



qriar labs

Curso de Introdução ao Linux

Introdução ao Linux

Aula 03

Tópico 7

- Acesso a VM e atividades no ambiente;

Tópico 8

- Script básico (uso de shell script);

Tópico 9

- Gerenciamento de processos e pacotes (ps, top, apt);

Tópico
10

- Containers (Docker);

Tópico
11

- Revisão geral dos conceitos aprendidos.

Redes e Conectividade no Linux

Acesso a VM e atividades no ambiente

Comando:

```
mqriar@mdell:~/aula03$ chmod go-wr instance-9_rsa  
mqriar@mdell:~/aula03$ ssh -i instance-9_rsa qriarlabs@EXTERNAL_IP
```

Dando permissões apenas de execução para a chave;

Acessando a VM.

```
mqriar@mdell:~/aula03$ chmod go-wr instance-9_rsa  
mqriar@mdell:~/aula03$ ssh -i instance-9_rsa qriarlabs@EXTERNAL_IP
```

NAME	ZONE	MACHINE_TYPE	ALUNO	INTERNAL_IP	EXTERNAL_IP
instance-1	us-central1-a	e2-small	Wenderson	10.128.0.2	35.226.157.110
instance-2	us-central1-a	e2-small	Edilson	10.128.0.3	34.69.71.235
instance-3	us-central1-a	e2-small	Jorge	10.128.0.4	35.184.22.225
instance-4	us-central1-a	e2-small	Layane	10.128.0.5	34.171.92.82
instance-5	us-central1-a	e2-small	Marcos Printes	10.128.0.6	35.193.194.71
instance-6	us-central1-a	e2-small	Andressa	10.128.0.7	34.30.46.48
instance-7	us-central1-a	e2-small	Allan	10.128.0.8	34.68.145.249
instance-8	us-central1-a	e2-small	Nathan	10.128.0.9	34.136.166.39
instance-9	us-central1-a	e2-small	Diego	10.128.0.10	34.31.215.205
instance-10	us-central1-a	e2-small	Eliene	10.128.0.11	34.30.28.133
instance-11	us-central1-a	e2-small	Ellen	10.128.0.12	34.29.75.34
instance-12	us-central1-a	e2-small	Marcos Okada	10.128.0.13	34.30.31.193
instance-13	us-central1-a	e2-small	Nattan	10.128.0.14	34.132.58.149
instance-14	us-central1-a	e2-small	Nicole	10.128.0.15	35.202.254.6
instance-15	us-central1-a	e2-small	João Victor	10.128.0.16	35.188.190.123
instance-16	us-central1-a	e2-small	Arlene	10.128.0.17	34.123.31.248
instance-17	us-central1-a	e2-small	Mateus	10.128.0.18	35.192.13.205
instance-18	us-central1-a	e2-small	Daiany	10.128.0.19	34.171.212.147
instance-19	us-central1-a	e2-small	Clarice	10.128.0.20	34.134.155.149
instance-20	us-central1-a	e2-small	Katyellen	10.128.0.21	34.135.98.120

Script básico (uso de shell script)

Exemplo de Script Simples

Comando:

nano scriptSimples.sh

Tornando o script executável

chmod +x scriptSimples.sh

Para rodar o script criado

./scriptSimples.sh

```
#!/bin/bash
echo "Meu nome de usuário:"
whoami

echo "Meu host:"
hostname

echo "Meu ip do host:"
hostname -i

echo "Verificando o uso de memória:"
free -m
```


Gerenciamento de Processos e Pacotes (ps, top, apt)

Listar Processos

Comando:

ps aux

Lista todos os processos

```
mqriar@mdell:~$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0 168612 13908 ?        Ss   09:37   0:02 /sbin/init sp
root           2  0.0  0.0      0      0 ?        S    09:37   0:00 [kthreadd]
root           3  0.0  0.0      0      0 ?        I<   09:37   0:00 [rcu_gp]
root           4  0.0  0.0      0      0 ?        I<   09:37   0:00 [rcu_par_gp]
root           5  0.0  0.0      0      0 ?        I<   09:37   0:00 [slub_flushwq]
root           6  0.0  0.0      0      0 ?        I<   09:37   0:00 [netns]
root           8  0.0  0.0      0      0 ?        I<   09:37   0:00 [kworker/0:0H]
root          11  0.0  0.0      0      0 ?        I<   09:37   0:00 [mm_percpu_wq]
root          12  0.0  0.0      0      0 ?        I    09:37   0:00 [rcu_tasks_kt
root          13  0.0  0.0      0      0 ?        I    09:37   0:00 [rcu_tasks_ru
root          14  0.0  0.0      0      0 ?        I    09:37   0:00 [rcu_tasks_tr
root          15  0.0  0.0      0      0 ?        S    09:37   0:00 [ksoftirqd/0]
```

Gerenciamento de Processos e Pacotes (ps, top, apt)

Listar Processos

Comando:

top

Lista todos os processos
em tempo real

```
top - 22:15:13 up 12:37, 1 user, load average: 0,25, 0,11, 0,07
Tarefas: 286 total, 1 em exec., 285 dormindo, 0 parado, 0 zumbi
%CPU(s): 0,2 us, 0,0 sy, 0,0 ni, 99,6 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MB mem : 15751,8 total, 5578,9 livre, 1980,1 usados, 8192,8 buff/cache
MB swap: 2048,0 total, 2048,0 livre, 0,0 usados, 13034,7 mem dispon.
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPO+	COMANDO
2982	mqriar	20	0	5999424	341380	132176	S	0,7	2,1	32:33.64	gnome-s+
16	root	20	0	0	0	0	I	0,3	0,0	0:30.12	rcu_pre+
585	systemd+	20	0	14832	6940	6100	S	0,3	0,0	0:36.82	systemd+
2849	mqriar	39	19	787736	187296	23508	S	0,3	1,2	1:29.64	tracker+
3228	mqriar	20	0	348796	30604	19500	S	0,3	0,2	0:01.14	ibus-ex+
12434	mqriar	20	0	32,7g	403596	256212	S	0,3	2,5	5:47.21	chrome
12481	mqriar	20	0	32,4g	131756	102384	S	0,3	0,8	1:17.59	chrome
21883	mqriar	20	0	555240	53072	40384	S	0,3	0,3	0:01.11	gnome-t+
1	root	20	0	168612	13908	8324	S	0,0	0,1	0:02.46	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.03	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par+
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	slub_fl+
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
8	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
11	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_perc+
12	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tas+

Conceitos Básicos do Docker

Imagens

- Uma imagem de container é um pacote leve, executável e independente que inclui tudo o que é necessário para rodar um pedaço de software, incluindo o código, as ferramentas de sistema, as bibliotecas e as configurações.

Containers

- Um container é a instância em execução de uma imagem. Você pode ter muitos containers rodando a partir da mesma imagem.

Dockerfile

- É um arquivo de texto que contém todas as instruções necessárias para construir uma imagem Docker.

Docker Hub

- É um repositório público para hospedar imagens Docker. Os usuários podem puxar (pull) imagens existentes ou enviar (push) suas próprias.

Containers (Docker)

Preparando ambiente para o Docker

Atualizar a lista de pacotes do APT:

Sudo apt-get update

Instale pacotes que permitem ao **apt** usar repositórios sobre HTTPS:

**sudo apt-get install **

**apt-transport-https **

**ca-certificates **

**curl **

software-properties-common



Containers (Docker)

Preparando ambiente para o Docker

Adicione a chave GPG oficial do Docker:

```
curl -fsSL
```

```
https://download.docker.com/linux/ubuntu/gpg |
```

```
sudo apt-key add -
```

Adicione o repositório do Docker às fontes do APT:

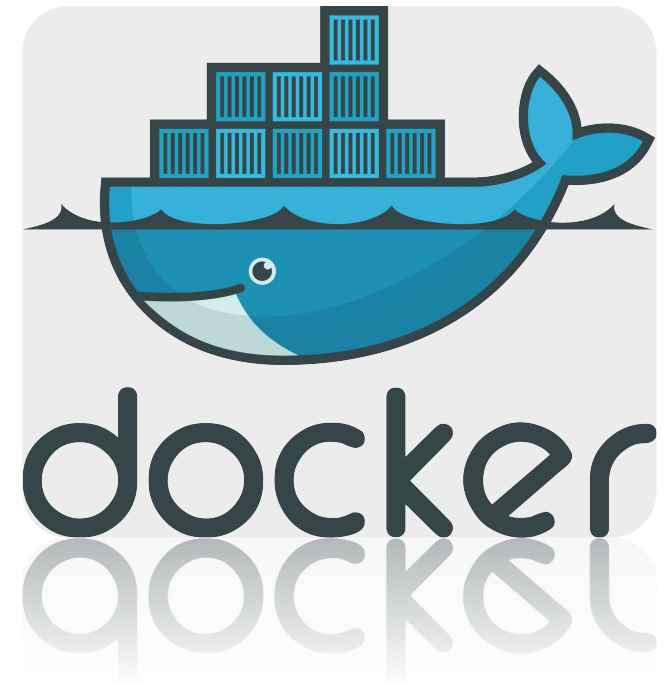
```
sudo add-apt-repository \
```

```
"deb [arch=amd64]
```

```
https://download.docker.com/linux/ubuntu \
```

```
$(lsb_release -cs) \
```

```
stable"
```



Containers (Docker)

Preparando ambiente para o Docker

Atualize a lista de pacotes do APT novamente com os novos repositórios do Docker:

```
sudo apt-get update
```

Instale o Docker Engine:

```
sudo apt-get install docker-ce docker-ce-cli  
containerd.io
```



Containers (Docker)

Preparando ambiente para o Docker

Executar o docker sem a necessidade de usar o sudo para dá permissão:

```
sudo usermod -aG docker seu_usuario
```

you can use the following command to apply the group changes without leaving the session:

```
newgrp docker
```



Containers (Docker)

Preparando ambiente para o Docker

Verifique se o Docker foi instalado corretamente e está rodando:

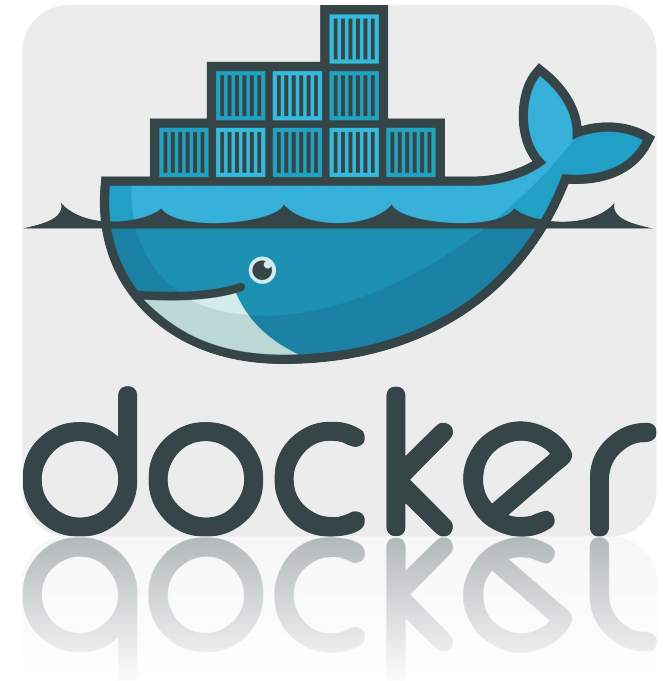
docker ps



Containers (Docker)

MySQL, phpMyAdmin e WordPress usando Docker

O Docker Compose permite definir e rodar aplicações multi-container Docker. Aqui está um exemplo de como o arquivo **docker-compose.yml** pode ser configurado para esse ambiente:



Containers (Docker)

MySQL, phpMyAdmin e WordPress usando Docker

Vamos entrar no diretório ~(home)

criar um diretório chamando **docker**

Criar um arquivo de configuração **dockerfile**

nano dockerfile.yml

Copiar e colar o código que eu passei



```
version: '3.1'

services:

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
    volumes:
      - wordpress_data:/var/www/html
```

```
db:
  image: mysql:5.7
  volumes:
    - db_data:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: somewordpress
    MYSQL_DATABASE: wordpress
    MYSQL_USER: wordpress
    MYSQL_PASSWORD: wordpress

phpmyadmin:
  depends_on:
    - db
  image: phpmyadmin/phpmyadmin
  ports:
    - "8080:80"
  environment:
    PMA_HOST: db
    MYSQL_ROOT_PASSWORD: somewordpress

volumes:
  wordpress_data:
  db_data:
```

Containers (Docker)

MySQL, phpMyAdmin e WordPress usando Docker

Rodando o **dockerfile** com as configurações

`docker compose -f dockerfile.yml up -d`

```
qriarlabs@instance-8:~/docker$ docker compose -f dockerfile.yml up
[+] Running 49/21
.: db 11 layers [██████████] 52.63MB/56.29MB Pulling 12.4s
.: phpmyadmin 18 layers [██████████] 0B/0B Pulling 12.4s
.: wordpress 21 layers [██████████] 0B/0B Pulling 12.4s
```



Comandos Básicos do Docker

1. Baixar uma Imagem Docker (Pull)

`docker pull nome_da_imagem`

Exemplo: **`docker pull hello-world`**

2. Listar Imagens Docker Locais

`docker images`

3. Rodar um Container Docker (Run)

`docker run nome_da_imagem`

Exemplo: **`docker run hello-world`**

4. Listar Containers em Execução

`docker ps`

5. Parar um Container em Execução

`docker stop id_ou_nome_do_container`

6. Construir uma Imagem a partir de um Dockerfile

`docker build -t nome_da_imagem .`





Comando	Descrição
ls	Lista arquivos e diretórios
ls -a	Lista todos os arquivos, incluindo ocultos
clear ou Ctrl+L	Limpa o terminal
mkdir	Cria diretórios. Ex: mkdir “NomeDoDiretório”
cd	Muda o diretório atual. Ex: cd “DestinoDoDiretório”
uso do TAB para completar	Autocompleta comandos e nomes de arquivos
pwd	Mostra o diretório atual
whoami	Mostra o nome do usuário atual
whoami >> turma2024.txt	Redireciona a saída para um arquivo
mkdir pasta1 pasta2	Cria múltiplos diretórios
mkdir "Nome composto"	Cria diretório com nome composto
Contrabarra (\) para entrar no diretório composto	Entra em diretório com nome composto usando barra invertida. Ex: cd “Nome\ composto”
cd com aspas ex: cd “Nome composto”	Entra em diretório com nome composto usando aspas
touch “nomedoarquivo”	Cria um arquivo vazio
nano “nomedoarquivo”	Edita um arquivo com o editor Nano
cat	Exibe o conteúdo de arquivos no terminal
mv para renomear	Renomeia ou move arquivos/diretórios
cp	Copia um arquivo para um diretório. Ex: cp “nomedoarquivo” ~/pasta1
find . -name nomedoarquivo	Procura arquivos pelo nome (O ponto (.), significa diretório atual



Comando	Descrição
head nomedoarquivo	Mostra as primeiras linhas de um arquivo
tail nomedoarquivo	Mostra as últimas linhas de um arquivo
cd /var/log	Muda para o diretório de logs
head kern.log	Mostra as primeiras linhas do arquivo de log do kernel
tail kern.log	Mostra as últimas linhas do arquivo de log do kernel
rm	Remover arquivo
rmdir	Remove diretórios não vazios
*rm -r (cuidado!)	Remove diretórios (nuca executar rm -rf /)
hostname	Mostra o nome do host
hostname -i	Mostra o ip do host
ls -la	Mostra o usuário e grupo a que pertence os arquivos
ping	Tempo de resposta
sudo add user NovoUsuio	Cria um novo usuário
cut -d: -f1 /etc/passwd	Lista usuários
ps -fp 10902	Identificar o Processo Ativo
kill 10902	Mata o processo
ps -u novo_usuario	Verificar Outros Processos do Usuário
pgrep -u novo_usuario	Verificar Outros Processos do Usuário
kill 10902	Encerrar o Processo