

# Comandos Linux

---

## 1. Comando **pwd**

Use o comando **pwd** para encontrar o caminho para o diretório atual (da pasta) em que você está. O comando vai retornar um caminho completo (cheio), que é basicamente um caminho que começa com uma barra inclinada (**/**). Um exemplo de um caminho completo é **/home/username**.

O comando **pwd** usa a seguinte sintaxe:

**pwd [opção]**

Ele tem duas opções aceitáveis:

- **-L** ou **-logical** imprime o conteúdo da variável de ambiente, incluindo links simbólicos.
- **-P** ou **-physical** imprime o caminho real do diretório atual.

## 2. Comando **cd**

Para navegar pelos arquivos e diretórios Linux, use o comando **cd**. Ele requer ou um caminho completo ou o nome de um diretório, dependendo do diretório atual em que você estiver.

Vamos dizer que você esteja em **/home/username/Documents** e quer ir para **Photos**, um subdiretório de **Documents**. Para fazer isso, simplesmente digite **cd Photos**.

Outro cenário em que você quer mudar completamente de diretório, digamos, para **/home/username/Movies**. Nesse caso, você tem que digitar **cd** seguido pelo caminho absoluto do diretório.

Existem alguns atalhos que você pode usar para navegar mais rapidamente.:

- Use **cd ..** (com dois pontos seguidos) para subir diretório acima
- Use **cd ~[username]** para acessar o diretório inicial de outro usuário.
- Use **cd\*-\*** (com um hífen) para mover para os diretórios anteriores.

## 3. Comando **ls**

O comando **ls** é usado para visualizar conteúdos em um diretório. Por padrão, esse comando vai mostrar os conteúdos apenas do diretório atual em que você estiver.

Se você quiser ver o conteúdo de outros diretórios, digite **ls** e, então, o caminho do diretório. Por exemplo, digite **ls /home/username/Documents** para ver os conteúdos de **Documents**.

Existem variações que você pode usar com o comando **ls**:

- **ls -R** vai listar todos os arquivos nos subdiretórios.
- **ls -a** vai mostrar todos os arquivos ocultos.
- **ls -lh** vai listar todos os tamanhos de arquivos em formatos fáceis, como MB, GB ou TB.

## 4. Comando cat

O **cat** (abreviação para concatenar) é um dos comandos Linux mais usados. Ele é usado para visualizar, criar e relacionar arquivos. Para executar esse comando, digite **cat** seguido pelo nome do arquivo e sua extensão. Por exemplo: **cat nomedoarquivo.txt**.

Aqui estão outras maneiras de usar o [comando cat](#):

- **cat > nomedoarquivo.txt** cria um novo arquivo
- **cat nomedoarquivo1.txt \*nomedoarquivo\*2.txt > nomedoarquivo3.txt** junta dois arquivos (1 e 2) em um novo (3).
- **tac nomedoarquivo.txt** exibe o conteúdo do arquivo em ordem reversa.

## 5. Comando cp

Use o comando **cp** para copiar arquivos ou diretórios e seu conteúdo. Abaixo, listamos alguns exemplos.

Para copiar um arquivo do diretório atual para outro, digite **cp** seguido do nome do arquivo e do diretório de destino. Por exemplo:

**cp nomedoarquivo.txt /home/username/Documents**

Para copiar arquivos para um diretório, digite os nomes dos arquivos seguidos do diretório de destino:

**cp nomedoarquivo1.txt \*nomedoarquivo\*2.txt \*nomedoarquivo\*3.txt  
/home/username/Documents**

Para copiar o conteúdo de um arquivo para um novo arquivo no mesmo diretório, digite **cp** seguido do arquivo de origem e do arquivo de destino:

**cp \*nomedoarquivo\*1.txt \*nomedoarquivo\*2.txt**

Para copiar um diretório inteiro, passe o sinalizador **-R** antes de digitar o diretório de origem, seguido pelo diretório de destino:

**cp -R /home/username/Documents /home/username/Documents\_backup**

## 6. Comando mv

O uso mais comum do comando **mv** é mover arquivos, mas ele também pode ser usado para renomear arquivos.

Basta digitar **mv** seguido do nome do arquivo e do diretório de destino. Por exemplo, você deseja mover o arquivo **\*nomedoarquivo\*.txt** para o diretório **/home/username/Documents**:

**mv \*\*\*nomedoarquivo\*\*\*.txt /home/username/Documents.**

Você também pode usar o [comando mv](#) para renomear um arquivo:

**\*mv\* nomedoarquivo\_antigo.txt nome\_novo.txt**

## 7. Comando mkdir

Use o comando **mkdir** para criar um ou vários diretórios de uma só vez e definir permissões para cada um deles. O usuário que executa esse comando deve ter o privilégio de criar uma nova pasta no diretório principal, caso contrário, poderá receber um erro de permissão negada.

Aqui está a sintaxe básica:

**mkdir [opção] nome\_do\_diretório**

Por exemplo, você deseja criar um diretório chamado **Music**:

**mkdir Music**

Para criar um novo diretório chamado **Songs** dentro de **Music**, use este comando:

**mkdir Music/Songs**

O comando **mkdir** aceita muitas opções, como:

- **-p** ou **-parents** cria um diretório entre duas pastas existentes. Por exemplo, **mkdir -p Music/2020/Songs** criará o novo diretório “2020”.
- **-m** define as permissões do arquivo. Por exemplo, para criar um diretório com permissões completas de leitura, gravação e execução para todos os usuários, digite **mkdir -m777 nome\_do\_diretório**.
- **-v** imprime uma mensagem para cada diretório criado.

## 8. Comando rmdir

Para excluir permanentemente um diretório vazio, use o [comando rmdir](#). Lembre-se de que o usuário que executa esse comando deve ter privilégios **sudo** no diretório pai.

Por exemplo, você deseja remover um subdiretório vazio chamado **personal1** e sua pasta principal **mydir**:

**rmdir -p mydir/personal1**

## 9. Comando rm

O comando **rm** é usado para excluir arquivos em um diretório. Certifique-se de que o usuário que executa esse comando tenha permissões de gravação.

Lembre-se do local do diretório, pois isso apagará o(s) arquivo(s) permanentemente e não há como desfazer a ação.

Aqui está a sintaxe geral:

**rm nome\_do\_arquivo**

Para remover vários arquivos, digite o seguinte comando:

**rm \*nome\_do\_arquivo\*1 \*nome\_do\_arquivo\*2 \*nome\_do\_arquivo\*3**

Aqui estão algumas opções que você pode adicionar:

- **-i** solicita a confirmação do sistema antes de excluir um arquivo.
- **-f** permite que o sistema faça a remoção sem confirmação.
- **-r** exclui arquivos e diretórios recursivamente.

## 10. Comando touch

O [comando touch](#) permite criar um arquivo vazio ou gerar e modificar um registro de data e hora na linha de comando do Linux.

Por exemplo, digite o seguinte comando para criar um arquivo HTML chamado **Web** no diretório **Documents**:

```
touch /home/username/Documents/Web.html
```

## 11. Comando locate

Você pode o [comando locate](#) para localizar um arquivo, assim como você faz para procurar um arquivo no Windows. Além disso, usando o argumento **-i** junto com esse comando faz com que ele se torne insensível a maiúsculas ou minúsculas, permitindo que você pesquise por um arquivo mesmo sem saber exatamente o nome dele.

Para procurar um arquivo que contém duas ou mais palavras, use um asterisco (\*). Por exemplo, use o comando **locate -i school\*note** para encontrar qualquer arquivo que tenha as palavras "school" e "note", não importando se existem letras maiúsculas ou minúsculas.

## 12. Comando find

Similar ao comando **locate**, o comando **find** ajuda você a procurar por arquivos. A diferença é que você usa o **find** para localizar arquivos dentro de um diretório específico.

Como exemplo, digite **find /home/ -name notes.txt** para procurar por um arquivo chamado **notes.txt** dentro do diretório home e seus subdiretórios.

Outras variações na hora de usar o **find** são:

- **find -name nomedoarquivo.txt** para localizar arquivos no diretório atual.
- **find ./ -type d -name nomedodiretorio** para procurar diretórios.

## 13. Comando grep

Outro comando básico do Linux que merece ser citado é o **grep**. Ele permite que você encontre uma palavra pesquisando todo o conteúdo de um arquivo específico.

Quando o [comando grep](#) encontra uma correspondência, ele imprime todas as linhas que contêm o padrão específico. Esse comando ajuda a filtrar arquivos de registro grandes.

Por exemplo, se você deseja pesquisar a palavra **blue (azul)** no arquivo **notepad.txt**:

```
grep blue notepad.txt
```

A saída do comando exibirá as linhas que contêm a palavra **blue**.

## 14. Comando sudo

Correspondente a **SuperUser Do**, **sudo** é um dos comandos básicos mais populares do Linux. Ele permite executar tarefas que exigem permissões administrativas ou de root.

Ao usar o **sudo**, o sistema solicitará que os usuários se autentiquem com uma senha. Em seguida, o sistema Linux registrará um registro de data e hora como um rastreador. Por padrão, todo usuário root pode executar comandos **sudo** por **15 minutos/sessão**.

Se você tentar executar o sudo na linha de comando sem se autenticar, o sistema registrará a atividade como um evento de segurança.

Aqui está a sintaxe geral:

### **sudo [comando]**

Você também pode adicionar uma opção, por exemplo:

- **-k** ou **-reset-timestamp** invalida o arquivo de registro de data e hora.
- **-g** ou **-group=group** executa comandos como um nome ou ID de grupo especificado.
- **-h** ou **-host=host** executa comandos no host.

## **15. Comando df**

Use o [comando df](#) para obter informações sobre o uso do espaço em disco do sistema, mostrado em porcentagem e quilobyte (KB). Esta é a sintaxe geral:

### **df [opções] [arquivo]**

Por exemplo, digite o seguinte comando se quiser ver quanto espaço o atual diretório ocupa em um formato legível por humanos:

**df -h**

Essas são algumas opções que você pode usar:

- **O df -m** exibe informações sobre o uso do sistema de arquivos em **MBs**.
- **df -k** exibe o uso do sistema de arquivos em **KBs**.
- **df -T** mostra o **tipo** de sistema de arquivos em uma nova coluna.

## **16. Comando du**

Se você quiser verificar quanto de espaço um arquivo ou diretório ocupa, use o comando **du**. Você pode executar esse comando para identificar qual parte do sistema usa excessivamente o armazenamento do seu sistema.

Lembre-se de que você deve especificar o caminho do diretório ao usar o comando **du**. Por exemplo, para verificar **/home/user/Documents**, digite:

**du /home/user/Documents**

Adicionar um sinalizador ao comando **du** modificará a operação, por exemplo:

- **-s** oferece o tamanho total de uma pasta especificada.
- **-m** fornece informações sobre pastas e arquivos em **MB**
- **k** exibe informações em **KB**.
- **-h** informa a data da última modificação das pastas e arquivos exibidos.

## **17. Comando head**

O comando **head** permite que você visualize as primeiras dez linhas de um texto. A adição de uma opção permite que você altere o número de linhas mostradas. O comando **head** também é usado para enviar dados canalizados para a CLI.

Aqui está a sintaxe geral:

## **head [opção] [arquivo]**

Por exemplo, se você deseja visualizar as primeiras dez linhas do arquivo **note.txt**, localizado no diretório atual:

**head note.txt**

Abaixo estão algumas opções que você pode adicionar:

- **-n** ou **-lines** imprime o primeiro número personalizado de linhas. Por exemplo, digite **head -n 5 nomedoarquivo.txt** para exibir as cinco primeiras linhas de **nomedoarquivo.txt**.
- **-c** ou **-bytes** imprime o primeiro número personalizado de bytes de cada arquivo.
- **-q** ou **-quiet** não imprimirá cabeçalhos que especifiquem o nome do arquivo.

## **18. Comando tail**

O [comando tail](#) tem função similar ao comando **head**, mas mostrando as últimas 10 linhas de um arquivo de texto.

Este é o formato geral:

## **tail [opção] [arquivo]**

Por exemplo, você deseja mostrar as últimas dez linhas do arquivo **colors.txt**:

**tail -n colors.txt**

## **19. Comando diff**

O comando **diff** (diferença) compara o conteúdo de dois arquivos linha por linha. Depois de analisar esses arquivos, ele vai mostrar as linhas que não são comuns entre eles.

Os programadores frequentemente usam este comando quando precisam fazer pequenas alterações em programas. Assim, eles não precisam reescrever o código inteiro.

Este é o formato geral:

## **diff [opção] arquivo1 arquivo2**

Por exemplo, você deseja comparar dois arquivos de texto — **note.txt** e **note\_update.txt**:

**diff note.txt note\_update.txt**

Aqui estão algumas opções para adicionar:

- **-q** mostra apenas se os arquivos são diferentes ou não, sem especificar as diferenças.
- **-i** torna o comando **diff** insensível a maiúsculas e minúsculas.
- **-b** ignora espaços em branco como possíveis diferenças.

## **20. Comando tar**

O [comando tar](#) reúne vários arquivos em um arquivo **TAR** — um formato do Linux semelhante ao **ZIP**, com compactação opcional.

Aqui está a sintaxe básica:

**tar [options] [archive\_file] [arquivo ou diretório a ser arquivado]**

Por exemplo, você deseja criar um novo arquivo **TAR** chamado **novoarquivo.tar** no diretório **/home/user/Documents**:

```
tar -cvf novoarquivo.tar /home/user/Documents
```

O comando **tar** aceita muitas opções, como:

- **-x** extrai um arquivo.
- **-t** lista o conteúdo de um arquivo.
- **-u** arquiva e adiciona a um arquivo existente.

## 21. Comando chmod

O **chmod** é um comando que modifica as permissões de leitura, gravação e execução de um arquivo ou diretório. No Linux, cada arquivo está associado a três classes de usuários:

**proprietário, membro do grupo e outros**.

Aqui está a sintaxe básica:

```
chmod [opção] [permissão] [nome_do_arquivo]
```

Por exemplo, o proprietário é atualmente o único com permissões totais para alterar o arquivo **note.txt**. Para permitir que os membros do grupo e outros possam ler, gravar e executar o arquivo, altere-o para o tipo de permissão **-rwxrwxrwx**, cujo valor numérico é **777**:

```
chmod 777 note.txt
```

O [comando chmod](#) oferece suporte a várias opções, incluindo:

- **-c** ou **-changes** exibe informações quando uma alteração é feita.
- **-f** ou **-silent** suprime as mensagens de erro.
- **-v** ou **-verbose** exibe um diagnóstico para cada arquivo processado.

## 22. Comando chown

No Linux, todos os arquivos são de propriedade de um usuário específico. O [comando chown](#) permite alterar a propriedade de um arquivo, diretório ou link simbólico para um nome de usuário específico.

Este é o formato básico:

```
chown [opção] proprietário[:grupo] arquivo(s)
```

Por exemplo, você deseja tornar **o linuxuser2** o proprietário do arquivo **filename.txt**:

```
chown linuxuser2 filename.txt
```

## 23. Comando jobs

Um job é um processo que iniciado pelo shell. O comando **jobs** exibirá todos os processos em execução juntamente com seus status. Lembre-se de que esse comando só está disponível nos shells **csh**, **bash**, **tcsh** e **ksh**.

Essa é a sintaxe básica:

```
jobs [opções] jobID
```

Para verificar o status dos trabalhos no shell atual, basta digitar **jobs** na CLI.

Aqui estão algumas opções que você pode usar:

- **-l** lista as IDs de processo juntamente com suas informações.
- **-n** lista os processos cujos status foram alterados desde a última notificação.
- **-p** lista somente IDs de processos.

## Grupo e dono: todo arquivo tem um

Quando um arquivo é criado ele possui um dono (quem o criou) e um grupo. Todo usuário pertence a um ou mais grupos.

Digitando `whoami` (quemsoueu) você descobrirá o nome do usuário atual, e digitando `groups` descobrirá a quais grupos este usuário pertence.

```
$ whoami  
magenteiro  
  
$ groups  
magenteiro www-data
```

O comando `ls -la` listará todos os arquivos do diretório atual, trazendo informações sobre o nome do dono, o grupo que cada arquivo e diretório pertencem e quais permissões cada um deles tem sobre cada item.

```
$ ls -la  
  
total 20320  
drwxr-xr-x 15 magenteiro www-data      4096 Feb 11 00:00 .  
drwxr-xr-x  3 root      root          4096 Jul  4 2019 ..  
-rw-r--r--  1 magenteiro www-data      11249 Nov 12 04:47 .htaccess  
-rw-r--r--  1 magenteiro www-data      11382 Nov 12 04:47 .htaccess.sample  
-rw-r--r--  1 magenteiro www-data     766543 Nov 12 04:47 CHANGELOG.md  
-rw-r--r--  1 magenteiro www-data       650 Nov 12 04:47 COPYING.txt  
drwxr-xr-x  3 magenteiro www-data      4096 Nov 12 06:00 app  
drwxrwsr-x 84 magenteiro www-data      4096 Nov 12 04:47 vendor
```

- Primeira coluna (drwxr-xr-x)  
Traz informações sobre as permissões que o dono, o grupo e outros usuários tem sobre o item. A primeira letra indica se o item é um diretório (d) ou um arquivo (-).
- Terceira coluna (magenta, root, etc)  
Nome do usuário que é o dono do arquivo ou diretório
- Quarta coluna (www-data, root, etc)  
Nome do grupo ao qual o arquivo ou diretório pertence

## Entendendo as permissões rwx

As permissões de um arquivo são divididas na seguinte ordem:



Cada letra representa uma permissão que cada um tem sobre o arquivo ou pasta.

As letras geralmente são:

- r – read (leitura)
- w – write (escrita)
- x – execute (execução)

E há ainda a letra **s** que é menos utilizada. Ela representa permissão de ajuste de user id (*setuid*). Esta é uma permissão especial que permite que qualquer usuário execute o arquivo como se fosse o dono. Em outras palavras, supondo que o dono do arquivo seja o root, qualquer usuário poderia executá-lo com o comando `sudo`.

## Todos os cursos Magento por R\$39/mês\*

Faça parte do Clube Magenteiro e tenha acesso a todos os cursos e conteúdos. \*pa

No exemplo da imagem acima, o dono do arquivo possui permissão para ler (r) e escrever/modificar(w) o arquivo. Usuários que estiverem no mesmo grupo que este arquivo pertence poderão apenas ler(r) o arquivo. Enquanto outros usuários, mesmo que não sejam o dono, nem façam parte deste grupo, poderão apenas ler (r) o arquivo.

## Entendendo os erros de permissão

Quando nosso servidor web (apache ou nginx por exemplo) serve uma aplicação (PHP por exemplo), ele faz isso usando um usuário que está inserido em um ou mais grupos. Sendo assim, se este usuário não tiver permissão de leitura nos arquivos da nossa aplicação, certamente teremos erros de permissão.

Permissão de escrita também é necessária quando nossa aplicação tenta escrever arquivos de sessão, gerar arquivos de cache, ou mesmo quando fazemos upload de um arquivo e o mesmo tenta ser salvo em determinada pasta que nosso servidor web não possui a permissão necessária.

## Alterando as permissões (um jeito simples de entender)

O comando `chmod` é o responsável por alterar as permissões de um arquivo ou pasta. Geralmente este comando é seguido de uma sequência de 3 números (ex: `chmod 754 arquivo.txt`).

No entanto, existe uma forma um pouco mais fácil de utilizá-lo e que facilita bastante a nossa vida.

`chmod [tipo][operador][permissão] arquivo`

Onde tipo pode ser:

- u: dono do arquivo (*user who owns it*)
- g: grupo do arquivo (*group*)
- o: outros (*others*)
- a: todos os acima (*all*)

Os operadores podem ser:

- +: o sinal de mais adicionará as permissões a seguir
- -: o sinal de menos removerá as permissões a seguir

E a permissão, pode ser a combinação de uma ou mais das letras que já vimos acima.

- r: para leitura (*read*)
- w: para escrita (*write*)
- x: para execução (*execute*)

Exemplos:

- `chmod u+rwx arquivo.txt`  
Adiciona permissão de leitura, escrita e execução para o dono do arquivo
- `chmod g-w arquivo.txt`  
Remove permissão de escrita para pessoas inseridas no grupo ao qual o arquivo pertence
- `chmod a+r arquivo.txt`  
Dá permissão de leitura para todo mundo
- `chmod u+x,g-w,o+r arquivo.txt`  
Acrescenta permissão de execução para o dono, remove permissão de escrita para o grupo, e adiciona permissão de leitura para outros.
- `chmod -R u+rw diretório`  
Dá permissão de leitura e escrita para o dono em todos os **arquivos e diretórios** dentro de uma pasta **recursivamente**.
- `chmod o-rwx *.sql`  
Remove todas as permissões de que não é o dono nem faz parte do grupo, em todos os arquivos do diretório atual com extensão .sql.

## Alterando o dono e o grupo

---

Por fim, temos o comando chown (*change owner*) que nos permite alterar o usuário e/ou grupo de um arquivo ou pasta.

- `chown magenteiro:www-data arquivo.txt`  
O dono passa a ser “magenteiro” e o grupo “www-data”
- `chown :www-data arquivo.txt`  
O grupo passa a ser “www-data” e o dono permanece o mesmo
- `chown magenteiro arquivo.txt`  
O dono passa a ser “magenteiro” e o grupo permanece o mesmo
- `chown -R :www-data diretório`  
Altera o grupo de um diretório e todos seus arquivos e subpastas recursivamente.

# Conclusão

---

Você pode “resolver” seus problemas de permissão com “chmod -R a+rwx” ou “chmod -R 777” seguidos do caminho da sua aplicação. No entanto, ao fazer isso você comprometerá a segurança da sua aplicação e de arquivos de configuração, facilitando a exposição de arquivos sensíveis que podem conter credenciais de banco de dados e outras informações importantes.

Um outro erro comum que vejo no mundo Magento é fazer tudo com usuário *root*. O Magento e nenhuma outra aplicação deveria ser instalada em um servidor com este usuário.

Idealmente devemos instalar o Magento com um usuário comum e depois ajustar as permissões como [mostrado na documentação oficial](#), e como explicado no [Magento 2: O Curso](#) (na seção onde configuramos um servidor Magento do zero na nuvem).

Uma regra que costumo seguir para não ter problemas de permissão no Magento é:

1. Configurar o grupo com o mesmo grupo usado pelo nosso servidor web (ex: `chown -R :www-data <pastamagento>`)
2. Nunca executar o `bin/magento` e nem alterar nada usando o usuário *root*.
3. Sempre usar um usuário que esteja inserido neste grupo para executar todas as tarefas administrativas, fazer deploys, alterar arquivos, etc. Isso pode ser feito com o comando `sudo -iu <usuario>`. Ex: `sudo -iu magenteiro`

## 24. Comando kill

Se você tem um programa que não está respondendo bem, você pode finalizá-lo manualmente pelo [comando kill](#). Ele vai mandar um certo sinal ao aplicativo com mau funcionamento e instruir que ele seja encerrado sozinho logo na sequência.

Existe um total de [64 avisos](#) que você pode usar, mas, geralmente, as pessoas usam apenas 2 deles:

- **SIGTERM (15)** – pede que um programa pare de rodar e dá algum tempo para salvar todo o seu progresso. Se você não especificar o aviso quando executar o comando *kill*, é este aviso que será usado.
- **SIGKILL (9)** – força um programa a parar imediatamente, em que todo o progresso não salvo será perdido.

Além de saber os avisos (sinais, notificações), você também precisa conhecer o número de identificação do processo (PID) do programa que você quer **matar** (*kill*). Se você não souber o PID, apenas execute o comando **ps ux**.

Depois de saber qual aviso você quer usar e o PID do programa, use a sintaxe abaixo:

**kill [signal option] PID.**

## 25. Comando ping

O [comando ping](#) é um dos comandos básicos do Linux mais usados para verificar se uma rede ou um servidor está acessível. Além disso, ele é usado para solucionar vários problemas de conectividade.

Este é o formato geral:

**ping [opção] [nome\_do\_hospedeiro\_ou\_endereço\_IP]**

Por exemplo, você quer saber se pode se conectar ao **Google** e medir seu tempo de resposta:

**ping google.com**

## 26. Comando wget

A linha de comando do Linux permite que você baixe arquivos da Internet usando o [comando wget](#). Ele funciona em segundo plano, sem atrapalhar outros processos em execução.

O comando **wget** baixa arquivos usando os protocolos HTTP, HTTPS e FTP. Ele pode executar downloads recursivos, que transferem partes de sites seguindo estruturas de diretórios e links, criando versões locais de páginas da web.

Para usá-lo, digite o seguinte comando:

**wget [opção] [url]**

Por exemplo, digite o seguinte comando para baixar a versão mais recente do [WordPress](#):

**wget <https://wordpress.org/latest.zip>**

## 27. Comando uname

O comando **uname**, que significa Unix Name, vai mostrar informações detalhadas sobre seu sistema Linux. Isso inclui o nome da máquina, do sistema operacional, do kernel e assim por diante.

**uname [opção]**

Essas são algumas das opções que você pode usar:

- **-a** exibe todas as informações do sistema.
- **-s** exibe o nome do kernel.
- **-n** exibe o hostname do node do sistema.

## 28. Comando top

Equivalente ao gerenciador de tarefas do Windows, o [comando top](#) vai mostrar uma lista de processos que estão em execução e o quanto de CPU cada processo usa. Ele é muito útil para monitorar o uso dos recursos do sistema, especialmente para saber qual processo deve ser encerrado por consumir muitos recursos. Basta digitar **top** na CLI para executá-lo.

## 29. Comando history

Com o **history**, o sistema listará até 500 comandos executados anteriormente, permitindo que você os reutilize sem precisar digitá-los novamente. Lembre-se de que somente os usuários com privilégios **sudo** podem executar esse comando. A forma de execução desse comando também depende do shell do Linux que você usa.

Para executá-lo, digite o comando abaixo:

**history [opção]**

Esse comando oferece suporte a várias opções, como:

- **-c** limpa a lista completa do histórico.

- **-d offset** exclui o registro do histórico na posição **OFFSET**.
- **-a** acrescenta linhas de histórico.

## 30. Comando man

O comando **man** fornece um manual completo para todos comandos ou utilitários que podem ser executados no Terminal, incluindo o nome, a descrição e as opções.

Ele consiste em nove seções:

- Programas executáveis ou comandos do shell
- Chamadas de sistema
- Chamadas da biblioteca
- Jogos
- Arquivos especiais
- Formatos e convenções de arquivos
- Comandos de administração do sistema
- Rotinas do kernel
- Diversos

Para exibir o manual completo, digite:

**man [nome\_do\_comando]**

Por exemplo, se você deseja acessar o manual do comando **ls**:

**man ls**

Digite esse comando se quiser especificar a seção exibida:

**man [opção] [número\_da\_seção] [\*nome\_do\_comando]\***

Por exemplo, você deseja ver a seção 2 do manual do comando **ls**:

**man 2 ls**

## 31. Comando echo

O comando **echo** é um utilitário nativo que exibe uma linha de texto ou cadeia de caracteres (string) usando a saída padrão. Veja a seguir a sintaxe básica:

**echo [opção] [string]**

Por exemplo, você pode exibir o texto **Hostinger Tutoriais** digitando:

**echo "Hostinger Tutoriais"**

Esse comando oferece suporte a várias opções, como:

- **-n** exibe a saída sem a nova linha final.
- **-e**

permite a interpretação dos seguintes escapes de barra invertida:

- **\a** reproduz um alerta sonoro.
- **\b** remove os espaços entre um texto.

- \c não produz mais nenhum resultado.
- -E exibe a opção padrão e desativa a interpretação dos escapes de barra invertida.

## 32. Comandos zip e unzip

Use o comando **zip** para compactar seus arquivos em um arquivo **ZIP**, um formato universal comumente usado no Linux. Ele pode escolher automaticamente a melhor taxa de compactação.

O comando **zip** também é útil para arquivar arquivos e diretórios e reduzir o uso do disco.

Para usá-lo, digite a seguinte sintaxe:

**zip [opções] zipfile file1 file2....**

Por exemplo, se você tem um arquivo chamado **note.txt** que deseja compactar em **archive.zip** no diretório atual:

**zip archive.zip note.txt**

Por outro lado, o [comando unzip](#) extrai os arquivos compactados de um arquivo. Este é o formato geral:

**unzip [opção] nome\_do\_arquivo.zip**

Portanto, para descompactar um arquivo chamado **archive.zip** no diretório atual, digite:

**unzip archive.zip**

## 33. Comando hostname

Execute o comando **hostname** para saber o nome do host do sistema. Você pode executá-lo com ou sem uma opção. Aqui está a sintaxe geral:

**hostname [opção]**

Há muitos sinalizadores opcionais a serem usados, incluindo:

- **-a** ou **-alias** exibe o alias do nome do host.
- **-A** ou **-all-fqdns** exibe o nome de domínio totalmente qualificado (FQDN) da máquina.
- **-i** ou **-ip-address** exibe o endereço IP da máquina.

Por exemplo, digite o seguinte comando para saber o endereço IP de seu computador:

**hostname -i**

## 34. Comandos useradd e userdel

O Linux é um sistema multiusuário, o que significa que mais de uma pessoa pode usá-lo simultaneamente. **useradd** é usado para criar uma nova conta, enquanto o comando **passwd** permite adicionar uma senha. Somente aqueles com privilégios de root ou **sudo** podem executar o comando **useradd**.

Quando você usa o comando **useradd**, ele realiza algumas alterações importantes:

- Edita os arquivos **/etc/passwd**, **/etc/shadow**, **/etc/group** e **/etc/gshadow** para as contas recém-criadas.
- Cria e preenche um diretório inicial para o usuário.
- Define as permissões e a propriedade dos arquivos no diretório inicial.

Aqui está a sintaxe básica:

**useradd [opção] nome\_de\_usuário**

Para definir a senha:

**passwd a\_senha\_escolhida**

Por exemplo, para adicionar uma nova pessoa chamada Paulo, digite o seguinte comando simultaneamente:

**useradd Paulo**

**passwd 123456789**

Para excluir uma conta de usuário, use o comando **userdel**:

**userdel nome\_de\_usuário**

## 35. Comando apt-get

O **apt-get** é uma ferramenta de linha de comando para lidar com as bibliotecas da Advanced Package Tool (APT) no Linux. Ele permite que você obtenha informações e pacotes de fontes autenticadas para gerenciar, atualizar, remover e instalar softwares e suas dependências.

A execução do comando **apt-get** exige que você tenha privilégios **sudo** ou root.

Aqui está a sintaxe principal:

**apt-get [opções] (comando)**

Esses são os comandos mais comuns que você pode adicionar ao **apt-get**:

- **update** sincroniza os arquivos de pacote de suas fontes.
- **upgrade** instala a versão mais recente de todos os pacotes instalados.
- **check** atualiza o cache de pacotes e verifica dependências quebradas.

## 36. comandos nano, vi e jed

O Linux permite que os usuários editem e gerenciem arquivos por meio de um editor de texto usando comandos como o **nano**, o **vi** ou o **jed**. O **nano** e o **vi** são nativos do sistema operacional, enquanto o **jed** precisa ser instalado.

O [comando nano](#) denota palavras-chave e pode funcionar com a maioria dos idiomas. Para usá-lo, digite o seguinte comando:

**nano [nome do arquivo]**

O **vi** usa dois modos operacionais para trabalhar: **insert** e **Command**. O **insert** é usado para editar e criar um arquivo de texto. Por outro lado, o **command** executa operações, como salvar, abrir, copiar e colar um arquivo.

Para usar o **vi** em um arquivo, digite:

**vi [nome do arquivo]**

O **jed** tem uma interface de menu suspenso que permite aos usuários executar ações sem digitar combinações ou comandos do teclado. Como o **vi**, ele tem modos para carregar módulos ou plugins para escrever textos específicos.

Para abrir o programa, basta digitar **jed** na linha de comando.

## 37. Comandos alias e unalias

O **alias** permite que você crie um atalho com a mesma funcionalidade de um comando, nome de arquivo ou texto. Quando executado, ele instrui o shell a substituir uma string por outra.

Para usar o comando **alias**, digite a seguinte sintaxe:

**alias Name=String**

Por exemplo, você deseja tornar **k** o alias do comando **kill**:

**alias k='kill'**

Por outro lado, o comando **unalias** exclui um alias existente.

Veja a seguir como é a sintaxe geral:

**unalias [nome\_do\_alias]**

## 38. Comando su

O comando switch user, ou **su**, permite executar um programa como um usuário diferente. Ele altera a conta administrativa na sessão de login atual. Esse comando é especialmente útil para acessar o sistema por meio de [SSH](#) ou usar o gerenciador de exibição da GUI quando o usuário raiz não está disponível.

Esta é a sintaxe geral do comando:

**su [opções] [nome de usuário [argumento]]**

Quando executado sem nenhuma opção ou argumento, o comando **su** é executado com privilégios de root. Ele solicitará que você se autentique e use os privilégios **sudo** temporariamente.

Aqui estão algumas que você pode usar:

- **-p** ou **-preserve-environment** mantém o mesmo ambiente de shell, composto por HOME, SHELL, USER e LOGNAME.
- **-s** ou **-shell** permite especificar um ambiente de shell diferente para execução.
- **-l** ou **-login** executa um script de login para mudar para um nome de usuário diferente. Para executá-lo, é necessário digitar a senha do usuário.

## 39. Comando htop

O comando **htop** é um programa interativo que monitora os recursos do sistema e os processos do servidor em tempo real. Ele está disponível na maioria das distribuições Linux e você pode instalá-lo usando o gerenciador de pacotes padrão.

Em comparação com o comando **top**, o **htop** tem muitos aprimoramentos e recursos adicionais, como a operação com o mouse e indicadores visuais.

Para usá-lo, execute o seguinte comando:

**htop [opções]**

Você também pode adicionar opções, como:

- **-d** ou **--delay** mostra o atraso entre as atualizações em décimos de segundos.
- **-C** ou **--no-color** ativa o modo monocromático.
- **-h** ou **--help** exibe a mensagem de ajuda e sai.

## 40. Comando ps

O status do processo, ou comando **ps**, produz um snapshot de todos os processos em execução em seu sistema. Os resultados estáticos são obtidos dos arquivos virtuais no sistema de arquivos **/proc**.

A execução do comando **ps** sem uma opção ou argumento listará os processos em execução no shell, juntamente com:

- A ID exclusiva do processo (**PID**)
- O tipo de terminal (**TTY**)
- O tempo de execução (**TIME**)
- O comando que inicia o processo (**CMD**)

Aqui estão algumas opções que você pode usar:

- **-T** exibe todos os processos associados à sessão atual do shell.
- **-u nome\_de\_usuário** lista os processos associados a um usuário específico.
- **-A** ou **-e** mostra todos os processos em execução.

## 41. Comando systemctl

### Listar Serviços

Listar todos os serviços no Linux. É só usar o comando:

```
sudo systemctl list-unit-files --type service --all
```

Caso deseje visualizar todos os serviços, tanto ativos quanto inativos, execute o seguinte comando:

```
systemctl list-units --type=service
```

Caso você queira saber apenas os serviços que estão ativos naquele momento, basta usar o comando junto com o **grep**. Assim:

```
sudo systemctl | grep running
```

### Verificando o Status de um Serviço

Para verificar o status de um serviço específico, use o seguinte comando:

```
systemctl status nome-do-servico
```

## Gerenciar Serviços Linux

Para começar um serviço no Linux, use o comando abaixo:

```
sudo systemctl start [service_name]
```

Se o serviço estiver configurado corretamente, ele vai iniciar logo em seguida. Agora, queremos pará-lo. Para isso, é só usar o comando:

```
sudo systemctl stop [service_name]
```

Enquanto isso, para verificar serviços Linux, ou seja, seu status de funcionamento, podemos usar:

```
sudo systemctl status [service_name]
```

Também é possível ter um serviço em execução enquanto o sistema operacional está sendo carregado. Use:

```
sudo systemctl enable [service_name]
```

Ou remova-o do carregamento inicial usando:

```
sudo systemctl disable [service_name]
```

Finalmente, é possível verificar qual porta está sendo usada por um serviço. Para isso, use o **netstat**.

Para instalá-lo no Ubuntu, execute:

```
sudo apt install netstat-nat
```

Caso esteja usando o CentOS 7, use:

```
yum install net-tools
```

Então, execute o seguinte comando:

```
sudo netstat -plnt
```

O resultado vai dar a você todas as informações sobre a rede de internet que estiver conectado.

Para listagem detalhada de algum erro que possa ocorrer, utilize o comando:

```
sudo journalctl -xeu tomcat.service
```

## Instalações

Antes de uma instalação execute uma atualização das listas de repositórios.

```
sudo apt update
```

# COMPARTILHAMENTO DE PASTAS COM O SAMBA - MÉTODO 1

Atualizar a lista de repositórios:

```
sudo apt update
```

Instalar o SAMBA

```
sudo apt install samba
```

Adicionar usuário ao grupo samba

```
sudo gpasswd -add $USER sambashare
```

Confirmar senha de usuário para o samba

```
sudo smbpasswd -a setic
```

Iniciar o serviço samba

```
sudo systemctl start smbd
```

Criar uma pasta para seu compartilhamento no diretório do usuário.

```
mkdir shared-folder
```

Configurar permissões da pasta

```
sudo chmod 777 nome_da_pasta
```

Para compartilhar a pasta podemos usar o arquivo do próprio Samba chamado: `smb.conf`. Este arquivo fica no diretório `/etc/samba`.

Algumas vezes esse arquivo vem com um exemplo de configuração, porém como vamos fazer a nossa própria configuração, podemos apagá-la.

```
cd /etc/samba
sudo cp smb.conf smb.conf.default # Cópia de segurança
sudo nano smb.conf # Abrir o arquivo smb.conf no editor de texto nano
```

Configura o arquivo da seguinte maneira:

```
#Adicione abaixo de workgroup
netbios name = O_nome_que_desejar_para_identificação_na_rede

# Na última linha adicione
```

```
path = caminho_da_pasta (pasta preferencialmente na pasta home do usuário)
public = yes
browseable = yes
writable = yes
comment = algum_comentário_que_desejar
read only = no
printable = no
guest ok = yes
create mask = 0700
directory mask = 0700
```

Reiniciar o samba

```
sudo systemctl restart smbd
```

## COMPARTILHAMENTO DE PASTAS COM O SAMBA - MÉTODO 2

### 1. Installing Samba

To install Samba, we run:

```
sudo apt update
sudo apt install samba
```

We can check if the installation was successful by running:

```
whereis samba
```

The following should be its output:

```
samba: /usr/sbin/samba /usr/lib/samba /etc/samba /usr/share/samba
/usr/share/man/man7/samba.7.gz /usr/share/man/man8/samba.8.gz
```

### 2. Setting up Samba

Now that Samba is installed, we need to create a directory for it to share:

```
mkdir /home/<username>/sambashare/
```

The command above creates a new folder `sambashare` in our home directory which we will share later.

The configuration file for Samba is located at `/etc/samba/smb.conf`. To add the new directory as a share, we edit the file by running:

```
sudo nano /etc/samba/smb.conf
```

At the bottom of the file, add the following lines:

```
[sambashare]
comment = Samba on Ubuntu
path = /home/username/sambashare
read only = no
browsable = yes
```

Then press `ctrl-o` to save and `ctrl-x` to exit from the *nano* text editor.

## What we've just added

- `comment`: A brief description of the share.
- `path`: The directory of our share.
- `read only`: Permission to modify the contents of the share folder is only granted when the value of this directive is `no`.
- `browsable`: When set to `yes`, file managers such as Ubuntu's default file manager will list this share under "Network" (it could also appear as browseable).

Now that we have our new share configured, save it and restart Samba for it to take effect:

```
sudo service smbd restart
```

Update the firewall rules to allow Samba traffic:

```
sudo ufw allow samba
```

## **3. Setting up User Accounts and Connecting to Share**

Since Samba doesn't use the system account password, we need to set up a Samba password for our user account:

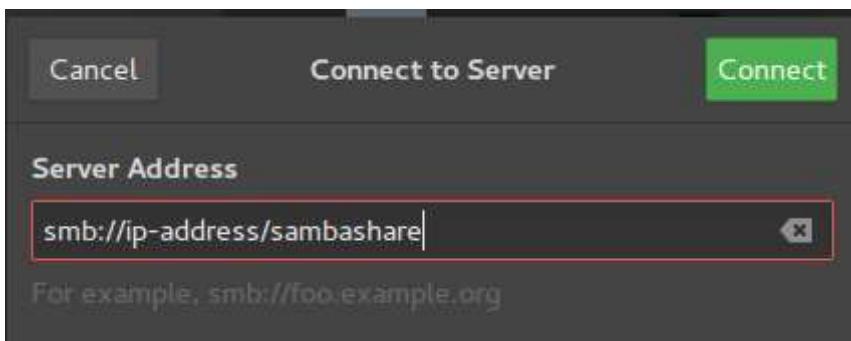
```
sudo smbpasswd -a username
```

### Note

Username used must belong to a system account, else it won't save.

## Connecting to Share

On Ubuntu: Open up the default file manager and click *Connect to Server* then enter:



On macOS: In the Finder menu, click *Go > Connect to Server* then enter:

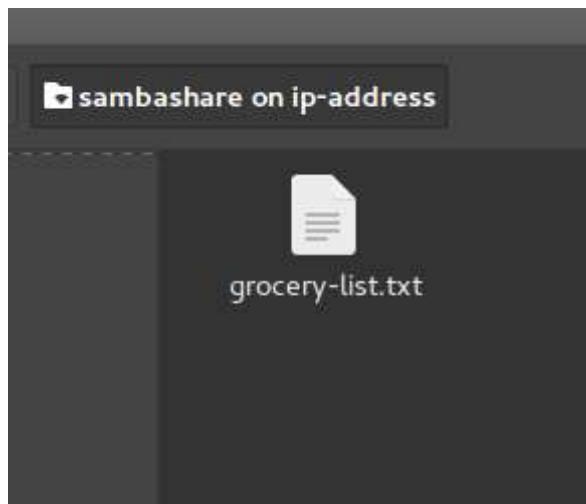


On Windows, open up File Manager and edit the file path to:

```
\ip-address\sambashare
```

Note: `ip-address` is the Samba server IP address and `sambashare` is the name of the share.

You'll be prompted for your credentials. Enter them to connect!



## If you'd like to take your configuration further...

- [Samba Server Guide](#)

## Tomcat

### Passo 1: Instalar o Java

Antes de instalarmos o Tomcat no Ubuntu, precisamos saber como instalar o Java no Ubuntu para executar o código da aplicação web em Java.

O OpenJDK é o padrão de desenvolvimento Java no Ubuntu 18.04. Instalar o Java é simples e rápido. Apenas siga os comandos abaixo:

```
sudo apt update
```

Instale o pacote [OpenJDK](#) executando:

```
sudo apt install default-jdk
```

Agora que o JDK está instalado no seu sistema, você pode criar o usuário Tomcat seguindo o próximo passo.

## Passo 2: Criar um Usuário Tomcat

Para sua segurança, você não deve usar o Tomcat sem um usuário único. Isso vai facilitar a instalação do Tomcat no Ubuntu. Crie um novo grupo do Tomcat que vai executar o serviço:

```
sudo groupadd tomcat
```

Agora, o próximo procedimento é criar um novo usuário Tomcat. Para criar usuários membros do grupo do Tomcat em um diretório principal, você precisa ir em **opt/tomcat** para executar o serviço Tomcat:

```
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

## Passo 3: Instalar Tomcat no Ubuntu

A melhor forma de instalar o Tomcat 9 no Ubuntu é baixar o mais recente lançamento dele da [página de download do Tomcat 9](#). E, assim, configurá-lo manualmente.

Se não for a versão 9.0.17 ou a versão mais recente, baixe, então, a última versão estável. Basta copiar o link do arquivo [tar.gz](#) principal na seção Distribuições Binárias.

Agora, mude para o diretório **/tmp** no seu servidor para baixar os itens que você não ai precisar depois de extrair o conteúdo Tomcat:

```
cd /tmp
```

Para baixar de um link copiado (do site do Tomcat), use o seguinte [comando curl](#):

```
curl -o https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.89/bin/apache-tomcat-9.0.89.tar.gz
```

Descompactar em /opt/tomcat

```
sudo tar xzvf apache-tomcat-9*tar.gz -c /opt/tomcat --strip-components=1
```

## Passo 4: Atualizar as Permissões

Agora que você acabou de instalar o Tomcat no Ubuntu, você precisa configurar o usuário Tomcat para ter total acesso à instalação do Tomcat.

Esse usuário precisa ter acesso ao diretório. Siga os seguintes passos abaixo:

```
sudo mkdir /opt/tomcat
```

```
cd /opt/tomcat
```

Agora, dê ao grupo Tomcat o controle sobre todo o diretório de instalação com o [comando chown](#):

```
sudo chgrp -R tomcat /opt/tomcat
```

Em seguida, você precisa dar ao usuário do Tomcat acesso ao diretório conf para pode visualizar seu conteúdo e executar o acesso à pasta dele:

```
sudo chmod -R g+r conf
```

```
sudo chmod g+x conf
```

Faça com que o usuário do Tomcat seja o dono dos diretórios webapps, work, tempo e logs:

```
sudo chown -R tomcat webapps/ work/ temp/ logs/
```

## Passo 5: Criar um Arquivo de Unidade do Systemd

(Continuar aqui)

Vamos precisar criar um novo arquivo único para executar o Tomcat como um serviço. Abra seu editor de texto e crie um arquivo com o nome de tomcat.service no diretório **/etc/systemd/system/**:

```
sudo nano /etc/systemd/system/tomcat.service
```

Em seguida, siga a seguinte configuração:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64/
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS.awt.headless=true -Djava.security.egd=file:/dev/urandom'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

Salve e feche o arquivo depois de terminar os comandos acima.

Em seguida, notifique o sistema que você criou um novo arquivo executando a seguinte linha de comando:

```
sudo systemctl daemon-reload
```

Os seguintes comandos vão permitir que você execute o serviço Tomcat:

```
sudo ./opt/tomcat/bin/startup.sh run
```

## Passo 6: Ajustar o Firewall

É essencial ajustar o firewall para que as solicitações cheguem ao serviço. O Tomcat usa a porta 8080 para aceitar solicitações convencionais. Permita tráfego para essa porta usando o [UFW](#):

```
sudo ufw allow 8080
```

Siga o comando abaixo para acessar a página de apresentação, indo pelo seu domínio ou endereço IP, em seguida por :8080 em um navegador – <http://IP:8080>

## Passo 7: Configurar a Interface Web de Gerenciamento Tomcat

Siga o seguinte comando abaixo para adicionar um login no seu usuário Tomcat e edite o arquivo **tomcat-users.xml**:

```
sudo nano /opt/tomcat/conf/tomcat-users.xml
```

Agora, defina o usuário que poderá acessar os arquivos e adicione o nome de usuário e senha:

```
tomcat-users.xml - Admin User

<tomcat-users . . .>

<tomcat-users . . .>

<user username="admin" password="password" roles="manager-gui,admin-gui"/>

</tomcat-users>
```

Para o gerenciador da aplicação, digite:

```
sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml
```

Para o aplicativo gerenciador do host, digite:

```
sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml
```

Para reiniciar o serviço Tomcat e visualizar os efeitos:

```
sudo systemctl restart tomcat
```

## Passo 8: Acessar a Interface Online

Agora que você já tem um usuário, você pode acessar a interface web de gerenciamento em um navegador.

Você pode acessar a interface pelo seu domínio ou endereço IP seguido pela porta 8080 no seu navegador – [http://dominio\\_servidor\\_ou\\_IP:8080](http://dominio_servidor_ou_IP:8080).

Vamos olhar para o gerenciador da aplicação, acessível pelo link – [http://dominio\\_servidor\\_ou\\_IP:8080/manager/html](http://dominio_servidor_ou_IP:8080/manager/html).

Garanta que você colocou as credenciais da conta no arquivo **tomcat-users.xml**.

Usamos um **Gerenciador de Aplicações Web** para gerenciar nossas aplicações Java. Com ele, você pode Começar, Parar, Recarregar, Lançar ou Remover todos os aplicativos. Por último, ele providencia dados sobre seu servidor na parte inferior da página.

Agora vamos para o **Gerenciador do Host**, acessível via [http://dominio\\_servidor\\_ou\\_IP:8080/host-manager/html](http://dominio_servidor_ou_IP:8080/host-manager/html).

Da página do **Gerenciador Virtual do Host**, você pode também adicionar novos hosts virtuais para seguir os formulários da sua aplicação, editá-los, criá-los ou apagá-los.

## Clickhouse(SGDB)

### 1. Download the binary (instalação para testes)

ClickHouse runs natively on Linux, FreeBSD and macOS, and runs on Windows via the [WSL](#). The simplest way to download ClickHouse locally is to run the following `curl` command. It determines if your operating system is supported, then downloads an appropriate ClickHouse binary:

```
curl https://clickhouse.com/ | sh
```

### 2. Start the server

Run the following command to start the ClickHouse server:

```
./clickhouse server
```

### 3. Start the client

Use the `clickhouse-client` to connect to your ClickHouse service. Open a new Terminal, change directories to where your `clickhouse` binary is saved, and run the following command:

```
./clickhouse client
```

You should see a smiling face as it connects to your service running on localhost:

```
my-host :)
```

## 4. Effective installation

```
sudo ./clickhouse install
```

# Como encontrar o PID de um processo que está usando uma porta no Linux?

## 1. Download the bi

### 1. Como listar processos e portas com o SS?

O comando `ss` é a [alternativa moderna](#) para nosso querido `netstat`. O funcionamento e parâmetros são muito parecidos, inclusive, utilizamos o mesmo `-tlunp` para obter as portas, PIDs e nomes dos processos.

Você pode utilizar o `ss` de duas maneiras, ou filtrando via `* | grep ":PORTA"` assim como fizemos no netstat ou então usando o próprio parâmetro de `sport` do ss. Eu fiz os dois exemplos de uso. Pessoalmente prefico usar o grep sempre.

```
# Comando ss utilizando o filtro source port
sudo ss -tlunp 'sport = :80'

# Output
Netid      State      Recv-Q      Send-Q
          Local Address:Port          Peer Address:Port
Process

tcp        LISTEN      0          511
          0.0.0.0:80          0.0.0.0:*
          users:(("nginx",pid=560,fd=8),("nginx",pid=559,fd=8),("nginx",pid=558,fd=8),
          ("nginx",pid=557,fd=8),("nginx",pid=556,fd=8))

tcp        LISTEN      0          511
          [::]:80            [::]:*
          users:(("nginx",pid=560,fd=9),("nginx",pid=559,fd=9),("nginx",pid=558,fd=9),
          ("nginx",pid=557,fd=9),("nginx",pid=556,fd=9))

# Comando ss utilizando filtro pelo | grep
sudo ss -tlunp | grep :80
```

```
# Output
tcp    LISTEN  0      511          0.0.0.0:80          0.0.0.0:*      users:
(("nginx",pid=560,fd=8),("nginx",pid=559,fd=8),("nginx",pid=558,fd=8),
 ("nginx",pid=557,fd=8),("nginx",pid=556,fd=8))
tcp    LISTEN  0      511          [::]:80          [::]:*      users:
(("nginx",pid=560,fd=9),("nginx",pid=559,fd=9),("nginx",pid=558,fd=9),
 ("nginx",pid=557,fd=9),("nginx",pid=556,fd=9))
```

O comando **netstat** foi descontinuado e não deve ser usado, usaremos então o comando **ss**.

```
sudo ss -tulwn | grep LISTEN
```

As opções usadas são:

- **-t** Mostra apenas conexões TCP
- **-u** Exibe apenas conexões UDP
- **-l** Mostra conexões de escuta (por exemplo, a porta 22 aberta pelo servidor SSHD)
- **-p** Lista o nome do processo que abriu a conexão
- **-n** Não resolve nomes de serviço

Se você visualiza uma porta, mas não sabe a qual serviço ela pertence. Consulte esta lista:

```
less /etc/services
```

## 2. Como listar processos e portas com o LSOF?

O comando **lsof** também é outro muito comum e muito utilizado por administradores Linux. Temos um artigo que fala somente sobre arquivos abertos no Linux.

O uso **lsof** é muito simples e, para o buscar uma porta aberta com ele, basta utilizar o parâmetro **-i :PORTA** que ele irá listar os PIDs, processos e portas de uma forma até mais completa que o próprio ss e netstat.

```
# Comando lsof com o filtro de porta -i
sudo lsof -i :80

# Output
COMMAND PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
nginx  556   root    8u  IPv4    909      0t0  TCP *:http (LISTEN)
nginx  556   root    9u  IPv6    910      0t0  TCP *:http (LISTEN)
nginx  557 www-data    8u  IPv4    909      0t0  TCP *:http (LISTEN)
nginx  557 www-data    9u  IPv6    910      0t0  TCP *:http (LISTEN)
nginx  558 www-data    8u  IPv4    909      0t0  TCP *:http (LISTEN)
nginx  558 www-data    9u  IPv6    910      0t0  TCP *:http (LISTEN)
nginx  559 www-data    8u  IPv4    909      0t0  TCP *:http (LISTEN)
nginx  559 www-data    9u  IPv6    910      0t0  TCP *:http (LISTEN)
nginx  560 www-data    8u  IPv4    909      0t0  TCP *:http (LISTEN)
nginx  560 www-data    9u  IPv6    910      0t0  TCP *:http (LISTEN)
```

### 3. Como listar processos e portas com o FUSER?

Das 3 opções anteriores, é possível que o comando `fuser` seja o menos comum de todos. Ele serve para exibir informações de processos e os usos de arquivos. Se você quiser APENAS o PID dos processos de uma determinada porta, sem dúvidas o `fuser` é a escolha certa.

O uso `**fuser**` é também muito simples e, para o buscar uma porta aberta com ele, basta utilizar o parâmetro `**-v PORTA/PROTOCOLO**` que ele irá listar os PIDs, processos e portas. O parâmetro `-v` é um verbose para retornar mais informações. Você pode também utilizar direto buscando a porta.

```
# Comando fuser com verbose e filtro de porta
sudo fuser -v 80/tcp

# Output
          USER      PID ACCESS COMMAND
80/tcp:    root      556 F.... nginx
           www-data  557 F.... nginx
           www-data  558 F.... nginx
           www-data  559 F.... nginx
           www-data  560 F.... nginx

# Comando fuser de maneira super simplificada, APENAS PIDs
sudo fuser 80/tcp

# Output
80/tcp:      556   557   558   559   560
```

## Instalar SSH

### Step 1: Prepare Ubuntu

The first thing you need to do before you start installing SSH on Ubuntu is to update all `apt` packages to the latest versions. To do this, use the following command:

```
sudo apt update && sudo apt upgrade
```

### Step 2: Install SSH on Ubuntu

OpenSSH is not pre-installed on the system, so let's install it manually. To do this, type in the terminal:

```
sudo apt install openssh-server
```

The installation of all the necessary components will begin. Answer "Yes" to all the system prompts.

After the installation is complete, go to the next step to start the service.

## Step 3: Start SSH

---

Now you need to enable the service you just installed using the command below:

```
sudo systemctl enable --now ssh
```

On successful startup, you will see the following system message.

The `--now` key helps you launch the service and simultaneously set it to start when the system boots.

To verify that the service is enabled and running successfully, type:

```
sudo systemctl status ssh
```

The output should contain the `Active: active (running)` line, which indicates that the service is successfully running.

If you want to disable the service, execute:

```
sudo systemctl disable ssh
```

It disables the service and prevents it from starting at boot.

## Step 4: Configure the firewall

---

Before connecting to the server via SSH, check the firewall to ensure it is configured correctly.

In our case, we have the UFW installed, so we will use the following command:

```
sudo ufw status
```

In the output, you should see that SSH traffic is allowed. If you don't have it listed, you need to allow incoming SSH connections. This command will help with this:

```
sudo ufw allow ssh
```

## Step 5: Connect to the server

---

Once you complete all the previous steps, you can log into the server using the SSH protocol.

To do this, you will need the server's IP address or domain name and the name of a user created on the server.

In the terminal line, enter the command:

```
ssh username@IP_address
```

Or:

```
ssh username@domain
```

Important: To successfully connect to a remote server, SSH must be installed and configured on the remote server and the user's computer from which you make the connection.