CAFÉ COM DOCKER

Folhetim 2.

INSTALAÇÃO DO DOCKER ENGINE NO UBUNTU.

A instalação do Docker Engine na distribuição Ubuntu exige alguns passos simples de instalação. Esse documento mostra apenas a parte da instalação do engine. Vale dizer que temos três modos de instalação do sistema operacional Linux ou Windows, para esse ambiente de estudo. São eles, máquina local, máquina virtual e Vagrant. Cada um desses processos tem suas vantagens e desvantagens e esse folhetim não descrever por questão de tempo.

Essa será a segunda máquina do ambiente de estudo do Docker que iremos desenvolver. Iremos focar no Guia do Docker 1.4 de 2020 para o estudo da certificação DCA.

A pratica começa ....

Vamos iniciar o processo de verificação se o sistema operacional está atualizado com o seguinte comando.

sudo apt update

```
vagrant@node01: ~
vagrant@node01:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:3 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 252 kB in 2s (117 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
98 packages can be upgraded. Run 'apt list --upgradable' to see them.
vagrant@node01:~$
```

Na imagem acima foi identificado que o sistema operacional não está atualizado. Iremos digitar o seguinte comando.

sudo apt upgrade -y

```
Get:86 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-twisted-bin amd64 17.9.0-
2ubuntu0.3 [11.5 kB]
Get:87 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-twisted all 17.9.0-2ubunt
u0.3 [1921 kB]
Get:88 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 landscape-common amd64 18.01-0ubu
ntu3.6 [85.6 kB]
Get:89 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsasl2-modules amd64 2.1.27~101
-g0780600+dfsg-3ubuntu2.4 [48.9 kB]
Get:90 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 linux-headers-4.15.0-187 all 4.15
```

Alguns pacotes sendo baixados e instalados no sistema GNU/Linux Ubuntu.

```
Unpacking lxd (3.0.3-0ubuntu1~18.04.2) over (3.0.3-0ubuntu1~18.04.1) ...
Preparing to unpack .../56-lxd-client_3.0.3-0ubuntu1~18.04.2_amd64.deb ...
Unpacking lxd-client (3.0.3-0ubuntu1~18.04.2) over (3.0.3-0ubuntu1~18.04.1) ...
Preparing to unpack .../57-snapd_2.54.3+18.04.2ubuntu0.2_amd64.deb ...
```

```
Setting up bind9-host (1:9.11.3+dfsg-1ubuntu1.17) ...
Setting up dnsutils (1:9.11.3+dfsg-1ubuntu1.17) ...
Processing triggers for systemd (237-3ubuntu10.53) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

Vamos agora repetir o comando de verificação de pacotes para verificação de pacotes pendentes.

sudo apt update

```
vagrant@node01:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
```

Finalizamos o processo de download e instalação dos pacotes do sistema operacional linux Ubuntu conforme a figura acima. Agora iremos verificar se o sistema operacional linux possui algum pacote ou software que faz referência ao Docker com o comando abaixo.

sudo apt list --installed | egrep docker

```
vagrant@node01:~$ apt list --installed | egrep docker

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

docker-ce/bionic,now 5:20.10.17~3-0~ubuntu-bionic amd64 [installed]
docker-ce-cli/bionic,now 5:20.10.17~3-0~ubuntu-bionic amd64 [installed]
docker-ce-rootless-extras/bionic,now 5:20.10.17~3-0~ubuntu-bionic amd64 [installed]
docker-compose-plugin/bionic,now 2.6.0~ubuntu-bionic amd64 [installed]
docker-scan-plugin/bionic,now 0.17.0~ubuntu-bionic amd64 [installed]
vagrant@node01:~$
```

Vimos que temos os pacotes docker já instalados no sistema. Mas esses pacotes são do repositório do docker ou do Ubuntu? Bom, esses pacotes são do repositório da distribuição Ubuntu. Dessa forma deveremos remover os pacotes mostrados na figura acima com o comando abaixo

sudo apt remove docker* containerd*

```
Package 'docker-engine' is not installed, so not removed
Note, selecting 'docker-doc' instead of 'docker.io-doc'
Package 'docker-engine-cs' is not installed, so not removed
Package 'docker-ee' is not installed, so not removed
Note, selecting 'containerd.io' for glob 'containerd*'
Note, selecting 'containerd' for glob 'containerd*'
Package 'docker' is not installed, so not removed
Package 'docker-compose' is not installed, so not removed
Package 'docker-containerd' is not installed, so not removed
Package 'docker-registry' is not installed, so not removed
Package 'docker-runc' is not installed, so not removed
Package 'docker2aci' is not installed, so not removed
Package 'containerd' is not installed, so not removed
Package 'docker-doc' is not installed, so not removed
Package 'docker.io' is not installed, so not removed
The following packages will be REMOVED:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  docker-scan-plugin
0 upgraded, 0 newly installed, 6 to remove and 0 not upgraded.
After this operation, 448 MB disk space will be freed.
Do you want to continue? [Y/n]
```

Escolha o “Y” para remover todas as referências citadas na pesquisa e assim removeremos tudo o que está relacionado a aplicação Docker do sistema operacional GNU/Linux Ubuntu

Na figura abaixo temos o informativo de sucesso do processo anterior.

```
The following packages will be REMOVED:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  docker-scan-plugin
0 upgraded, 0 newly installed, 6 to remove and 0 not upgraded.
After this operation, 448 MB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 90971 files and directories currently installed.)
Removing docker-ce-rootless-extras (5:20.10.17~3-0~ubuntu-bionic) ...
Removing docker-ce (5:20.10.17~3-0~ubuntu-bionic) ...
Warning: Stopping docker.service, but it can still be activated by:
  docker.socket
Removing containerd.io (1.6.6-1) ...
Removing docker-ce-cli (5:20.10.17~3-0~ubuntu-bionic) ...
Removing docker-compose-plugin (2.6.0~ubuntu-bionic) ...
Removing docker-scan-plugin (0.17.0~ubuntu-bionic) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
vagrant@node01:~$
```

Agora iremos adicionar e instalar o Docker Engine através do repositório oficial Docker com alguns comandos abaixo.

sudo apt-get install ca-certificates curl gnupg lsb-release

```
vagrant@node01:~$ sudo apt-get install ca-certificates curl gnupg  lsb-release
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (9.20170808ubuntu1).
lsb-release set to manually installed.
ca-certificates is already the newest version (20211016~18.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.18).
gnupg is already the newest version (2.2.4-1ubuntu1.5).
gnupg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
vagrant@node01:~$
```

sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

```
vagrant@node01:~$ sudo mkdir -p /etc/apt/keyrings
 https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
vagrant@node01:~$  curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
gpg: WARNING: unsafe ownership on homedir '/home/vagrant/.gnupg'
File '/etc/apt/keyrings/docker.gpg' exists. Overwrite? (y/N) y
vagrant@node01:~$
```

echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

```
vagrant@node01:~$ echo \
>   "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://downloa
d.docker.com/linux/ubuntu \
>   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
vagrant@node01:~$
```

Vamos listar os repositórios adicionados e verificar o repo do docker com os comandos abaixo

ll /etc/apt/source.list.d/
cat /etc/apt/source.list.d/docker.list

```
vagrant@node01:~$ ll /etc/apt/sources.list.d/
total 12
drwxr-xr-x 2 root root 4096 Jun 18 23:13 ./
drwxr-xr-x 8 root root 4096 Jun 18 23:13 ../
-rw-r--r-- 1 root root  113 Jun 19 00:53 docker.list
vagrant@node01:~$ cat /etc/apt/sources.list.d/docker.list
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu
bionic stable
vagrant@node01:~$
```

O repositório que queremos ver e se está instalado é o abaixo:

```
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu
bionic stable
```

Visto que o mesmo foi adicionado com sucesso. Agora iremos rodar um comando de atualização antes de iniciar o download e instalação dos pacotes do engine docker.

sudo yum update

```
vagrant@node01:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
vagrant@node01:~$
```

Todos os pacotes e aplicativos do sistema GNU/Linux estão atualizados e prontos para receber os pacotes de instalação e configuração do Engine Docker Com o comando abaixo iremos fazer o download e a instalação dos pacotes do docker engine.

sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin libltdl7 pigz
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
Recommended packages:
  slirp4netns
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  docker-scan-plugin libltdl7 pigz
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 96.2 kB/108 MB of archives.
After this operation, 449 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 libltdl7 amd64 2.4.6-2 [38.8 kB]
Fetched 96.2 kB in 1s (88.3 kB/s)
Selecting previously unselected package pigz.
(Reading database ... 90737 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.6.6-1_amd64.deb ...
Unpacking containerd.io (1.6.6-1) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../2-docker-ce-cli_5%3a20.10.17~3-0~ubuntu-bionic_amd64.deb ...
Unpacking docker-ce-cli (5:20.10.17~3-0~ubuntu-bionic) ...
```

O ultimo comando e mais importante é o Hello-World Docker que iremos chamar ele da seguinte forma.

sudo docker run hello-world

```
vagrant@node01:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:13e367d31ae85359f42d637adf6da428f76d75dc9afeb3c21faea0d976f5c651
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

vagrant@node01:~$
```

Se a imagem acima ficar idêntica em seu host, podemos considerar que o docker foi instalado com sucesso no GNU/Linux Ubuntu.

Nivel: Iniciante
Autor: Renato Rossi Ferreira
Email: renato.rossi.ferreira@gmail.com
Linkedin: www.linkedin.com/in/renatoticloud