

**Aluno: Gabriel Sebold dos Santos**

**Turma: 30**

**Curso: Sistemas de Informação**

*Parte 1: O objetivo dessa atividade é entender como funciona o esquema de virtualização usando o VirtualBox.*

*Instalar o VirtualBox*

*Efetuar o Download do Ubuntu Server*

*Instalar o Ubuntu Server No Virtual Box*

*Instalar o apache2*

*Acessar o website através do navegador local*

*Em anexo consta alguns links com manuais para o procedimento.*

*Parte 2: Entender como funciona a execução de um container.*

*Acessar o endereço:*

*<https://labs.play-with-docker.com/>*

*Rodar uma imagem Nginx exposta na porta 80*

*Dica: Pode ser usado IA generativa para a execução dos comandos.*

## **Atividade 1:**

Após seguir o tutorial de instalação do Ubuntu Server na máquina virtual rodei o seguinte comando:

- **`sudo apt update && sudo apt upgrade -y`**

1. Com comando ***sudo*** dando permissão de administrador;
2. ***Apt update*** verifica se há novas versões de pacotes disponíveis nos servidores;
3. ***&&*** Operador lógico para apenas fazer a segunda condição se a primeira for atendida.
4. ***apt upgrade -y*** atualiza todos os pacotes do sistema para as versões mais recentes disponíveis nos repositórios.  
(O ***-y*** responde "sim" automaticamente para todas as perguntas de confirmação, evitando que o usuário tenha que confirmar manualmente).

Em seguida utilizei:

- ***sudo apt install apache2 -y***

Esse comando instala o Apache2 sem precisar de interação do usuário. Após a instalação, o Apache pode ser iniciado e configurado para hospedar sites e aplicações web.

Logo após foi utilizado o comando:

- ***sudo systemctl status apache2***

O comando `sudo systemctl status apache2` serve para verificar se o Apache2 está rodando no sistema. Ele mostra se o serviço está ativo, seu ID de processo e detalhes sobre sua execução.

```

127.0.1.1
gabrielsebold@ubuntu:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-03-20 02:03:51 UTC; 3min 32s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 846 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 874 (apache2)
    Tasks: 55 (limit: 1064)
   Memory: 7.7M (peak: 8.0M)
      CPU: 68ms
   CGroup: /system.slice/apache2.service
           └─874 /usr/sbin/apache2 -k start
             └─876 /usr/sbin/apache2 -k start
               └─877 /usr/sbin/apache2 -k start

Mar 20 02:03:50 ubuntu:server systemd[1]: Starting apache2.service - The Apache HTTP Server...
Mar 20 02:03:51 ubuntu:server apachectl[867]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the
Mar 20 02:03:51 ubuntu:server systemd[1]: Started apache2.service - The Apache HTTP Server.
Quit
gabrielsebold@ubuntu:~$

```

- ***sudo ufw allow 'Apache'***
- ***ufw***: É a ferramenta de firewall simplificado do Ubuntu e outras distribuições baseadas no Debian.
- ***allow***: Permite que o tráfego para uma determinada aplicação ou porta passe pelo firewall.
- ***'Apache'***: Especifica a aplicação Apache. Quando você usa 'Apache', o ufw já sabe que deve permitir as portas associadas ao Apache, que são as portas 80 (HTTP) e 443 (HTTPS).

- ***sudo ufw enable***

- ***sudo ufw enable***

Ativa o firewall. Após esse comando, o UFW começará a filtrar o tráfego de rede conforme as regras configuradas.

- ***sudo ufw status***

Exibe o status atual do firewall e as regras configuradas no sistema.

Segue print dos 3 comandos:

```
lines 1-17/17 (END)
gabrielsebold@ubuntu:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-03-20 02:03:51 UTC; 3min 32s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 846 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 874 (apache2)
      Tasks: 55 (limit: 1064)
     Memory: 7.7M (peak: 8.0M)
        CPU: 68ms
    CGroup: /system.slice/apache2.service
            └─874 /usr/sbin/apache2 -k start
              └─876 /usr/sbin/apache2 -k start
                └─877 /usr/sbin/apache2 -k start

Mar 20 02:03:50 ubuntu:systemd[1]: Starting apache2.service - The Apache HTTP Server...
Mar 20 02:03:51 ubuntu:apachectl[867]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the
Mar 20 02:03:51 ubuntu:systemd[1]: Started apache2.service - The Apache HTTP Server.
Quit
gabrielsebold@ubuntu:~$
gabrielsebold@ubuntu:~$
gabrielsebold@ubuntu:~$
gabrielsebold@ubuntu:~$ sudo ufw allow 'Apache'
Skipping adding existing rule
Skipping adding existing rule (v6)
gabrielsebold@ubuntu:~$
gabrielsebold@ubuntu:~$
gabrielsebold@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
gabrielsebold@ubuntu:~$
gabrielsebold@ubuntu:~$ sudo ufw status
Status: active

To Action From
--
apache ALLOW Anywhere
apache (v6) ALLOW Anywhere (v6)

gabrielsebold@ubuntu:~$
gabrielsebold@ubuntu:~$
gabrielsebold@ubuntu:~$
gabrielsebold@ubuntu:~$
```

Logo depois utilizei:

- ***ip a***
- ***Hostname -I***

Ao rodar ***ip a***, você obterá informações detalhadas sobre todas as interfaces de rede do seu sistema;

Ao rodar **hostname -I**, exibe apenas o endereço IP da máquina, sem informações extras sobre interfaces ou estado da rede.

Para verificar o IP no qual o servidor estava hospedado, segue print:

```
└─876 /usr/sbin/apache2 -k start
└─877 /usr/sbin/apache2 -k start

Mar 20 02:03:50 ubuntu:server systemd[1]: Starting apache2.service - The Apache HTTP Server...
Mar 20 02:03:51 ubuntu:server apachectl[867]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the
Mar 20 02:03:51 ubuntu:server systemd[1]: Started apache2.service - The Apache HTTP Server.
Quit
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$ sudo ufw allow 'Apache'
Skipping adding existing rule
Skipping adding existing rule (v6)
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$ sudo ufw enable
Firewall is active and enabled on system startup
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$ sudo ufw status
Status: active

To Action From
--
Apache ALLOW Anywhere
Apache (v6) ALLOW Anywhere (v6)

gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$ $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:15:7a:20 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.101/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 6825sec preferred_lft 6825sec
    inet6 fe80::a00:27ff:fe15:7a20/64 scope link
        valid_lft forever preferred_lft forever
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$
gabrielsebold@ubuntu:server~$ $ hostname -I
192.168.1.101
gabrielsebold@ubuntu:server~$
```

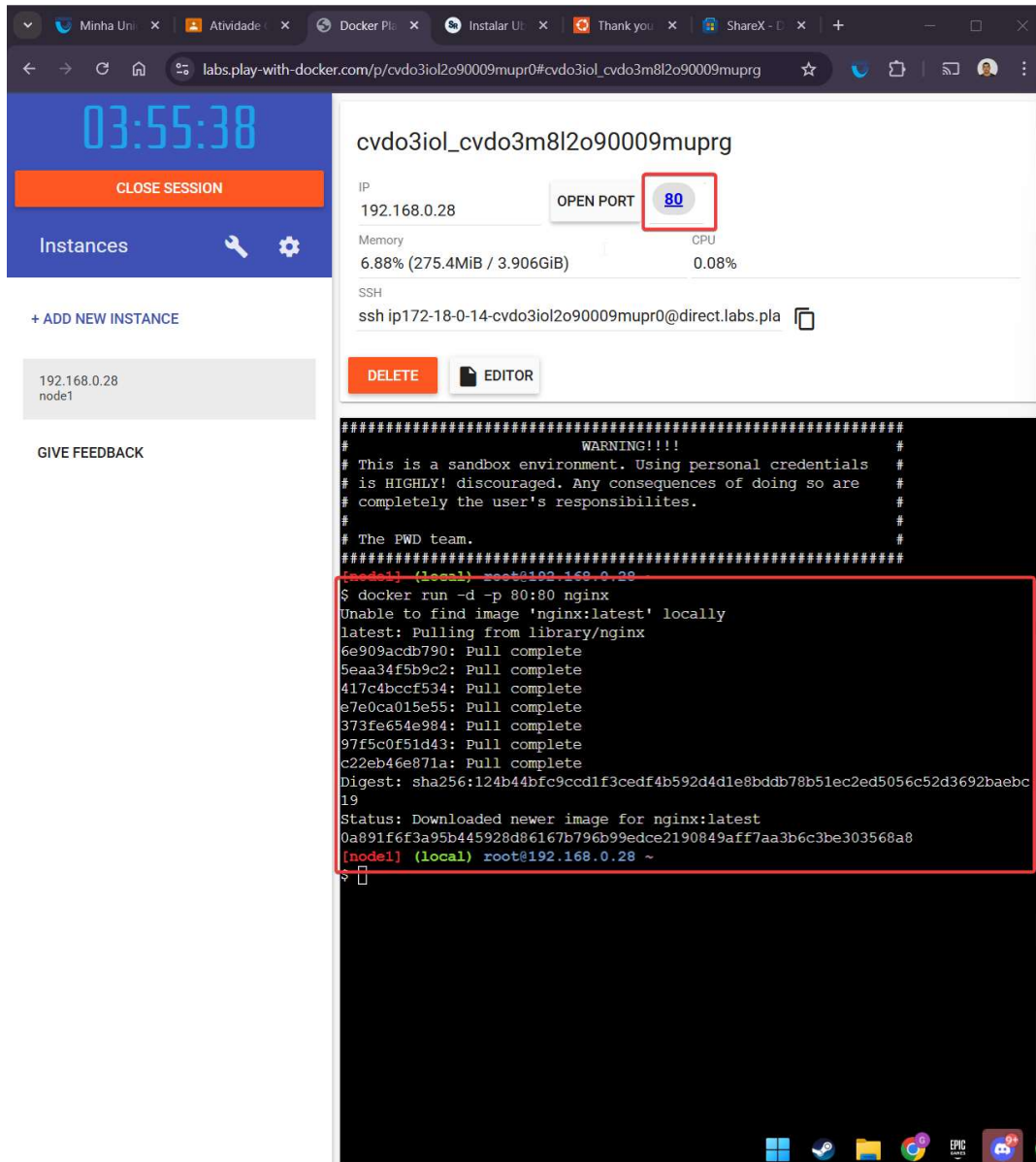
Para conclusão acessei o **IP** em questão:



## Atividade 2:

Após inicializar o container docker utilizei o comando:

- **`docker run -d -p 80:80 nginx`**
- **`docker run`**: Comando para iniciar um container.
- **`-d`**: Executa o container em modo detached (em segundo plano).
- **`-p 80:80`**: Mapeia a porta 80 do container para a porta 80 da instância, permitindo acesso externo.
- **`nginx`**: Nome da imagem que será utilizada



Após isso cliquei na opção “OPEN PORT” e adicionei 80 para abrir a porta 80 e então precionei “OK”

03:53:42

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.28  
node1

GIVE FEEDBACK

labs.play-with-docker.com diz

What port would you like to open?

80

OK

Cancelar

SSH

ssh ip172-18-0-14-cvdo3iol2o90009mupr0@direct.labs.pla

DELETE

EDITOR

```
#####
#                               #
#   WARNING!!!!                #
#   This is a sandbox environment. Using personal credentials         #
#   is HIGHLY! discouraged. Any consequences of doing so are         #
#   completely the user's responsibilites.                             #
#   The PWD team.                                                      #
#####
[node1] (local) root@192.168.0.28 ~
$ docker run -d -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
6e909acdb790: Pull complete
5eaa34f5b9c2: Pull complete
417c4bccf534: Pull complete
e7e0ca015e55: Pull complete
373fe654e984: Pull complete
97f5c0f51d43: Pull complete
c22eb46e871a: Pull complete
Digest: sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc
19
Status: Downloaded newer image for nginx:latest
0a891f6f3a95b445928d86167b796b99edce2190849aff7aa3b6c3be303568a8
[node1] (local) root@192.168.0.28 ~
$
```

We'd love to hear about your usage of Play with Docker. Please take a moment to fill out our [survey](#).



**Segue print da imagen nginx:**

