

# Linux

Comandos, dicas e outras cositas más que uso de vez em quando

- [Copiar e mesclar duas pastas no terminal](#)
- [dig dig dig dig](#)
- [Erro ao atualizar certificados no let's encrypt](#)
- [Comprimindo pastas](#)
- [Alterando dependências de um pacote deb](#)
- [Encontrando e substituindo em arquivos no shell](#)
- [Removendo um ip do Fail2ban](#)
- [Mini-curso mega-rápido de Torrents e Piratebay](#)
- [Operações com arquivos no shell](#)
- [Exibir qual processo está usando uma determinada porta](#)
- [Agendando desligamento no Debian](#)
- [Vídeos de Timelapse](#)
- [Criando e distribuindo um torrent usando o Transmission](#)
- [Convertendo de MP4 para GIF no Debian](#)
- [Montando drives usb na linha de comando](#)
- [Executando um comando repetidas vezes no terminal](#)
- [Acesso ssh via tor](#)
- [Removendo arquivos com interrogação \(?\) no nome](#)
- [Trabalhando com arquivos no terminal](#)
- [Virtualbox - faltando headers](#)
- [Mexendo com arquivos wbfs para usar no Nintendo Wii](#)
- [Gravando um arquivo ISO em um pendrive](#)
- [Erro NO\\_PUBKEY no Ubuntu/Debian](#)
- [Backup com rsync](#)
- [Listando portas X serviços](#)
- [Um pouco de windows](#)
- [Baixando arquivos recursivamente via FTP](#)

- [nohup](#)
- [Renomeando arquivos sequencialmente](#)
- [Mexendo com o Swap](#)
- [DDNS com a Digital Ocean](#)
- [Atualizando o Debian/Ubuntu](#)
- [Criando link simbólico para uma pasta](#)
- [Trocando o editor do crontab para o nano](#)
- [Configurando o Percona Monitoring and Management usando Docker](#)
- [Habilitando APM no log do Nginx](#)
- [Gerando chaves ssh](#)
- [Docker](#)
- [Usando ffmpeg](#)
- [Bloqueando ip no iptables](#)
- [Comprimir imagens no terminal](#)

# Copiar e mesclar duas pastas no terminal

Usando rsync:

```
rsync -a /origem/ /destino/
```

No caso acima, vai copiar o conteúdo de origem para destino.

O detalhe é que o destino precisa terminar com /, senão a pasta vai ser copiada dentro da outra, e não mesclada.

Fonte: <https://unix.stackexchange.com/questions/149965/how-to-copy-merge-two-directories>

# dig dig dig dig

..ou brincando com dns:

Verificando se uma entrada CNAME existe:

```
dig subdominio.dominio.com cname
```

Forçando a verificação a passar por um servidor DNS (pra ver se já propagou por ele):

```
dig @1.0.0.1 subdominio.dominio.com cname
```

(no caso, 1.0.0.1 é o ip do servidor DNS)

Alterando os servidores DNS do servidor:

```
nano /etc/resolv.conf
```

alterar (ou apenas comentar) as linhas que possuem "nameserver", apontando para o ip desejado

# Erro ao atualizar certificados no let's encrypt

Isso no servidor Debian.

Se aparecer a mensagem:

```
("bad handshake: Error([('SSL routines', 'tls_process_server_certificate', 'certificate verify failed')],),). Skipping
```

Primeiro, precisa tentar instalar os certificados raiz (ou algo assim):

```
apt install ca-certificate
```

Se aparecer a mensagem "ca-certificates is already the newest version", precisa instalar manualmente:

1) Baixar o arquivo de instalação dos certificados:

[https://security.debian.org/debian-security/pool/main/c/ca-certificates/ca-certificates\\_20200601~deb9u2\\_all.deb](https://security.debian.org/debian-security/pool/main/c/ca-certificates/ca-certificates_20200601~deb9u2_all.deb)

(novo link: [https://archive.debian.org/debian-security/pool/updates/main/c/ca-certificates/ca-certificates\\_20200601~deb9u2\\_all.deb](https://archive.debian.org/debian-security/pool/updates/main/c/ca-certificates/ca-certificates_20200601~deb9u2_all.deb))

2) instalar os certificados:

```
dpkg -i ca-certificates_20200601~deb9u2_all.deb
```

3) Pedir para o let's encrypt atualizar os certificados dos sites:

```
certbot renew
```

Se não aparecer a mensagem de que os certificados já estão na última versão, basta pedir para o let's encrypt atualizar os certificados. Eu acho.

# Comprimindo pastas

```
tar -zcvf archive-name.tar.gz directory-name
```

extrair:

```
tar -zxvf archive-name.tar.gz
```

Se o formato utilizado for **xz**, precisa trocar o **z** por **J**:

```
tar -Jxvf arquivo.tar.xz
```

# Alterando dependências de um pacote deb

1. Descompactar deb: `ar x nomedoarquivo.deb` (serão criados três arquivos: debian-binary control.tar.gz data.tar.gz)
2. Descompactar arquivo control: `tar xzf control.tar.gz` (vai criar alguns arquivos como: postinst postrm preinst prerm md5sums control)
3. Corrigir dependências no arquivo control (exemplo, alterando de **Depends** para **Recommends**)
4. Recompactar control.tar.gz: `tar c postinst postrm md5sums control | gzip -c > control.tar.gz` (verificar todos os arquivos gerados no passo 2)
5. Recompactar arquivo deb: `ar rcs novopacote.deb debian-binary control.tar.gz data.tar.gz` (a ordem é importante!)

Fonte: <https://coderwall.com/p/hes3ha/change-the-dependencies-of-a-deb-package>

# Encontrando e substituindo em arquivos no shell

Procurar um determinado texto em vários arquivos:

```
find . -type f -exec grep -l 'gerar_boletos' {} \;
```

Procurar um determinado texto em um arquivo e salvar o resultado em outro arquivo:

```
grep '177.141.123.210' access_log20140604 > encontrou.log
```

Para apenas listar os nomes dos arquivos, informar o parâmetro -l para o grep:

```
grep -l procurar *
```

Procurando e substituindo um texto em vários arquivos:

```
find . -type f -name "*.php" -exec sed -i 's/procurar/substituir/g' {} +
```

Não esquecer de escapar as barras!



# Removendo um ip do Fail2ban

1) Listar os ips bloqueados:

```
iptables -L -n
```

2) Listar os nomes das regras do fail2ban:

```
fail2ban-client status | grep "Jail list" | sed -E 's/^[^:]+:[ \t]+//' | sed 's/,//g'
```

3) Desbloquear o ip, informando também o nome da regra:

```
fail2ban-client set REGRA unbanip ENDERECO
```

exemplo:

```
fail2ban-client set sshd unbanip 186.204.26.11
```

# Mini-curso mega-rápido de Torrents e Piratebay

Aqui vai um tutorial resumido de como baixar filmes no piratebay.

Primeiro, precisa de um client de bit torrent. Eu sempre uso o Transmission, porque é bem simples:

<https://transmissionbt.com/download/>

Depois de instalar, abra o navegador e acesse o <https://thepiratebay.org/>. Não esqueça de ativar bloqueador de anúncios, pra não ser bombardeado com anúncios chatos.

Pesquise pelo conteúdo que quiser baixar. Na tela do resultado da busca, preste atenção na coluna "**SE**" - essa coluna indica a quantidade de Seeds, ou seja, pessoas que tem o arquivo completo e estão distribuindo. Se não tiver nenhum, a chance de conseguir baixar o arquivo completo é menor. A coluna "**LE**" indica a quantidade de Leeches, que são as pessoas que estão transferindo o arquivo mas não possuem ele completo. Quanto mais Seeds e mais Leeches, mais rápido vai ser o download.

Também é bom prestar atenção no tamanho do arquivo. Para filmes, o tamanho ideal fica entre 800Mb e 4 ou 5 Gb. Menos do que isso pode ser algo com a qualidade muito baixa, e mais do que isso pode ser qualidade muito alta - não que seja um problema, mas nem todos os tocadores conseguem reproduzir vídeos com definição alta demais, sem contar que o tempo de download vai ficar muito maior.

E fuja de torrents que tem as palavras **TS** ou **CAM** no título - estes são aqueles filmes em que alguém grava dentro do cinema, e geralmente a qualidade é terrível.

Após decidir qual torrent vai ser baixado, ainda na tela de resultado da busca, clique com o botão direito no ícone de ímã (🧲) e copie o link magnético - é um link que começa com **magnet:?**.

No programa **Transmission**, clique em **File - Open URL** e cole o link magnético, e é isso. Lembrando que a velocidade vai depender de quantos Seeds e quantos Leeches estão distribuindo o torrent.

Alguns tocadores de vídeo baixam automaticamente as legendas. Caso o seu tocador não faça isso, ou não encontre a legenda, eu costumo copiar o nome completo do arquivo de vídeo e pesquiso no google, adicionando a palavra "legendas".

# Operações com arquivos no shell

Excluindo arquivos com mais de 90 dias:

```
find pasta -mtime +90 -type f -exec rm -f {} \;
```

(altere o +90 para -90 para pegar arquivos com menos de 90 dias)

Excluindo arquivos com uma determinada extensão:

```
find pasta -type f -name *.sql -exec rm -f {} \;
```

Excluindo pastas vazias:

```
find pasta -type d -empty -exec rm -Rf {} \;
```

Excluindo arquivos vazios:

```
find pasta -type f -empty -exec rm -f {} \;
```

Para listar os arquivos ao invés de excluir, troque o `rm -f` por `ls -l`.

Extraindo arquivos existentes em sub-pastas:

```
for file in `find *`; do  
gunzip "${file}" ; done
```

Mesclando vários arquivos em um só:

```
find . -type f -name 'access-*.log' -exec cat {} + >> output.file
```

Gerando log com o goaccess:

```
goaccess output.log -a -o log.html
```

Substituindo texto em um arquivo

```
sed -i 's/procurar/substituir/g' arquivo.txt
```

Dividindo um arquivo em partes:

Dividir o arquivo lista.txt em partes de 1000 linhas cada.  
Cada parte do arquivo será chamada parte1.txt, parte2.txt etc.

```
split -l 1000 lista.txt parte
```

Extrair um arquivo tgz:

```
tar -zxvf arquivo.tgz
```

Extrair um arquivo específico que está dentro do tgz:

```
tar -zxvf arquivo.tgz arquivo_interno.txt
```

Comprimindo uma pasta/arquivo no formato tgz:

```
tar -cvpzf arquivo.tgz pasta
```

Caso o arquivo seja .tar, basta tirar o z do comando:

```
tar -xvf arquivo.tar
```

# Exibir qual processo está usando uma determinada porta

Assim:

```
lsof -w -n -i tcp:9000
```

# Agendando desligamento no Debian

Editar o arquivo /etc/rc.local:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# shutdown everyday at 6 am
shutdown -h 06:30 &

exit 0
```

# Vídeos de Timelapse

Itens necessários:

- Celular para capturar as imagens;
- Algum programa para tirar fotos de minuto a minuto (<http://survey7.com/cameraupload/>)
- FFmpeg;

Girando as imagens (caso elas tenham sido capturadas com o celular na horizontal):

```
for file in *.jpg; do convert "$file" -rotate -90 "${file%.jpg}"_rotated.jpg; done
```

As imagens convertidas terão o nome "\_rotated.jpg".

Agora, precisamos criar o vídeo:

```
ffmpeg -r 5 -pattern_type glob -i '*_rotated.jpg' -vcodec libx264 timelapse_5fps.mp4
```

Altere o parâmetro `-r 5` para definir a quantidade de quadros por segundo. Quando for um vídeo longo, é melhor usar uma taxa maior, tipo 15.

# Criando e distribuindo um torrent usando o Transmission

Usando o terminal, é óbvio.

Primeiro, instale o transmission. Junto com ele, vai vir o daemon e alguns programas. Os que precisamos são o transmission-create e o transmission-remote.

Depois, precisamos de uma lista de trackers. Neste link tem alguns:

<https://github.com/ngosang/trackerslist>

Navegue até um nível acima da pasta/arquivo que será compartilhada, e execute este comando, para criar o arquivo .torrent:

```
transmission-create pasta_a_ser_compartilhada/ -o arquivo.torrent -t http://umtracker:1337/announce -t  
udp://outrotracker:1337/announce
```

Informe quantos trackers puder, para facilitar a distribuição. Neste caso tem só dois, mas no teste que fiz, coloquei 57.

Agora é preciso começar a semear o torrent, para que as outras pessoas possam baixar.

```
transmission-remote --add arquivo.torrent --download-dir  
/caminho_completo_ate_a_pasta_compartilhada/ -n  
usuario_do_transmission:senha_do_transmission
```

Aqui tem um detalhe: É preciso informar o caminho para chegar na pasta/arquivo compartilhado na opção `--download-dir`. Com isso, o Transmission vai perceber que tu já tem todos os arquivos do torrent, e vai começar a semear. Isso é importante caso tu esteja semeando um arquivo/pasta que não esteja na pasta padrão de downloads do Transmission.

Por exemplo, se tu vai compartilhar a pasta `/home/usuario/Downloads/blariblabla`, os comandos ficam assim:

```
cd ~/Downloads
```



```
transmission-create blariblabla/ -o arquivo.torrent -t http://umtracker:1337/announce -t  
udp://outrotracker:1337/announce
```

```
transmission-remote --add arquivo.torrent --download-dir  
/home/usuario/Downloads/ -n  
usuario_do_transmission:senha_do_transmission
```

Depois disso, é só distribuir o arquivo .torrent - seja através do Piratebay, ou de qualquer outro modo.

Um jeito bom de testar se o torrent vai funcionar é baixando ele através do <https://seedr.cc/>.

# Convertendo de MP4 para GIF no Debian

Com o ffmpeg:

```
ffmpeg -i video.mp4 imagem.gif
```

Provavelmente também funciona com outros formatos de vídeo.

# Montando drives usb na linha de comando

Listando os dispositivos e partições encontrados:

```
sudo fdisk -l
```

Montando a partição do dispositivo em uma pasta:

```
sudo mount /dev/sdb1 /media/usb/
```

Obs: A pasta /media/usb precisa ter sido criada antes.

Listando o espaço em disco dos dispositivos:

```
df
```

Desmontando a partição para remover o drive:

```
sudo umount /media/usb
```

# Executando um comando repetidas vezes no terminal

Assim:

```
while sleep TEMPO; do COMANDO; done
```

Exemplo: exibir a data, a cada segundo:

```
while sleep 1; do date; done
```

# Acesso ssh via tor

Primeiro, instalamos tor e o connect-proxy:

```
sudo apt install tor connect-proxy
```

Depois, configuramos o arquivo `~/.ssh/config` para usar o proxy:

```
Host *
```

```
CheckHostIP no
```

```
Compression yes
```

```
Protocol 2
```

```
ProxyCommand connect -4 -S localhost:9050 $(tor-resolve %h localhost:9050) %p
```

Por fim, conectamos no ssh:

```
ssh user@server
```

Se der a mensagem de erro `ERROR: You must specify the destination port correctly.`, então precisa usar o ip do servidor, ao invés do hostname.

# Removendo arquivos com interrogação (?) no nome

Às vezes aparecem uns arquivos chamados `?.pub`, ou apenas `?` na pasta home.

Na verdade, esse não é o nome dos arquivos. O que acontece é que o nome deles é formado por caracteres unicode que não podem ser exibidos no console.

O jeito é descobrir o inode dos arquivos, e excluir através dele.

Listando o inode dos arquivos:

```
ls -li
```

Excluindo através do inode:

```
find . -inum 15681478197 -exec rm -i {} \;
```

Fonte: <https://unix.stackexchange.com/a/230030>

# Trabalhando com arquivos no terminal

Listando:

```
ls -l
```

Ordenando pela data:

```
ls -l --sort=time
```

Ordenando inversamente:

```
ls -l --sort=time -r
```

Exibindo data e hora completos do arquivo:

```
ls -l --time-style=full-iso
```

Contando quantos arquivos tem na pasta:

```
ls | wc -l
```

Alterando data de modificação de arquivos:

```
touch -d "2 hours ago" filename
```

Para remover arquivos com uma determinada extensão em várias sub-pastas, recursivamente:

```
find pasta -name '*.extensao' -exec rm -f {} \;
```

Obviamente, pode ser usado na pasta atual:

```
find . -name '*.extensao' -exec rm -f {} \;
```

Encontrando uma linha específica em um arquivo:

```
sed -n '12345p' < arquivo.txt
```

Procurar arquivos que contenham um texto:

```
find . -exec grep -l '_0xaae8' {} \;
```

(tenho a impressão de que já tenho um comando parecido com esse em algum lugar, mas não tenho certeza)

Fontes: <https://unix.stackexchange.com/questions/1125/how-can-i-get-a-count-of-files-in-a-directory-using-the-command-line>

<https://askubuntu.com/questions/62492/how-can-i-change-the-date-modified-created-of-a-file>



# Virtualbox - faltando headers

Ao iniciar o Virtualbox, aparece uma mensagem dizendo para executar o `/sbin/vboxconfig` como root.

Ao executar o comando indicado, aparece a seguinte mensagem de erro:

```
This system is currently not set up to build kernel modules.  
Please install the Linux kernel "header" files matching the current kernel  
for adding new hardware support to the system.  
The distribution packages containing the headers are probably:  
linux-headers-amd64 linux-headers-4.9.0-6-amd64  
This system is currently not set up to build kernel modules.  
Please install the Linux kernel "header" files matching the current kernel  
for adding new hardware support to the system.  
The distribution packages containing the headers are probably:  
linux-headers-amd64 linux-headers-4.9.0-6-amd64
```

```
There were problems setting up VirtualBox. To re-start the set-up process, run  
/sbin/vboxconfig  
as root.
```

Para resolver, basta instalar os headers indicados na mensagem de erro.

```
sudo apt install linux-headers-amd64 linux-headers-4.9.0-6-amd64
```

E depois, rodar de novo o comando para configurar o Virtualbox:

```
sudo /sbin/vboxconfig
```

O detalhe aqui é verificar o header descrito na mensagem de erro.

# Mexendo com arquivos wbfs para usar no Nintendo Wii

Extraindo de rar para iso:

```
unrar x arquivo.part01.rar
```

Convertendo de ISO, WDF, WIA, CISO, WBFS, GCZ out FST para wbfs:

```
wit copy --source game.iso --wbfs -d game.wbfs --split -v
```

Encontrando o ID correto no arquivo wbfs:

```
wwt LIST-L game.wbfs
```

Fonte: <https://negativo17.org/wit-wiimms-iso-tools/>

<https://wit.wiimm.de/wit/>

# Gravando um arquivo ISO em um pendrive

Usando o comando dd:

```
sudo dd if=arquivo.iso of=/dev/sdc status=progress
```

Só que isso nem sempre faz com que o pendrive seja "bootável". Por isso, o melhor jeito é instalar o [Ventoy](#) em um pendrive, depois disso, basta copiar os arquivos iso nele que o programa faz o resto. Faz uns dois anos que não preciso mais ficar criando pendrives bootáveis.

# Erro NO\_PUBKEY no Ubuntu/Debian

Mensagem de erro:

```
The following signatures couldn't be verified because the public key is not available: NO_PUBKEY  
978228591BD3A65C
```

Comando:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 978228591BD3A65C
```

Fonte: <https://www.vivaolinux.com.br/topico/Linux-Basico/erro-com-chave>

# Backup com rsync

```
rsync -azP --bwlimit=5000 --exclude=git user@host:/origem destino/
```

Quando esquecer para que servem os parâmetros, uma dica é usar o site Explain Shell:

<https://explainshell.com/explain?cmd=rsync+-azP+--bwlimit%3D5000+--exclude%3Dgit+user%40host%3A%2Forigem+destino%2F>

# Listando portas X serviços

```
sudo lsof -iTCP -sTCP:LISTEN -P
```

# Um pouco de windows

Alguns comandos úteis (ou não) para o windows 10.

Girar a tela:

Ctrl+Alt+Seta para os lados

Usar vários desktops:

Win + Tab

P.S.: eu sei que essa seção se chama "Linux", mas não vou criar uma seção pra postar dois comandos, né

# Baixando arquivos recursivamente via FTP

Porque nem sempre dá pra usar o sftp.

```
wget -v -r -t 1 --include-directories=public_html,public,httpdocs,httpsdocs,Web --user=usuario --password=senha ftp://dominio.com/
```

no parâmetro include-directories, passar as pastas que serão baixadas. Se for pra pegar tudo, não precisa passar nada.



# nohup

Para executar um comando e sair sem esperar o retorno.

Pode parecer falta de educação deixar o comando falando sozinho, mas às vezes é necessário.

```
nohup comando > /dev/null 2>&1 &
```

# Renomeando arquivos sequencialmente

Pega vários arquivos chamados "parte1", "parte2" etc. e renomeia para 0001, 0002 e assim por diante.

```

a=1
for i in parte*; do
new=$(printf "%d" "$a") #04 pad to length of 4
mv -- "$i" "$new"
let a=a+1
done
```

# Mexendo com o Swap

```
sudo swapon --show //consultando a utilização do swap
```

```
sudo swapoff -a //desligando swap
```

```
sudo swapon --show //confirmando se desligou
```

```
sudo fallocate -l 4G /swapfile //criando um arquivo de 4Gb para usar como swap
```

```
ls -lh /swapfile //conferindo o tamanho do arquivo
```

```
sudo chmod 600 /swapfile //dando permissão de acesso apenas ao root
```

```
sudo mkswap /swapfile //transformando o arquivo em swap
```

```
sudo swapon /swapfile //habilitando o swap
```

```
sudo swapon --show //conferindo se habilitou
```

```
cat /proc/sys/vm/swappiness //conferindo o swappiness
```

```
sudo sysctl vm.swappiness=30 //alterando o valor do swappiness para 30
```

```
gksudo gedit /etc/sysctl.conf //alterando permanentemente o valor de swappiness
```

```
vm.swappiness=10 //se nao existir esta linha, criar
```

```
echo 'swapfile none swap sw 0 0' | sudo tee -a /etc/fstab //mantendo a alteração permanente
```

Obs: não esquecer de editar o /etc/fstab, comentando a linha da partição de swap, caso já exista uma antes.

# DDNS com a Digital Ocean

Como o DynDNS agora é pago, o jeito é usar alguma outra solução de dns dinâmico, de preferência gratuita.

No caso não é exatamente gratuita - afinal, a Digital Ocean cobra pelo servidor - mas enfim.

Primeiro, siga os passos da instalação:

[https://github.com/digitalm3/DigitalOcean\\_dyndns](https://github.com/digitalm3/DigitalOcean_dyndns)

Depois de seguir os passos, precisa criar um subdomínio no painel da DO.

Não esqueça que este subdomínio precisa ter uma entrada do tipo A apontado para o IP dinâmico, e não para o IP do servidor.

Para atualizar esse IP, instale o ddclient.

Configure o arquivo `/etc/ddclient.conf` da seguinte forma:

```
# /etc/ddclient.conf
script=/nic/update.php
protocol=dyndns2
#use=if, if=enp2s0
use=web, web=checkip.dyndns.org/
server=dyndns.meudominio.com
login=qualquercoisa
password='qualquercoisa'
meusubdominio.meudominio.com
```

na linha **server=dyndns.meudominio.com**, informe a url de onde ficou instalado o script do primeiro passo. No meu caso, eu criei um subdomínio chamado dyndns, mas poderia ter instalado no domínio principal.

O login e senha podem ser qualquer coisa, já que o script não faz nenhuma autenticação (o que pode ser perigoso, caso vá rodar algum serviço sigiloso nesse ddns).

Já a última linha é o subdomínio que foi criado no segundo passo.

Depois, é só rodar um **sudo ddclient update** que o IP vai ser atualizado.

Ah, a linha **use=web, web=checkip.dyndns.org/** é importante para que o ddclient pegue o IP externo da tua rede. Caso contrário, ele pode pegar o IP interno (192.168.0.6, por exemplo).

# Atualizando o Debian/Ubuntu

Comando pra atualizar o Ubuntu ou Debian.

Eu sempre esqueço...

```
sudo apt update -y && sudo apt upgrade -y && sudo apt dist-upgrade -y && sudo apt-get autoremove -y  
sudo reboot
```

# Criando link simbólico para uma pasta

Assim:

```
ln -s /var/log/apache2/ ~/logs
```

O detalhe é que a pasta ~/logs não deve existir na hora em que for rodar o comando.

# Trocando o editor do crontab para o nano

Só vale para a sessão atual do ssh, mas já ajuda:

```
env EDITOR=nano crontab -e
```

# Configurando o Percona Monitoring and Management usando Docker

As instruções são um pouco confusas, então achei melhor criar um guia.

Requisitos:

- 1) Uma máquina rodando Docker com alguma porta exposta (no exemplo usei a **4443**)
- 2) Uma ou mais máquina rodando MariaDB ou Mysql, com acesso root

## 1) Instalar o Servidor PMM

Baixar a imagem do docker:

```
docker pull percona/pmm-server:2
```

Criar o volume para armazenar os dados:

```
docker volume create pmm-data
```

Iniciar o container:

```
docker run --detach --restart always --publish 4443:443 --volume pmm-data:/srv --name pmm-server percona/pmm-server:2
```

## 2) Configurar o servidor

Acesse a administração e crie uma senha. Os dados do primeiro acesso são admin / admin:  
<https://ip-do-servidor:8443>

## 3) Instalar o cliente

Em cada máquina que será monitorada, instale o pmm-client. Por exemplo, no Debian fica assim:

```
wget https://repo.percona.com/apt/percona-release_latest.generic_all.deb
```

```
dpkg -i percona-release_latest.generic_all.deb
```

```
sudo apt update
```

```
sudo apt install -y pmm2-client
```



Nesse link tem mais instruções para outras plataformas: <https://docs.percona.com/percona-monitoring-and-management/setting-up/client/index.html#install>

## 4) Configurar os clientes

Crie um usuário no banco com algumas permissões básicas. O exemplo abaixo é para o MariaDB:

```
CREATE USER 'pmm'@'127.0.0.1' IDENTIFIED BY 'pass';  
UPDATE mysql.user set MAX_USER_CONNECTIONS = 10 WHERE user = 'pmm';  
GRANT SELECT, PROCESS, REPLICATION CLIENT, RELOAD ON *.* TO 'pmm'@'127.0.0.1';  
FLUSH PRIVILEGES;
```

Aqui tem mais exemplos: <https://docs.percona.com/percona-monitoring-and-management/setting-up/client/mysql.html#create-a-database-account-for-pmm>

Conecte o agente com o servidor:

```
pmm-admin config --server-insecure-tls --server-url=https://admin@localhost:senha@ip-do-servidor:4443 --force
```

**admin@localhost** é o usuário que você criou no painel, **senha** é a senha desse usuário.

Conecte o agente com o banco de dados:

```
pmm-admin add mysql --username=pmm --password=senha --server-insecure-tls --server-url=https://admin:senha@ip-do-servidor:4443 --query-source=perfschema hostname localhost:3306
```

**--username** e **--password** são as credenciais de acesso ao banco. As outras credenciais são as mesmas informadas acima (pois é). **perschema** indica que o cliente vai usar a tabela **performance\_schema** para as estatísticas (também pode ser pelo arquivo de slow query, mas não achei exemplo de como funciona). **hostname** é o nome que vai ser exibido no servidor, e é bom que seja diferente para cada instância monitorada.

Fontes:

<https://github.com/percona/pmm>

<https://docs.percona.com/percona-monitoring-and-management/index.html>

# Habilitando APM no log do Nginx

APM é um modo de medir a performance de um sistema.

Por padrão, o log do nginx mostra apenas a hora em que a requisição foi servida, mas não mostra o tempo que ela levou para ser processada.

Essa configuração serve para que o nginx exiba esse tempo, para que seja possível identificar as requisições mais demoradas e que precisam de otimização.

O primeiro passo é encontrar o arquivo nginx.conf (geralmente em /etc/nginx/nginx.conf) e adicionar o seguinte template de log:

```
log_format apm '$remote_addr - $remote_user [$time_local] '
    '$request' $status $body_bytes_sent '
    '$http_referer' '$http_user_agent' ($request_time);
```

A variável [\\$request\\_time](#) possui o tempo que levou para processar a requisição. Coloquei entre parênteses apenas para ficar mais fácil de identificar.

O próximo passo é adicionar a configuração no arquivo vhost do site, para que o log seja gerado:

```
access_log /var/log/nginx/site.com.br/apm.log apm;
```

# Gerando chaves ssh

Eu sempre esqueço.

```
ssh-keygen -t rsa
```

Ler o conteúdo do arquivo:

```
cat ~/.ssh/id_rsa.pub
```

Depois precisa adicionar no arquivo `~/.ssh/authorized_keys` da máquina remota.

Caso apareçam aqueles arquivos chamados **?.pub**, ou apenas **?** na pasta home.

Na verdade, esse não é o nome dos arquivos. O que acontece é que o nome deles é formado por caracteres unicode que não podem ser exibidos no console.

O jeito é descobrir o inode dos arquivos, e excluir através dele.

Listando o inode dos arquivos:

```
ls -li
```

Excluindo através do inode:

```
find . -inum 15681478197 -exec rm -i {} \;
```

Fonte: <https://unix.stackexchange.com/a/230030>

# Docker

Vários comandos do curso de Docker do senai:

acessar o terminal de um container que já está rodando:

```
docker run -it ubuntu
```

iniciar um container e acessar o terminal:

```
docker start -ai 05025384675e
```

iniciar um container sem travar o terminal:

```
docker run -d dockersamples/static-site
```

iniciar um container atribuindo uma porta aleatória:

```
docker run -d -P dockersamples/static-site
```

iniciar um container, mas mapeando uma porta definida com o container. no caso, a porta 8080 é da máquina host, e 80, a do container:

```
docker run -d -p 8080:80 --name site-docker dockersamples/static-site
```

exibindo as portas atreladas ao container:

```
sudo docker port 325b4cbf1b6e
```

atribuindo um nome ao container, para não precisar depender do id:

```
docker run -d -P --name site-docker dockersamples/static-site
```

```
docker stop site-docker
```

parando todos os containers em execução:

```
docker stop $(docker ps -q)
```

examinar um container:

```
docker inspect iddocontainer
```

pode ser usado os primeiros caracteres do id, não precisa ser completo:

```
docker inspect 456b0f275ee1
```

```
docker inspect 456b
```

usando uma pasta específica na hora de montar o volume:

```
docker run -v "/tmp/www:/var/www/" ubuntu
```

neste caso, a pasta **/tmp/www** da máquina host será utilizada como **/var/www** no container

criar uma rede:

```
docker network create --driver bridge minha-rede-docker
```

criar um container nessa rede:

```
docker run -it --name meu-container-1 --network minha-rede-docker ubuntu
```

```
docker run -it --name meu-container-2 --network minha-rede-docker ubuntu
```

agora dá pra acessar um container à partir de outro, usando o hostname ao invés do IP

# Usando ffmpeg

Para comprimir um vídeo:

```
ffmpeg -i input.mp4 -crf 28 output.mp4
```

Esse outro comando especifica o codec de saída como h265:

```
ffmpeg -i input.mp4 -vcodec libx265 -crf 28 output.mp4
```

Para transformar vários arquivos que estão em formato webm (ou qualquer outro formato) para mp4:

```
for file in *.webm; do ffmpeg -i "$file" -vcodec libx265 -crf 28 "${file%.webm}".mp4; done
```

Para apenas converter rapidamente um arquivo em mp4, sem processamento:

```
ffmpeg -i input.mp4 -codec copy output.mp4
```

Para mais opções: <https://askubuntu.com/questions/352920/fastest-way-to-convert-videos-batch-or-single>

# Bloqueando ip no iptables

Bloqueando uma faixa de ip:

```
iptables -A INPUT -s 35.203.210.0/24 -j DROP
```

Bloqueando apenas um endereço:

```
iptables -A INPUT -s [IP] -j DROP
```

(obs.: ainda não testei esse comando)

Exibindo os ips bloqueados:

```
iptables -L -v
```

# Comprimir imagens no terminal

Usando o imagemagick:

```
convert imagem.png -resize 80x convertida.png
```

80x é para deixar com 80px de largura

Fonte: <https://opensource.com/article/21/9/resize-image-linux>