyAdmin e suas dependencias

```
sham@sham: ~  
root@relatorio:/# apt-get install apache2 php5 libapache2-mod-php5 mysql-server php5-mysql phpmyadmin
```

Ativando o apache2

```
root@relatorio:/home/sham# a2enmod php5
```

Ativando o servidor do apache2

```
root@relatorio:/home/sham# service apache2 start
```

Concedendo permissão para acesso

```
root@relatorio:/home/sham# chmod 777 /var/www
```

Configurando

apache para funcionar com o PHPMyAdmin

```
Include /etc/phpmyadmin/apache.conf
```

Conectando ao PHPMyAdmin

No endereço do browser digite <http://ipDoServidor/phpmyadmin>

Página Inicial de Login



Welcome to phpMyAdmin

Language

English

Log in

Username:

Password:

Go

# Recursos PHPMyAdmin

## Controle de Usuario

### Users overview

	User	Host	Password	Global privileges	Grant	Action
	debian-sys-maint	localhost	Yes	ALL PRIVILEGES	Yes	Edit Privileges  Export
	fatec	%	Yes	USAGE	No	Edit Privileges  Export
	root	127.0.0.1	Yes	ALL PRIVILEGES	Yes	Edit Privileges  Export
	root	:::1	Yes	ALL PRIVILEGES	Yes	Edit Privileges  Export
	root	localhost	Yes	ALL PRIVILEGES	Yes	Edit Privileges  Export
	root	relatorio	Yes	ALL PRIVILEGES	Yes	Edit Privileges  Export

## Controle de Trafego

[Server](#) [Query statistics](#) [All status variables](#) [Monitor](#) [Advisor](#)

**Network traffic since startup: 395 KiB**

This MySQL server has been running for 0 days, 0 hours, 45 minutes and 18 seconds. It started up on Jun 09, 2017 at 06:20 PM.

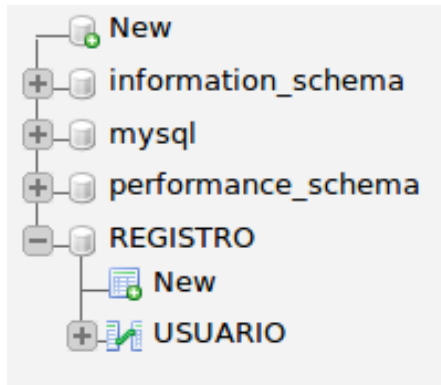
Traffic	Ø per hour	
Received	184.4 KiB	138.3 KiB
Sent	290.5 KiB	384.8 KiB
Total	395 KiB	523.1 KiB

Connections	Ø per hour	%
max. concurrent connections	3	---
Failed attempts	45	59.6 34.09%
Aborted	1	1.32 0.76%
Total	132	174.83 100.00%

Processes	ID	User	Host	Database	Command	Time	Status	SQL query
Kill	131	root	localhost	mysql	Query	0	---	

[SHOW PROCESSLIST](#)

## Criar tabelas e bancos de dados



## Importar databases

### Importing into the current server

#### File to Import:

File may be compressed (gzip, bzip2, zip) or uncompressed.

A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer:  Nenhum arquivo selecionado. (Max: 2,048KiB)

Character set of the file:

#### Partial Import:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

Skip this number of queries (for SQL) or lines (for other formats), starting from the first one:

#### Format:

#### Format-Specific Options:

SQL compatibility mode:

☒ Do not use AUTO\_INCREMENT for zero values

## Exportar Databases

# Exporting databases from the current server

## Export Method:

---

- ☒ Quick - display only the minimal options
- ☐ Custom - display all possible options

## Format:

---

SQL

Go





# MongoDB


## Instalando servidor MongoDB

### Download do MongoDB para seu sistema

**Current Stable Release (3.4.4)**  
04/20/2017: [Release Notes](#) | [Changelog](#)  
Download Source: [tgz](#) | [zip](#)

 Windows

 Linux

 OSX

Solaris

Select your distribution from the list or the legacy Linux 64-bit version if your distribution is unavailable. Keep in mind that this legacy Linux 64 build may lack the performance optimizations present in targeted builds.


Version:  

Debian 8 Linux 64-bit x64 ▾

The binary of this version has been compiled with SSL enabled and dynamically linked. This requires that SSL libraries be installed separately. See [here](#) for more information on installing OpenSSL.

Package Manager:  
[Instructions for installing with apt](#)

Binary: [Installation Instructions](#) | [All Version Binaries](#)

 [DOWNLOAD \(tgz\)](#) [https://fastdl.mongodb.org/linux/mongodb-linux-x86\\_64-debian81-3.4.4.tgz](https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-debian81-3.4.4.tgz)

### Descompactando arquivo tipo .tgz

```
root@relatorio:/home/sham# tar -xvzf mongodb-linux-x86_64-debian81-3.4.4.tgz
```

### No diretorio criado /bin

```
root@relatorio:/home/sham/mongodb-linux-x86_64-debian81-3.4.4/bin#
```

### Executar o servidor mongoDB

```
root@relatorio:/home/sham/mongodb-linux-x86_64-debian81-3.4.4/bin# ./mongod
```

### Conexão iniciada

```
2017-06-09T19:26:25.984+0000 I NETWORK [thread1] connection accepted from 127.0.0.1:
33500 #1 (1 connection now open)
2017-06-09T19:26:25.985+0000 I NETWORK [conn1] received client metadata from 127.0.0
.1:33500 conn1: { application: { name: "MongoDB Shell" }, driver: { name: "MongoDB In
ternal Client", version: "3.4.4" }, os: { type: "Linux", name: "PRETTY_NAME="Debian G
NU/Linux 8 (jessie)"" , architecture: "x86_64", version: "Kernel 4.8.0-54-generic" } }
```

Acessando servidor mongoDB

Atravez de acesso via SSH

```
sham@sham:~/Documentos$ ssh sham@172.17.0.2
```

Conectando ao servidor MongoDB

```
sham@relatorio:~/mongodb-linux-x86_64-debian81-3.4.4/bin$ ./mongo
```

Explorando MongoDB

```
> use usuario
switched to db usuario
> db.usuario.save({nome: "Pedro Pilla"})
WriteResult({ "nInserted" : 1 })
> db.usuario.save({nome: "Aruã"})
WriteResult({ "nInserted" : 1 })
> db.usuario.find()
{ "_id" : ObjectId("593af7e75e3b411ea04f4403"), "nome" : "Sham" }
{ "_id" : ObjectId("593af8bb5e3b411ea04f4404"), "nome" : "Pedro" }
{ "_id" : ObjectId("593af8d15e3b411ea04f4405"), "nome" : "Pedro Pilla" }
{ "_id" : ObjectId("593af8db5e3b411ea04f4406"), "nome" : "Aruã" }
>
```

# Docker

Instalando o Docker através do script disponibilizado em <https://get.docker.com>

Instalando o Curl

```
sham@sham: ~  
root@relatorio:/# apt install curl
```

Fazendo download do script

```
sham@sham: ~  
root@relatorio:/# curl https://get.docker.com > /tmp/install.sh
```

Concedendo permissão

```
sham@sham: ~  
root@relatorio:/# chmod +x /tmp/install.sh
```

Conferindo Script

```
sham@sham: ~  
root@relatorio:/# cat /tmp/install.sh
```

Executando o Script

```
sham@sham: ~  
root@relatorio:/# ./tmp/install.sh
```

Usando o Docker

Iniciando e baixando um container

```
root@sham:/home/sham# docker run -h NomeDaMaquina -it debian /bin/bash
```

Criando uma imagem de container

```
root@sham:/home/sham# docker commit 916e4dd4b01a debian:relatoriov1
```

Iniciando imagem existente

```
root@sham:/home/sham# docker run -h relatorio -it debian:relatoriov1
```

Mostrando containers sendo executados

```
root@sham:/home/sham# docker ps
```

Acessando container iniciado

```
root@sham:/home/sham# docker attach infallible_brahmagupta
```

Exportando imagem de container para .tar

```
root@sham:/home/sham# docker save -o /home/sham/Documentos/RelatorioFinal.tar skatesham/debian:completo
```

Importando container .tar

```
root@sham:/home/sham/Documentos# docker load -i debianrelatorio.tar
```

Fazendo upload de um container para o Docker Hub

Fazendo login no Docker

```
root@sham:/home/sham# docker login -u login -p senha
```

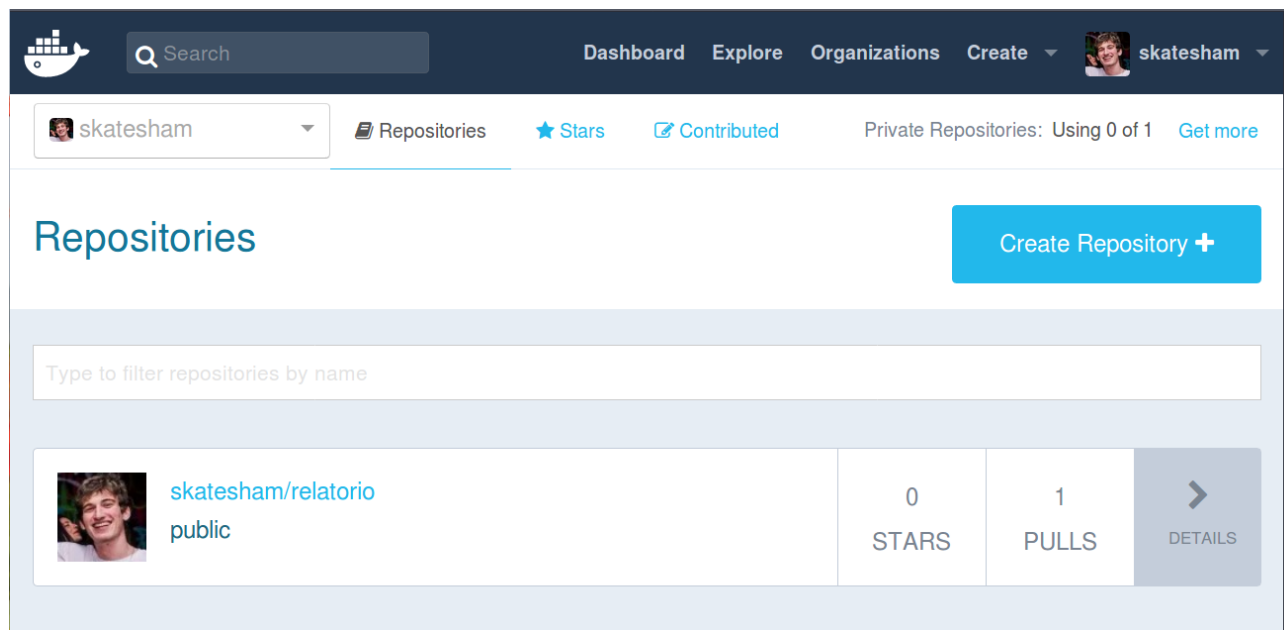
Criando uma tag

```
root@sham:/home/sham# docker tag debian:relatorio skatesham/debian:completo
```

Enviando para o Docker Hub

```
#docker push skatesham/debian:completo
```

Visualizando o Docker Hub



Download da Imagem do container do trabalho

Para fazer download da Imagem do container debian da trabalho com os servidores configurado, está disponivel em skatesham/debian:completo

```
root@sham:/home/sham# docker pull skatesham/debian:completo
completo: Pulling from skatesham/debian
Digest: sha256:33475ac3217ec3567163c9782f4ad390a79768f82214625dd9c5f6fe708ab835
Status: Image is up to date for skatesham/debian:completo
root@sham:/home/sham#
```

Repositório skatesham/debian

Tag completo 425 MB

PUBLIC REPOSITORY

skatesham/debian ☆

Last pushed: 20 minutes ago

[Repo Info](#)

[Tags](#)

[Collaborators](#)

[Webhooks](#)

[Settings](#)

Tag Name	Compressed Size	Last Updated
completo	425 MB	20 minutes ago

## CONCLUSÃO

Neste trabalho abordamos o assunto da instalação e configuração de alguns servidores linux em um container Docker do Debian, podemos notar que sempre que vamos usar pela primeira vez um serviço em um container Docker que acabou de ser iniciado necessita-se de iniciá-lo anteriormente, nota-se para acessar a máquina temos o SSH, para troca de arquivo temos FTP, para Banco de dados o MySQL, acessos via Browser o PHPMyAdmin, MongoDB para bancos não relacionais e um Docker para concluir aprendizagem de instalação do Docker, esse trabalho foi de grande importância para aprendizagem de conhecimento de sistemas Linux, servidores linux, bash, scripts, linha de comando, conexão com bancos de dados e fez migrar meu dispositivo para um sistema linux.

## Referencias

Docker Hub. Acesso em: 09/06/2017. Disponível em: <https://hub.docker.com/>