Guia de Implantação: Dashboard de Streaming Multicast em VM Ubuntu 22.04 no Proxmox

Este guia detalhado fornecerá instruções passo a passo para implantar o Dashboard de Streaming Multicast em uma Máquina Virtual (VM) rodando Ubuntu Server 22.04 LTS dentro de um ambiente Proxmox Virtual Environment. Ao final deste guia, seu dashboard estará acessível publicamente através do domínio `4menlive.supersamsm.com.br`, com todas as funcionalidades de backend, frontend e WebRTC operacionais e seguras.

1. Preparação da Máquina Virtual no Proxmox

Antes de iniciar a configuração do software, é crucial preparar adequadamente a máquina virtual no seu ambiente Proxmox. Certifique-se de que seu servidor Proxmox esteja configurado e acessível.

1.1. Download da Imagem ISO do Ubuntu Server 22.04 LTS

Primeiramente, você precisará da imagem ISO do Ubuntu Server 22.04 LTS. É recomendável baixar a versão LTS (Long Term Support) para garantir estabilidade e suporte a longo prazo. Você pode obter a imagem diretamente do site oficial do Ubuntu [1].

> *[1]* ***Ubuntu Server Download****:* `https://ubuntu.com/download/server`

Após o download, faça o upload da imagem ISO para o armazenamento ISO do seu Proxmox. Isso geralmente é feito através da interface web do Proxmox, navegando até `Datacenter > [Seu Nó Proxmox] > Local (ou seu armazenamento ISO) > ISO Images > Upload`.

1.2. Criação da Nova Máquina Virtual

Na interface web do Proxmox, clique em `Create VM` (Criar VM) no canto superior direito. Siga os passos abaixo, preenchendo as informações conforme as recomendações:

Guia Geral (General)

- **Node**: Selecione o nó Proxmox onde a VM será criada.
- **VM ID**: Um ID único para sua VM (ex: `100`).
- **Name**: Um nome descritivo para a VM (ex: `streaming-dashboard-ubuntu`).

Guia OS (Operating System)

- **ISO image**: Selecione a imagem ISO do Ubuntu Server 22.04 LTS que você acabou de fazer upload.
- **Guest OS**: Defina `Type` como `Linux` e `Version` como `5.x - 2.6 Kernel (lts)`.

Guia System (Sistema)

- **Graphic card**: `Default` (padrão).
- **SCSIF**: `VirtIO SCSI` (melhor desempenho).
- **Qemu Agent**: Marque `Qemu Agent` (recomendado para melhor gerenciamento da VM).

Guia Disks (Discos)

- **Bus/Device**: `VirtIO Block` (melhor desempenho).
- **Storage**: Selecione o armazenamento onde o disco da VM será criado.
- **Disk size**: Recomenda-se um mínimo de `40GB` para o sistema operacional e o projeto. Se planeja armazenar muitos logs ou dados, considere mais espaço.

Guia CPU

- **Cores**: Atribua pelo menos `2` núcleos de CPU. Para um desempenho ideal, especialmente com o servidor WebRTC e múltiplas transmissões, `4` núcleos ou

mais são recomendados.

- **Type**: `host` (para melhor compatibilidade e desempenho).

Guia Memory (Memória)

- **Memory (MiB)**: Atribua pelo menos `4096 MiB` (4GB) de RAM. Para cargas de trabalho mais pesadas ou muitos convidados simultâneos, `8192 MiB` (8GB) ou mais seria ideal.

Guia Network (Rede)

- **Bridge**: `vmbr0` (ou sua bridge de rede principal).
- **Model**: `VirtIO (paravirtualized)` (melhor desempenho).
- **Firewall**: Desmarcado (o firewall será configurado dentro da VM).

Guia Confirm (Confirmar)

- Revise todas as configurações. Marque `Start after creation` (Iniciar após a criação) se desejar que a VM inicie automaticamente após ser criada.

1.3. Instalação do Ubuntu Server 22.04 LTS

Após a criação da VM, o Proxmox iniciará o processo de instalação do Ubuntu Server. Conecte-se ao console da VM através da interface web do Proxmox (`Console` no menu da VM) e siga as instruções na tela:

- **Idioma**: Selecione `Português (Brasil)`.
- **Layout do Teclado**: Detecte ou selecione o layout correto.
- **Configuração de Rede**: Se você usa DHCP, a VM deve obter um IP automaticamente. Para um ambiente de produção, é **altamente recomendável configurar um IP estático** para a VM. Anote o endereço IP, máscara de sub-rede, gateway e servidores DNS.
- **Configuração de Proxy**: Deixe em branco, a menos que sua rede exija um proxy.
- **Mirror do Arquivo**: Use o padrão ou um mirror mais próximo de você.
- **Configuração de Armazenamento**: Selecione `Use an entire disk` (Usar um disco inteiro) e confirme. Para usuários avançados, é possível configurar LVM.

- **Configuração de Perfil**: Crie um nome de usuário (ex: `ubuntu` ou `admin`), nome do servidor (ex: `streaming-dashboard`), e uma senha forte. **Anote essas credenciais.**
- **Instalação de SSH**: Marque a opção `Install OpenSSH server`. Isso permitirá que você acesse a VM remotamente via SSH, o que é essencial para as próximas etapas.
- **Recursos do Servidor**: Não selecione nenhum recurso adicional neste momento. A instalação será mais rápida e você instalará apenas o necessário posteriormente.

Após a instalação, a VM será reiniciada. Remova a imagem ISO do drive virtual da VM no Proxmox para evitar que ela inicialize novamente a partir da ISO. Agora você pode acessar sua VM via SSH usando o IP que você configurou (ou o IP DHCP) e as credenciais que você criou.

2. Configuração Inicial do Servidor Ubuntu

Após a instalação do Ubuntu Server 22.04 LTS e o primeiro login via SSH, é fundamental realizar algumas configurações iniciais para garantir a segurança, a atualização do sistema e a preparação para a implantação do dashboard.

2.1. Atualização do Sistema

É crucial manter o sistema operacional atualizado para garantir que você tenha as últimas correções de segurança e melhorias de desempenho. Execute os seguintes comandos no terminal da sua VM:

```
 sudo apt update
sudo apt upgrade -y
sudo apt autoremove -y
```

- `sudo apt update`: Atualiza a lista de pacotes disponíveis nos repositórios.
- `sudo apt upgrade -y`: Instala as novas versões dos pacotes que já estão instalados no sistema. O `-y` confirma automaticamente todas as perguntas de instalação.

- `sudo apt autoremove -y`: Remove pacotes que foram instalados automaticamente para satisfazer dependências de outros pacotes e que não são mais necessários.

2.2. Configuração do Firewall (UFW)

O Uncomplicated Firewall (UFW) é uma interface amigável para o `iptables` e é a ferramenta de firewall padrão no Ubuntu. É essencial configurá-lo para permitir apenas o tráfego necessário, aumentando a segurança da sua VM.

Primeiro, permita as conexões SSH para que você não perca o acesso à sua VM:

```
sudo ufw allow ssh
```

Em seguida, você precisará permitir o tráfego para as portas que serão usadas pelo seu Dashboard de Streaming Multicast:

- **Porta 80 (HTTP)**: Para acesso inicial ao site e para o Certbot.
- **Porta 443 (HTTPS)**: Para acesso seguro ao site após a configuração do SSL.
- **Porta 5002 (Backend Flask)**: A porta que o backend Flask estará escutando (ajuste se você configurou uma porta diferente).
- **Porta 3002 (WebRTC Server)**: A porta que o servidor WebRTC estará escutando (ajuste se você configurou uma porta diferente).

Execute os comandos para permitir essas portas:

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw allow 5002/tcp
sudo ufw allow 3002/tcp
```

Após permitir as portas necessárias, habilite o firewall:

```
sudo ufw enable
```

Você será avisado de que habilitar o firewall pode interromper as conexões SSH existentes. Confirme com `y` e pressione Enter. Sua conexão SSH não será interrompida porque você já permitiu a porta SSH.

Para verificar o status do firewall e as regras ativas:

```
sudo ufw status verbose
```

2.3. Instalação de Ferramentas Essenciais

Para o desenvolvimento e implantação do projeto, você precisará de algumas ferramentas e dependências. Instale-as usando `apt`:

```
sudo apt install -y git curl wget build-essential
```

- `git`: Essencial para clonar o repositório do seu projeto do GitHub.
- `curl` e `wget`: Utilitários de linha de comando para baixar arquivos da internet.
- `build-essential`: Pacotes que incluem compiladores C/C++ e outras ferramentas necessárias para compilar software a partir do código-fonte, o que pode ser útil para algumas dependências.

2.4. Instalação do Node.js e pnpm

O frontend React e o servidor WebRTC dependem do Node.js. É recomendável instalar uma versão LTS do Node.js e o gerenciador de pacotes `pnpm` (que você usou no desenvolvimento).

2.4.1. Instalação do Node.js

Use o `curl` para baixar e executar o script de instalação do NodeSource, que adicionará o repositório oficial do Node.js ao seu sistema. Certifique-se de instalar uma versão LTS (Long Term Support), como a 18.x ou 20.x:

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
```

Verifique a instalação:

```
node -v
npm -v
```

2.4.2. Instalação do pnpm

O `pnpm` é um gerenciador de pacotes rápido e eficiente para Node.js. Instale-o globalmente usando `npm`:

```
sudo npm install -g pnpm
```

Verifique a instalação:

```
pnpm -v
```

2.5. Instalação do Python e pip

O backend Flask é desenvolvido em Python. O Ubuntu 22.04 já vem com Python 3 pré-instalado, mas é bom garantir que `pip` (o gerenciador de pacotes do Python) esteja disponível e atualizado.

```
sudo apt install -y python3-pip python3-venv
```

- `python3-pip`: Instala o `pip` para Python 3.
- `python3-venv`: Permite a criação de ambientes virtuais Python, o que é uma boa prática para isolar as dependências do projeto.

Verifique a instalação:

```
python3 --version
pip3 --version
```

Com essas configurações iniciais, seu servidor Ubuntu estará pronto para receber os arquivos do projeto e iniciar a implantação do Dashboard de Streaming Multicast.

3. Transferência e Configuração do Projeto

Com o servidor Ubuntu devidamente configurado e as ferramentas essenciais instaladas, o próximo passo é transferir os arquivos do seu projeto Dashboard de Streaming Multicast para a VM e configurar cada um dos seus componentes (backend, frontend e servidor WebRTC).

3.1. Transferência dos Arquivos do Projeto

Você pode transferir os arquivos do projeto para a VM de diversas maneiras. A mais comum e recomendada é usar `scp` (Secure Copy Protocol) ou clonar diretamente do seu repositório Git (se o projeto estiver versionado).

3.1.1. Usando `scp` (se você tem o .zip localmente)

Se você tem o arquivo `streaming-dashboard-project.zip` (que eu gerei para você anteriormente) em sua máquina local, pode transferi-lo para a VM usando `scp`. Substitua `seu_usuario` pelo nome de usuário que você criou na VM e `seu_ip_da_vm` pelo endereço IP da sua VM Ubuntu.

```
scp /caminho/para/streaming-dashboard-project.zip
seu_usuario@seu_ip_da_vm:/home/seu_usuario/
```

Após a transferência, conecte-se via SSH à sua VM e descompacte o arquivo:

```
ssh seu_usuario@seu_ip_da_vm
cd /home/seu_usuario/
unzip streaming-dashboard-project.zip
```

Isso criará um diretório `streaming-dashboard-project` contendo o backend, frontend e o servidor WebRTC.

3.1.2. Clonando do GitHub (se o projeto estiver em um repositório)

Se o seu projeto estiver em um repositório GitHub (ou outro Git), você pode cloná-lo diretamente na VM. Certifique-se de ter o `git` instalado (seção 2.3).

```
ssh seu_usuario@seu_ip_da_vm
cd /home/seu_usuario/
git clone https://github.com/seu_usuario/seu_repositorio.git streaming-
dashboard-project
```

Substitua `https://github.com/seu_usuario/seu_repositorio.git` pela URL real do seu repositório.

3.2. Configuração do Backend (Flask)

O backend Flask é responsável pela API, autenticação e integração com as plataformas de streaming. Ele será executado usando `gunicorn` para produção e `supervisor` para gerenciar o processo.

Navegue até o diretório do backend:

```
cd /home/seu_usuario/streaming-dashboard-project/streaming-dashboard-backend
```

3.2.1. Criação do Ambiente Virtual e Instalação de Dependências

É uma boa prática usar um ambiente virtual para isolar as dependências do Python:

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

3.2.2. Configuração das Variáveis de Ambiente

Crie um arquivo `.env` na raiz do diretório `streaming-dashboard-backend` com as variáveis de ambiente necessárias para produção. Este arquivo **não deve ser versionado** no Git por conter informações sensíveis.

```
nano .env
```

Adicione o seguinte conteúdo (ajuste os valores conforme necessário):

```
SECRET_KEY=sua_chave_secreta_muito_forte_e_unica
DATABASE_URL=sqlite:///./database.db  # Ou sua URL de banco de dados de
produção (PostgreSQL, MySQL, etc.)
FLASK_ENV=production
```

- `SECRET_KEY`: Uma chave secreta longa e aleatória, usada para segurança de sessões e JWT.
- `DATABASE_URL`: A URL de conexão com seu banco de dados. Para produção, é altamente recomendável usar um banco de dados robusto como PostgreSQL ou MySQL em vez de SQLite.
- `FLASK_ENV`: Define o ambiente como produção.

3.2.3. Instalação do Gunicorn

O Gunicorn é um servidor WSGI Python que será usado para servir sua aplicação Flask em produção. Instale-o dentro do seu ambiente virtual:

```
pip install gunicorn
```

3.2.4. Criação do Serviço Supervisor para o Backend

O Supervisor é um sistema de controle de processos que permite que você gerencie processos de longa duração, como o Gunicorn, garantindo que eles permaneçam em execução e reiniciem automaticamente em caso de falha.

Instale o Supervisor:

```
sudo apt install -y supervisor
```

Crie um arquivo de configuração para o seu serviço Flask em `/etc/supervisor/conf.d/streaming_backend.conf`:

```
sudo nano /etc/supervisor/conf.d/streaming_backend.conf
```

Adicione o seguinte conteúdo (ajuste `seu_usuario` e `seu_ip_da_vm`):

```
[program:streaming_backend]
command=/home/seu_usuario/streaming-dashboard-project/streaming-dashboard-
backend/venv/bin/gunicorn -w 4 -b 0.0.0.0:5002 src.main:app
directory=/home/seu_usuario/streaming-dashboard-project/streaming-dashboard-
backend
user=seu_usuario
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
stderr_logfile=/var/log/supervisor/streaming_backend_err.log
stdout_logfile=/var/log/supervisor/streaming_backend_out.log
environment=PATH="/home/seu_usuario/streaming-dashboard-project/streaming-
dashboard-
backend/venv/bin",FLASK_ENV="production",SECRET_KEY="sua_chave_secreta_muito_fort
```

- `command`: O comando para iniciar o Gunicorn. `-w 4` significa 4 workers (ajuste conforme os núcleos da sua CPU). `-b 0.0.0.0:5002` faz com que o Gunicorn escute em todas as interfaces na porta 5002.
- `directory`: O diretório raiz do seu projeto Flask.

- `user`: O usuário sob o qual o processo será executado (recomenda-se um usuário não-root).
- `environment`: É crucial passar as variáveis de ambiente aqui, incluindo `SECRET_KEY` e `DATABASE_URL`.

Após salvar o arquivo, atualize e inicie o Supervisor:

```
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start streaming_backend
```

Verifique o status do serviço:

```
sudo supervisorctl status streaming_backend
```

3.3. Configuração do Frontend (React)

O frontend React é a interface do usuário. Ele será servido por um servidor web (Nginx, que será configurado na próxima seção).

Navegue até o diretório do frontend:

```
cd /home/seu_usuario/streaming-dashboard-project/streaming-dashboard-frontend
```

3.3.1. Instalação de Dependências e Build de Produção

```
pnpm install
pnpm run build
```

Isso criará uma pasta `dist` contendo os arquivos estáticos otimizados para produção. Estes arquivos serão servidos pelo Nginx.

3.3.2. Configuração das Variáveis de Ambiente para o Frontend

Crie um arquivo `.env` na raiz do diretório `streaming-dashboard-frontend` para as variáveis de ambiente do frontend. Este arquivo também **não deve ser versionado**.

```
nano .env
```

Adicione o seguinte conteúdo (ajuste as URLs para as URLs públicas do seu backend e servidor WebRTC após a implantação):

```
 VITE_API_URL=https://4menlive.supersamsm.com.br/api
VITE_WEBRTC_URL=https://4menlive.supersamsm.com.br/webrtc
```

Importante: As URLs acima são exemplos. Elas devem apontar para os seus domínios reais após a configuração do Nginx e DNS. Por enquanto, você pode usar `http://seu_ip_da_vm:5002` para `VITE_API_URL` e `http://seu_ip_da_vm:3002` para `VITE_WEBRTC_URL` para testes internos, mas para acesso público, elas precisarão ser as URLs baseadas no domínio `4menlive.supersamsm.com.br` que você configurará com Nginx.

Após alterar o `.env`, você precisará rodar `pnpm run build` novamente para que as novas variáveis de ambiente sejam incorporadas ao build de produção.

3.4. Configuração do Servidor WebRTC (Node.js)

O servidor WebRTC é responsável pela comunicação em tempo real com baixa latência. Ele também será gerenciado pelo Supervisor.

Navegue até o diretório do servidor WebRTC:

```
cd /home/seu_usuario/streaming-dashboard-project/webrtc-server
```

3.4.1. Instalação de Dependências

```
npm install
```

3.4.2. Criação do Serviço Supervisor para o Servidor WebRTC

Crie um arquivo de configuração para o seu serviço WebRTC em `/etc/supervisor/conf.d/streaming_webrtc.conf`:

```
sudo nano /etc/supervisor/conf.d/streaming_webrtc.conf
```

Adicione o seguinte conteúdo (ajuste `seu_usuario` e `seu_ip_da_vm`):

```
[program:streaming_webrtc]
command=node server.js
directory=/home/seu_usuario/streaming-dashboard-project/webrtc-server
user=seu_usuario
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
stderr_logfile=/var/log/supervisor/streaming_webrtc_err.log
stdout_logfile=/var/log/supervisor/streaming_webrtc_out.log
environment=PORT="3002"
```

- `command`: O comando para iniciar o servidor Node.js.
- `directory`: O diretório raiz do seu projeto WebRTC.
- `user`: O usuário sob o qual o processo será executado.
- `environment`: Define a porta que o servidor WebRTC irá escutar.

Após salvar o arquivo, atualize e inicie o Supervisor:

```
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start streaming_webrtc
```

Verifique o status do serviço:

```
sudo supervisorctl status streaming_webrtc
```

Com todos os componentes do projeto configurados e rodando via Supervisor, o próximo passo é configurar o Nginx como um reverse proxy para expor esses serviços publicamente e configurar o SSL com Certbot.

4. Configurar Nginx como Reverse Proxy e SSL com Certbot

Para que seu Dashboard de Streaming Multicast seja acessível publicamente via `4menlive.supersamsm.com.br` e utilize HTTPS para segurança, você precisará configurar o Nginx como um reverse proxy e obter um certificado SSL gratuito com o Certbot.

4.1. Instalação do Nginx

O Nginx é um servidor web de alto desempenho que será usado para rotear as requisições do seu domínio para os serviços de backend, frontend e WebRTC que estão rodando internamente na sua VM.

```
sudo apt install -y nginx
```

Após a instalação, o Nginx deve iniciar automaticamente. Você pode verificar o status:

```
sudo systemctl status nginx
```

Se estiver rodando, você verá `active (running)`.

4.2. Configuração do Nginx como Reverse Proxy

Você precisará criar um arquivo de configuração para o seu domínio no Nginx. Crie um novo arquivo em `/etc/nginx/sites-available/4menlive.supersamsm.com.br`:

```
sudo nano /etc/nginx/sites-available/4menlive.supersamsm.com.br
```

Adicione o seguinte conteúdo. Este arquivo configurará o Nginx para servir o frontend, rotear requisições `/api` para o backend Flask e requisições `/webrtc` para o servidor WebRTC.

```
server {
    listen 80;
    server_name 4menlive.supersamsm.com.br;

    # Redireciona HTTP para HTTPS (será habilitado após o Certbot)
    # return 301 https://$`host`$request_uri;

    root /home/seu_usuario/streaming-dashboard-project/streaming-dashboard-
frontend/dist;
    index index.html index.htm;

    location / {
        try_files $`uri `$uri/ /index.html;
    }

    # Proxy para o Backend Flask
    location /api/ {
        proxy_pass http://127.0.0.1:5002/api/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_redirect off;

        # Configurações para WebSockets (se o Flask usar)
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }

    # Proxy para o Servidor WebRTC
    location /webrtc/ {
        proxy_pass http://127.0.0.1:3002/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # Configurações para WebSockets (essencial para WebRTC)
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_read_timeout 86400s; # Longo timeout para conexões WebRTC
        proxy_send_timeout 86400s;
    }
}
```

Ajuste `seu_usuario` para o nome de usuário que você está usando na VM.

Crie um link simbólico do seu arquivo de configuração para o diretório `sites-enabled` para ativá-lo:

```
sudo ln -s /etc/nginx/sites-available/4menlive.supersamsm.com.br
/etc/nginx/sites-enabled/
```

Remova o arquivo de configuração padrão do Nginx para evitar conflitos:

```
sudo rm /etc/nginx/sites-enabled/default
```

Teste a sintaxe da configuração do Nginx e reinicie o serviço:

```
sudo nginx -t
sudo systemctl restart nginx
```

Neste ponto, se o seu DNS já estiver configurado (veja a próxima seção), você já poderá acessar o site via HTTP. No entanto, ainda não é seguro.

4.3. Instalação do Certbot e Obtenção do Certificado SSL

O Certbot é uma ferramenta que automatiza a obtenção e renovação de certificados SSL/TLS gratuitos do Let's Encrypt. Ele pode configurar automaticamente o Nginx para usar HTTPS.

```
sudo snap install core
sudo snap refresh core
sudo snap install --classic certbot
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

Agora, execute o Certbot para obter e instalar o certificado SSL para o seu domínio. O Certbot detectará sua configuração do Nginx e fará as alterações necessárias, incluindo o redirecionamento de HTTP para HTTPS.

```
sudo certbot --nginx -d 4menlive.supersamsm.com.br
```

Siga as instruções na tela: - Digite seu endereço de e-mail. - Aceite os termos de serviço. - Escolha se deseja compartilhar seu e-mail com a EFF (opcional). - Quando perguntado sobre o redirecionamento de HTTP para HTTPS, escolha a opção para **redirecionar todo o tráfego HTTP para HTTPS** (geralmente opção `2`).

Após a conclusão, o Certbot informará que o certificado foi instalado com sucesso e que o Nginx foi configurado para usar HTTPS. Ele também configurará a renovação automática do certificado.

Você pode verificar o status da renovação automática:

```
sudo systemctl status snap.certbot.renew.service
```

Com o Nginx configurado e o SSL ativado, seu Dashboard de Streaming Multicast estará acessível de forma segura via HTTPS. O próximo passo é garantir que seu domínio aponte para o IP da sua VM.

5. Configuração de DNS

Para que seu Dashboard de Streaming Multicast seja acessível através do domínio `4menlive.supersamsm.com.br`, você precisará configurar os registros DNS no seu provedor de domínio. Este é um passo crucial que conecta o nome do seu domínio ao endereço IP público da sua VM.

5.1. Encontrando o Endereço IP Público da Sua VM

Se sua VM está em uma rede local (atrás de um roteador), você precisará garantir que o endereço IP que você configurou para a VM (ou o que ela obteve via DHCP) seja acessível externamente. Isso geralmente envolve a configuração de **Redirecionamento de Portas (Port Forwarding)** no seu roteador.

Você precisará redirecionar as portas **80 (HTTP)** e **443 (HTTPS)** do seu roteador para o endereço IP interno da sua VM Ubuntu. Consulte o manual do seu roteador ou procure por tutoriais específicos para o seu modelo para configurar o redirecionamento de portas.

Após configurar o redirecionamento, você pode encontrar o seu endereço IP público (o IP do seu roteador que é visível para a internet) usando sites como `whatismyip.com` ou `meuip.com.br`.

5.2. Adicionando Registros DNS

Acesse o painel de controle do seu provedor de domínio (onde você registrou `supersamsm.com.br`). Procure pela seção de gerenciamento de DNS ou

gerenciamento de zonas DNS. Você precisará adicionar ou modificar os seguintes registros:

Registro A para o Domínio Principal

Este registro mapeia o seu domínio principal para o endereço IP público da sua VM.

- **Tipo**: `A`
- **Nome/Host**: `@` ou `4menlive` (dependendo do provedor, `@` geralmente representa o domínio raiz, ou você pode precisar criar um subdomínio `4menlive`)
- **Valor/Endereço IP**: O endereço IP público da sua VM Ubuntu.
- **TTL (Time To Live)**: Deixe o padrão (geralmente 3600 segundos ou 1 hora), ou um valor menor (ex: 300 segundos) se você precisar que as alterações se propaguem mais rapidamente.

Registro CNAME (Opcional, para www)

Se você deseja que o site também seja acessível via `www.4menlive.supersamsm.com.br`, você pode adicionar um registro CNAME.

- **Tipo**: `CNAME`
- **Nome/Host**: `www`
- **Valor/Destino**: `4menlive.supersamsm.com.br` (o seu domínio principal)
- **TTL (Time To Live)**: Padrão

Exemplo de Configuração DNS (pode variar ligeiramente entre provedores):

Tipo	Nome/Host	Valor/Endereço IP	TTL
`A`	`@` ou `4menlive`	`SEU_IP_PUBLICO_DA_VM`	`3600`
`CNAME`	`www`	`4menlive.supersamsm.com.br`	`3600`

Após salvar as alterações nos registros DNS, pode levar algum tempo (de alguns minutos a 48 horas, dependendo do TTL e do seu provedor) para que as alterações se propaguem pela internet. Você pode verificar a propagação do DNS usando ferramentas online como `DNS Checker` [2].

> *[2] **DNS Checker**: `https://dnschecker.org/`*

Uma vez que o DNS esteja propagado e apontando para o IP público da sua VM, e com o Nginx e Certbot configurados, seu Dashboard de Streaming Multicast estará totalmente acessível via `https://4menlive.supersamsm.com.br`.