

Conhecendo o Ambiente:

1. Faça o login via cliente de SSH com o usuário ubuntu e senha cloudsenha (caso não tenha trocado).
2. No terminal digite o comando **screen** caso você precise de uma ajuda.
(este comando permitirá que o instrutor acesse o seu terminal, para tirar dúvidas)
3. O terminal deve aparecer da seguinte maneira `"ubuntu@t[1-20]:~$ "`
4. Para se tornar o administrador de sua máquina digite o seguinte comando: **sudo -i**
(é necessário para a maioria de nossas atividades)
5. Agora você pode fazer todas as atividades utilizando este terminal ;)

Atividade 1 - Conhecendo o ambiente Docker

No seu servidor, em um terminal com poderes administrativo "root" verifique os seguintes elementos:

- A. Verificar os pacotes instalados: **dpkg -l | grep docker**
OBS: Caso o Docker não esteja instalado utilize:
apt -y install docker.io # ou utilize a instalação docker-ce
- B. Verifique se o docker está instalado em seu servidor com o comando: **docker -v**
- C. Verifique o serviço que habilita o Docker engine com o comando:
systemctl status docker
- D. Configure para iniciar em todos os boot de sua máquina:
systemctl enable docker
- E. Verifique a API Docker com o comando: **netstat -lnp | grep docker.sock**
OBS: Caso o netstat não esteja instalado utilize: *apt -y install net-tools*
- F. Veja quem tem permissão de acessar a API do socket unix com o comando:
ls -l /run/docker.sock

Atividade 2 - Imagens para seu container Docker

No seu servidor, em um terminal com poderes administrativo "root" verifique os seguintes elementos:

- A. Verifique as imagens disponíveis localmente utilizando o seguinte comando:
docker images
- B. Faça o download da imagem oficial do serviço web nginx da versão 1.18 com o comando: **docker pull nginx:1.18**

- C. Verifique novamente as imagens locais disponíveis em seu ambiente (passo A).
- D. Faça o download da imagem oficial do nginx sem passar a versão com o comando:
docker pull nginx
- E. Verifique novamente as imagens locais disponíveis em seu ambiente (passo A).
- F. Apague a imagem do nginx versão 1.18 com o comando: **docker rmi nginx:1.18**
- G. Veja as camadas da imagem do nginx com o comando: **docker image history nginx**

Atividade 3 - Iniciando seu primeiro container Docker

No seu servidor, em um terminal com poderes administrativo “root” verifique os seguintes elementos:

Vamos iniciar o seu primeiro container com o serviço web apache (http) com os seguintes comandos:

- A. Download da imagem do Apache com o comando: **docker pull centos**
- B. Iniciar a execução do container com a imagem do Apache com o comando:
docker run -it -d --name meucentos centos
- C. Vamos verificar a situação de nosso container CentOS com o comando: **docker ps**
- D. Vamos verificar a saída da aplicação default do container CentOS com o comando:
docker attach --detach-keys 'ctrl-y' meucentos
- E. Execute CTRL e Y para fechar (desatachar) a saída da aplicação default do container CentOS de seu (pseudo-TTY).
- F. EXTRA: Quais comandos podemos verificar qual é a aplicação default do container CentOS ?
- G. Vamos parar o container CentOS com o comando: **docker stop meucentos**
- H. Vamos verificar a situação de nosso container Apache com o comando: **docker ps**
- I. Verifique a situação de todos os container criados com o comando: **docker ps -a**
- J. Inicie novamente o container com o comando: **docker start meucentos**
- K. Verifique a situação de todos os container criados com o comando: **docker ps -a**
- L. Faça o attach novamente na aplicação default do seu container centos e execute o seguinte comando: **sleep 9876 &**
- M. Execute CTRL + Y para desatachar do terminal (pseudo-TTY) do container centos.
- N. Busque na árvore de processo do seu Ubuntu, o processo sleep com o argumento 9876 com o comando: **ps tree -a -l -n**
- O. Execute o seguinte comando e analise a saída: **docker top meucentos**
- P. Execute o seguinte comando e analise a saída: **docker top meucentos ef**
- Q. Pare o container meucentos e depois remova ele completamente.

Limpendo todo o ambiente:

- A. Pare todos os container com o comando: `docker stop $(docker ps -q)`
- B. Apague todos os container que estão parados: `docker rm $(docker ps -a -q)`

Atividade 4 - Criando a sua aplicação PHP

No seu servidor, em um terminal com poderes administrativo “root” verifique os seguintes elementos:

- A. Inicie um container PHP com Apache utilizando o seguinte comando: `docker run -it -d --name meuphp -p 80[01-20]:80 php:apache`
Exemplo: Minha instância de ssh é de porta 2011 então meu número é 8011 acima.
- B. Em seu browser acesse a URL `http://tcontainer.ic.unicamp.br:80[01-20]/` (lembrando de trocar o número de porta da sua instância)
- C. Execute um nova aplicação em seu container (a aplicação que vamo executar é o bash) com o comando: `docker exec -it meuphp bash`
- D. Verifique se está no documentroot do Apache com o comando: `pwd`
- E. Se você não estiver em `/var/www/html` navegue até lá: `cd /var/www/html`
- F. Execute o seguinte comando para alterar a página inicial: `echo "Meu nome é _____" > index.php`
- G. Acesse novamente em seu browser a URL de seu servidor Apache.
- H. No bash do seu container meuphp sai com o comando: `exit`
- I. No terminal em seu servidor crie um arquivo chamado `info.php` com o comando: `nano info.php`
- J. Na edição do arquivo `info.php` inclua o seguinte conteúdo:
`<?php phpinfo(); ?>`
- K. Tecle CTRL e X para sair, seguido Y para salvar e Enter para confirmar
- L. Vamos copiar o arquivo para dentro do container com o comando: `docker cp info.php meuphp:/var/www/html/`
- M. Acesse novamente em seu browser a URL de seu servidor Apache com o caminho `http://tcontainer.ic.unicamp.br:80[01-20]/info.php`

Limpe o ambiente conforme a Atividade 3