



# BORN2BEROOT

SERVIDOR SSH

Bônus





# Sistema

**1** CPU Virtualização?  
Configurou a BIOS?



**2** Virtual Machine?  
Rede em Bridge!  
Arquivo VDI ?  
HD dinamico ?



Estude os sistemas e escolha um para seguir com o projeto!  
\*Eu escolhi o Debian, então o este guia é para Debian



**5** Nome do Host  
deverá ser seu  
login + 42  
ex: lucasmar42



**4** Proibido interface  
gráfica!  
instale o mínimo de  
serviços!

**7** **LVM**



Pastas '/' no linux?  
Pasta '/home' no linux? \*Montar separado!  
criptografia de disco ?

Suas partições deve ter essa estrutura  
comando : # lsblk

```
wil@wil:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda    HD PHYSICAL        8:0   8G  0 disk
└─sda1      8:1   0 487M 0 part /boot
└─sda2 PARTIÇÕES        8:2   0 1K   0 part
└─sda5      8:5   0 7.5G 0 part
 └─sda5_crypt 254:0   0 7.5G 0 crypt
   ├─wil--vg-root 254:1   0 2.8G 0 lvm  /
   ├─wil--vg-swap_1 254:2   0 976M 0 lvm  [SWAP]
   └─wil--vg-home 254:3   0 3.8G 0 lvm  /home
sr0      11:0   1 1024M 0 rom
```



**8** Grub ?  
instale!



**6** CRIAR O USUÁRIO!  
DEVE TER SEU LOGIN  
EX: lucasmar



# CRON / MONITORING.SH

Monitoring.sh, este é o nome do arquivo que deve rodar com o cron nele devemos mostrar informações na tela de todos os usuários logados aqui não tem como monstrar passo a passo como fazer , mais vou te orientar ao que estudar para fazer essas informações aparecer na tela.

Ferramentas úteis para a construção do Monitoring.sh

manipulação de texto:  
grep  
cut  
awk  
wc  
awk com printf  
uniq  
xargs  
if em script shell

Informações:  
uname -snrvmo  
arquivo /proc/cpuinfo  
top -bn1  
who -b  
users  
hostname  
free -m  
df -Bm  
df \_Bg

a estrutura do arquivo é quase isso, lembrando que você pode monstrar a sua da maneira que achar melhor o importante é trazer as informações corretas,o arquivo está disponível para consulta.

```
#!/bin/bash

TCP=$(grep 'TCP:' /proc/net/sockstat | awk '{print $3}')
MSG_TCP=$(if [ ${TCP} -eq 0 ]; then echo NOT ESTABLISHED; else echo ESTABLISHED; fi)

T_RAM=$(free -m | grep '^Mem.:.' | awk '{print $2}')
U_RAM=$(free -m | grep '^Mem.:.' | awk '{print $3}')
P_RAM=$(free -m | grep '^Mem.:.' | awk '{printf("%.2f"),$3/$2*100}')

DISKUSE=$(df -Bm | grep /dev/ | grep -v /boot | awk '{ud = ud + $3} END {print ud}')
DISKFULL=$(df -Bg | grep /dev/ | grep -v /boot | awk '{fd = fd + $2} END {print fd}')
DISK=$(df -Bm | grep /dev/ | grep -v /boot | awk '{ud = ud + $3} {fd = fd + $2} END {printf("%d")}

echo "
echo "
echo "                               System Control Panel"
echo "-----"
echo "#Architecture:      : $(uname -snrvmo | cut -c -56)"
echo "#Architecture:      : $(uname -snrvmo | cut -c 57-)"
echo "#CPU physical       : $(grep '^physical id' /proc/cpuinfo | uniq | wc -l)"
echo "#vCPU                : $(grep '^processor' /proc/cpuinfo | uniq | wc -l)"
echo "#Memory Usage        : ${U_RAM}/${T_RAM}MB (${P_RAM}%)"
echo "#Disk Usage          : ${DISKUSE}/${DISKFULL}Gb (${DISK}%)"
echo "#CPU load            : $(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%")})"
echo "#Last boot           : $(who -b | cut -c 37-)"
```

- \* o arquivo pode ser criado em qualquer pasta basta indicar o caminho no crontab -e
- \* lembra de habilitar o arquivo para a execução

```
# chmod 755 monitoring.sh
```

## Resultados:

```
lucasmar@lucasmar42:~$
```

System Control Panel

---

#Architecture:	: Linux lucasmar42 5.10.0-9-amd64 #1 SMP Debian 5.10.70-1 (2021-09-30) x86_64 GNU/Linux
#CPU physical	: 1
#vCPU	: 3
#Memory Usage	: 85/4956MB (1,72%)
#Disk Usage	: 1180/Gb (%)
#CPU load	: 0,0%
#Last boot	: 2021-12-06 16:22
#LVM use	: yes
#connexions TCP	: 2 ESTABLISHED
#User log	: 2
#Network	: IP 192.168.1.162 (08:00:27:b4:aa:90)
#Sudo	: 21 cmd

---



# CRON / MONITORING.SH

Para indicar qual o periodo que queremos que o cron execute o trabalho, devemos seguir uma sintese onde indica Minutos, horários, dia, mês, e o dia da semana, lembrando que aqui vamos abordar o assunto superficialmente então se aprofunde para ter mais segurança ao editar seu arquivo.

Imagens, tirada do vídeo que está na descrição.

PERIODICIDADE					COMANDO
MINUTOS	HORAS	DIAS DO MÊS	MÊS	DIAS DA SEMANA	COMANDO
0 - 59	0 - 23	1 - 31	1 - 12	0 - 7	

além dos números podemos usar alguns operadores como :

- / Significa a cada, como exemplo “\* / 10” a cada 10min.
- \* Por padrão \* significa todo de forma simples \* todos minutos.
- , Podemos separar por listas, indicando os minutos 0,10,20,30 ...
- intervalos podemos definir o intervalo como 0-10 minutos.

lembmando que os exemplos acima é só base para entendimento, com isso nosso aquivo deve ser algo como isso:

```
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
#MINUTOS 0-59 , HORAS 0-23 , DIAS DO MÊS 1-31 , MÊS 1-12 , DIAS DA SEMANA 0-7 , COMANDOS
#
#evitar arquivo log pesados msg de tela direcione a saida >/dev/null >2>&1
#
#MIN HRS DDM MDA DDS COMANDO
*/10 * * * * bash /root/monitoring.sh | wall
*/10 * * * * sleep $(who -b | awk '{print $5}' | cut -c 5-)m && bash /root/monitoring.sh | wall
```

Para resolver o problema de rodar apartir do sistema usamos o comando sleep, assim o cron executa depois de 10 minutos de quando o sistema foi iniciado, a instrução deve vir seguida do bash .....

**sleep \$(who -b | awk '{print 5}' | cut -c 5-)**

Comandos uteis para o cron

**# systemctl stop cron**

para o sistema de agendamento de forma temporaria

**# systemctl start cron**

iniciar novamente o serviço

**# systemctl enable cron**

inicia o serviço cron junto ao sistema

**# systemctl restart cron**

reinicia o serviço cron

Ferramenta Wall, basicamente permite você transmitir mensagens a todos usuários do sistema, muito usado por administradores de sistema para informar manutenções e outros, em nosso projeto vamos usar o mesmo para transmitir a informação do monitoring.sh a todos.  
comandos do wall uteis em caso que queira tirar a bandeira

**# wall oi**

```
root@lucasmar42:~# wall oi
```

```
Mensagem de broadcast de root@lucasmar42 (tty1) (Tue Dec 7 14:55:37 2021):
```

```
oi
```

**#wall -n oi**

```
root@lucasmar42:~# wall -n oi
```

```
oi
```



# CRON / MONITORING.SH

crontab?  
wall?  
script shell?

> Criar um script em bash nomeando de monitoring.sh  
ele deve mostrar as informações abaixo:

```
Broadcast message from root@wil (tty1) (Sun Apr 25 15:45:00 2021):  
  
#Architecture: Linux wil 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux  
#CPU physical : 1  
#vCPU : 1  
#Memory Usage: 74/987MB (7.50%)  
#Disk Usage: 1009/2Gb (39%)  
#CPU load: 6.7%  
#Last boot: 2021-04-25 14:45  
#LVM use: yes  
#Connexions TCP : 1 ESTABLISHED  
#User log: 1  
#Network: IP 10.0.2.15 (08:00:27:51:9b:a5)  
#Sudo : 42 cmd
```

> Na inicialização do servidor, o script exibirá algumas informações (listadas abaixo) em todos os terminais a cada 10 minutos (dê uma olhada no wall). O banner é opcional.  
Nenhum erro deve Seja visível.

Nesta etapa do projeto devemos exibir as informações acima para todos os usuários logados no servidor de 10 em 10 minutos a contar da inicialização do servidor. Aqui temos que compreender algumas ferramentas e vamos dividir isso em três fases, primeira coisa que devemos conhecer é o Cron depois a Wall e por fim script shell. Vamos lá !

## CRON

Software de agendamento de tarefas, que faz execução das mesmas sem intervenção de um humano tornando assim a administração do sistema mais autómatica, em nosso projeto vamos usar ela para exibir de 10 em 10 minutos o script, então começamos verificando se está instalada.

```
# systemctl status cron
```

caso não esteja instalado instale o mesmo assim :

```
# apt-get install cron
```

Retomando o comando status a tela deve aparecer assim:

```
root@lucasmar42:~# systemctl status cron  
● cron.service - Regular background program processing daemon  
  Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: enabled)  
  Active: active (running) since Mon 2021-12-06 16:22:08 -03; 41min ago  
    Docs: man:cron(8)  
   Main PID: 569 (cron)  
     Tasks: 1 (limit: 5904)  
    Memory: 1.0M  
      CPU: 39ms  
     CGroup: /system.slice/cron.service  
           └─569 /usr/sbin/cron -f  
  
dez 06 16:30:01 lucasmar42 CRON[675]: (root) CMD (wall oi)  
dez 06 16:30:01 lucasmar42 CRON[674]: pam_unix(cron:session): session closed for user root  
dez 06 16:40:01 lucasmar42 CRON[682]: pam_unix(cron:session): session opened for user root  
dez 06 16:40:01 lucasmar42 CRON[683]: (root) CMD (wall oi)  
dez 06 16:40:01 lucasmar42 CRON[682]: pam_unix(cron:session): session closed for user root
```

Cron ativo, vamos para a ferramenta crontab abaixo alguns comandos úteis para a configuração do Cron em nosso projeto, a idéia aqui é simples existe um arquivo onde colocaremos nossas regras para então exibir a mensagem:

```
# crontab -e  
-e edita o tal arquivo que vamos usar!  
# crontab -l  
-l exibe o arquivo, mas sem permitir edição.  
# crontab -r  
-r exclui o arquivo, ou melhor retorna ao valor padrão.
```

tendo esses comandos em mãos vamos editar o arquivo conforme solicitado, veja que as instruções no arquivo de como informar o periodo que queremos que tal tarefa seja executada.



# SSH LADO CLIENT

Para concluir de vez nossa configuração do SSH , devemos testar agora usando uma maquina client ou seja acessando de fora o nosso servidor

informação importante qualquer sistema hoje é capaz de acessar um servidor utilizando o protocolo ssh, no meu caso tenho instalado ubunto , wsl no windows e o proprio windows e todos funcionam bem pois seguem o mesmo protocolo ssh.

algumas máquinas não possuem o pacote por nativo então devemos instalar  
`# apt-get install openssh-client`

\* no caso do windows as versões a partir da 10 já possuem e vamos usar o PowerShell pra acessar nosso servidor Debian.

padrão para acessar o ssh é:

`ssh NOME_USUÁRIO_SERVIDOR@IP_DOSERVIDOR -P PORTA_CONFIGURADA`

Então em meu caso

ex:

`~$ ssh lucasmar@192.168.1.162 -p 4242`

## WSL UBUNTO NO WINDOWS

```
apontes19@JARVIS:~$ ssh lucasmar@192.168.1.162 -p 4242
lucasmar@192.168.1.162's password:
Linux lucasmar42 5.10.0-9-amd64 #1 SMP Debian 5.10.70-1 (2021-09-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  2 19:19:11 2021 from 192.168.1.167
lucasmar@lucasmar42:~$
```

para fechar a conexão pode usar o `~$ exit`, agora vamos testar acessando como usuário root o ssh deve negar isso.

```
apontes19@JARVIS:~$ ssh root@192.168.1.162 -p 4242
root@192.168.1.162's password:
Permission denied, please try again.
root@192.168.1.162's password:
Permission denied, please try again.
root@192.168.1.162's password:
```

você também pode enviar arquivos entre as máquinas usando o comando da ferramenta SCP Secure Copy que já foi instalada junto ao ssh

Enviando do Servidor a máquina local :

```
~$ spc -P 4242 lucasmar@192.168.1.162:/home/arq.txt nomearqnaclient.txt
```

resalvas do comando acima, deve ser executada da máquina cliente, -P para indicar a porta no servidor usuário e nome do servidor depois o local onde está o arquivo no servidor e por fim o caminho onde vai ser salvo o arquivo não esqueça de dar um nome a ele e se não por o caminho ele salva no home por padrão nos dois casos.

Agora enviando da máquina para o servidor:

```
~$ spc -P 4242 nomearqnaclient.txt lucasmar@192.168.1.162:~
```

Você também consegue enviar pastas inteiras apenas colocando o -r na frente

```
~$ spc -r -P 4242 home/nomepastanaclient lucasmar@192.168.1.162:~
```

>  
SSH

# IP ESTÁTICO

Antes das configurações vamos verificar qual é o nosso IP , netmask e também o getway, uma ferramenta que vai nós ajudar com isso é o net tools que não vem por padrão no Debian, então devemos installar.

```
# apt-get install net-tools
```

após a instalação vamos usar a ferramenta ifconfig que está inclusa no net.

```
# ifconfig -a
```

```
root@lucasmar42:/# ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.162 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::a00:27ff:feb4:aa90 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:b4:aa:90 txqueuelen 1000 (Ethernet)
            RX packets 1320 bytes 228869 (223.5 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 226 bytes 19028 (18.5 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Loopback Local)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
# route -n
```

```
root@lucasmar42:/# route -n
Tabela de Roteamento IP do Kernel
Destino     Roteador      MáscaraGen.    Opcões Métrica Ref   Uso Iface
0.0.0.0     192.168.1.1   0.0.0.0        UG      0      0       0 enp0s3
192.168.1.0 0.0.0.0      255.255.255.0  U        0      0       0 enp0s3
root@lucasmar42:/# _
```

IP?  
NETMASK?  
BROADCAST?  
GETWAY?  
REDE LOCAL ?  
SOCKET ?

comandos uteis

```
# ifconfig enp0s3 up
# ifconfig enp0s3 down
```

ativa e desativa a placa de rede veja o nome da sua placa antes.

Com as informações em mãos vamos agora para a configuração do IP estático

```
# nano /etc/network/interfaces
```

deixe seu arquivo parecido com a imagem abaixo, ressaltando que lógicamente você deve usar as suas informações obtidas no passo acima

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
#iface enp0s3 inet dhcp
#auto enp0s3

iface enp0s3 inet static
    address 192.168.1.162
    broadcast 192.168.1.255
    netmask 255.255.255.0
    gateway 192.168.1.1

iface enp0s3 inet6 auto
```

Após a configuração vem os testes, verifique se tem internet e depois as portas abertas e por último veja novamente as informações de IP

```
# ping 8.8.8.8
```

```
root@lucasmar42:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=111 time=85.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=111 time=47.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=111 time=38.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=111 time=42.8 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=111 time=22.9 ms
```

Agora devemos verificar se somente o socket 4242 está sendo usado no ssh para verificar isso usamos o comando

```
# ss -tunlp
```

```
ss -tunlp
Netid State Recv-Q Send-Q Local Address:Port    Peer Address:Port Process
tcp   LISTEN 0      128          0.0.0.0:4242      0.0.0.0:*      users:(("sshd",pid=633,fd=3))
tcp   LISTEN 0      128          [::]:4242        [::]:*       users:(("sshd",pid=633,fd=4))
root@lucasmar42:/#
```



# SSH

O que é Serviço SSH?  
o que posso fazer com SSH?  
Acesso Root?  
ip estático ?

- > Configurar porta 4242 e somente ela para o ssh
- > Não permitir conexão com usuário root

Nesta etapa do projeto vamos cuidar da nossa conexão remota, para isso usaremos um sistema seguro. Servidores comumente fica em locais com temperaturas controlada e restritos por segurança e isso faz com que tenhamos de configurar manutenções de forma remota sem perder a segurança dos dados que trafega nesta conexão.

Como servidor devemos instalar o serviço ssh como sever

```
# apt-get install openssh-server
```

Logo após vamos editar o arquivo de configuração conforme solicitado

```
# nano /etc/ssh/sshd_config
```

```
Port 4242 ←  
#AddressFamily any  
#ListenAddress 0.0.0.0  
#ListenAddress ::  
  
#HostKey /etc/ssh/ssh_host_rsa_key  
#HostKey /etc/ssh/ssh_host_ecdsa_key  
#HostKey /etc/ssh/ssh_host_ed25519_key  
  
# Ciphers and keying  
#RekeyLimit default none  
  
# Logging  
#SyslogFacility AUTH  
#LogLevel INFO  
  
# Authentication:  
  
#LoginGraceTime 2m  
PermitRootLogin no ←  
#StrictModes yes
```

Tudo configurado devemos agora habilitar o serviço e definir para que ele inicie sempre junto ao sistema.

```
# systemctl start sshd  
# systemctl enable sshd
```

\*Vale aqui lembrar que estamos estabelecendo conexão local e via autenticação via senha, podemos deixar isso mais seguro e automático além de conectar na rede mundial, mais não é o propósito da parte mandatória.

vamos agora fazer a configuração do IP para estático, isso vai nos possibilitar se conectar de forma mais facil na máquina cliente além de aumentar a segurança pois em algumas máquinas pode deixar mais socket abertos além do 4242 solicitado, veremos isso a frente



# APT X APTITUDE

Etapa onde devemos apenas buscar informações de como usar as ferramentas apresentadas , entender as diferenças e optar por uma delas, eu optei por permanecer com a apt que vem como padrão no Debian , pois para o que temos que fazer é o suficiente.

os dois são gerenciadores de pacotes, assim ele mantem uma lista no sistema uma biblioteca onde ao solicitar a instalação de uma aplicação como o python eles buscam as dependencias desta app e instala pra você sem que você tenha que ler o manual pra saber o que baixar na máquina.

Principal diferença deles é o fato do aptitude possuir uma interface para que o administrador possa gerenciar melhor os pacotes, como já foi mencionado eles instala pacotes , exclui, busca e gerência a suas atualizações bem como limpa os arquivos baixados quando não mais utilizados , no caso do aptitude faz automático pra você. Outra diferença é que o aptitude trabalha com um número maior de bibliotecas assim atende mais sistemas

## comandos úteis do apt

```
# apt-get install vim
```

Instala pacotes somente com o nome 'vim'

```
# apt-get remove vim
```

Desinstala pacotes também com o nome

```
# apt search vim
```

Pesquisa dentre os pacotes aqueles que tem a palavra 'vim'

```
# apt-get update
```

atualiza somente os pacotes de aplicações e não sistema

```
# apt-get upgrade
```

atualiza pacotes de sistema e não de aplicações

```
# apt autoremove
```

remove arquivos de pacotes que já foi instalados

```
# apt-get full-upgrade
```

atualiza pacotes de aplicações, sistemas e também remove os já instalados

```
# apt-get aptitude
```

```
# aptitude
```

Tela do aptitude

```
Ações Desfazer Pacote Resolvedor Pesquisar Opções Visões Ajuda
C-T: Menu ?: Ajuda q: Sair u: Actualizar g: Visualizar/Download/Instalar/Remover Pacotes
aptitude 0.8.13 @ lucasmar42
```

```
--- interpreters  Interpretadores para linguagens interpretadas (1)
--- kernel       Kernel and kernel modules (4)
--- libs          Coleções de rotinas de software (135)
--- localization Language packs (5)
--- metapackages Packages which solely depend on other packages (1)
--- misc          Miscelânia de softwares (6)
--- net           Programas para se conectar e fornecer diversos serviços (11)
--- perl          Interpretador e bibliotecas Perl (8)
--- python        Interpretador e bibliotecas Python (7)
--- shells        Shells de comando e ambientes de console alternativos (2)
--- text          Utilitários de processamento de textos (8)
--- utils         Diversos utilitários de sistema (35)
--- x11           O X window system e softwares relacionados (3)
--- Pacotes não instalados (58318)
--- Pacotes virtuais (19526)
--- Tarefas (222)
```

```
Packages in the 'kernel' section provide the core of the operating system. They include the
operating system kernel itself, along with extension modules providing features such as support for
unusual hardware and support for running virtual machines.
```

```
Este grupo contém 4 pacotes.
```



# AppArmor x SELinux

Apparmor como funciona ?  
SELinux tem em todos sistemas e ou só o centOS?  
protege exatamente o que ?  
rsyslog?

Neste requesito do sistema não temos nada a fazer, tirando o fato de conferir se está funcionando, abaixo segue alguns comandos para isso.

```
# cd /etc/apparmor.d/  
# ls -la
```

```
root@lucasmar42:/etc/apparmor.d# ls -la  
total 36  
drwxr-xr-x 7 root root 4096 nov 24 19:19 .  
drwxr-xr-x 68 root root 4096 dez  1 12:00 ..  
drwxr-xr-x  4 root root 4096 nov 24 19:19 abstractions  
drwxr-xr-x  2 root root 4096 abr  3 2021 disable  
drwxr-xr-x  2 root root 4096 abr  3 2021 force-complain  
drwxr-xr-x  2 root root 4096 nov 24 19:19 local  
-rw-r--r--  1 root root 1313 abr  3 2021 lsb_release  
-rw-r--r--  1 root root 1167 abr  3 2021 nvidia_modprobe  
drwxr-xr-x  5 root root 4096 nov 24 19:19 tunables  
root@lucasmar42:/etc/apparmor.d#
```

## Exemplo Lista de Perfil AppArmor

A manutenção é feita alterando os arquivos e e ou criando um novo perfil para monitorar e para isso tem comando específicos consulte o manual

```
# aa-status
```

Comando para verificar o status do AppArmor

```
root@lucasmar42:/# aa-status  
apparmor module is loaded.  
3 profiles are loaded. ←  
3 profiles are in enforce mode.  
    lsb_release  
    nvidia_modprobe  
    nvidia_modprobe//kmod  
0 profiles are in complain mode.  
0 processes have profiles defined.  
0 processes are in enforce mode.  
0 processes are in complain mode.  
0 processes are unconfined but have a profile defined.  
root@lucasmar42:/# _
```

```
# less nvidia_modprobe
```

```
# vim:syntax=apparmor  
  
#include <tunables/global>  
  
profile nvidia_modprobe {  
    #include <abstractions/base>  
  
    # Capabilities  
  
    capability chown,  
    capability mknod,  
    capability setuid,  
    capability sys_admin,  
  
    # Main executable  
  
    /usr/bin/nvidia-modprobe mr,  
  
    # Other executables  
  
    /usr/bin/kmod Cx -> kmod,  
  
    # System files  
  
    /dev/nvidia-modeset w,  
    /dev/nvidia-uvm w,  
    /dev/nvidia-uvm-tools w,  
    @sys/bus/pci/devices/ r,  
    @sys/devices/pci[0-9]*/**/config r,  
    @PROC/devices r,  
    @PROC/driver/nvidia/params r,  
    @PROC/modules r,  
    @PROC/sys/kernel/modprobe r,  
  
    # Child profiles  
  
    nvidia_modprobe_
```

## Exemplo de Perfil AppArmor

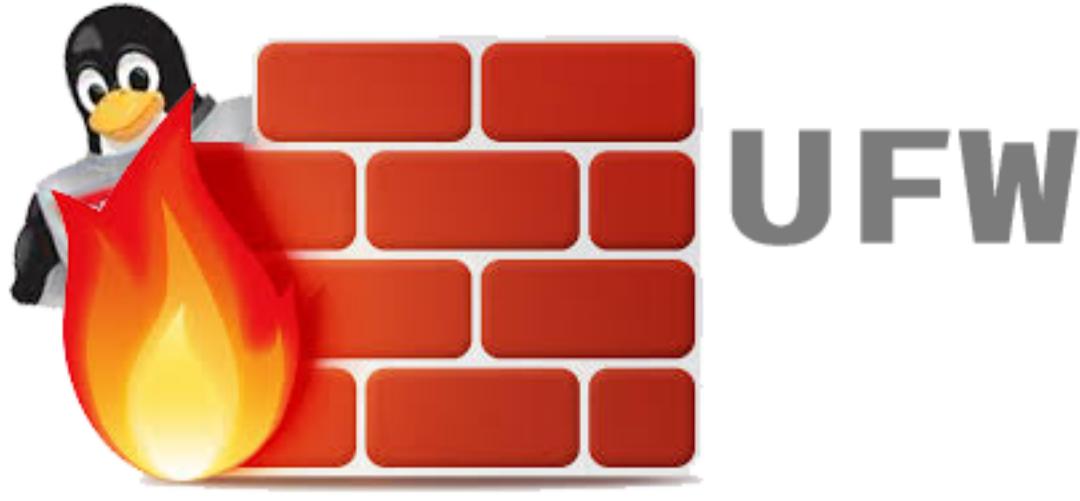
```
# systemctl enable apparmor
```

Comando para iniciar com o sistema

Vale ressaltar que é obrigação saber o que cada um deles faz e quais as diferenças entre eles , recomendo ir mais a fundo neste tema pois tem bastante coisa para saber sobre.Vou deixar aqui um pequeno resumo mais indico ler a fundo os manuais.

AppArmor:um aplicativo que vem instalado por padrão no Debian desde 2019 versão 10 do Debian, ele vem como uma reforço a segurança do sistema onde seu foco é gerenciar os recursos das aplicações e o faz de forma a gerar perfil AppArmor assim facilitando a vida do administrador do sistema, é importante dizer que ele não impede comportamentos estranhos só registra os mesmos, e se destaca na comunidade por ser mais simples de usar alem de funcinar em qualquer sistema de arquivo software mantido pela Caninocal desde 2009.

SELinux: tem o mesmo propósito que é controlar atividades dos apps, porem é totalmente diferente no modo de funcionamento ele busca no kernel e por esse motivo é mais complexo e completo ele não funciona muito bem junto com o AppArmor devido ambos controlar o comportamento dos softwares e vem ativado por padrão no CentOS , porem é possivel usar no Debian.



UFW ?  
FireWall como Funciona?  
como computadores se comunica na rede ?  
o que são porta de rede ?

## Política do UFW

- > Instale um firewall UFW
- > o Firewall deve estar ativo no inicio da Máquina
- > Configurar apenas a porta 4242 aberta, **\*se fizer o Bônus isso muda**

Instalando o UFW

```
# apt-get install ufw
```

Configurando o Start do ufw ao ligar o sistema

```
# systemctl enable ufw
```

Permitindo a porta 4242

```
# ufw allow 4242
```

```
root@lucasmar42:/# ufw status
Status: active

To                         Action      From
--                         ----       ---
4242                       ALLOW      Anywhere
4242 (v6)                  ALLOW      Anywhere (v6)
```

comandos uteis do ufw, você pode conferir no **m# man ufw**

```
# ufw status
```

Mostra as regras atuais do ufw

```
# ufw deny 4242
```

Bloqueia um a porta desejada

```
# ufw status numbered
```

Mostra as regras atuais com o numero de registro dela  
assim é possivel excluir utilizando o número.

```
# ufw delete 1
```

Deleta uma regra do ufw, veja que ao excluir os outros  
registros tambem atualiza os numeros

```
# ufw enable
```

Habilita o serviço de ufw

```
# ufw disable
```

Desabilita o serviço UFW

```
# ufw reset
```

Reinicia o serviço ufw, veja que isso desconecta a máquina  
para que ele possa verificar novamente as regras atualizadas



# SUDO

SUDO X SU ?  
TTY ? e no sudo ?  
por que estes caminhos deve ser restritos?

## Política do SUDO

- > Limitar a 3 tentativas de acesso para senha incorreta
- > Mensagem personalizada ao errar a senha de acesso
- > Cada ação com sudo deve ser salvo em /var/log/sudo/**sudo.log**
- > Modo TTY deseja ser ativado
- > Caminhos restritos ao sudo deve ser:  
`/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin`

Primeiro instalamos o sudo caso não tenha instalado

```
# apt-get install sudo
```

Agora editamos o arquivo /etc/sudoers, veja que temos que editar com visudo para que ele verifique se está ok

```
# visudo
```

```
# This file MUST be edited with the 'visudo' command as root.  
#  
# Please consider adding local content in /etc/sudoers.d/ instead of  
# directly modifying this file.  
#  
# See the man page for details on how to write a sudoers file.  
#  
Defaults      env_reset  
Defaults      mail_badpass  
Defaults      passwd_tries=3  
Defaults      badpass_message="Aqui não meu Jovem , volte com a senha correta!"  
Defaults      log_host,log_year,logfile="/var/log/sudo/ologsudo"  
Defaults      requiretty  
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"  
  
# Host alias specification  
  
# User alias specification  
  
# Cmnd alias specification  
  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
#alterar  
# Allow members of group sudo to execute any command  
%sudo   ALL=(ALL:ALL) ALL  
  
# See sudoers(5) for more information on "@include" directives:  
  
#includedir /etc/sudoers.d
```

\*USUÁRIO MÁQUINAS=(USUÁRIOS\_DA MAQUINA:GRUPOS) COMANDOS  
root ALL=(ALL:ALL) ALL

Não é solicitado mas caso queira pode restringir a comando  
lucasmar ALL=(ALL:ALL) apt,visudo

Atenção você deve criar o arquivo para os logs caso contrário ele não  
cria sozinho! **sudo.log**

```
# touch /var/log/sudo/sudo.log
```

Devemos neste passo colocar o usuário lucasmar criado no grupo sudo:  
antes conforme solicita o pdf devemos criar os grupos sudo e user42

```
# addgroup user42  
  
# adduser lucasmar user42  
# adduser lucasmar sudo
```

\*VEJA ADM USER E GROUPS



# Política de Senha

- > Expirar a cada **30** Dias
- > Tempo mínimo para alterar a senha **2** dias.
- > Avisar o usuário com **7** dias antes.

para esses parâmetros vamos definir como padrão editando o arquivo:

```
# nano /etc/login.defs
```

```
#
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#      PASS_WARN_AGE    Number of days warning given before a password expires.
#
PASS_MAX_DAYS     30
PASS_MIN_DAYS     2
PASS_WARN_AGE     7
```

\*os usuários que já existia tem que ser modificados um a um devido esse ser um arquivo padrão para novos usuários.

comandos para alterar

```
# chage -M 30 username
# chage -m 2 username
# chage -W 7 username
# chage -l username
```

```
root@lucasmar42:/# chage -l lucasmar
Última mudança de senha : nov 26, 2021
Senha expira : dez 26, 2021
Senha inativa : nunca
Conta expira : nunca
Número mínimo de dias entre troca de senhas : 2
Número máximo de dias entre troca de senhas : 30
Número de dias de avisos antes da expiração da senha : 7
```

para o restante das regras teremos de instalar o pam-pwquality

```
# apt-get install libpam-pwquality
```

- > Mínimo de **10** caracteres
  - > Letra Maiúscula
  - > Número
  - > Não pode conter mais que 3 idênticos e consecutivos ex:qqqq X
  - > Não deve incluir o nome do usuário
  - > Deve ter ao menos 7 caracteres diferentes da senha anterior
- \* Somente essa regra não aplica ao ROOT

```
# nano /etc/security/pwquality.conf
```

```
# Number of characters in the new password that must not be present in the
# old password.
difok = 7
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
minlen = 10
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
dcredit = -1
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
ucredit = -1
#
# The maximum number of allowed consecutive same characters in the new password.
# The check is disabled if the value is 0.
maxrepeat = 3
#
# Whether to check if it contains the user name in some form.
# The check is enabled if the value is not 0.
usercheck = 1
#
# Prompt user at most N times before returning with error. The default is 1.
retry = 3
#
# Enforces pwquality checks on the root user password.
# Enabled if the option is present.
enforce_for_root
```

Agora deve **mudar a senha** dos usuários para a nova política de senha.

```
# passwd lucasmar
```

\*com o usuário Root não é necessário digitar senha antiga!

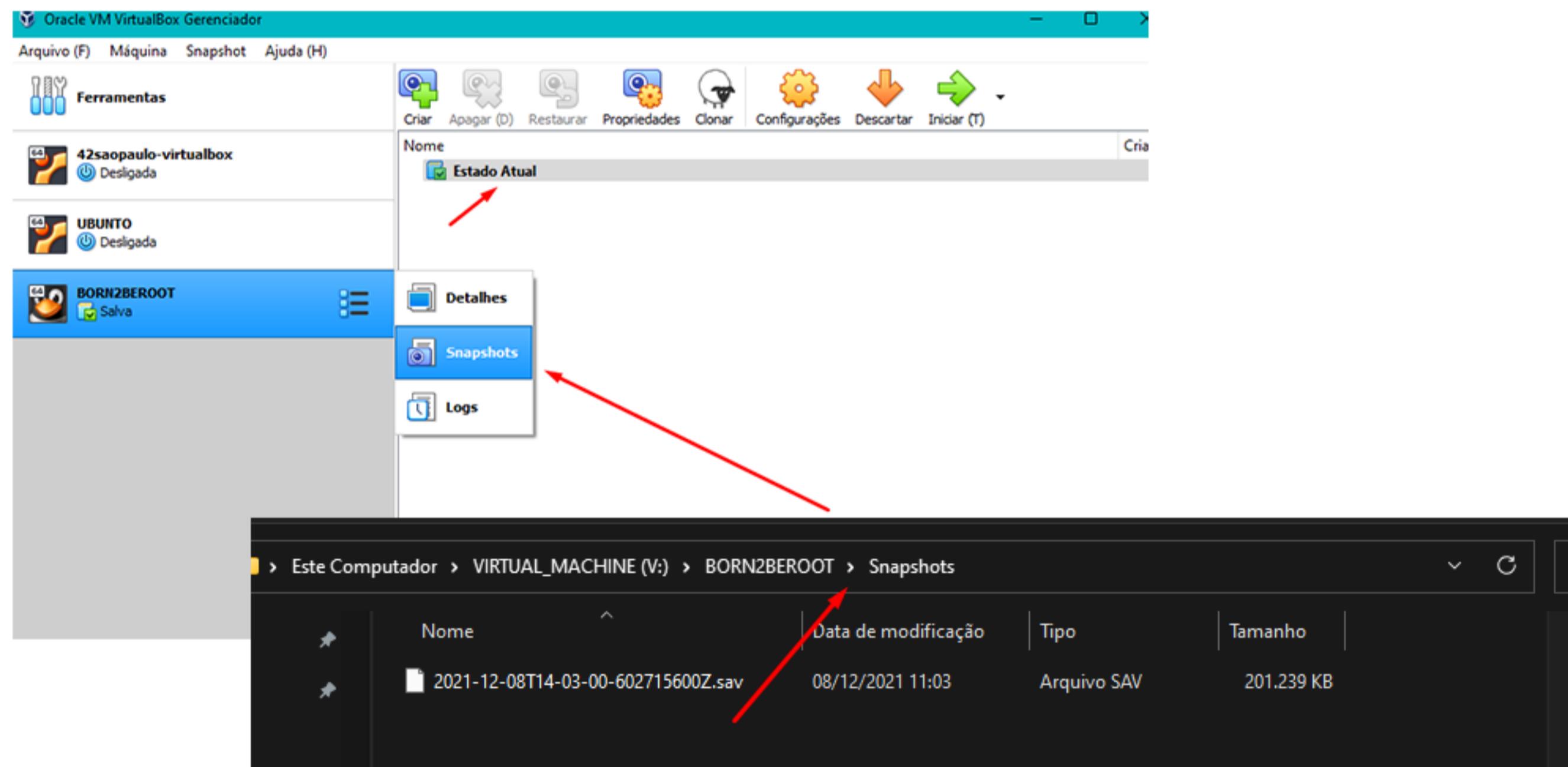
```
root@lucasmar42:/# passwd
```

```
Nova senha: _
```

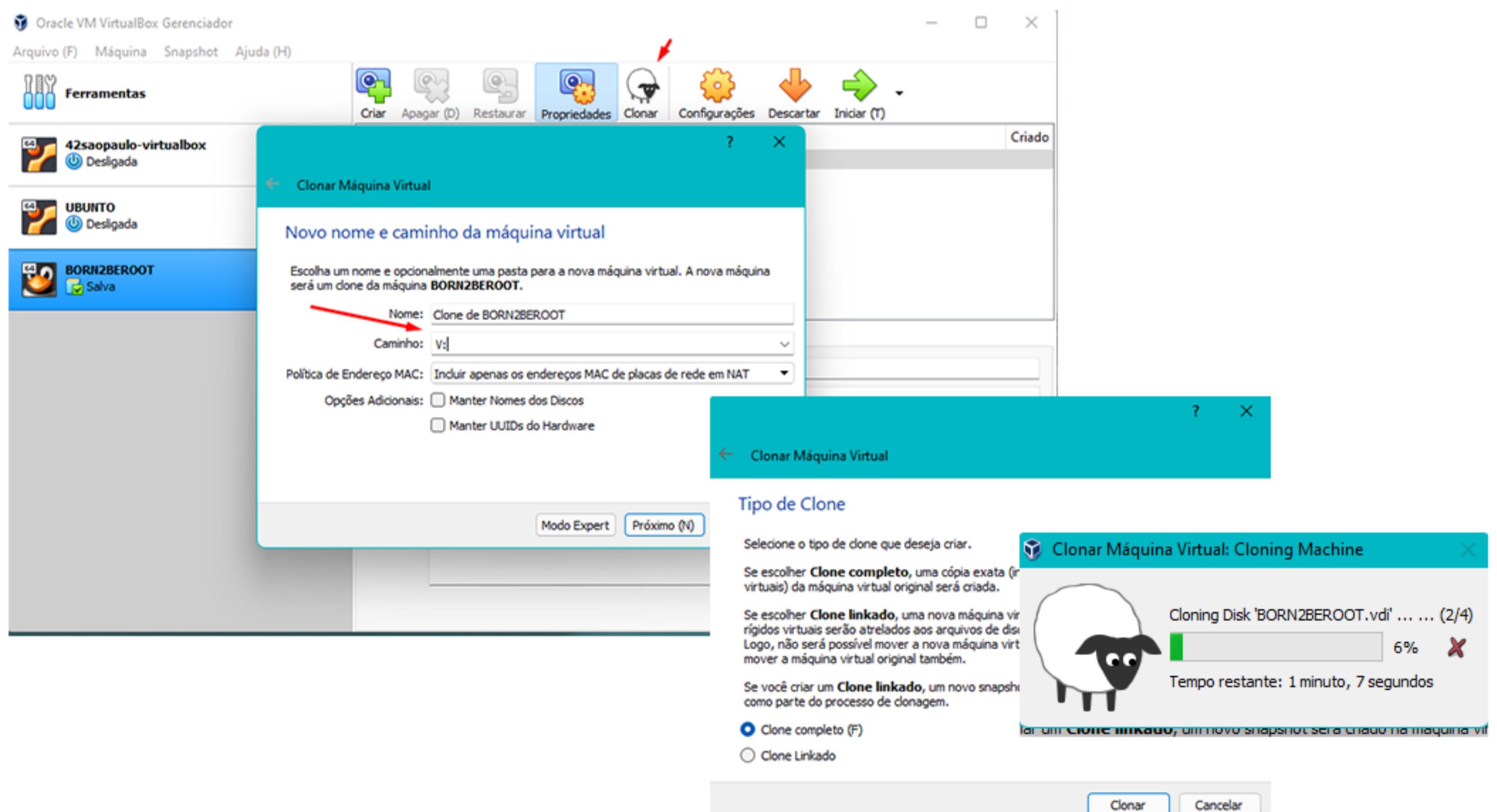


# signature.txt

Existe outras formas de fazer isso é clonando a máquina virtual diretamente pelo programa virtual box e ou salvando o estao da máquina no momento que fez a chave, e sempre se inicial a máquina você usa este estado salvo.



Assim você pode tirar a chave deste arquivo, que é o retrato do sistema.





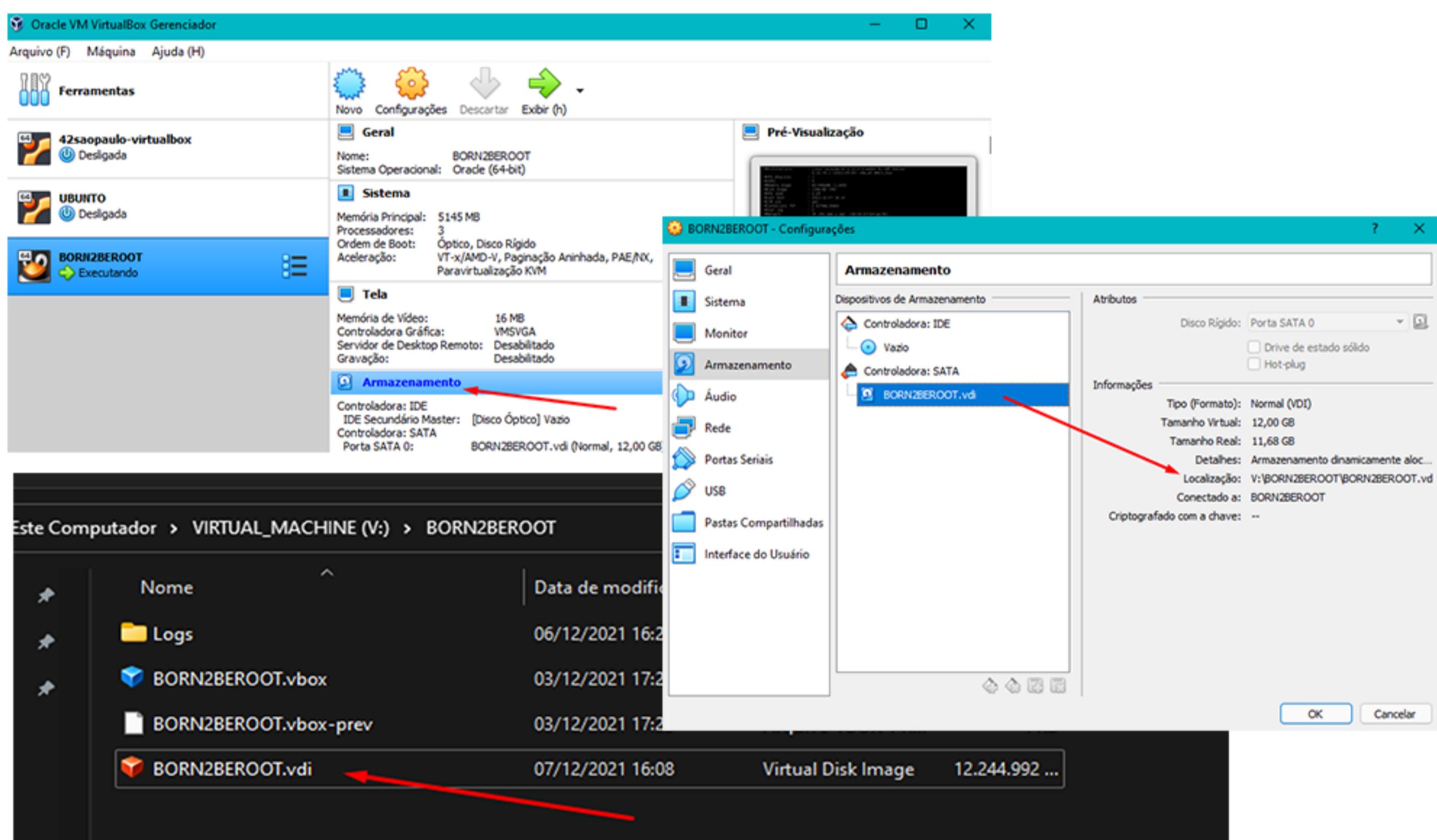
# signature.txt

disco virtual ?  
assinatura de disco ?

> Você deve colar no arquivo chamado signature.txt a assinatura do disco virtual de sua máquina.

> Você só precisa entregar um arquivo signature.txt na raiz do seu repositório Git.

Encontre o caminho onde está o seu disco virtual , existe pastas padrão veja no virtualbox onde salvou o mesmo.



veja um exemplo de como obter a chave apartir do wsl no windows

```
apontes19@JARVIS:~$ sha1sum /mnt/f/LUCAS\ DADOS/BORN2BEROOT.vdi
5c1a1f3eb1e6d5933449398c540eee68f507c4d4 /mnt/f/LUCAS DADOS/BORN2BEROOT.vdi
apontes19@JARVIS:~$
```

veja um exemplo de como obter a chave apartir do windows no PS

```
PS F:\LUCAS DADOS> certUtil -hashfile BORN2BEROOT.vdi sha1
SHA1 hash de BORN2BEROOT.vdi:
5c1a1f3eb1e6d5933449398c540eee68f507c4d4
CertUtil: -hashfile : comando concluído com êxito.
PS F:\LUCAS DADOS>
```

Veja isso é uma certificação que o disco não foi alterado, porém no caso do nosso projeto temos de apresentar a máquina a três pessoas e só o simples fato de abrir e fechar a máquina pode alterar o arquivo vdi que em nosso caso é o disco.

Para que esse número não altere, se possível podemos copiar este disco "arquivo vdi" a um pasta e não mexer mais nele e assim fazer a apresentação normalmente com o arquivo original e gerar a chave com o arquivo copia.



# USER / GROUPS COMANDOS

Vou deixar aqui alguns comandos úteis para a administração de usuários e grupos dentro do sistema, bem como comandos para verificar o status de disco e outros:

```
# adduser nome_novo_usuário
criar um novo usuário

# passwd nome_usuário
altera a senha do usuário

# deluser nome_usuário
deleta um usuário

# ls /home/
lista de pastas de usuários

# adduser nome_usuário nome_grupo
adicionar um usuário a um grupo

# addgroup nome_novo_grupo
adicionar um grupo

# delgroup nome_grupo
deletar um grupo

# groups nome_usuario
lista de grupos do usuário

# nano /etc/group
lista de grupos

# reboot
reinicia o sistema

# exit
faz logoff no sistema

# poweroff
desliga o sistema

# lsblk
lista as partições do sistema

# head -n 2 /etc/os-release
informa a versão do sistema

#hostnamectl set-hostname nome_novo
altera o nome do hostname do sistema

#hostnamectl
consulta informações do hostname
```