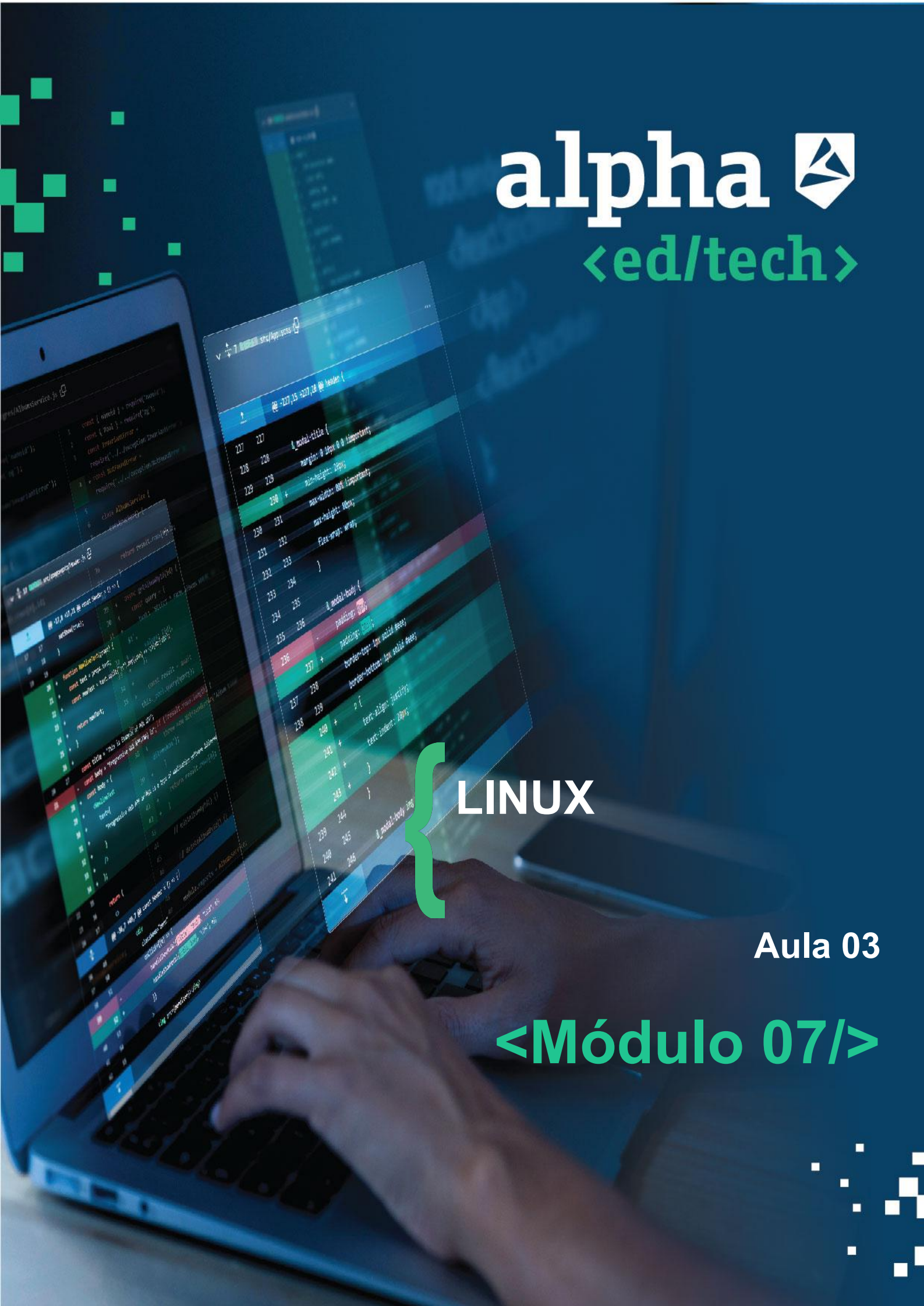

alpha
<ed/tech>
LINUX
Aula 03
<Módulo 07/>

Introdução Usuários e Grupos de Usuários

Abordaremos aspectos fundamentais para o gerenciamento de usuários e permissões no Linux. Exploraremos a criação e administração de usuários, juntamente com a atribuição de permissões específicas a arquivos e diretórios. Veremos também como estabelecer e gerenciar grupos, agregando usuários para facilitar a gestão de permissões em larga escala. Além disso, discutiremos a troca eficiente entre usuários no ambiente Linux, proporcionando uma compreensão abrangente do controle de acessos e segurança no sistema operacional. Prepare-se para aprofundar seus conhecimentos em administração de usuários e permissões no Linux.

Iniciar a VM

Você pode iniciar a VirtualBox em background. Nesse caso para iniciar a VM headless mode (lembre-se de substituir pelo nome da sua VM), aplique esse comando:

VBoxManage startvm "UbuntuServer22.04-alpha" --type headless

```
letonio.silva@BRRIOLN043879:~/Documentos/alphaedtech/examples $ VBoxManage startvm "UbuntuServer2
2.04-alpha" --type headless
Waiting for VM "UbuntuServer22.04-alpha" to power on...
VM "UbuntuServer22.04-alpha" has been successfully started.
letonio.silva@BRRIOLN043879:~/Documentos/alphaedtech/examples $
```

Nesse ponto, é irrelevante se você iniciou o software da VirtualBox clicando no programa ou em background. Entretando, recomenda-se que login na VM seja feito através do terminal da sua máquina local, para que você possa fazer uso dos recursos de copiar e colar texto no terminal.

Já aprendemos como fazer o login usando secure shell (SSH). Se a sua máquina local é Linux, MacOS ou Windows, ela já conta com um client SSH embutido. Portanto, você pode fazer acesso à máquina virtual através desse comando:

ssh usuario@endereco_ip_da_vm

A autenticação é feita com login/senha. A imagem a seguir mostra o terminal da minha máquina local após o login. O nome do meu usuário é "lets", enquanto "serverlets" é o nome do servidor.

```
lets@serverlets:~$
```

Comando whoami

É utilizado para exibir o nome do usuário atual, ou seja, exibe seu nome de login. Basta usar whoami e pressionar Enter.

```
lets@serverlets:~$ whoami
lets
lets@serverlets:~$
```

Usuários do sistema

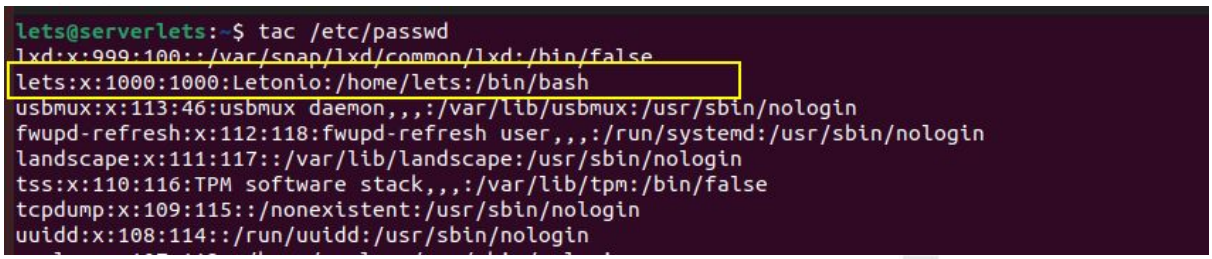
O arquivo **/etc/passwd** é um arquivo de texto que armazena informações sobre usuários do sistema. Cada linha no arquivo representa um usuário e contém várias informações separadas por dois-pontos (:).

A estrutura básica de uma linha no /etc/passwd é a seguinte:

username:password:UID:GID:GECOS:directory:shell

Na imagem abaixo, aplicou-se o comando **cat /etc/passwd** para mostrar as linhas do arquivo na ordem inversa (da última para a primeira). Destacou-se a segunda linha:

lets:x:1000:1000:Letonio:/home/lets:/bin/bash



Uma breve explicação sobre cada elemento:

- **username:** Nome do usuário.
- **password:** Atualmente, fixa um carácter **x** ou ***** para indicar que o usuário tem uma senha associada a ele. Essa senha, por razões de segurança, tem uma referência criptografada armazenada em /etc/shadow.
- **UID (User ID):** Identificador único do usuário, usado internamente pelo sistema para identificar usuários.
- **GID (Group ID):** Identificador do grupo primário ao qual o usuário pertence.
- **GECOS (General Electric Comprehensive Operating System):** Tradicionalmente, esse campo continha informações como o nome completo do usuário, número de telefone, etc. No caos da imagem acima, está salvo apenas o nome. Se você tem uma boa memória, lembrará que durante a instalação do Ubuntu server, fornecemos nossos nomes. Muitas distribuições Linux usam esse campo para informações adicionais sobre o usuário.
- **directory:** O diretório inicial (home directory) do usuário.
- **shell:** O interpretador de comandos padrão associado ao usuário. Na imagem acima é um executável em /bin/bash.

Adicionar usuário

O comando oficial para criar novos usuários no sistema é **useradd**. Trata-se de um comando de mais baixo nível que adiciona o usuário, mas não configura automaticamente alguns aspectos adicionais, como a criação do diretório inicial do usuário ou a atribuição de grupos. Por conta disso, algumas pessoas preferem um comando alternativo, que é o **adduser**. Na verdade, **adduser** é um script de alto nível que executa o useradd com algumas configurações adicionais. Ele simplifica o processo de adição de um usuário, automatizando tarefas comuns, como a criação do diretório inicial do usuário, a adição do usuário a grupos padrão, entre outras.

Comando useradd

Você pode usar o argumento **--help** ou **-h** para ver as opções disponíveis. Para fazer a adição de um usuário, incluindo um diretório home para este novo usuário, podemos usar a flag **-m**. Portanto, para adicionar um usuário chamado joao, podemos aplicar o comando a seguir:

sudo useradd -m joao

```
lets@serverlets:~$ sudo useradd -m joao
[sudo] password for lets:
lets@serverlets:~$
```

Comando passwd

Na seção anterior, apenas criamos um usuário novo. Para completar o registro, precisamos adicionar uma senha para ele. Isso é feito através do comando:

sudo passwd joao

```
lets@serverlets:~$ sudo passwd joao
New password:
Retype new password:
passwd: password updated successfully
lets@serverlets:~$
```

Comando adduser

Alternativamente, usando o script de alto nível adduser, recebemos algumas perguntas e baseado nisso o novo usuário é criado. Vamos criar um segundo usuário, dessa vez chamado felipe.

sudo adduser felipe

```
lets@serverlets:~$ sudo adduser felipe
Adding user `felipe' ...
Adding new group `felipe' (1002) ...
Adding new user `felipe' (1002) with group `felipe' ...
Creating home directory `/home/felipe' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for felipe
Enter the new value, or press ENTER for the default
        Full Name []: felipe santos
        Room Number []: 14
        Work Phone []: 2199999123
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

Usuário criado e fornecemos diversas informações. Se preferir, pode deixar vazio (pressione Enter), o importante é criar uma senha.

Podemos visualizar os usuários criados no arquivo /etc/passwd. Optei por aplicar esse comando:

tac /etc/passwd

```
lets@serverlets:~$ tac /etc/passwd
felipe:x:1002:1002:felipe santos,14,2199999123,:/home/felipe:/bin/bash
joao:x:1001:1001::/home/joao:/bin/sh
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
lets:x:1000:1000:Letonio:/home/lets:/bin/bash
usbmux:x:113:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
fwupd-refresh:x:112:118:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
landscape:x:111:117::/var/lib/landscape:/usr/sbin/nologin
```

Além disso, nota-se que foram criados diretórios para cada usuário, conforme ilustrado na imagem abaixo:

cd /home && ls

```
lets@serverlets:/home$ cd /home && ls
felipe  joao  lets
lets@serverlets:/home$
```

Remover usuário

Para ilustrar como remover um usuário, vamos remover o felipe. Aplica-se esse comando:

sudo userdel -r felipe

O argumento -r serve para remover o diretório inicial associado àquele usuário e sua caixa de correio. Caso você desejasse manter o diretório /home/felipe, não passaria a flag -r.
Na figura abaixo, nota-se que o diretório foi removido e não tem nenhuma caixa de correio associada ao felipe.

```
lets@serverlets:/home$ sudo userdel -r felipe
[sudo] password for lets:
userdel: felipe mail spool (/var/mail/felipe) not found
lets@serverlets:/home$ cd /home && ls
joao  lets
lets@serverlets:/home$
```

Trocar de usuário

Para trocar de usuário você pode aplicar o comando **su <usuario>**. O termo **su** vem de **substitute user**. Por exemplo, você pode utilizar o comando **sudo adduser fulano** para adicionar um novo usuário chamado fulano. Em seguida, aplique:

su fulano

```
lets@serverlets:~$ su fulano
Password:
fulano@serverlets:/home/lets$ ls /home
fulano  joao  lets
fulano@serverlets:/home/lets$ su lets
Password:
lets@serverlets:~$ pwd
/home/lets
lets@serverlets:~$
```

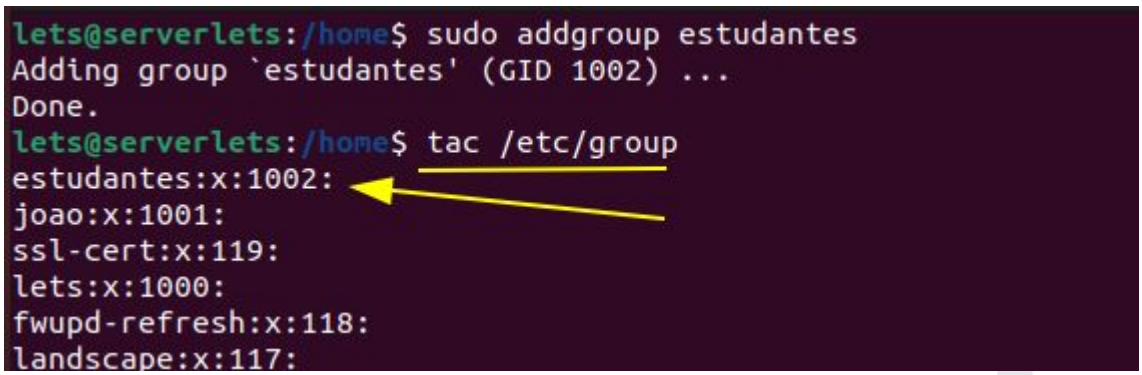
Criar grupo

Criar grupos pode simplificar a administração em ambientes com um grande número de usuários ou quando é necessário segmentar (por exemplo, por departamentos, equipes, divisões). Essa prática ajuda a organizar de forma mais eficiente o acesso a arquivos e recursos. Vamos criar um grupo chamado estudantes:

sudo addgroup estudantes

Para confirmar que o grupo foi criado com sucesso, basta acessar o arquivo /etc/group.

tac /etc/group



Remover grupo

Para remover um grupo, use esse comando:

sudo delgroup <grupo>

Adicionar usuário a um grupo

Para ilustrar um exemplo de adição de um usuário a um grupo existente, vamos considerar que desejamos adicionar joao ao grupo de estudantes.

sudo usermod -a -G <grupo> <usuário>
sudo usermod -a -G estudantes joao

Opcionalmente, poderíamos escrever -aG em vez de -a -G (--append --groups). Essas opções indicam que a intenção é de adicionar ao grupo.

Listar grupo

É possível listar os grupos dos quais o usuário atual faz parte por meio do comando **groups**.

groups

Para listar os grupos de um usuário específico, podemos usar o comando id <usuário>. Por exemplo, para listar os grupos do usuário joao, podemos aplicar esse comando:

id joao

```
lets@serverlets:/home$ groups
lets adm cdrom sudo dip plugdev lxd
lets@serverlets:/home$ id joao
uid=1001(joao) gid=1001(joao) groups=1001(joao),1002(estudantes)
lets@serverlets:/home$
```

Remover usuário de um grupo

A sintaxe básica para remover um usuário de um grupo é a seguinte:

sudo gpasswd -d <usuário> <grupo>

Na imagem abaixo, é possível observar o usuário "joao" sendo removido do grupo "estudantes".

```
lets@serverlets:~$ sudo gpasswd -d joao estudantes
[sudo] password for lets:
Removing user joao from group estudantes
lets@serverlets:~$ id joao
uid=1001(joao) gid=1001(joao) groups=1001(joao)
lets@serverlets:~$
```

A Importância de usuários e grupos

Você pode estar se questionando sobre a necessidade de tantos comandos para configurar grupos e usuários, e por que não adotar apenas um usuário? A explicação é direta: usuários e grupos possuem a capacidade de receber diferentes conjuntos de permissões. Em outras palavras, certos usuários podem enfrentar restrições específicas quanto ao acesso a pastas e arquivos. Essa distinção também pode ser estendida ao nível de grupos. Esses elementos desempenham funções cruciais na administração de permissões, garantindo a segurança e organização eficientes dos recursos do sistema.

Cada usuário é atribuído a suas próprias permissões, e a utilização de grupos facilita a agregação de usuários com necessidades semelhantes, simplificando a administração de permissões. Ao empregar grupos, os administradores têm a capacidade de conceder apenas as permissões estritamente necessárias para a realização de tarefas específicas, seguindo o princípio do mínimo privilégio. Essa abordagem contribui significativamente para a redução dos riscos de segurança. A organização de usuários em grupos otimiza a administração do sistema, permitindo que as configurações sejam aplicadas a conjuntos completos de usuários, em vez de individualmente, o que simplifica as operações de gerenciamento.

Permissões

Cada arquivo e diretório tem um **user owner** e um **group owner**. O **user owner** tem permissões especiais sobre o arquivo ou diretório, incluindo a capacidade de alterar as permissões e o nome do arquivo, bem como excluir o arquivo. A identificação do user owner é geralmente associada ao UID (User ID) do usuário no sistema.

O **group owner** é o grupo ao qual o arquivo ou pasta está associado. Os membros do grupo têm permissões específicas em relação ao arquivo ou pasta, dependendo das configurações de permissões do grupo. A identificação do group owner está vinculada ao GID (Group ID) do grupo no sistema. O GID é um identificador único atribuído a cada grupo.

As permissões de leitura (**r**->read), gravação (**w**->write) e execução (**x**->execute) são atribuídas user owner, ao group owner e a outros usuários (others), e essas permissões determinam quem pode realizar quais ações em um determinado arquivo ou pasta.

Mudar o dono de um arquivo/diretório

Para alterar o proprietário de um arquivo ou diretório no Linux, podemos utilizar o comando **chown** (change owner). Aqui está a sintaxe básica:

sudo chown <novo_proprietario>:<novo_grupo> <arquivo_ou_diretorio>

Para exemplificar, primeiro vamos criar um diretório chamado "page" com dois arquivos dentro dele: index.html e style.css.

```
lets@serverlets:~$ mkdir page
lets@serverlets:~$ ls
page
lets@serverlets:~$ cd page && touch index.html style.css
lets@serverlets:~/page$ ls
index.html  style.css
lets@serverlets:~/page$
```

Vamos aplicar o comando **cd .. && ls -R -l** para subirmos um nível na hierarquia de diretórios e, em seguida, listar de forma recursiva os detalhes sobre os arquivos e diretório que acabamos de criar.

```
lets@serverlets:~/page$ cd .. && ls -R -l
.:
total 4
drwxrwxr-x 2 lets lets 4096 Feb 11 12:10 page

./page:
total 0
-rw-rw-r-- 1 lets lets 0 Feb 11 12:10 index.html
-rw-rw-r-- 1 lets lets 0 Feb 11 12:10 style.css
lets@serverlets:~$
```

Observe que na linha do diretório "page", temos à esquerda **drwxrwxr-x**. Trata-se de um sistema para identificar as permissões relacionadas ao diretório "page". A letra **d** indica que o elemento é um diretório. No caso de arquivos, a letra é substituída por traço (-).

Em seguida, vemos **rwx rwx r-x**. O primeiro conjunto indica que o dono (**user owner**) tem permissão de ler, escrever e executar o conteúdo do diretório; O segundo conjunto indica que o grupo (por padrão, quando não é informado o **group owner**, ele passa a ser um grupo com o mesmo nome do usuário) tem as mesmas permissões (ler o conteúdo, gravar alterações e executar). Por fim, o terceiro conjunto está indicando as permissões de outros usuários (others). O traço (-) no lugar do w indica que outros usuários não têm permissão para gravar alterações nos arquivos, apenas ler e executar (script são exemplos de arquivos que podem ser executados).

De forma semelhante, percebemos que o arquivo index.html permite que o dono e pessoas pertencentes ao grupo possam ler e escrever (rw-), mas não executar. Isso acontece porque o index.html não é um arquivo executável. Por fim, outros usuários podem apenas ler o conteúdo (r--).
Usando o comando chown troque o dono do arquivo index.html

sudo chown joao:estudantes ./page/index.html

```
lets@serverlets:~$ sudo chown joao:estudantes ./page/index.html
[sudo] password for lets:
lets@serverlets:~$ cd page && ls -l
total 0
-rw-rw-r-- 1 joao estudantes 0 Feb 11 12:10 index.html
-rw-rw-r-- 1 lets lets       0 Feb 11 12:10 style.css
lets@serverlets:~/page$
```

Tendo em mente que o usuário atual é o lets, que não faz parte do grupo estudantes e nem é o dono do arquivo index.html, percebe-se que ele tem apenas permissão para leitura. Vamos tentar abrir o arquivo usando o editor de texto "nano". Uma notificação aparece na parte inferior, indicando que o arquivo não permite escrita.

```
 GNU nano 6.2                    index.html

                [ File 'index.html' is unwritable ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Podemos usar a opção -R para modificar de forma recursiva o dono/grupo de todos os arquivos/subdiretórios. Por exemplo, para adicionar joao e estudantes como dono e grupo, respectivamente, aplica-se esse comando:

sudo chown -R joao:estudantes ./page

```
lets@serverlets:~$ ls
page
lets@serverlets:~$ sudo chown -R joao:estudantes ./page
[sudo] password for lets:
lets@serverlets:~$ ls -R -l
.:
total 4
drwxrwxr-x 2 joao estudantes 4096 Feb 11 12:40 page

./page:
total 0
-rw-rw-r-- 1 joao estudantes 0 Feb 11 12:10 index.html
-rw-rw-r-- 1 joao estudantes 0 Feb 11 12:10 style.css
lets@serverlets:~$
```

Mudar apenas o grupo/usuário

Para modificar apenas o grupo omita o nome do usuário, conforme ilustrado abaixo:

sudo chown :<grupo> <arquivo_ou_diretorio>

Para modificar apenas o dono a ideia é parecida, omita o grupo:

sudo chown <novo_dono>: <arquivo_ou_diretorio>

A seguir, mostra-se um exemplo onde mudamos o nome do grupo para "aspirantes" e depois, voltamos ao "estudantes":

sudo chown :<grupo> <arquivo_ou_diretorio>

```
lets@serverlets:~/page$ sudo addgroup aspirantes
Adding group `aspirantes' (GID 1004) ...
Done.
lets@serverlets:~/page$ sudo chown :aspirantes index.html
lets@serverlets:~/page$ ls -l
total 4
-rwxrw-r-- 1 joao aspirantes 30 Feb 11 13:20 index.html
-rw-rw-r-- 1 joao estudantes  0 Feb 11 12:10 style.css
lets@serverlets:~/page$ sudo chown :estudantes index.html
lets@serverlets:~/page$ ls -l
total 4
-rwxrw-r-- 1 joao estudantes 30 Feb 11 13:20 index.html
-rw-rw-r-- 1 joao estudantes  0 Feb 11 12:10 style.css
lets@serverlets:~/page$
```

Modificar permissões

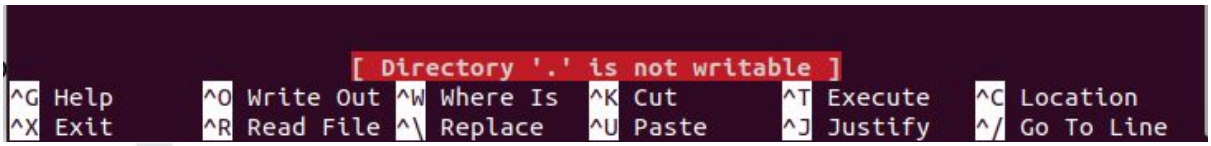
Note que as permissões mostradas para os arquivos e diretórios criados na seção anterior são o padrão que o Linux adota. No entanto, podemos personalizar essas permissões a vontade. Por exemplo, vamos habilitar que outros usuários (others) possam gravar alterações no arquivo index.html. Isso é feito através do comando **chmod** (change mode). O comando ficaria assim:

sudo chmod o+w index.html

Sobre o comando acima, estamos indicando que others (o) deve receber permissão de escrita (+w) em relação ao arquivo index.html.

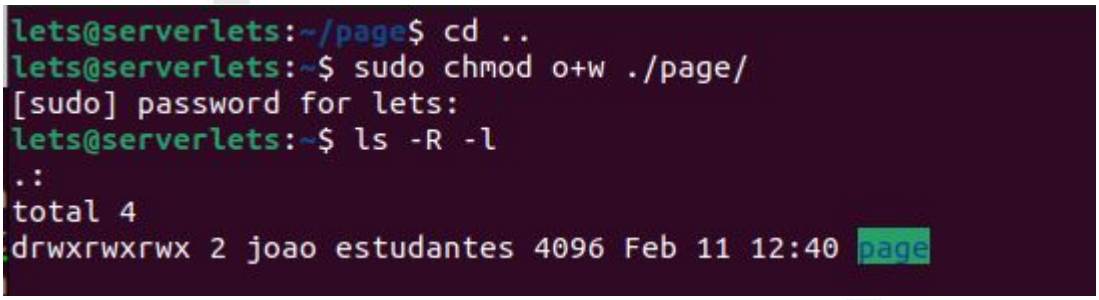
```
lets@serverlets:~/page$ ls -l
total 0
-rw-rw-r-- 1 joao estudantes 0 Feb 11 12:10 index.html
-rw-rw-r-- 1 joao estudantes 0 Feb 11 12:10 style.css
lets@serverlets:~/page$ sudo chmod o+w index.html
lets@serverlets:~/page$ ls -l
total 0
-rw-rw-rw- 1 joao estudantes 0 Feb 11 12:10 index.html
-rw-rw-r-- 1 joao estudantes 0 Feb 11 12:10 style.css
lets@serverlets:~/page$
```

Com a mudança aplicada, tente usar o editor de texto "nano" para fazer alguma alteração no arquivo index.html. Observe que a interface do nano tem uma mensagem no canto inferior. Essa mensagem indica que o diretório atual não tem permissão de escrita para outros usuários (others).

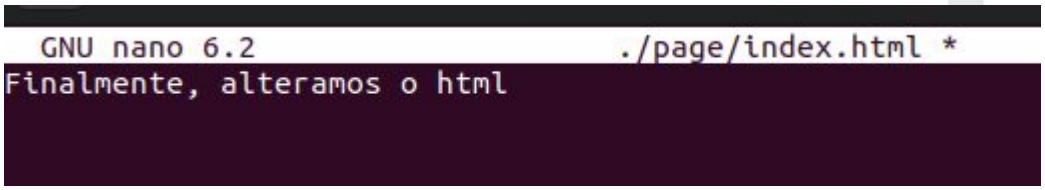


Vamos aplicar a mudança de permissão no diretório também:

sudo chmod o+w ./page



Finalmente, ao acessar o arquivo index.html através do editor nano, não temos nenhum aviso sobre problemas com gravação.



Tendo Imagine que desejamos remover as permissões de leitura e gravação para o grupo "estudantes" e outros usuários (others) em relação ao arquivo index.html. O comando ficaria assim:

sudo chmod go-rw index.html

Alternativamente, em vez de informar símbolos como g, o, go, a, +r, +w, -r, -w, -x, -rw, etc. Podemos usar números para indicar especificamente as permissões que desejamos aplicar.
Neste modo, as permissões de arquivo não são representadas como caracteres, mas como um número **octal** de três dígitos. A tabela a seguir ilustra as possibilidades de um octal:

Number	Tipo de permissão	Símbolo
0	No permission	---
1	Execute	--x
2	Write	-w-
3	Execute+Write	-wx
4	Read	r--
5	Read+Execute	r-x
6	Read+Write	rw-
7	Read+Write+Execute	rwx

Considere a seguinte situação:

- user owner - Pode ler, gravar e executar (7);
- group owner - Pode ler e gravar (6);
- others - Pode apenas ler (4);

Logo, o comando para aplicar esse nível de permissão ficaria:

sudo chmod 764 index.html

```
lets@serverlets:~/page$ ls -l
total 4
-rw------- 1 joao estudantes 30 Feb 11 13:20 index.html
-rw-rw-r-- 1 joao estudantes  0 Feb 11 12:10 style.css
lets@serverlets:~/page$ sudo chmod 764 index.html
[sudo] password for lets:
lets@serverlets:~/page$ ls -l
total 4
-rwxrw-r-- 1 joao estudantes 30 Feb 11 13:20 index.html
-rw-rw-r-- 1 joao estudantes  0 Feb 11 12:10 style.css
lets@serverlets:~/page$
```

Mudar dados de login

Para mudar o nome de usuário, aplique o comando usermod com a opção -l (--login):

sudo usermod -l <username_novo> <username_antigo>

sudo usermod -l joao_admin joao

```
lets@serverlets:~/page$ sudo usermod -l joao_admin joao
lets@serverlets:~/page$ ls -l
total 4
-rwxrw-r-- 1 joao_admin estudantes 30 Feb 11 13:20 index.html
-rw-rw-r-- 1 joao_admin estudantes  0 Feb 11 12:10 style.css
```

Para mudar a senha de um usuário, use a seguinte sintaxe:

sudo passwd <usuario>

```
lets@serverlets:~/page$ sudo passwd joao_admin
New password:
Retype new password:
passwd: password updated successfully
lets@serverlets:~/page$
```

Bloquear e desbloquear uma conta

Imagine que um invasor teve acesso às credenciais de um funcionário da sua companhia e acessa de forma indevida o seu sistema. Você pode bloquear aquele login para que ele não cause mais estragos. Use esse comando para bloquear (lock):

sudo usermod -L <usuário>

Utilize a opção -U para desbloquear (unlock):

sudo usermod -U <usuário>

Para ilustrar essa situação, primeiro vamos bloquear o usuário joao_admin.

sudo usermod -L joao_admin

```
lets@serverlets:~/page$ sudo usermod -L joao_admin
lets@serverlets:~/page$
```

Podemos ver o status da senha de joao_admin através desse comando:

sudo passwd -S joao_admin

```
lets@serverlets:~/page$ sudo usermod -L joao_admin
lets@serverlets:~/page$ sudo passwd -S joao_admin
joao_admin L 02/11/2024 0 99999 7 -1
```

A letra L indica que a senha está bloqueada.

Em seguida, via outro terminal, vamos tentar fazer o login usando joao_admin:

```
letonio.silva@BRRIOLN043879:~ $ ssh joao_admin@192.168.0.102
joao_admin@192.168.0.102's password:
Permission denied, please try again.
joao_admin@192.168.0.102's password:
```

Nota-se que a permissão foi negada.

Habilitando novamente o login do joao_admin:

sudo usermod -U joao_admin

```
lets@serverlets:~/page$ sudo usermod -U joao_admin
lets@serverlets:~/page$ sudo passwd -S joao_admin
joao_admin P 02/11/2024 0 99999 7 -1
```

A letra P indica que está habilitado, joao_admin tem uma senha definida e válida. Tentando fazer o login através do outro terminal, desta vez conseguiremos fazer com sucesso, pois está desbloqueado.

```
letonio.silva@BRRIOLN043879:~ $ ssh joao_admin@192.168.0.102
joao_admin@192.168.0.102's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)
```

Introdução ftp

A transferência de arquivos é uma atividade essencial no ambiente digital, e uma ferramenta amplamente utilizada para essa finalidade é o **FileZilla**, um cliente **FTP** (File Transfer Protocol) confiável e eficiente. Ao explorar as funcionalidades do FileZilla, mergulhamos em um universo de possibilidades para transferir dados de maneira segura e eficaz entre computadores conectados à internet. Este material visa fornecer uma visão abrangente sobre o que é o FTP. Posteriormente, vamos instalar o FileZilla e realizar transferências de arquivos, destacando sua importância na gestão eficiente de dados.

FTP

O **File Transfer Protocol** (FTP) ou Protocolo de Transferência de Arquivos, em português, é um protocolo padrão utilizado para transferir arquivos entre computadores em uma rede. Ele opera sobre uma arquitetura **cliente-servidor**, onde o cliente inicia a comunicação solicitando a transferência de arquivos e o servidor responde fornecendo acesso ou enviando os arquivos solicitados.

O FTP suporta dois modos de transferência: o **modo ativo** e o **modo passivo**. No modo ativo, o cliente abre uma porta para receber dados do servidor. No modo passivo, o servidor abre uma porta para o cliente se conectar. A porta padrão utilizada para transferências via FTP é a 21.

Sobre o FTP no Lado do Servidor (Server-side), o software é instalado e configurado para gerenciar as solicitações dos clientes. O servidor FTP responde a comandos, autentica usuários e gerencia o acesso aos diretórios. Ainda nesta aula, vamos instalar na máquina virtual o **Very Secure FTP Daemon** (vsftpd).

Sobre o FTP no lado do cliente, um programa FTP é usado para se conectar ao servidor FTP. Esses clientes podem ser aplicativos dedicados, como FileZilla. Na aula de hoje vamos instalar o FileZilla na nossa máquina local e transferir arquivos para a máquina virtual via FTP.

Instalação do vsftpd

A sua máquina virtual com o Ubuntu Server instalado será o seu Server-side. O objetivo é instalar o **Very Secure FTP Daemon** (vsftpd), que é um servidor FTP para sistemas Linux, conhecido por sua ênfase em segurança e eficiência. Ele é projetado para ser um servidor FTP leve, rápido e, como o nome sugere, muito seguro.

Ligue a sua máquina virtual. Recomendamos ligá-la via termianl com o headless mode, para que o software fique ligado em background.

VBoxManage startvm "UbuntuServer22.04-alpha" --type headless

Em seguida, vamos acessar a máquina virtual:

ssh lets@192.168.0.102

```
letonio.silva@BRRIOLN043879:~ $ ssh lets@192.168.0.102
lets@192.168.0.102's password:
```

Para instalar o vsftpd, basta aplicar esse comando:

sudo apt install vsftpd

```
lets@serverlets:~$ sudo apt install vsftpd
[sudo] password for lets:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
```

É interessante salvar uma cópia do arquivo de configuração do vsftpd com as configurações padrão. Antes de continuar, responda mentalmente a seguinte pergunta:

Qual diretório contém arquivos de configuração?

Para fazer a cópia, podemos usar esse comando:

sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.original

```
lets@serverlets:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.original
lets@serverlets:~$
```

Configurar o firewall

Caso o **Uncomplicated Firewall** (ufw) esteja ativo é necessário habilitar o tráfego em algumas portas. Para verificar o status do firewall, aplique o seguinte comando:

sudo systemctl status ufw

```
lets@serverlets:~$ sudo systemctl status ufw
● ufw.service - Uncomplicated firewall
     Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enabled)
     Active: active (exited) since Sun 2024-02-11 19:55:27 UTC; 27min ago
       Docs: man:ufw(8)
   Main PID: 565 (code=exited, status=0/SUCCESS)
        CPU: 640us

Feb 11 19:55:27 serverlets systemd[1]: Starting Uncomplicated firewall...
Feb 11 19:55:27 serverlets systemd[1]: Finished Uncomplicated firewall.
lets@serverlets:~$
```

Para o tráfego FTP, vamos usar as portas 20 e 21. O FTP recebeu oficialmente as portas 20 e 21. Se estiver usando especificamente uma configuração de **conexão "ativa"**, isso significa que enquanto um computador cliente faz a solicitação de conexão e envia os comandos primeiro na porta 21, conhecida como **"porta de controle"**, uma conexão para o servidor na porta 20, a "porta de dados", também é aberta automaticamente para transferir os dados do arquivo.

Se estiver usando uma configuração de conexão **FTP "passiva"**, o computador cliente também se conecta ao servidor na porta FTP 21. No entanto, o servidor responde com um número de porta aleatório, em um intervalo livre de portas, para usar como porta de dados para transferências de arquivos. Vamos definir o intervalo de portas 40000-50000 para conexão passiva.

O FTP original não é seguro, pois envia informações, incluindo senhas, em texto simples (plain text). Versões mais recentes, como FTPS (FTP Seguro) e SFTP (SSH File Transfer Protocol), foram desenvolvidas para adicionar criptografia e autenticação segura. Posteriormente, vamos aprender como usar o FTPS. O FTPS utiliza a porta 990.

A seguir, apresenta-se a lista de comandos para liberar o tráfego em todas essas portas:

sudo ufw allow 20/tcp
sudo ufw allow 21/tcp
sudo ufw allow 990/tcp
sudo ufw allow 40000:50000/tcp

```
lets@serverlets:~$ sudo ufw allow 20/tcp
[sudo] password for lets:
Rules updated
Rules updated (v6)
lets@serverlets:~$ sudo ufw allow 21/tcp
Rules updated
Rules updated (v6)
lets@serverlets:~$ sudo ufw allow 990/tcp
Rules updated
Rules updated (v6)
lets@serverlets:~$ sudo ufw allow 40000:50000/tcp
Rules updated
Rules updated (v6)
lets@serverlets:~$
```

Idealmente, para razões de segurança, é recomendável restringir o FTP a um diretório específico. O vsftpd utiliza a técnica de c**adeia chroot** para alcançar esse objetivo. Com o **Chroot** ativado, o usuário local é **confinado ao diretório inicial**, definido como padrão. No entanto, devido às considerações de segurança do vsftpd, pode ocorrer a impossibilidade de um usuário escrever no diretório inicial.

Em vez de retirar os privilégios de gravação do diretório inicial, optaremos por criar um diretório FTP separado que funcionará como um ambiente chroot, em conjunto com um diretório gravável de arquivos responsável por armazenar os dados relevantes. Utilize o comando a seguir para criar a pasta FTP:

mkdir /home/lets/ftp

```
lets@serverlets:~$ mkdir /home/lets/ftp
lets@serverlets:~$ ls -l
total 8
drwxrwxr-x 2 lets lets 4096 Feb 11 22:45 ftp
```

Defina a propriedade utilizando:

sudo chown nobody:nogroup /home/lets/ftp

```
lets@serverlets:~$ sudo chown nobody:nogroup /home/lets/ftp
[sudo] password for lets:
lets@serverlets:~$
```

Finalmente, remova as permissões para gravar:

sudo chmod a-w /home/lets/ftp

Agora, use o seguinte comando para verificar as permissões:

sudo ls -la /home/lets/ftp

Nota-se que a pasta ftp não pertence a ninguém e nenhum grupo. Além disso, foram removidas as permissões de escritas para todos os usuários (a-w). Ninguém pode gravar conteúdo lá.

```
lets@serverlets:~$ sudo ls -la /home/lets/ftp
total 8
dr-xr-xr-x 2 nobody nogroup 4096 Feb 11 22:45 .
drwxr-x--- 7 lets   lets    4096 Feb 11 22:45 ..
lets@serverlets:~$
```

Dentro do diretório ftp, vamos criar um subdiretório para onde os arquivos serão transferidos.

sudo mkdir /home/lets/ftp/arquivos

Vamos usar o comando chown para mudar o dono da pasta "arquivos" de root para "lets" (lets será o usuário usado na hora de fazer transferência da máquina local para a virtual).

sudo chown lets:lets /home/lets/ftp/arquivos

```
lets@serverlets:~$ ls -la /home/lets/ftp/arquivos/
total 8
drwxr-xr-x 2 root   root    4096 Feb 11 22:58 .
dr-xr-xr-x 3 nobody nogroup 4096 Feb 11 22:58 ..
lets@serverlets:~$ sudo chown lets:lets /home/lets/ftp/arquivos
lets@serverlets:~$ ls -la /home/lets/ftp/arquivos/
total 8
drwxr-xr-x 2 lets   lets    4096 Feb 11 22:58 .
dr-xr-xr-x 3 nobody nogroup 4096 Feb 11 22:58 ..
lets@serverlets:~$
```

Configurar o vsftpd

Usando um editor de texto da sua escolha, abra o arquivo /etc/vsftpd.conf. Por exemplo, isso pode ser feito usando nano, conforme mostrado a seguir:

sudo nano /etc/vsftpd.conf

Procure e remova o comentário (#) dessa linha: #write_enable=YES, ficando assim:

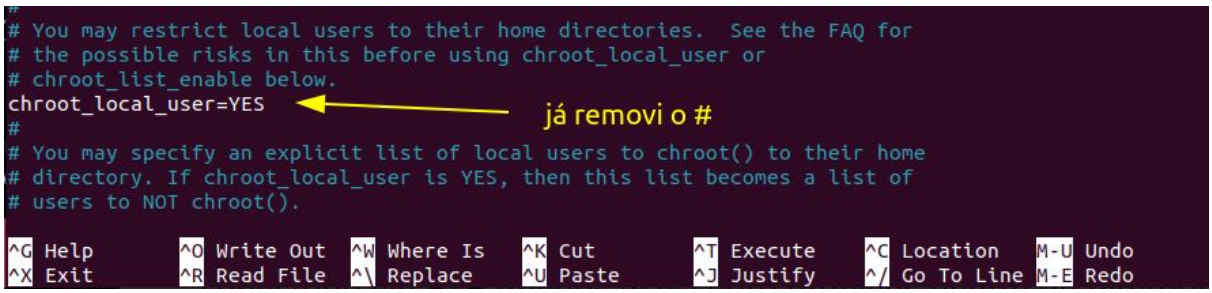
write_enable=YES

```
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES          <-- aqui já removi o #
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)

^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
```

Essa opção, quando habilitada, determina que os usuários têm permissão para realizar operações de gravação (escrever) no servidor FTP.

O **chroot_local_user=YES** também será descomentado para garantir que o usuário conectado via FTP apenas acessará arquivos dentro do diretório permitido, impedindo-o de acessar diretórios acima desse ponto na hierarquia de diretórios.



Vá ao final do arquivo de configuração para adicionar algumas novas linhas. Primeiramente, será incluído um **user_sub_token** no caminho do diretório local_root. Isso garantirá que a configuração funcione tanto para o usuário atual quanto para quaisquer outros usuários que sejam adicionados posteriormente.

user_sub_token=$USER
local_root=/home/$USER/ftp

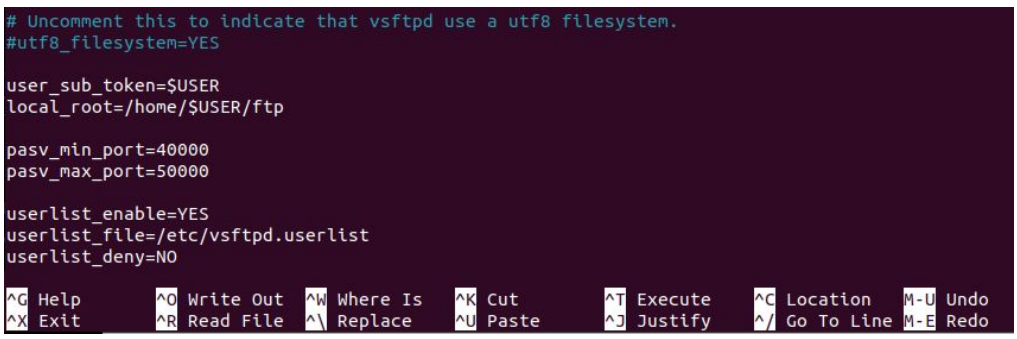
Com o objetivo de assegurar a disponibilidade de um número significativo de conexões, iremos restringir o número de portas utilizadas no arquivo de configuração.

pasv_min_port=40000
pasv_max_port=50000

Neste guia, nossa abordagem será permitir o acesso de forma seletiva; portanto, configuraremos para garantir que o acesso seja concedido apenas aos usuários que tenham sido explicitamente adicionados a uma lista.

userlist_enable=YES
userlist_file=/etc/vsftpd.userlist
userlist_deny=NO

A variável booleana **userlist_deny** é responsável por alternar a lógica; quando configurado para "não", somente os usuários especificados na lista terão acesso permitido.
A Figura abaixo mostra as configurações que foram adicionadas no final do arquivo:



Uma vez feito isso, clique em CTRL + X e confirme as alterações no arquivo.

Por fim, vamos adicionar o usuário "lets" (lembre-se de substituir pelo nome do seu usuário) ao final da lista de usuários que podem criar e adicionar arquivos usando o servidor FTP, por meio desse comando:

echo "lets" | sudo tee -a /etc/vsftpd.userlist

```
lets@serverlets:~$ echo "lets" | sudo tee -a /etc/vsftpd.userlist
[sudo] password for lets:
lets
lets@serverlets:~$ cat /etc/vsftpd.userlist
lets
lets@serverlets:~$
```

Vamos reiniciar o servidor vsftpd para que as novas configurações sejam aplicadas:

sudo systemctl restart vsftpd

```
lets@serverlets:~$ sudo systemctl restart vsftpd
lets@serverlets:~$
```

Instalação do Filezilla

Se você quiser fazer upload de um arquivo maior ou transferir arquivos com maior praticidade, você precisa usar um cliente FTP como o **FileZilla**. Este software livre de código aberto ajuda a transferir arquivos da sua máquina local para servidores web e vice-versa.

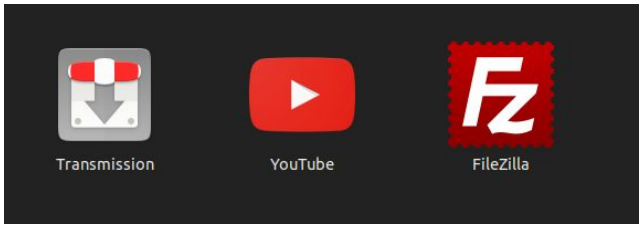
Sistemas operacionais Windows e MacOS: Caso o sistema operacional da sua máquina local seja Windows ou MacOS faça o download do instalador Filezilla Client, através do link:

https://filezilla-project.org/

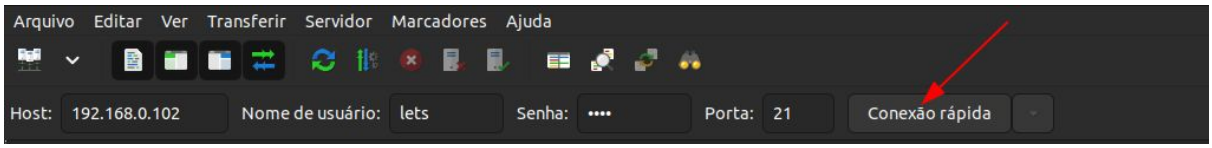
Sistema Linux: Se você Caso o sistema operacional da sua máquina local seja Linux, você pode instalar o Filezilla via:

sudo apt install filezilla

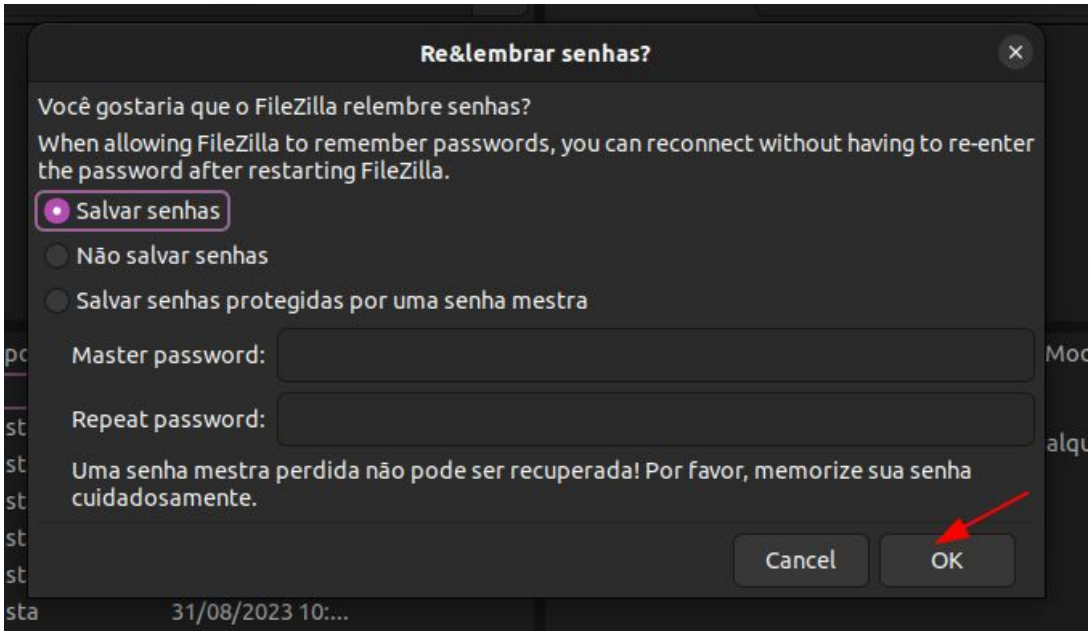
Após instalar o Filezilla client, abra o programa.



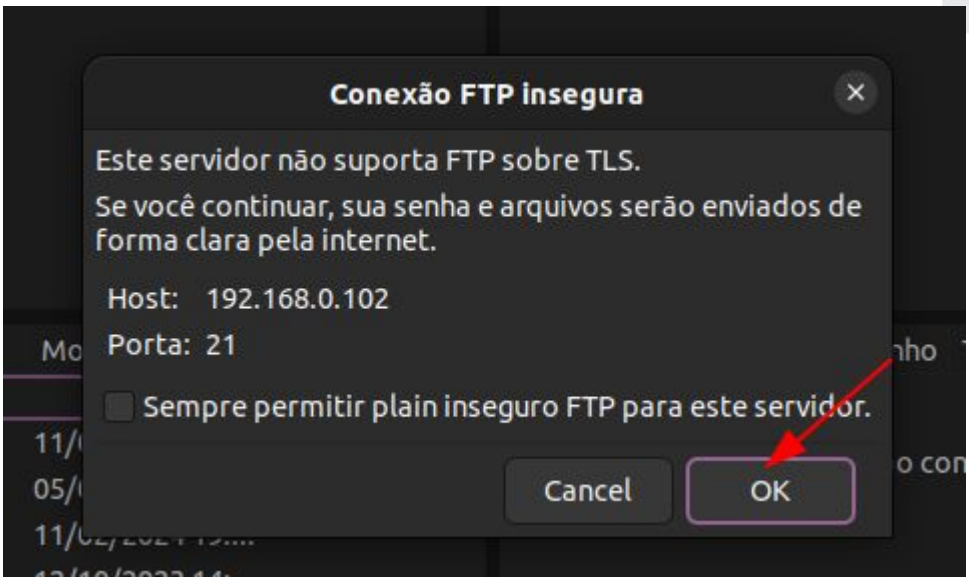
No Filezilla, preencha os campos com os dados do servidor, inserindo também as credenciais do usuário que pode usar o FTP. Por fim, clique em conexão rápida:



É possível salvar as senhas para não precisar inserir novamente os mesmos dados. Tome cuidado caso opte por criar uma senha mestra.



Conforme explicado mais cedo, o FTP puro não tem criptografia. Uma mensagem aparece, ativando o usuário sobre isso. Mesmo assim, clique em OK.

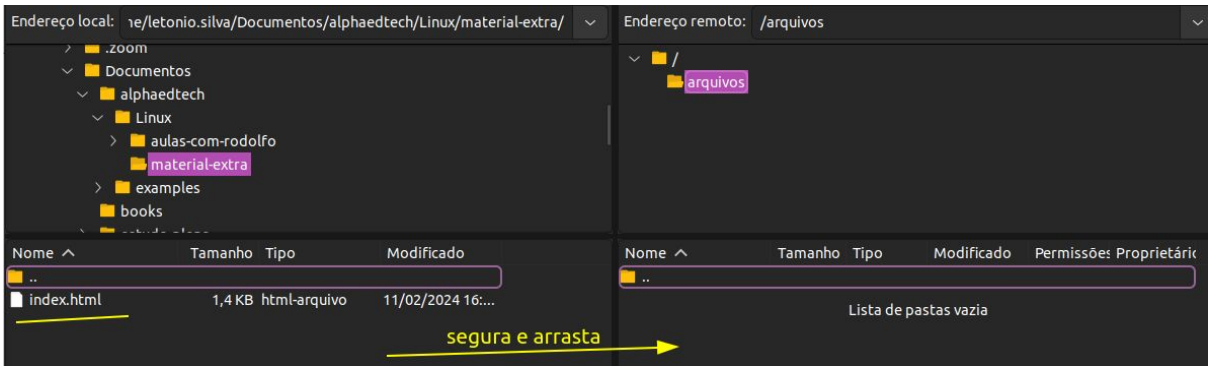


A conexão foi um sucesso, conforme ilustrado na imagem a seguir:

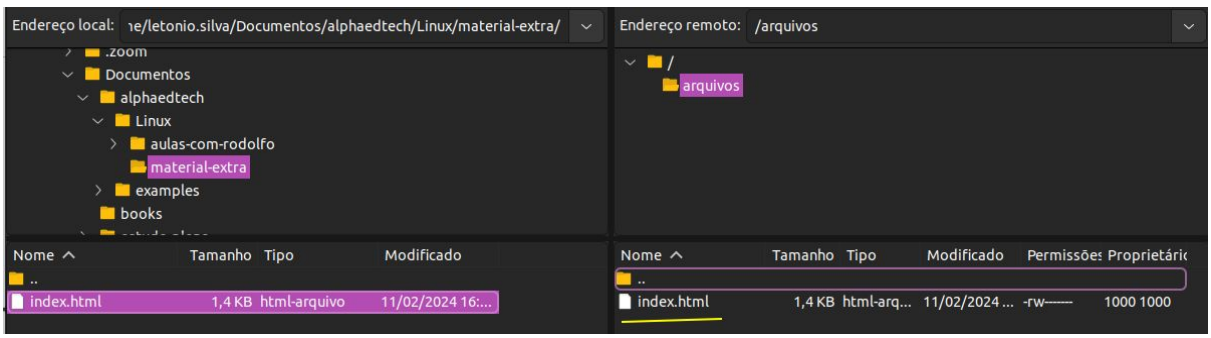
À esquerda, temos o sistema de pastas e arquivos da máquina local. À direita, temos a "raiz" do diretório que podemos acessar via FTP. O caminho absoluto é /home/lets/ftp/arquivos. Porém, nos passos iniciais, definimos uma pasta ftp como raiz (**chroot**) e o usuário lets, ao fazer conexão ftp não teria acesso a nenhum arquivo/diretório acima dessa hierarquia. Portanto, estamos limitados a essa pasta e somente podemos fazer transferências para esse diretório.

Transferindo arquivos

Nosso objetivo é transferir um arquivo **index.html** simples (html e css). Posteriormente, vamos pegar esse arquivo transferido via FTP e substituir o arquivo padrão que o Apache está servindo.
Localize na sua máquina local o arquivo index.html que deseje transferir. Em seguida, arraste o arquivo para o diretório da máquina virtual (arquivos), conforme ilustrado a seguir:



Na imagem abaixo, mostra-se que a transferência foi bem-sucedida.



Substituição da página web

Com a transferência sendo bem-sucedida, vamos copiar o arquivo recém-transferido para a pasta que o apache usa para servir como padrão.
Apenas relembrando:
O caminho absoluto até o arquivo recém-transferido é **/home/lets/ftp/arquivos/index.html**
O caminho absoluto até o arquivo servido pelo apache é **/var/www/html/index.html**

Vamos fazer a cópia, o que substituirá o arquivo atual do apache:

sudo cp /home/lets/ftp/arquivos/index.html /var/www/html/index.html

Acesse **http://endereco_ip** para visualizar a nova página html.