



Linux Laravel

Virtual Machine

- [Introdução](#)
- [Passo a Passo](#)
 - [Chave SSH na IBM Cloud](#)
 - [Acesso externo ao Banco de Dados](#)
 - [DNS na IBM Cloud](#)
 - [Acessando a VM](#)
 - [Instalando o Apache](#)
 - [Instalando o PHP](#)
 - [Configurando o ambiente](#)
 - [Criando o projeto](#)
 - [.env](#)
 - [Criando as telas](#)
- [Referências](#)
- [Autores](#)
- [Licença](#)

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.

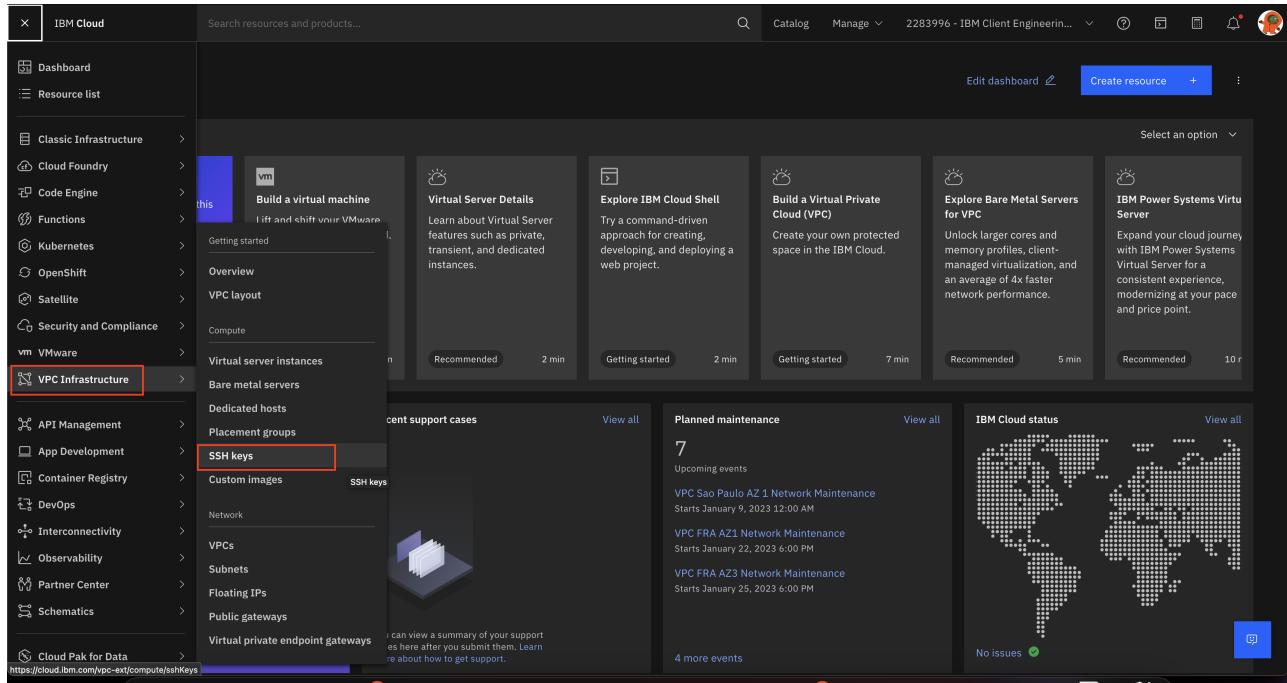
Passo a Passo

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.

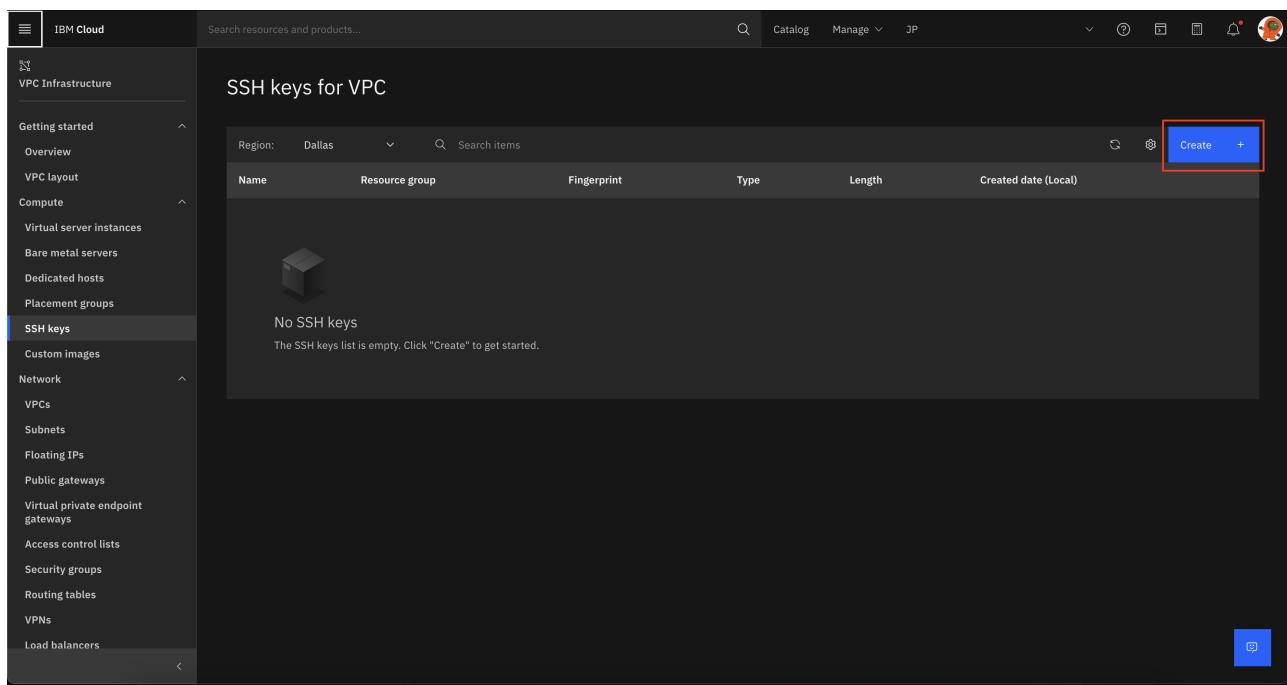
De posse das chaves acesse a IBM Cloud: <https://cloud.ibm.com/>

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



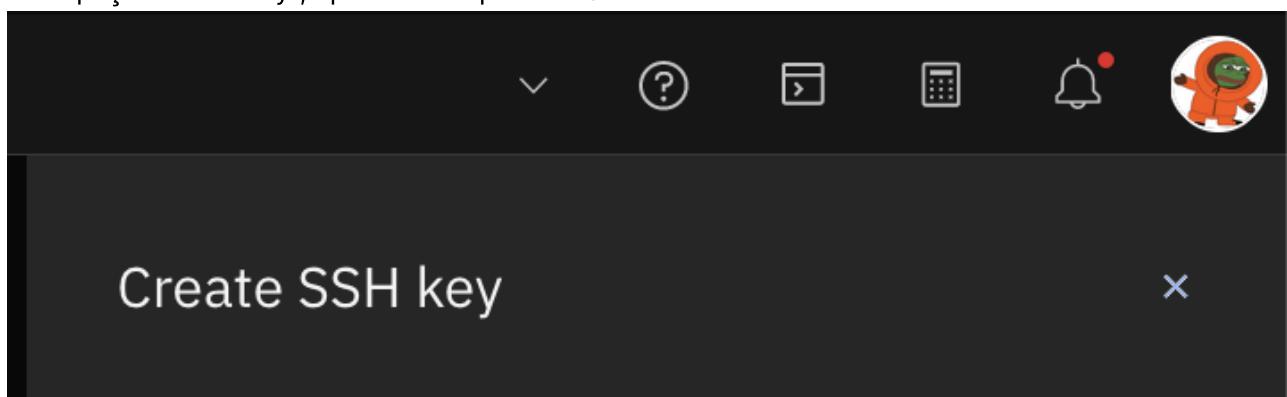
The screenshot shows the IBM Cloud dashboard. On the left, there is a navigation sidebar with various service categories. Under the 'Compute' section, 'VPC Infrastructure' is highlighted with a red box. Below it, under 'Compute', is a sub-menu for 'SSH keys', which is also highlighted with a red box. The main content area displays several cards related to VPCs and other cloud services, along with a world map showing 'No issues'.

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 'SSH keys for VPC' page. The left sidebar has 'SSH keys' selected. The main area shows a table header with columns: Name, Resource group, Fingerprint, Type, Length, and Created date (Local). A large blue 'Create' button is located at the top right of the table area, highlighted with a red box. Below the table, a message says 'No SSH keys' and 'The SSH keys list is empty. Click "Create" to get started.'

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 a modal dialog titled 'Create SSH key'. At the top, there is a toolbar with icons for back, help, forward, calculator, and notifications. Below the toolbar, the title 'Create SSH key' is displayed. The main area of the dialog is currently empty, showing a dark background.

Create an SSH key that you'll use to access your virtual server instance. [Learn more](#)

Location

Geography

North America



Region

Dallas



Details

Name

Enter key name

Resource group [\(i\)](#)

Default



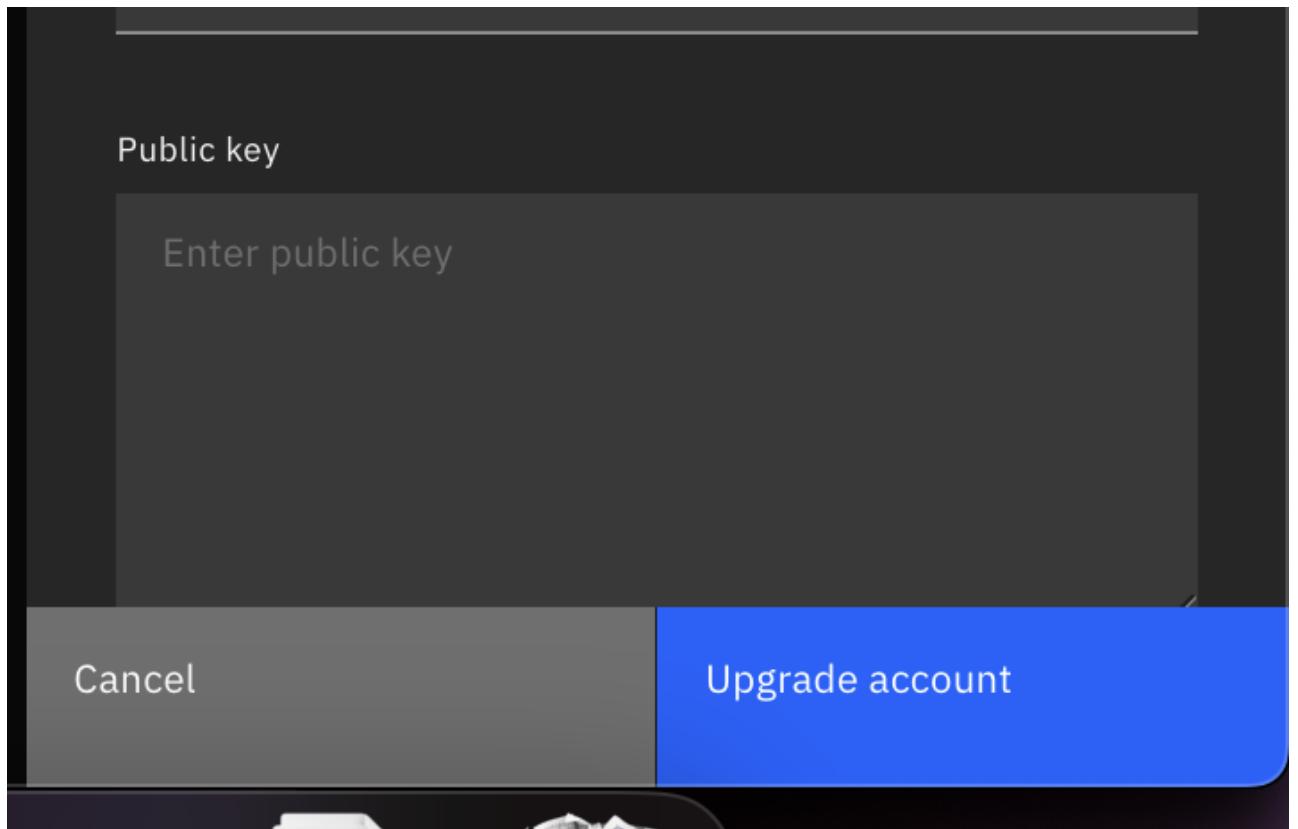
[View all resource groups](#)

Tags [\(i\)](#)

Examples: env:dev, version-1

Access management tags [\(i\)](#)

Examples: access:dev, proj:version-1



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.

A screenshot of the IBM Cloud VPC Infrastructure interface. On the left, there is a navigation sidebar with various options like "Getting started", "Compute", "Network", and "VPCs". The "VPCs" link is highlighted with a red box. The main content area is titled "SSH keys for VPC" and shows a message "No SSH keys" with the sub-instruction "The SSH keys list is empty. Click "Create" to get started." There is a "Create" button at the top right of the table header.

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

The screenshot shows the IBM Cloud interface for VPC Infrastructure. On the left, there's a sidebar with various options like Overview, VPC layout, Compute, Virtual server instances, and Network. Under Network, the 'VPCs' option is selected and highlighted with a blue border. The main area is titled 'Virtual private clouds' and shows a message: 'No virtual private clouds' and 'The virtual private clouds list is empty. Click "Create" to get started.' In the top right corner of the main area, there's a red box highlighting the 'Create' button.

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

The screenshot shows the 'Create' dialog for a new Virtual Private Cloud. The 'Location' section includes 'Geography' (set to North America) and 'Region' (set to Dallas). The 'Details' section includes 'Name' (input field 'Enter a unique name'), 'Resource group' (dropdown set to 'default'), and 'Tags' (input field 'Examples: env=dev, version=3'). Below these are sections for 'Access management tags' and 'VPC default access control list'. The 'Subnets' section lists three existing subnets: 'sn-20221226-01' (Zone Dallas 1, IP range 10.240.0.0/18), 'sn-20221226-02' (Zone Dallas 2, IP range 10.240.64.0/18), and 'sn-20221226-03' (Zone Dallas 3, IP range 10.240.128.0/18). On the right side, there's a summary panel showing 'Virtual private cloud' and 'provided' status, along with buttons for 'Apply a code', 'Total estimated cost \$0.00/mo', and 'Add to estimate'.

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"

Virtual private clouds

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-devolve-demotion	audacious-freebee-elated-crawling
na-cop-vpc	Available	tg-refinitiv-us	3	trunks-definite-leverage-culotte	fencing-unsafe-tabloid-strength
vpc-ip	Available	default	3	footboard-dividing-purifier-elusive	spoils-heroism-autumn-marsupial
telusmecvpc	Available	tg-telus-communications	3	unmoved-essence-habitual-platform	defiant-headlock-false-legwarmer
lumen2	Available	ce-lumen2	3	trio-coastine-pushchair-unhappily	life-nullify-wrangle-itinerary
vpc-lumen2	Available	ce-lumen2	3	decaf-refinery-backyard-tantrum	hardener-getaway-citadel-rectangle

Items per page: 10 | 1-10 of 10 items | 1 of 1 page | < >

What do you want to do next?
Since you already created a virtual private cloud, you can add other services.

<https://cloud.ibm.com/vpc-ext/compute/vs>

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

Virtual server instances for VPC

Name	Status	Resource group	Virtual Private Cloud	Profile	Reserved IP ⓘ	Floating IP
No virtual server instances The virtual server instances list is empty. Click "Create" to get started.						

What do you want to do next?
Learn how to connect to your new instance or add extra services.

10. Configure a máquina de acordo com sua necessidade. Aqui o importante é associar a essa máquina 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.

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.

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

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 IBM Cloud interface for managing VPC infrastructure. On the left, there's a navigation sidebar with categories like Overview, VPC layout, Compute, Network, and Floating IPs. The Floating IPs section is currently selected. The main area displays a table of floating IP details, including Name, Status, Address, Location, Targeted device, and Target type. A search bar at the top right allows filtering by address. To the right of the table is a detailed panel for creating a new floating IP. It includes fields for Floating IP name, Resource group (set to Default), Tags (with examples like env.dev.version-1), and a checkbox for Select instance or server which has "Virtual server instance" and "php" selected. Below these fields are sections for "No available network interface" and "Get sample API call". At the bottom right of the panel are "Total estimated cost" (\$1.00/mo) and "Reserve" buttons.

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 screenshot shows the IBM Cloud VPC Infrastructure homepage. It features a navigation bar with icons for VPC, Compute, Network, and Storage. Below the navigation is a main content area with sections for Getting started, Overview, VPC layout, Compute, and Bare metal servers. The "Virtual server instances" link under the Compute section is highlighted with a red box. There are also sections for Dedicated hosts and Floating IPs.

Dedicated hosts

Placement groups

SSH keys

Custom images

Network



VPCs

Subnets

Floating IPs

Public gateways

**Virtual private endpoint
gateways**

Access control lists

Security groups

Routing tables

VPNs

Load balancers

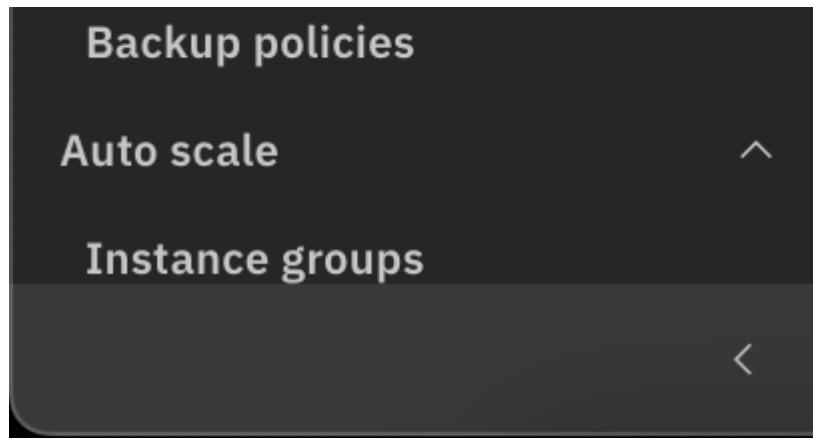
Flow logs

Storage



Block storage volumes

Block storage snapshots



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	bx2-2x8	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".

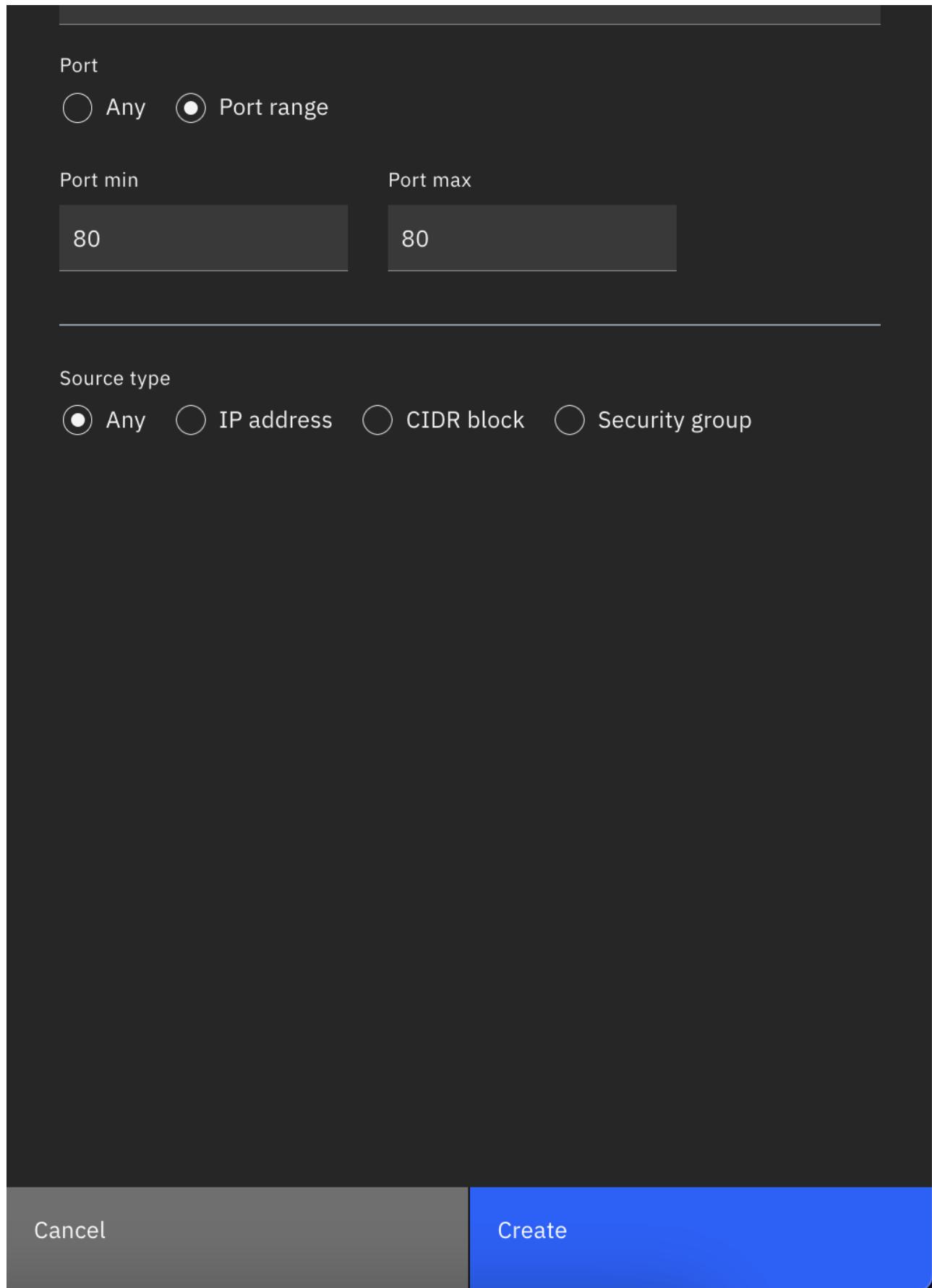
The screenshot shows the IBM Cloud interface for managing security groups. The left sidebar is collapsed. The main area displays the 'frequency-pectin-implosion-fondue' security group details. The 'Rules' tab is selected. In the 'Inbound rules' section, there are five rules listed. At the top right of 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.

The screenshot shows the IBM Cloud VPC Infrastructure interface. The left sidebar is collapsed. The main area displays the 'Network interfaces' section. A specific security group, 'frequency-pectin-implosion-fondue', is selected and its details are visible in the center panel. The 'Rules' tab is highlighted with a red box. In the 'Attached resources' section, a network interface 'eth0' is listed under 'Display resource'.

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

The screenshot shows the IBM Cloud VPC Infrastructure interface, specifically the 'Security groups' section for the 'frequency-pectin-implosion-fondue' group. The 'Rules' tab is active and highlighted with a red box. The 'Attached resources' section shows a single network interface 'eth0' attached to the virtual server 'php' in subnet 'sn-20211124-02'. The 'Rules' section indicates there are 5 inbound rules and 9 outbound rules.

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

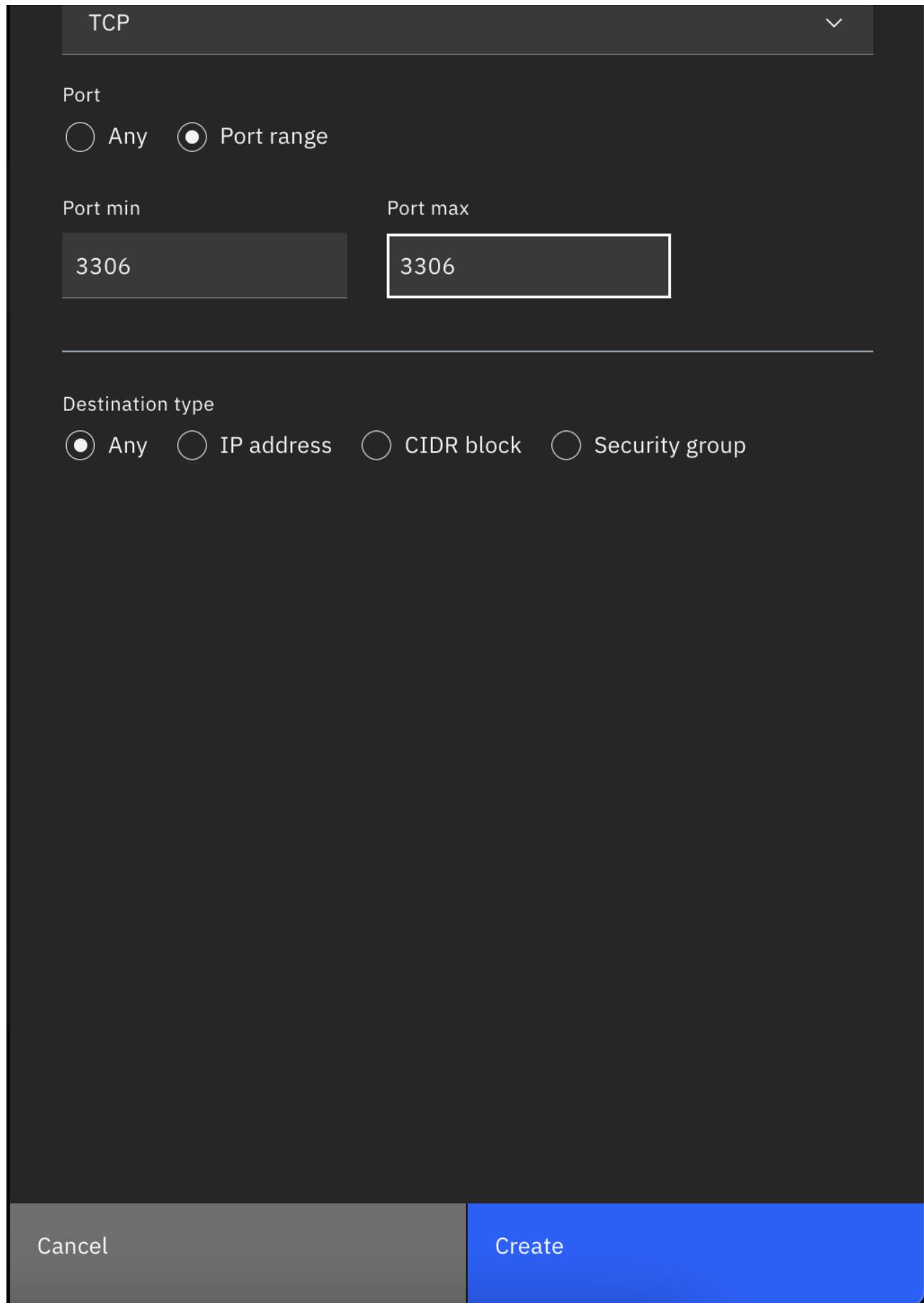
The screenshot shows the IBM Cloud interface for managing security groups. The left sidebar is collapsed. The main area displays the 'frequency-peptic-implosion-fondue' security group. Under the 'Inbound rules' tab, there is a table with columns: Protocol, Source type, Source, and Value. The table contains several rows for ICMP and TCP protocols. At the top right of the table, there is a 'Create' button with a plus sign, which is highlighted with a red box.

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

The screenshot shows the same IBM Cloud interface for managing security groups. Under the 'Outbound rules' tab, there is a table with columns: Protocol, Destination type, Destination, and Value. The table contains several rows for ICMP and TCP protocols. At the top right of the table, there is a 'Create' button with a plus sign, which is highlighted with a red box.

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

The screenshot shows a modal dialog titled 'Create outbound rule'. At the bottom left, there is a 'Protocol' input field. The top right corner of the dialog has a standard close button represented by an 'X'.



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 search bar and a red box highlights the 'Catalog' tab in the top navigation bar. Below the dashboard, there are several service cards: 'Build' (Explore IBM Cloud with this selection of easy starter tutorials and services), 'Get started with Watson Discovery' (Get up to speed on Watson Discovery with step-by-step tutorials, deep-dive videos, and complete examples of working code), 'Augment AI search in your domain' (Deploying Watson Knowledge Studio custom models into Watson Discovery can significantly boost the quality of Watson Discovery query responses.), 'Plan your infrastructure' (Design and size your infrastructure to make sure it fits your workloads.), 'Get started with VPN' (VPN access enables users to manage all servers remotely and securely over the IBM Cloud private network.), 'IBM Log Analysis with LogDNA' (Troubleshoot logs in real-time to diagnose issues and identify problems. Leverage hundreds of integrations and built-in IBM Cloud logging.), and 'Explore II' (Try a common approach developing web projects). Below these cards 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. A red box highlights the search bar at the top left, which contains the text 'Internet Services'. An arrow points from the search bar to the left sidebar where the 'Internet Services' category is selected. The sidebar also lists other categories like 'Kubernetes Service', 'Secured Enterprise File Services Platform', 'Secure Gateway', 'Compose Enterprise', 'Analytics (10)', 'Blockchain (1)', 'Databases (30)', 'Developer tools (43)', 'Logging and monitoring (5)', 'Migration (8)', and 'Integration (10)'. The main area displays several service cards: 'Analytics Engine' (By IBM), 'Annotator for Clinical Data' (By IBM), 'AnonTech ViziVault Platform' (By Anon Technology, Inc.), and 'API Connect' (By IBM). A tooltip for the 'Analytics Engine' card states: 'Pricing might reflect internal IBM pricing, which should not be shared with external users. Please log in to your external IBM Cloud account to view external pricing details.'

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" service overview page. The sidebar on the left has tabs for Overview (which is selected), Getting started, Account, Metrics, Security, Reliability, Performance, Edge Functions, and Plan. The main content area features a large "Start by adding a domain." callout with a "Add domain" button highlighted with a red box. Below this, there's a link to "Learn more about Internet Services." A blue "Edit" icon is visible in the bottom right corner of the page.

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 (indicated by a red box). In the main content area, the 'Plan' section shows a trial period of '30 days remaining'. The 'Domain ID' is listed as 'ff37be50fb5a96338760747a338f1cda'. The 'CRN' (Cloud Resource Name) is listed as 'crn:v1:bluemix:public:internet-svcs:global:a/c409b38a21f14c33a0d1964b06334bb:d56e8172-771a-4cca-8507-14b323d8fbce:'. Under 'Name servers', the values 'ns122.name.cloud.ibm.com' and 'ns169.name.cloud.ibm.com' are shown. The 'Service modes' section contains 'Defense mode' (disabled) and 'Development mode' (disabled). The 'Events' section shows a 3D cube icon and a blue 'View events' button.

7. Entre na aba de DNS.

The screenshot shows the IBM Cloud dashboard for a service named 'Test'. The 'Reliability' section is currently selected. Under 'Global load balancers', the 'DNS' tab is highlighted with a red box. A sub-section titled 'Load balancers' is visible, showing a table with columns: Health, Hostname, Available pools, TTL, Proxy, and Enabled. A blue 'Create +' button is located at the top right of this table. Below the table, it says 'There are no load balancers set up.'

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

The screenshot shows the 'DNS records' page. At the top, there is a note: 'application names may cause conflicts with your DNS resolution.' Below this, 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 highlights a 'Domain pending NS record:' note: '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](#)'.

Below the note, there is a search bar and a table header with columns: Type, Name, Value, TTL, and Proxy. A blue 'Add +' button is highlighted with a red box. The table body below the header shows the message: 'There are no records. You may add new records above or import a DNS file in the table settings menu.' At the bottom, there are pagination controls: 'Items per page: 25', '0 - 0 of 0 items', '1 of 1 page', and navigation arrows.

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

The screenshot shows the 'Add record' dialog box. The title 'Add record' is at the top left, and a close 'X' icon is at the top right. The dialog has two main input fields: 'Type' and 'TTL'. The 'Type' field dropdown is set to 'A' and has a downward arrow. The 'TTL' field dropdown is set to 'Automatic' and also has a downward arrow. There is a large blue 'Add' button at the bottom right of the dialog.

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

```
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.