**COMO COPIAR IMAGENS DO DOCKER DE UM HOST PARA OUTRO HOST**

PASSO 1 - Execute o comando abaixo para listarmos todas as imagens existentes atualmente no host que desejamos realizar a cópia da imagem para outro host, para que possamos identificar a imagem e então iniciar o processo de cópia nos próximos passos.

COMANDO: docker images

EXEMPLO:

Texto alternativo gerado por máquina:
rooteteandrocgst : /home/leandro# 
REPOSITOR*' 
ubun tu 
latest 
latest 
centos 
fedor a 
fedora 
r awh ide 
fedora 
fedor a 
latest 
fedor a 
fedora 
fedor a 
branched 
fedora 
fedor a 
fedora 
25-modutar 
modula r 
fedora 
fedor a 
fedora 
hetsenbug 
fedora 
fedor a 
rootmeandrocgst : /home/leandrog 
docker 
1 
vmages 
cfef3ca922eo 
ef3e07ce138f 
e13631cee1a8 
e13e31cee1a8 
98c51911eeac 
9eff96f4c827 
9eff96f4c827 
d093e2f77cfc 
f35d549d2474 
f89698S3S456 
4c5b21baa7ao 
30190780b56e 
9cffd21a45e3 
4f8db6S3ea36 
4f8db6S3ea36 
971eofoa8b71 
ba74bddb63ee 
ba74bddb636e 
ba5369d567d1 
ela9fe974dba 
60ba3309bebb 
CREATED 
days ago 
3 weeks ago 
4 weeks ago 
4 weeks ago 
4 ago 
4 weeks ago 
4 weeks ago 
ru)nths ago 
8 months ago 
8 months ago 
9 months ago 
19 months ago 
24 months ago 
24 months ago 
24 ago 
years 
3 years 
years 
years 
3 years 
years 
ago 
ago 
ago 
ago 
ago 
SIZE 
64.2MB 
220HB 
261MB 
201148 
195MB 
248HB 
248HB 
275MB 
232MB 
235MB 
261MB 
249HB 
232MB 
85 . 3MB 
85 . 3MB 
204MB 
291HB 
291MB 
241HB 
189MB 
214HB 

PASSO 2 - Após identificar qual imagem do docker desejamos realizar a cópia, então execute o comando abaixo, para que seja realizado a leitura do container em questão e que seja armazeado as devidas informações do mesmo em um arquivo de texto.

COMANDO: **docker run DIGITE\_O\_NOME\_DA\_IMAGEM\_DO\_CONTAINER /bin/bash -c "echo 'DIGITE\_O\_QUE\_DESEJA\_QUE\_SEJA\_ESCRITO\_NO\_CONTEÚDO\_DO\_ARQUIVO' > /tmp/nice-content"**

EXEMPLO:



PASSO 3 - Concluindo o passo anterior, execute o comando abaixo, para que possamos validar se foi executado a ação determinada no passo anterior no container que foi inserido no comando do passo anterior.

COMANDO: docker ps -a

EXEMPLO:

Texto alternativo gerado por máquina:
rootmeandrocgst : /home/leandrog 
/btn/bash -c "echo 
rootmeandracgst : /home/ leandro# 
CONTAINER ID 
rooteleandrocgst : /home/leamfro# 
CONTO TNFR 
57a34b349113 
7c685de398ae 
9c624cbdbc6S 
fa59S4aafe2a 
rootmeandrocgst : /home/leandro# 
ubuntu 
centos 
ubuntu 
docker 
docker 
docker 
1 
run 
ps 
ps • 
ubuntu 
Nice 
content' > Itmp/ntce-content" 
STATUS 
NANES 
" Ibtn/bash -c 
" ,'btn/bash" 
" Ibtn/bash" 
•echo 4' 
IS seconds ago 
4s hours ago 
2 days ago 
2 days ago 
ted 
Extted 
Ext ted 
(6) 14 seconds ago 
(127) 4s hours ago 
(e) 47 ago 
(e) 2 days ago 
tender mirzakhant 
a g ta ted_y a Ow 
gal lant_ardtngheUt 
gtfted_etnstetn 

PASSO 4 - Execute o comando abaixo para criarmos uma nova imagem do docker a partir do conteúdo do arquivo que foi gerado no PASSO 2, onde devemos inserir o HASH ID da imagem que foi executado o comando do passo 2, conforme destacado no exemplo do passo 3.

COMANDO: **docker commit DIGITE\_O\_HASH\_ID\_DA\_IMAGEM DIGITE\_UM\_NOME\_QUE\_DESEJA\_DEFINIR\_PARA\_A\_NOVA\_IMAGEM**

EXEMPLO:

IDENTIFICANDO O HASH\_ID DA IMAGEM

Texto alternativo gerado por máquina:
rootmeandrocgst : /home/leandrog 
/btn/bash -c "echo 
rootmeandracgst : /home/ leandro# 
CONTAINER ID 
rooteleandrocgst : /home/leamfro# 
CONTO TNFR 
57a34b349113 
7c685de398ae 
9c624cbdbc6S 
fa59S4aafe2a 
rootmeandrocgst : /home/leandro# 
ubuntu 
centos 
ubuntu 
docker 
docker 
docker 
1 
run 
ps 
ps • 
ubuntu 
Nice 
content' > Itmp/ntce-content" 
STATUS 
NANES 
" Ibtn/bash -c 
" ,'btn/bash" 
" Ibtn/bash" 
•echo 4' 
IS seconds ago 
4s hours ago 
2 days ago 
2 days ago 
ted 
Extted 
Ext ted 
(6) 14 seconds ago 
(127) 4s hours ago 
(e) 47 ago 
(e) 2 days ago 
tender mirzakhant 
a g ta ted_y a Ow 
gal lant_ardtngheUt 
gtfted_etnstetn 

EXECUTANDO O COMANDO APÓS IDENTIFICAR O HASH\_ID DA IMAGEM QUE DESEJAMOS "CLONAR"



PASSO 5 - Execute o comando abaixo para validarmos a criação da nova imagem, conforme definições executadas no passo 4.

COMANDO: docker images

EXEMPLO:

Texto alternativo gerado por máquina:
rooteteandrocgst : /home/ leandr 
REPOS T v,' 
foo•bar 
latest 
uoun 
centos 
latest 
fedora 
fedora 
fedora 
fedo ra 
latest 
fedora 
fedor a 
fedora 
fedora 
fedo ra 
br anched 
fedara 
fedor a 
26 - r 
fedora 
mc%fular 
fedora 
fedora 
er 
umages 
b93ec21 f10dc 
7 seconds ago 
64.2MB 
orscayzz 
6f3eB7cB138f 
e13e31cee1a8 
e13e31cee1a8 
98c51911eeac 
9eff96f4c827 
9eff96f4cB27 
da9332f77cfc 
f35d549d2474 
f89698SB5456 
4c5b21baa 7ao 
3019078Bb56e 
9cffd21a4Se3 
4f8db6S3ea36 
4fBdb6S3ea36 
971eofoa8b71 
ba74bddb636e 
3 weeks ago 
4 weeks ago 
4 ago 
4 weeks ago 
4 weeks ago 
4 weeks ago 
7 months ago 
8 ago 
8 months ago 
9 months ago 
19 months ago 
24 months ago 
24 ago 
24 months ago 
2 years ago 
3 years ago 
22BMB 
201148 
195HB 
248148 
248MB 
275MB 
232MB 
235HB 
261148 
249MB 
232148 
as . 3MB 
. 3MB 
204HB 
291MB 

PASSO 6 - Execute o comando abaixo para validarmos quais processos foram executados pela determinada imagem que criamos com base no arquivo gerado no passo 2.

COMANDO: **docker history DIGITE\_O\_NOME\_DA\_IMAGEM\_DESEJADA**

EXEMPLO:

Texto alternativo gerado por máquina:
CREATEO 
b93ec21f10dc 
31 seconds 
7 days ago 
cf3f3ca922eO 
7 days ago 
«mvsnng> 
7 days ago 
cmtsstnq. 
7 days ago 
7 days ago 
emtsstngs 
rootmeandrocgst : /home/leandrog 
docker 
ago 
1 
hi Story f 00-bar 
EATEO 
/btn/bash 
-c echo 'Nice content' 
/tmp/ntce— 
/btnJsh 
c CHO - /btn,'bash•) 
/btn/sh 
c nkdtr -p /run/systemd echo 'do— 
Ibtn/sh -c [ -z tndextargets)" 
/btn/sh - 
ADD ft1e:d13b09e8b3cc98bfe- 
SIZE 
78 
7453 
987kB 
63.2hE 
COMMENT 

PASSO 7 - Execute o comando abaixo para compactarmos todo o conteúdo da imagem que realizamos a criação dentro de um arquivo, onde este arquivo será utilizado no outro host que desejamos subir esta imagem.

COMANDO: **docker save -o /DIGITE\_O\_CAMINHO\_DO\_DIRETÓRIO\_ABSOLUTO\_QUE\_DESEJA\_SALVAR\_O\_ARQUIVO\_DA\_IMAGEM\_COMPACTADA/DEFINA\_UM\_NOME\_PARA\_O\_ARQUIVO.tar DIGITE\_O\_NOME\_DA\_IMAGEM\_QUE\_DESEJAMOS\_COMPACTAR\_O\_CONTEÚDO\_PARA\_UM\_ARQUIVO**

EXEMPLO:

Texto alternativo gerado por máquina:
rootmeandrocgst:/home/leandr docker save -o /mnt/shared/foo-bar . tar foo-bar 
rootmeandrocgst : /home/leandrog 

PASSO 8 - Realize a cópia do arquivo compactado criado no passo anterior, copiando o mesmo para o host destino no qual desejamos subir esta imagem.

COMANDO: **scp -p DIGITE\_O\_CAMINHO\_ABSOLUTO\_COM\_O\_NOME\_ARQUIVO\_A\_SER\_ENVIADO DIGITE\_O\_NOME\_DO\_USUARIO\_COM\_PERMISSAO\_ADMIN\_NO\_HOST\_DESTINO@DIGITE\_O\_HOSTNAME\_OU\_IP-ADDRESS\_DO\_HOST\_DESTINO:/DIGITE\_O\_CAMINHO\_ABSOLUTO\_QUE\_RECEBERA\_O\_ARQUIVO\_NO\_HOST\_DESTINO/**

EXEMPLO:

Texto alternativo gerado por máquina:
SCP -p 
compactado/ 
vinicius . nunes@192.168.0.200is password: 
fedora-prod-24-Q9-2Q23.tar 
/mnt/ imagens -docker/compactado/fedora-prod-24-09-2023. tar vintcuus.nunes@192.168. O. 200 :/mnt/ imagens -docker/ 
166% 188MB 343 .9MB/s 

PASSO 9 - Após realizar a cópia do arquivo .tar para o host destino via SCP, no host destino, execute o comando abaixo para descompactarmos o arquivo .tar que contém o conteúdo da imagem do docker compactada.

COMANDO: **tar tf /DIGITE\_O\_NOME\_DO\_DIRETÓRIO\_QUE\_O\_ARQUIVO.tar\_ESTA\_ALOCADO**

EXEMPLO:

Texto alternativo gerado por máquina:
I teandrocgsil 
,'mnt 
a2aa9b58d434242762b2fcc6486aoa86e735 69132 a43 6e17 6572c 91 cd/ 
a2aa9b58d434242762b2fcc648õaoa86e735d69132fa43b6e17f6572cfb91 
02aa9b58d434242762b2f j son 
ayer . tar 
07298a lea5 f3e9c84489d3 f e89b f f cec/VERS10N 
07298a 92d4e4 lea5 f3e9c8448gd3 fe89b f3cec8df f354a53354ca3612b36e / j son 
072B8a92d4e4 lea5f3e9c8448Yd3 tee9b ec8df 054053354ca3612b36e ayer . tar 
3ca5561219a47ce8ac0b0476de3ttb6ed75897c2te0860b4ec905058b6e4f1ea/ 
3ca5561219a47 f leal son 
3 ca ce8acoba476de3f fb6ed75897c2f e88eab4ec9B5058b6e4f lea/l aver . tar 
f8U5bcfa9c7bd4f0J3d4db75aob3e539bacbff44nabbw 
3e1a95d173b89bet7tt885bcta9c7bd4t073d4db75a0b3eS39bacbtt4471abbg/VERS10N 
son 
3e1a95d173bB9bef7ff885bcfa9c7bd4 f073d4db75aob3e539bacbff4471abbg/1 ayer tar 
7e344 efdeocbd71b97cb1798b282f35a99a9bdea178dac3gf51bb/ 
7 efdeocbd7 282t35ag9a9bde0178doc3gf51bb/VERS10N 
7 e3441b426998a7tetdeocbd71b97cb17g8b202t35ag9a9bd60178doc3gt51bb/ j son 
ayer . tar 
b93ec21 f 18dc39a74a5f8ca ff35181 f 6c488986baaa f51 f3827 854da f471b3fe. j son 
manifest. json 
repositories 
[root@localhost teandrocgsil* 

PASSO 10 - Identificando as 4 camadas que são responsáveis pela imagem em questão que acabamos de descompactar no passo anterior.

**LAYER 0 (IMAGEM 0):** Nesta camada está o BASE IMAGE (SISTEMA OPERACIONAL\S.O), sendo a imagem de base, sendo responsável por armazenar o sistema de arquivos raiz (rootfs) também conhecido como root filesystem, e todos os diretórios e arquivos necessários para funcionamento dos containers são reduzidos ao mínimo, deixando a imagem de forma MINIMALISTA.

EXEMPLO:

Texto alternativo gerado por máquina:
[ root@localhost teandrocgsil# Is -Ih 'mnt/Shared/foo-bar. tar 
- rwxrwxrux. I root root 64" Oct 25 20:12 /mnt/Shared/too-bar.tar 
2aa9b58d434242762b2fcc6486aBa86e735d69132fa43b6e17f6572cfb91bcd/ 
2aa9b58d434242762b2f bcd/VERS10N 
2aa9b58d434242762b2f son 
2a39b58d434242762b2tccb486aoa86e735d6g132ta43b6e1Vt65J2ctb91bcd/1a er . tar 
e cec 
07298a f3e9c84489d3 f e89b /VERSION 
07298a 92d4e4 lea5 f3e9c8448gd3 fe89b f3cec8df f354a53354ca3612b36e / j son 
07298a 92d4e41ea5 f3e9c8448sd3 teE9b t3cec8df ayer . tar 
3ca5561219a47 
3 ca ce8acoba476de3f fb6ed75897c2f sor, 
3 ca5561219a47 ce8acoba476de3f fb6ed75897c2f e88eab4ec9B5058b6e4f lea,'layer . tar 
f8e5bcfa9c7bd4f073d4db75aob3eS39bacbff4471abbg' 
3 ela95d173b39b9f7ff885bcfa9c7bd4 
3e1a95d173b89b8f7f f885bcfa9c7bd4fB73d4db75aab3e539bacbff4471abb9/1 ayer . tar 
7 Ib42eg98a7f efdeacbd7 Ib97cb1798b282f35a99a9bdea178dac39f51bb/ 
7 efdeocbd71b97cb17B8b262t35ag9a9bdG0178d0c3gf51bb/VERS10N 
7 tar 
b93ec21f ff35181 f 6c488986baB8 f51 f3827 f471b3fe. son 
manifest. json 
repositories 
1 
[ root@localhost teandrocgsil& 

**LAYER 0 (IMAGEM 0):** Nesta camada está o BASE IMAGE (SISTEMA OPERACIONAL\S.O), sendo a imagem de base, sendo responsável por armazenar o sistema de arquivos raiz (rootfs) também conhecido como root filesystem, e todos os diretórios e arquivos necessários para funcionamento dos containers são reduzidos ao mínimo, deixando a imagem de forma MINIMALISTA.

Texto alternativo gerado por máquina:
teandrocgsil# tar -t f tar 
a2aa9b58d434242762b2 f f6572cfb91 bcd/ 
a2aa9b58d434242762b2f f6572cfb91 bcd/VERS10N 
02aa9b58d434242762b2f son 
7298a 92d4e41ea5f3e9c84489d3fe89bf3cec8dff354053354ca3612b36ecec/ 
7298a f3e9c84489d3 f e89b f354a53354ca3612b36ecec /VERSION 
7298a f3e9c8448gd3 fe89b f3cec8df f354a53354ca3612b36e son 
12B8a92d4e41ea5 . tar 
a ce ac 
3 ca ce8acoba476de3f fb6ed75897c2f sor, 
3 ca5561219a47 ce8acoba476de3f fb6ed75897c2f e88eab4ec9B5058b6e4f lea,'layer . tar 
f8e5bcfa9c7bd4f073d4db75aob3eS39bacbff4471abbg' 
3 ela95d173b39b9f7ff885bcfa9c7bd4 
3e1a95d173b89b8f7f f885bcfa9c7bd4fB73d4db75aab3e539bacbff4471abb9/1 ayer . tar 
7 Ib42eg98a7f efdeacbd7 Ib97cb1798b282f35a99a9bdea178dac39f51bb/ 
7 efdeocbd71b97cb17B8b262t35ag9a9bdG0178d0c3gf51bb/VERS10N 
7 tar 
b93ec21f ff35181 f 6c488986baB8 f51 f3827 f471b3fe. son 
manifest. json 
repositories 
1 
troot@localhost teandrocgsil& 

**LAYER 2 (IMAGEM 2):** Nesta camada é onde aplicamos as atualizações de patch\packges e posteriores configurações.

Texto alternativo gerado por máquina:
teandrocgsil# tar -t f tar 
a2aa9b58d434242762b2 f f6572cfb91 bcd/ 
a2aa9b58d434242762b2f f6572cfb91 bcd/VERS10N 
02aa9b58d434242762b2f son 
02aa9b58d434242762b2f cc6486aoa86e bcd/layer . tar 
07298a lea5 f3e9c84489d3 fe89b f /VERSION 
07298a f3e9c8448gd3 fe89b f3cec8df son 
012Y8a92d4e41ea5t3e9c8448sd3tee9bt3cecUdft354053J34ca3612b36ecec 1a er .tar 
ca 5561219a4Vce8ac0b047bde3ttb6ed75897c2te0860b4ec90S058b6e4t1eaf 
ca 5561219a47 ce8acoba476de3f fb6ed75897c2f sor, 
ca5561219a47ceeacob0476de3ffb6ed75897c2fe8Beab4ec905058b6e4f1ea/1a er . tar 
3±1395 89 c a9c7 4 4 ac 
447 La g,' 
labbg/VERS10N 
3 ela95d173b39b9f7ff885bcfa9c7bd4 
3e1a95d173b89b8f7f f885bcfa9c7bd4fB73d4db75aab3e539bacbff4471abb9/1 ayer . tar 
7 Ib42eg98a7f efdeacbd7 Ib97cb1798b282f35a99a9bdea178dac39f51bb/ 
7 efdeocbd71b97cb17B8b262t35ag9a9bdG0178d0c3gf51bb/VERS10N 
7 tar 
b93ec21f ff35181 f 6c488986baB8 f51 f3827 f471b3fe. son 
manifest. json 
repositories 
1 
[ root@localhost teandrocgsil& 

**LAYER 3 (IMAGEM 3):** Esta é a camada de UNION MOUNTS, onde o docker lida com o UNION MOUNTS da seguinte maneira: O docker monta como somente leitura todas as 3 camadas padrões que formam uma imagem, e então, cria uma 4º camada acima das demais, onde está 4º camada possui permissão de leitura e gravação (RW (READ e WRITE)), e quando um container é iniciado, as questões de permissões na sessão do shell do container em qestão rodam sobre esta 4º camada.

Texto alternativo gerado por máquina:
teandrocgsil# tar -t f tar 
a2aa9b58d434242762b2 f bcd/ 
a2aa9b58d434242762b2f f6572cfb91 bcd/VERS10N 
02aa9b58d434242762b2f son 
02aa9b58d434242762b2f cc6486aoa86e ayer . tar 
07298a lea5 f3e9c84489d3 f e89b f /VERSION 
07298a 92d4e4 lea5 f3e9c8448gd3 fe89b f3cec8df f354a53354ca3612b36e son 
t3e9c84489d3 tee9b t3cecUdf ayer . tar 
3 ca ce8acoba476de3f fb6ed75897c2 f f leal son 
3ca5561219a47ceeacob0476de3ffb6ed75897c2feQBeab4ec9B5058b6e4f1ea/1a er . tar 
e1395 89 c a9c7 4 4 75ao 32539 ac 
4471a g/ 
fa73d4db75a0b3e539bacbff4471abb9/json 
ela95d173bB9b8f7ff885bcfa9c7bd4fa73d4db75aab3e539bacbff4471abbg 1a er.tar 
7e3441b42@998a7fefdeocbd71b97cb17g8b2Q2f35ag9a9bdea17Bdac3gf51bb/ 
7 efdeocbdJ Ibsrcb17B8b202t35ag9a9bd80178doc3gf51bb/vERS10N 
ayer tar 
b93ec21f 18dc39a74a5f8ca ff35181f f 51 f3827 854da f471b3fe. j son 
manifest. json 
repositories 
1 
[ root@localhost teandrocgsil& 

Texto alternativo gerado por máquina:
I teandrocgsil# tar -t f tar 
a2aa9b58d434242762b2 f bcd/ 
a2aa9b58d434242762b2f f6572cfb91 bcd/VERS10N 
02aa9b58d434242762b2f son 
02aa9b58d434242762b2f ayer . tar 
07298a f3e9cB4489d3 fe89b f3cec8df /VERSION 
07298a lea5 f3e9c8448gd3 fe89b f3cec8df 53354ca3612b36e son 
f3e9c84489d3 tee9b t3cecUdf ayer . tar 
3 ca 5561219a47ce8ac0b0476de3ttb6ed75897c2te08e0b4ec90S058b6e4t1ea/ 
3 ca ce8acoba476de3f f f leal son 
3 ca fb6ed75897c2f lea,'layer . tar 
f8E5bcta9crbd4f0J3d4db75aob3e539bacbtf4471abbY' 
3e1a95d173b89bet7tt885bcta9c 
j son 
3e1a95d173b89b8f7f ayer . tar 
7 efdeacbd71 b97cb1798b282f35a99a9bdea178dac3gf51bb/ 
7 efdeocbdJ Ibgrcb17B8b2e2t35ag9a9bd80178doc3gt51bb/vERS10N 
7e3441b42e998a7fefdeocbd71b97cb17g8b2e2f35ag9a9bd90178doc3gf51bb 1a er.tar 
93ec21 19 c3ga74a5 8ca 35181 aoa 51 3827854 a 471 3 c. son 
repositories 

PASSO 11 - Execute o comando abaixo, para que o docker possa ler o conteúdo do arquivo compactado e então, importar as informações para o diretório das imagens do docker de forma automática.

COMANDO: **docker load -i /DIGITE\_O\_NOME\_DO\_DIRETÓRIO\_QUE\_O\_ARQUIVO.tar\_ESTA\_ALOCADO**

EXEMPLO:

Texto alternativo gerado por máquina:
[rootesv-SRV 
-LABOI 
-LABOI docker load 
[rootesv-SRV 
Loading Iayer 
4723949838ee : 
Loaded image: 
fedora-prod:latest 
[rootesv-SRV 
-LABOI 
-i /mnt/imagens-docker/compactado/fedora-prod-24-99-2923.tar 

PASSO 12 - Execute o comando abaixo para validarmos se a imagem do docker que está compactada foi iniciada como imagem de container.

COMANDO: docker images

EXEMPLO:

Texto alternativo gerado por máquina:
[rootesv-SRV 
[rootesv-SRV 
REPOSITORY 
fedora-pro 
Teaora 
fedora 
fedora 
fedora 
fedora 
fedora 
fedora 
fedora 
-LABOI 
-LABOI 
TAG 
atest 
r awh ide 
latest 
docker umages 
ed74795a3299 
CREATED 
45 minutes ago 
SIZE 
191MB 
18 
94381T11a 
ays ago 
2d94381f11a3 
2f4dc7b1e2a6 
62131eb5b7d8 
62131eb5b7d8 
adf585e2308a 
Id81a0524f7d 
fddade41cfbe 
le days ago 
16 days ago 
16 days ago 
16 days ago 
16 days ago 
4 months ago 
9 months ago 
186MB 
186MB 
191MB 
191MB 
185MB 
165MB 
155MB 

PASSO 13 - Execute o comando abaixo para acessarmos o shell do container que realizamos a subida, com base no arquivo compactado.

COMANDO: docker run -it NOME\_DA\_IMAGEM\_DO\_CONTAINER /bin/bash

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 docker run -it fedora-prod /bin/bash 
Croot@14148e45df13 
Croot@14148e45df13 • 

PASSO 14 - Basta começar a utilizar o container em questão, conforme a devida necessidade.