Relatório AP01 - Implementação de Administração de Sistemas e Serviços de Rede.

Alunos: Rafael dos Santos | Hermes Soares de Jesus | Fábio Mendes Crepaldi

1. Administre um Sistema GNU com Linux com as seguintes especificações:

1º) Debian ou derivados com instalação mínima (apenas com interface de texto)

2º) Além do usuário administrador root, adicione mais dois usuários ao sistema:

- um usuário com suporte administrativo através do sudo;
- um usuário sem suporte administrativo;

3º) Crie também dois grupos adicionais para dois tipos de usuários:

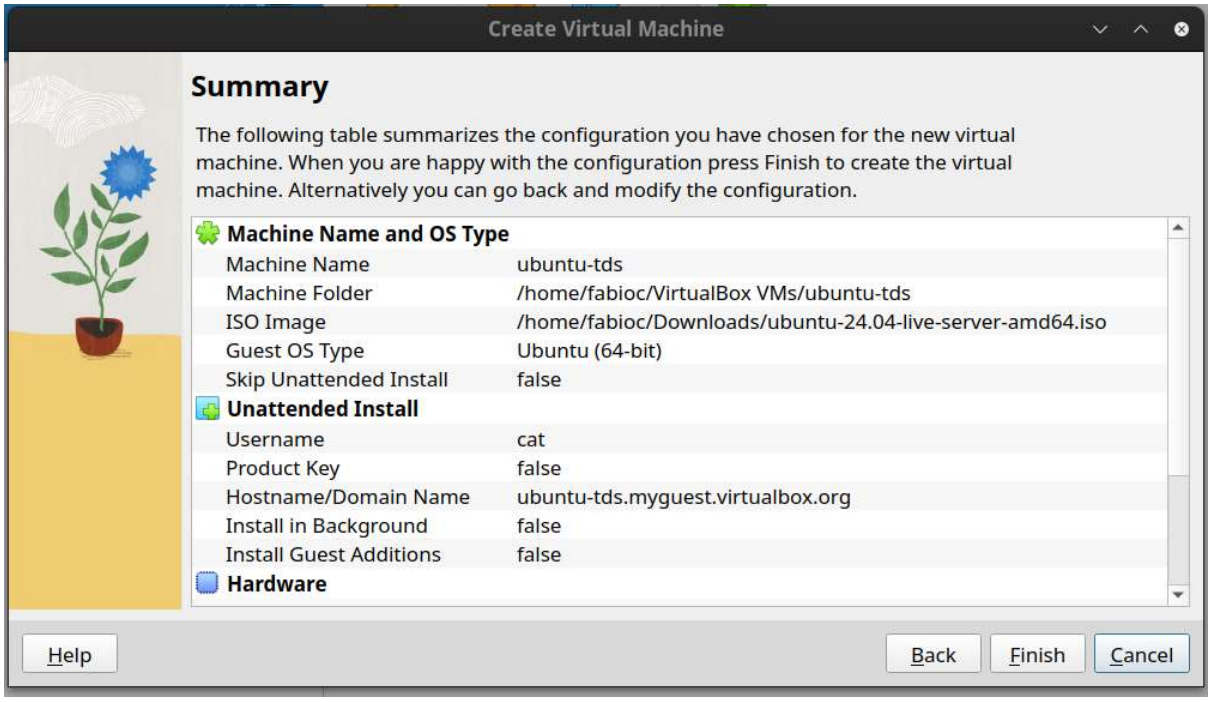
- um grupo chamado ‘admins’ para usuários com suporte administrativo;
- um grupo chamado ‘users’ para usuários sem suporte administrativo;
- configure os usuários adequadamente aos grupos criados;

4º) Configure uma quota de 500MB para os usuários sem suporte administrativo;

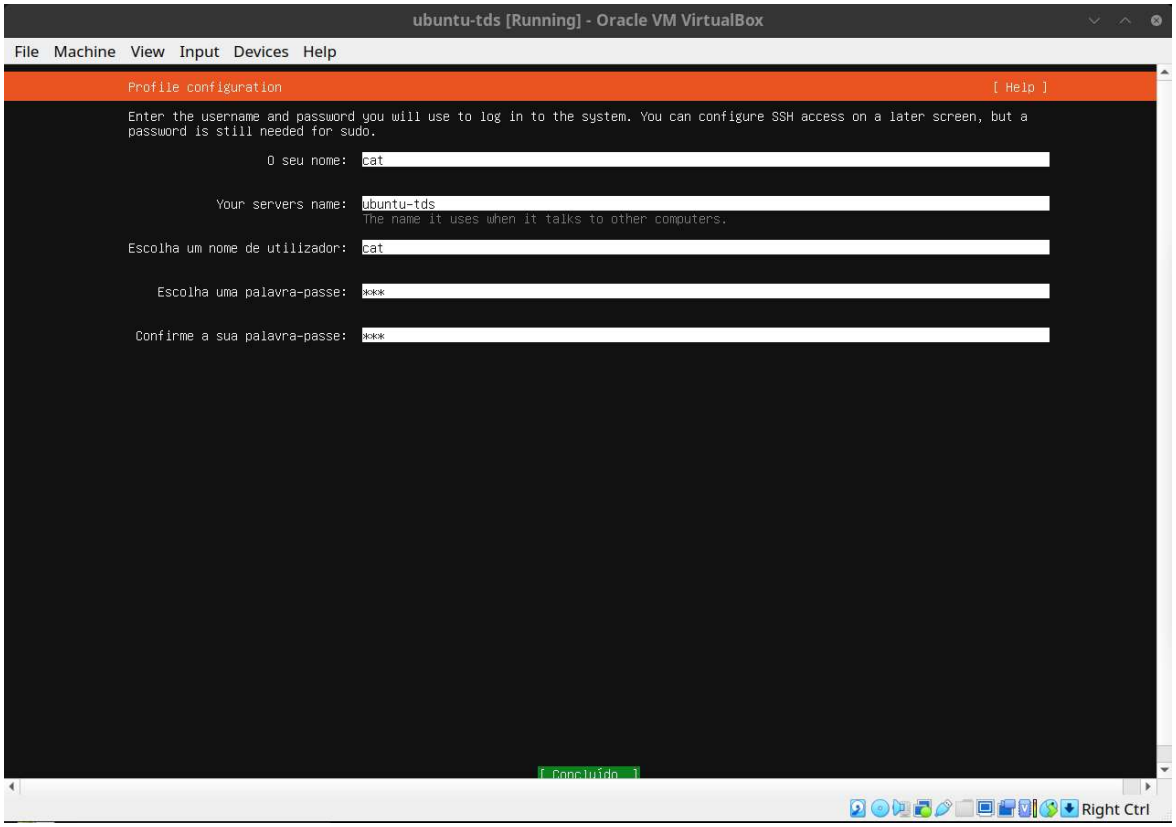
5º) Configure uma crontab para escrever a data diariamente em um arquivo date.txt na home do usuário sem suporte administrativo.

1º) Inicialmente instalamos uma máquina virtual utilizando uma distribuição Ubuntu Server (64-bits), com 4gb de RAM e 20gb de armazenamento interno reservados. Para a instalação da máquina virtual utilizamos a interface Oracle VM Virtual Box.

Nomeamos a máquina virtual como “ubuntu-tds” e usuário padrão como “cat”.



Como nome do servidor, reutilizamos a nomenclatura ”ubuntu-tds” e o usuário ”cat”



Depois de instalado a distribuição Ubuntu e logado no sistema, mudamos a senha do root para ”root” utilizando o comando:

```
# passwd root
```

2º) Criamos um usuário chamado “dog” com o seguinte comando:

```
# useradd –m dog
```

E trocamos a senha do usuário dog para “dog”, utilizando o comando:

```
# passwd dog
```

3º) Foi criado o grupo “admins” e foi realizado uma tentativa de criação de um grupo “users”, porém esbarramos em dificuldades pois já existia um grupo com a denominação “users”. Em seguida adicionamos o usuário “cat” para o grupo de admins. Utilizando o comando:

```
# usermod –aG admins cat
```

E para o usuário "dog", adicionamos o mesmo para o grupo “users” já criado pela distribuição. Para isso repetimos o comando anterior:

```
# usermod –aG users dog
```

Para dar permissão de sudo para o grupo “admins”, utilizamos o comando:

```
# EDITOR=vim sudo –e /etc/sudoers
```

E adicionamos a seguinte linha no terminal para a inclusão do mesmo:

```
# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults        use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
%admins ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
```

4º) Antes de configurar a quota para grupos, instalamos a quota na máquina utilizando o seguinte comando:

```
# apt install quota
```

Configuramos uma quota de espaço no disco para o grupo “users”

```
# vim /etc/fstab
```

```
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/e399364c-d1eb-404a-8124-6b2ae903f7a1 / ext4 defaults usrquota,grpquota 0 1
/swap.img       none    swap    sw      0       0
```

Para criar o arquivo quota de grupos utilizamos o comando:

```
# quotacheck -gm /
```

Para colocarmos uma quota no grupo “users” utilizamos o comando:

```
# EDITOR=vim edquota –g users
```

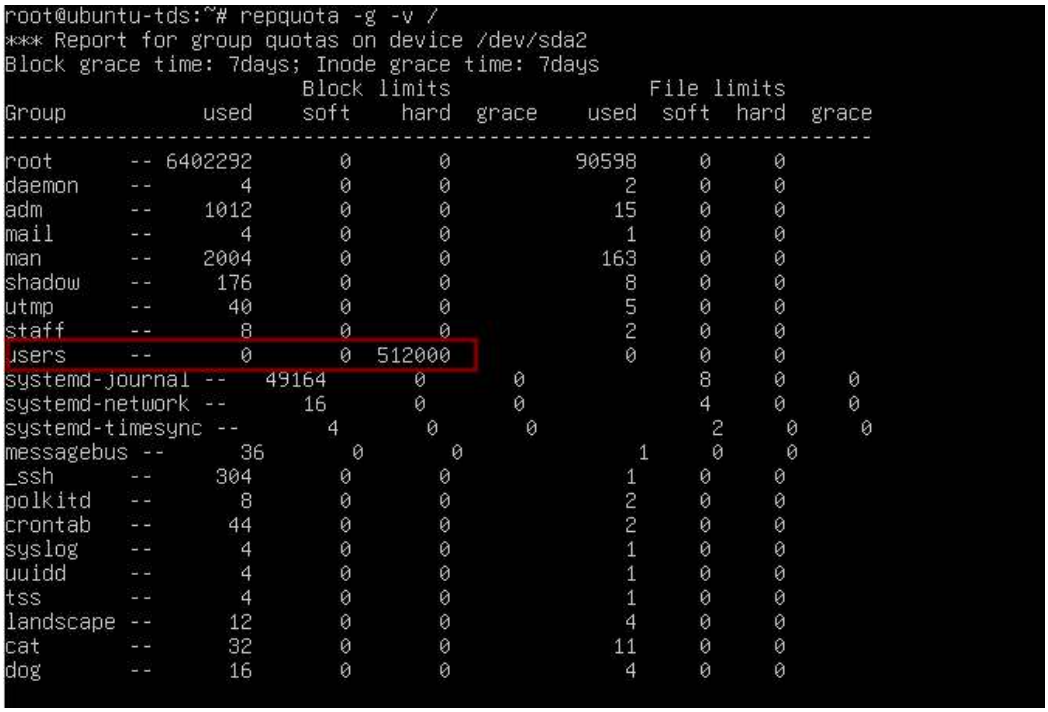
E definimos o valor máximo de utilização de disco para cada participante deste grupo, neste caso no tamanho de 500MB.

Em seguida habilitamos o sistema de quotas para o grupo “users”, pois ele não é linkado automaticamente. Para isto, utilizamos o comando:

```
# quotaon /
```

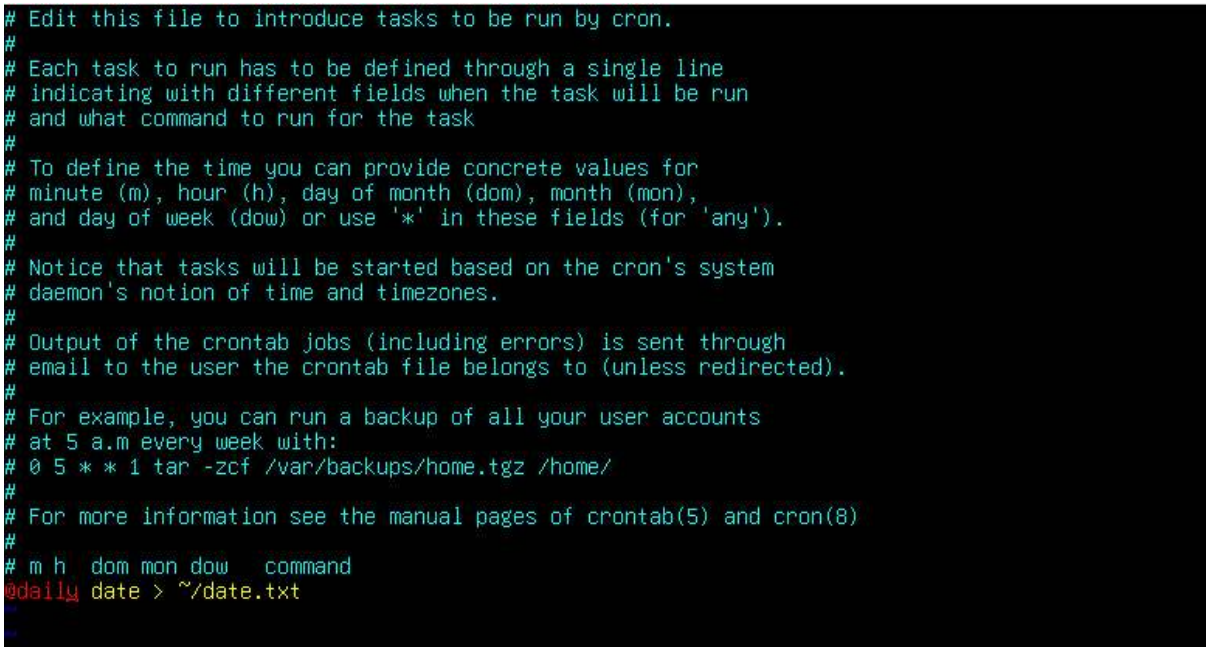
E para validarmos a existência da quota no grupo “users”, utilizamos o comando abaixo e notamos os 500MB reservados ao grupo:

```
# repquota –g –v /
```



5º) Agora partimos para a configuração do crontab para o usuário “dog”, pertencente ao grupo “users”, ou seja, sem permissão administrativa, para isto utilizamos o comando:

```
# EDITOR=vim crontab –u dog -e
```



Após a configuração do crontab no usuário “dog”, partimos para a obtenção e redirecionamento da data para o arquivo “date.txt”, feito de maneira automática e diária com a seguinte linha:

```
@daily date > ~/date.txt
```

Esta configuração é feita diretamente no arquivo do crontab.

2. Implemente serviços de rede web e de acesso remoto com as seguintes especificações:

1º) Servidor Web com Apache que suporte:

- PHP
- Virtual Host

2º) O Servidor Web deve hospedar dois sites:

- Um site HTML
- Um site PHP

3º) Servidor de Acesso Remoto com SSH

- Acesso por senha
- Acesso por chave

1º) Para instalar o servidor apache digitamos o seguinte comando no terminal:

```
# apt install apache2
```

E para configurar o suporte de php com o apache:

```
# apt install php libapache2-mod-php
```

Depois da instalação, foi realizado a reinicialização do servidor apache com o seguinte comando:

```
# service apache2 restart
```

Após a reinicialização, o apache está habilitado para interpretar scripts php.

2º) Um servidor default para HTML é configurado em /var/www/html/, com o nome index.html, como mostrado a na imagem a seguir, usaremos ele como parte do exercício:

E para a configuração do servidor PHP, criamos um diretório para hospedar a página PHP, utilizando o seguinte comando:

```
# mkdir /var/www/php
```

Com a pasta criada, podemos dar sequência criando o arquivo index.php, para isso, usamos o comando:

```
$ cd /var/www/php
```

```
# vim index.php
```

Com o seguinte conteúdo:

Em seguida, editamos o arquivo 000-default.conf, localizado no diretório:

/etc/apache2/sites-available/

Na edição do arquivo adicionamos essas linhas de texto:

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

<VirtualHost *:81>
        ServerName unioeste.br

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/php

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Para configurar o servidor (na porta 81) onde despejaremos um arquivo php. Além disso, editamos o arquivo /etc/apache2/mods-enabled/dir.conf para colocarmos prioridade sobre os arquivos php em cima dos de outro tipo:

vim /etc/apache2/mods-enabled/dir.conf

Original:

```
DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
```

Modificado:

```
DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
```

Com a prioridade definida, passamos a testar nosso servidor PHP que acabou de ser criado, primeiramente é necessário adicionarmos uma porta, para isso precisamos editar o seguinte arquivo:

vim /etc/apache2/ports.conf

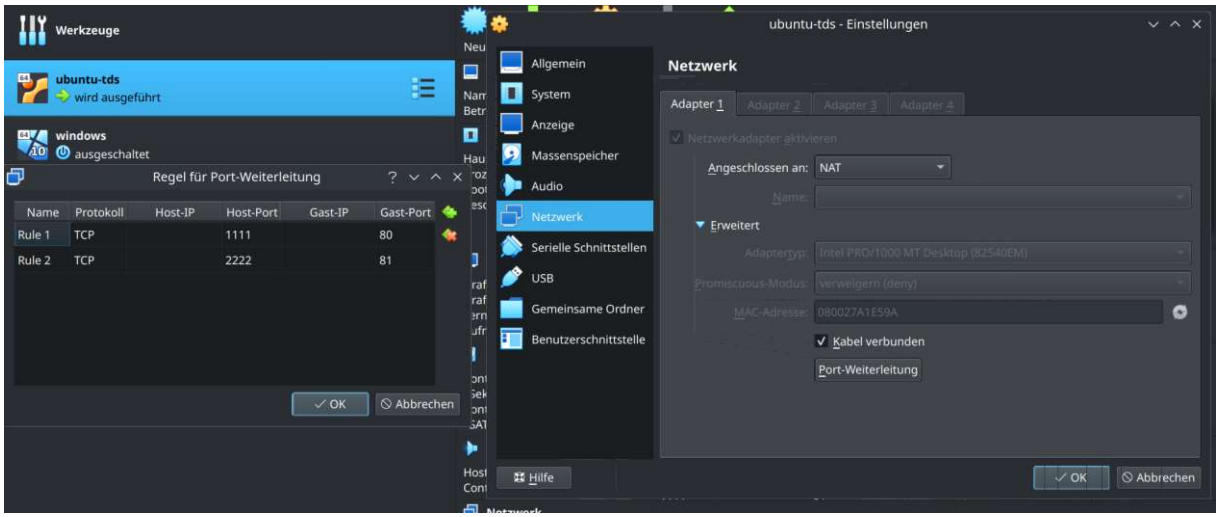
E adicionamos a seguinte linha:

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
Listen 81

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>
```

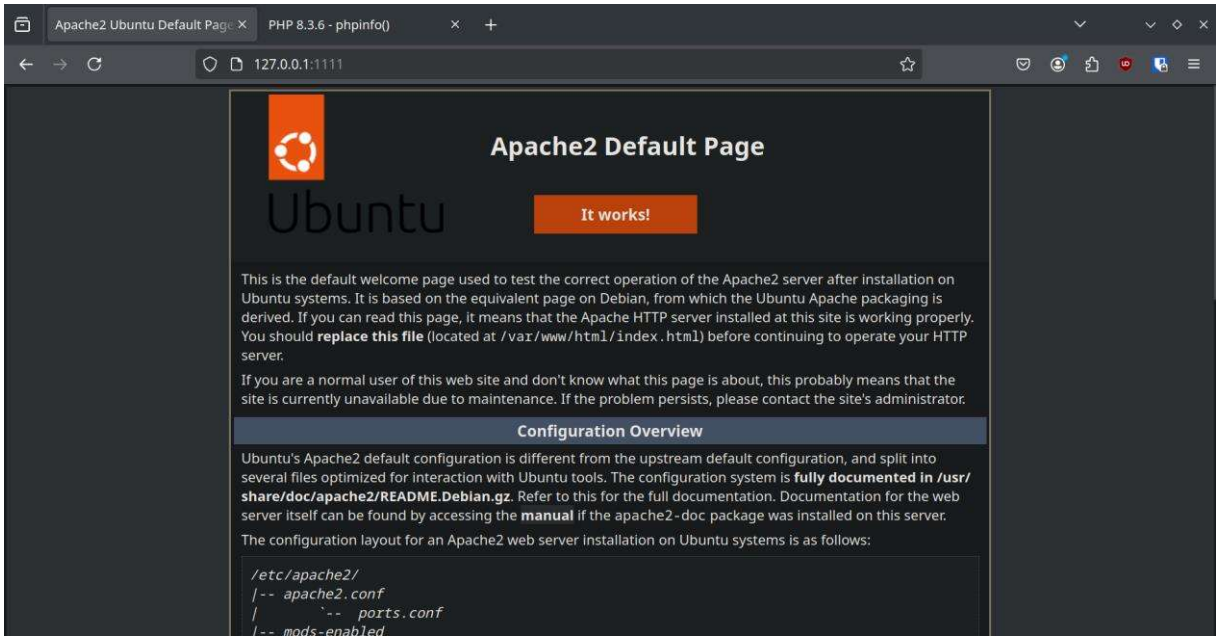
Posteriormente, reiniciamos novamente o servidor Apache:

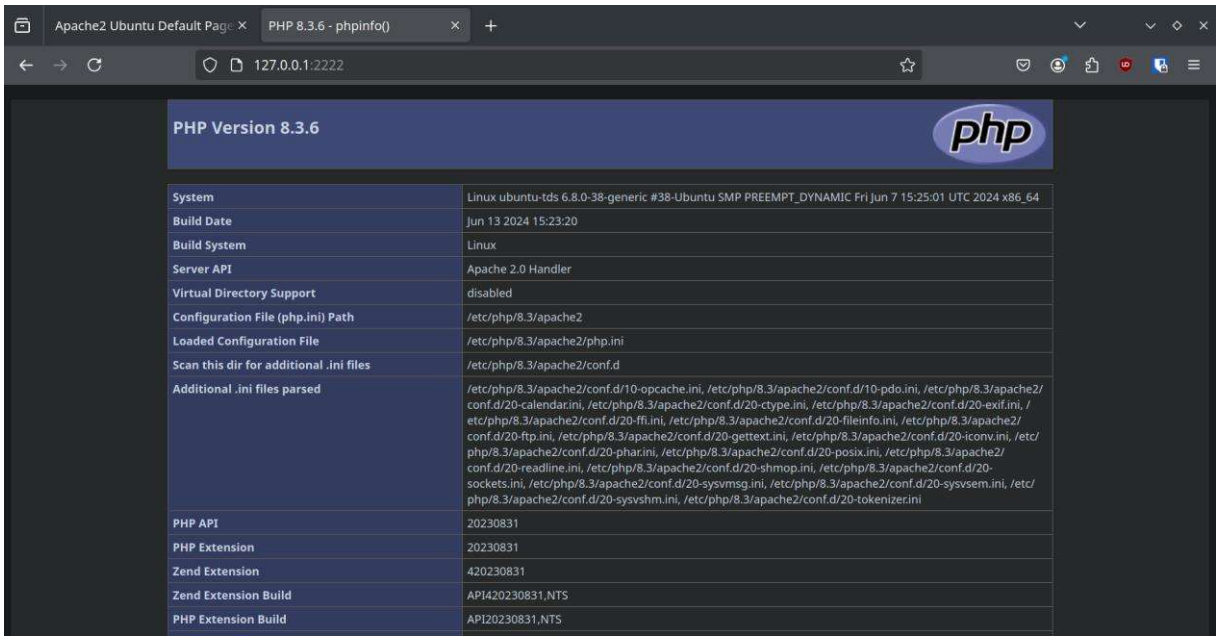
service apache2 restart

Para testar, encaminhamos as portas da máquina virtual para a máquina original:



E acessamos as páginas em um navegador comum:





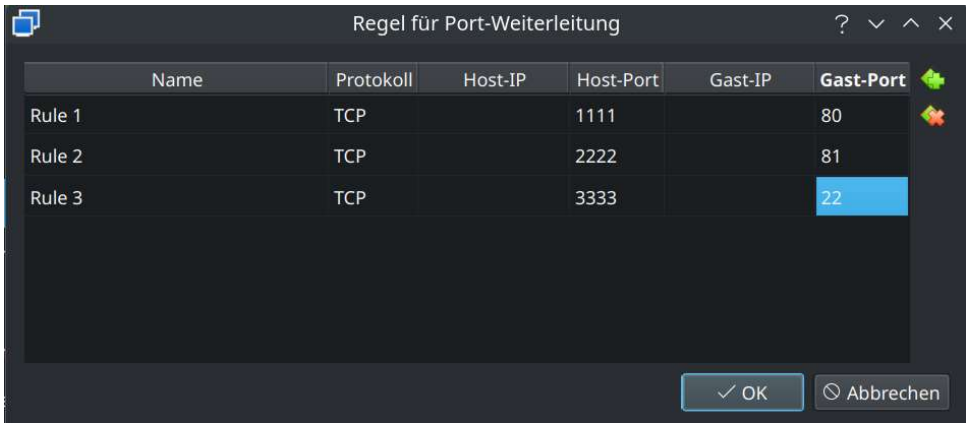
3º) Para começarmos a configuração do servidor de acesso remoto, precisamos inicialmente instalar o SSH na máquina virtual, para isso utilizamos o seguinte comando:

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

Logo após a instalação habilitamos e iniciamos o serviço do SSH com o comando:

```
# systemctl  enable –now ssh
```

Agora, precisamos adicionar uma porta para o serviço do SSH, assim como demostrado na imagem a seguir:



Assim, o login por senha já foi configurado, podemos testar usando o comando:

```
# ssh cat@127.0.0.1 -p 3333
```

```
[fabioc@selkirkk ~]$ ssh cat@127.0.0.1 -p 3333
cat@127.0.0.1's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of ter 06 ago 2024 21:00:49 UTC

  System load:  0.39               Processes:               137
  Usage of /:   32.3% of 19.51GB   Users logged in:         1
  Memory usage: 7%                 IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Manutenção de Segurança Expandida para Applications não está ativa.

70 as atualizações podem ser aplicadas imediatamente.
26 dessas atualizações são atualizações de segurança padrão.
Para ver as actualizações adicionais corre o comando: apt list --upgradable

Ativar ESM Apps para poder receber possiveis futuras atualizações de segurança.
Consulte https://ubuntu.com/esm ou execute: sudo pro status


Last login: Tue Aug  6 20:56:38 2024 from 10.0.2.2
cat@ubuntu-tds:~$
```

Posteriormente geramos uma chave na máquina original utilizando o seguinte comando:

```
$ ssh-keygen
```

```
[fabioc@selkirkk ~]$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/fabioc/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/fabioc/.ssh/id_ed25519
Your public key has been saved in /home/fabioc/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:5mYOWN9EWSlKGfDu/QX2smoztPNjJfnWOOslyHhzn/o fabioc@selkirkk
The key's randomart image is:
+--[ED25519 256]--+
|      ...o  ..   |
|       .o .o.    |
|       ...o.     |
|        ...      |
|       . S . o.  |
|      o = +.+o+. |
|     . . *.+.*+=o.|
|        =  *oo*=+o|
|         ...B+==E.|
+----[SHA256]-----+
```

E para copiar a chave gerada para a máquina virtual usamos o comando:

$ ssh-copy-id -p 3333 cat@127.0.0.1

```
[fabioc@selkirkk ~]$ ssh-copy-id -p 3333 cat@127.0.0.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/fabioc/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alrea
dy installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to instal
l the new keys
cat@127.0.0.1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh -p 3333 'cat@127.0.0.1'"
and check to make sure that only the key(s) you wanted were added.
```

Agora conseguimos logar automaticamente, sem utilizar senha: