

Recomendações: Os textos escritos nesse bagulho (no caso os textos descrevendo certos objetos e comandos) foram feitos com a minha visão das informações, aconselho a quem utilizar esse documento que leia os conteúdos das fontes e elabore suas próprias conclusões.(é a boa brother)

- **O que é uma Máquina Virtual e pra que serve**

Uma máquina virtual é um recurso que simula um ambiente computacional capaz de executar sistemas operacionais e programas como se fosse uma máquina física, basicamente um computador dentro de um computador. Esse recurso é extremamente útil pelo fato de te dar acesso a recursos de outro sistema operacional em uma janela dentro do seu próprio sistema operacional, o que te permite testar programas em outro ambiente, ter acesso às coisas nativas de outro sistema e muito mais.

Fontes:

<https://tecnoblog.net/302438/o-que-e-uma-maquina-virtual/>

<https://www.tecmundo.com.br/maquina-virtual/232-o-que-sao-maquinas-virtuais-.htm>

Criando a Máquina Virtual

- **Escolhendo Sistema Operacional**

Você deve escolher qual sistema irá utilizar entre o CentOS e o Debian
→ CentOS vs. Debian

Os dois sistemas são de distribuição Linux, logo de código aberto, porém possuem algumas diferenças

Os CentOS é considerado mais de classe empresarial, isso pelo fato de receber atualizações com pouca frequência, o que torna ele um sistema mais estável que o Debian. Já o Debian pode ser considerado mais de classe pessoal/doméstica, já que é utilizado pela maioria da comunidade nesse ambiente. O Debian ser menos estável que o CentOS não significa que ele é instável, ele continua sendo muito estável também, mesmo recebendo atualizações com muito mais frequência que o CentOS

A comunidade do Debian é bem maior nas redes que a do CentOS, o que possibilita muitos fóruns de ajuda pra diversos problemas

Como o Debian recebe atualizações com mais frequência que o CentOS, ele acaba tendo pacotes mais atualizados, coisa que não acontece com o CentOS, logo pra ter a versão mais recente de certos recursos com o CentOS, a atualização teria que ser feita manualmente. Temos também que a quantidade de pacotes oferecidos pelo Debian geralmente é maior que a quantidade oferecida pelo CentOS, o que pode ser considerado uma vantagem pro Debian

O Debian possui uma interface gráfica mais amigável que o CentOS e trabalha com os gerenciadores de pacote distribuidores de .deb, enquanto o CentOS trabalha com os distribuidores .rpm

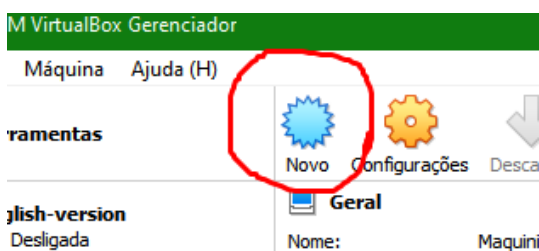
Faça o download da iso do sistema de sua preferência. O Sistema Operacional escolhido nesse texto foi o Debian <https://www.debian.org/index.pt.html>

Fontes:

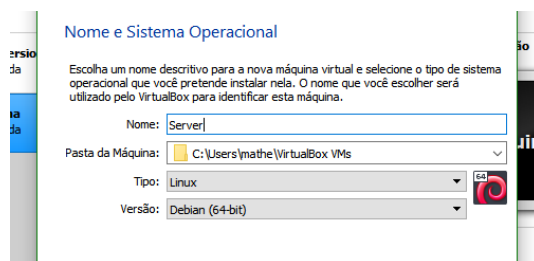
<https://www.educba.com/centos-vs-debian/>

<https://www.hostinger.com.br/tutoriais/centos-vs-ubuntu-qual-escolher-para-servidor-web/>

• Setando as Configurações Iniciais



Selecione a opção de criar máquina virtual na VM



Selecione o nome, o tipo e a versão compatível com a iso que você baixou

Tamanho da memória

Selecione a quantidade de memória (RAM) em megabytes que será alocado para a máquina virtual.

O tamanho recomendado para memória é de **1024MB**.



Selecione uma quantidade de RAM pra máquina ao seu gosto (recomendo por 1GB que já será o suficiente para os planos do projeto)

Selecione as opções:

- Criar um novo disco rígido virtual agora
- VDI (VirtualBox Disk Image)

Armazenamento em disco rígido físico

Escolha se o arquivo contendo o disco rígido virtual deve crescer à medida em que é utilizado (dinamicamente alocado) ou se ele deve ser criado já com o tamanho máximo (tamanho fixo).

Um arquivo de disco rígido virtual **dinamicamente alocado** irá utilizar espaço em seu disco rígido físico à medida em que for sendo utilizado (até um **tamanho máximo pré-definido**), mas não irá encolher caso seja liberado espaço nele.

Um arquivo de disco rígido virtual de **tamanho fixo** pode levar mais tempo para ser criado em alguns sistemas, mas geralmente possui acesso mais rápido.

- ☒ Dinamicamente alocado
☐ Tamanho Fixo

Selecione o tipo de armazenamento da máquina ao seu gosto (eu preferi optar por dinamicamente alocado, da a VM a oportunidade de expansão mais facilmente caso necessário)

Localização e tamanho do arquivo

Informe o nome do arquivo em disco que conterá o disco virtual no campo abaixo ou clique no ícone da pasta para selecionar uma localização diferente para o arquivo.

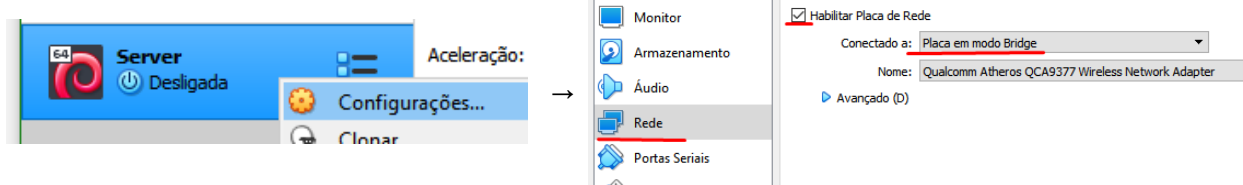
C:\Users\mathe\VirtualBox VMs\Server\Server.vdi

Selecione o tamanho da imagem de disco virtual em megabytes. Este tamanho é o limite máximo de dados que uma máquina virtual poderá armazenar neste disco rígido.

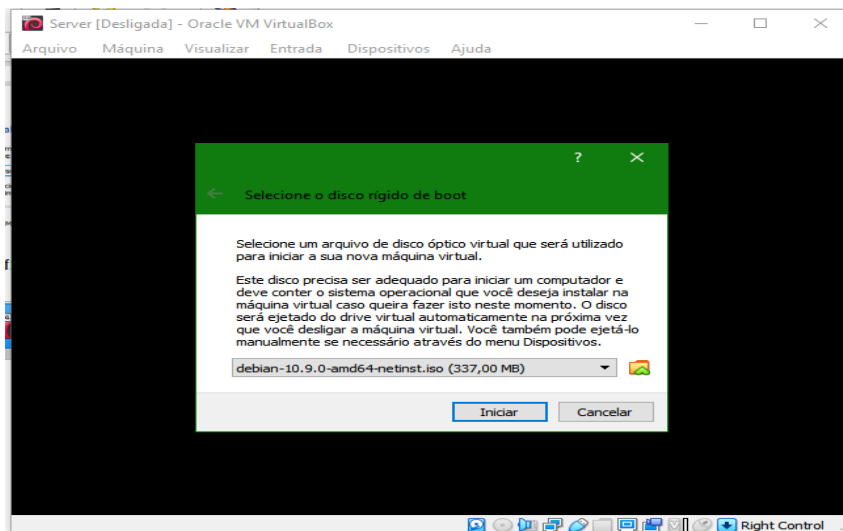


Selecione uma quantidade de ROM pra máquina ao seu gosto (as configurações do subject mostram que foi alocado 8GB nessa parte, caso você queira seguir)

Configure a rede em modo bridge para as conexões



Inicie a máquina e escolha a iso que você fez o download



Fontes:

<https://blog.remontti.com.br/1134>

- **Instalando o Sistema Operacional (Debian)**

Selecione a opção install

Selecione a língua de preferência (minha escolha foi o inglês)

Selecione sua localização

(o sistema tentara combinar sua localização com a linguagem escolhida, logo se você escolher alemão e por um país da Ásia, ele vai ficar pedindo mais informações de localização pra tentar casar os dois, então minha recomendação é por um país que tenha como língua principal a escolhida anteriormente)

Selecione linguagem do teclado

(eu recomendo você conhecer a linguagem do seu teclado, pois caso você ponha uma linhagem diferente da sua tu vai passar raiva digitando “[“ e aparecendo “{“, mas mesmo que você não ponha a certa, ao finalizar a instalação da pra alterar essa opção então tá suave)

Defina um hostname

(o subject pede pra ser seu user+42 ,exemplo mmoreira42, porém teoricamente você pode pôr o que quiser já que tu vai ter que aprender a mudar o hostname dentro do sistema mesmo, só não esquece de deixar user+42 quando for entregar)

Não é necessário definir nenhum nome de domínio

(não tenho certeza se não é permitido definir um nome de domínio, não achei nada do subject, mas como eu sou escaldado e só faço o que pedem, eu não botei nada)

Defina uma senha root e confirme ela

(mais tarde você vai precisar mudar a senha do root pra se adequar as politicas de senha, então teoricamente pode por qualquer coisa simples pra facilitar a vida na configuração do servidor, só não pode esquecer de alterar antes da entrega)

Não é necessário definir nenhum nome completo de usuário

(essa opção é só uma flag que vai entrar na descrição do usuário que você vai criar na próxima opção, é opcional você por informações adicionais sobre os usuários do seu servidor, então mesmo esquema do nome de domínio, se não pedir eu não boto)

Crie um usuário além do root

(mesmo tendo como fazer isso dentro do sistema, essa parte é obrigatória pra instalação. Como no subject também pede pra ter no mínimo os usuários root e um com seu login na intra, eu recomendo você já criar esse usuário com seu login na intra, exemplo mmoreira)

Defina uma senha pra esse usuário e confirme ela

(mesmo esquema da senha root, vai ter que mudar então pode definir qualquer coisa)

Selecione sua zona de fuso horário

(as opções vão se basear no país que você escolheu lá no começo)

Selecione como método de partição a utilização do disco inteiro com LVM criptografada

(obrigatório pelo subject. LVM É um sistema de gerenciamento de volumes lógicos do kernel do Linux, com ele é possível realizar operações com os volumes do disco de forma mais flexível e inteligente que os métodos tradicionais. O LVM por trabalhar com volumes lógicos pode, por exemplo, redimensionar partições do disco enquanto eles são ocupados ou estão sendo utilizados, coisa que outros métodos só conseguem fazer com partições livres ou fora de uso.

Fontes:

<https://www.youtube.com/watch?v=k5ZrQzwHW88>

<https://wiki.ubuntu.com/Lvm>

<https://www.certificacaolinux.com.br/logical-volume-manager-lvm-no-linux/>)

Selecione o disco pra particionar

(só tem uma opção mesmo)

Selecione partição home separada

(obrigatória pelo subject)

Selecione sim pra escrever no disco de acordo com as opções selecionadas**Defina uma senha encriptada de 20 caracteres e confirme ela**

(como não tem nenhuma restrição dessa senha no subject, pode ser livre pra escolher)

Defina o volume do disco que vai ser usado no particionamento guiado

(o tipo de partição mostrada no subject sugere que nessa configuração foi escolhido 100% do disco disponível caso você queira seguir)

Selecione a opção de finalizar o particionamento e escrever as alterações no disco**Confirme a escrita no disco****Recuse a leitura de outro CD/DVD pra pacotes adicionais****Defina um país para puxar o repositório de pacotes do debian**

(eu recomendo escolher o país que você definiu lá no início ou um perto dele)

Selecione um espelho do repositório de pacotes debian

(eu segui o recomendado e escolhi o deb.debian.org)

Não selecione nenhum HTTP proxy

(se não pedir eu não faço)

Recuse o popularity-contest

(se não pedir eu não faço)

Selecione somente os softwares de servidor web, servidor ssh e utilitários do sistema padrão

(o subject pede pra não instalar mais que o necessário, então lança no máximo esses 3 aí, sendo que o servidor web só é necessário pra quem for fazer o bônus, no meu caso eu não botei)

Selecione sim para a instalação do GRUB

(ele é um sistema de bootloader, trabalha na escolha do sistema operacional que vai ser usado quando a máquina iniciar, então é necessário)

Selecione o disco que vai ser instalado o carregador de inicialização

(seleciona o /dev/sda mesmo)

Instalação completa corre pro abraço no continue

Fontes:

500 tentativa e erro + Aninha, Tomás, Tuco e Tuca (geral brabo demais)

Configurando o Servidor

Ao iniciar o sistema, será pedido a senha de 20 caracteres e depois o login do usuário que você deseja iniciar a sessão (recomendo que você logue no usuário root pois vai precisar configurar muita coisa com permissão de superusuário)

Caso você queira trocar de usuário utilize **logout** ou **exit** pra deslogar e logue no usuário desejado

Depois de logado, confira com o comando **lsblk** se o particionamento do disco ficou igual ao do subject. Se não estiver, inicie o processo todo de novo e siga os passos namoralzinha.

```
wil@wil:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                  8:0    0   8G  0 disk
├─sda1                              8:1    0 487M  0 part  /boot
├─sda2                              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
```

As informações sobre o sistema operacional que você escolheu estão no arquivo **/etc/os-release**. Caso você queira consultar utilize

head -n 2 /etc/os-release → consultar as linhas que dizem qual é o sistema

cat /etc/os-release → consultar o arquivo todo

Comandos uteis para o boot

reboot → Reinicia a máquina (precisa de permissão root)

poweroff → Desliga a máquina (precisa de permissão root)

- **Escolhendo um Gerenciador de Pacotes**

→ Apt vs. Aptitude

Talvez a principal diferença entre os dois seja o comportamento em relação ao tratamento de pacotes dependentes, o aptitude lida melhor com eles na hora da remoção enquanto o apt-get só remove os pacotes solicitados (apesar de isso ser trabalhado com o apt-get autoremove)

O apt-get trabalha com uma interface em linhas de comando, mostrando os procedimentos que estão ocorrendo enquanto o aptitude além disso também possui uma interface gráfica no terminal mostrando os pacotes instalados e os que ainda não foram instalados. O aptitude possui ferramentas mais intuitivas e é mais convidativo, em contra partida o apt conta com uma busca de pacotes mais completa utilizando o apt-cache

O apt vem por padrão no Debian e é mais utilizado pela comunidade, logo possui muito mais fóruns de ajuda que o aptitude

Gerenciador de Pacotes Escolhido: Apt

Fontes:

<https://www.ubuntudicas.com.br/2009/05/as-diferencas-entre-apt-get-e-aptitude/>

<https://www.ubuntudicas.com.br/2015/06/apt-get-dpkg-e-aptitude/>

<http://pthree.org/2007/08/12/aptitude-vs-apt-get/>

<https://www.hardware.com.br/guias/ubuntu/apt-get-aptitude.html>

- **Instalando e Configurando o Sudo**

Utilize o seu gerenciador de pacotes para instalar o sudo (é necessário estar no root)

apt-get install sudo

Para adicionar as regras do sudo pedidas no subjecto é preciso editar o arquivo **/etc/sudoers** através do comando **visudo** (para realizar operações que precisam de sudo com outro usuário além do root, você precisa incluir esse usuário no grupo sudo: **gpasswd -a <usuário> sudo**)
sudo visudo

Existe também a opção de incluir um novo arquivo no diretório **/etc/sudoers.d** (que é escaneado e incluído no **/etc/sudoers**) e depois editá-lo utilizando o comando **visudo -f**
(cd /etc/sudoers.d ; touch <arquivo de sua escolha>)
sudo visudo -f /etc/sudoers.d/<arquivo de sua escolha>

Pra configurar as regras pro sudo inclua as seguintes linhas em um dos arquivos

#Limite de tentativas de senha igual a 3
Defaults passwd_tries=3

#Mensagem personalizada de erro de senha
Defaults badpass_message="<mensagem de sua escolha>"


```
#Salvar as informações de log do sudo no diretório "/var/log/sudo/." (não esqueça de criar esse diretório)
```

```
Defaults logfile="/var/log/sudo/<arquivo de sua escolha>"
```

```
#Habilitar o modo TTY
```

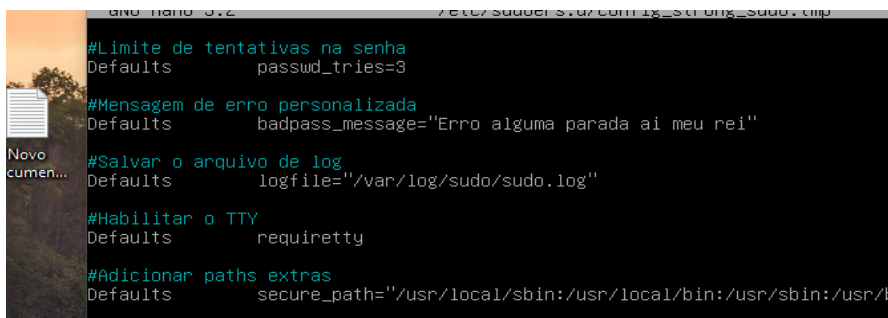
```
Defaults requiretty
```

Já existe uma linha no documento `/etc/sudoers` com as informações de `secure_path`, você tem a opção de alterar ela acrescentando o diretório `/snap/bin` ou comentar ela e pôr a nova com todos os diretórios pedidos (eu optei por comentar a linha e escrever a nova)

```
#Adicionar o path extra pro sudo
```

```
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
```

```
/snap/bin "
```



Eu botei os comentários por vontade própria, mas eles não são necessários, optei também por criar o arquivo no sudoers.d (também não esqueci de comentar a linha do `secure_path` no sudoers)

Fontes:

<https://www.todoespacoonline.com/w/2015/10/su-sudo-e-sudoers-no-linux/>

<https://www.hostinger.com.br/tutoriais/arquivo-sudoers>

<https://www.digitalocean.com/community/tutorials/how-to-edit-the-sudoers-file-pt>

• Sistema de Controle de Acesso Obrigatório (MAC)

→ AppArmor

O AppArmor é um sistema de controle de acesso obrigatório pelo qual o sistema operacional ou o banco de dados restringe a capacidade do sujeito ou iniciador de acessar ou executar certos objetos ou alvos no sistema, sendo esses objetos arquivos, diretórios, portas TCP/UDP, seguimentos de memória compartilhada, dispositivos e etc.

O AppArmor utiliza o path do objeto para identificar ele e aplicar as restrições configuradas, isso significa que hard-links referenciando o objeto podem não ser sujeitos as mesmas restrições que existem pro objeto.

Nos sistemas mas atualizados do Debian, o AppArmor vem instalado por padrão, mas podemos instalá-lo utilizando o gerenciador de pacotes de sua escolha

```
sudo apt-get install apparmor
```

Pra listar todos os perfis e status do AppArmor carregados pros apps e processos utilize

```
sudo apparmor-status ou sudo aa-status
```

Pra ativar/desativar o AppArmor pro sistema inteiro utilize (ele precisa estar ativo)

sudo systemctl enable/disable apparmor

Pras mudanças no AppArmor serem aplicadas, o sistema precisa ser reiniciado

Fontes:

https://en.wikipedia.org/wiki/Mandatory_access_control

<https://en.wikipedia.org/wiki/AppArmor>

<https://wiki.debian.org/AppArmor/HowToUse>

https://linuxhint.com/debian_apparmor_tutorial/

<https://goto-linux.com/pt/2019/7/11/como-desativar-o-apparmor-no-ubuntu-20.04-focal-fossa-linux/>

- **Firewall**

→ UFW

O UFW é uma ferramenta de configuração de firewall, que é um dispositivo de segurança de rede que cuida do tráfego de informações bloqueando ou permitindo passagens de dados dependendo das regras configuradas.

Utilize o gerenciador de pacotes selecionado para instalar o UFW

sudo apt-get install ufw

O comando abaixo pode ser utilizado para verificar o status do UFW

sudo ufw status verbose

Pra ativar/desativar o UFW utilize os comandos (ele deve ser ativado)

sudo ufw enable/disable

Pra permitir/bloquear a entrada em todas as portas utilize (o subject pede pra deixar só a porta 4242 aberta, logo bloqueei tudo)

sudo ufw default allow/deny incoming

Pra permitir/bloquear a saída em todas as portas utilize (eu recomendo não alterar o padrão já definido, eu bloqueei a saída dos bagulho e meu gerenciador de pacotes parou de funcionar, então não recomendo mexer)

sudo ufw default allow/deny outgoing

Pra permitir/bloquear a conexão em uma porta especifica utilize (o subject pede pra habilitar somente a porta 4242)

sudo ufw allow/deny <número da porta>

Ao habilitar ou desabilitar alguma porta ou algo do tipo, você acaba criando uma regra pro UFW seguir, o comando “***sudo ufw status verbose***” além de mostrar o status do firewall também mostra as regras ativas, pra deletar uma regra ativa utilize

sudo ufw status numbered → Mostra as regras com seu número do lado

sudo ufw delete <número da regra> → Deleta a regra

Fontes:

https://www.cisco.com/c/pt_br/products/security/firewalls/what-is-a-firewall.html

<https://www.tecmint.com/setup-ufw-firewall-on-ubuntu-and-debian/>

<https://wiki.debian.org/Uncomplicated%20Firewall%20%28ufw%29>

- **Conexão SSH**

→ SSH

O SSH é um protocolo utilizado pra troca de dados entre cliente e servidor remoto de forma segura e dinâmica. Ele possibilita a comunicação criptografada através da rede permitindo acessar e fazer alterações em outro computador através do terminal

Utilize o gerenciador de pacotes selecionado pra instalar o servidor e o cliente SSH

sudo apt-get install openssh-server openssh-client

Pra verificar o status do serviço SSH utilize o comando

sudo service ssh status

Pra ativar/desativar o serviço SSH utilize os comandos (ele precisa estar ativado)

sudo service ssh start/stop

Pra mudar a porta padrão do servidor SSH e desabilitar o acesso ao root é preciso editar o arquivo **/etc/ssh/sshd_config** (brother, não confunde com o arquivo /etc/ssh/ssh_config, fiquei meia hora tentando descobrir qual era o problema. Utilize também o editor de sua preferência, eu escolhi o nano)

nano /etc/ssh/sshd_config

Altere a linha 13 pra SSH funcionar somente pela porta 4242 (obrigatório pelo subject)

```
# OpenSSH is to speed up the
# possible, but leave it
# default value.


#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

→

```
# OpenSSH is to speed up the
# possible, but leave it
# default value.

Port 4242
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Altere a linha 32 pra negar o acesso ao root pelo SSH (obrigatório pelo subject)



```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

 →

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Após realizar as mudanças, reset o serviço SSH pra aplicá-las utilizando (dá um reboot no sistema também só pra garantir)

sudo service ssh restart

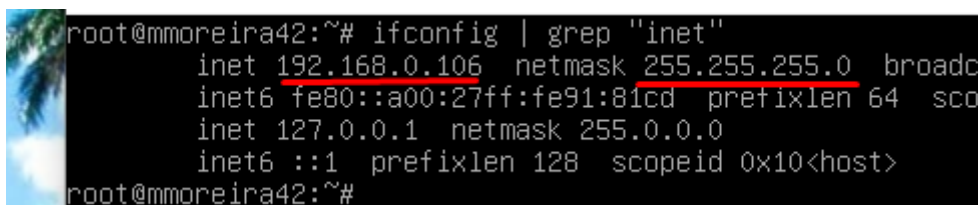
Mesmo após esse procedimento todo, a saída do comando pra listar as conexões ativas não fica idêntica à do subject devido a uma conexão UDP que aparece, pra resolver esse problema é preciso mudar as configurações do ip de DHCP pra Estático. Pra realizar essa mudança é necessário editar o arquivo `/etc/network/interfaces` ou criar um arquivo com as alterações no diretório `/etc/network/interfaces.d/` (mesmo esquema da configuração do sudo). Antes de fazer isso você precisa ter em mãos o ip, a netmask e o gateway da sua máquina. (pra realizar essas consultas, eu utilizei comandos do pacote net-tools, logo vou seguir com esse modo).

Utilize o seu gerenciador de pacotes para instalar o net-tools

sudo apt-get install net-tools

Agora pra pegar o ip e a netmask, olhe a primeira linha exibida no terminal ao utilizar

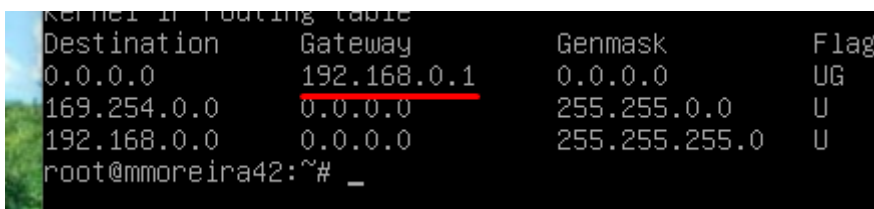
sudo ifconfig | grep "inet"



```
root@mmoreira42:~# ifconfig | grep "inet"
inet 192.168.0.106 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::a00:27ff:fe91:81cd prefixlen 64 scopeid 0x20<eth>
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
root@mmoreira42:~#
```

Pra pegar o gateway, olhe a segunda coluna da segunda linha exibida no terminal ao utilizar

sudo route -n



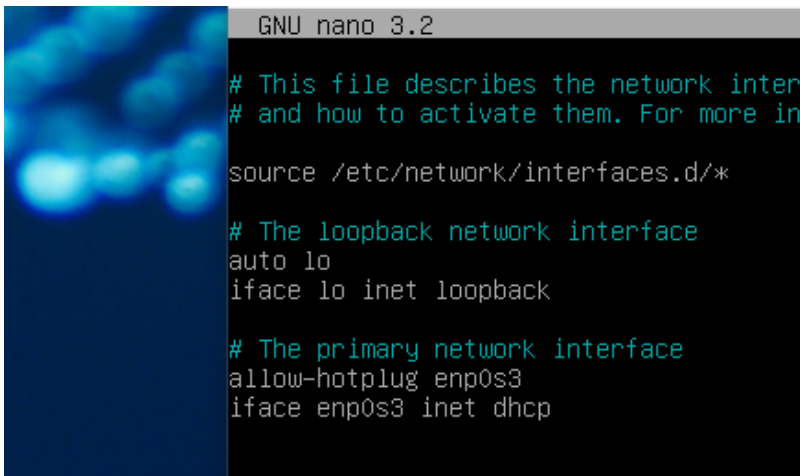
```
Kernel IP routing table
Destination Gateway Genmask Flag
0.0.0.0 192.168.0.1 0.0.0.0 UG
169.254.0.0 0.0.0.0 255.255.0.0 U
192.168.0.0 0.0.0.0 255.255.255.0 U
root@mmoreira42:~#
```

Com essas informações na mão abra o arquivo `/etc/network/interfaces` ou crie um arquivo no diretório `/etc/network/interfaces.d` e faça as seguintes alterações. Na última linha altere o ip de DHCP pra static e logo abaixo, insira suas informações com a seguinte formatação

```
iface enp0s3 inet static
    address <seu ip>
    netmask <sua mascara>
    gateway <seu gateway>
```

Pra também configurar o ipv6 adicione a linha (caso ela já não esteja lá)

```
iface enp0s3 inet6 auto
```

A screenshot of a terminal window showing the original content of the `/etc/network/interfaces` file. The terminal title is "GNU nano 3.2". The file content includes comments about network interfaces, a source line for `/etc/network/interfaces.d/*`, and configurations for the loopback interface `lo` and the primary interface `enp0s3` using DHCP.

```
GNU nano 3.2

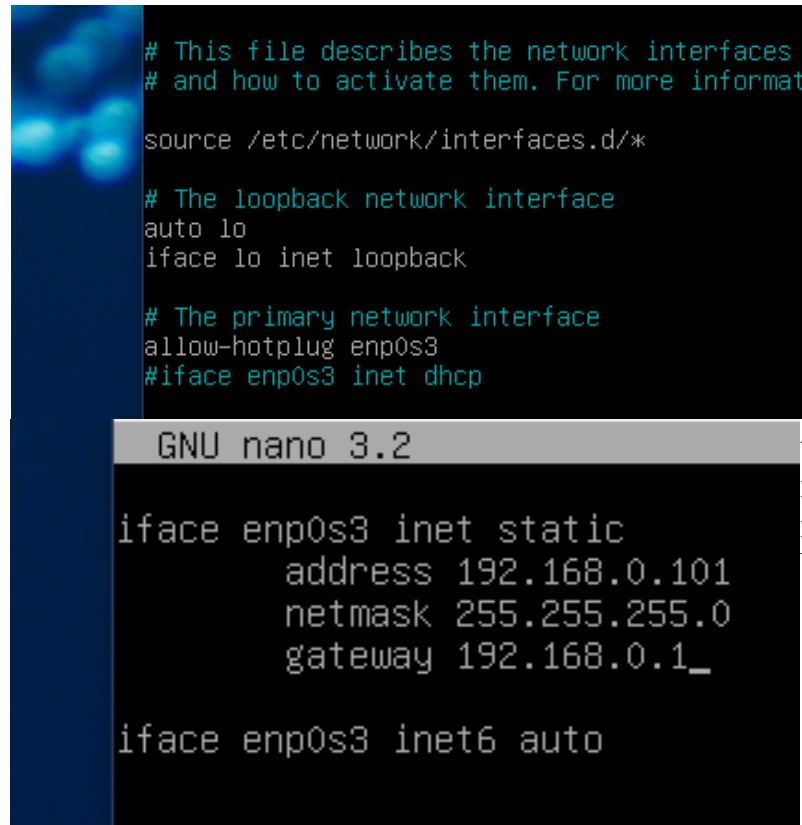
# This file describes the network inter
# and how to activate them. For more in

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
```

Original

A screenshot of a terminal window showing the modified content of the `/etc/network/interfaces` file. The terminal title is "GNU nano 3.2". The original DHCP configuration for `enp0s3` has been commented out, and a new static configuration with specific IP, netmask, and gateway has been added. The `inet6 auto` configuration has also been added.

```
GNU nano 3.2

# This file describes the network interfaces
# and how to activate them. For more informat

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

iface enp0s3 inet static
    address 192.168.0.101
    netmask 255.255.255.0
    gateway 192.168.0.1_

iface enp0s3 inet6 auto
```

Depois das alterações (eu optei por comentar a linha no arquivo original, criar um arquivo no diretório `interfaces.d` e editá-lo. O tab nos parâmetros de rede não é necessário, eu botei porque os sites mostravam assim e porque fica bonitinho)

Verifique com o comando **ss -tunlp** se as conexões estão iguais à do subject

```
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
tcp LISTEN 0 128 0.0.0.0:4242 0.0.0.0:* users:(("sshd",pid=523,fd=3))
tcp LISTEN 0 128 :::4242 :::* users:(("sshd",pid=523,fd=4))
```

Depois de realizar todos esses procedimentos resta testar a conexão SSH

De um terminal fora da VM (essa simplicidade é pro Linux ou o WSL do Windows. No terminal do Windows vai precisar instalar o PuTTY e no Mac eu nem faço ideia se precisa instalar algum cliente SSH), você vai utilizar o comando SSH com a seguinte estrutura (faça todos os testes como conectar no seu usuário da intra, no usuário root, conectar em uma porta que não seja a 4242 e essas parada toda)

ssh [usuário da VM a ser logado]@[ip da sua VM] -p [porta da conexão]

Pra enviar arquivos pro servidor através da SSH utilize a seguinte estrutura

scp -P [porta da conexão] [arquivo] [usuário da VM a ser logado]@[ip da sua VM]:[diretório que vai receber o arquivo]

Pra sair da conexão SSH utilize **logout** ou **exit**

Fontes:

Tuco e Tuca (novamente brabos demais)

<https://rockcontent.com/br/blog/ssh/>

<https://www.dialhost.com.br/blog/acesso-ssh/>

<https://bonino.com.br/configurar-ssh-servidor-linux-debian/#inicia-servidor>

<https://devconnected.com/how-to-install-and-enable-ssh-server-on-debian-10/>

<https://linuxconfig.org/how-to-setup-a-static-ip-address-on-debian-linux>

https://wiki.debian.org/pt_BR/NetworkConfiguration

<https://www.hostinger.com.br/tutoriais/usar-comando-scp-linux-para-transferir-arquivos>

• Configurando Política de Senha Forte

A implementação da política de senha forte é feita em 2 partes, a parte de ajuste dos temporalizadores das senhas e a parte de restrição na criação/alteração das senhas

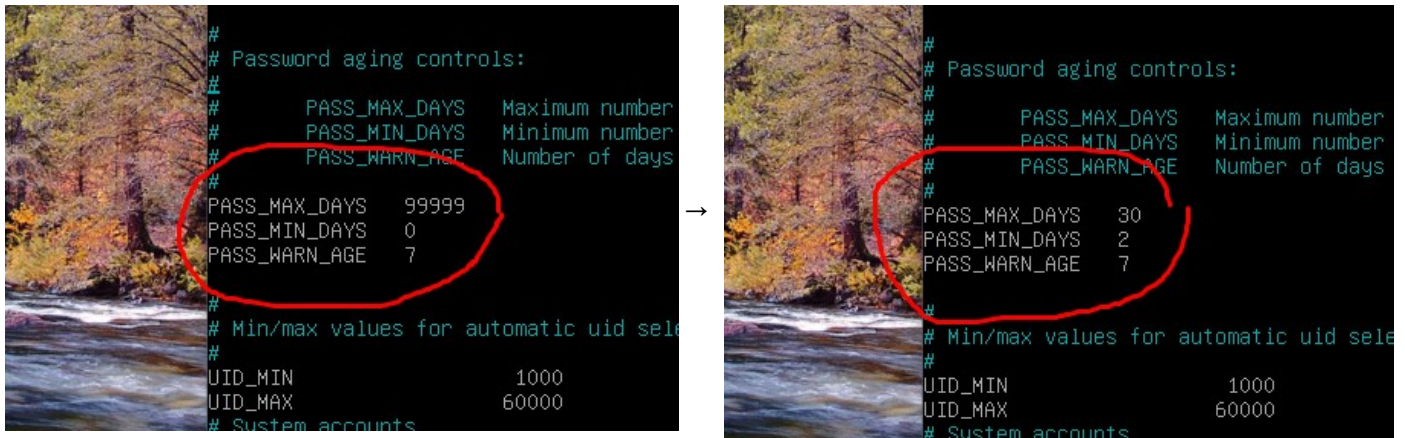
A primeira parte consiste na alteração do arquivo **/etc/login.defs** em 3 linhas diferentes. Elas representam as seguintes configurações

PASS_MAX_DAYS = total de dias até a senha inspirar

PASS_MIN_DAYS = mínimo de dias necessários pra poder realizar outra alteração na senha

PASS_WARN_AGE = número de dias antes da senha expirar pra alertar que vai expirar

Você vai realizar as alterações pros valores pedidos no subject dessa forma



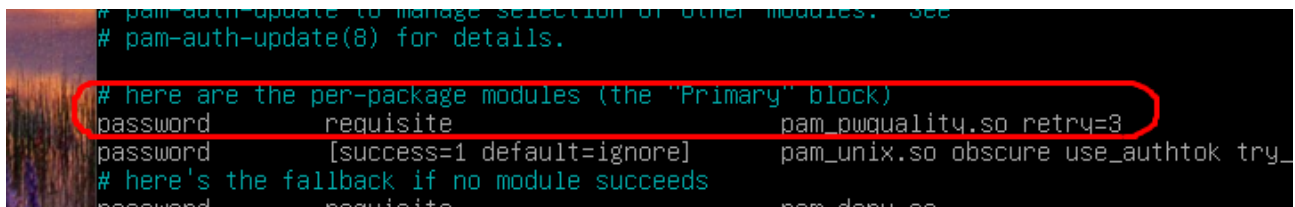
```
# Password aging controls:
#
# PASS_MAX_DAYS Maximum number
# PASS_MIN_DAYS Minimum number
# PASS_WARN_AGE Number of days
#
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_WARN_AGE 7
#
# Min/max values for automatic uid selection
#
UID_MIN 1000
UID_MAX 60000
# System accounts
```

```
# Password aging controls:
#
# PASS_MAX_DAYS Maximum number
# PASS_MIN_DAYS Minimum number
# PASS_WARN_AGE Number of days
#
PASS_MAX_DAYS 30
PASS_MIN_DAYS 2
PASS_WARN_AGE 7
#
# Min/max values for automatic uid selection
#
UID_MIN 1000
UID_MAX 60000
# System accounts
```

Pra segunda parte é necessário instalar um pacote chamado `libpam-pwquality`, com ele teremos as ferramentas necessárias pra realizar as restrições

`sudo apt-get install libpam-pwquality`

Após instalá-lo, nós teremos um novo arquivo que será responsável por implementar as configurações que nós desejamos, ele será o `/etc/pam.d/common-password`, pra editar ele precisamos alterar a linha abaixo



```
# here are the per-package modules (the "Primary" block)
password requisite pam_pwquality.so retry=3
password [success=1 default=ignore] pam_unix.so obscure use_authtok try_
# here's the fallback if no module succeeds
password requisite pam_deny.so
```

A alteração vai ser feita acrescentando os seguintes parâmetros

`minlen=10` → Restringe a senha a ter no mínimo 10 caracteres.

`ucrcdit = -1` → Restringe a senha a ter no mínimo um carácter maiúsculo.

`dcrcdit = -1` → Restringe a senha a ter no mínimo um carácter numérico.

`maxrepeat = 3` → Restringe a senha a ter no máximo 3 caracteres consecutivos

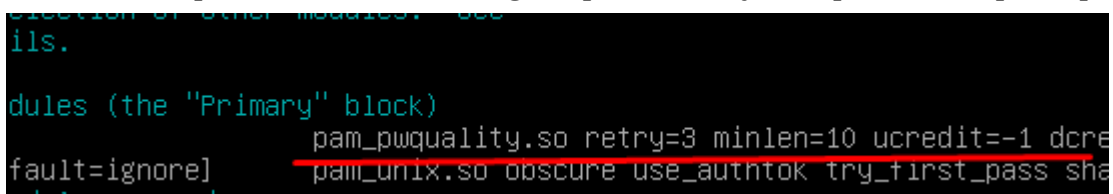
idênticos.

`reject_username` → Restringe a senha a não ter o nome do usuário.

`difok = 7` → Restringe a senha a ter no mínimo 7 caracteres que não fazem parte da senha antiga.

`enforce_for_root` → Aplica as restrições de criação de senha até para quando o root configura a senha.

O acréscimo desses parâmetros deve ser logo depois do `retry=3` separando-os por espaço



```
modules (the "Primary" block)
password requisite pam_pwquality.so retry=3 minlen=10 ucrcdit=-1 dcrcdit=-1 maxrepeat=3 reject_username difok=7 enforce_for_root pam_unix.so obscure use_authtok try_first_pass sha
password [success=1 default=ignore]
# here's the fallback if no module succeeds
```

Após isso a política de senha forte já está implementada, porém as alterações feitas não serão aplicadas automaticamente aos usuários já existentes. Como esses usuários também precisam estar na política de senha forte, você terá que aplicar manualmente neles

Pra modificar as configurações da primeira parte da senha utilize os códigos

chage -M 30 <usuário> → Número de dias pra senha expirar

chage -m 2 <usuário> → Número de dias mínimos pra trocar senha

chage -W 7 <usuário> → Número de dias pra mensagem de senha expirando

chage -l <usuário> → Mostra as configurações que estão setadas pra esse usuário

Pra modificar as configurações da segunda parte das senhas fortes só é necessário alterar a senha do usuário e seguir o que as restrições impõem. Pra alterar a senha do usuário utilizei ***passwd <usuário>***

Fontes:

Tuco e Tuca (olha eles ai de novo)

<https://tiparaleigo.wordpress.com/2020/08/13/como-aplicar-politicas-de-senha-no-linux-ubuntu-centos/>

<https://www.cyberciti.biz/faq/securing-passwords-libpam-cracklib-on-debian-ubuntu-linux/>

https://linux.die.net/man/8/pam_pwquality

• **Hostname, Usuários e Grupos**

É preciso alterar seu hostname pro login na intra + 42 (mesmo se ele já estiver assim, aprenda como fazer pra avaliação), logo utilize e aprenda os comandos abaixo (depois de mudar o hostname eu dava um reboot pra garantir)

hostnamectl status → mostra o hostname atual com mais algumas informações

hostnamectl set-hostname <novo hostname> → altera o hostname atual

É preciso ter no mínimo um usuário root e um com seu login na intra na máquina, logo utilize e estude os comandos abaixo pra essa tarefa.

sudo awk -F": " '{print \$1}' /etc/passwd → mostra só a primeira coluna do arquivo /etc/passwd. Essa coluna contem os usuários existentes na máquina

adduser <usuário> → cria um novo usuário com home, recolhe informações do novo usuário como nome completo e aquelas coisa, completinho esse bagulho

useradd -m <usuário> → cria um novo usuário somente, com o parâmetro -m cria uma home pra esse usuário também

userdel -r <usuário> → deleta o usuário, com o parâmetro -r deleta todos os arquivos vinculados a esse usuário, inclusive a home dele

usermod -l <novo nome usuário> <usuário> → altera o nome de usuário

usermod -g <grupo(nome ou número)> <nome> → altera o GID do grupo principal do usuário utilizando ou o nome do novo grupo ou o GID

id -u <usuário> → mostra o UID do usuário.

users → mostra os usuários conectados no sistema naquele momento

É preciso ter o usuário com seu login na linha inserido nos grupos sudo e user42, logo utilize e estude os comandos abaixo pra essa tarefa.

`sudo awk -F":" '{print $1}' /etc/group` → mostra só a primeira coluna do arquivo /etc/group. Essa coluna contém os grupos existentes na máquina

`groupadd <grupo>` → cria um grupo

`groupdel <grupo> --` deleta o grupo

`gpasswd -a <usuário> <grupo>` → adiciona o usuário ao grupo

`gpasswd -d <usuário> <grupo>` → remove o usuário do grupo

`groups <usuário>` → mostra os grupos que esse usuário está

`getent group <grupo>` → mostra os usuários que estão nesse grupo

`id -g <usuário> --` mostra o GID do grupo principal do usuário

Fontes:

<https://thomasdiego.com/como-alterar-nome-da-maquina-hostname-no-linux/>

<https://www.infowester.com/usuarioslinux.php>

<https://e-tinet.com/linux/gerenciar-usuarios-linux/>

<https://www.vivaolinux.com.br/dica/Adicionando-ou-excluindo-um-usuario-de-um-grupo>

<https://www.ppgia.pucpr.br/pt/arquivos/techdocs/linux/foca-intermediario/ch-cmdc.html>

• Cron e Wall

→ Cron

O Cron é um programa que executa em segundo plano na máquina, ele é utilizado pra realizar tarefas recorrentes através de um controlador de tempo. Com ele é possível agendar comandos pra serem dados em horários específicos em datas específicas de forma repetida, os chamados cron jobs.(é a forma mais prática pra executar seu script a cada 10m como manda o subject)

Utilize o seu gerenciador de pacotes pra instalar o Cron

`sudo apt-get install cron`

O Cron utiliza os arquivos Crontab pra agendar as tarefas que serão dadas. Cada usuário pode possuir um Crontab próprio e eles são simples de editar. Pra editar um arquivo Crontab utilize

`crontab -e` (na primeira execução do comando, vai aparecer uma tela perguntando qual editor de texto você vai querer deixar como padrão pra editar o Crontab, digite o número equivalente ao editor de sua preferência)

Pra agendar um comando no Cron nós utilizamos a sintaxe

MI H D ME DS <comando>

Essas siglas representam

MI = **Minuto** → (varia de 00 a 59)

H = **Hora** → (varia de 0 a 23)

D = **Dia do Mês** → (varia de 1 a 31)

ME = **Mês** → (varia de 1 a 12)

DS = **Dia da Semana** → (varia de 0 a 7, onde domingo é representado tanto por 0 quanto por 7)

Junto a essa sintaxe nós temos os caracteres utilizados pra flexibilizar o agendamento de eventos

Asterisco = * → na sintaxe do Cron representa “todo”. Ex: * * * * * <comando> (executa o comando em todos os minutos, em todas as horas, em todos os dias...)

Virgula = , → na sintaxe do Cron é utilizada pra por mais de um parâmetro em um dos argumentos. Ex: 25 * 10,15 * * <comando> (executa o comando no vigésimo quinto minuto de todas as horas dos dias 10 e 15)

Hifen = - → na sintaxe do Cron é utilizado pra por um intervalo entre valores nos argumentos. Ex: 15 10 * 2-5 * <comando> (executa o comando as 10:15 todos os dias de fevereiro até maio)

Barra = / → na sintaxe do cron é utilizada pra criar intervalos pré-determinados de tempos. Ex 30 */3 * * * * <comando> (executar o comando no trigésimo minuto a cada 3 horas todos os dias)

Por padrão o Cron executa o comando no segundo 0 dos minutos definidos no agendamento da tarefa, mas você pode alterar esse padrão utilizando o sleep com a seguinte sintaxe.

Ex: 10 * * * * sleep 45 ; <comando> (executar o comando no quadragésimo quinto segundo do décimo minuto de todas as horas de todos os dias ...)

Assim caso você queira criar uma tarefa que execute a cada 20 segundos você pode definir 3 rotinas seguidas nos segundos 0(padrão), 20 e 40 pra executar em todos os minutos. Ex:

* * * * * <comando>

* * * * * sleep 20 ; <comando>

* * * * * sleep 40 ; <comando>

Alguns desses comandos ainda podem ser abreviados por algumas sintaxes especiais, como

@hourly <comando> = 0 * * * * <comando> (uma vez em todas as horas)

@daily <comando> = 0 0 * * * <comando> (uma vez em todos os dias)

@weekly <comando> = 0 0 * * 0 <comando> (uma vez em toda semana)

@monthly <comando> = 0 0 1 * * <comando> (uma vez em todo mês)

@yearly <comando> = 0 0 1 1 * <comando> (uma vez em todo ano)

Outros comandos necessários pra edição e manipulação dos cron job são

crontab -r → deleta o crontab do usuário logado

crontab -l → exibe o conteúdo do crontab do usuário logado

crontab -u <usuario> -e ou **-r** ou **-l** → permite fazer ações com o crontab de outro usuário

/etc/init.d/cron start/stop/status → ativa/desativa o cron ou mostra seu status (as alterações de ativado ou desativado do cron só entram em vigor no minuto depois do comando ser dado. O cron também sempre inicializa ativo quando o sistema liga, logo se você der um **/etc/init.d/cron stop** e der um **reboot**, ele ativara de novo. Assim caso você queira fazer o script parar de executado mesmo depois do **reboot**, você precisa comentar a linha que executam o script)

Caso você va executar um script com o Cron, ponha o path completo do script, porque o Cron é complicado **Ex: * * * * * bash /home/user/arqu/monitoring.sh**

→ Comando wall

O wall exibe uma mensagem (como o echo) ou o conteúdo de um arquivo (como o cat) no terminal de todos os usuários conectados na máquina naquele momento. O comando quebrará as linhas com mais de 79 caracteres e preenche com espaços em branco as linhas com menos de 79 caracteres. No final da exibição o comando quebra linha e põe o espaço pra inserir novos textos no terminal. (o subject quer que o script exiba as informações em todos os terminais, logo o wall é a boa pra isso)

A primeira linha escrita na exibição é o banner (o “broadcast messa...”). Utilizando a flag -n, o banner é suprimido

wall -n <entrada>

Fontes:

Tomás, Tuco e Tuco (esses cara não para)

<https://www.hostinger.com.br/tutoriais/cron-job-guia#O-Que-e-Cron-Job>

<https://www.digitalocean.com/community/tutorials/how-to-use-cron-to-automate-tasks-ubuntu-1804-pt>

<https://rafaelbiriba.com/2010/08/01/crontab-rodando-um-script-a-cada-15-segundos.html>

<https://www.todoespacoonline.com/w/2015/11/cron-e-crontab-no-linux/>

<https://man7.org/linux/man-pages/man1/wall.1.html>

- **Script**

A forma como será construída o script é pessoal e cada um faz da forma que achar melhor. Porém ele precisa seguir os seguintes critérios

Ele precisa ser em bash

Precisa ter o nome monitoring.sh

Deve ser exibido em todos os terminais a cada 10 minutos

Uma forma interessante de construir o script é escrever as informações necessárias em um arquivo temporário e depois utilizar o wall pra exibir o conteúdo desse arquivo

Comandos uteis pra construção do script.

uname -a → mostra a arquitetura e o kernel da máquina

grep 'physical id' /proc/cpuinfo | uniq | wc -l → conta o número de CPUs

grep 'processor' /proc/cpuinfo | uniq | wc -l → conta o número de vCPUs

free -m → mostra informações da memória RAM (flag -m mostra em MB)
df -h/m --total → mostra informações do disco (flag -m em MB, -h em GB)
mpstat → mostra porcentagem dos processos (necessita do pacote sysstat)
who -b → mostra a data e horário do último boot no sistema
if [\$(blkid | grep -c '/dev/mapper') -eq 0]; then echo "no"; else echo "yes"; fi → uma expressão boa pra verificar se existem partições utilizando lvm
ss -s | awk '/TCP:/ {print \$2} → mostra o número de conexões TCP
who → mostra os usuários logados
hostname -I → somente número do ip
ifconfig | awk '/ether/ {print \$2}' → número MAC
grep -c 'COMMAND' /var/log/sudo/<arquivo que tu guardo os log> → conta o número de ações feitas com sudo

Em determinado momento o cronjob começou a não ler os comandos **ifconfig** e **blkid** do meu script, parece que os comandos do diretório **/usr/sbin** (que só o root e o sudo tem acesso) param de funcionar mesmo se o crontab for do root, eu resolvi o problema utilizando o caminho todo do comando, no caso seria **/usr/sbin/ifconfig** e **/usr/sbin/blkid**

- **Entrega do Projeto**

Apos finalizar todas as instruções desse texto, verifique com o subject se foi tudo atendido e desligue a máquina virtual.

Va até o diretório onde a VirtualBox guarda as VMs

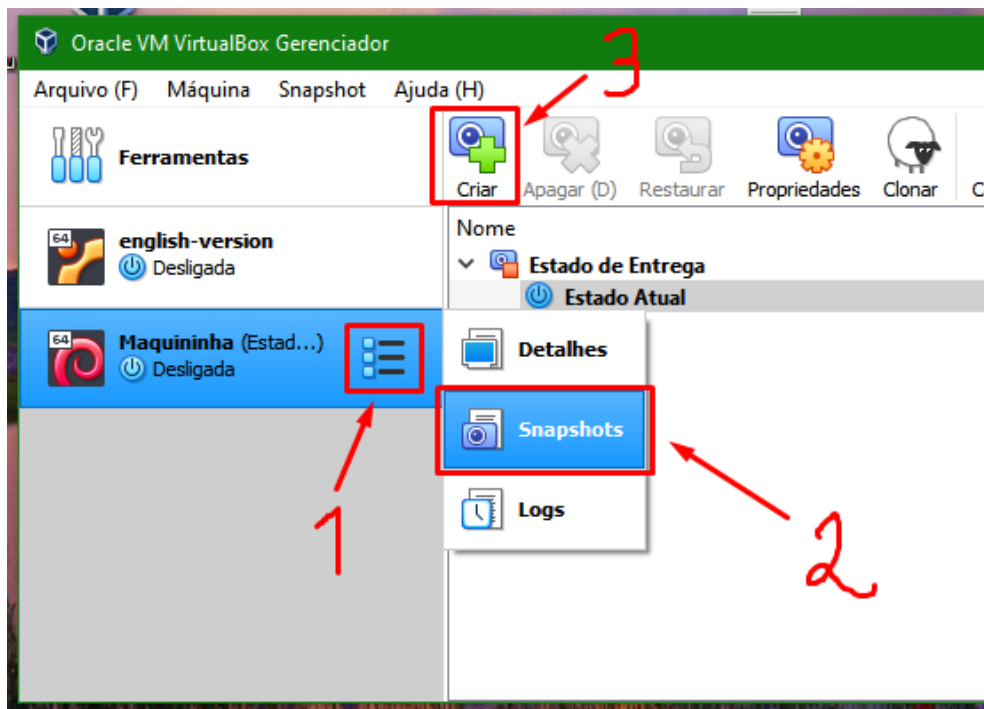
Windows: %HOMEDRIVE%%HOMEPATH%\VirtualBox VMs\
Linux: ~/VirtualBox VMs/
MacM1: ~/Library/Containers/com.utmapp.UTM/Data/Documents/
MacOS: ~/VirtualBox VMs/

Abra o terminal no diretório da VM que você criou pro projeto e pegue a assinatura no formato sha1 do arquivo .vdi (ou ".qcow2" pra usuários UTM). Utilize os comandos abaixo feitos com o exemplo cent_serv.vdi (do subject) pra isso

Windows: **certUtil -hashfile centos_serv.vdi sha1**
Linux: **sha1sum centos_serv.vdi**
For Mac M1: **shasum Centos.utm/Images/disk-0.qcow2**
MacOS: **shasum centos_serv.vdi**

Um exemplo de resultado dessa operação é **6e657c4619944be17df3c31faa030c25e43e40af**

Toda vez que alguma alteração é feita na VM a assinatura muda, logo caso você faça alterações na primeira avaliação que você fizer, pra segunda avaliação a assinatura no seu computador não vai bater com a assinatura no repositório, pra resolver isso você pode tirar um snapshot do estado da VM antes da entrega Depois de criar a snapshot use a sua VM e



Faça as alterações que quiser, porém quando encerrar, desligue a VM na parte gráfica da janela e marque a opção “Restaurar o snapshot”, assim nenhuma alteração vai ser salva e a assinatura não muda

