

Configurando uma Aplicação MVC PHP em Servidor Ubuntu com Apache

Este guia apresenta os passos necessários para configurar uma aplicação web MVC PHP de um repositório GitHub em um servidor Ubuntu com Apache, tornando-a acessível via endereço IP.

1. Atualizar o Sistema

Primeiro, atualize os pacotes do sistema:

```
bash  
  
sudo apt update  
sudo apt upgrade -y
```

2. Instalar o Apache e PHP

Instale o servidor web Apache e o PHP com as extensões necessárias:

```
bash  
  
sudo apt install apache2 -y  
sudo apt install php php-cli php-common php-mysql php-zip php-gd php-mbstring php-curl php-xml
```



3. Verificar o Status do Apache

Certifique-se de que o Apache está rodando:

```
bash  
  
sudo systemctl status apache2
```

Se não estiver ativo, inicie-o:

```
bash  
  
sudo systemctl start apache2  
sudo systemctl enable apache2
```

4. Configurar o Firewall

Libere a porta HTTP (80) no firewall:

```
bash
```

```
sudo ufw allow 'Apache'  
sudo ufw enable  
sudo ufw status
```

5. Clonar o Repositório GitHub

Instale o Git e clone seu repositório:

```
bash
```

```
sudo apt install git -y  
cd /var/www/html  
sudo rm -rf index.html # Remover página padrão do Apache  
sudo git clone [URL-DO-SEU-REPOSITORIO] .
```

Substitua `[URL-DO-SEU-REPOSITORIO]` pelo URL do seu repositório GitHub.

6. Configurar Permissões

Configure permissões adequadas para os arquivos da aplicação:

```
bash
```

```
sudo chown -R www-data:www-data /var/www/html/  
sudo chmod -R 755 /var/www/html/
```

7. Configuração do Virtual Host do Apache

Crie um arquivo de configuração para o site:

```
bash
```

```
sudo nano /etc/apache2/sites-available/minha-aplicacao.conf
```

Adicione a seguinte configuração:

apache

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    <Directory /var/www/html>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

8. Habilitar o Virtual Host e o Módulo Rewrite

bash

```
sudo a2ensite minha-aplicacao.conf
sudo a2dissite 000-default.conf
sudo a2enmod rewrite
sudo systemctl restart apache2
```

9. Configurar arquivo .htaccess

Se sua aplicação MVC precisa de um arquivo .htaccess para as rotas funcionarem corretamente, verifique se ele está presente ou crie um na raiz do projeto:

bash

```
sudo nano /var/www/html/.htaccess
```

Conteúdo básico para um MVC PHP (ajuste conforme necessário):

apache

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?url=$1 [QSA,L]
```

10. Configuração da Aplicação

Se sua aplicação precisar de um arquivo de configuração (como conexão com banco de dados), ajuste-o para corresponder ao ambiente:

```
bash
```

```
sudo nano /var/www/html/config/database.php # Ajuste o caminho conforme necessário
```

11. Instalação de Dependências (se usar Composer)

Se seu projeto usa Composer para gerenciar dependências:

```
bash
```

```
sudo apt install composer -y  
cd /var/www/html  
sudo composer install
```

12. Testar a Aplicação

Agora você deve conseguir acessar a aplicação de qualquer navegador digitando o endereço IP do servidor:

```
http://[IP-DO-SEU-SERVIDOR]
```

Dicas de Solução de Problemas

1. Verificar logs do Apache:

```
bash
```

```
sudo tail -f /var/log/apache2/error.log
```

2. Verificar configuração do PHP:

```
bash
```

```
php -i | grep php.ini
```

3. Verificar conectividade:

```
bash
```

```
sudo netstat -tulpn | grep apache2
```

4. Caso ocorra problemas com permissões:

bash

```
sudo find /var/www/html -type d -exec chmod 755 {} \;
```

```
sudo find /var/www/html -type f -exec chmod 644 {} \;
```