



## Linux Laravel

### Virtual Machine

## Introdução

Este documento visa fornecer uma guia sobre como configurar um ambiente IaaS na IBM Cloud de forma a configurar uma Máquina Virtual e configurar uma aplicação laravel e expô-la na internet.

## Passo a Passo

### Chave SSH na IBM Cloud

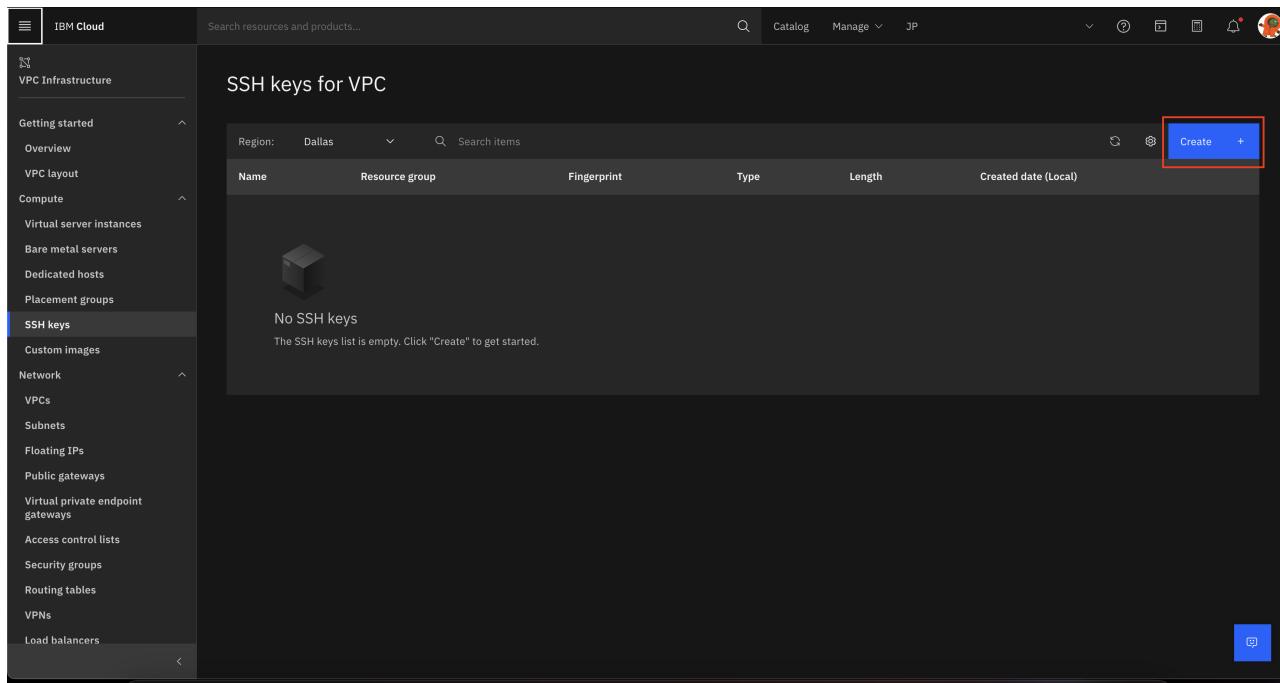
Antes de mais nada para que seja possível seguir corretamente esse tutorial é preciso criar uma chave ssh pois ela será utilizada para acessar a máquina virtual criada e realizar as operações necessárias para disponibilizar a aplicação. Existem muitos métodos e tutoriais de como criar uma chave sshm recomendamos fortemente o disponibilizado pelo [GitHub](#), siga somente os passos da seção "Gerar uma nova chave SSH", ao final desses passos você deve ter em seu diretório dois arquivos, uma chave pública e uma privada. A chave privada você deverá renomear para `cred.pem` pois este será o nome utilizada pelo tutorial.

De posse das chaves acesse a IBM Cloud: <https://cloud.ibm.com/>

1. No painel inicial, clique no menu no canto superior esquerdo, selecione a opção VPC e depois SSH.

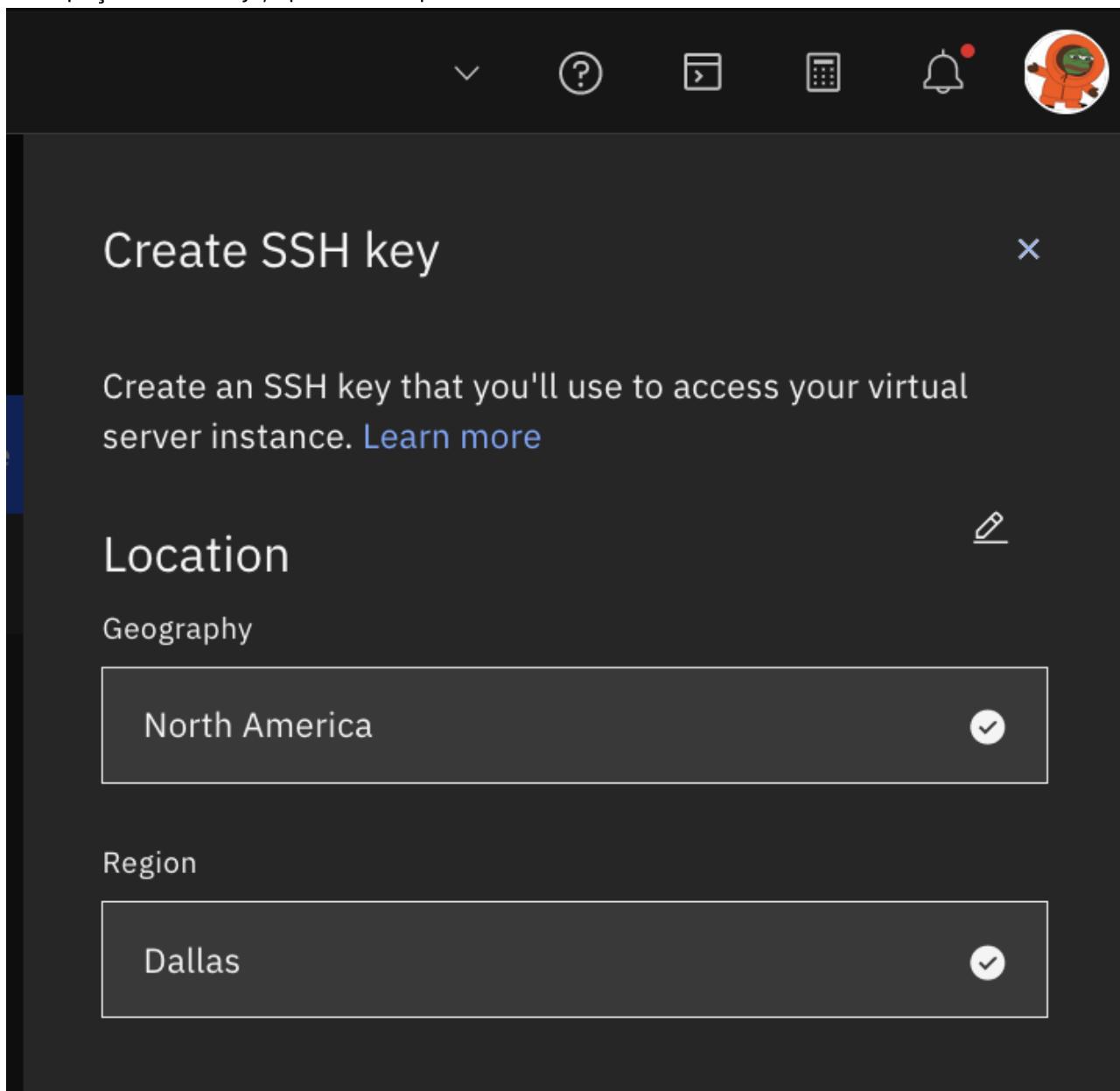
The screenshot shows the IBM Cloud dashboard with the navigation menu open. The 'VPC Infrastructure' section is highlighted with a red box. Within 'VPC Infrastructure', the 'SSH keys' option is also highlighted with a red box. The main content area displays various cloud services like 'Build a virtual machine', 'Virtual Server Details', 'Explore IBM Cloud Shell', 'Build a Virtual Private Cloud (VPC)', 'Explore Bare Metal Servers for VPC', and 'IBM Power Systems Virtu Server'. Below the main content, there's a 'Planned maintenance' section listing events and an 'IBM Cloud status' map.

2. Com o painel de chaves aberto vamos carregar nossa nova chave, clique no botão "Create" do lado direitiro na tela.



The screenshot shows the IBM Cloud VPC Infrastructure SSH keys page. The left sidebar is titled "VPC Infrastructure" and includes sections for Getting started, Overview, VPC layout, Compute, Virtual server instances, Bare metal servers, Dedicated hosts, Placement groups, SSH keys (which is selected and highlighted in blue), Custom images, Network, VPCs, Subnets, Floating IPs, Public gateways, Virtual private endpoint gateways, Access control lists, Security groups, Routing tables, VPNs, and Load balancers. The main content area is titled "SSH keys for VPC" and shows a table with columns: Name, Resource group, Fingerprint, Type, Length, and Created date (Local). A message at the bottom says "No SSH keys" and "The SSH keys list is empty. Click "Create" to get started." A "Create" button is located in the top right corner of the table header, with a red box highlighting it.

3. Escolha a localização de armazenamento da chave, de um nome para a chave e cole a chave pública no espaço "Public Key", apos isso clique em "Create"



The screenshot shows the "Create SSH key" dialog. At the top, there are several icons: a dropdown arrow, a question mark, a forward arrow, a calculator, a bell with a red dot, and a user profile icon. The title of the dialog is "Create SSH key". Below the title, a sub-instruction reads "Create an SSH key that you'll use to access your virtual server instance. [Learn more](#)". The main configuration section is titled "Location" and has a "Geography" sub-section. Under "Geography", the option "North America" is selected, indicated by a checked checkbox icon in the top right corner of its input field. Another section titled "Region" shows the option "Dallas" with a checked checkbox icon in its input field.

# Details

Name

Enter key name

Resource group (i)

Default



[View all resource groups](#)

Tags (i)

Examples: env:dev, version-1

Access management tags (i)

Examples: access:dev, proj:version-1

Public key

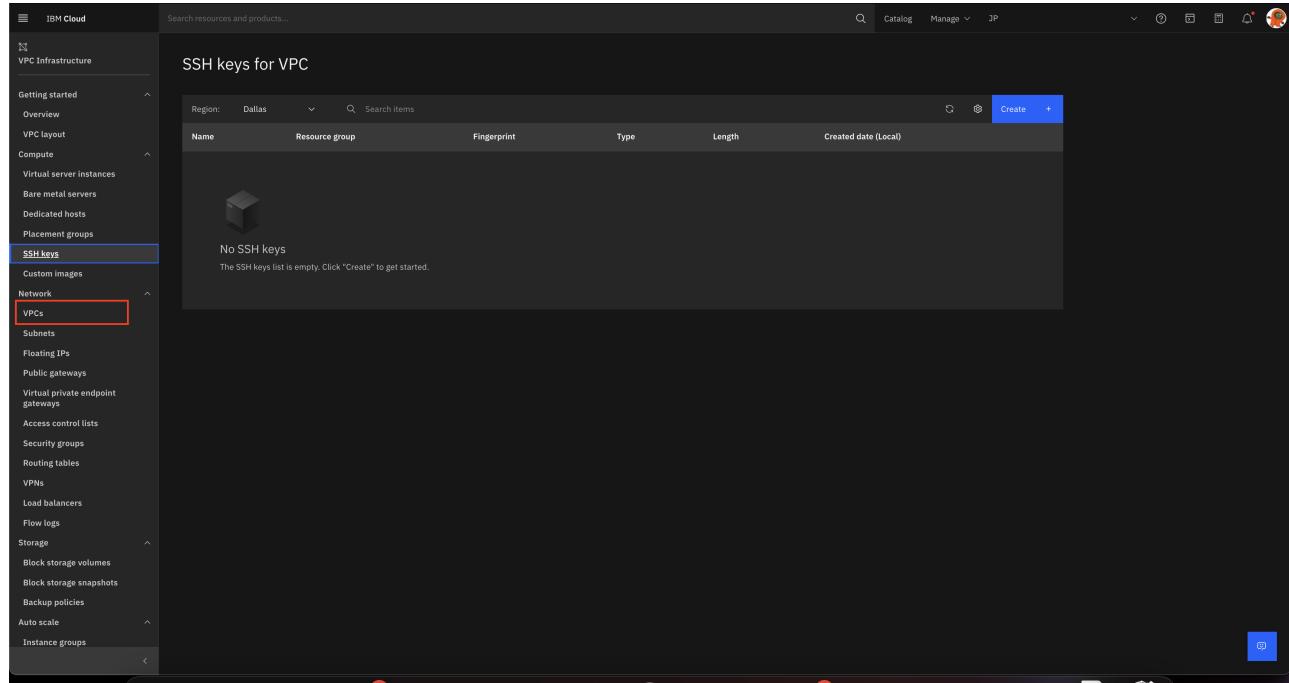
Enter public key

Cancel

Upgrade account

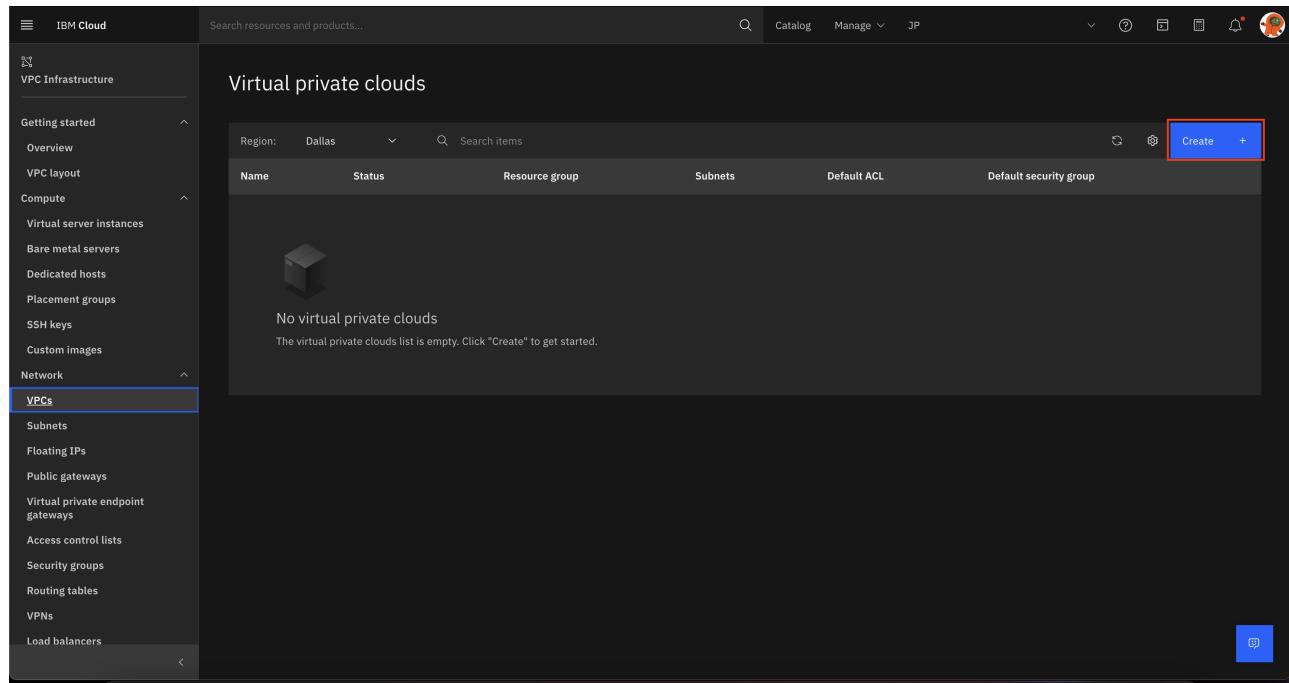
#### 4. A chave deve aparecer no painel de chaves

5. Com chave criada e carregada agora é preciso criar a rede privada onde a nossa aplicação irá rodar.  
Do lado esquerdo, clique em VPCs.



The screenshot shows the IBM Cloud VPC Infrastructure section. On the left, the navigation menu is expanded, with 'VPCs' highlighted and a red box drawn around it. The main panel title is 'SSH keys for VPC'. It displays a table with columns: Name, Resource group, Fingerprint, Type, Length, and Created date (Local). A message at the bottom says 'No SSH keys' and 'The SSH keys list is empty. Click "Create" to get started.' A blue 'Create' button is located in the top right corner of the table header.

6. Do lado direito, clique no botão "Create" para criar uma nova VPC



The screenshot shows the IBM Cloud VPC Infrastructure section. On the left, the navigation menu is expanded, with 'VPCs' highlighted and a red box drawn around it. The main panel title is 'Virtual private clouds'. It displays a table with columns: Name, Status, Resource group, Subnets, Default ACL, and Default security group. A message at the bottom says 'No virtual private clouds' and 'The virtual private clouds list is empty. Click "Create" to get started.' A blue 'Create' button is located in the top right corner of the table header.

7. Escolha onde ela será criada e de um nome a ela, em seguida basta clicar no botão "Create virtual private cloud"

The screenshot shows the 'Create' interface for a Virtual Private Cloud (VPC) in the IBM Cloud. The 'Location' section is set to North America (Dallas). The 'Details' section includes fields for Name (Enter a unique name), Resource group (default), Tags (Examples: env=dev, version=1), and Access management tags (Examples: access=dev, propagation=1). Under 'VPC default access control list', it shows Default security group (Allow SSH, Allow ping selected). Under 'Classic access', it shows Enable access to classic resources (Create a default prefix for each zone selected). The 'Subnets' section lists three subnets: sn-20231226-01 (Dallas 1, default, 10.240.0.0/18, 256 addresses, IP range 10.240.0.0/24, Public gateway Detached), sn-20231226-02 (Dallas 2, default, 10.240.64.0/18, 256 addresses, IP range 10.240.64.0/24, Public gateway Detached), and sn-20231226-03 (Dallas 3, default, 10.240.128.0/18, 256 addresses, IP range 10.240.128.0/24, Public gateway Detached). On the right, there's a sidebar with 'Summary' and 'Virtual private cloud provided' buttons, and a 'Total estimated cost \$0.00/mo' section.

8. Uma vez com a rede criada, agora é necessário criar a máquina virtual para executar a aplicação. Do lado esquerdo, clique em "Virtual server instances"

The screenshot shows the 'Virtual server instances' list interface in the IBM Cloud. The left sidebar navigation bar is expanded, showing categories like VPC Infrastructure, Getting started, Compute, and VPCs. The 'Virtual server instances' link under Compute is highlighted with a red box. The main content area displays a table of existing virtual server instances:

Name	Status	Resource group	Subnets	Default ACL	Default security group
ce-browser-vpc	Available	tg-ups-api-journey	3	prudent-neuron-frozen-canal	septet-mammal-nearby-purist
demo-vpc	Available	tg-estimating-edge	3	crowns-gloomy-amusement-banknote	glory-eagle-motivate-colocate
telus-vpc	Available	tg-telus-communications	3	reward-rug-ocelot-cough	twenty-latterly-monetize-isolated
lumen-vpc	Available	ce-lumen	3	expenses-dreamt-pampers-lent	companion-humvee-refund-antiquity
refinitiv-vpc	Available	tg-refinitiv-us	0	outcast-blast-devote-demotion	audacious-freebee-elated-crawling
na-cop-vpc	Available	tg-refinitiv-us	3	trunks-definite-leverage-culotte	fencing-unsafe-tabloid-strength
vpc-jp	Available	default	3	footboard-dividing-purifier-elusive	spoils-heroism-autumn-marsupial
telusmevcpc	Available	tg-telus-communications	3	unmoved-essence-habital-platform	defiant-headlock-false-legwarmer
lumen2	Available	ce-lumen2	3	trio-coastline-pushchair-unhappily	life-nullify-wrangle-itinerary
vpc-lumen2	Available	ce-lumen2	3	decaf-refinery-backyard-tantrum	hardener-getaway-citadel-rectangle

Below the table, it says 'Items per page: 10' and '1-10 of 10 items'. At the bottom, there's a 'What do you want to do next?' section with a 'Create' button.

9. Clique no botão "Create" do lado direto da tela.

The screenshot shows the IBM Cloud interface for managing VPC infrastructure. On the left, there's a sidebar with various options like Overview, Compute, and Network. Under Compute, 'Virtual server instances' is selected and highlighted with a blue box. The main area displays a table with columns: Name, Status, Resource group, Virtual Private Cloud, Profile, Reserved IP, and Floating IP. A message at the top says 'Virtual server instances for VPC' and 'Region: Dallas'. Below the table, it says 'No virtual server instances' and 'The virtual server instances list is empty. Click "Create" to get started.' A 'Create' button is located in the top right corner of the main content area, which is also highlighted with a red box.

10. Configure a máquina de acordo com sua necessidade. Aqui o importante é associar a essa maquina a VPC criada no passo 7 e com a chave ssh do passo 1 para que tenhamos acesso. Selecione a o sistema operacional que melhor se adeque a suas necessidades, neste tutorial foi utilizado o Ubuntu.

This screenshot shows the 'Create Virtual server for VPC' configuration page. It includes sections for Server type (Intel x86 architecture selected), Architecture (IBM Z, LinuxONE), Hosting type (Public), Location (Geography: North America, Region: Dallas, Zone: Dallas 2), and Details (Name, Resource group: default). On the right side, there's a summary table showing costs for a Virtual server instance (\$0.096/hr), Image (\$0.096/hr), Boot volume (\$0.013/hr), and Network interface (\$0.001/hr). The 'Create' button at the bottom left is highlighted with a red box.

11. Com a máquina virtual criada, agora é necessário expor ela para que seja possível acessá-la. Para isso clique em "Floating IPs" do lado esquerdo.

The screenshot shows the IBM Cloud interface for managing VPC Infrastructure. On the left sidebar, under 'Compute', 'Virtual server instances' is selected. The main panel displays a single virtual server instance named 'php'. The 'Overview' tab is active, showing details like Name (php), Virtual private cloud (automation-system-vpc), Created date (December 19 2022 at 2:22:57 PM), Profile (bx2-2x8), and Image (Debian GNU/Linux 11.x Bullseye/Stable - Minimal Install (amd64)). The 'Monitoring' tab is also visible. A 'Monitoring preview' section shows CPU usage (0.22%) and Memory used (9.9%). Below the instance details, there's a 'Storage volumes' section with a table and a 'Launch monitoring' button.

12. Vamos criar um IP para acessar a máquina. Clique em "Reserve" do lado direito da tela.

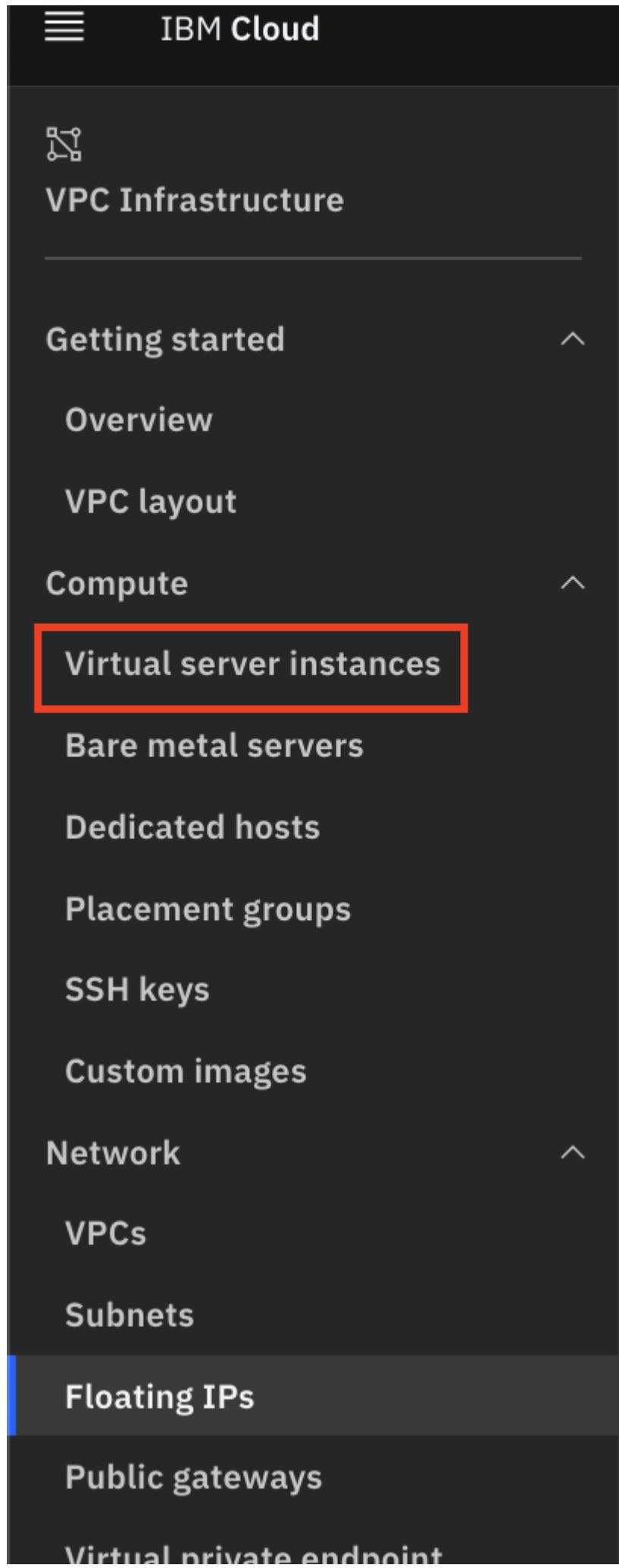
The screenshot shows the 'Floating IPs for VPC' page. Under the 'Region' dropdown, 'Dallas' is selected. The table lists one floating IP entry: 'Name' (php-test), 'Status' (Bound), 'Address' (52.118.206.177), 'Location' (Dallas 2), 'Targeted device' (php - eth0), and 'Target type' (Virtual server instance). A 'Reserve' button is highlighted with a red box on the right side of the table.

13. Selecione a região, de um nome e selecione a vm criada para que o IP seja associado a ela. Em seguida clique em "Reserve".

The screenshot shows the 'Reserve' dialog for the floating IP 'php-test'. In the 'Details' section, the 'Target instance or server' dropdown is set to 'Virtual server instance' and 'php'. Other fields include 'Enter floating ip name' (empty), 'Resource group' (Default), 'Tags' (empty), 'Access management tags' (empty), 'Resource to bind' (checkbox checked), and 'No available network interface' (checkbox unchecked). At the bottom, the 'Total estimated cost' is shown as \$1.00/mo. The 'Reserve' button is at the bottom right.

14. Agora com o IP alocado já é possível acessar a VM via ssh. Guarde esse IP pois ele será utilizado para acessar sua VM

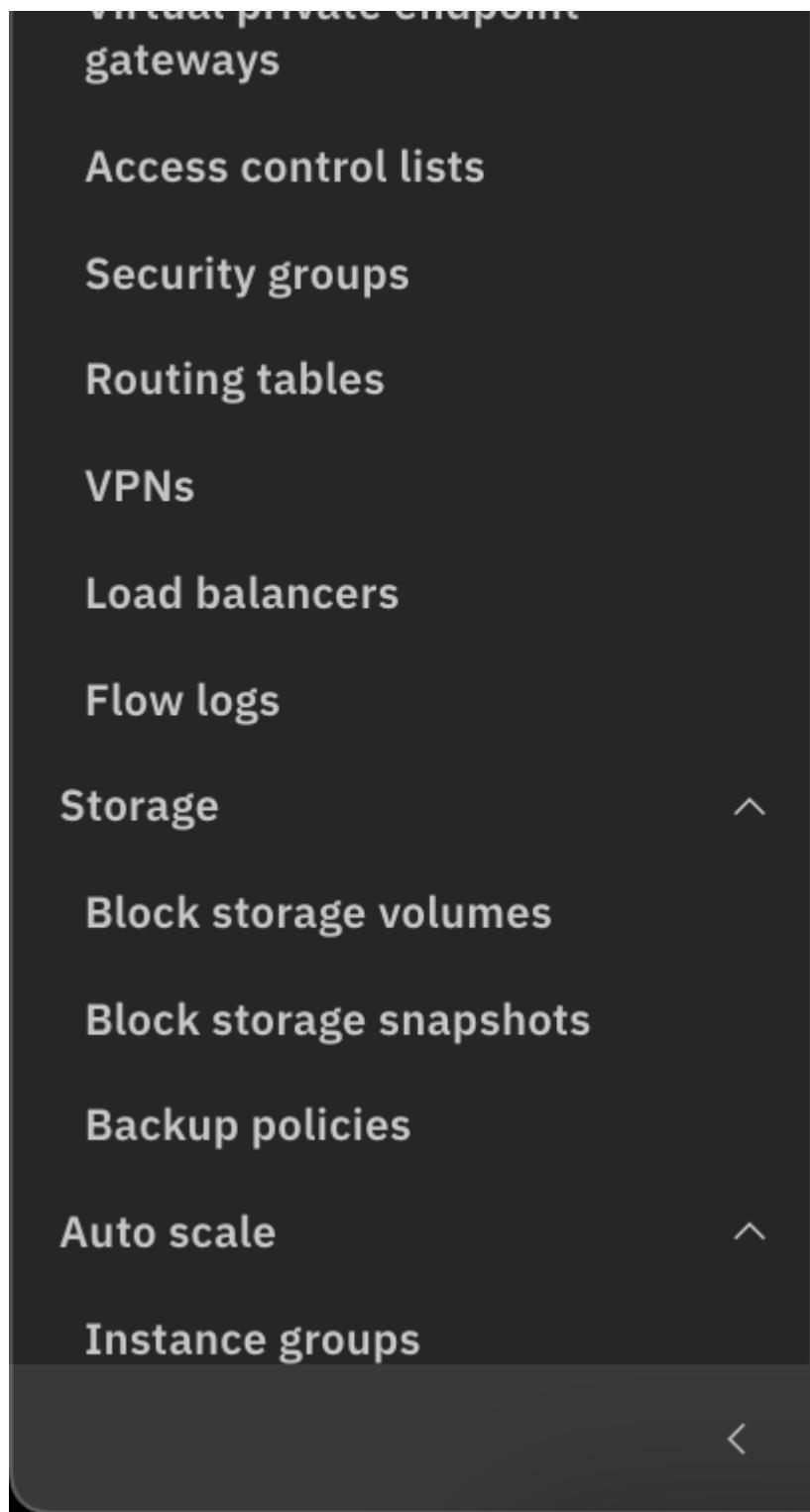
15. Para que seja possível disponibilizar sua aplicação na internet é preciso liberar a porta 80 e 443. Para isso acesse sua vm clicando em "Virtual server instances" do lado esquerdo.



The image shows a mobile view of the IBM Cloud VPC Infrastructure menu. At the top, there's a navigation bar with three horizontal lines on the left and the text "IBM Cloud" in the center. Below this is a header section with a network icon and the text "VPC Infrastructure". The main content area contains several menu items, each preceded by a small upward arrow icon:

- Getting started
- Overview
- VPC layout
- Compute
  - Virtual server instances
  - Bare metal servers
  - Dedicated hosts
  - Placement groups
  - SSH keys
  - Custom images
- Network
  - VPCs
  - Subnets
  - Floating IPs
  - Public gateways
  - Virtual private endpoint

A red rectangular box highlights the "Virtual server instances" item under the Compute section. A blue vertical bar is visible on the far left edge of the screen.



16. Clique no nome da VM para acessar suas configurações.

Name	Status	Resource group	Virtual Private Cloud	Profile	Reserved IP	Floating IP
php	Running	Sandbox Brazil	automation-system-vpc	b2x2-2x8	10.240.64.15	52.118.206.177

17. Na parte inferior da página, existe uma seção para configurar a interface de rede. Clique no grupo de segurança.

Interface	Subnet	Reserved IP	Floating IP	Security groups	Allow IP spoofing
eth0	sn-20211124-02	10.240.64.15	52.118.206.177	frequency-peclin-implosion-fondue	Disabled

18. Precisamos criar uma regra nova habilitar a exposição da vm a internet. Para isso clique em "Rules".

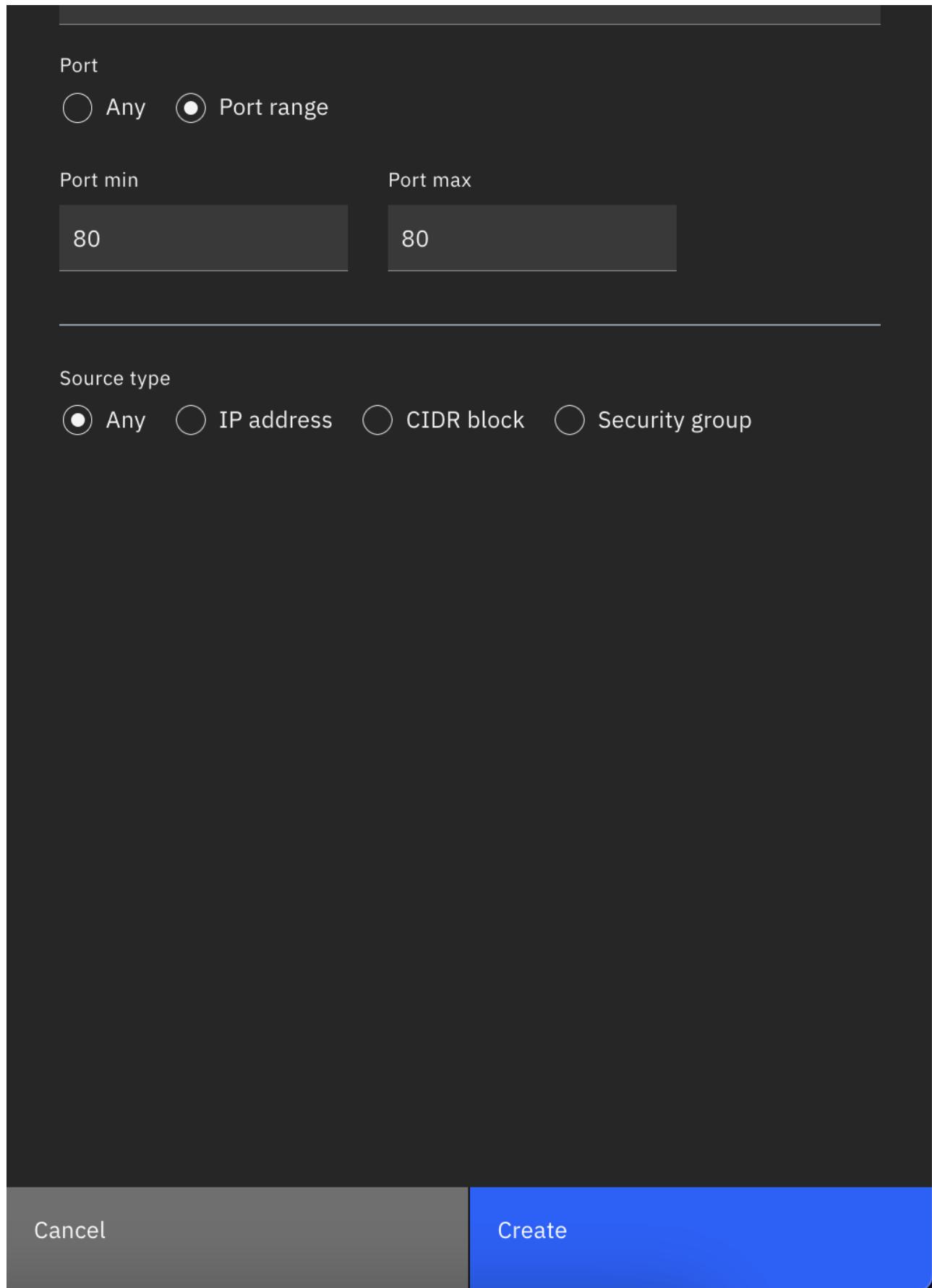
The screenshot shows the IBM Cloud interface for managing security groups. The left sidebar is collapsed. The main area displays the 'frequency-pectin-implosion-fondue' security group details. The 'Rules' tab is selected. In the 'Inbound rules' section, there are five rules listed. At the top right of this section, there is a 'Create' button with a plus sign, which is highlighted with a red box.

19. Em Rules, clique no botão "Create" na seção "Inbound rules"

This screenshot is identical to the previous one, showing the 'frequency-pectin-implosion-fondue' security group page. The 'Rules' tab is selected, and the 'Inbound rules' section is visible. The 'Create' button at the top right of the 'Inbound rules' table is highlighted with a red box.

20. Mantenha o protocolo como TCP, e adicione a porta 80. Em seguinda clique em "Create".

A modal dialog box titled 'Create inbound rule' is shown. The 'Protocol' dropdown menu is open, and 'TCP' is selected. At the bottom right of the dialog, there is a 'Create' button with a plus sign, which is highlighted with a red box.



21. Repita o passo 20 caso queria liberar a porta 443.

Ao final dessa etapa voce dever ter em mãos a chave privada ssh e o IP de acesso da VM, com isso está pronto para seguir em frente.

## Acesso externo ao Banco de Dados

A IBM Cloud oferece alguns bancos de dados em sua nuvem, um deles é o [MySQL](#). Nele é possível customizar algumas features para de acordo com a necessidade. Com o produto instanciado é será possível acessá-lo através do seu link, com as credenciais de usuário e a porta correta fornecida na aba de credenciais do produto.

1. Para liberar a conexão com o Banco de Dados externo é necessário criar uma regra no grupo de segurança que permita isso. Na sua VPC acesso o grupo de segurança o qual pertence sua máquina.

Interface	Subnet	Reserved IP	Floating IP	Security groups	Allow IP spoofing
eth0	sn-20211124-02	10.240.64.15	52.118.206.177	frequency-peclin-implosion-fondue	Disabled

2. Precisamos criar uma regra que habilite o acesso externo, clique em "Rules".

Interface	Virtual server	Subnet
eth0	php	sn-20211124-02

3. Aqui vamos precisar criar uma regra para inbound, clique no botão "Create" na seção "Inbound rules"

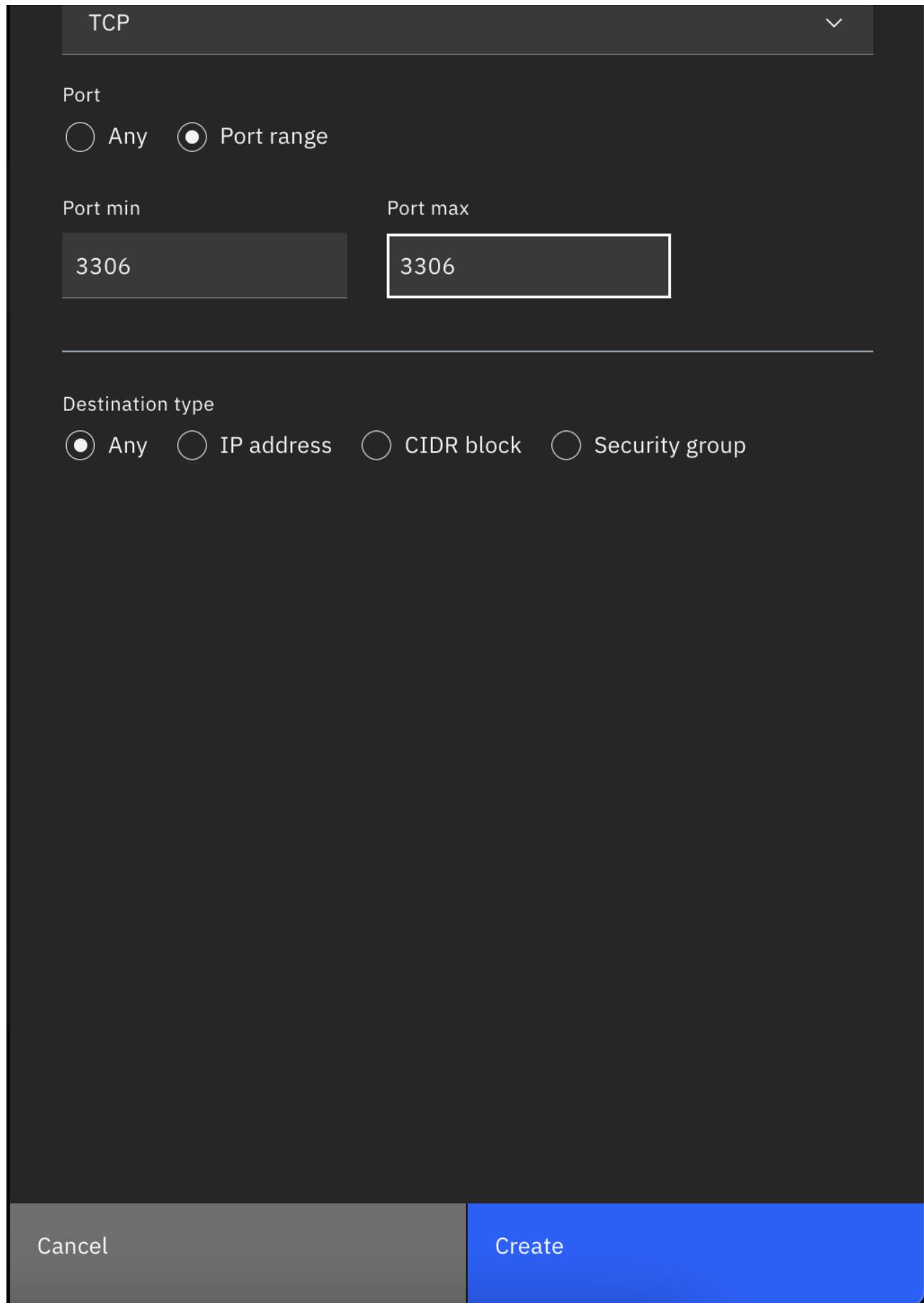
The screenshot shows the IBM Cloud interface for managing security groups. The left sidebar is collapsed. The main area displays the 'frequency-peptic-implosion-fondue' security group. Under the 'Inbound rules' section, there is a table with columns: Protocol, Source type, Source, and Value. Several rows are listed, including entries for ALL, TCP, ICMP, and multiple TCP entries with specific port ranges. At the top right of this table, there is a blue 'Create' button with a '+' icon, which is highlighted with a red box.

4. Tambem é necessário criar uma regra de outbound, clique no botão "Create" na seção "Outbound rules"

This screenshot shows the same IBM Cloud interface, but the focus is on the 'Outbound rules' section. The table has columns: Protocol, Destination type, Destination, and Value. It lists several rules for ICMP, TCP, and UDP protocols to various IP addresses. Similar to the inbound rules, a blue 'Create' button with a '+' icon is located at the top right of the table, highlighted with a red box.

5. Selecione o protocolo e estabelece a porta que deve ser exposta.

A modal dialog box titled 'Create outbound rule' is shown. At the top left, it displays the session ID '2283996 - IBM Client Engineerin...'. The top right features several icons: a question mark, a magnifying glass, a clipboard, a calculator, a bell with a red dot, and a user profile. At the bottom left of the modal, the word 'Protocol' is visible. The bottom right corner contains a large 'X' symbol for closing the dialog.

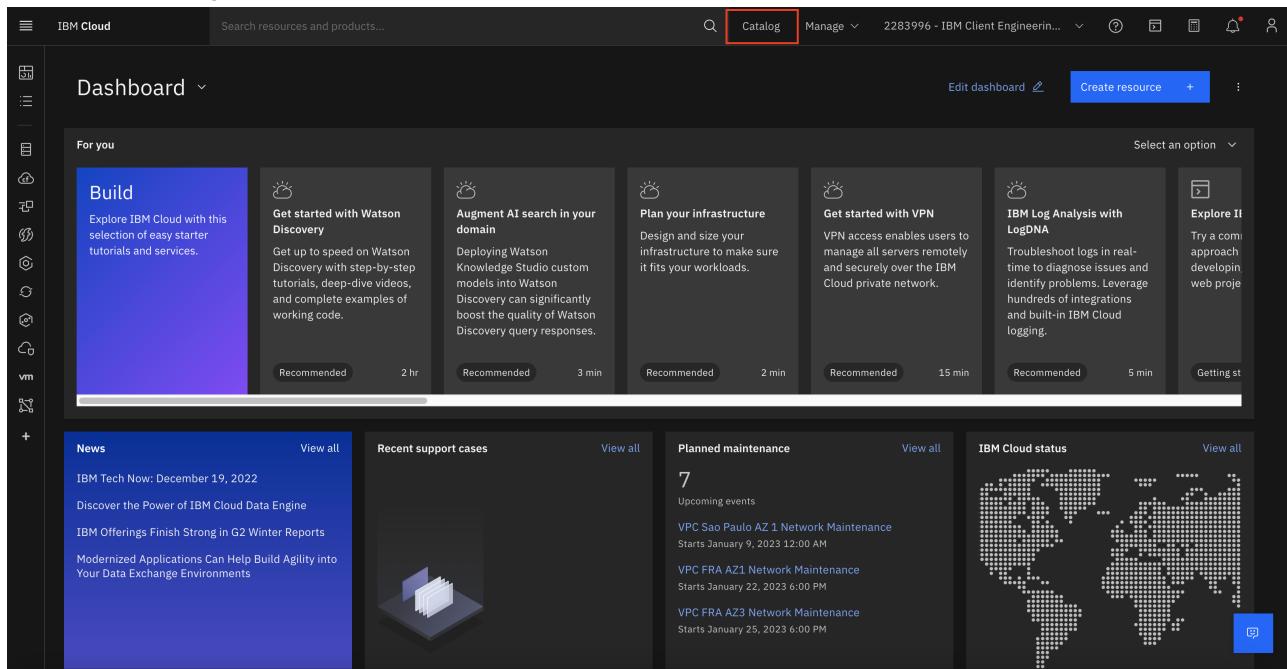


Dessa será possível habilitar o acesso externo da VM ao banco de dados na nuvem.

DNS na IBM Cloud

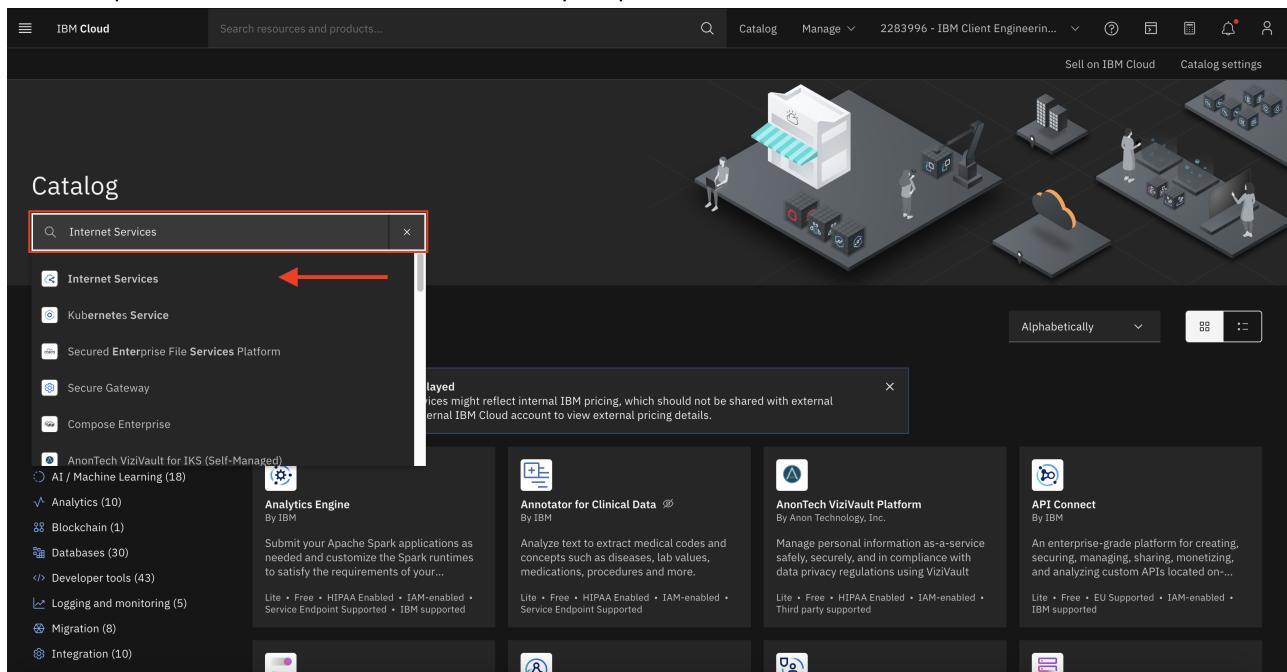
Para fazer uso dos diversos endereços DNS, vamos precisar instanciar o serviço **Internet Services** na IBM Cloud, então acesse sua conta em <https://cloud.ibm.com/> e siga os passos abaixo:

## 1. Acesse o catálogo da IBM Cloud.



The screenshot shows the IBM Cloud Dashboard. At the top, there's a navigation bar with 'IBM Cloud' and a search bar. Below it is a 'Catalog' tab, which is highlighted with a red box. To the right of the catalog tab are 'Manage', '2283996 - IBM Client Engineerin...', and other account-related icons. The main area is a 'Dashboard' section with various cards. One card for 'Build' is recommended and takes up most of the space. Other cards include 'Get started with Watson Discovery', 'Augment AI search in your domain', 'Plan your infrastructure', 'Get started with VPN', 'IBM Log Analysis with LogDNA', and 'Explore II'. Below the dashboard are sections for 'News', 'Recent support cases', 'Planned maintenance', and 'IBM Cloud status' (a world map).

## 2. Procure por "Internet Services" na barra de pesquisa.



The screenshot shows the IBM Cloud Catalog page. At the top, there's a search bar with 'Internet Services' typed in, which is also highlighted with a red box. Below the search bar is a sidebar with a tree view of categories like 'Internet Services' (which is expanded and highlighted with a red arrow), 'Kubernetes Service', 'Secured Enterprise File Services Platform', 'Secure Gateway', 'Compose Enterprise', 'Analytics', 'Blockchain', 'Databases', 'Developer tools', 'Logging and monitoring', 'Migration', and 'Integration'. The main area displays several service cards. One card for 'Analytics Engine' is visible, along with others for 'Annotator for Clinical Data', 'AnonTech ViziVault Platform', and 'API Connect'. A tooltip for the 'Analytics Engine' card states: 'Submit your Apache Spark applications as needed and customize the Spark runtimes to satisfy the requirements of your...'. Another tooltip for the 'Annotator for Clinical Data' card states: 'Analyze text to extract medical codes and concepts such as diseases, lab values, medications, procedures and more.' The bottom of the screen shows a footer with 'Alphabetically' and other navigation icons.

### 3. Aceite os termos e crie o serviço.

The screenshot shows the IBM Cloud Catalog interface. On the left, there's a sidebar with filters: Type (Infrastructure), Provider (IBM), Last updated (12/14/2022), Category (Networking), Compliance (IAM-enabled), Location (Global), and Related links (API docs, Docs, Terms). The main area displays the "Internet Services" catalog entry, which includes a brief description of Cloudflare's services and a "Create" button. Below this, a table compares two plans: "Free Trial" and "Standard". The "Free Trial" plan is highlighted. The "Create" button is located at the bottom right of the page, with a red arrow pointing to it from the right side of the interface.

### 4. Adicione o seu domínio DNS.

The screenshot shows the "Test" resource overview page. The left sidebar has tabs: Getting started, Account, Overview (which is selected and highlighted in blue), Metrics, Security, Reliability, Performance, Edge Functions, and Plan. The main content area features a "Start by adding a domain." section with a "Add domain" button, which is highlighted with a red box. Below this, there's a link to "Learn more about Internet Services." The top right of the page has "Details" and "Actions..." buttons, and the bottom right has a blue "Edit" icon.

## 5. Escolha o endereço do domínio e prossiga.

The screenshot shows two overlapping windows. On the left is the 'Test' service's Overview page, which includes sections for Getting started, Account, Overview (highlighted), Metrics, Security, Reliability, Performance, Edge Functions, and Plan. A central panel displays a 3D cube icon and text about connecting a domain to Cloud Internet Services (CIS) for improved response speeds and security. Below this is a large blue 'Add domain' button and a link to learn more about Internet Services. On the right is a 'Setup your domain' dialog box with tabs for Connect domain, DNS records, and Domain management. The 'Connect domain' tab is active, showing a 'Domain name' input field containing 'www.test.com' (which is highlighted with a red box). Below it is a smaller 'example.com' placeholder. At the bottom of the dialog are 'Cancel' and 'Next' buttons.

## 6. Para criar registros de subdomínios que apontam para os endereços que irá utilizar, acesse a aba Reliability.

This screenshot shows the same 'Test' service Overview page as the previous one, but with the 'Reliability' tab (highlighted with a red box) selected instead of 'Overview'. The main content area displays the 'Details' section, which includes the 'Plan' (Trial: 30 days remaining), 'Domain ID' (ff37be50fb5a96338760747a338f1cda), 'CRN' (crn:v1:bluemix:public:internet-svcs:global:a/c409b38a21f14c33a0d1964b06334bbd:d56e8172-771a-4cca-8507-14b323d8fbce:), and 'Name servers' (ns122.name.cloud.ibm.com and ns169.name.cloud.ibm.com). Below these are 'Service modes' (Defense mode and Development mode, both inactive) and an 'Events' section. The top navigation bar includes 'IBM Cloud', a search bar, 'Catalog', 'Manage', and 'Gabriel Pires's Account'.

## 7. Entre na aba de DNS.

The screenshot shows the IBM Cloud dashboard for a service named 'Test'. The left sidebar has a 'Reliability' section selected. The main content area is titled 'Global load balancers' and contains a sub-section 'Load balancers'. A red box highlights the 'DNS' tab in the top navigation bar of this section. Below the tabs, there's a table header with columns: Health, Hostname, Available pools, TTL, Proxy, and Enabled. A blue box highlights the 'Create' button at the top right of the table area.

## 8. Desça na página até DNS Records e clique em Adicionar.

The screenshot shows the 'DNS records' page. At the top, there's a note: 'application names may cause conflicts with your DNS resolution.' Below it, a 'View lists' button is shown. The main section is titled 'DNS records' and contains a note: 'Manage your DNS records and control whether to proxy traffic through Cloud Internet Services. Proxied traffic can leverage WAF, DDoS, and Caching capabilities.' A callout box points to a 'Domain pending NS record' message: 'Your domain is pending NS record updates. Ensure your DNS records are properly configured, then return to the Overview page to finish configuring your domain. [Go to Overview](#)'. A red box highlights the 'Add' button at the top right of the table area.

## 9. Defina as configurações básicas, escolha o nome do subdomínio e informe seu IPv4.

The screenshot shows the 'Add record' dialog box. The title is 'Add record' with a close button 'X' in the top right. It has two main input fields: 'Type' and 'TTL'. The 'Type' field is set to 'A' and has a dropdown arrow. The 'TTL' field is set to 'Automatic' and also has a dropdown arrow. The background of the dialog is dark gray.

Name

WWW

Reusing a name that is used by an enabled load balancer or range application may cause a conflict.

IPv4 address

145.229.30.65

Close

Add

## Acessando a VM

É necessário criar o arquivo da credencial (cred.pem) dentro de uma pasta em sua máquina, após isso é preciso acessá-la através do terminal e utilizar o comando

```
ssh -i cred.pem root@<IPdaVM>
```

Caso tenha problemas de permissão para acessar a VM, utilize o comando

```
chmod 700 cred.pem para resolver
```

## Instalando o Apache

Primeiro precisamos instalar o Apache Web Server usando o comando

```
apt install apache2
```

Caso esteja utilizando algum firewall, é necessário utilizar o comando

```
ufw allow "Apache Full"
```

Ele irá fornecer as permissões necessárias.

Após isso, para verificar o status do Apache e garantir que ele está funcionando, execute o comando

```
systemctl status apache2
```

E então acesse o IP de sua VM pelo browser para ver a página criada pelo Apache.

## Instalando o PHP

Para instalar o PHP utilize o comando

```
apt install php libapache2-mod-php php-mbstring php-xmlrpc php-soap php-gd  
php-xml php-cli php-zip php-bcmath php-tokenizer php-json php-pear
```

e atualize seu sistema com

```
apt-get update
```

## Configurando o ambiente

Para criar um arquivo na raiz pasta do Apache, utilize o comando

```
nano /var/www/html/<nomeDoArquivo>
```

e adicione ao arquivo as seguintes linhas

```
<?php  
    phpinfo();  
?>
```

Também vamos precisar do banco de dados, para isso basta o comando

```
apt install mariadb-server
```

Instale o Composer que será utilizado para criar o projeto Laravel com o comando

```
curl -sS https://getcomposer.org/installer | php
```

e após isso, para garantir que o Composer será utilizado globalmente, execute os seguintes comandos

```
mv composer.phar /usr/local/bin/composer
```

```
chmod +x /usr/local/bin/composer
```

Agora iremos criar o projeto, vamos executar uma série de comandos

- Para criar o projeto

```
composer create-project --prefer-dist laravel/laravel <NomeDoProjeto>
```

- Para acessar o diretório do projeto

```
cd <NomeDoProjeto>
```

- Para configurar o ambiente

```
php artisan serve --host=<SeuIP> --port=<PortaDesejada>
```

Para rodar o projeto precisamos move-lo para a pasta raiz do Apache com o comando

```
mv <NomeDoProjeto> /var/www/html/
```

Também vamos fornecer as permissões necessárias para que o projeto funcione corretamente com os comandos

```
chgrp -R www-data /var/www/html/<NomeDoProjeto>/
```

```
chmod -R 775 /var/www/html/<NomeDoProjeto>/storage
```

Agora para criar um host virtual, vamos acessar o diretório correto com

```
cd /etc/apache2/sites-available
```

Em seguida o comando para criar o arquivo de configuração

```
nano laravel_project.conf
```

e adicione o seguinte código à ele

```
<VirtualHost *:80>
    ServerName thedomain.com
    ServerAdmin webmaster@thedomain.com
    DocumentRoot /var/www/html/example/public

    <Directory /var/www/html/example>
        AllowOverride All
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Após isso desative a configuração padrão do host com

```
a2dissite 000-default.conf
```

Ative nosso host virtual com

```
a2ensite laravel_project
```

Por fim, ative o modulo de reescrita com

```
a2enmod rewrite
```

e reinicie o Apache usando

```
systemctl restart apache2
```

## Criando o projeto

Agora que temos todo o ambiente criado e configurado, vamos criar um modulo de exemplo para ver melhor o funcionamento do Laravel e testar a conexão com o banco.

Primeiro vamos criar um model chamado **Company**, acesse o diretório do projeto com

```
cd /var/www/html/<NomeDoProjeto>/
```

Agora execute

```
php artisan make:model Company -m
```

Em seguida acesse o diretório

```
database/migrations/
```

e adicione ao arquivo de migration de **Company** o seguinte código

```
public function up() {  
    Schema::create('companies', function (Blueprint $table) {  
        $table->id();  
        $table->string('name');
```

```
$table->string('email');

$table->string('address');

$table->timestamps();

});

}
```

Depois acesse a pasta app/Models e adicione o seguinte código ao arquivo **Company.php**

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Company extends Model {

    use HasFactory; protected $fillable = ['name', 'email', 'address'];

}

?>
```

Acesse a raiz do projeto e execute o comando abaixo para criar as tabelas no banco de dados

```
php artisan migrate
```

Vamos então, criar o Controller para esse modelo, utilizaremos o comando

```
php artisan make:controller CompanyController
```

Depois disso acesse o repositório

```
app/Http/Controllers
```

e adicione o seguinte código ao arquivo **CompanyController.php**

```
<?php

namespace App\Http\Controllers;

use App\Models\Company;

use Illuminate\Http\Request;

class CompanyController extends Controller

{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $companies = Company::orderBy('id', 'desc')->paginate(5);

        return view('companies.index', compact('companies'));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('companies.create');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([
            'name' => 'required',
            'email' => 'required',
            'address' => 'required',
        ]);
    }
}
```

```
]);  
  
    Company::create($request->post());  
  
    return redirect()->route('companies.index')-  
>with('success','Company has been created successfully.');//  
}  
  
/**  
 * Display the specified resource.  
 *  
 * @param \App\company $company  
 * @return \Illuminate\Http\Response  
 */  
  
public function show(Company $company)  
{  
    return view('companies.show',compact('company'));  
}  
  
/**  
 * Show the form for editing the specified resource.  
 *  
 * @param \App\Company $company  
 * @return \Illuminate\Http\Response  
 */  
  
public function edit(Company $company)  
{  
    return view('companies.edit',compact('company'));  
}  
  
/**  
 * Update the specified resource in storage.  
 *  
 * @param \Illuminate\Http\Request $request  
 * @param \App\company $company  
 * @return \Illuminate\Http\Response  
 */  
  
public function update(Request $request, Company $company)  
{  
    $request->validate([  
  
        'name' => 'required',  
  
        'email' => 'required',  
  
        'address' => 'required',  
    ]);  
  
    $company->fill($request->post())->save();  
  
    return redirect()->route('companies.index')-
```

```

>with('success','Company Has Been updated successfully');
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Company $company
 * @return \Illuminate\Http\Response
 */

public function destroy(Company $company)
{
    $company->delete();

    return redirect()->route('companies.index')-
>with('success','Company has been deleted successfully');
}
}

```

Vamos então criar as rotas necessárias para acessar esse controller, acesse o diretório **routes** e adicione o código abaixo ao arquivo **web.php**

```

use App\Http\Controllers\CompanyController;

Route::resource('companies', CompanyController::class);

```

.env

É preciso ter certeza que o arquivo **.env** está configurado corretamente para a conexão com o banco de dados

```

DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=<PortaSQL>
DB_DATABASE=<NomeBanco>
DB_USERNAME=<UsuarioDoBanco>
DB_PASSWORD=<SenhaDoUsuario>

```

## Criando as telas

Agora, iremos criar as telas para acessar através dessa rota, primeiro acesse o diretório **resources/views** e crie uma nova pasta chamada **companies**, dentro delas vamos precisar de 3 telas:

- index.blade.php

```

<!DOCTYPE html>
<html lang="en">

```

```
<head>

    <meta charset="UTF-8">

    <title>Laravel 9 CRUD Tutorial Example</title>
    <link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
    .min.css" >

</head>
<body>
    <div class="container mt-2">
        <div class="row">
            <div class="col-lg-12 margin-tb">
                <div class="pull-left">

                    <h2>Laravel 9 CRUD Example Tutorial</h2>

                </div>
                <div class="pull-right mb-2">

                    <a class="btn btn-success" href="{{
                    route('companies.create') }}> Create Company</a>

                </div>
            </div>
        </div>

        @if ($message = Session::get('success'))
            <div class="alert alert-success">
                <p>{{ $message }}</p>
            </div>
        @endif

        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>S.No</th>

                    <th>Company Name</th>

                    <th>Company Email</th>

                    <th>Company Address</th>

                    <th width="280px">Action</th>
                </tr>
            </thead>
            <tbody>
                @foreach ($companies as $company)
                    <tr>
                        <td>{{ $company->id }}</td>

                        <td>{{ $company->name }}</td>
```

```

        <td>{{ $company->email }}</td>

        <td>
            <form action="{{ route('companies.destroy', $company->id) }}" method="Post">

                <a class="btn btn-primary" href="{{ route('companies.edit', $company->id) }}>Edit</a>

                @csrf

                @method('DELETE')

                <button type="submit" class="btn btn-danger">Delete</button>
            </form>
        </td>
    </tr>
    @endforeach
</tbody>
</table>

{!! $companies->links() !!}

</div>
</body>
</html>

```

- `create.blade.php`

```

<!DOCTYPE html>
<html lang="en">
    <head>

        <meta charset="UTF-8">
        <title>Add Company Form – Laravel 9 CRUD</title>
        <link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    </head>
    <body>
        <div class="container mt-2">
            <div class="row">
                <div class="col-lg-12 margin-tb">
                    <div class="pull-left mb-2">

                        <h2>Add Company</h2>

```

```
</div>
<div class="pull-right">

    <a class="btn btn-primary" href="{{
route('companies.index') }}> Back</a>

    </div>
</div>
</div>

@if(session('status'))

<div class="alert alert-success mb-1 mt-1">

{{ session('status') }}

</div>

@endif

<form action="{{ route('companies.store') }}" method="POST"
enctype="multipart/form-data">
    @csrf
    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">

                <strong>Company Name:</strong>

                <input type="text" name="name" class="form-
control" placeholder="Company Name">

                @error('name')

                    <div class="alert alert-danger mt-1 mb-1">{{
$message }}</div>

                @enderror

            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">

                <strong>Company Email:</strong>

                <input type="email" name="email" class="form-
control" placeholder="Company Email">

                @error('email')

                    <div class="alert alert-danger mt-1 mb-1">{{
$message }}</div>

                @enderror

            </div>
        </div>
    </div>
</form>
```

```
        @enderror

            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">

                <strong>Company Address:</strong>

                <input type="text" name="address" class="form-control" placeholder="Company Address">

                @error('address')

                    <div class="alert alert-danger mt-1 mb-1">{{ $message }}</div>

                @enderror
            </div>
        </div>

        <button type="submit" class="btn btn-primary ml-3">Submit</button>

    </div>
</form>
</div>
</body>
</html>
```

- edit.blade.php

```
<!DOCTYPE html>
<html lang="en">
    <head>

        <meta charset="UTF-8">
        <title>Edit Company Form - Laravel 9 CRUD Tutorial</title>
        <link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    </head>
    <body>
        <div class="container mt-2">
            <div class="row">
                <div class="col-lg-12 margin-tb">
                    <div class="pull-left">

                        <h2>Edit Company</h2>

                    </div>
```

```
<div class="pull-right">

    <a class="btn btn-primary" href="{{ route('companies.index') }}" enctype="multipart/form-data">
        Back</a>
    </div>
</div>

@if(session('status'))

<div class="alert alert-success mb-1 mt-1">
    {{ session('status') }}
</div>

@endif

<form action="{{ route('companies.update', $company->id) }}" method="POST" enctype="multipart/form-data">
    @csrf
    @method('PUT')

    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">

                <strong>Company Name:</strong>
                <input type="text" name="name" value="{{ $company->name }}" class="form-control"
                    placeholder="Company name">
                @error('name')
                    <div class="alert alert-danger mt-1 mb-1">{{ $message }}</div>
                @enderror

            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">

                <strong>Company Email:</strong>
                <input type="email" name="email" class="form-control" placeholder="Company Email">
            </div>
        </div>
    </div>
</form>
```

```
        value="{{ $company->email }}>

        @error('email')

            <div class="alert alert-danger mt-1 mb-1">{{
                $message }}</div>

        @enderror

    </div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">

        <strong>Company Address:</strong>

        <input type="text" name="address" value="{{
            $company->address }}>" class="form-control"

            placeholder="Company Address">

        @error('address')

            <div class="alert alert-danger mt-1 mb-1">{{
                $message }}</div>

        @enderror

    </div>
</div>

    <button type="submit" class="btn btn-primary ml-3">Submit</button>

</div>
</form>
</div>
</body>
</html>
```

Finalmente, para executar seu projeto, utilize o comando

```
php artisan serve
```

e acesse via URL

```
http://<SeuIP>/companies
```

## Referências

- Criação de Frontend publico e backend privado
- "Como Instalar Laravel no Ubuntu"
- CRUD no Laravel

## Autores

- Gabriel Felisberto Pires
- João Pedro Poloni Ponce

## Licença



Este obra está licenciado com uma Licença Creative Commons Atribuição 4.0 Internacional.