# Tutorial para Instalação e

# Utilização do Docker

**Sistema Operacional:** Debian 10

**Fonte:** https://phoenixnap.com/kb/how-to-install-docker-on-debian-10

## Instalação:

### Instalar os pré-requisitos:

sudo apt-get update

sudo apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common

### Instalar o Docker:

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian buster stable"

sudo apt-get update

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

### Verificar o status do serviço após instalação:

sudo systemctl status docker

### Rodando seu primeiro container:

sudo docker run hello-world

### Verificar informações, será possível constatar que um container foi adicionado:

sudo docker info

## Executando Docker sem sudo (recomendável):

### Verifique quais grupos estão disponíveis no seu sistema, rode o comando (procure pelo grupo docker):

cat /etc/group | cut -d: -f1

ou

cat /etc/group | grep docker

ou

Se quiser rodar um comando que lhe diz se existe ou não o grupo, rede o comando:

[[ $(grep 'docker' /etc/group) ]] && echo 'Existe o grupo docker' || echo 'Nao existe o grupo docker'

### Então, crie o grupo (caso não exista) com o comando:

sudo groupadd docker

### Agora adicione seu usuario ao grupo docker com o comando:

sudo usermod -aG docker $USER

### Verifique se o seu usuário está no grupo com o comando:

groups $USER

### Reiniciar a máquina e fazer o comando "docker info" para validar que seu usuário pode usar o docker sem o sudo.

## Build do Dockerfile:

### O script do Dockerfile está na pasta “Docker” no Sharepoint:

<https://tinyurl.com/4zuhjz6b>

### Faça o download da pasta **ubuntu18-developer** e copie-o para uma pasta vazia de seu computador.

### Acesse essa pasta pelo Terminal e realize a build através do comando (há um ponto no final do comando):

docker build \  
--build-arg username=${USER} \  
--build-arg uid=$(id -u) \  
--build-arg gid=$(id -g) \  
-t ubuntu18-developer .

### Nesta etapa, será efetuado, via script, o download da imagem **ubuntu18-environment**, cujo Dockerfile também se encontra no Sharepoint. Por isso, não será necessário efetuar a build.

### Uma vez concluído, será possível conferir todas as imagens Docker através do comando:

docker images

### A pasta **persist**, contém arquivos importantes para a utilização dos simuladores e do VS Code. Ela pode ser salva em qualquer local do host. Para este tutorial, a pasta será salva em:

${HOME}/Documentos

### Para executar a imagem compilada, utilize o comando a seguir (adequações podem ser necessárias dependendo do local onde as pastas **persist** e **scratch** estiverem no **host**):

docker run -it --rm --name container\_ubu18-dev -e DISPLAY=$DISPLAY \  
-v "${HOME}/.Xauthority:${HOME}/.Xauthority:rw" \  
-v "/tmp/.X11-unix:/tmp/.X11-unix" \  
-v "${HOME}/source/ns-3.30.1/scratch:/simulators/source/ns-3.30.1/scratch" \  
-v "${HOME}/Documentos/persist/ndnlua:/simulators/ndnSIM/.local/lib/wireshark/plugins/2.6" \  
-v "${HOME}/Documentos/persist/udpclient:/simulators/source/ns-3.30.1/src/applications/model" \  
-v "${HOME}/Documentos/persist/Fapesp:${HOME}/Fapesp" \  
-v "${HOME}/Documentos/persist/vscode:/simulators/ndnSIM/ns-3/scratch/.vscode" \  
-v "${HOME}/Documentos/persist/vscode:/simulators/source/ns-3.30.1/scratch/.vscode" \  
ubuntu18-developer

**Obs.:** A criação dos volumes permite que haja persistência dos dados utilizados no container. Quando o container é encerrado e novamente executado, ele volta ao estado original, garantindo a integridade e a inviolabilidade do ambiente. Mudanças definitivas na imagem deverão ser realizadas via build. Dados a serem persistidos deverão permanecer em volumes.

## Lista de alguns comandos úteis:

### Listar as imagens:

docker images

### Remover uma imagem:

docker rmi <nome ou ID da imagem>

### Listar os processos ou contêineres:

docker ps -a

### Remover um contêiner:

docker rm <nome ou ID do contêiner>

### Remover todos os contêineres (usar com cuidado):

docker rm $(docker ps -a -q)

### Parar a execução de um contêiner:

docker stop <nome ou ID do contêiner>

### Iniciar e executar um contêiner:

docker start <nome ou ID do contêiner>

docker exec -it <nome ou ID do contêiner> bash

### Para efetuar o login, preparando-se para realizar um push (crie a conta no https://hub.docker.com/):

docker login

### Para fazer o push, primeiramente atribua uma tag à imagem e, por fim, realize o push:

docker tag exemplo repositorioquefoicriado/exemplo

docker push repositorioquefoicriado/exemplo