FICHA TRABALHO 3

MODALIDADE:	Aprendizagem +	Não aplicável	
CURSO:	Técnico/a de Cibersegurança		
UC:	Gerir sistemas de deteção de intrusos (IDS)	CÓDIGO UC:	UC01486
FORMADOR/A:	Bruno Silva	DATA:	

OBJETIVOS

- Saber como instalar e configurar um servidor SSH
- Utilização do software SSH e interligação com a aplicação Putty para acesso remoto

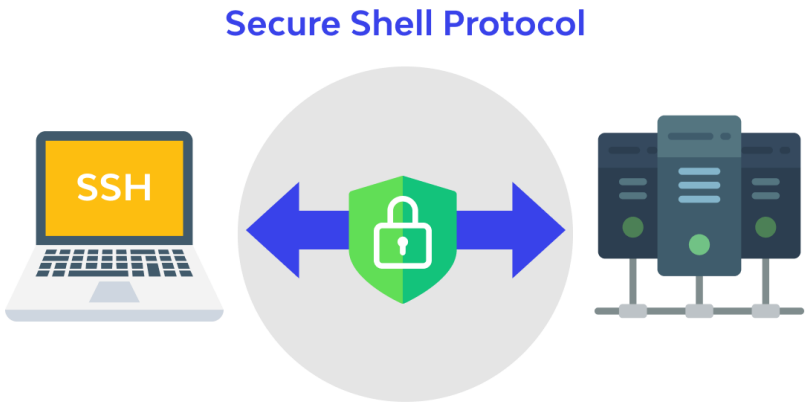
SSH (Secure Socket Shell)

SSH (Secure Socket Shell), é um dos protocolos específicos de segurança para troca de informações entre cliente e servidor que utiliza criptografia. O objetivo do SSH é permitir que desenvolvedores ou outros utilizadores realizem alterações em websites e acedam aos servidores utilizando uma conexão simples, mas com segurança.

De uma forma geral, este protocolo cria um método seguro e impede que as informações sejam expostas ou corrompidos por terceiros. É aí que as criptografias são usadas, permitindo apenas que dois pontos acedam as informações, que são o servidor e o computador que enviou os dados para esse local remoto.

O OpenSSH é uma versão gratuita da implementação do SSH no Linux.

O OpenSSH é um conjunto de ferramentas que nos permite gerir remotamente máquinas, recorrendo ao protocolo SSH. Ao contrário de outras ferramentas como o Telnet, rcp, rlogin e ftp, o OpenSSH garante que as comunicações entre máquinas sejam seguras, pois, recorre à criptografia para cifrar todo o tráfego (incluindo passwords).



Benefícios da utilização do SSH:

1. ***Encriptação:*** O SSH fornece criptografia de todos os dados trocados entre o cliente e o servidor, garantindo que informações confidenciais, como senhas e dados confidenciais, sejam protegidas contra-espionagem e intercetação por agentes mal-intencionados.

2. ***Autenticação***: Suporta vários mecanismos de autenticação, incluindo autenticação baseada por senha e métodos mais seguros, como autenticação de chave pública, que permite usar um par de chaves digitais em vez de uma senha.

3. ***Encaminhamento/túnel de porta***: O SSH permite aos utilizadores encaminhar ou proteger com segurança as portas de comunicação e tráfego de rede. Isto pode ser usado para proteger transferências de arquivos com SFTP, configurar VPNs e aceder com segurança a aplicações como websites, base de dados entre outros serviços.

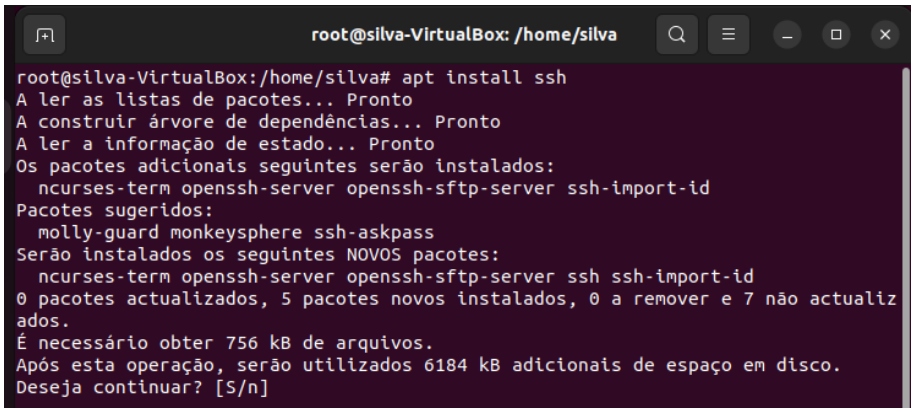
4. ***Comando e controlo***: Através do SSH, os utilizadores podem executar comandos nos servidores remotos, gerir configurações dos sistemas, gerir aplicações e realizar tarefas administrativas, tudo por meio de um canal seguro e sem a necessidade de aceder fisicamente ao servidor.

5. ***Interoperabilidade***: O SSH é compatível com uma ampla variedade de plataformas, incluindo Linux, Unix, Windows e macOS, o que torna esta ferramenta muito versátil.

Parte 1 – Instalação do servidor SSH

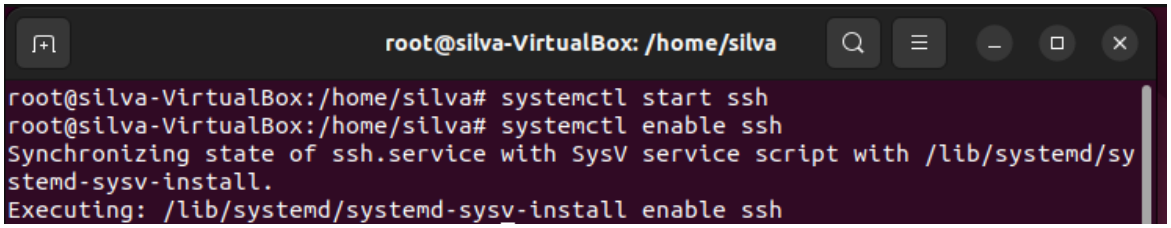
Passo 1 – Atualizar os repositórios do Ubuntu, utilize o comando: **apt-get update**

Passo 2 – Para instalar o SSH no Ubuntu, use o comando: **apt-get install ssh**

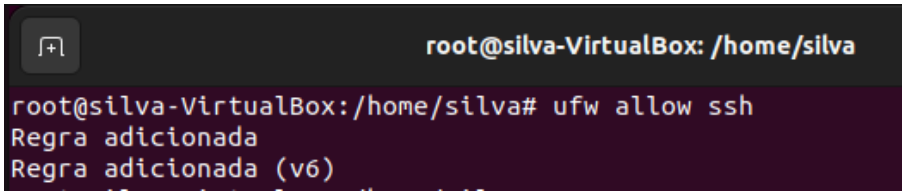


Passo 3 - Em seguida vamos iniciar e ativar o serviço. Para tal usem os seguintes comandos:

- ***Iniciar o serviço:*** systemctl start ssh
- ***Ativar o serviço no arranque do ubuntu:*** systemctl enable ssh



Passo 4 - Abrir o porto de comunicação 22 na firewall. Use o seguinte comando: ***ufw allow ssh***



Se desejar ver se a porta foi bem adicionada, deve utilizar o comando: ufw status

Aspetos importantes na configuração:

No diretório ***/etc/ssh*** vamos ter dois ficheiros de configuração***:***

- ***ssh_config*** ficheiro com as configurações do cliente;
- ***sshd_config*** ficheiro com as configurações do servidor;

```
root@silva-VirtualBox:/home/silva# ls -l /etc/ssh/
total 540
-rw-r--r-- 1 root root 505426 jan  2 16:54 moduli
-rw-r--r-- 1 root root   1650 jan  2 16:54 ssh_config
drwxr-xr-x 2 root root   4096 jan  2 16:54 ssh_config.d
-rw-r--r-- 1 root root   3254 jan  2 16:54 sshd_config
drwxr-xr-x 2 root root   4096 jan  2 16:54 sshd_config.d
-rw------- 1 root root    513 abr  6 15:54 ssh_host_ecdsa_key
-rw-r--r-- 1 root root    183 abr  6 15:54 ssh_host_ecdsa_key.pub
-rw------- 1 root root    411 abr  6 15:54 ssh_host_ed25519_key
-rw-r--r-- 1 root root    103 abr  6 15:54 ssh_host_ed25519_key.pub
-rw------- 1 root root   2610 abr  6 15:54 ssh_host_rsa_key
-rw-r--r-- 1 root root    575 abr  6 15:54 ssh_host_rsa_key.pub
-rw-r--r-- 1 root root    342 dez  7  2020 ssh_import_id
```

Parte 2 – Configurações de segurança importantes do OpenSSH

Passo 1 – Criar nova conta de utilizador (para além daquele que já tem no Ubuntu)

Para esta parte, é importante que tenho um utilizador que tenha acessos administrativos. Por exemplo, permita que o utilizador ***admin*** efetue login como root usando o comando sudo. Para tal deve fazer as seguintes configurações:

- ***Criar novo utilizador:*** adduser admin

```
root@silva-VirtualBox:/home/silva# adduser admin
A adicionar o utilizador `admin' ...
A adicionar o novo grupo `admin' (1003) ...
A adicionar o novo utilizador `admin' (1003) com grupo `admin' ...
A criar directório home `/home/admin' ...
A copiar ficheiros de `/etc/skel' ...
Nova palavra-passe:
```

- ***Dar permissões administradoras ao novo utilizador:*** adduser admin sudo

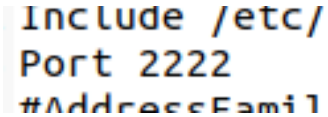
```
root@silva-VirtualBox:/home/silva# adduser admin sudo
A adicionar o utilizador `admin' ao grupo `sudo' ...
A adicionar o utilizador admin ao grupo sudo
Concluído.
```

Passo 2 – Em seguida basta ir ao ficheiro de configuração do OpenSSH (/etc/ssh/sshd_config)

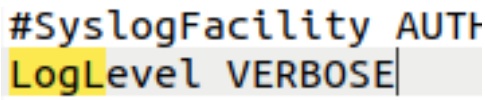
Ao editar seu arquivo de configuração, algumas opções podem ser comentadas por padrão usando um caractere (#) no início da linha. Para editar essas opções, ou habilitá-las, vai precisar descimentá-las removendo o #.

Aceder ao ficheiro da configuração: gedit /etc/ssh/sshd_config

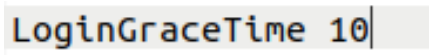
1. ***Mudar o porto lógico de acesso ao OpenSSH (linha 23):*** por omissão é o 22. Para mudar o porto, deve ir a instrução Port e mudar o porto de 22 para 2222.



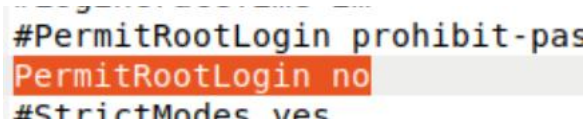
2. ***Mais informações pormenorizadas (linha 37):*** Por padrão, o SSH regista tudo. Se quiser registar mais informações, como tentativas de login mal sucedidas, no campo ***LogLevel*** altere o valor de INFO para ***VERBOSE***.



3. ***Tempo de espera para autenticação (linha 41):*** é possível definir o tempo que um utilizador tem para concluir a autenticação depois de ligar inicialmente ao seu servidor SSH. Para tal basta mudar o parâmetro ***LoginGraceTime*** e definir o tempo em segundos. Neste caso, coloque 10 segundos.



4. ***Desabilitar o login via SSH com o utilizador root (linha 42)***: defina a opção ***PermitRootLogin*** com o valor ***NO*** (veja se é necessário remover o comentário no início da linha);



5. ***Indicar número de tentativas para autenticar por ligação (linha 44):*** descomente a linha e no campo ***MaxAuthTries*** especifique o número máximo de tentativas de autenticação permitidas por cada ligação ***são apenas 3 tentativas***.



6. ***Desativar o acesso SSH sem password (linha 67):*** tirar o comentário e colocar o ***PermitEmptyPasswords*** com o valor ***NO;***

```
PermitEmptyPasswords no
```

7. ***Ocultar a informação de quem fez o último login (linha 104)***: no campo ***PrintLastLog*** coloque o valor ***NO.***

```
PrintLastLog no
```

8. ***Tempo de inatividade (linhas 108 e 109)***: é uma boa prática definir um tempo limite de inatividade adequado para evitar uma sessão autônoma. Para tal, descomente as linhas e mude os valores dos campos:

   - ClientAliveInterval 300;
   - ClientAliveCountMax 0

```
 99 #Compression delayed
100 ClientAliveInterval 300
101 ClientAliveCountMax 0
102 #UseDNS no
```

9. ***Listas de permissões de Utilizadores:*** poderá ser necessário dar ou negar acessos aos utilizadores façam login no SSH, ***melhorando assim a sua segurança***. Por padrão, esta opção não está disponível no arquivo de configuração SSH. Para tal, podemos colocar os seguintes comandos:

   - Para ***permitir*** utilizadores: ***AllowUsers*** user1 user2 .... (também pode ser endereços IP)
   - Para ***negar*** utilizadores: ***DenyUsers*** user3 user4 ..... (também pode ser endereços IP)

   Outros exemplos prático:

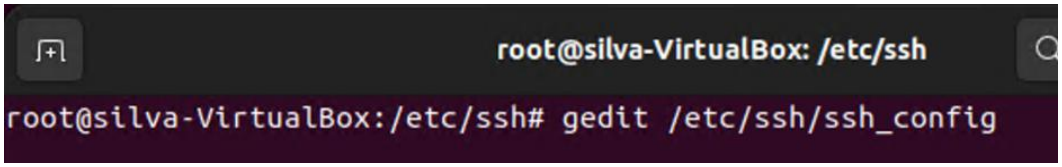
   - ***Restringir*** utilizadores a um endereço IP específico: AllowUsers *@203.0.113.1
   - ***Restringir*** utilizadores a um intervalo de endereços IP específico usando a máscara de rede: AllowUsers *@203.0.113.0/24
   - ***Restringir*** utilizadores a vários endereços e intervalos IP específicos:

     AllowUsers *@203.0.113.1 *@203.0.113.2 *@192.0.2.0/24 *@172.16.*.1

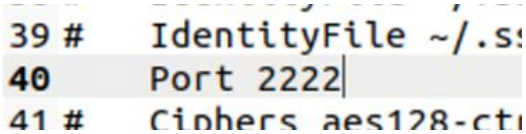
- ***Restringir*** todos os utilizadores, ***exceto*** utilizadores nomeados, de endereços IP específicos:

  AllowUsers sammy@203.0.113.1 alex@203.0.113.2

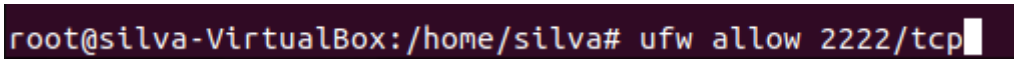
MUITO IMPORTANTE 1: **Se *mudar a porta no ficheiro anterior (configuração servidor)***, também temos de mudar os dados ***no ficheiro do cliente (ficheiro ssh_config). Vamos fazer isto logo após a configuração deste ficheiro***;



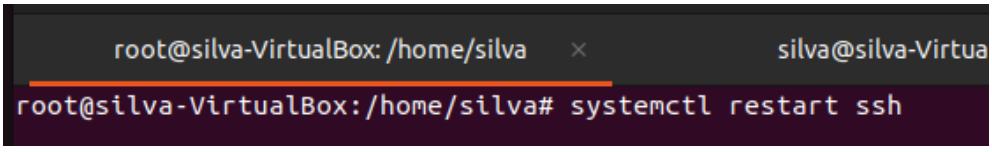
Mudar o valor do parâmetro Port para o que indicou no ficheiro do servidor (neste caso foi o 2222):



MUITO IMPORTANTE 2: Se mudar a porta neste ficheiro, também temos de adicionar a regra na firewall. Use o comando: ***ufw allow 2222/tcp → Vamos fazer isto logo após a configuração deste ficheiro***



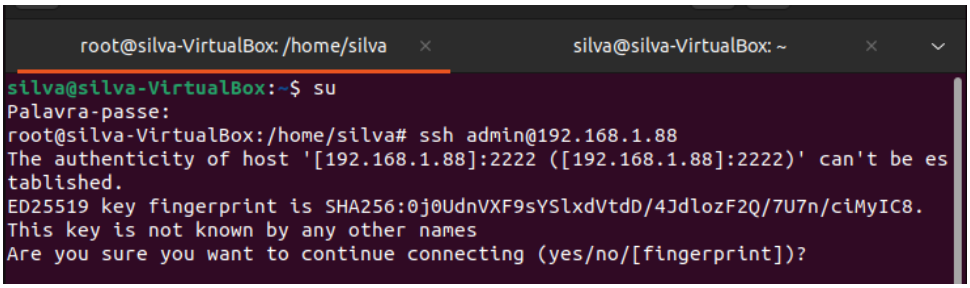
Passo 3 – No final, vamos gravar as alterações e reinicie o serviço: ***systemctl restart ssh***



Para confirmar se o serviço está a funcionar normalmente, use o comando: **systemctl status ssh**

Passo 4 – Para ligarem a uma máquina com SSH podem usar ferramentas como o Putty ou WinSCP, ***ou então***, via o terminal (quer Windows (a partir do Windows 10), quer Linux), com o comando:

ssh utilizador@endereco_ip_do_servidor



MUITO IMPORTANTE 3: Isto não deve funcionar (ver a mensagem do erro que não conseguiu estabelecer a ligação. Deve cancelar a operação (CTRL+C) e temos de verificar se o socket do ssh no sistema operativo também está a ouvir a porta 2222 (pois mudamos e ainda deve estar na porta 22).

Passo 5 – Por vezes só a ativação do porto no ficheiro de configuração não é o suficiente. Como tal, temos de ir ao serviço que corre no sistema **ssh.socket** (que está a ouvir o canal de configurações no sistema operativo) e ainda pode estar na porta 22. Como tal e se alterar a porta, vamos ter de editar o ficheiro do socket com o seguinte comando: **systemctl edit ssh.socket**

```
root@linsrv:/etc/ssh# systemctl edit ssh.socket_
```

Quando abrir o ficheiro, ative as linhas mais abaixo e na porta mude a opção 22 para 2222 (de acordo com o que foi feito anteriormente):

```
  GNU nano 7.2                                    /etc/systemd/system/ssh.socket.d/.#o
### Editing /etc/systemd/system/ssh.socket.d/override.conf
### Anything between here and the comment below will become the contents of the drop-i



### Edits below this comment will be discarded


### /usr/lib/systemd/system/ssh.socket
# [Unit]
# Description=OpenBSD Secure Shell server socket
# Before=sockets.target ssh.service
# ConditionPathExists=!/etc/ssh/sshd_not_to_be_run
#
 [Socket]
 ListenStream=0.0.0.0:2222
 ListenStream=[::]:2222
# BindIPv6Only=ipv6-only
# Accept=no
```

Vamos reiniciar o serviço do ssh:

```
root@linsrv:/etc/systemd/system# systemctl restart ssh
```

Veja se o serviço já assumiu a nova porta serviço:

```
root@linsrv:/etc/systemd/system# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Wed 2026-11-11 11:41:39 UTC; 3s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 3594 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 3596 (sshd)
      Tasks: 1 (limit: 4600)
     Memory: 1.2M (peak: 1.4M)
        CPU: 10ms
     CGroup: /system.slice/ssh.service
             └─3596 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

nov 11 11:41:39 linsrv systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
nov 11 11:41:39 linsrv sshd[3596]: Server listening on 0.0.0.0 port 2222.
nov 11 11:41:39 linsrv sshd[3596]: Server listening on :: port 2222.
nov 11 11:41:39 linsrv systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@linsrv:/etc/systemd/system#
```

Teste a conexão ssh, Deve utilizar o parâmetro ***-p (de port ou porta)*** para indicar a porta:

ssh utilizador@endereco_ip_do_servidor -p porto_comunicação

```
root@silva-QEMU-Virtual-Machine: /home/silva
root@silva-QEMU-Virtual-Machine:/home/silva# ssh admin@192.168.1.88 -p 2222
```

Se ao tentar aceder, der o erro "... broken pipe", deve editar o ficheiro SSH (sshd_config) do servidor e adicionar o seguinte parâmetro: ***UsePrivilegeSeparation yes***

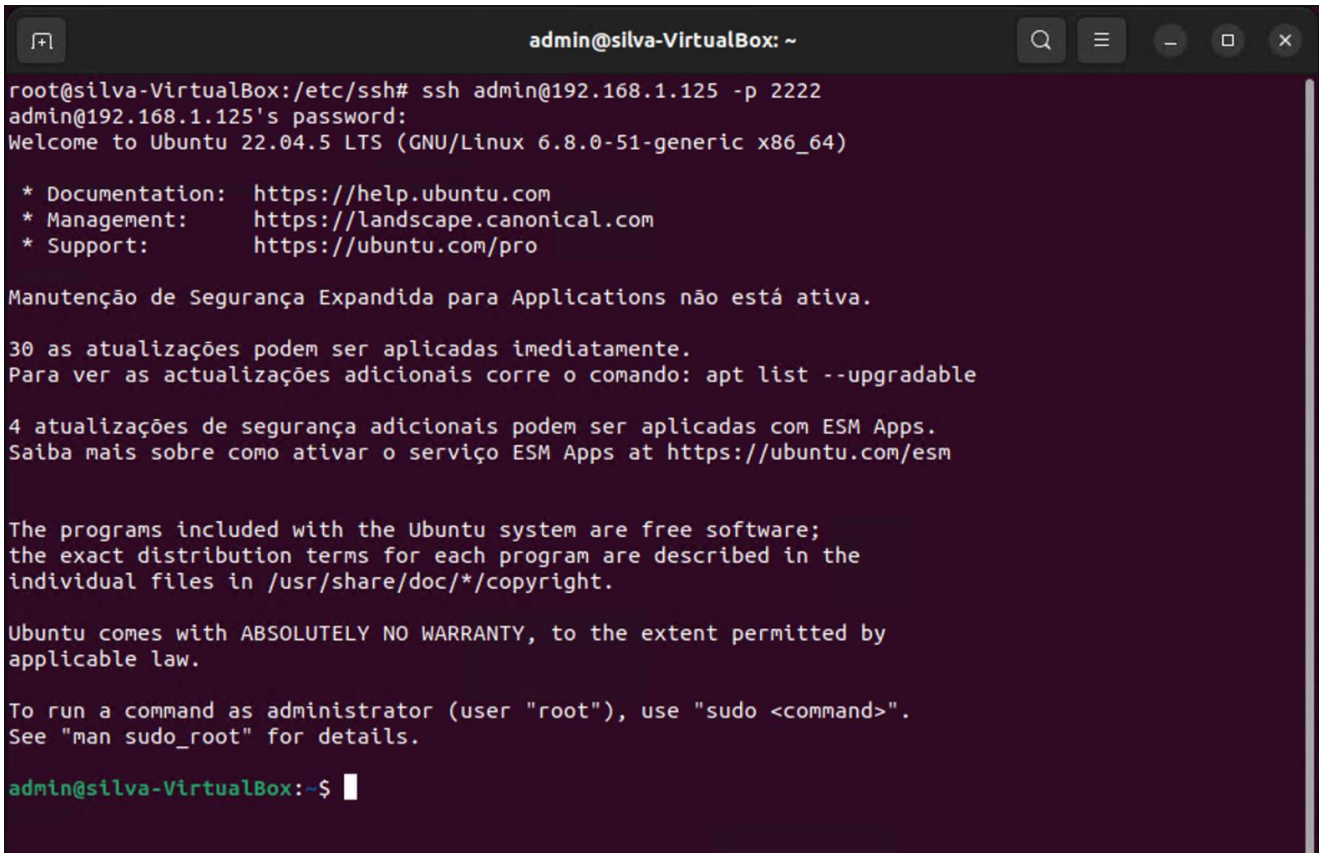
```
hosts.
ssh_dispatch_run_fatal: Connection to 192.168.1.88 port 2222: Broken pipe
```

```
10 # default value.
11 UsePrivilegeSeparation yes
12
```

No final, vamos gravar as alterações e reiniciar o serviço com o comando: ***systemctl restart ssh***

```
root@silva-VirtualBox: /home/silva          silva@silva-Virtua
root@silva-VirtualBox:/home/silva# systemctl restart ssh
```

Resultado após a ligação:



MUITO IMPORTANTE: não feche a janela da autenticação do ssh, pois ainda vamos necessitar da mesma!!