

Guia de Instalação: Dashboard de Streaming Multicast em Ubuntu 22.04

Este guia detalha o processo de instalação e configuração do Dashboard de Streaming Multicast em uma máquina virtual (VM) com Ubuntu Server 22.04 LTS. O objetivo é configurar o ambiente para que o frontend seja acessível via `4menlive.supersamsm.com.br` e o backend via `4menapi.supersamsm.com.br`, garantindo uma operação robusta e segura.

1. Introdução

O Dashboard de Streaming Multicast é uma solução completa para gerenciar transmissões ao vivo simultâneas para diversas plataformas (Twitch, YouTube, Facebook, Instagram, TikTok) e integrar convidados com controle de latência e qualidade. A arquitetura da aplicação é composta por três componentes principais:

- **Backend (Flask):** Responsável pela lógica de negócios, autenticação, gerenciamento de plataformas e convidados, e a API para o frontend.
- **Frontend (React):** A interface do usuário que interage com o backend e exibe as métricas e controles.
- **Servidor WebRTC (Node.js):** Gerencia as conexões de áudio e vídeo em tempo real com os convidados.

Este guia fornecerá instruções passo a passo para configurar cada um desses componentes, além de configurar um servidor web (Nginx) como proxy reverso e SSL para comunicação segura.

2. Pré-requisitos

Antes de iniciar a instalação, certifique-se de que sua VM Ubuntu 22.04 atenda aos seguintes pré-requisitos:

- **Sistema Operacional:** Ubuntu Server 22.04 LTS (mínimo).
- **Acesso SSH:** Você deve ter acesso SSH à VM com um usuário que possua privilégios `sudo`.
- **Recursos de Hardware:**
 - **CPU:** Mínimo de 2 vCPUs (recomenda-se 4 vCPUs ou mais para transmissões com múltiplos convidados).
 - **RAM:** Mínimo de 4 GB (recomenda-se 8 GB ou mais).
 - **Armazenamento:** Mínimo de 40 GB de espaço em disco (SSD recomendado).
- **Conectividade de Rede:** A VM deve ter acesso à internet para baixar pacotes e dependências.
- **Domínios:** Dois subdomínios configurados para apontar para o endereço IP público da sua VM:
 - `4menlive.supersamsm.com.br` (para o frontend)
 - `4menapi.supersamsm.com.br` (para o backend)
- **Portas Abertas:** As seguintes portas devem estar abertas no firewall da sua VM e, se aplicável, no seu roteador (para acesso externo):
 - `80` (HTTP - para Certbot)
 - `443` (HTTPS - para Nginx)
 - `5000` (Backend Flask - porta interna, será proxyada pelo Nginx)
 - `3000` (Frontend React - porta interna, será proxyada pelo Nginx)
 - `3002` (Servidor WebRTC - porta interna, será proxyada pelo Nginx)

3. Configuração Inicial do Servidor

Conecte-se à sua VM via SSH e execute os seguintes comandos para atualizar o sistema e instalar pacotes essenciais:

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y curl git unzip nginx python3-pip python3-venv nodejs npm
```

Verifique as versões instaladas:

```
python3 --version
node --version
npm --version
nginx -v
```

4. Instalação do Backend (Flask)

O backend é uma aplicação Flask que gerencia a lógica de negócios.

4.1. Clonar o Repositório e Configurar o Ambiente

```
cd /opt
sudo git clone https://github.com/seu-usuario/streaming-dashboard-backend.git #
Substitua pelo seu repositório real
sudo chown -R $USER:$USER streaming-dashboard-backend
cd streaming-dashboard-backend
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

4.2. Configurar Variáveis de Ambiente

Crie um arquivo `.env` para as variáveis de ambiente do Flask. Este arquivo **não deve ser versionado** no Git.

```
nano .env
```

Adicione o seguinte conteúdo, substituindo `your-secret-key-here` por uma chave secreta forte e única:

```
FLASK_APP=src/main.py
FLASK_ENV=production
SECRET_KEY=your-secret-key-here
PORT=5000
```

Salve e feche o arquivo (`Ctrl+X`, `Y`, `Enter`).

4.3. Testar o Backend

Para garantir que o backend está funcionando corretamente, inicie-o temporariamente:

```
source venv/bin/activate
flask run --host=0.0.0.0 --port=5000
```

Você deverá ver uma mensagem indicando que o servidor está rodando. Pressione `ctrl+c` para parar o servidor.

4.4. Configurar o Gunicorn com Supervisor

Para gerenciar o processo do Flask em produção, usaremos Gunicorn e Supervisor.

Instale o Gunicorn:

```
pip install gunicorn
```

Crie um script de inicialização para o Gunicorn:

```
nano gunicorn_start.sh
```

Adicione o seguinte conteúdo:

```
#!/bin/bash

NAME="streaming_dashboard_backend"
FLASKDIR=/opt/streaming-dashboard-backend
SOCKFILE=/tmp/gunicorn.sock
USER=$USER
GROUP=$USER
NUM_WORKERS=3

echo "Starting $NAME as `whoami`"

# Activate the virtual environment
cd $FLASKDIR
source venv/bin/activate

# Start Gunicorn
exec gunicorn main:app \
  --workers $NUM_WORKERS \
  --user=$USER --group=$GROUP \
  --bind=unix:$SOCKFILE \
  --log-level=info \
  --log-file=- \
  --timeout 120
```

Torne o script executável:

```
chmod +x gunicorn_start.sh
```

Instale o Supervisor:

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

Crie um arquivo de configuração para o Supervisor:

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

Adicione o seguinte conteúdo:

```
[program:streaming_dashboard_backend]  
command=/opt/streaming-dashboard-backend/gunicorn_start.sh  
directory=/opt/streaming-dashboard-backend  
user=$USER  
autostart=true  
autorestart=true  
stopasgroup=true  
killasgroup=true  
stderr_logfile=/var/log/streaming_dashboard_backend.err.log  
stdout_logfile=/var/log/streaming_dashboard_backend.out.log  
environment=FLASK_APP="src/main.py",FLASK_ENV="production",SECRET_KEY="your-  
secret-key-here",PORT="5000"
```

Substitua `your-secret-key-here` pela sua chave secreta real. Salve e feche o arquivo.

Atualize e inicie o Supervisor:

```
sudo supervisorctl reread  
sudo supervisorctl update  
sudo supervisorctl start streaming_dashboard_backend
```

Verifique o status:

```
sudo supervisorctl status streaming_dashboard_backend
```

5. Instalação do Frontend (React)

O frontend é uma aplicação React que será servida pelo Nginx.

5.1. Clonar o Repositório e Instalar Dependências

```
cd /opt
sudo git clone https://github.com/seu-usuario/streaming-dashboard-frontend.git
# Substitua pelo seu repositório real
sudo chown -R $USER:$USER streaming-dashboard-frontend
cd streaming-dashboard-frontend
npm install -g pnpm # Instala o pnpm globalmente
pnpm install
```

5.2. Configurar Variáveis de Ambiente do Frontend

Crie um arquivo `.env.production` para as variáveis de ambiente do frontend. Este arquivo **não deve ser versionado** no Git.

```
nano .env.production
```

Adicione o seguinte conteúdo, apontando para a URL do seu backend e WebRTC:

```
VITE_API_URL=https://4menapi.supersamsm.com.br
VITE_WEBRTC_URL=https://4menlive.supersamsm.com.br:3002 # A porta 3002 será
proxyada pelo Nginx
```

Salve e feche o arquivo.

5.3. Build da Aplicação Frontend

```
pnpm run build
```

Os arquivos estáticos da aplicação serão gerados na pasta `dist/`.

6. Instalação do Servidor WebRTC (Node.js)

O servidor WebRTC é uma aplicação Node.js que gerencia as conexões em tempo real.

6.1. Clonar o Repositório e Instalar Dependências

```
cd /opt
sudo git clone https://github.com/seu-usuario/webrtc-server.git # Substitua
pelo seu repositório real
sudo chown -R $USER:$USER webrtc-server
cd webrtc-server
pnpm install
```

6.2. Configurar o Supervisor para o Servidor WebRTC

Crie um arquivo de configuração para o Supervisor:

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

Adicione o seguinte conteúdo:

```
[program:webrtc_server]
command=node server.js
directory=/opt/webrtc-server
user=$USER
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
stderr_logfile=/var/log/webrtc_server.err.log
stdout_logfile=/var/log/webrtc_server.out.log
environment=PORT="3002"
```

Salve e feche o arquivo.

Atualize e inicie o Supervisor:

```
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start webrtc_server
```

Verifique o status:

```
sudo supervisorctl status webrtc_server
```

7. Configuração do Nginx como Proxy Reverso e SSL

O Nginx será usado para servir o frontend, proxyar as requisições para o backend e o servidor WebRTC, e gerenciar os certificados SSL.

7.1. Configurar o Nginx para o Frontend e Backend

Crie um novo arquivo de configuração para o Nginx:

```
sudo nano /etc/nginx/sites-available/streaming_dashboard
```

Adicione o seguinte conteúdo:


```

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

    location / {
        root /opt/streaming-dashboard-frontend/dist;
        try_files $uri `uri/` /index.html;
    }

    location /api/ {
        proxy_pass http://unix:/tmp/gunicorn.sock;
        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;
    }
}

server {
    listen 80;
    server_name 4menapi.supersamsm.com.br;

    location / {
        proxy_pass http://unix:/tmp/gunicorn.sock;
        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;
    }
}

server {
    listen 80;
    server_name 4menlive.supersamsm.com.br:3002; # Para o servidor WebRTC

    location / {
        proxy_pass http://127.0.0.1:3002;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        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;
    }
}

```

Crie um link simbólico para habilitar o site:

```

sudo ln -s /etc/nginx/sites-available/streaming_dashboard /etc/nginx/sites-enabled

```

Teste a configuração do Nginx e reinicie o serviço:

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

7.2. Instalar Certbot e Obter Certificados SSL

Instale o Certbot e o plugin Nginx:

```
sudo snap install core  
sudo snap refresh core  
sudo snap install --classic certbot  
sudo ln -s /snap/bin/certbot /usr/bin/certbot  
sudo apt install python3-certbot-nginx -y
```

Obtenha os certificados SSL para ambos os domínios:

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

Siga as instruções do Certbot. Ele irá configurar automaticamente o Nginx para usar HTTPS e adicionar redirecionamentos HTTP para HTTPS.

Após a conclusão, o arquivo de configuração do Nginx (`/etc/nginx/sites-available/streaming_dashboard`) será atualizado para incluir as configurações SSL. Verifique-o para confirmar.

7.3. Configurar o Nginx para o Servidor WebRTC com SSL

O Certbot não configura automaticamente o SSL para a porta não padrão (3002) do WebRTC. Você precisará adicionar manualmente a configuração SSL para o servidor WebRTC no arquivo de configuração do Nginx.

Edite o arquivo de configuração do Nginx novamente:

```
sudo nano /etc/nginx/sites-available/streaming_dashboard
```

Localize o bloco `server` para `4menlive.supersamsm.com.br:3002` e modifique-o para incluir as configurações SSL, similar ao que o Certbot fez para as portas 80/443. O bloco final deve se parecer com isto:

```

server {
    listen 443 ssl;
    server_name 4menlive.supersamsm.com.br;

    ssl_certificate
/etc/letsencrypt/live/4menlive.supersamsm.com.br/fullchain.pem;
    ssl_certificate_key
/etc/letsencrypt/live/4menlive.supersamsm.com.br/privkey.pem;

    location / {
        root /opt/streaming-dashboard-frontend/dist;
        try_files $uri $uri/ /index.html;
    }

    location /api/ {
        proxy_pass http://unix:/tmp/gunicorn.sock;
        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;
    }

    location /socket.io/ {
        proxy_pass http://127.0.0.1:3002/socket.io/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        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;
    }
}

server {
    listen 443 ssl;
    server_name 4menapi.supersamsm.com.br;

    ssl_certificate
/etc/letsencrypt/live/4menapi.supersamsm.com.br/fullchain.pem;
    ssl_certificate_key
/etc/letsencrypt/live/4menapi.supersamsm.com.br/privkey.pem;

    location / {
        proxy_pass http://unix:/tmp/gunicorn.sock;
        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;
    }
}

# Redireccionamento HTTP para HTTPS para 4menlive.supersamsm.com.br
server {
    listen 80;
    server_name 4menlive.supersamsm.com.br;
    return 301 https://$host$request_uri;
}

# Redireccionamento HTTP para HTTPS para 4menapi.supersamsm.com.br
server {

```

```

listen 80;
server_name 4menapi.supersamsm.com.br;
return 301 https://$host$request_uri;
}

# Configuração para o servidor WebRTC (porta 3002)
server {
    listen 3002 ssl;
    server_name 4menlive.supersamsm.com.br;

    ssl_certificate
/etc/letsencrypt/live/4menlive.supersamsm.com.br/fullchain.pem;
    ssl_certificate_key
/etc/letsencrypt/live/4menlive.supersamsm.com.br/privkey.pem;

    location / {
        proxy_pass http://127.0.0.1:3002;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        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;
    }
}

```

Importante: O Certbot cria os certificados em `/etc/letsencrypt/live/seu-dominio/`. Certifique-se de que os caminhos `ssl_certificate` e `ssl_certificate_key` estejam corretos para o seu domínio.

Teste a configuração do Nginx e reinicie o serviço:

```

sudo nginx -t
sudo systemctl restart nginx

```

8. Configuração de DNS

Certifique-se de que seus registros DNS estejam configurados corretamente no seu provedor de domínio para apontar para o endereço IP público da sua VM:

- **Tipo:** A
- **Nome:** 4menlive
- **Valor:** SEU_IP_PUBLICO_DA_VM
- **Tipo:** A

- **Nome:** `4menapi`
- **Valor:** `SEU_IP_PUBLICO_DA_VM`

Se você estiver usando um roteador em sua rede local, certifique-se de que as portas `80`, `443` e `3002` (para o WebRTC) estejam redirecionadas para o endereço IP interno da sua VM.

9. Teste Final

Após concluir todos os passos, acesse seus domínios no navegador:

- **Frontend:** `https://4menlive.supersamsm.com.br`
- **Backend API:** `https://4menapi.supersamsm.com.br`

Verifique se o Dashboard carrega corretamente, se você consegue fazer login, gerenciar convidados e baixar as configurações do OBS. Teste também a funcionalidade de convidados, enviando um link para um amigo e verificando a conexão.

10. Solução de Problemas Comuns

- **"502 Bad Gateway" no Nginx:** Geralmente indica que o Gunicorn (backend Flask) não está rodando ou não está acessível. Verifique os logs do Supervisor (`/var/log/streaming_dashboard_backend.err.log` e `.out.log`) e o status do Supervisor (`sudo supervisorctl status streaming_dashboard_backend`).
- **"404 Not Found" no Frontend:** Verifique se os arquivos do frontend estão na pasta correta (`/opt/streaming-dashboard-frontend/dist`) e se a configuração `root` no Nginx está correta.
- **Problemas com SSL:** Verifique os logs do Certbot (`/var/log/letsencrypt/`) e certifique-se de que os caminhos dos certificados no Nginx estão corretos.
- **Problemas com WebRTC:** Verifique os logs do servidor WebRTC (`/var/log/webrtc_server.err.log` e `.out.log`) e o status do Supervisor (`sudo supervisorctl status webrtc_server`). Certifique-se de que a porta `3002` está aberta e que o Nginx está proxyando corretamente para ela.

- **Permissões:** Certifique-se de que o usuário Nginx (`www-data`) tenha permissão para ler os arquivos do frontend e os sockets do Gunicorn. Se necessário, ajuste as permissões com `sudo chown -R www-data:www-data /opt/streaming-dashboard-frontend/dist` e `sudo chmod -R 755 /opt/streaming-dashboard-frontend/dist`.
-

Autor: Manus AI **Data:** 19 de Julho de 2025