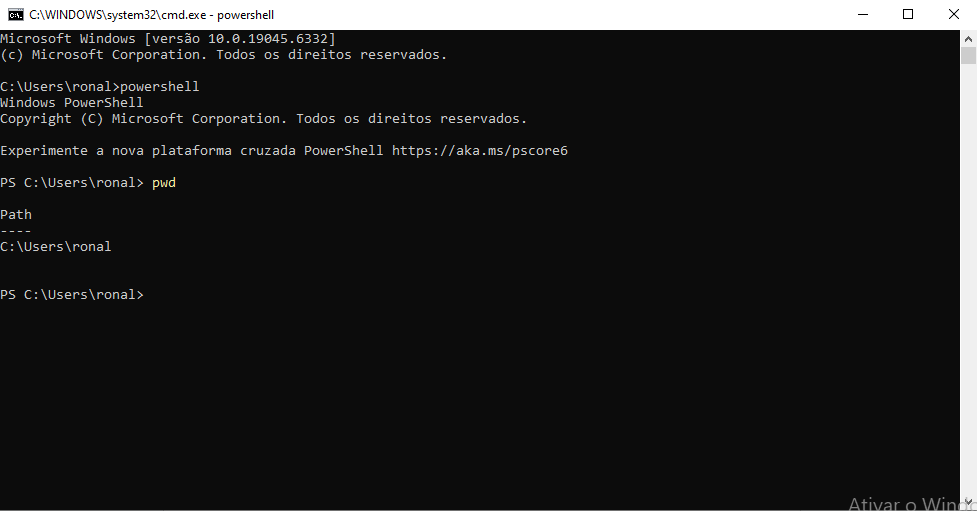
**Módulo 1: Comandos Básicos e Navegação no Sistema** 1. **Navegação e Visualização de Diretórios**

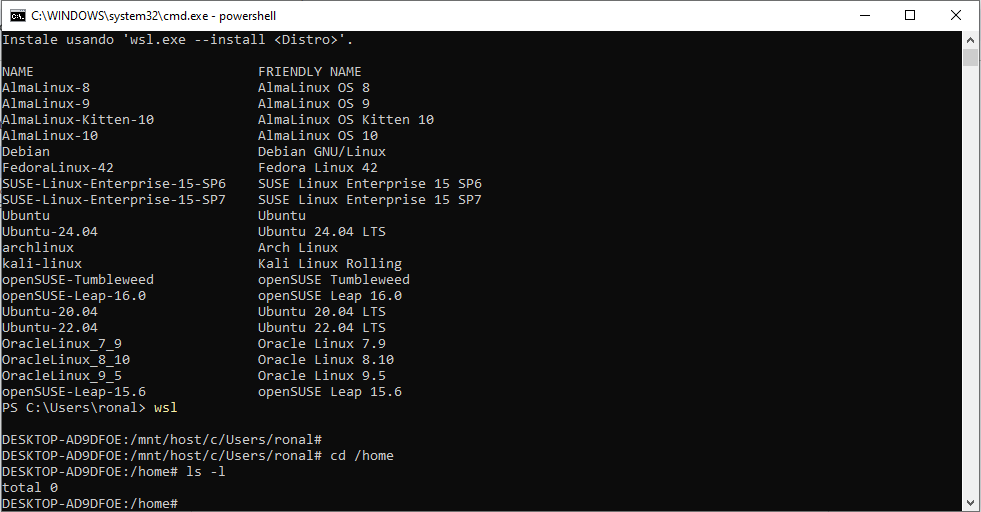
Verifique o diretório atual:

pwd



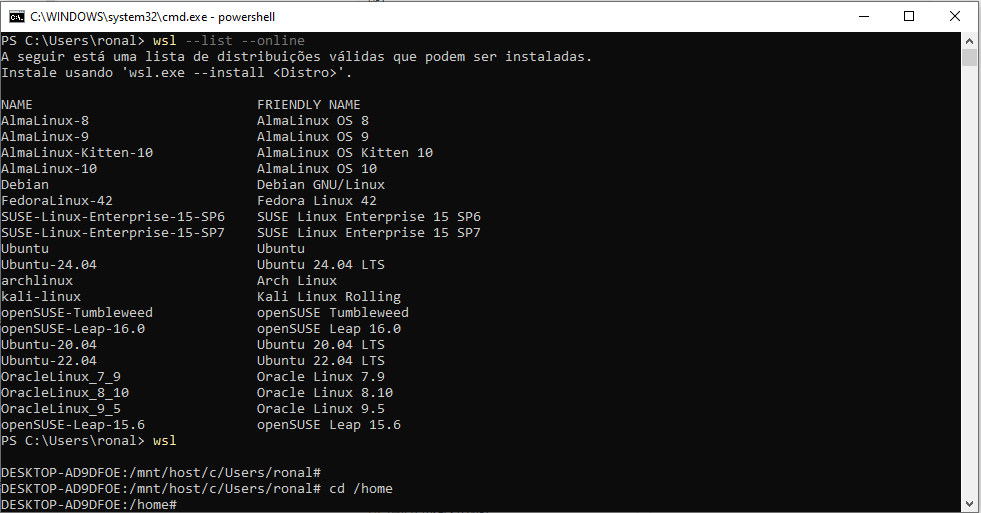
Liste arquivos e pastas no diretório atual:

ls -l



Entre no diretório /home:

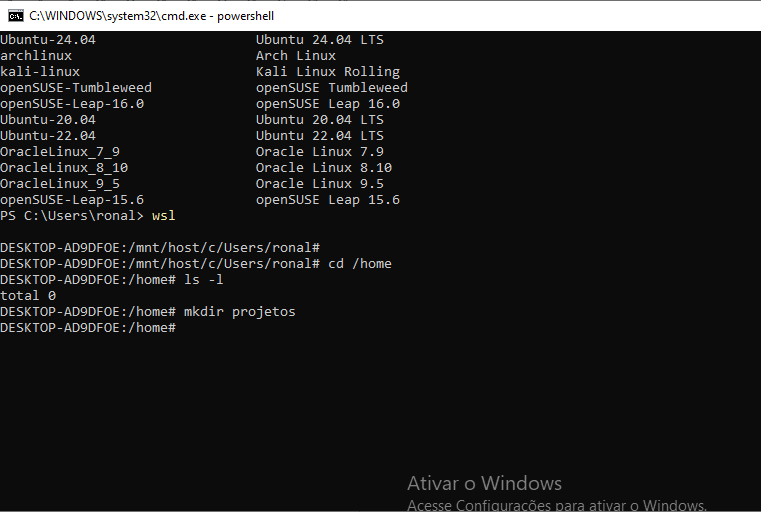
cd /home



2. **Manipulação de Arquivos e Diretórios**

Crie um novo diretório chamado projetos:

mkdir projetos

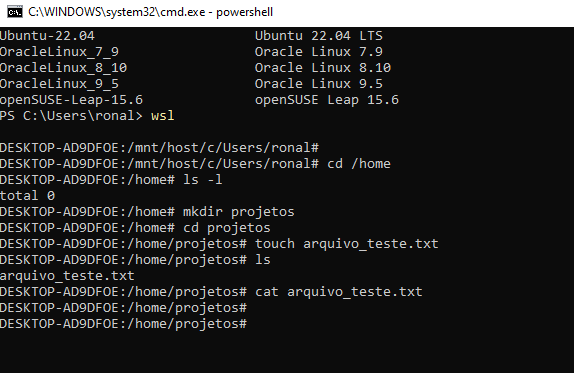


Mude para o diretório projetos e crie um arquivo de teste:

cd projetos

touch arquivo\_teste.txt

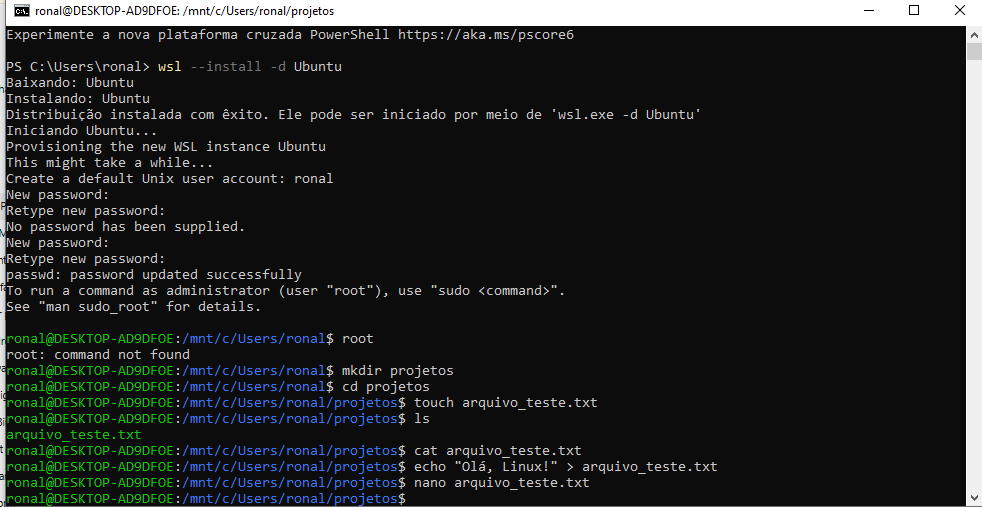
Visualize o conteúdo de arquivo\_teste.txt (ainda vazio): cat arquivo\_teste.txt

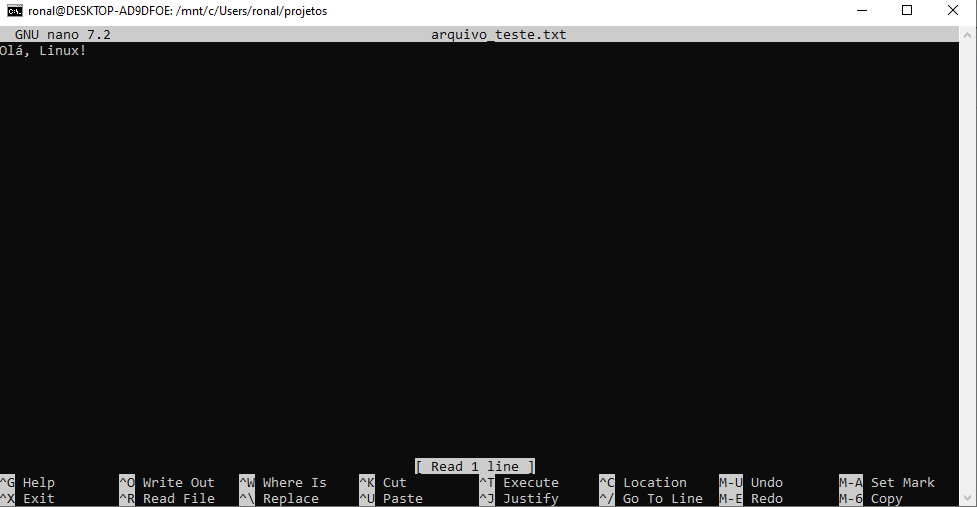


3. **Editar Arquivos com echo e nano**

Adicione texto ao arquivo:

echo "Olá, Linux!" > arquivo\_teste.txt Abra

o arquivo no editor nano para editar manualmente: nano arquivo\_teste.txt



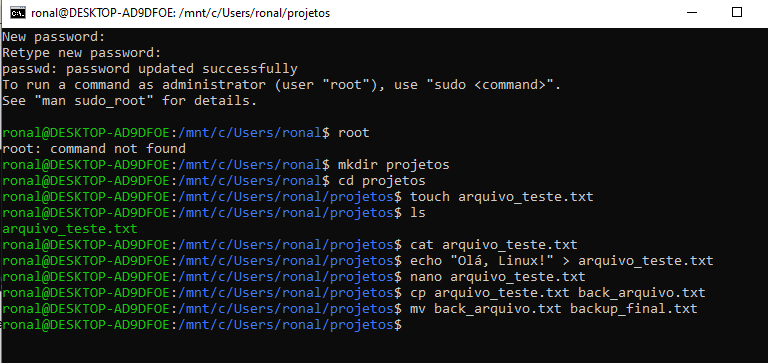
**Módulo 2: Gerenciamento de Arquivos, Permissões e Usuários** 1. **Copiar e Mover Arquivos**

Copie arquivo\_teste.txt para backup\_arquivo.txt:

cp arquivo\_teste.txt backup\_arquivo.txt

Renomeie o arquivo backup\_arquivo.txt para backup\_final.txt:

mv backup\_arquivo.txt backup\_final.txt



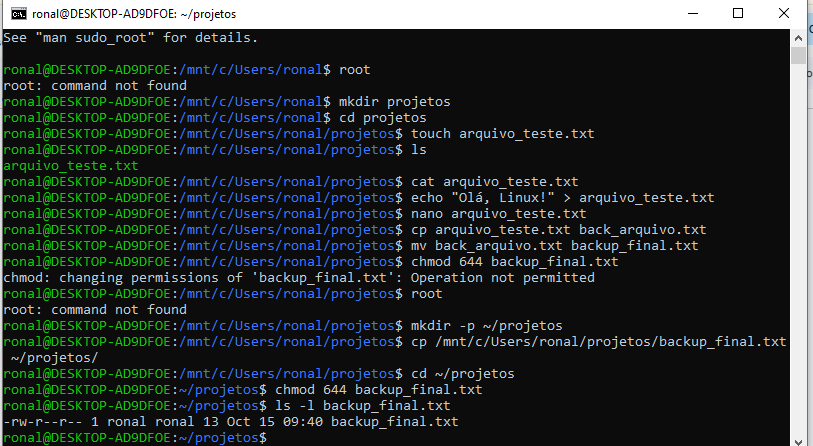
2. **Gerenciar Permissões**

Altere as permissões de backup\_final.txt para leitura e escrita pelo proprietário e leitura para outros:

chmod 644 backup\_final.txt

Verifique as permissões:

ls -l backup\_final.txt



3. **Criação e Gerenciamento de Usuários**

Crie um novo usuário chamado aluno:

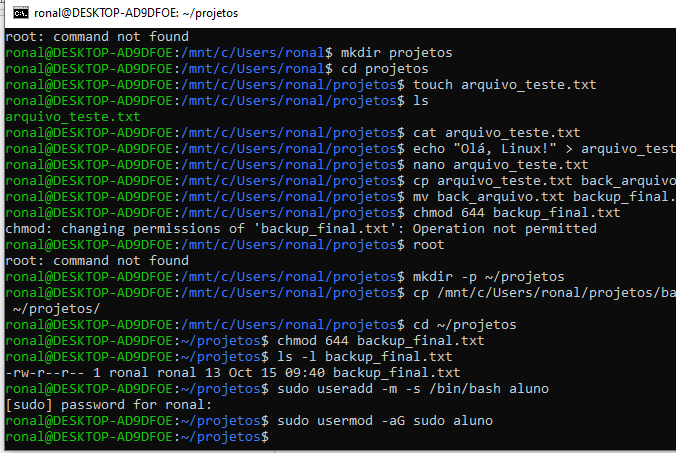
sudo useradd -m -s /bin/bash aluno

Defina uma senha para o novo usuário:

sudo passwd aluno

Adicione aluno ao grupo sudo:

sudo usermod -aG sudo aluno

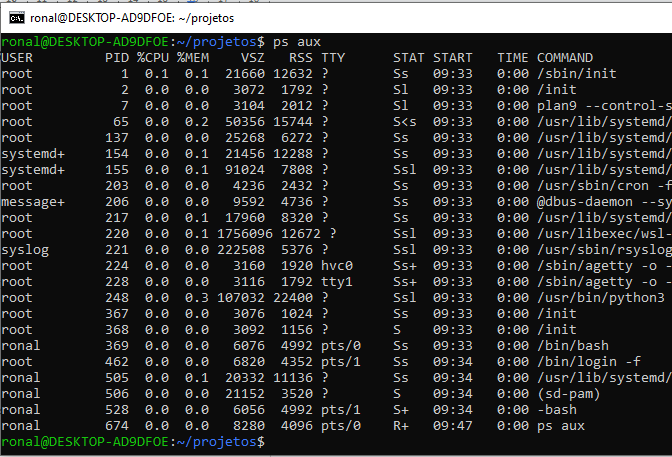


**Módulo 3: Processos, Tarefas e Configuração do SSH**

1. **Monitoramento de Processos**

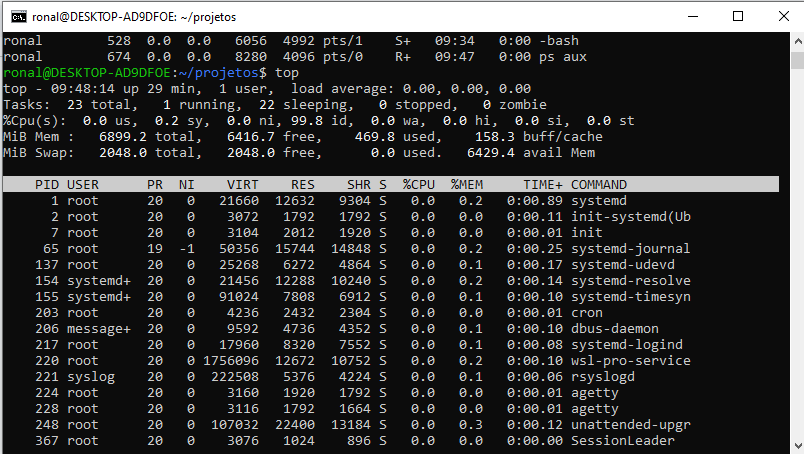
Liste processos ativos:

ps aux

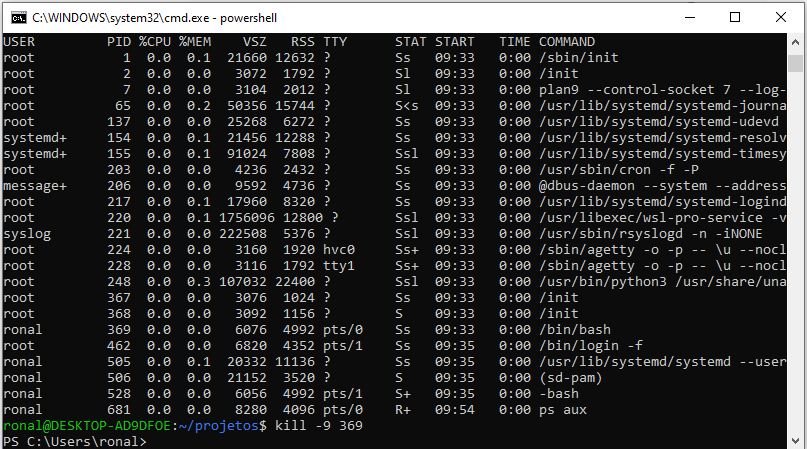


Execute top para monitorar o uso de recursos:

top



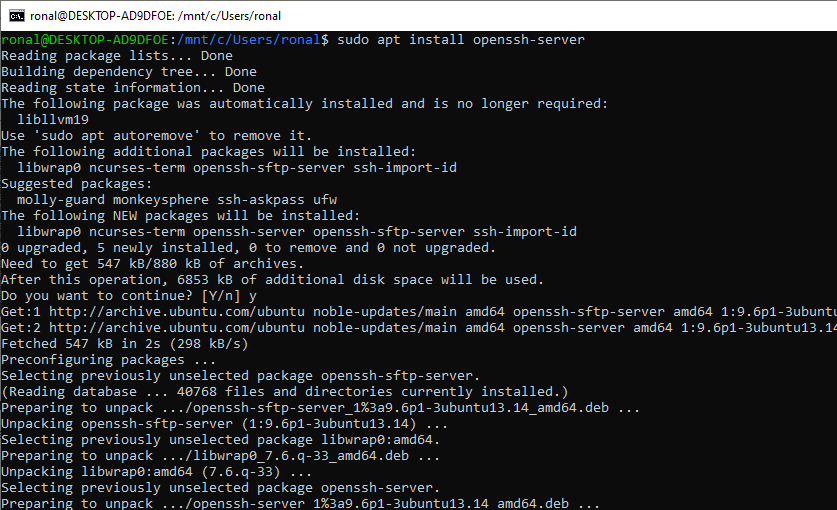
Finalize um processo (substitua PID pelo ID do processo): kill -9 PID



2. **Configuração do SSH**

Instale o OpenSSH (se necessário):

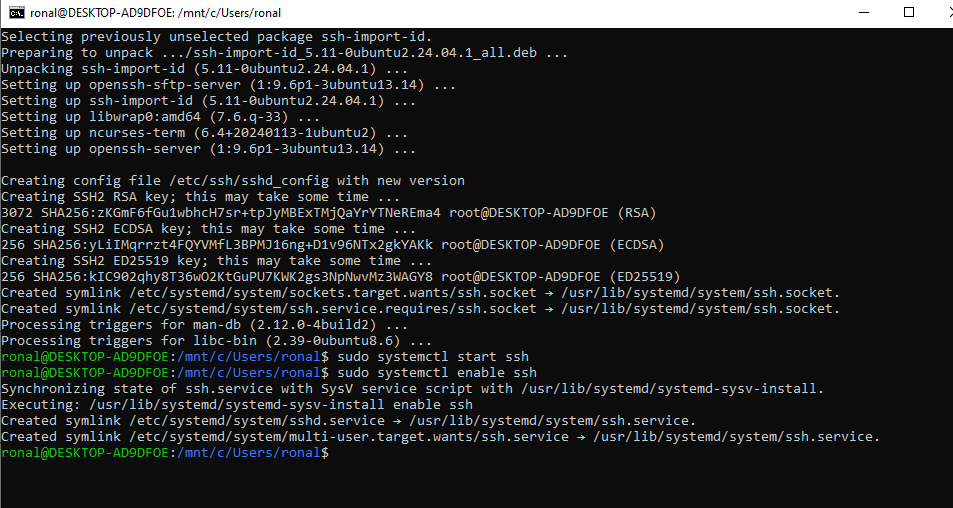
sudo apt install openssh-server



Inicie e ative o SSH:

sudo systemctl start ssh

sudo systemctl enable ssh



OU

sudo service ssh start

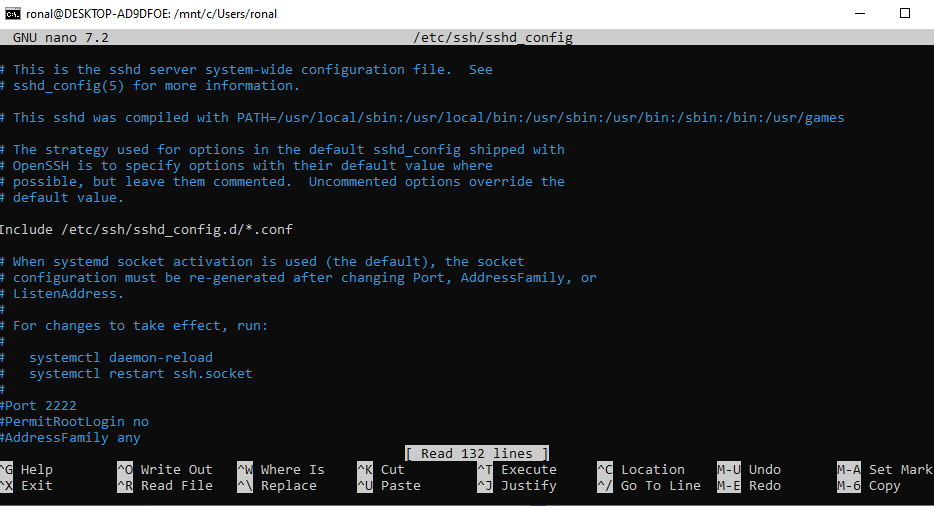
sudo service ssh enable

Edite as configurações SSH para maior segurança:

sudo nano /etc/ssh/sshd\_config

■ Alterar a porta (ex: Port 2222).

■ Desativar login root (PermitRootLogin no).

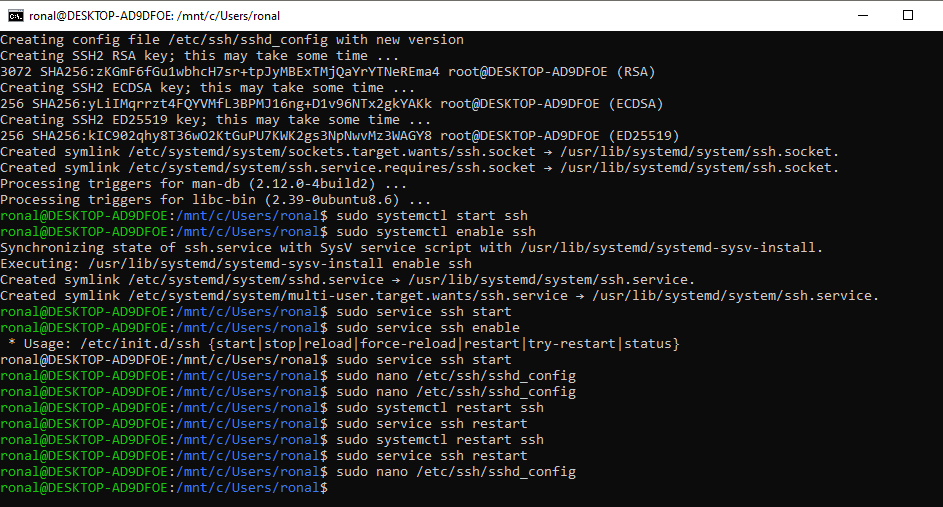


Reinicie o SSH para aplicar mudanças:

sudo systemctl restart ssh

OU

sudo service ssh restart



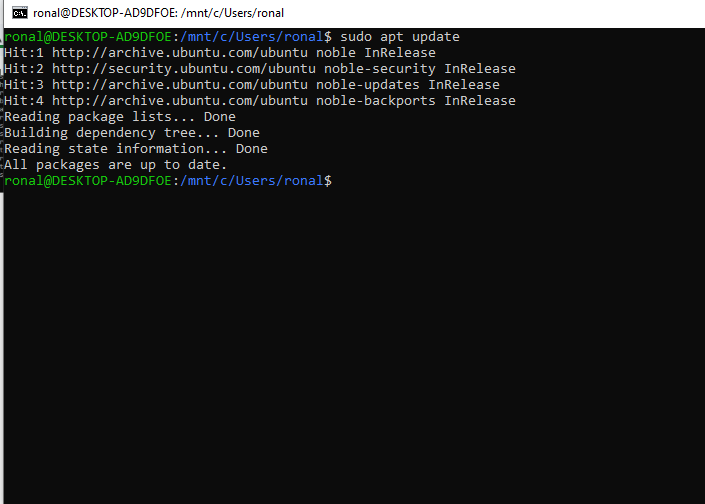
**Módulo 4: Gerenciamento de Pacotes**

1. **Atualizar o Sistema**

Atualize os repositórios e o sistema:

sudo apt update

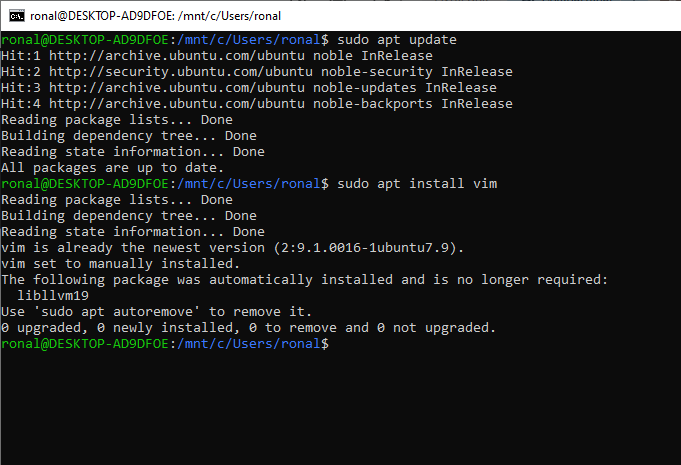
sudo apt upgrade



2. **Instalar e Remover Pacotes**

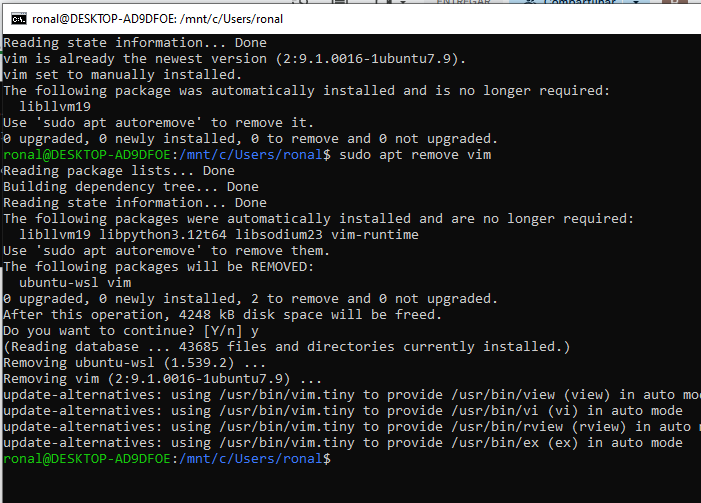
Instale o editor vim:

sudo apt install vim



Remova o editor vim:

sudo apt remove vim

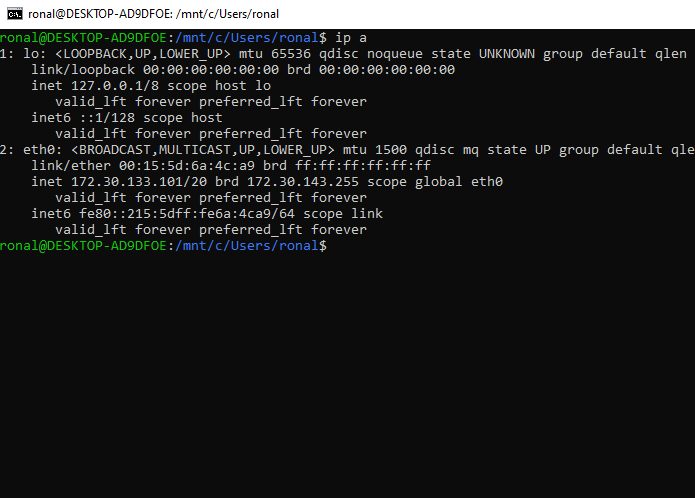


**Módulo 5: Configuração e Diagnóstico de Rede**

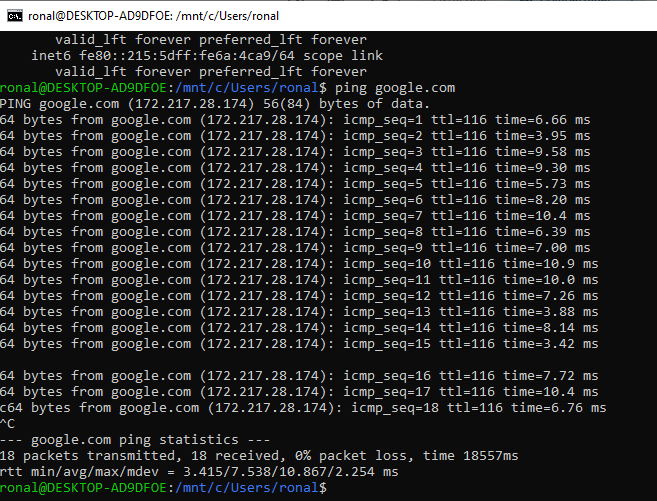
1. **Configuração de Rede**

Exiba as interfaces de rede e seu status:

ip a



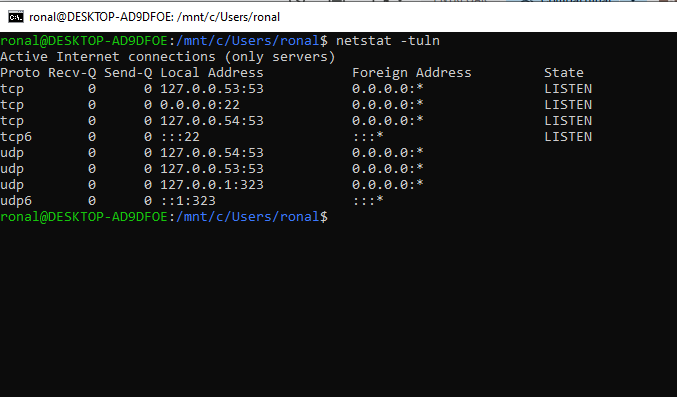
Ping para verificar conectividade com um servidor externo (ex: Google): ping google.com



2. **Diagnóstico de Rede**

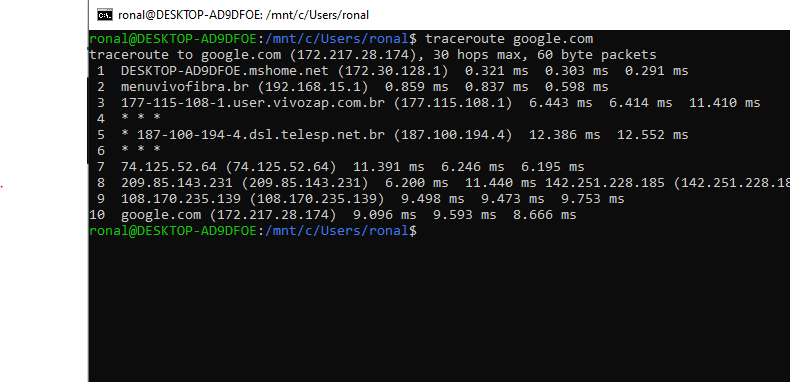
Liste portas abertas e conexões ativas:

netstat -tuln



Rastreie a rota até um destino:

traceroute google.com



**Módulo 6: Shell Scripting Básico**

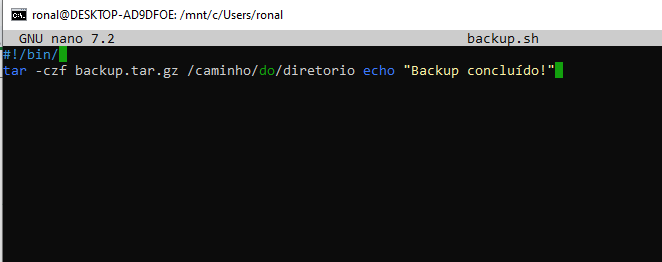
1. **Criar um Script Simples**

Crie um novo arquivo de script chamado backup.sh: nano backup.sh

Insira o seguinte código no script:

#!/bin/

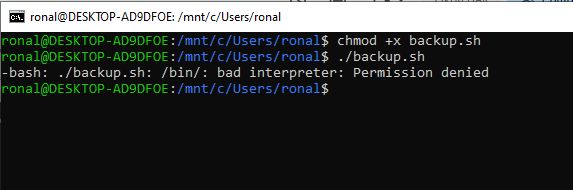
tar -czf backup.tar.gz /caminho/do/diretorio echo "Backup concluído!"



Salve, feche e execute o script:

chmod +x backup.sh

./[backup.sh](http://backup.sh)



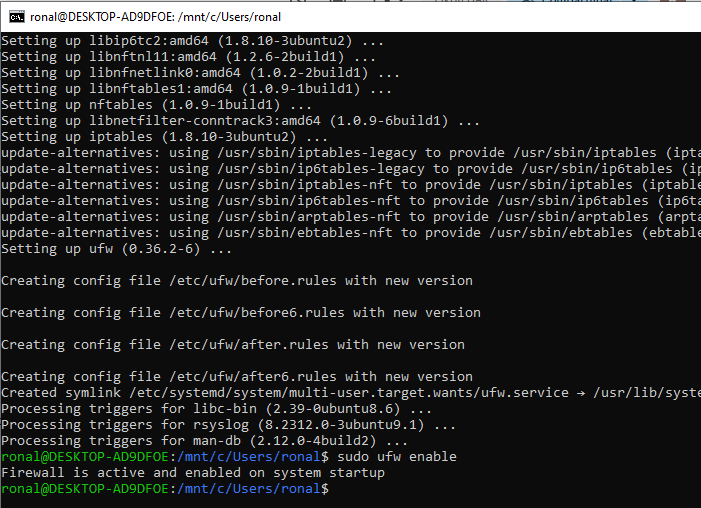
**Módulo 7: Segurança no Linux**

1. **Configuração Básica de Firewall com UFW**

Instale e habilite o UFW (se necessário):

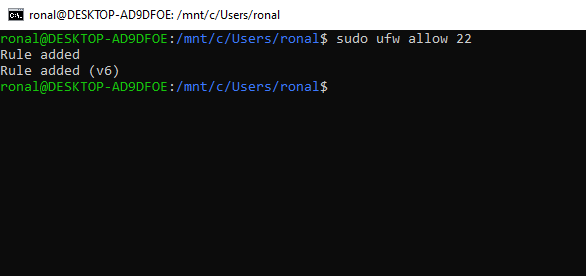
sudo apt install ufw

sudo ufw enable



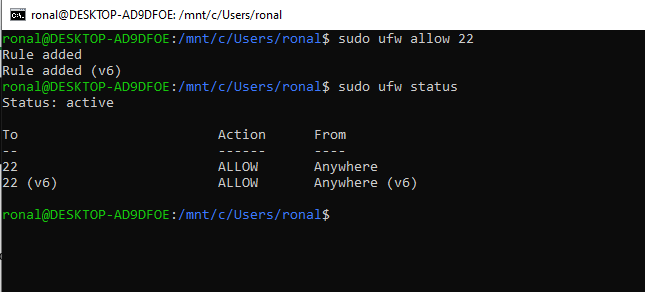
Permita o tráfego na porta 22 para SSH:

sudo ufw allow 22



Exiba o status do UFW para verificar as regras:

sudo ufw status

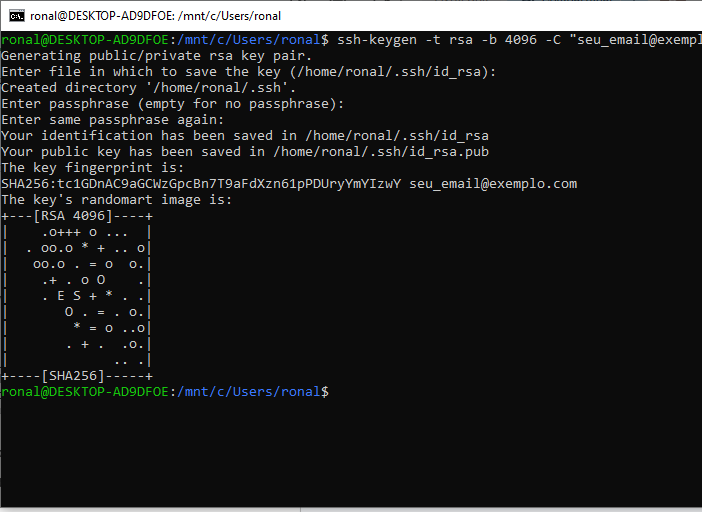


2. **Configuração Avançada de Segurança no SSH**

○ Configure chaves SSH:

No cliente (máquina local), abra o terminal e gere uma chave SSH com o seguinte comando:

ssh-keygen -t rsa -b 4096 -C "seu\_email@exemplo.com"



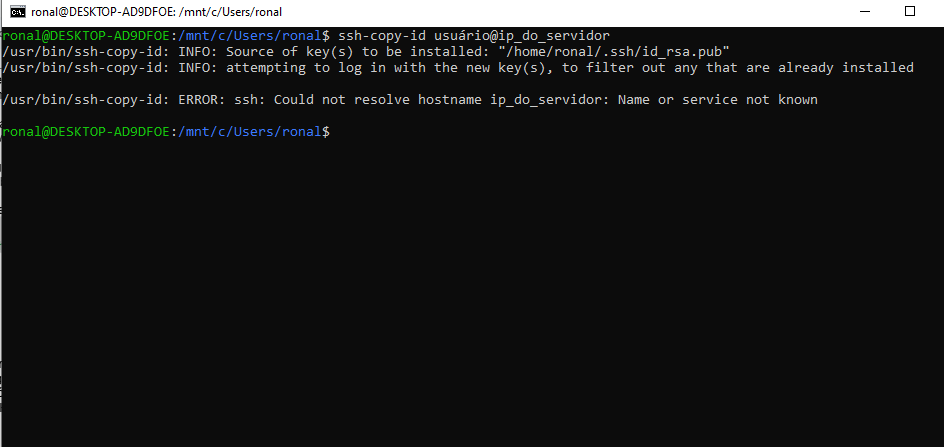
Pressione Enter para salvar a chave no local padrão (~/.ssh/id\_rsa), ou especifique um caminho se preferir.

Quando solicitado, você pode configurar uma passphrase (senha para proteger a chave privada). Isso adiciona uma camada extra de segurança:

Digite uma senha, ou pressione Enter para deixar em branco (não recomendado para ambientes de produção).

Use o comando ssh-copy-id para transferir a chave pública para o servidor:

ssh-copy-id usuário@ip\_do\_servidor



● Substitua o usuário pelo nome do usuário que você está configurando no servidor e ip do servidor pelo endereço IP ou domínio do servidor.

● Ao executar este comando, você precisará digitar a senha do usuário remoto. Isso adiciona automaticamente a chave pública ao arquivo ~/.ssh/authorized\_keys no servidor.

Configurar SSH para Usar Apenas Autenticação por Chave (Opcional, para maior segurança)

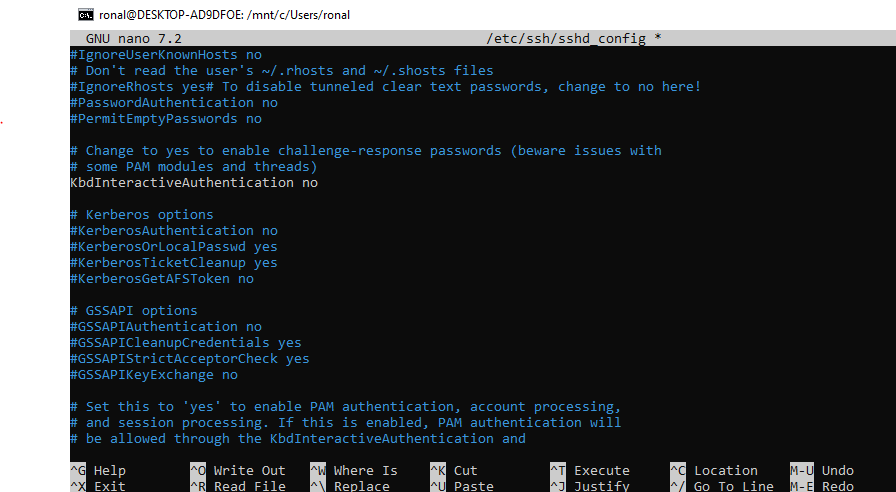
Após a chave ser configurada e testada, você pode desativar a autenticação por senha no SSH para garantir maior segurança.

No servidor, edite o arquivo de configuração SSH:

sudo nano /etc/ssh/sshd\_config

Localize a linha com PasswordAuthentication e altere para no:

PasswordAuthentication no



Reinicie o serviço SSH para aplicar as alterações:

sudo systemctl restart ssh

Conectar ao Servidor Usando SSH com Chave

Agora, você pode se conectar ao servidor usando SSH sem precisar da senha (se a chave estiver configurada corretamente):

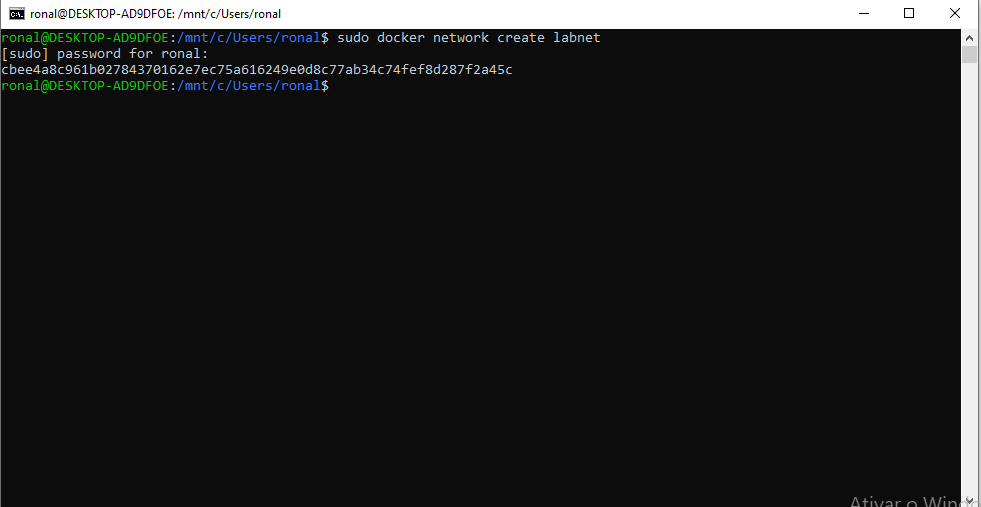
ssh usuário@ip\_do\_servidor

Caso tenha configurado uma passphrase para a chave, você será solicitado a inseri-la durante a conexão.

■ **PREPARAÇÃO DO AMBIENTE DOCKER (PASSO A PASSO)** Siga exatamente na ordem para evitar erros.

# 1■■ Criar rede para os contêineres

docker network create labnet



# 2■■ Criar o arquivo Dockerfile (no diretório atual)

nano Dockerfile

# Cole o conteúdo abaixo e salve (CTRL+O, ENTER, CTRL+X)

FROM ubuntu:22.04

ENV DEBIAN\_FRONTEND=noninteractive

RUN apt-get update && apt-get install -y \

nano vim net-tools iputils-ping traceroute curl wget tar \

openssh-server openssh-client \

iptables iptables-persistent \

nginx openssl

# Usuário padrão do lab

RUN useradd -m -s /bin/bash aluno && echo "aluno:123" | chpasswd

# Pastas de serviço

RUN mkdir -p /var/run/sshd /var/www/html

CMD ["/bin/bash"]

# 3■■ Construir a imagem

docker build -t lab-linux .

# 4■■ (re)Criar o contêiner do servidor com portas publicadas (SSH 2222, HTTP 8080, HTTPS 8443) # Se já existir, remova-o primeiro:

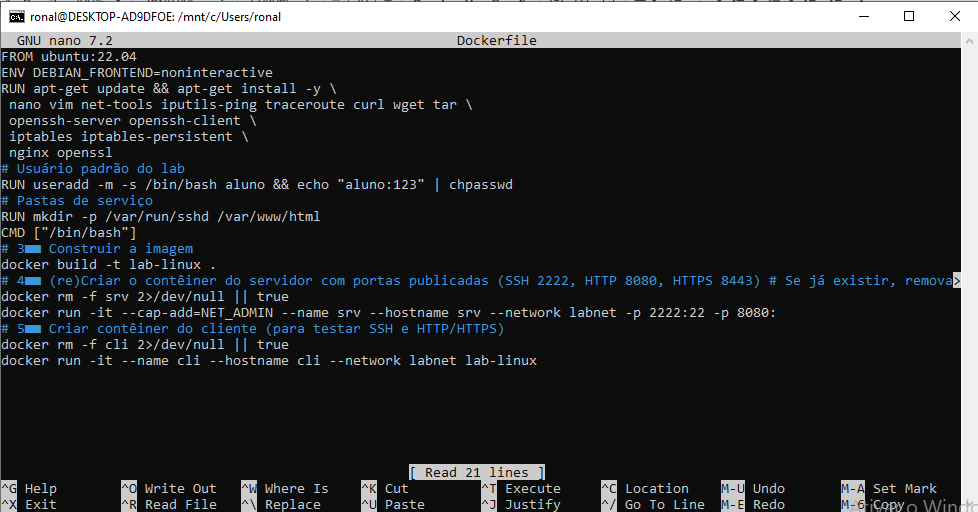
docker rm -f srv 2>/dev/null || true

docker run -it --cap-add=NET\_ADMIN --name srv --hostname srv --network labnet -p 2222:22 -p 8080:

# 5■■ Criar contêiner do cliente (para testar SSH e HTTP/HTTPS)

docker rm -f cli 2>/dev/null || true

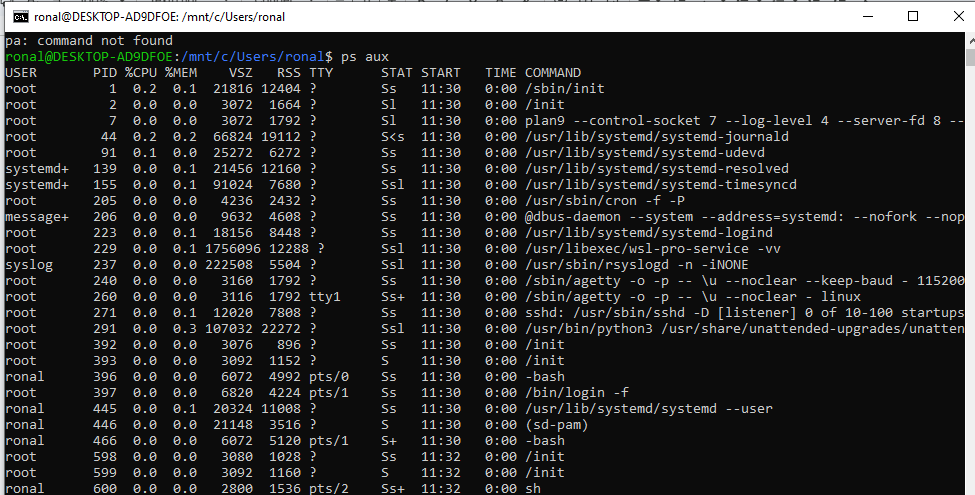
docker run -it --name cli --hostname cli --network labnet lab-linux



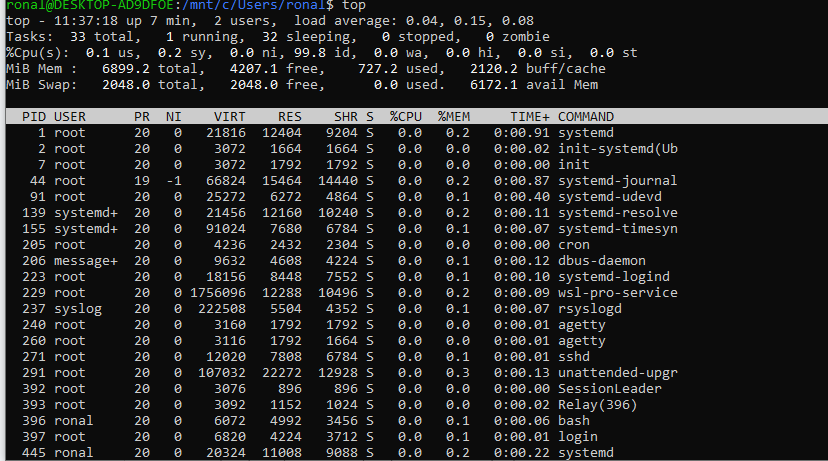
**Módulo 3: Processos e SSH (dois contêineres: srv e cli)**

# 3.1 Processos

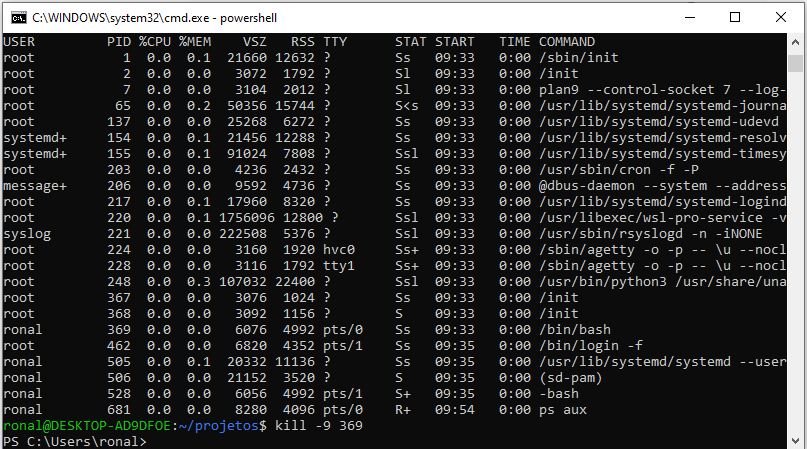
ps aux



top



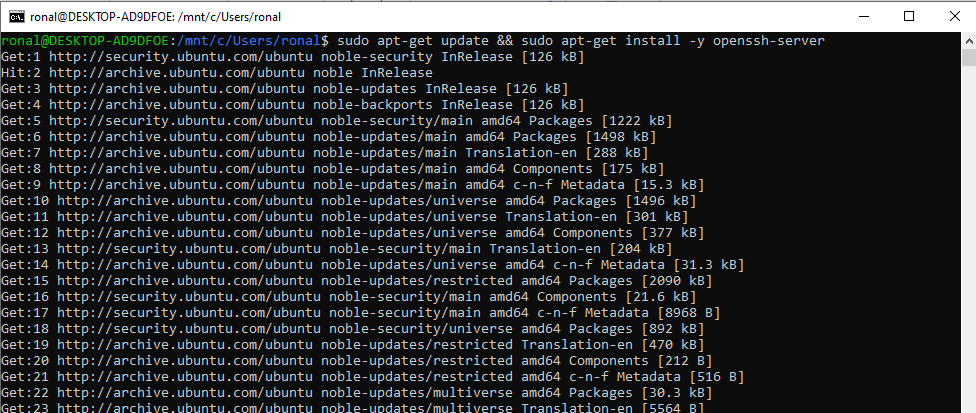
kill -9 PID



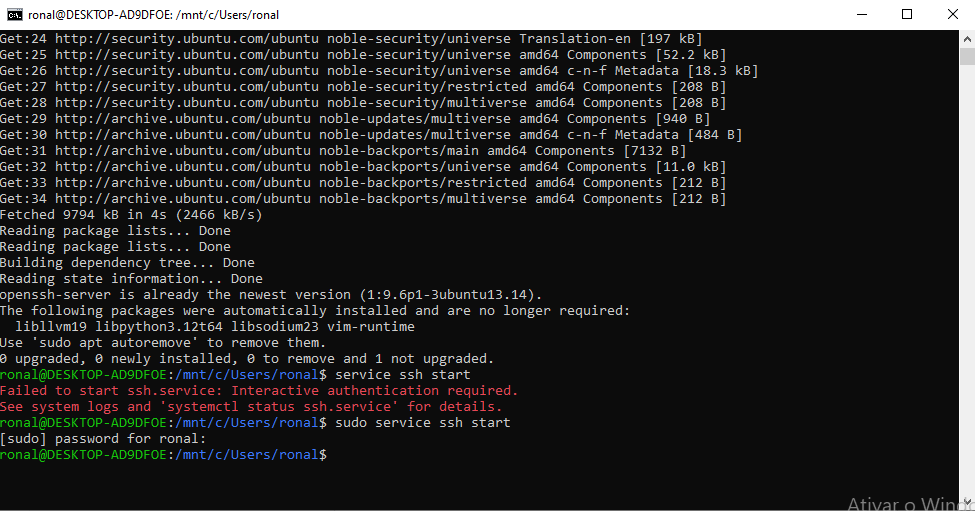
# 3.2 SSH

# No servidor (srv):

apt-get update && apt-get install -y openssh-server



service ssh start

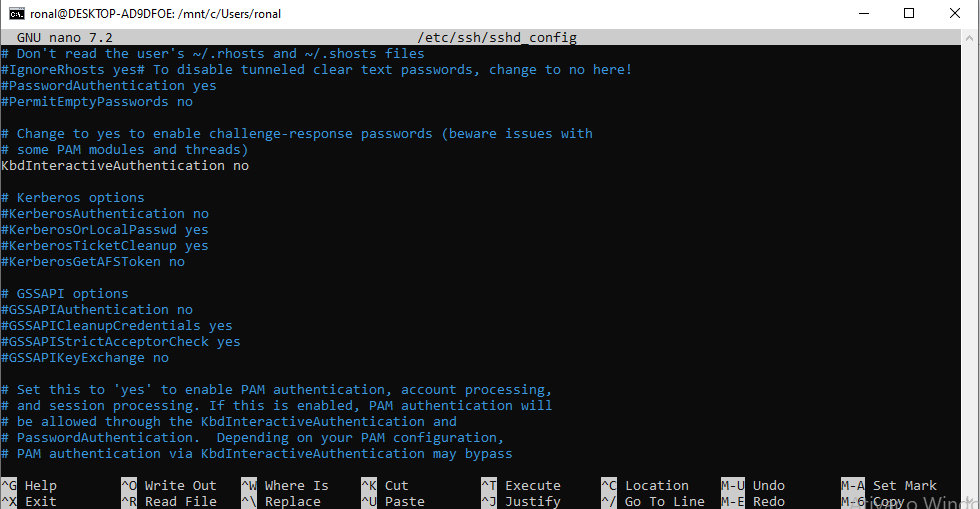


nano /etc/ssh/sshd\_config

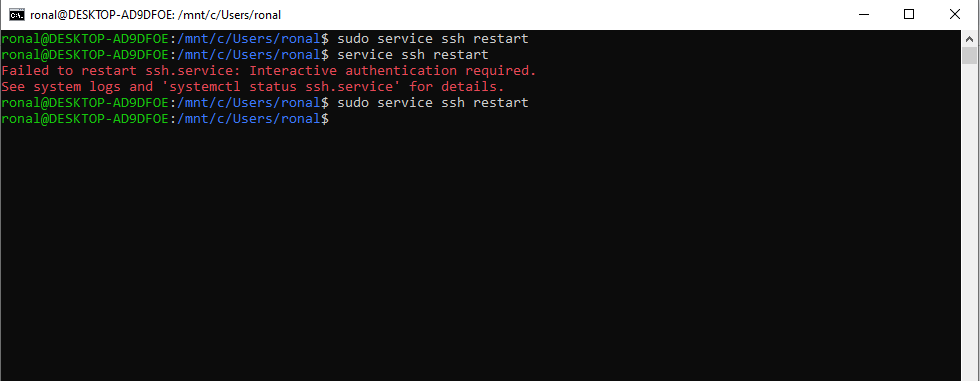
# Recomendações:

# PermitRootLogin no

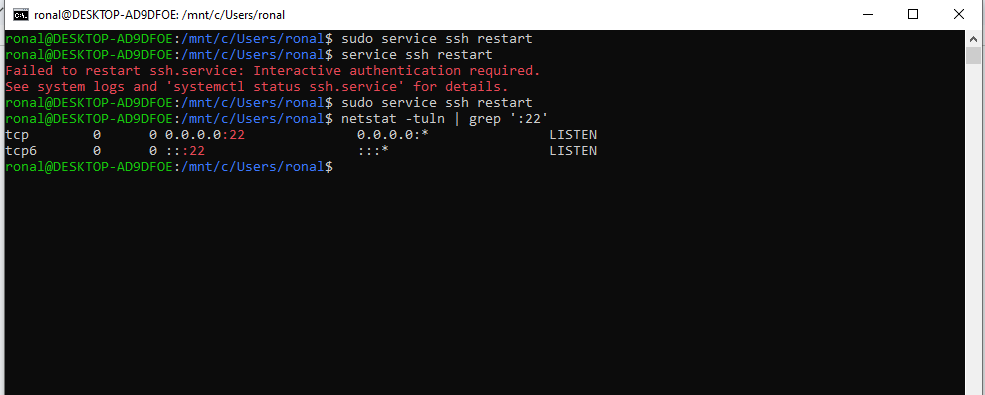
# PasswordAuthentication yes (mantenha yes para primeiro acesso)



service ssh restart

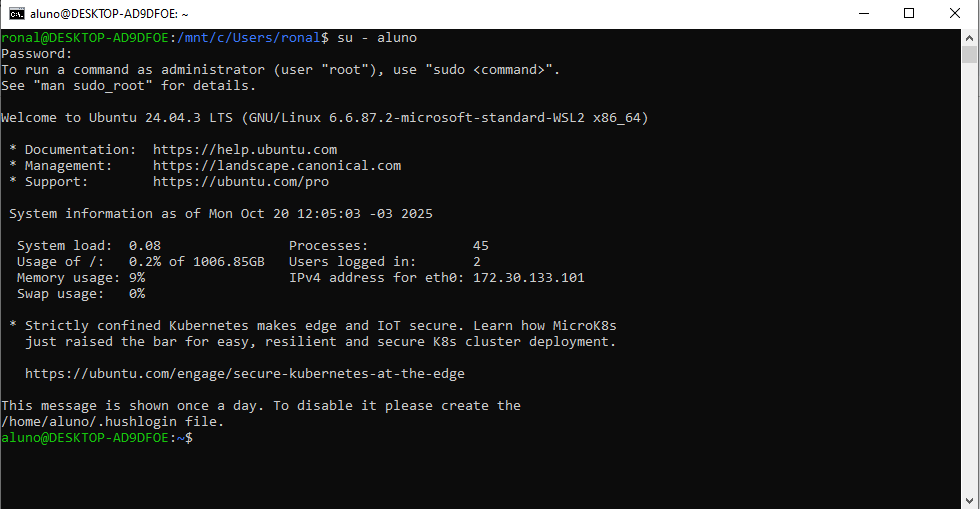


netstat -tuln | grep ':22'

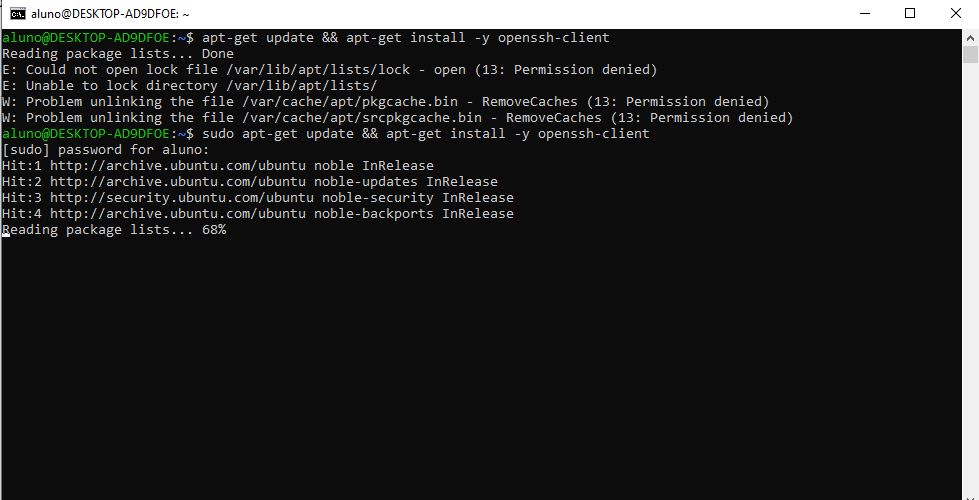


# No cliente (cli) como 'aluno':

su - aluno



apt-get update && apt-get install -y openssh-client



ssh-keygen -t rsa -b 4096 -C "aluno@example.com" -N "" -f ~/.ssh/id\_rsa

ssh-copy-id aluno@srv # senha: 123456

ssh aluno@srv

exit

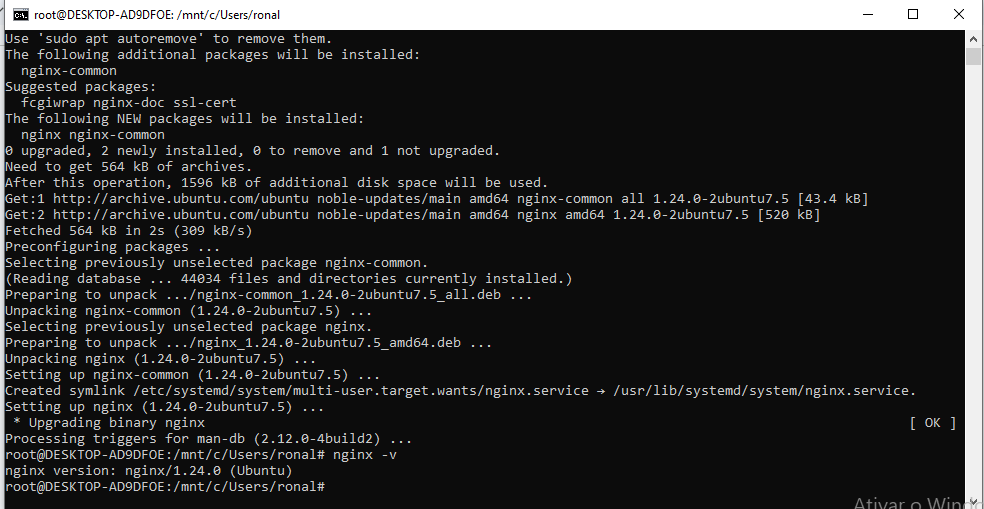


**Módulo 8: Servidor Web (Nginx) + HTTPS**

# No servidor (srv)

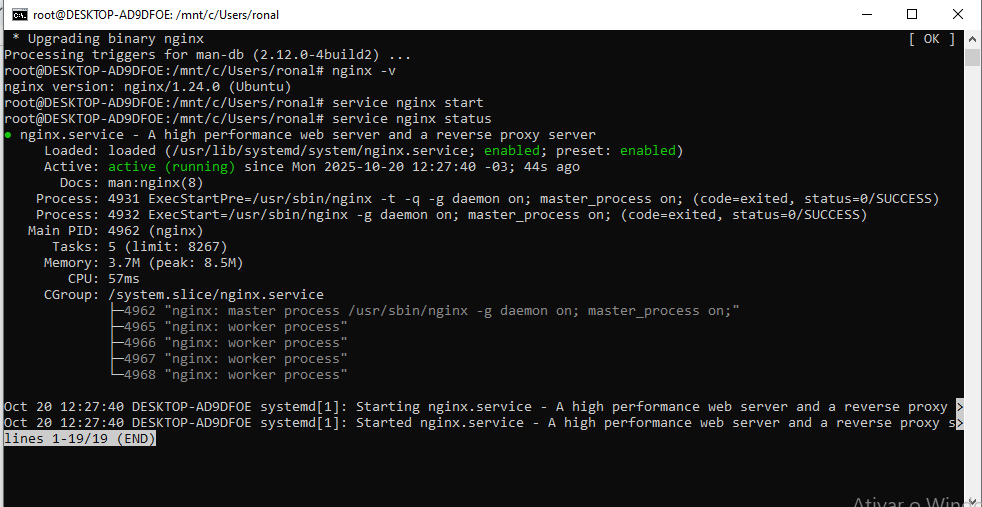
apt-get update && apt-get install -y nginx

nginx -v



service nginx start

service nginx status

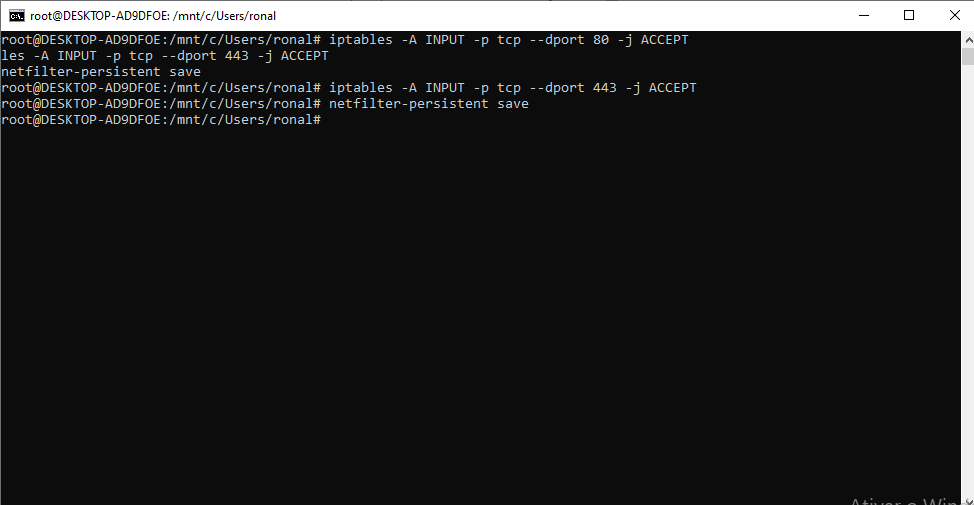


# Liberar HTTP/HTTPS

iptables -A INPUT -p tcp --dport 80 -j ACCEPT

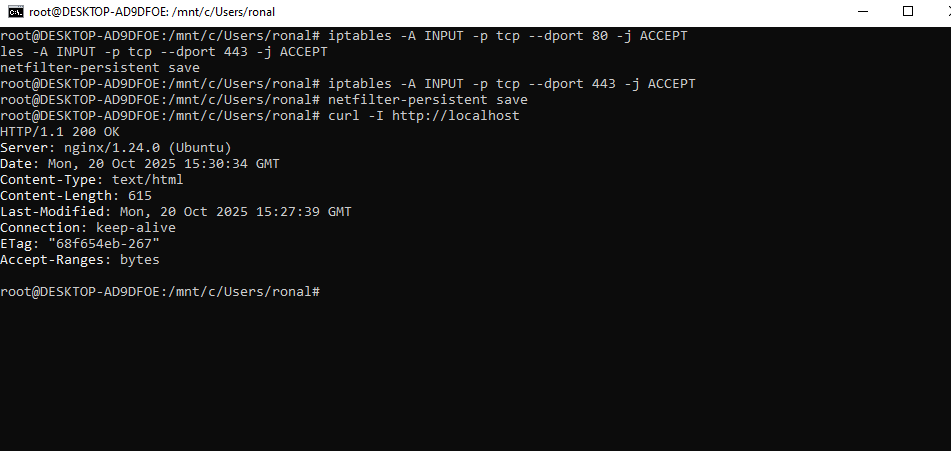
iptables -A INPUT -p tcp --dport 443 -j ACCEPT

netfilter-persistent save

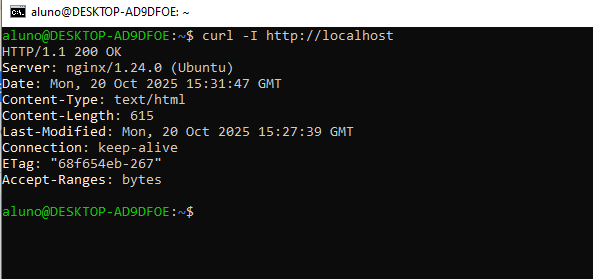


# Teste HTTP local (no srv) e externo (no cli/host)

curl -I http://localhost



# Do host: <http://localhost:8080>



# Site simples

mkdir -p /var/www/meusite.com

echo "<h1>Meu Site</h1>" > /var/www/meusite.com/index.html nano /etc/nginx/sites-available/meusite.com

# Conteúdo:

# server {

# listen 80;

# server\_name \_;

# root /var/www/meusite.com;

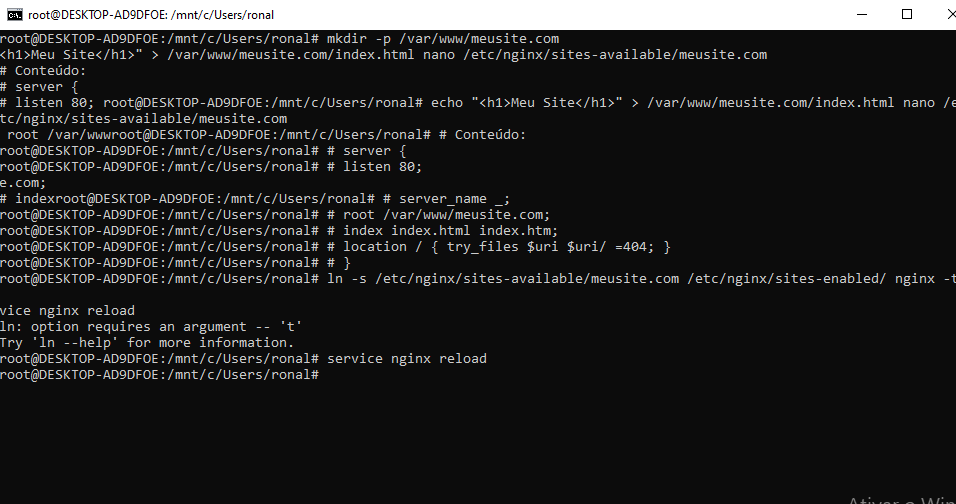
# index index.html index.htm;

# location / { try\_files $uri $uri/ =404; }

# }

ln -s /etc/nginx/sites-available/meusite.com /etc/nginx/sites-enabled/ nginx -t

service nginx reload



# Teste HTTP (no cli)

curl -I http://srv

# Do host: curl -I http://localhost:8080

#############################################

# A) HTTPS com certificado AUTOASSINADO (OFFLINE)

#############################################

# 1) Gerar chave e certificado (válido por 365 dias)

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/meusite.key -out

# 2) Configurar servidor HTTPS (novo bloco)

nano /etc/nginx/sites-available/meusite\_ssl

# Conteúdo:

# server {

# listen 443 ssl;

# server\_name \_;

# root /var/www/meusite.com;

# index index.html index.htm;

# ssl\_certificate /etc/ssl/certs/meusite.crt;

# ssl\_certificate\_key /etc/ssl/private/meusite.key;

# location / { try\_files $uri $uri/ =404; }

# }

ln -s /etc/nginx/sites-available/meusite\_ssl /etc/nginx/sites-enabled/

nginx -t

service nginx reload

# 3) Teste HTTPS

curl -k -I https://localhost

# Do host (porta publicada): https://localhost:8443 (use -k por ser autoassinado) curl -k -I https://localhost:8443

#############################################

# B) HTTPS com Let's Encrypt (OPCIONAL, requer domínio público)

#############################################

# Pré-requisitos:

# - Domínio apontado para o IP público da sua máquina

# - Publicar portas 80 e 443 diretamente (ex.: -p 80:80 -p 443:443)

# - DNS propagado

# 1) Instalar Certbot + plugin Nginx

apt-get update && apt-get install -y certbot python3-certbot-nginx

# 2) Emitir e configurar automaticamente

# Substitua "meusite.com" pelo seu domínio

certbot --nginx -d meusite.com -d www.meusite.com

# 3) Testar renovação

certbot renew --dry-run

# 4) Teste HTTPS via navegador: https://seu\_dominio