

CAFÉ COM DOCKER

Folhetim 1.

INSTALAÇÃO DO DOCKER ENGINE NO CENTOS 7.

A instalação do Docker Engine na distribuição CentOS exige alguns passos simples de instalação. Esse documento mostra apenas a parte da instalação do engine. Hoje, não irei descrever os passos de instalação do sistema operacional Linux CentOS 7. Em outros artigos, foram descritos os processos de instalação e configuração básica do sistema no host. 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 primeira 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. Ela cobre 6 tópicos. Vamos para prática do folhetim.

A prática começa

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

`sudo yum update`

```
xfsprogs           x86_64      4.5.0-22.el7      base           897 k
yum                noarch     3.4.3-168.el7.centos base           1.2 M
yum-plugin-fastestmirror noarch    1.1.31-54.el7_8   base            34 k
zlib               x86_64     1.2.7-20.el7_9    updates         90 k
Installing for dependencies:
bc                 x86_64     1.06.95-13.el7    base           115 k

Transaction Summary
=====
Install  1 Package (+1 Dependent package)
Upgrade 134 Packages

Total download size: 254 M
Is this ok [y/d/N]:
```

Nessa imagem verificamos que o sistema operacional não está atualizado. Quando digitarmos o `y` o Linux fará atualização dos pacotes mencionados na pesquisa anterior.

```
(100/136): openssl-1.0.2k-25.el7_9.x86_64.rpm | 494 kB 00:00:00
(101/136): openssl-libs-1.0.2k-25.el7_9.x86_64.rpm | 1.2 MB 00:00:00
(102/136): polkit-0.112-26.el7_9.1.x86_64.rpm | 170 kB 00:00:00
(103/136): pyldb-1.5.4-2.el7.x86_64.rpm | 49 kB 00:00:00
(104/136): python-2.7.5-90.el7.x86_64.rpm | 96 kB 00:00:00
```

Parte do processo de download e instalação dos pacotes da lista acima.

```

xfsprogs.x86_64 0:4.5.0-22.el7
yum.noarch 0:3.4.3-168.el7.centos
yum-plugin-fastestmirror.noarch 0:1.1.31-54.el7_8
zlib.x86_64 0:1.2.7-20.el7_9

Complete!
[vagrant@node02 ~]$

```

Finalizamos o processo de download e instalação dos pacotes da atualização do sistema operacional linux CentOS 7 conforme a figura acima. Agora iremos verificar se o sistema operacional linux possui algum pacote ou software que faz referencia ao Docker com o comando abaixo.

sudo yum list installed | egrep docker

```

[vagrant@node02 ~]$ sudo yum list installed | egrep docker
containerd.io.x86_64                1.6.6-3.1.el7                @docker-ce-stable
docker-ce.x86_64                    3:20.10.17-3.el7             @docker-ce-stable
docker-ce-cli.x86_64                1:20.10.17-3.el7             @docker-ce-stable
docker-ce-rootless-extras.x86_64  20.10.17-3.el7               @docker-ce-stable
docker-compose-plugin.x86_64        2.6.0-3.el7                  @docker-ce-stable
docker-scan-plugin.x86_64           0.17.0-3.el7                 @docker-ce-stable
[vagrant@node02 ~]$

```

Vimos que temos os pacotes docker já instalados no sistema. Mas esses pacotes são do repositório do docker ou do CentOS? Bom, esses pacotes são do repositório da distribuição epel 7. Para a prova DCA o docker recomenda utilizar o seu repositório como oficial. Dessa forma deveremos remover os pacotes mostrados na figura acima com o comando abaixo

sudo yum remove docker* containerd*

```

Resolving Dependencies
--> Running transaction check
----> Package containerd.io.x86_64 0:1.6.6-3.1.el7 will be erased
----> Package docker-ce.x86_64 3:20.10.17-3.el7 will be erased
----> Package docker-ce-cli.x86_64 1:20.10.17-3.el7 will be erased
----> Package docker-ce-rootless-extras.x86_64 0:20.10.17-3.el7 will be erased
----> Package docker-compose-plugin.x86_64 0:2.6.0-3.el7 will be erased
----> Package docker-scan-plugin.x86_64 0:0.17.0-3.el7 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                Arch          Version           Repository        Size
=====
Removing:
containerd.io                          x86_64        1.6.6-3.1.el7     @docker-ce-stable 125 M
docker-ce                              x86_64        3:20.10.17-3.el7  @docker-ce-stable  96 M
docker-ce-cli                          x86_64        1:20.10.17-3.el7  @docker-ce-stable 140 M
docker-ce-rootless-extras              x86_64        20.10.17-3.el7    @docker-ce-stable  19 M
docker-compose-plugin                  x86_64        2.6.0-3.el7       @docker-ce-stable  25 M
docker-scan-plugin                     x86_64        0.17.0-3.el7      @docker-ce-stable  13 M
=====

Transaction Summary
=====
Remove 6 Packages

Installed size: 418 M
Is this ok [y/N]:

```

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

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

```
Is this ok [y/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Erasing      : docker-compose-plugin-2.6.0-3.el7.x86_64                1/6
  Erasing      : 3:docker-ce-20.10.17-3.el7.x86_64                      2/6
  Erasing      : docker-ce-rootless-extras-20.10.17-3.el7.x86_64       3/6
  Erasing      : docker-scan-plugin-0.17.0-3.el7.x86_64                 4/6
  Erasing      : 1:docker-ce-cli-20.10.17-3.el7.x86_64                 5/6
  Erasing      : containerd.io-1.6.6-3.1.el7.x86_64                     6/6
  Verifying    : containerd.io-1.6.6-3.1.el7.x86_64                     1/6
  Verifying    : docker-ce-rootless-extras-20.10.17-3.el7.x86_64       2/6
  Verifying    : 1:docker-ce-cli-20.10.17-3.el7.x86_64                 3/6
  Verifying    : 3:docker-ce-20.10.17-3.el7.x86_64                     4/6
  Verifying    : docker-scan-plugin-0.17.0-3.el7.x86_64                 5/6
  Verifying    : docker-compose-plugin-2.6.0-3.el7.x86_64               6/6

Removed:
  containerd.io.x86_64 0:1.6.6-3.1.el7                docker-ce.x86_64 3:20.10.17-3.el7
  docker-ce-cli.x86_64 1:20.10.17-3.el7                docker-ce-rootless-extras.x86_64 0:20.10.17-3.el7
  docker-compose-plugin.x86_64 0:2.6.0-3.el7            docker-scan-plugin.x86_64 0:0.17.0-3.el7

Complete!
[vagrant@node02 ~]$
```

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

```
sudo yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo
```

Na figura abaixo temos o processo de adição do repositório

```
[vagrant@node02 ~]$ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
[vagrant@node02 ~]$
```

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

sudo yum repolist

```
[vagrant@node02 ~]$ sudo yum repolist
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.uepg.br
 * extras: mirror.uepg.br
 * updates: mirror.uepg.br
repo id                                repo name                                status
base/7/x86_64                          CentOS-7 - Base                          10,072
docker-ce-stable/7/x86_64               Docker CE Stable - x86_64                163
extras/7/x86_64                         CentOS-7 - Extras                        512
updates/7/x86_64                       CentOS-7 - Updates                       3,863
repolist: 14,610
[vagrant@node02 ~]$
```

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

```
docker-ce-stable/7/x86_64 Docker CE Stable - x86_64 163
```

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@node02 ~]$ sudo yum update
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.uepg.br
 * extras: mirror.uepg.br
 * updates: mirror.uepg.br
No packages marked for update
[vagrant@node02 ~]$
```

sudo yum upgrade -y

```
[vagrant@node02 ~]$ sudo yum upgrade -y
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.uepg.br
 * extras: mirror.uepg.br
 * updates: mirror.uepg.br
No packages marked for update
[vagrant@node02 ~]$
```

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

sudo yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin

```
=====
Package                                Arch      Version      Repository    Size
=====
Installing:
containerd.io                          x86_64    1.6.6-3.1.el7  docker-ce-stable 33 M
docker-ce                              x86_64    3:20.10.17-3.el7  docker-ce-stable 22 M
docker-ce-cli                          x86_64    1:20.10.17-3.el7  docker-ce-stable 29 M
docker-compose-plugin                  x86_64    2.6.0-3.el7      docker-ce-stable 7.0 M
Installing for dependencies:
docker-ce-rootless-extras              x86_64    20.10.17-3.el7  docker-ce-stable 8.2 M
docker-scan-plugin                     x86_64    0.17.0-3.el7     docker-ce-stable 3.7 M
Transaction Summary
=====
Install 4 Packages (+2 Dependent packages)

Total download size: 104 M
Installed size: 418 M
Is this ok [y/d/N]:
```

Escolha a opção “y” para dar como ok.

A confirmação dos pacotes instalados no sistema após o download segue na imagem abaixo.

```
vagrant@node02:~$ sudo dnf install docker
(6/6): docker-scan-plugin-0.17.0-3.el7.x86_64.rpm | 3.7 MB 00:00:00
-----
Total | 27 MB/s | 104 MB 00:00:03
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : docker-scan-plugin-0.17.0-3.el7.x86_64 1/6
  Installing : 1:docker-ce-cli-20.10.17-3.el7.x86_64 2/6
  Installing : containerd.io-1.6.6-3.1.el7.x86_64 3/6
  Installing : docker-ce-rootless-extras-20.10.17-3.el7.x86_64 4/6
  Installing : 3:docker-ce-20.10.17-3.el7.x86_64 5/6
  Installing : docker-compose-plugin-2.6.0-3.el7.x86_64 6/6
  Verifying : containerd.io-1.6.6-3.1.el7.x86_64 1/6
  Verifying : docker-ce-rootless-extras-20.10.17-3.el7.x86_64 2/6
  Verifying : 1:docker-ce-cli-20.10.17-3.el7.x86_64 3/6
  Verifying : 3:docker-ce-20.10.17-3.el7.x86_64 4/6
  Verifying : docker-scan-plugin-0.17.0-3.el7.x86_64 5/6
  Verifying : docker-compose-plugin-2.6.0-3.el7.x86_64 6/6

Installed:
  containerd.io.x86_64 0:1.6.6-3.1.el7          docker-ce.x86_64 3:20.10.17-3.el7
  docker-ce-cli.x86_64 1:20.10.17-3.el7        docker-compose-plugin.x86_64 0:2.6.0-3.el7

Dependency Installed:
  docker-ce-rootless-extras.x86_64 0:20.10.17-3.el7  docker-scan-plugin.x86_64 0:0.17.0-3.el7

Complete!
[vagrant@node02 ~]$
```

Agora é necessário habilitar o serviço do docker engine durante o boot do sistema com o comando abaixo.

```
sudo systemctl enable docker
```

Após habilitar o serviço do docker durante a inicialização do sistema é necessário também rodar um start no serviço do docker com o comando abaixo.

```
sudo systemctl start docker
```

Após o start do serviço é recomendado verificar o status do serviço com o comando abaixo.

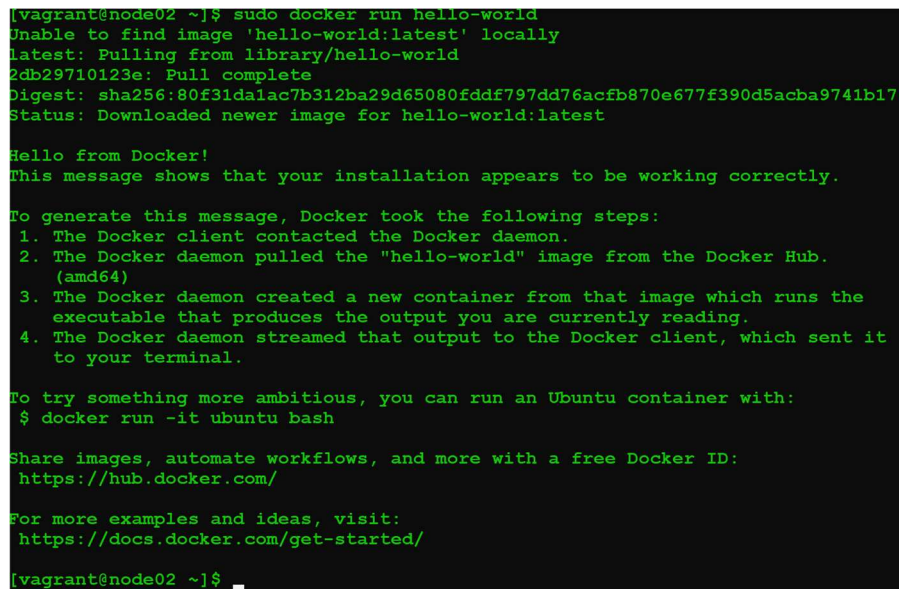
```
Sudo systemctl status docker
```

```
[vagrant@node02 ~]$ sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[vagrant@node02 ~]$ sudo systemctl start docker
[vagrant@node02 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2022-06-12 22:58:55 UTC; 9s ago
     Docs: https://docs.docker.com
  Main PID: 1034 (dockerd)
    Tasks: 8
   Memory: 36.4M
    CGroup: /system.slice/docker.service
            └─1034 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jun 12 22:58:54 node02.docker-dca.example dockerd[1034]: time="2022-06-12T22:58:54.908078177Z"...pc
```

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

```
sudo docker run hello-world
```

A terminal window with a black background and green text. The prompt is [vagrant@node02 ~]\$. The command sudo docker run hello-world is entered. The output shows that the 'hello-world:latest' image was not found locally, so it was pulled from the Docker Hub. The pull is complete, and the digest is sha256:80f31dalac7b312ba29d65080fddf797dd76acfb870e677f390d5acba9741b17. The status is 'Downloaded newer image for hello-world:latest'. Then, 'Hello from Docker!' is printed, followed by a message stating that the installation appears to be working correctly. A list of four steps is shown: 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. Then, a suggestion to try something more ambitious is given: 'To try something more ambitious, you can run an Ubuntu container with: \$ docker run -it ubuntu bash'. Below that, a link to 'https://hub.docker.com/' is provided, followed by a link to 'https://docs.docker.com/get-started/'. The prompt [vagrant@node02 ~]\$ is shown at the bottom.

```
[vagrant@node02 ~]$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:80f31dalac7b312ba29d65080fddf797dd76acfb870e677f390d5acba9741b17
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@node02 ~]$
```

Se a imagem acima ficar idêntica em seu host, podemos considerar que o docker foi instalado com sucesso no GNU/Linux CentOS 7. Esse foi um passo a passo na forma pratica. Em outros folhetins de Café com Docker irei colocar mais conteúdo técnico.

Nivel: Iniciante

Autor: Renato Rossi Ferreira

Email: renato.rossi.ferreira@gmail.com

Linkedin: www.linkedin.com/in/renatoticloud