Projeto formação essencial em Tecnologia da Informação.

Mês 2 - Linux Essencial para TI

Autor: André Luis de Freitas Ribeiro

Curso: Gestão de Tecnologia da Informação

Ferramentas: Oracle VirtualBox, Ubuntu Server 24.04

Período: `15/01/2026 - 24/01/2026`

---

1. Objetivo

Este relatório tem como objetivo a criação e configuração de um servidor Linux básico, utilizando uma máquina virtual que hospeda o sistema operacional **Ubuntu Server**, previamente criada no módulo anterior do projeto. Além disso, contempla a aplicação de conceitos de **Administração Básica de Sistemas Linux**, com foco no aprendizado prático de fundamentos essenciais, como gerenciamento de usuários e permissões, estrutura de diretórios, administração de pacotes e gerenciamento de serviços.

---

2. Escopo

2.1 O que está incluso

1. **Domínio de competências técnicas em Linux**
    - Adquirir autonomia na utilização de sistema Linux
    - Utilizar o Terminal de forma eficiente
2. **Conteúdos de administração de sistema Linux**
    - Compreender estruturas de diretórios
    - Gerenciar grupos,usuários e permissões de arquivos e diretórios
    - Utilizar o gerenciador de pacotes `apt` para instalar, atualizar e remover software.
    - Administração de serviços do sistema com `systemctl`.
3. **Realização do Projeto - Servidor Linux Básico**
    - Configurar um Servidor Linux do zero
        - Criação de usuários
        - Configuração de SSH
        - Configuração de atualizações automáticas
        - Implementar firewall básico com UFW (Uncomplicated Firewall)
    - Entrega de checklist e relatório técnico

2.2 O que não está incluso

1. Configuração de serviços avançados
2. Implementação de monitoramento ou sistemas de backup.
3. Configuração de redes complexas (VPN, VLANs, roteamento avançado).
4. Tópicos de segurança avançada.
5. Automação com ferramentas.

---

3. Ambiente Utilizado

Componente	Sistema Operacional	Processador	RAM	Armazenamento	Software de Virtualização
Ambiente Host	Windows 11	Intel(R) Processor U300 (1.20 GHz)	8 GB	256 GB	—
Ambiente Virtual	Ubuntu Server 24.04 LTS	1 CPU Alocada	2 GB	30 GB	Oracle VirtualBox

---

4. Etapas do Projeto

4.1 Fundamentos do Sistema

Estrutura de diretório Linux - Teoria

Foi realizada inicialmente uma análise e estudo dos conceitos relacionados aos **diretórios** do sistema Linux, com o objetivo de obter uma melhor compreensão da base necessária para a execução dos testes futuros.

Para isso, foi adotada uma metodologia de estudo que mescla a análise teórica do funcionamento dos **principais diretórios** (`/`, `/home`, `/bin`) com a utilização de serviços online voltados ao aprendizado e à realização de testes por meio de questões de banca, visando uma compreensão mais sólida e aplicada do conteúdo.

```
# /
# /home
# /bin
# /etc
# /var
```

Usuários e Grupos

Como primeiro passo, foi necessária a criação de um novo usuário definindo senha e seguindo os protocolos padrões, denominado `usuario_teste`, destinado à execução das etapas subsequentes do projeto. Essa abordagem permite a configuração direta de permissões e propriedades, garantindo maior controle do ambiente e possibilitando a simulação de cenários profissionais.

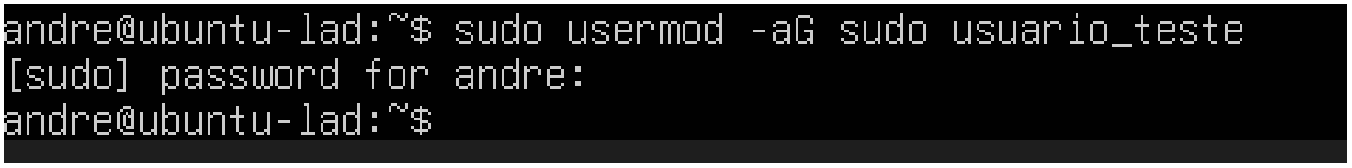
> **Comando Utilizado**

```
sudo adduser usuario_teste
```

Em seguido concedi a permissão administrativa ao usuário de teste criado, adicionando-o ao grupo `sudo`, permitindo a execução segura de comandos administrativos quando necessário.

> **Comando Utilizado**

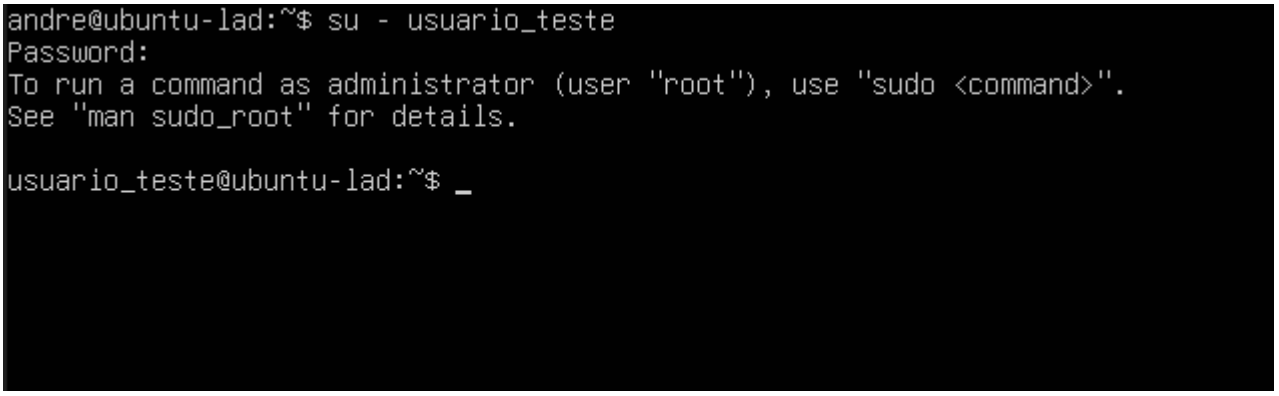
```
sudo usermod -aG sudo usuario_teste
```



Com a criação do usuário de teste completa realizei o teste de `login` para verificar se o acesso ao novo usuária estava ocorrendo da forma esperada.

> **Comando Utilizado**

```
su - usuario_teste
```



Permissões

Para realizar o teste de permissões inicialmente criei uma `pasta` simples para ser usada como base de acesso aos usuários.

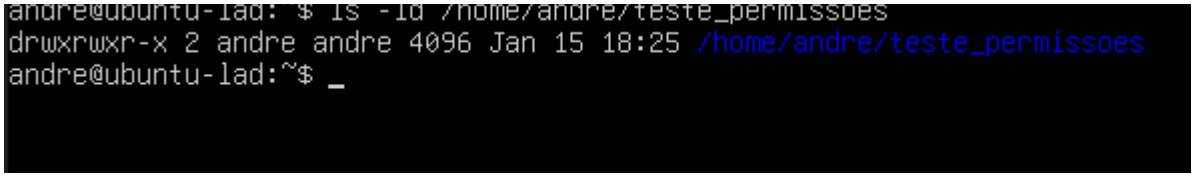
> Comando utilizado

```
mkdir /home/andre/teste_permissoes
```

Para a confirmação da permissão atual executei um comando de visualização da pasta.

> Comando utilizado

```
ls - ld /home/andre/teste_permissoes
```



> ***A saída indica o proprietário do diretório e o grupo, sendo ambos como `andre`, coerente com diretórios criado pelo usuário***

Como última etapa do teste de gerenciamento de permissões, foi realizada a alteração do proprietário e do grupo da pasta `teste_permissoes`. Por meio do comando `sudo chown`, foi possível definir o novo proprietário e o novo grupo associados ao diretório, especificando diretamente quais usuários e grupos passariam a ser responsáveis por seus arquivos e permissões.

> **Comando utilizado**

```
sudo chown usuario_teste:usuario_teste teste_permissoes
```

Em seguida foi alterado as permissões da pasta utilizando o comando `chmod 700` para restringir o acesso ao diretório `teste_permissoes`, garantindo que apenas o usuário proprietário possua permissões de leitura, escrita e execução, enquanto grupos e outros usuários não possuem qualquer tipo de acesso.

```
chmod 700 teste_permissoes
```

4.2 Gerenciamento de Serviços

Pacotes

Como preparação para a etapa de gerenciamento de pacotes, foram inicialmente executadas boas práticas de **administração de sistemas Linux**, por meio de comandos como `apt update` e `apt upgrade`, responsáveis pela atualização da lista de pacotes disponíveis e pela aplicação de atualizações do sistema utilizando o gerenciador de pacotes APT.

> **Comando utilizado**

```
sudo apt update #atualizou lista de pacote
sudo apt upgrade #atualizou pacotes instalados
```

Para o teste de instalação, foi escolhido o software **Nginx**, um serviço de servidor web amplamente utilizado em ambientes Linux. A instalação foi realizada por meio do comando `sudo apt install`, permitindo a validação do processo de gerenciamento e instalação de pacotes utilizando o gerenciador **apt.**

> **Comando utilizado**

```
sudo apt install nginx
```

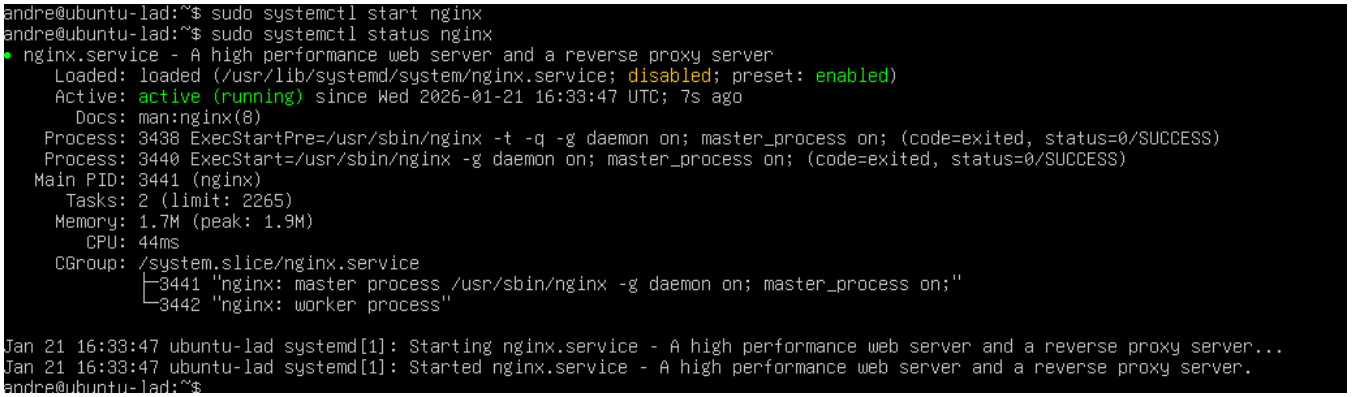
> *para a verificação de instalação do software foi executado o comando `nginx -v` ,comprovando a instalação e a versão adquirida.*

Serviços

Com o software já instalado, foi realizada a inicialização do serviço por meio do comando `sudo systemctl start nginx`. Em seguida, foi feita a verificação do status de execução utilizando o comando `sudo systemctl status nginx`, o qual forneceu informações detalhadas sobre o estado do serviço, seu funcionamento e possíveis mensagens de erro.

> **Comando utilizado**

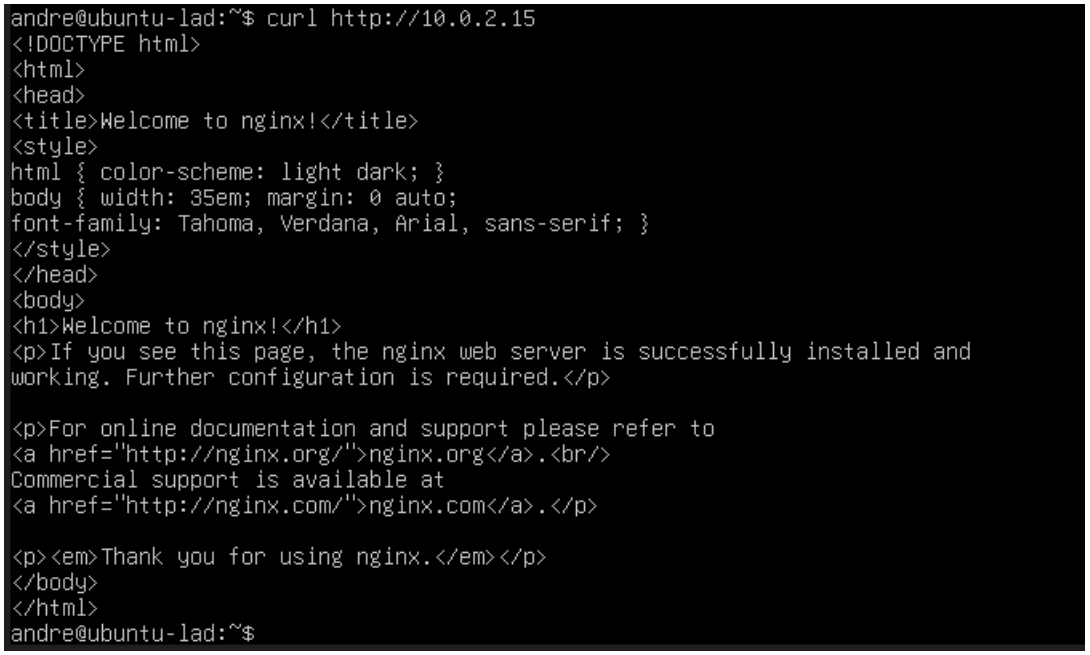
```
sudo systemctl start nginx #Iniciou o programa
sudo systemctl status nginx #Mostrou informaçõs sobre o estado do programa
```



Para o teste de funcionamento do serviço na máquina virtual, foi utilizado o método de verificação por meio do **endereço IP** da máquina que hospeda o **Nginx** dentro do ambiente virtualizado. O endereço IP foi obtido através do comando `ip a` e, em seguida, foi realizada a validação do serviço utilizando o comando `curl http://<IP_da_maquina>`, confirmando a disponibilidade do servidor web.

> **Comando utilizado**

```
ip a
curl http://
```



> *O acesso externo via host não foi configurado nesta etapa devido a rede da vm estar em* `NAT`.

Por fim, foi executado o comando `sudo systemctl stop nginx` e `sudo systemctl disable nginx` para encerrar as atividades do serviço e impedir a inicialização do serviço no boot, retornando a máquina virtual a um estado neutro, sem serviços em execução.

```
sudo systemctl stop nginx # Parar serviço
sudo systemctl disable nginx # Desabilitar serviço no boot
```

4.3 Segurança e Acesso

SSH

*Para a realização e documentação dos testes de **SSH e Firewall**, foi criado um novo usuário, denominado* `teste`*, utilizado como base para a execução e validação das conectividades entre os diferentes ambientes.*

Inicialmente conferi o estado de serviço e fiz a instalação do serviço responsável pela conexões `SSH`.

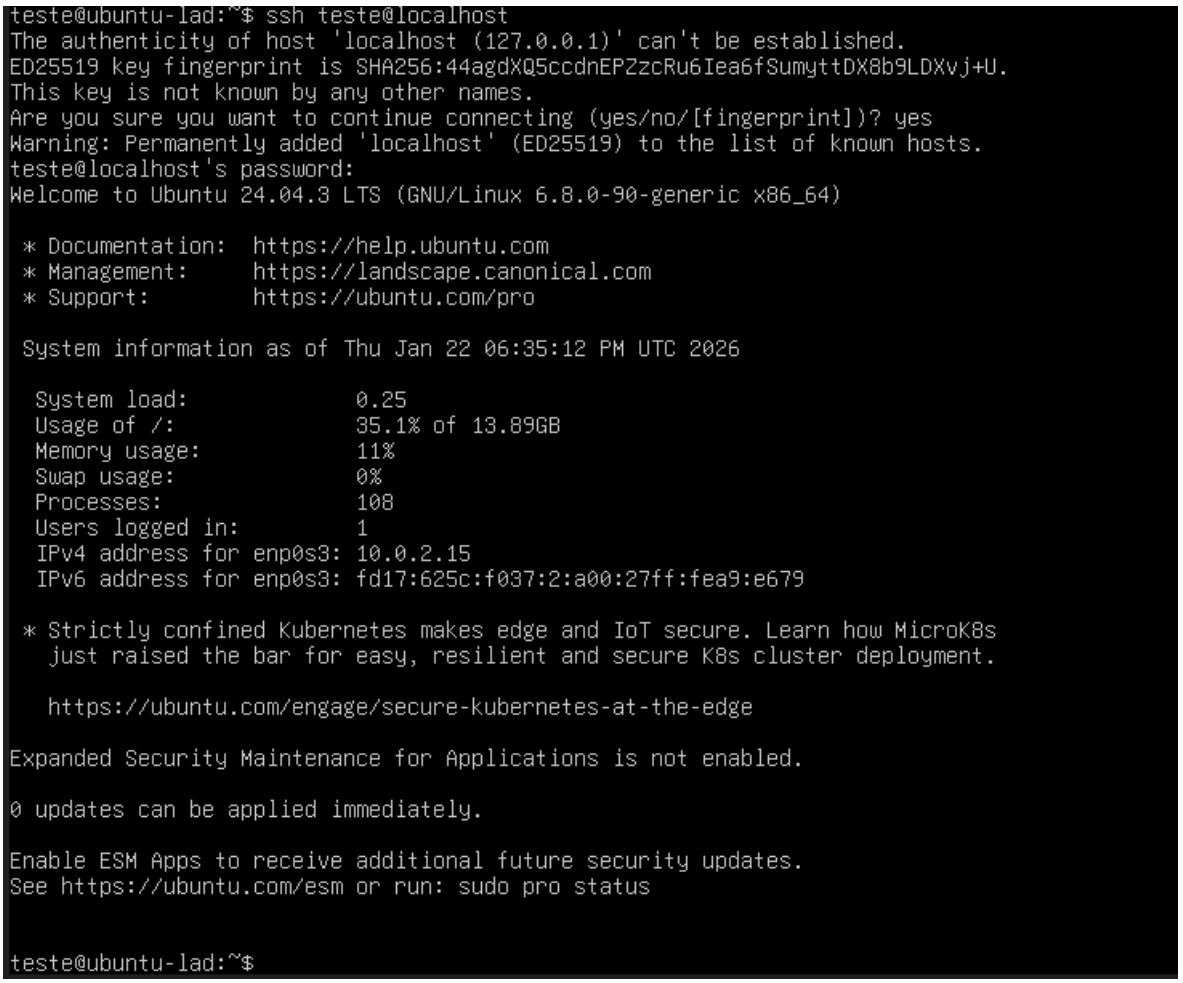
> **Comando utilizado**

```
sudo systemctl status ssh
sudo apt install openssh-server
```

Como primeiro teste de conectividade para verificar se o serviço **SSH** estava ativo, foi realizada uma conexão a partir da própria máquina dentro do ambiente virtual. Conforme esperado, a conexão foi realizada com sucesso, confirmando o funcionamento adequado do serviço.

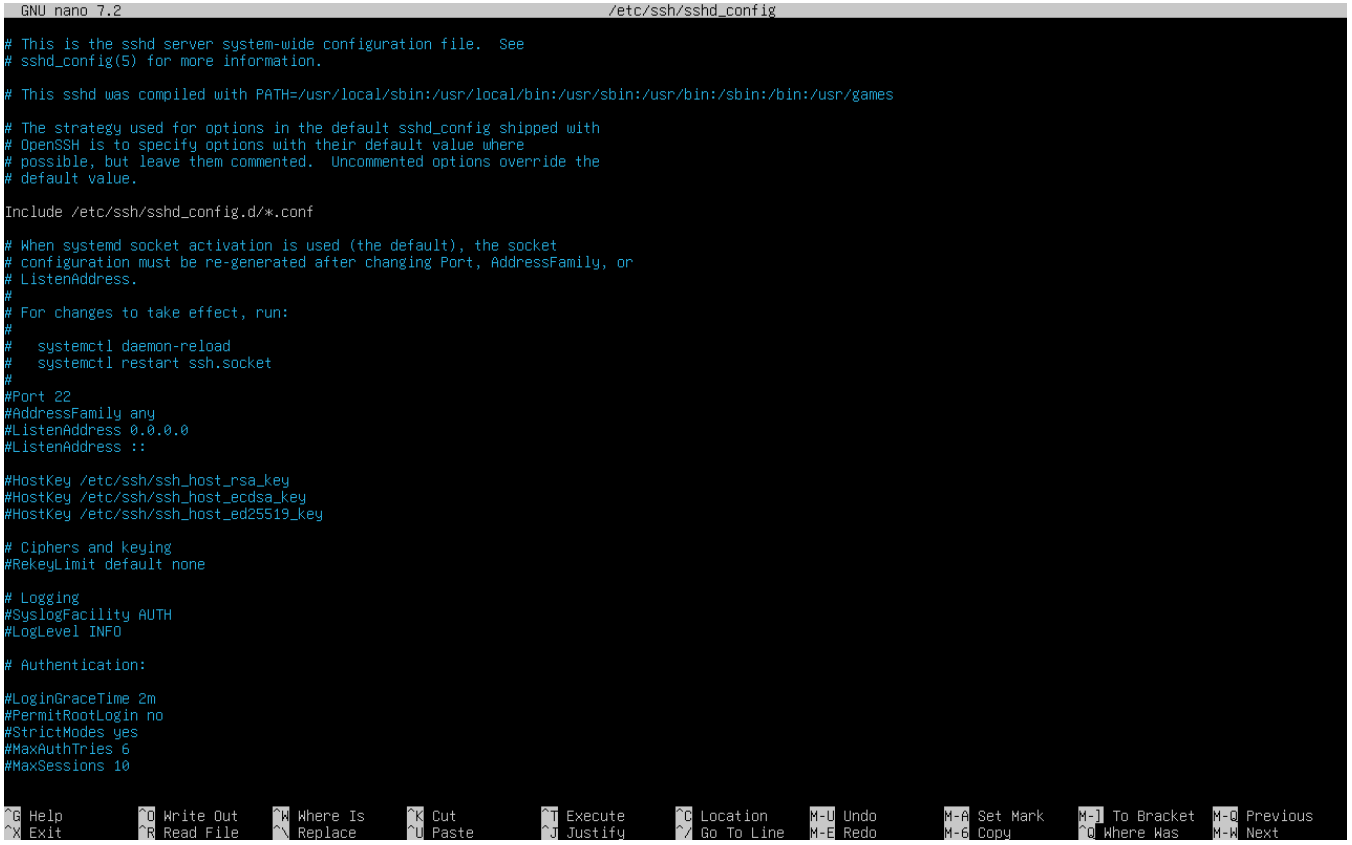
> **Comando utilizado**

```
ssh teste@localhost
```



O arquivo `/etc/ssh/sshd_config` foi utilizado para configurar políticas de acesso remoto. A opção `PermitRootLogin no` foi aplicada para impedir o login direto do usuário root via SSH.

```
sudo nano /etc/ssh/sshd_config
sudo systemctl restart ssh # Aplicar as modificações
```

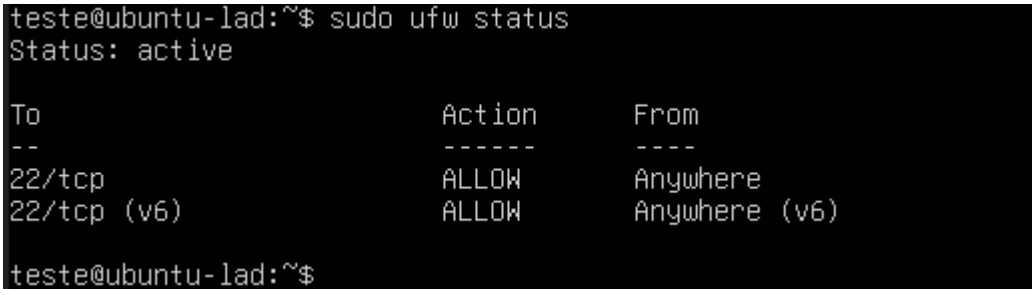


Firewall (UFW)

Com o serviço **SSH** configurado, a próxima etapa foi o gerenciamento do **firewall**. Inicialmente, foi utilizado o comando `sudo ufw status` para verificar o estado atual do firewall. Em seguida, foi executado o comando `sudo ufw allow ssh`, liberando a porta **22** e permitindo conexões SSH mesmo com o firewall ativo.

> **Comando utilizado**

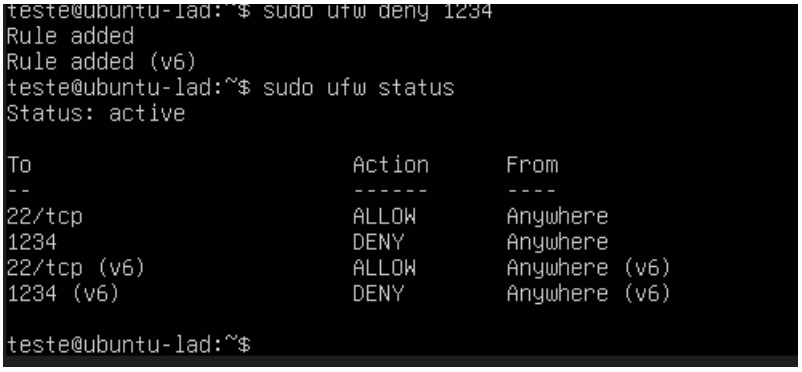
```
sudo ufw status
sudo ufw allow ssh
```



Por fim, o firewall foi ativado para bloquear todo o tráfego que não estivesse explicitamente permitido. Como teste simples de bloqueio, foi configurada a porta **1234** com o status `Deny`, impedindo qualquer tentativa de acesso a esse serviço.

> **Comando utilizado**

```
sudo ufw deny 1234 #Configura a porta 1234 para deny
```



Teste de SSH Host –> VM

Para finalização, realizei o teste de conectividade entre a máquina host com a máquina virtual. Para isso foi necessário configurar **Port Forwarding**, através das configurações do **Oracle VirtualBox**, devido a VM estar configurada em NAT ,impossibilitando o acesso direto do host sem configuração extra.

> No VirtualBox:
>
> > **Configurações da VM > Rede > Adaptador 1 > Redirecionamento de Portas > Acrescentar Nova Regra**

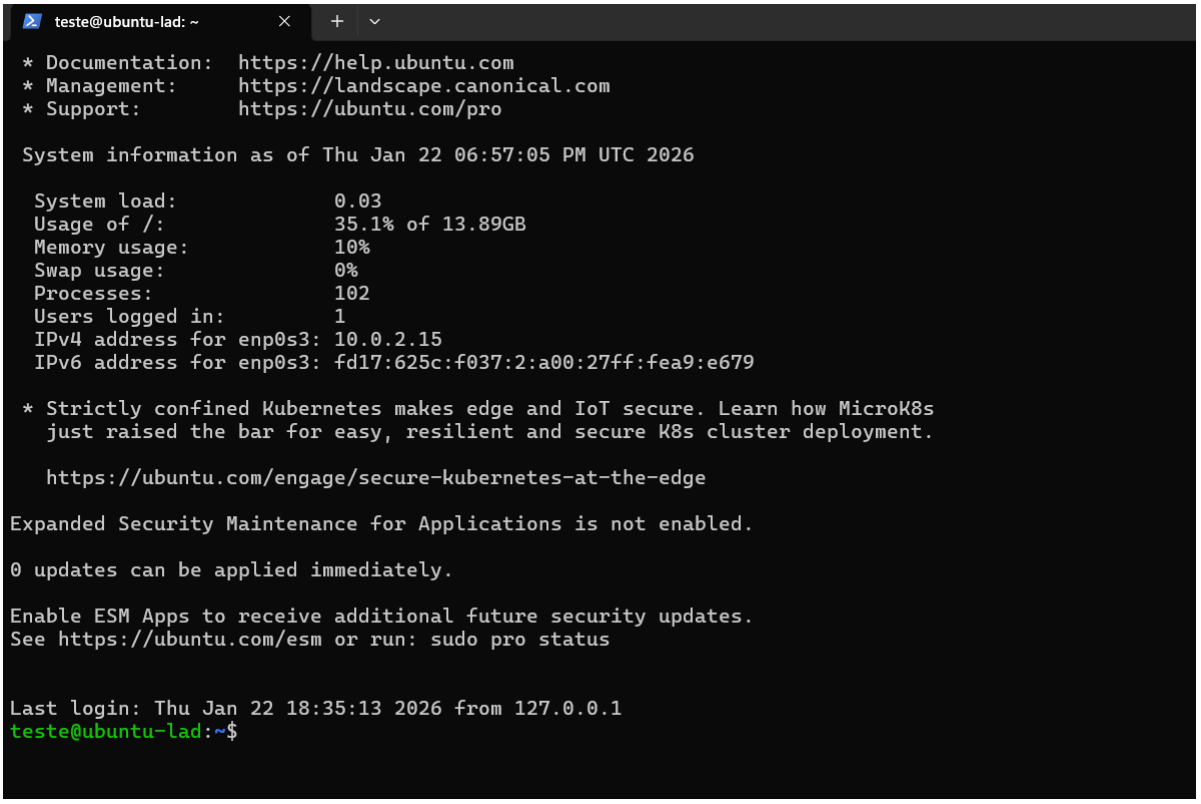
- Regras estabelecidas;

NOME	PROTOCOLO	IP HOST	PORTA HOST	IP CONVIDADO	PORTA CONVIDADO
SSH	TCP	—	2222	10.0.2.15	22

Após a realização das configurações, foi utilizado o **PowerShell** no sistema host para acessar a máquina virtual por meio de uma conexão **SSH**, validando a comunicação entre os ambientes host e VM.

> **Comando utilizado**

```
ssh teste@localhost -p 2222
```



> *Como configurado anteriormente foi solicitado a senha de login do usuário para o acesso ao ambiente.*

5.Conclusão

Após o estudo e a execução técnica das etapas deste projeto, foi possível aprimorar a autonomia na utilização do terminal **Linux**, além de desenvolver uma compreensão mais clara sobre o funcionamento interno do sistema e sua forma de operação.

Por meio do gerenciamento de **usuários** e **grupos**, foi possível compreender como um ambiente pode ser organizado de forma **pré-definida**, utilizando a separação de funções baseada em **permissões**, o que contribui para a criação de um ambiente controlado, seguro e eficiente.

A utilização e o gerenciamento de **pacotes** possibilitaram o entendimento de como um sistema pode ser expandido, modificado e mantido em bom estado operacional, assim como o uso de **serviços** para a execução de processos específicos. Por fim, a implementação de **mecanismos de segurança de acesso** permitiu transformar o ambiente em um laboratório seguro, com controle efetivo sobre quem pode acessar e operar o sistema.

---

6. Melhorias Futuras

1. **Atualizações Automáticas**
   Implementar `unattended-upgrades` para manter o sistema sempre atualizado com patches de segurança.
2. **Estrutura de Diretórios Corporativa**
   Criar hierarquia organizacional (ex: `/empresa/{financeiro,ti,rh}`) com permissões específicas por departamento.
3. **Gerenciamento de Pacotes Avançado**

Explorar mais recursos do `apt`: versionamento, limpeza automática, repositórios customizados.

4. **Monitoramento Básico**
   Adicionar verificação de logs, uso de recursos (CPU, memória, disco) com comandos como `top`, `df`, `journalctl`.
5. **Script de Provisionamento**
   Criar um script Bash que automatize toda a configuração deste servidor do zero.

---

7. Referências

Site oficial Oracle VirtualBox: VirtualBox site

Site Ubuntu Server:Ubuntu Server site

Gerenciamento de pacotes:Pacotes Linux

Fundamentos Linux: Princípios Básicos do Linux. Uma visão abrangente do Sistema... | by Hugo Habbema | Medium

SSH: Linux SSH

Firewall (UFW): Firewall UFW Linux