Servidor Linux para Banco de Dados

Instalação e Configuração Básica

Dominar a administração de servidores Linux é essencial para profissionais de dados

Terminal SSH
Acesso remoto seguro e criptografado

Segurança
Firewall, usuários e privilégios

Administração
Gerenciamento de sistema e rede

"Todo banco de dados robusto roda em um servidor estável, e a espinha dorsal dos servidores é o Linux."

Linux no Servidor

Estabilidade, **segurança** e **open source** fazem do Linux o padrão da indústria para servidores, especialmente em ambientes cloud.

Acesso Remoto SSH

O protocolo **SSH (Secure Shell)** permite conectar e controlar um servidor Linux à distância de forma **criptografada e segura**, usando apenas o terminal.

Ambiente Remoto

Você controlará um computador em outro lugar, sem instalar nada localmente. Pode ser uma conta cloud (AWS, Azure, GCP) ou um sandbox online.

Fundamentos: Linux no Servidor e Acesso Remoto SSH

Linux no Servidor

Linux é um sistema operacional **open source** que domina o mercado de servidores. Suas características principais são:

- → **Estabilidade**: Pode rodar 24/7 sem reiniciar
- → **Segurança**: Modelo de permissões robusto
- → **Customização**: Você controla tudo
- → **Custo**: Gratuito e sem licenças

Protocolo SSH

SSH (Secure Shell) é o protocolo padrão para acesso remoto seguro. Funciona assim:

- → Você conecta ao servidor usando **terminal/cliente SSH**
- → A conexão é **criptografada** (ninguém pode ver sua senha)
- → Você recebe um **prompt ($)** do servidor
- → Você digita **comandos** e o servidor executa

Conceito de Acesso Remoto ao Terminal

Você está controlando um computador em outro lugar. Não instala nada localmente. Tudo acontece via terminal:

- → **Seu computador**: Terminal SSH (cliente)
- → **Conexão**: Criptografada e segura
- → **Servidor remoto**: Linux (servidor SSH)
- → **Resultado**: Você controla o servidor

Atividade 1: Conexão e Navegação Básica no Terminal

Passo a Passo: Conexão SSH

1

Abra o Terminal ou Cliente SSH

Use o terminal do seu computador (Mac/Linux) ou PuTTY/Terminal (Windows)

2

Digite o Comando de Conexão

```
$ ssh usuario@ip_do_servidor
```

Exemplo: **ssh ubuntu@192.168.1.100**

3

Confirme a Autenticidade

Digite "yes" quando perguntado sobre a chave SSH

4

Digite sua Senha

Insira a senha do usuário (não aparecerá na tela por segurança)

5

Pronto! Você está Conectado

Você verá um **prompt ($)** - agora pode digitar comandos

Comandos Essenciais de Navegação

pwd (Print Working Directory)

Mostra o diretório atual onde você está

```
$ pwd
/home/ubuntu
```

ls (List)

Lista arquivos e pastas do diretório atual

```
$ ls -la
drwxr-xr-x 5 ubuntu ubuntu 4096 arquivo.txt
```

cd (Change Directory)

Navega para outro diretório

```
$ cd /home/ubuntu
$ cd .. (volta um nível)
```

clear

Limpa a tela do terminal

```
$ clear
```

O Comando sudo: Privilégios Administrativos

O que é sudo?

sudo significa **"Super User Do"**. É um comando que permite executar outros comandos com privilégios administrativos (de root).

```
$ sudo apt update [sudo] password for ubuntu:
```

Quando você digita **sudo** antes de um comando:

- → O sistema pede sua **senha**
- → Verifica se você tem **permissão** (grupo sudo)
- → Executa o comando como **root**
- → Registra a ação em **logs** de auditoria

sudo vs root

sudo	root
Uso: Um comando por vez	**Uso:** Múltiplos comandos
Auditoria: Registra cada ação	**Auditoria:** Sem registro
Segurança: Mais seguro	**Segurança:** Muito perigoso
Exemplo: sudo apt update	**Exemplo:** su - (vira root)

Nunca use root diretamente!

Um comando errado como root pode destruir todo o sistema. Use **sudo** para cada comando administrativo.

Atividade 2: Atualização do Sistema

Primeiro Ato Administrativo em um Servidor Linux

A **primeira ação** em um servidor novo é sempre atualizar o sistema. Isso garante **segurança** (patches de vulnerabilidades) e **estabilidade** (correções de bugs).

Passo a Passo Detalhado

Passo 1: Atualizar Lista de Pacotes

```
$ Digite este comando:
sudo apt update
```

Este comando **baixa a lista de pacotes** disponíveis dos repositórios. É como verificar o catálogo da loja.

Passo 2: Atualizar Pacotes Instalados

```
$ Digite este comando:
sudo apt upgrade -y
```

Este comando **atualiza todos os pacotes** para suas versões mais recentes. O **-y** responde "sim" automaticamente.

O que Observar no Output

Após apt update: Você verá linhas como "Hit", "Get", "Reading package lists". Isso significa que o sistema está baixando informações sobre pacotes disponíveis.

Após apt upgrade: Você verá a lista de pacotes a atualizar. O sistema pode pedir confirmação (responda "y" ou use -y para automático). Patches de segurança são críticos!

Atividade 3: Configuração de Rede - Parte 1

Por que IP Estático é Importante?

Servidores de banco de dados **precisam de um IP fixo (estático)**. Se o IP mudar, as aplicações não conseguem conectar. Em ambientes cloud, você pode simular isso, mas é importante entender a lógica.

Tarefa: Verificar o IP atual usando comandos essenciais.

Passo 1: Verificar o IP Atual

Comando 1: Mostra todos os endereços IP da máquina

```
$ ip addr
# Detalhado e completo
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536
  inet 127.0.0.1/8 scope host lo
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500
  inet 192.168.1.100/24 brd 192.168.1.255
```

Comando 2: Alternativa mais simples e legível

```
$ ifconfig
# Mais simples de ler
```

```
eth0: flags=67<UP,BROADCAST,RUNNING>
  inet 192.168.1.100
  netmask 255.255.255.0
  broadcast 192.168.1.255
```

Agora que você verificou o IP atual, na **Parte 2** vamos simular como seria uma configuração estática usando o editor nano e a sintaxe Netplan.

Atividade 3: Configuração de Rede - Parte 2

Passo 2: Simular Configuração Estática com Netplan

Em um servidor cloud, você não pode editar o arquivo de rede real. Mas vamos **simular a sintaxe** que seria usada em um servidor local. Abra o nano e crie um arquivo temporário:

```
$ nano /tmp/rede_config.txt

# Conteúdo do arquivo (sintaxe Netplan - Ubuntu/Debian):
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: no
      addresses:
        - address: 192.168.1.100/24
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Instruções Críticas: Usando o Nano Editor

- → **Passo 1:** Digite o comando acima e pressione Enter. O nano abrirá um editor de texto
- → **Passo 2:** Copie e cole (ou digite) a sintaxe Netplan acima no editor
- → **Passo 3:** Pressione **Ctrl+O** para salvar (Write Out)
- → **Passo 4:** Pressione **Enter** para confirmar o nome do arquivo
- → **Passo 5:** Pressione **Ctrl+X** para sair do nano
- → **Passo 6:** Verifique o arquivo com: **cat /tmp/rede_config.txt**

O que Significa Cada Parâmetro?

- **dhcp4: no** - Desativa DHCP (atribuição automática de IP). Você define o IP manualmente
- **address: 192.168.1.100/24** - IP fixo (192.168.1.100) com máscara /24 (255.255.255.0)
- **gateway4: 192.168.1.1** - Rota padrão (porta de saída para internet)
- **nameservers: [8.8.8.8, 8.8.4.4]** - Servidores DNS (Google DNS para resolver nomes)

Atividade 4: Criação de Usuário para Administração do BD - Parte 1

Nunca administre o banco de dados com a conta root! Crie um usuário específico (ex: bd_admin) com privilégios administrativos limitados. Isso aumenta a segurança e deixa um rastro de auditoria.

Passo a Passo: Criando o Usuário bd_admin (Parte 1)

1

Criar novo usuário

Use o comando sudo adduser para criar um novo usuário chamado bd_admin:

```
$ sudo adduser bd_admin
```

2

Definir senha

O sistema pedirá uma senha. Digite uma senha forte (mínimo 12 caracteres, com números e símbolos):

```
New password: ****
Retype new password: ****
```

3

Confirmar informações

O sistema pedirá nome completo, telefone, etc. Pressione Enter para pular ou preencha. No final, confirme com "y":

```
Is the information correct? [Y/n] y
```

Continua na próxima página: Você adicionará o usuário ao grupo sudo e testará os privilégios administrativos.

Atividade 4: Criação de Usuário para Administração do BD - Parte 2

Continuação: Passos 4 e 5

4

Adicionar ao grupo sudo

Agora dê privilégios administrativos ao bd_admin adicionando-o ao grupo sudo:

```
$ sudo usermod -aG sudo bd_admin
```

5

Testar o novo usuário

Mude para o novo usuário e teste se ele tem privilégios sudo:

```
$ su - bd_admin
$ sudo whoami
root
```

Checklist de Validação

- ✓ Usuário bd_admin foi criado com sucesso
- ✓ Você conseguiu fazer login com o novo usuário (su - bd_admin)
- ✓ O comando "sudo whoami" retornou "root" (privilégios confirmados)
- ✓ Você voltou ao usuário original (exit)

Atividade 5: Configuração de Firewall Mínimo (UFW) - Parte 1

O que é Firewall?

Pense no firewall como um **porteiro do servidor**. Ele controla quem entra e sai. Por padrão, bloqueia tudo. Você libera apenas o que é **essencial**: SSH (porta 22) para administração remota e a porta do seu banco de dados (ex: 5432 para PostgreSQL, 3306 para MySQL).

Comandos Essenciais do UFW

1. Verificar Status

```
$ sudo ufw status
```

Mostra se o firewall está ativo ou inativo

2. Ativar Firewall

```
$ sudo ufw enable
```

Ativa o firewall (cuidado: pode bloquear SSH!)

3. Permitir SSH

```
$ sudo ufw allow 22/tcp
```

Libera a porta 22 (SSH) - FAÇA ISSO ANTES DE ATIVAR!

4. Permitir PostgreSQL

```
$ sudo ufw allow 5432/tcp
```

Libera porta 5432 (PostgreSQL)

5. Permitir MySQL

```
$ sudo ufw allow 3306/tcp
```

Libera porta 3306 (MySQL)

6. Remover Regra

```
$ sudo ufw delete allow 22/tcp
```

Remove uma regra (use com cuidado!)

7. Listar Regras

```
$ sudo ufw show added
```

Mostra todas as regras adicionadas

8. Desativar Firewall

```
$ sudo ufw disable
```

Desativa o firewall (apenas em emergências)

Na **Parte 2**, vamos aplicar esses comandos em um passo a passo prático para configurar o firewall mínimo do seu servidor.

Atividade 5: Configuração de Firewall Mínimo (UFW) - Parte 2

Passo a Passo: Configurar Firewall Mínimo

1. **Verificar status:** sudo ufw status
2. **Permitir SSH:** sudo ufw allow 22/tcp
3. **Permitir BD:** sudo ufw allow 5432/tcp (ou 3306 para MySQL)
4. **Ativar firewall:** sudo ufw enable
5. **Verificar regras:** sudo ufw status numbered
6. **Testar conexão:** Tente conectar via SSH novamente

Checklist de Validação

- ✓ Firewall UFW está ativo (sudo ufw status mostra "Status: active")
- ✓ SSH (porta 22) está permitido
- ✓ Porta do BD (5432 ou 3306) está permitido
- ✓ Você conseguiu conectar via SSH após ativar o firewall

Consolidação: Habilidades Essenciais Conquistadas

Acesso Remoto SSH

Conectar a um servidor remoto de forma segura e criptografada

Navegação Terminal

Usar comandos essenciais (pwd, ls, cd, clear) para explorar o sistema

Atualização Sistema

Manter o servidor seguro com apt update e apt upgrade

Configuração Rede

Entender e verificar configurações de IP (estático e dinâmico)

Gerenciamento Usuários

Criar usuários específicos com privilégios administrativos limitados

Firewall Básico

Configurar UFW para permitir apenas tráfego essencial (SSH e BD)

Você Conseguiu Fazer Tudo Isso?

- ✓ Conectou ao servidor remoto via SSH com sucesso
- ✓ Atualizou o sistema com apt update e apt upgrade
- ✓ Criou um novo usuário (bd_admin) com privilégios sudo
- ✓ Navegou pelos diretórios usando pwd, ls e cd
- ✓ Verificou o IP atual com ip addr ou ifconfig
- ✓ Configurou firewall básico com UFW (SSH e porta BD)

Conclusão: Próximos Passos e Reflexão Final

O que você conquistou nesta aula

Você aprendeu os fundamentos essenciais para administrar um servidor Linux remoto. Desde conectar via SSH até configurar firewall, você agora domina as habilidades que profissionais de dados usam diariamente em ambientes de produção.

Próximos Passos Práticos

Instalar Banco de Dados

Instale PostgreSQL, MySQL ou MongoDB no servidor. Use os comandos apt que aprendeu.

Monitoramento

Aprenda a monitorar CPU, memória e disco usando top, df e ferramentas como Prometheus.

Segurança Avançada

Configure SSH com chaves públicas, implemente fail2ban e audite logs de segurança.

Continua na próxima página: Entenda por que Linux é essencial para profissionais de dados e a mensagem final de conclusão.

Por que Linux é Essencial para Profissionais de Dados

Razões Fundamentais

- → **Mercado de Trabalho:** 96% dos servidores em nuvem rodam Linux
- → **Automação:** Scripts bash e cron jobs para processos automáticos
- → **Big Data:** Hadoop, Spark e ferramentas de dados rodam em Linux
- → **DevOps:** Docker, Kubernetes e CI/CD rodam em Linux

🚀

Você agora está pronto para gerenciar servidores Linux e trabalhar em ambientes profissionais de dados. Continue praticando, explorando e aprendendo. O futuro da tecnologia é Linux!

⭐