



Portforwarding -

Portforwarding é quando queremos expor uma porta da máquina virtual para que ela seja acessada pela máquina local.

Pra isso ficar mais claro, vou subir um serviço na máquina virtual e acessar ele pelo meu windows local. Primeiro eu vou me conectar à máquina virtual com o comando **“vagrant ssh”**:

```
C:\Users\lucas\Desktop\ambiente_dev\Bionic>vagrant ssh
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Feb  7 20:38:02 UTC 2022

System load:  0.0               Processes:    90
Usage of /:   2.5% of 61.80GB    Users logged in: 0
Memory usage: 11%              IP address for eth0: 10.0.2.15
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

Last login: Thu Jan 27 23:27:10 2022 from 10.0.2.2
vagrant@vagrant:~$
```

Agora eu posso instalar um serviço nessa máquina, a ideia é que esse serviço pode ser qualquer coisa, como um **banco de dados por exemplo**. Mas como é fins didáticos vou instalar o **servidor web ginex**(ele é um software de servidor web bem leve, funciona bem pra requisições HTTP).

Como isso é uma máquina linux antes é preciso fazer algumas configurações, a primeira é **atualizar todos os pacotes**, dá pra fazer isso usando o comando **“sudo apt-get update”**:

```
vagrant@vagrant:~$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security/main i386 Packages [1,106 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [1,415 kB]
19% [6 Packages 448 kB/1,415 kB 32%] [5 Packages 531 kB/1,106 kB 48%] 24.2 kB/s 10min 2s
```

Agora basta instalar o **nginx**, dá pra fazer isso com o comando **“sudo apt-get install -y nginx”**:

```
vagrant@vagrant: ~
vagrant@vagrant:~$ sudo apt-get install -y nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0 lib
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-
  libx11-6 libx11-data libxcb1 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
```

Pra testar se o servidor do **nginx** está rodando dá pra fazer um comando **“netstat -lntp”** esse comando lista portas que estão em **listening** com o **número da porta** em **protocolo tcp** e o **número do processo**. **A primeira linha já mostra que o nginx está escutando na porta 80:**

```
vagrant@vagrant:~$ vagrant@vagrant:~$ netstat -lntp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Pr
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
vagrant@vagrant:~$
```

Melhor ainda, que tal só fazer uma requisição com o comando “`curl http://localhost`” localhost é a própria máquina dârr, é retornado uma **página HTML**:

```
vagrant@vagrant:~$ vagrant@vagrant:~$ curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
width: 35em;
margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
vagrant@vagrant:~$
```

Agora que a máquina virtual está configurada com **nginx** vem a parte principal de tudo que é acessar essa porta da minha **máquina virtual** pela minha **máquina local**, a ideia é que eu use uma porta específica da **máquina local(host)** e quando a **máquina local** recebe uma requisição nessa porta ele sabe que tem que chamar a **máquina virtual**

Essa configuração pode ser feita via **Vagrantfile** usando a **Portforwarding**, novamente montando a **infraestrutura como código**, onde podemos especificar qual porta da máquina virtual (guest) será chamada ao usar a porta da máquina local (host):

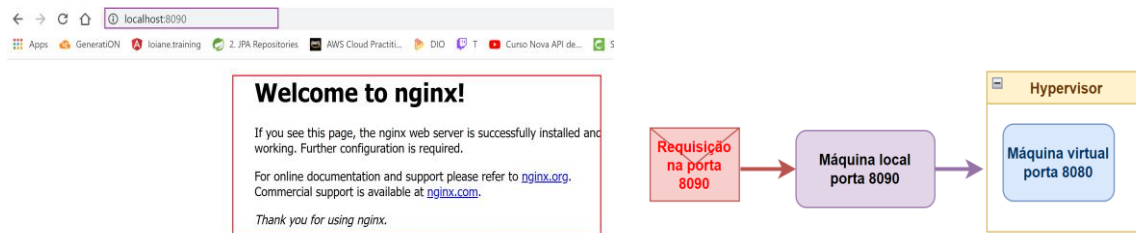
```
Vagrant.configure("2") do |config|
  config.vm.box = "hashicorp/bionic64"
  config.vm.network "forwarded_port", guest: 80, host: 8090
end
```

Reiniciando a máquina virtual já dá pra ver no console a configuração da forwarded port:

```
:\Users\lucas\Desktop\ambiente_dev\Bionic>vagrant halt
=> default: Attempting graceful shutdown of VM...

:\Users\lucas\Desktop\ambiente_dev\Bionic>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
=> default: Checking if box 'hashicorp/bionic64' version '1.0.282' is up to date...
=> default: Clearing any previously set forwarded ports...
=> default: Clearing any previously set network interfaces...
=> default: Preparing network interfaces based on configuration...
  default: Adapter 1: nat
=> default: Forwarding ports...
  default: 80 (guest) => 8090 (host) (adapter 1)
  default: 22 (guest) => 2222 (host) (adapter 1)
=> default: Booting VM...
=> default: Waiting for machine to boot. This may take a few minutes...
  default: SSH address: 127.0.0.1:2222
  default: SSH username: vagrant
  default: SSH auth method: private key
```

Basta agora eu fazer uma **requisição** na minha máquina local (**host**) na porta **8090** que ele irá redirecionar para a máquina virtual (**guest**) na porta **80** que irá devolver a requisição para minha máquina local:



*****Aqui foi instalado somente um nginx, mas a ideia é que qualquer recurso possa ser instalado na máquina virtual, então fazemos um portforwarding para acessar esse recurso*****