**ARMAZENAMENTO**

**IMAGENS: ls -lhs /var/lib/docker/NOME\_DO\_STORAGE\_DRIVE\_EM\_QUESTÃO/**

Texto alternativo gerado por máquina:
r ootmeandruxgst 'home 'teandr 
total 116 
dr.x------ 
Ivar t b,'docker/oVerIay2,' 
527417 
56S7E2 
152152 
ss7se5 
sgi0E4 
548957 
SAAOS4 
5409as 
565976 
573914 
Tez339 
52140 
dr•x••• 
dr•x--•--- 
dr•x•--- 
drõ------ 
dr•a--- 
drvx••-- • 
dr«x-- 
dr•x- 
14 
3 coat 
root 
3 r Oot 
3 r OOt 
4 coot 
r oot 
4 roOt 
4 reot 
root 
4 ronr 
root 
4 r oot 
cont 
root 
root 
foot 
3 r oot 
4 r oot 
2 root 
roo t 
roo t 
roo t 
roo t 
4095 
40.9 s 
4096 
4095 
4090 
4095 
4095 
408 b 
1 
out 
out 
out 
out 
out 
out 
00 r 
out 
out 
Out 
out 
ena t 
Out 
00 t 
out 
out 
00 t 
24 
24 
22 
22 
24 
23 
28 
24 
22 
22 
22 
24 
24 
24 
24 
19 SI 
00:02 
ea:oz 
19:59 
00 ; 05 
zl•.qo 
08:27 
23:59 
oe:as 
21:40 
59 
21:40 
og;os 
ee:02 
95 
ee;03 
ae:93 
00:04 
22:32 
60:oS 
"deu t 
roo ; I 

**CONTAINERS:** /var/lib/docker/containers/

Texto alternativo gerado por máquina:
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
[root@SV-SRV-LABe1 
Crootesv-SRv-LABe1 
Is -Ihs /var/l ib/docker/containers/ 
total 144K 
26 : 
20:39 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:30 
20:39 
20:30 
20:30 
20:30 
20:30 
20:30 
20 : 30 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
d rwx 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
-x- 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
4 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
. OK 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
25 
QIaQ3af257c7723dba8ea74b69efcf36bQ37a7dQQ8d63d139eQ2e8Q86c76ba49 
14148045df 13ffQ44agdQQaQdb5Qf94b833QQb4116daade783Qe62c88397beae 
226b66Qbe3QQ2886fc82a2261cbQ6b8d2a176e3QaQd53897gaQb77ee2acfd3e5 
261a3cgacfedbc822234e712bbd94ce4b8852c86d581d1Qb5e6d8cabQ6b8b7a5 
4022995f57d44f33fQ4853fQe2QaQb776753f5beggb86f8aaed93c3861cQc94e 
42edb16bd85baQ77a1a3beedcbf8e82abb417b41a85c93d4bcba912ad42ee387 
43159d35af7645293cd4cb2Qfa7b98c 72e5e52bbcfQ7Qdd4Q3aQed1e88e56147 
476ddgb3bb6b879Q36a21eQc81a4baf483239955fbcffa632fQ8c4d12d4b8588 
488c 111 lei f82365a478218a86gb8eaf443b2cQ56gd94af4dc49f 
4beQ14Q8dgf8864d66c67c7fgd26dfd5ggab728de31db454cb563767765QQ7b3 
55bQ998ac6caa35b972ca2Qdee48c 7cdbeQQ55b161b3 
56b6f66a84cb3cggf9427af4ac1Q8fe16e23123af92c5726b4eQ487d4ed86992 
60a 719ed7Qge6f3d1bff9311 If7eac b34 fd27bd1 Ic bdQa24 faea3ba6abga Ib41 
6155ca86721f5cb1c3e1248ba92Q41fd6dddb4b896dbff4d386aQQ2735c6cb98 
8ae91cef15a6Q2c971db9933883c 7961a36373a7Qd2cd77Q1ae3eQd519354c46 
8e6bdf8898a8abgdb419542824f4237dQd4635ge2791gd7e6cc23a26499d932e 
9684caagff5d85ca7237c4c2fccgbfeQ93db26Qc3gededegbd76b32774a17Qe3 
7522673b1a224ac85Q6138b76fe8fead5b 
a825da3c3foggggcQQb14576c98Q2527b4e95Q36gcaae4421Q46f9Q696497b6f 
aae4c4bffc6c57eb44gff6degea3d82c8232257877gc4f66eeaga616aQ5d2dQQ 
b28a7326Q4f323a8292geecdc4312683f461Q2c49283Q85cbe84daaa4Q2f5dcQ 
b646fdQbda5Q8bggf586e62Qfa31feQece638fda3142ec 165e94Q538aa41e24b 
b6f8d1ea8425e451cc Icb2664QeQd3621gc5aef3a7294442aQ7dd57a3452b22f 
bf5Q95913Q6d75cd172418a2c6Q5db2cga3c8Q85b6511482cQ81fdf7ed16dgc4 
ce8Q25c65ge2bQe33277Q4gcec616Q35a16b6ea764b413452e1gcaf6b65eeaaf 
dgbcaa3fb8cd6f332ee3626ggga82egb66dQ86Q455fQQf21ccff1Q68ce84f56b 
dadc684de771c 7f3fda55e2baeac6d6f9835b5233266617bQ6eQcb74c5f852ad 
dd5aQ2cagdgc4dbf2657Qabgb33dfdad37367gbQQ79958c f456a6a9Qe866cdf8 
ddb2efe58cf9996c82f8d7efa518adc 13b5526f2596fbfb198396eQ5447b5d8a 
e4a7fbggd373b5992652c3d27a5425e8Q4fQ2995f287c73f4Q3ea1Qddbb74Qe8 

**ARQUIVO .JSON DE UM CONTAINER**

PASSO 1 - Execute o comando abaixo para localizar o HASH\_ID do container que deseja coletar a informação da configuração json do determinado container.

COMANDO: docker ps -a

Texto alternativo gerado por máquina:
[rootesv-SRv-LABe1 
CONTAINER ID 
226b66ebe3ee 
Séc1ceeee95 
IMAGE 
deb ian 
ubuntu. 
docker 
• 19.64 
ps -a 
" /b in/bash 
"/bin/bash 
"/bin/bash" 
-c 
-c 
apt 
yum 
um" 
CREATED 
23 hours 
23 hours 
23 hours 
ago 
ago 
ago 
STATUS 
Exited 
Exited 
Exited 
PORTS 
( 127) 23 hours ago 
(6) 23 hours ago 
(6) 23 hours ago 
NAMES 
kind hawking 
sleepy_noether 
jov ial_brahmagupta 

PASSO 2 - Copie o HASH\_ID do container e execute o comando abaixo, para localizar qual é o arquivo responsável pelo conteiner em questão.

COMANDO:  **ls -lhs /var/lib/docker/containers/ | grep COLE\_OU\_DIGITE\_O\_HASH\_ID\_DO\_CONTAINER**

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
Crootesv-SRv-LABe1 
4. OK drwx--x 
4 root 
Crootesv-SRv-LABe1 
Is -Ihs var I ib/docker containers 
re 226b66ebe3ee 
.0K sep 25 20:30 226 660 e3002886 c82a2261cb06b8 2a176e3eaed538979aeb77ee2acfd3e5 
root 

PASSO 3 - Execute o comando abaixo, para que possamos localizar o arquivo .jason dentro do diretório responsável pelo container que copiamos o HASH\_ID no PASSO 1.

COMANDO: **ls -lhs /var/lib/docker/containers/DIGITE\_O\_NOME\_DO\_ARQUIVO\_APRESENTADO\_NA\_SAÍDA\_STD\_DO\_PASSO\_2/**

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
Crootesv-SRv-LABe1 
total 28K 
IS -lhs /var/lib/docker/containers/226b66ebe3992886fc82a2261cb96b8d2a176e3eaed538979aeb77ee2acfd3e5/ 
4.9K -rw-r- 
e drwx 
4.91< -rw- 
4.9K -rw- 
4.9K -rw-r- 
4.6K -rw-r- 
drwx--x 
4.9K -rw-r--r- 
4. OK -rw-r--r- 
2 
2 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
root 
1.6K 
6 
1.4K 
13 
174 
6 
87 
71 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
Sep 
24 
24 
25 
25 
24 
24 
24 
24 
24 
43 
43 
20. 
• 30 
20 
: 30 
43 
43 
43 
43 
43 
226b66ebe3662886fc82a2261cb 
8d2a 176e3eaed538979aeb77ee2acfd3e5 - j son . log 
check oints 
con .v2. son 
os con ug.json 
hos tname 
hos ts 
mounts 
resolv . conf 
resolv . conf .hash 
Crootesv-SRv-LABe1 
[root@SV-SRV-LABe1 

PASSO 4 - Execute o comando abaixo, para que possamos realizar a leitura do conteúdo do arquivo com extensão .json, onde este arquivo representa as configurações em jason do conteiner em questão.

COMANDO: **cat /var/lib/docker/containers/DIGITE\_O\_NOME\_DO\_ARQUIVO\_APRESENTADO\_NA\_SAÍDA\_STD\_DO\_PASSO\_2/config.v2.json**

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 cat ,'var/ / oc er/contauners/226 660 e3002886 c82a226 c 06 8 2a 76e3eae 538979ae 77ee2ac 3e5/con -cg. v2.json 
{"Streamconfig" "Staféú'"é'k" ¯ 
a se, "Hestartung 
a se, "Hemovallnprogress 
unn ung 
a se, 
a se, 
, "Exitcode" "Error" : " " , "StartedAt 
' : "2023-09-25TOO : 07 .296464743Z" , "FinishedAt" : "2023-eg-25Tee : 23 .955704022& , "Health" : null} , "10" : "226b6 
60be3002886fc82a2261cb06b8d2a176e3eaed538979aeb77ee2acfd3e5 
" , "Created" 
: "2023-eg-25Tee . 16416010521 , "Managed" 
• false, "Path" : "/bin/bash" , "Ar 
":" " , "User" 
' , "IPAddress " : "" , "IPPrefixLen" , "IPv6Gateway" : "" , "GlobalIPv6Address " : " " , "GlobalIPv6PrefixLen" , "MacAddress " : " " , "Driver0pts " : null , "IPAM0perat 
gs " 
name 
C" c" , "yum insta II iputils -y ; ping 
yum insta II btop -y ; hostname'l], "Config" :{"Hostname 
" : "226b66ebe3ee 
8.8.8.8 -c 
" , "Doma un 
' , "Attachstdin" : false, "Attachstdout" : false, "Attachstderr" : false 
false, "Openstdin" : false, "Stdin0nce" : false 
, "Env" : C 'IPATH= 
, "Tty"• 
"Cmd" : C'l/bin/bash"," c" , "yum insta II iputils -y ; ping 
8.8.8.8 -c 
yum insta II btop 
-Y ; hostname'l], "Image 
. " , "Volumes" :nuII, "WorkingDir":"" , "Entrypoint" 
" : "centos • 7 
ma . build-date 
" , "org.label-schema . license" : "GPLv2" , "org.label-schema . name" • "Centos Base Image" , "org.label-schema . schema-vers ton" : "1 
" : "20201113 
O" , "org.label-schema . vendor 
. image.created" 
" , "org.opencontauners . image.licenses": "GPL- 
: "2020-11-13 
" : "Centos 
2.e-onIy" , "org.opencontainers . image .tutle" • "Centos Base Image" , "org.opencontauners . umage . vendor" : "Centos"}} , "Image" 
: "sha256:eeb6ee3f44bdeb5103 
bb561b4c16bcb82328cfe5809ab675bb17ab3a16c517c9 
" , "ImageManifest" : null , "Networksettings " : {'IBridge" : , "SandboxID 
" : "096a04fb9865447cfd82c7185762a 
16eQ5d9f8fa7f953ef686e1be16fb69ffd4" , "Ha irp inMode" : false , "L inkLocaIIPv6Address " : " " , "L inkLocaIIPv6Pref ixLen" : O , "Networks " : {"bridge" : IPAMConfi 
:null 
, "L inks " :nuII, "AI iases " :nuII, "NetworkID 
" : "ca4dd7b957ae492bf4ea1e9b7461aeebc5e917b73ec351f3f6268b88474c45ce 
" , "EndpointID" , "Gateway" 
tonal " : false}} , "Service" : null , "Ports " : null , "SandboxKey" : "/var/run/docker/netns/096ae4fb9865" , "SecondaryIPAddresses " : 
nuli 
, "SecondaryIPv6Address 
: nuli 
, "IsAnonymousEndpoint" : true , "HasswarmEndpoint" : false} , "Logpath 
" : " ,'var/l ib/docker/conta iners ,'226b66ebe3002886fc82a2261cb06b8d2a176e3eae 
d538979aeb77ee2acfd3e5/226b66ebe3002886fc82a2261cb06b8d2a176e3eaed538979aeb77ee2acfd3e5-json . log 
" , "Name" : "/sleepy_noether" , "Driver" : "overIay2" 
, "OS" : "I inux" , "RestartCount" "HasBeenStartedBefore" : true , "HasBeenManuaIIyStopped" 
false , "Mountpoints " , "SecretReferences " : null , "ConfigRefe 
" :nuII , "MountLabeI " : , "ProcessLabeI " : " " , "AppArmorprofiIe'I : " " , "Seccompprofile" • , "NoNewpriviIeges" false, "Hostnamepath" : "/var/l ib/docke 
r ences 
r / conta iners ,'226b66ebe3002886fc82a2261cb06b8d2a176e3eaed538979aeb77ee2acfd3e5/hos tname 
" , "Hostspath" : " /var/l ib/docker/containers/226b66ebe30028 
86fc82a2261cb06b8d2a176e3eaed538979aeb77ee2acfd3e5/hosts " 
, "Shmpath" : " " , "Resolvconfpath" : " /var/l ib/docker/containers/226b66ebe3992886fc82a2261c 
b06b8d2a176e3eaed538979aeb77ee2acfd3e5/ resolv . conf" 
, "LocalLogCacheMeta" : false}} 
[root@SV-SRV-LAB01 

PASSO 5 - Fim, o arquivo apresentado no PASSO 4 é o arquivo .json contendo as configurações .json do determinado conteiner.

**COMANDOS DE DOCKER CLIENT**

**docker -v** (Este comando apresenta no stdout a versão atual do daemon docker que está em execução no momento).

EXEMPLO:

Texto alternativo gerado por máquina:
docker 
Docker ver suon 24.e.5, 
-v 
.64.1 

**docker version** (Este comando além de apresentar no stdout a versão atual do daemon docker que está em execução no momento, também apresenta detalhadamente informações de versões da cadeia de sub-serviços do daemon docker (Server Docker) e client docker).

EXEMPLO:

Texto alternativo gerado por máquina:
root@SV-SRV-ZABBIX: 
CI ient: 
Vers von: 
API vers von: 
Go vers ton: 
Git commit: 
Bui I t: 
OS/Arch 
Context : 
Server: 
Eng ine: 
Vers ton: 
API versuon: 
Go versuon: 
Gut commut: 
Bui I t: 
OS/Arch: 
Exper imental : 
conta unerd: 
Vers ton: 
Gitcommut: 
runc: 
Vers ton: 
Gitcommut : 
docker- inut: 
Vers ton: 
Gitcommut : 
docker vers ton 
24.6.5 
1.43 
gol.2e.3 
24.e.5-eubuntu1-,22 .64.1 
Mon Aug 21 2623 
I inux/amd64 
default 
24.6.5 
1.43 (minimum versuon 1. 12) 
gol.2e.3 
24. e. 5-eubuntu1-,22 .64.1 
Mon Aug 21 2623 
I inux/amd64 
false 
1.7.2 
1. 1.7 -eubuntu1-,22 .64. 1 
e. 19.6 

**docker info** (Este comando apresenta na saída stdout informações sobre o estado atual do ambiente docker, como quantidade de containers em uso, número de images, versões do Go, versões do Daemon Docker (Server Docker) e client Docker, Storage Driver localização do diretório atual do Docker Root e muito mais).

EXEMPLO:

Texto alternativo gerado por máquina:
rootesv-SRv-ZABBIX: 
CI ient: 
24.6.5 
Vers von: 
default 
Context: 
Debug Mode: 
false 
Se rver: 
Conta uners: 
docker info 
Running: 
paused: 
Stopped: 
Images: 
Server Versuon: 24.6.5 
Storage Driver: overIay2 
Backing Filesystem: extfs 
Supports d_type: true 
Us ing metacopy: 
false 
Native Overlay Diff: true 
false 
userxattr: 
Logging Driver: 
j son-file 
Cgroup Driver: systemd 
Cgroup Vers von: 2 
plugins: 
Volume: local 
Network: bridge host ipvlan macvlan null overlay 
Log: awslogs fluentd gcplogs gelf journald json-file local Iogentrues splunk syslog 
Swarm: unactive 
Runtumes: to. . runc .v2 runc 
Default Runtime: runc 
Init Binary: docker-inut 
contavnerd vers ton: 
runc vers ton: 
inut versuon: 
Securuty Optuons : 
apparmor 
seccomp 
profile: 
bu i It in 
cgroupns 

**docker run -it DIGITE\_O\_NOME\_DA\_DISTRO\_PARA\_CONTAINERIZAR /bin/bash** (Este comando irá realizar o download MINIMAL da imagem de container da distribuição desejada, onde irá criar o ID HASH para este determinado containado container, disponibilizando o mesmo para acesso e gerenciamento via shel bash).

COMANDO: docker run -it DIGITE\_O\_NOME\_DA\_DISTRO\_PARA\_CONTAINERIZAR /bin/bash

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
Crootesv-SRv-LABe1 docker 
CONTAINER 
OMAND 
f1555532cfa1 
ubuntu 
" /b in/bash" 
[root@SV-SRV-LAB01 
Croot@SV-SRV-LABe1 • 
Less than a second ago 
Exited (e) 4 seconds ago 
focused colden 

**docker run -it DIGITE\_O\_NOME\_DA\_DISTRO\_PARA\_CONTAINERIZAR:DIGITE\_A\_VERSÃO\_DESEJADA /bin/bash** (Este comando irá realizar o download MINIMAL da imagem de container da distribuição NA VERSÃO DESEJADA, CASO NÃO EXISTA, ONDE IRÁ INICIAR UM CONTAINER NA VERSÃO DESEJADA, CONFORME FORNECIDA NO COMANDO).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
Croot@SV-SRV-LABe1 docker run -it ubuntu: 19.64 /bin/bash 
root@ce8025c659e2 : 

**docker run -d DIGITE\_O\_NOME\_DA\_IMAGEM\_DE\_CONTAINER\_DESEJADO /bin/bash -c "INSIRA O COMANDO QUE VOCE DESEJA QUE SEJA EXECUTADO DENTRO DO CONTAINER APENAS"** (Este comando permite que seja inicado um novo container com base em uma determinada imagem do docker, onde este container irá executar e processar o comando fornecido no comando em questão, e após o comando ser executado com sucesso, então o container em questão será terminado, ou seja, o processo responsável por manter o container em questão em execução será encerrado de forma automática).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 docker run -d fedora-prod /bin/bash 
dgbcaa3fb8cd6f332ee3626889a82eSb66dbÉ6bdS5tbb+hh"h+h068ce 
[root@SV-SRV-LAB01 
um 
ins ta II 
ut i Is 
8.8.8.8 

**docker pull DIGITE\_O\_NOME\_DA\_DISTRO\_PARA\_CONTAINERIZAR** (Este comando irá realizar o download MINIMAL da imagem MAIS RECENTE no repositório DockerHUB do docker).

COMANDO:

EXEMPLO:

Texto alternativo gerado por máquina:
docker ulI fedora 
root@SV-SRV-ZABBIX: 
Us ing default tag: Ia es 
latest: Pulling from library/fedora 
Digest: sha256:6fceef83a1b6526b1c6562e3ef552d169ba8e269 
Status: Image is up to date for fedora:latest 
42a26efab Ib7aef59e 
docker . io/l ibrary/fedora:latest 
ocker ps -a 
CONTAINER 
d3480fc506do 
fedora 
'l 'b in/bash" 
4b517254bced 
hello-world 
" /hello 
9 minutes ago 
h'-i munures ago 
24 hours ago 
Exited 
Exited 
(255) 39 seconds ago 
9 munures ago 
(6) 24 hours ago 
confident morse 
locusen colrlen 
relaxed torvalds 

**docker pull -a DIGITE\_O\_NOME\_DA\_DISTRO\_PARA\_CONTAINERIZAR** (Este comando irá realizar o download MINIMAL DE TODAS AS IMAGENS DESTA DETERMINADA DISTRO NO REPOSITÓRIO DO DockerHUB do docker).

COMANDO:

EXEMPLO:

Texto alternativo gerado por máquina:
docker pull 
5: Pulling from libra 
38892065247a: Extracting L 
-a centos 
52 .36MB/87 .34MB 

**COMBINAÇÃO DE TECLAR CTRL + P + Q** (Este comando realiza a saída do container sem interrompe-lo (para-lo\stop)).

COMBINAÇÃO DE TECLAS: CTRL + P + Q

EXEMPLO:



**usermod -aG docker NOME\_USUÁRIO\_DESEJADO** (Este comando adicionará o usuário desejado ao grupo do docker como grupo secundário, e não removerá os demais grupos secundários que já estavam associados ao usuário em questão, onde devido ao arquivo de socket /var/run/docker.sock possui o usuário root como dono e o grupo DOCKER como grupo do arquivo, ao adicionarmos o usuário no grupo docker, permitiremos que o determinado usuário execute comandos (binários) do docker sem a necessidade de elevar a sua sessão para root (sudo)).

COMANDO: usermod -aG NOME\_USUÁRIO\_DESEJADO docker

EXEMPLO:

Texto alternativo gerado por máquina:
Crootesv 
-SRV-LABOI 
Crootesv 
-SRV-LABOI 
Crootesv 
-SRV-LABOI 
[rootesv 
-SRV-LABOI 
docker:x. 
'981. 
usermod 
-aG docker vintcuus .nunes 
getent rou 
-u oc er 
• vunuc tus . nunes 
Croot@sv-SRv-LABe1 

**docker ps** (Este comando lista e apresenta na saída (stdout) todos os containers em execução no momento).

COMANDO: docker ps

EXEMPLO:

Texto alternativo gerado por máquina:
Crootesv-SRv-LABe1 
Crootesv-SRV 
-LABOI docker 
CONTAINER ID 
IMAGE 
Crootesv-SRv-LABe1 
-bash: /root: 
Is a directory 
Croot@SV-SRV-LAB01 
ps 
CREATED 
STATUS 
PORTS 
NAMES 

**docker ps -a** (Este comando lista e apresenta na saída (stdout) todos os containers em execução no momento e TODOS OS CONATINERS QUE ESTÃO PARADOS OU FORAM CRIADOS E NÃO ESTÃO EM EXECUÇÃO NO MOMENTO).

COMANDO: docker ps -a

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
Crootesv-SRv-LABe1 docker 
CONTAINER 
OMAND 
f1555532cfa1 
ubuntu 
" /b in/bash" 
[root@SV-SRV-LAB01 
Croot@SV-SRV-LABe1 • 
Less than a second ago 
Exited (e) 4 seconds ago 
focused colden 

**docker ps -l** (Este comando irá apresentar na saída (stdout) somente o ultimo container que foi iniciado e\ou interrompido, onde auxilia o administrador durante o troubleshooting no dia a dia).

EXEMPLO:

Texto alternativo gerado por máquina:
Croot@SV-SRV-LABe1 
CONTAINER 0_ IMAGE 
b28a732604f3 
deb ian 
um" 
rd6t@SV-SRV-LABe1 
[root@SV-SRV-LAB01 
docker ps 
"/b in/bash 
-c 
apt 
CREATED 
6 minutes 
ago 
STATUS 
Exited 
( 127) 
6 
minutes 
PORTS 
ago 
NAMES 
k ind hawking 

**docker start DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_DESEJADO** (Este comando realiza a inicialização de um determinado container que estava PARADO, mas não realizará o acesso a sessão do shell do determinado container).

COMANDO: **docker start DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_DESEJADO**

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
root@SV-SRV docker ps - a 
COMAND 
CREATED 
Less than 
Less than 
a secon 
a second 
STATUS 
ago 
PORTS 
f1555532cfa1 
ubuntu 
" /b in/bash" 
"/he110" 
munu 
es 
ago 
NAMES 
focused colden 
relaxed torvalds 
4b517254bce 
el 
[root@SV-SRV-LABe1 
C rootesv-SRv-LABe1 
Crootesv-SRv-LABe1 
f1555532cfa1 
Croot@SV-SRV-LAB01 
[root@SV-SRV-LABe1 
o -world 
docker 
start f1555532cfa 

**docker restart DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_DESEJADO** (Este comando realiza o restart do container).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
028a732604f3 
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
docker restart b28a732694f3 

**docker stop DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_DESEJADO** (Este comando realiza o envio do sinal de interrupção SIGTERM (-15) para o PID responsável pela execução do determinado container, fazendo com que o container seja encerrado mas não de forma abrupta, por conta de utilizar o sinal de interrupção SIGTERM (-15)).

EXEMPLO:

Texto alternativo gerado por máquina:
99c1ceeee95f 
un / b Ás h " 
-LABOI a$ 
aamun . aocKer@SV-SRV 
[admin . docker@SV-SRV 
-LABOI a$ 
[admin.docker@SV-SRV-LABe1 docker s top 
99c1ceeee95f 
[admin . docker@SV-SRV 
-LABOI a$ 
CONTAINER ID 
IMAGE 
COMAND 
CREATED 
[admin . docker@SV-SRV 
-LABOI 
[admin . docker@SV-SRV 
-LABOI a$ 
[admin.docker@SV-SRV-LAB01 a$ 
[admin.docker@SV-SRV-LABe1 a$ oc er s 
[admin.docker@SV-SRV-LABe1 docker ps 
IMAGE 
COMAND 
12 minutes ago 
99c1ceeee95f 
STATUS 
12 minutes 
NAMES 
PORTS 
NAMES 
jovial brahmagupta 
STATUS 
PORTS 

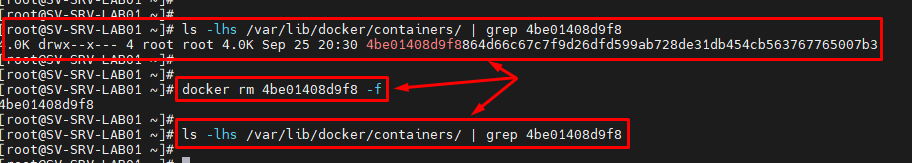
**docker rm DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_DESEJADO** (Este comando realiza a remoção de um container da distro em questão, **DESDE QUE ESTE CONTAINER NÃO ESTEJA EM EXECUÇÃO**, e então, deletará até mesmo o arquivo responsável por este container no diretório /var/lib/docker/containers/).

EXEMPLO:

Texto alternativo gerado por máquina:
trasv- 
-LABOI Is -Ihs /var/lib/docker/contauners/ 
grep 14148945df13 
[rootesv-SRV 
. OK drwx--x- 
4 root root 4.6K sep 25 20:36 14148045df13ff644a9deeaedb5ef94b833eeb4116daade7836e62c88397beae 
root@SV-SRV-LABe1 
[root@SV-SRV-LAB01 
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
C rootesv-SRv-LABe1 
14148045df13 
[rootesv 
Crootesv 
[rootesv 
[rootesv 
[rootesv 
[rootesv 
-SRV-LABOI 
-SRV-LABOI 
-SRV-LABOI 
-SRV-LABOI 
-SRV-LABOI 
-SRV-LABOI 
docker rm 14148945df13 
Is -Ihs /var/l ib/docker/containers/ gr ep 14148945df13 

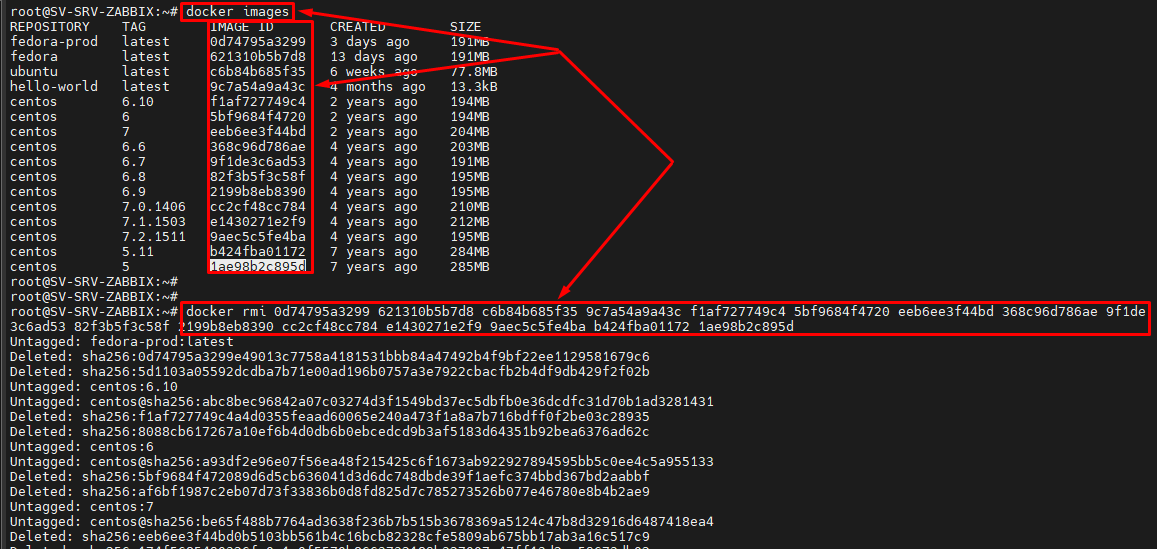
**docker rm DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_DESEJADO -f** (Este comando realiza a remoção de um container da distro em questão, onde a flag -f forçara a remoção do container,  **MESMO QUE O ONTAINER ESTEJA EM EXECUÇÃO**, e então, deletará até mesmo o arquivo responsável por este container no diretório /var/lib/docker/containers/).

EXEMPLO:



**docker rmi DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_DESEJADO** (Este comando realiza a remoção de um container da distro em questão, **DESDE QUE ESTE CONTAINER NÃO ESTEJA EM EXECUÇÃO**, e então, deletará até mesmo o arquivo responsável por este container no diretório /var/lib/docker/containers/).

EXEMPLO:



VALIDANDO SE AS IMAGENS DO DOCKER FORAM REMOVIDAS:

Texto alternativo gerado por máquina:
root@SV-SRV-ZABBIX: 
REPOSITORY 
TAG 
root@SV-SRV-ZABBIX: 
docker 
uma es 
ATED 
SIZE 

**docker rmi -f DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_DESEJADO** (Este comando realiza a remoção de um container da distro em questão, onde a flag -f forçara a remoção da IMAGEM DO DOCKER,  **MESMO QUE O ONTAINER ESTEJA EM EXECUÇÃO**, e então, deletará até mesmo o arquivo responsável por este container no diretório /var/lib/docker/containers/).

EXEMPLO:

Texto alternativo gerado por máquina:
docker 
-f ed74795a3299 62131eb5b7d8 c6b84b685f35 9c7a54a9a43c flaf727749c4 5bf9684f4726 eeb6ee3f44bd 368c96d786ae 9f 
Ide3c6ad53 82f3b5f3c5 f 2199b8eb83ge cc2cf48cc784 e1430271e2f9 gaec5c5fe4ba b424fbae1172 1ae98b2c895d 
Untagged: 
Untagged: 
Deleted: 
Untagged: 
Untagged: 
Deleted: 
Untagged: 
Untagged: 
Deleted: 
fedora:late 
fedora@sha256 : 6fceef83a1b6526b1c6562e3ef552d199ba8e269259c6742a26efab1b7aef59e 
sha256 : 62131 eb5b7d8188469265c9eece793ee72128e5e5ff5cb935a527b73873e4898 
ubuntu:latest 
ubuntu@sha256 : aabed3296a3d45cede Idc866a24476c4d7e993aa896263c27ddaadbdce3c 1654 
sha256 : c6b84b685f35f 1a5d63661f5d4aa662ad9b7ee4f4b8c394c022f25023c907b65 
hello -world:latest 
sha256:9c7a54a9a43cca047013b82af109fe963fde787f63f9e016fdc33845eec2823d 

**docker rmi $(docker images -q)** (Este comando irá remover todas as imagens não vinculadas a containers no docker).

EXEMPLO:

Texto alternativo gerado por máquina:
Croot@SV-SRV-LAB02 
Croot@SV-SRV-LAB92 docker rmi $(docker images 

**docker rmi $(docker images -q) -f** (Este comando irá remover **FORÇADAMENTE** todas as imagens do docker, inclusive as imagens vinculadas a containers no docker).

EXEMPLO:

Texto alternativo gerado por máquina:
Croot@SV-SRV-LAB02 
Croot@SV-SRV-LAB92 docker rmi $(docker images -q) -f 

**docker rmi $(docker ps -a -q)** (Este comando irá remover todos os containers que não estão em execução no docker).

EXEMPLO:

Texto alternativo gerado por máquina:
Croot@SV-SRV-LAB02 
Croot@SV-SRV-LAB92 docker rmi $(docker ps -a 

**docker rmi $(docker ps -a -q) -f** (Este comando irá remover **FORÇADAMENTE** todos os containers do docker, inclusive os containers que estão em execução no docker).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LAB02 
Croot@SV-SRV-LAB02 docker rmi $(docker ps -a -q) -f' 

**docker exec -it DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_A\_SER\_ACESSADO /bin/bash** (Este comando estabelece um acesso a sessão do shell de um determinado container, permitindo que o container seja gerenciado por nós neste determinado momento, **CONTUDO, DIFRENETEMENTE DO COMANDO docker attach, ESTE COMANDO docker exec AO SAIR DO CONTAINER, O PROCESSO DO MESMO NÃO É ENCERRADO NO MESMO INSTANTE, SE MANTENDO EM EXECUÇÃO**).

EXEMPLO:

Texto alternativo gerado por máquina:
Croot@SV-SRV-LABe1 
NTATNFR 
IMAGE 
05ef72a65953 
-LABOI 
roo tes 
[root@SV-SRV-LABe1 
Crootesv-SRv-LABe1 
root@05ef72a65953 : 
docker ps 
"(h*n/bas 
docker exec 
CREATED 
7 seconds ago 
STATUS 
Up 6 seconds 
PORTS 
NAMES 
s illy_engelba rt 
05ef72a65953 /bin/bash 

**nsenter -m -u -n -p -i -t DIGITE\_O\_PID\_DO\_DETERMINADO\_CONTAINER\_A\_SER\_ACESSADO /bin/bash** (Este comando estabelece um acesso a sessão do shell de um determinado container, permitindo que o container seja gerenciado por nós neste determinado momento, **CONTUDO, DIFRENETEMENTE DO COMANDO docker attach, ESTE COMANDO docker exec AO SAIR DO CONTAINER, O PROCESSO DO MESMO NÃO É ENCERRADO NO MESMO INSTANTE, SE MANTENDO EM EXECUÇÃO, EVITANDO QUE UM CONTAINER EXECUÇÃO SEJA REMOVIDO DE FORMA ERRONEA**).

EXEMPLO:

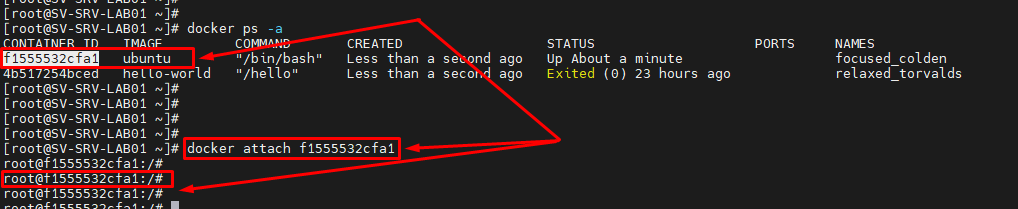
**SEGUE INSTRUÇÕES ABAIXO EM COMO CAPTURAR O PID DO DETERMINADO CONTAINER:**

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
Croot@SV-SRV-LABe1 
r nNTó 
IMAGE 
05ef72a65953 
ian 
-LABOI 
roo tes 
[root@SV-SRV-LABe1 
Croot@SV-SRV-LABe1 
"pid' 
3834 
"PidMo e 
"Pidsl_imit" 
Croot@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
[rootesv-SRV 
docker ps 
I 'b in/bash" 
ocker ins ect 
nuli, 
CREATED 
3 minutes ago 
05ef72a65953 
STATUS 
Up 3 minutes 
PORTS 
NAMES 
s illy_engelba rt 
-LABOI nsenter -m -u 
-n -p -i -t 3834 /bin/bash 

**docker attach DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_A\_SER\_ACESSADO** (Este comando estabelece um acesso a sessão do shell de um determinado container, permitindo que o container seja gerenciado por nós neste determinado momento, **CONTUDO, AO SAIR DO CONTAINER, O PROCESSO DO MESMO SERÁ ENCERRADO NO MESMO INSTANTE**).

COMANDO: docker attach **DIGITE\_O\_HASH-ID\_DO\_CONTAINER\_A\_SER\_ACESSADO**

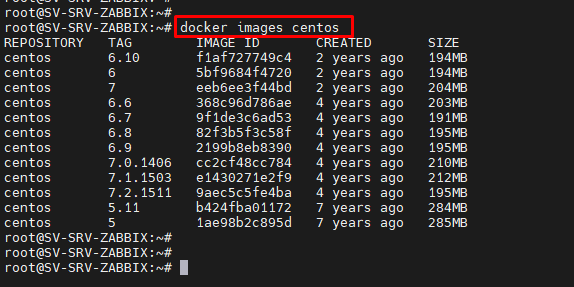
EXEMPLO:



**docker images DIGITE\_O\_NOME\_DA\_DISTRO\_PARA\_VERIFICAR\_A\_QUANTIDADE\_DE\_IMAGENS** (Este comando apresenta na saída (stdout) todas as imagens disponíveis no momento para a determinada DISTRO LINUX digitada).

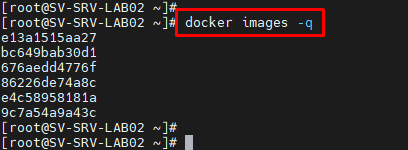
COMANDO: **docker images DIGITE\_O\_NOME\_DA\_DISTRO\_PARA\_VERIFICAR\_A\_QUANTIDADE\_DE\_IMAGENS**

EXEMPLO:



**docker images -q** (Este comando lista todos os ID's de cada imagem do docker).

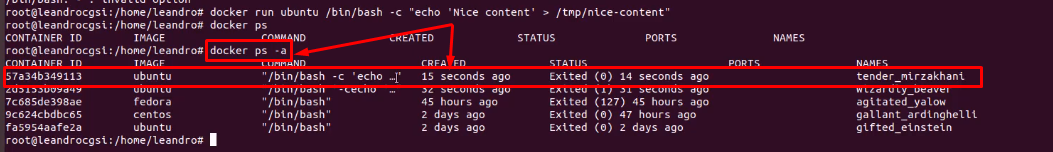
EXEMPLO:



**docker commit DIGITE\_O\_HASH\_ID\_DA\_IMAGEM DIGITE\_UM\_NOME\_QUE\_DESEJA\_DEFINIR\_PARA\_A\_NOVA\_IMAGEM** (Este comando permite criarmos uma nova imagem de container a partir de uma determinada imagem de container existente).

EXEMPLO:

IDENTIFICANDO O HASH\_ID DA IMAGEM



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



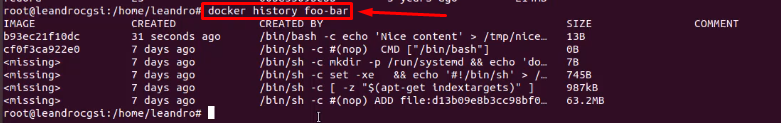
**docker save -o /DIGITE\_O\_CAMINHO\_DO\_DIRETÓRIO\_ABSOLUTO\_QUE\_DESEJA\_SALVAR\_O\_ARQUIVO\_DA\_IMAGEM\_COMPACTADA/DEFINA\_UM\_NOME\_PARA\_O\_ARQUIVO.tar DIGITE\_O\_NOME\_DA\_IMAGEM\_QUE\_DESEJAMOS\_COMPACTAR\_O\_CONTEÚDO\_PARA\_UM\_ARQUIVO** (Este comando irá compactar todo o conteúdo de uma determinada imagem em execução para dentro de um arquivo .tar, onde este arquivo compactado pode ser copiado para outro host e então, subir a imagem em questão neste outro host).

EXEMPLO:



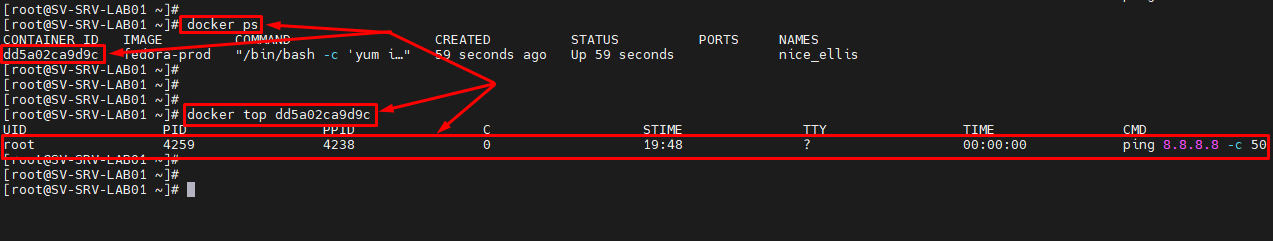
**docker history DIGITE\_O\_NOME\_DA\_IMAGEM\_DESEJADA** (Este comando permite identificarmos quais processos estão sendo e\ou foram executados por uma determinada imagem de container do docker).

EXEMPLO:



**docker top DIGITE\_O\_HASH\_ID\_DO\_CONTAINER** (Este comando permite visualizarmos qual ou quais comandos estão em execução em um determinado container do docker).

EXEMPLO:



**docker inspect DIGITE\_O\_HASH\_ID\_DO\_CONTAINER** (Este comando irá apresentar a configuração geral do container, demonstrando informações de quando foi criado e muito mais).

EXEMPLO:

Texto alternativo gerado por máquina:
[rootesv 
C roo tesv 
-SRV-LABOI 
-SRV-LABOI 
docker inspect ce8025c659e2 
"Id": "ce8625c659e2bee33277649cec616635a16b6ea764b413452e19caf6b65eeaaf" , 
"Created 
'I: 16:06 , 
"Path" . 
"/b in/bash" , 
"Args" 
"State" 
"Status 
"Runn ing" 
"exited", 
false, 
"Paused": 
false, 
false, 
"Restart ung" 
"00MKilled" : 
false, 
"Dead" 
false, 
"p idil 
"Exitcode" 
"Error" 
"StartedAt" 
. "2623-99-25Tee: 16:06 .24765724Z" , 
"FinishedAt": 
: 18 : 26 . 563737328Z" 
"sha256:c88ac1f841b72add46f5a8bee77c2ad6864d47e5663686ea64375acd55e27966" , 
" Image 
"Resolvconfpath" : 
" Iva r / I ib/docke r / conta ine r s /ce8Q25c659e2bee33277Q49cec616035a 16b6ea764b413452e 19caf6b65eeaaf / resolv . conf" , 
"Hostnamepath"• "/var/l ib/docker/containers/ce8925c659e2bee33277e49cec616935a16b6ea764b413452e19caf6b65eeaaf/hostname" , 
"Hostspath": 
" /var/l ib/docker/conta iners /ce8025c659e2bOe33277049cec616035a16b6ea764b413452e19caf6b65eeaaf lhos ts " , 
"Logpath" : 
" /var/l ib/docker/conta iners/ce8925c659e2bee33277e49cec616935a16b6ea764b413452e19caf6b65eeaaf/ce8925c659e2bee33277649cec616635a16b6ea764b413452 
e19caf6b65eeaaf-json . log" 
• "/busy_chandrasekhar" , 
"Name " 
"Res ta rtCount" 
"overIay2", 
"Driver 
"Platform": "I inux" , 
"MountLabeI" 
"Processl_abel"• 
"AppArmorprofiIe" 
nuli, 
"Exec1Ds " 
"Hos tconfig" 

**docker logs DIGITE\_O\_HASH\_ID\_DO\_CONTAINER** (Este comando irá apresentar em logs TODAS AS EXECUÇÕES\AÇÕES que o determinado container realizou, desde a sua inicialização até seu término).

EXEMPLO:

Texto alternativo gerado por máquina:
dd5a02ca9d9c 
agocaa3T08c 
43159d35af76 
60a719ed709e 
dadc684de771 
ddb2efe58cf9 
4022995f57d4 
ececeec3a567 
ela03af257c7 
e6782c78ab72 
6155ca86721f 
aae4c4bffc6c 
42edb16bd85b 
4bee1408d9f8 
b6f8d1ea8425 
14148945df13 
CV in ic ius . nunes@SV-SRV-LAB01 
r v in ius . nunes@SV-SRV-LAB01 
: 69 
ao : 04 
06:01 
a$ docker 
II/ un/bas 
"/bin/bash 
bin/bash" 
"/b in/bas 
"/bin/bash 
"/bin/bash 
"/bin/bash" 
"/bin/bash 
"/bin/bash 
"/bin/bash 
"/bin/bash 
"/bin/bash 
"/bin/bash 
"/bin/bash 
"/b in/bash" 
-i fedora- 
fedora-prod 
rod 
fedora 
-pr 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
fedora-prod 
-c 
-c 
-c 
-c 
-c 
-c 
-c 
-c 
-c 
-c 
-c 
-c 
yum um" 
i yum um" 
echo 
pung 
pung m" 
pung m" 
pung m" 
pung m" 
pung m" 
pung m" 
pung m" 
pung m" 
2 
2 
2 
2 
2 
2 
2 
2 
2 
2 
2 
2 
2 
2 
ours 
hours 
hours 
hours 
hours 
hours 
hours 
hours 
hours 
hours 
hours 
hours 
hours 
hours 
hours 
2.5 
2.8 
32 
2.1 
rod 
ago 
ago 
ago 
ago 
ago 
ago 
ag 
ago 
ago 
ago 
ago 
ago 
ago 
ago 
ago 
kB 
MB 
MB 
MB 
Exited 
Exited 
Exited 
ited 
Exu 
Exite 
Exited ( 
Exited 
Exite 
Exited 
Exited 
Exited 
Exited 
Exited 
Exited 
hours ago 
hours ago 
hours ago 
hours ago 
(127) 
27) 
( 130) 
d (127) 2 
(127) 2 
(127) 
(127) 
(127) 
(127) 
( 136) 
2 
2 
2 
2 
2 
2 
2 
2 
hours 
hours 
hours 
hours 
hou 
hours 
hours 
hours 
hours 
hours 
ago 
ago 
ago 
ago 
ago 
ago 
ago 
ag 
nuce ellis 
p r act ical_mccl untock 
charming_dij kstra 
practical_leakey 
affectionate_payne 
nuce lichterman 
zen_feynman 
affectionate nobel 
unruffled wescoff 
flamboyant_lewin 
vig ilant_lalande 
Iaugh ing_swanson 
eloquent_allen 
dreamy_brown 
keen_j ohnson 
reve rent_n ight ingale 
[v in ic ius . nunes@SV-SRV-LABe1 
CV in u: ius . nunes@SV-SRV-LAB01 
CV in ic ius .nunes@SV-SRV-LABe1 
Fedora 38 
x86 64 
a$ docker Io s 
Fedora 38 openh564 (From Cisco) 
x86 64 
Fedora Modular 38 
x86 64 
Fedora 38 
x86_64 - Updates 
Fedora Modular 38 
x86_64 - Updates 
Last metadata expurat ion check 
Dependencues resolved. 
dd5a02ca9d9c 
8 a MB/s 
1.6 kB/s 
2.2 MB/s 
7.2 MB/s 
1.7 MB/s 
ago 
on 
sun sep 24 
package 
Ins ta II ing : 
iput i Is 
Architecture 
x86 64 
Vers ton 
26221126-2. fc38 
2623. 
Repos itory 
fedo ra 
S Ize 
185 k 
Transact von Summary 
Insta II 1 package 

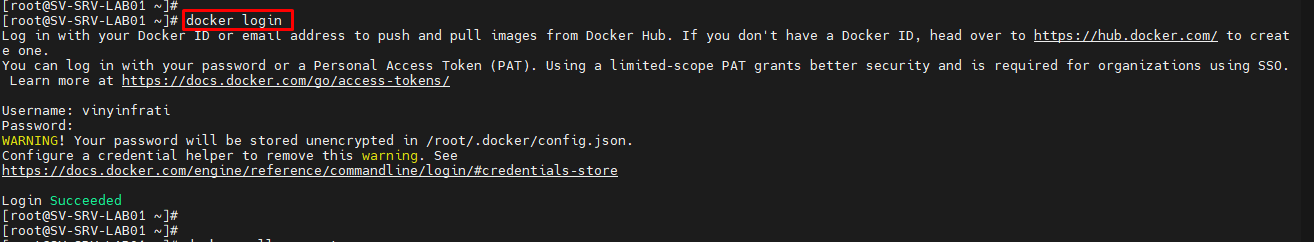
**docker logs -f DIGITE\_O\_HASH\_ID\_DO\_CONTAINER** (Este comando irá **apresentar de forma contínua** e em tempo real em logs TODAS AS EXECUÇÕES\AÇÕES que o determinado container realizou, desde a sua inicialização até seu término).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
-LABOI docker ps 
[rootesv-SRV 
CONTAINER ID 
57c18e161175 
6fd0867b6155 
[root@SV-SRV-LABe1 
[root@SV-SRV-LAB01 
[rootesv-SRV 
IMAGE 
my_apache92 
my_apache62 
COMAND 
"apache2ctl -D FOREG„" 
"apache2ctl -D FOREG„" 
CREATED 
2 hours 
2 hours 
PORTS 
192 . 168 . O. 
ago 
ago 
STATUS 
Up 2 hours 
Up 2 hours 
NAMES 
web -apache - 92 
web - apache62 
-LABOI docker Iogs - f 
57c180161175 
AH00558: apache2: coul 
globally to suppress this message 
server is fully 
qual ified doma in 
e.e.e.e 
name , 
using 172.17.0.3 
Set the 
Se rve rName 
d trect uve 

**docker login** (Este comando permite realizarmos login de autenticação no dockerhub, para que seja possível realizarmos downloads de imagens do docker no DockerHUB sem utilizar login anônimo, pois quando utilizamos login anônimo, ou seja, não se autenticamos com usuário e senha válidos no dockerhub, existe então um LIMITE de imagens que podemos realizar o download por dia, e muita das vezes isto impacta na produção da operação, onde o comando docker login irá conter estes problemas).

EXEMPLO:



**REDES NO DOCKER**

**brctl show docker0** (Este comando irá nos apresentar na saída stdout os detalhes do tráfego de rede do virtual switch docker0, ou seja, informações sobre o trágefo de redes de containers docker conectados ao SWITCH VIRTUAL em questão).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LAB02 
[root@SV-SRV-LAB02 
CONTAINER ID 
IMAGE 
e31b7239161f 
ng Inx 
[root@SV-SRV-LAB02 
Croot@SV-SRV-LAB02 
Croot@SV-SRV-LAB02 
Crootesv-SRv-LAB02 
bridge name 
bridge 
dockerO 
[root@SV-SRV-LAB02 
Croot@SV-SRV-LAB02 
docker ps 
COMAND 
"/docker -entrypo int . m" 
brctl show 
dockere 
CREATED 
25 minutes 
ago 
STATUS 
Up 25 minutes 
PORTS 
86/ tcp 
NAMES 
interesting bell 
enabled 
.92427b9eb686 
interfaces 
veth84dfd3d 

**A COLUNA "INTERFACES**" indica as interfaces de cada container conectada diretamente ao virtual switch DOCKER0, ou seja, conectados a interface docker0 do S.O (DOCKER HOST (SERVIDOR FÍSICO OU VIRTUAL) ) em questão.

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 docker 
COMAND 
5816cec61798 my_apache: 1. e 
ash" 
dd813937bea9 
my apache: 1.0 
" /b in/bas 
Iroot@SV-SRVLABOidoc er- ut 
ps 
CREATED 
53 seconds 
ago 
STATUS 
Up 52 seconds 
Up 2 minutes 
PORTS 
NAMES 
my_net_2 
my_net 1 
[rootesv-SRV 
[rootesv-SRV 
Crootesv-SRV 
bridge name 
dockere 
C roo tesv-SRV 
[rootesv-SRV 
[rootesv-SRV 
-LABOI 
-LABOI 
-LABOI 
bridge id 
.9242a6216fe8 
-LABOI 
-LABOI 
-LABOI 
brctl show dockere 
STP ena e 
terfac 
veth5a8f2b6 
veth67ae193 

Texto alternativo gerado por máquina:
Crootesv-SRV 
Crootesv-SRV 
bridge name 
dockere 
Crootesv-SRV 
Crootesv-SRV 
Crootesv-SRV 
-LABOI 
-LABOI 
bridge id 
.9242a9216fe8 
-LABOI 
-LABOI 
-LABOI 
brctl show dockere 
STP ena e 
veth5a8f2b6 
. Io: , LOWER mtu 65536 qdisc noqueue state UNKNOWN g 
link/loopback brd 
inet 127.0. O. 1/8 scope host Io 
valid Ift forever preferred Ift forever 
:i/128 scope host 
inet6 
valid Ift forever preferred Ift forever 
up default qlen 1666 
2: 
3: 
enpes3: <BROADCAST ,MULTICAST , UPTLOWER mtu 1566 qdisc fq_codel state UP roup default glen 1666 
I ink/ether 
brd 
inet 192. 168.0.200/24 brd 192. 168.0.255 scope global noprefixroute 
valid Ift forever preferred_lft forever 
dockere: <BROADCAST , LOWER mtu 1566 qdisc n 
ue state UP group default 
link/ether brd 
inet 172.17. O. 1/16 brd 172.17.255.255 scope global 
valid Ift forever preferred Ift forever 
inet6 feãe: scope I in 
<BROADCAST ,MULTICAST , LOWER UP> mtu 1500 qdisc noqueue master dockere state UP group default 
brd 
link-netnsid 
inet6 fe80: :accb:a2ff:fe98:88aa/64 scope link 
valid Ift forever 
.orefpr.red I ft for ever 
troot@SV-SRV-LABe1 
Croot@SV-SRV-LABe1 

Percebam no exemplo abaixo que acessamos um determinado container, onde identificamos que a interface VETH (Virtual Ethernet conectada ao Virtual Switch docker0 do DOCKER HOST) está utilizando uma faixa de IP ADDRESS dentro do barramento (/16) que está configurado no virtual Switch docker0 no DOCKER HOST.

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 docker 
r nNTOTN 
COMAND 
ps 
CREATED 
53 seconds ago 
2 minutes ago 
STATUS 
Up 52 seconds 
Up 2 minutes 
PORTS 
5816cec61798 
bin/bash" 
NAMES 
my_net_2 
my_net 1 
apac 
[root@SV-SRV-LABQi 
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
Croot@SV-SRV-LABe1 
bridge name 
dockere 
bridge id 
.9242a6216fe8 
brctl show dock 
STP enab d 
interfaces 
veth5a8f2b6 
veth67ae193 
Croot@SV-SRV-LAB61 
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
1: Io: 
docker attach 5816cec61798 
6 qdisc noqueue state UNKNOWN group default qlen 1666 
18: 
link/loopback br 
inet 127.0. O. 1/8 scope host Io 
ft forevqr_pref 
eth0@if19 ,MULTICAS 
oreu 
WER UP> mtu 1566 qdisc noqueue state UP group default 
inet 172.17.0 
va 
link-netnsid e 
. 3/16 rd 172.17.255.255 scope global ethe 
orever preferred Ift forever 
root@5816cec61798: 
root@5816cec61798: 
root@5816cec61798: 

**docker port DIGITE\_O\_NOME\_OU\_HASH-ID\_DO\_CONTAINER** (Este comando irá apresentar as portas do docker container (**portas a ESQUERDA do STDOUT do comando**) que estão sendo NETEADAS para as portas do DOCKER HOST (**portas a DIREITA do STDOUT do comando**), onde está espécie de "NAT" ocorre devido ao DOCKER\_PROXY que o serviço docker está executando para a troca de informação em questão nas portas do docker container vs docker host).

EXEMPLO:

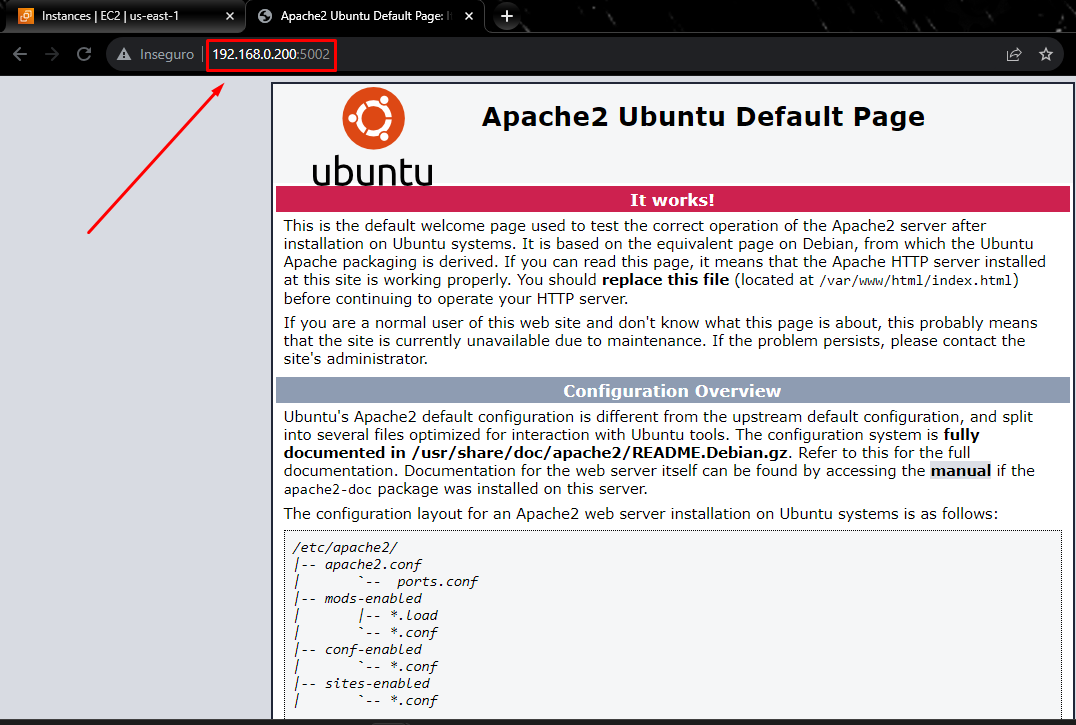
Texto alternativo gerado por máquina:
[rootesv-SRV 
[rootesv-SRV 
CONTAINER ID 
6fd0867b6155 
[rootesv-SRV 
[rootesv-SRV 
r rpptasv-SRV 
-LABOI 
-LABOI my_apache02] docker ps 
IMAGE 
C OM•IAN 
CREATED 
4 minutes 
ago 
STATUS 
Up 4 minutes 
PORTS 
my apache62 
"apache2ctl -D FOREG„" 
-LABOi 
-LABOI 
NAMES 
web-apac eo 
-LABOI m a docker port web-apache92 
80/tcp e.a.o.o:5001 
80/tcp l: : ] 
: 5001 
C root@SV-SRV-LABe1 

**docker run -d -p DIGITE\_O\_ENDEREÇO\_IP\_DESEJADO:DIGITE\_O\_NÚMERO\_DA\_PORTA\_DO\_DOCKER-HOST\_QUE\_SERÁ\_UTILIZADA:DIGITE\_O\_NÚMERO\_DA\_PORTA\_DO\_DOCKER-CONTAINER\_QUE\_SERÁ\_UTILIZADA --name=DIGITE\_O\_ALIAS (apelido ou nome que deseja definir)\_PARA\_O\_SEU\_CONTAINER DIGITE\_O\_NOME\_DA\_IMAGEM\_DESEJADA\_PARA\_O\_CONTAINER\_SE\_BASEAR\_NA\_CRIAÇÃO** (Este comando irá realizar a criação de um container baseado em uma determinada imagem, onde através do docker\_proxy, será definido uma espécie de NAT, fazendo com que uma determinada porta do DOCKER HOST envie as requisições feitas nesta determinada porta, para uma determinada porta dentro do DOCKER CONTAINER, assim funcionando como uma espécie de NAT através do docker\_proxy).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
57c180161175babfee8a05febb5029a4 
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
[root@SV-SRV-LABe1 
80/tcp 192.168.0.200:5002 
L root@S 
[root@SV-SRV-LABe1 
docker run 
d -p 192.168.0.200 
- -name=web-apache-92 my_apache02 
docker port web-apache-02 

ACESSANDO VIA BROWSER (NAVEGADOR) O ENDEREÇO IP DEFINIDO ACIMA NA DETERMINADA PORTA QUE TAMBÉM FOI DEFINIDA, PARA VALIDARMOS A ABERTURA DO SERVIÇO APACHE POR EXEMPLO:



**docker network inspect bridge** (Este comando irá apresentar na saída stdout os detalhes de conectividade do virtual switch (docker0) que está atualmente em produção).

EXEMPLO:

Texto alternativo gerado por máquina:
Crootesv 
C rootesv 
mtu": "1500" 
-SRV-LAB02 
- SRV-LAB02 
docker network inspect bridge 
"Name": "bridge" 
"Id". "98d66f457gef3f3672a748c2c85a71 6da8feb9284d2dc2f6211275e19a52e4" , 
"Created 
'I: "2923-16-23Tee: 12:04. 1846771 
-63 , 
"Scope": "local" , 
": "bridge" 
"Driver 
"Enable1Pv6" : 
false, 
"IPAM": { 
. "default" , 
"Driver" 
"Options" 
nuli, 
"Config" 
"Subnet": 
"Gateway" 
"172.17.0.0/16", 
: "172.17.0 1 
"Internal": 
false, 
"Attachable" 
false, 
false, 
"Ingress" 
"ConfigFrom": { 
"Netwo rk" 
"Config0nIy" 
false, 
o, 
"Contauners" 
"com.docker.network.bridge . default bridge 
"com.docker.network.bridge.enable icc": "true" 
"com.docker.network.bridge . enable_ ip_masquerade 
"com.docker.network.bridge . host binding_ipv4" 
"Opt ions " 
"Labels " 
"true" 
"true" 
. "6.e.e.6", 
"com.docker.network.bridge 
"com.docker.network . driver. 
. name 
71: "dockere", 

**docker system prune** (Este comando **APLICA UM RESET (REMOÇÃO) POR COMPLETO NO AMBIENET DOCKER**, onde irá remover todos os containers parados, remove todas as interfaces de rede sem uso por pelo menos um container e remove todas as imagens e cache sem vínculo com nenhum container).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LAB02 
Croot@SV-SRV-LAB92 docker system prune 

**docker system prune -a** (Este comando **APLICA UM RESET (REMOÇÃO) POR COMPLETO NO AMBIENET DOCKER**, onde irá remover todos os containers parados, remove todas as interfaces de rede sem uso por pelo menos um container, remove todas as imagens e cache sem vínculo com nenhum container, remove todas as imagens e limpa todo o cache de build do docker).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LAB02 
Croot@SV-SRV-LAB92 docker system prune -a' 

**COMANDOS DE DOCKER COMPOSE**

**docker compose up** (**Este comando deve ser executado dentro do mesmo diretório em que o arquivo docker-compose.yml está armazenado**, e então, este comando irá construir a infraestrutura definida dentro do arquivo docker-compose.yml)

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 cd 
/etc/docker/docker-compose/ 
[root@SV-SRV-LAB01 docker 
-compose]# 
[root@SV-SRV-LABe1 docker 
-compose]# 
[root@SV-SRV-LABe1 docker 
-compose]# docker compose up 
Running 0/2 
wordpress Pulling 
2.Qs 
db Pull ing 
2. as 

**docker compose up -d** (**Este comando deve ser executado dentro do mesmo diretório em que o arquivo docker-compose.yml está armazenado**, e então, irá construir a infraestrutura definida dentro do arquivo docker-compose.yml, contudo, **NÃO IRÁ PRENDER A SESSÃO DO USUÁRIO EM QUESTÃO AO SHELL DA SESSÃO DO DETERMINAOD USUÁRIO, MAS IRÁ EXECUTAR O COMANDO EM QUESTÃO EM MODO DETACHADO, OU SEJA, SEM PRENDER A SESSÃO DO USUÁRIO NO SHELL EM QUESTÃO, IMPEDINDO O USO DO SHELL**).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 docker-compose]# 
Croot@SV-SRV-LABe1 docker-compose]# oc er compose up 
[+] Running 2/2 
v' Contauner docker-compose-db-l 
Started 
Conta iner docker-compose-wordpress-l Started 
Croot@SV-SRV-LABe1 docker-compose]# 
[root@SV-SRV-LABe1 docker-compose]# 
o.os 
a.as 

**docker compose up -d** (**Este comando deve ser executado dentro do mesmo diretório em que o arquivo docker-compose.yml está armazenado**, e então, irá construir a infraestrutura definida dentro do arquivo docker-compose.yml **SEM PASSAR PELO CACHE DE BUILD DO DOCKER**, ou seja, irá construir totalmente os containers do zero).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 docker-compose]# 
C root@SV-SRV-LABe1 docker-compose]# docker compose up -d 
- -bui Id 

**docker compose down** (**Este comando deve ser executado dentro do mesmo diretório em que o arquivo docker-compose.yml está armazenado**, e então, irá realizar a PARADA de todos os containers e configurações que estão associados a infraestrutura definida dentro do arquivo docker-compose.yml).

EXEMPLO:

Texto alternativo gerado por máquina:
[root@SV-SRV-LABe1 docker-compose] 
Croot@SV-SRV-LABe1 docker-compose] docker compose down 
[+] Running 3/3 
Conta uner docker-compose-wordpress-l 
Conta iner docker-compose-db-l 
Removed 
Removed 
1.2s 
1.4s 
o.2s 
Network docker-compose_wordpress -network Removed 
Croot@SV-SRV-LABe1 docker-compose]# 
Croot@SV-SRV-LABe1 docker-compose]# 

**docker compose ps** (**Este comando deve ser executado dentro do mesmo diretório em que o arquivo docker-compose.yml está armazenado**, e então, irá apresentar somente os conatiners (services) que estão sendo gerenciados pelo DOCKER COMPOSE).

EXEMPLO:

Texto alternativo gerado por máquina:
Crootesv-SRv-LABe1 docker 
Croot@SV-SRV-LABe1 docker 
STATUS 
docker -compose- db-l 
Up About a munute 
ago 
-compose]# 
-compose] docker com ose ps 
IMAGE 
PORTS 
mysql:5.7.22 
COMAND 
"docker- entrypo unt . sh 
mysqld 
- de f ault_authent icat ion_plug in=mysql 
apache2 - fo reg round " 
nat uve_password" 
e .e.e.e: , 
SERVICE 
db 
wordp ress 
CREATED 
About a minute 
About a minute 
docker -compose-wordpress - 1 wordpress : latest 
"docker-entrypoint.sh 
Up About a minute 
ago 
Croot@SV-SRV-LABe1 