UNIVERSIDADE FEDERAL DE LAVRAS

Departamento de Ciência da Computação
Disciplina: Redes de Computadores - GCC125
Trabalho de Instalação - Etapa 1
Professor: Hermes Pimenta de Moraes Júnior
Data de entrega: 10/01/2023

Grupo T - Alunos:

- Guilherme Grego Santos - 202020417 - 10A
- Lucas Neves Sáber Gabriel - 202020459 - 10A
- Thiago Odilon de Almeida - 202021025 - 14A
- Otávio Augusto Trindade Fonseca - 202020551 - 10A

Passos Preliminares: Acessando a máquina virtual via ssh

O primeiro passo para ter acesso às máquinas virtuais foi conectar no laboratório utilizando OpenVPN. Por conseguinte, foi atualizado o repositório local, pois é recomendo fazer essa atualização antes de instalar qualquer pacote em um sistema linux.

```
sudo apt-get update
```

Depois disso, é necessário instalar o servidor ssh. Por padrão os sistema linux mais recentes já possuem o cliente ssh. Entretanto, nem todos vêm com o servidor ssh previamente instalados.

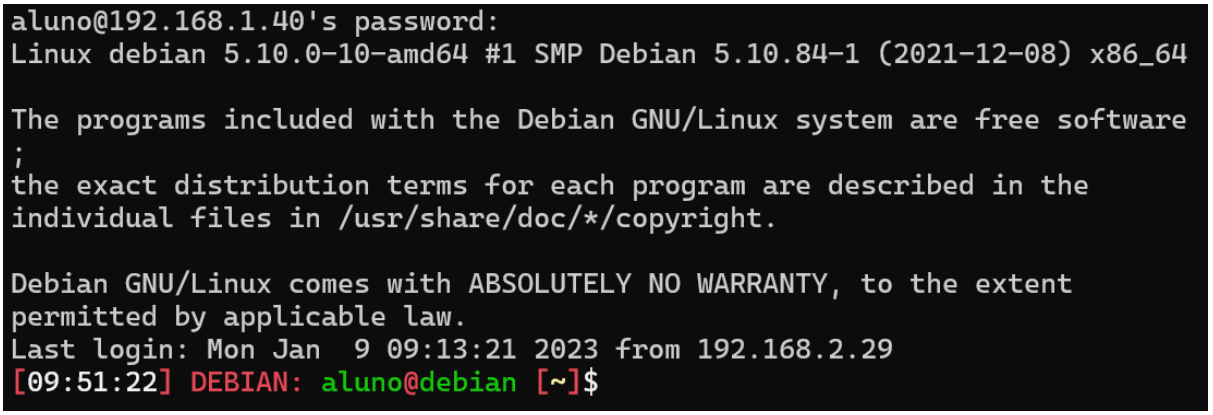
Instale o servidor ssh executando o seguinte comando.

```
sudo apt-get install openssh-server
```

Por fim, abra um terminal e digite o comando abaixo, substituindo o numero_IP_VM pelo o número IP da máquina que deseje acessar.

```
ssh aluno@numero_IP_VM
```

Se tudo der certo, você terá acessado a máquina virtual. A figura abaixo mostra a mensagem recebida após a conexão.



Alteração das senhas de acesso

Em primeiro lugar, como foi sugerido, a senha do usuário **aluno** para acessar às VMs com IP **192.168.1.39** e **192.168.1.40** foi alterada utilizando o comando **passwd**. A senha anterior era aluno, agora passou a ser **grupot**.

Instalação do serviço de sincronização de hora

Sincronizando a hora VM 192.168.1.39 com um servidor do NTP.br

A VM **192.168.1.39** se conectará em um servidor do NTP.br e a VM **192.168.1.40** será a cliente de hora. Foi utilizado o seguinte comando para visualizar a data e o horário local nas VMs:

```
date
```

Instalamos o chrony com o comando:

```
sudo apt install chrony
```

Paramos o serviço do chrony para alterar o arquivo de configuração:

```
sudo systemctl stop chrony
```

Por conseguinte, modificamos o valor da hora para um valor errado (00:00:00), com a finalidade de testar os serviços. Depois, usamos o comando date para conferir a atualização. Isto foi feito com os comandos abaixo:

```
sudo timedatectl set-time 00:00:00
date
```

Depois, abrimos o arquivo de configuração do chrony na máquina para editá-lo:

```
sudo nano /etc/chrony/chrony.conf
```

E configuramos para o seguinte estado, conforme o tutorial do NTP.br:

```
# servidores publicos do NTP.br com NTS disponível
server a.st1.ntp.br iburst
server b.st1.ntp.br iburst
server c.st1.ntp.br iburst
server d.st1.ntp.br iburst
server gps.ntp.br iburst

# arquivo usado para manter a informação do atraso do seu
relógio local
driftfile /var/lib/chrony/chrony.drift

# local para as chaves e cookies NTS
ntsdumpdir /var/lib/chrony

# erro máximo tolerado em ppm em relação aos servidores
maxupdateskew 100.0

# habilita a sincronização via kernel do real-time clock a
cada 11 minutos
rtcsync

# ajusta a hora do sistema com um "salto", de uma só vez, ao
invés de
```

```
# ajustá-la aos poucos corrigindo a frequência, mas isso
apenas se o erro
# for maior do que 1 segundo e somente para os 3 primeiros
ajustes
makestep 1 3

# diretiva que indica que o offset UTC e leapseconds devem
ser lidos
# da base tz (de time zone) do sistema
leapsectz right/UTC
```

Após salvar as mudanças, inicializamos o serviço do chrony:

```
sudo systemctl start chrony
```

Em sequência, verificamos os servidores NTP.br adicionados como fonts:

```
chronyc sources
```

E utilizando o seguinte comando, verificamos o status da sincronização do relógio da VM, que mostrará a mesma sincronizada de acordo com o servidor gps.jd.ntp.br:

```
chronyc tracking
```

Depois utilizamos o comando date para verificar a data da VM e concluímos que ela foi sincronizada, e depois tentamos alterá-la novamente com o comando sudo timedatectl set-time 00:00:00, rmas recebemos a seguinte mensagem de erro:

```
Failed to set time: Automatic time synchronization is enabled
```

Sincronizando a hora da VM 192.168.1.40 com a 192.168.1.39

Primeiramente, abrimos o arquivo de configuração na VM 192.168.1.39 com o comando sudo nano /etc/chrony/chrony.conf e adicionamos o seguinte trecho, permitindo que a VM 192.168.1.40 consiga acessar a primeira como cliente NTP:

```
[...]
# permite o acesso aos seguintes clientes NTP
allow 192.168.1.40
```

Após isso, nos desconectamos da VM 192.168.1.39 e acessamos a VM 192.168.1.40. Feito isso, realizamos os mesmos passos para a instalação do chrony:

```
sudo apt install chrony
sudo systemctl stop chrony
```

Alteramos a data para um valor incorreto novamente, para fins de teste e verificamos:

```
sudo timedatectl set-time 00:00:00
date
```

Abrimos e editamos o arquivo de configuração do chony com o comando *sudo nano /etc/chrony/chrony.conf*, para que a VM 192.168.1.40 possa reconhecer a VM 192.168.1.39 como um servidor a ser buscado:

```
# servidores publicos do NTP.br com NTS disponível
server 192.168.1.39 iburst

[...]
```

Por fim, ativamos o serviço do chrony e verificamos com os seguintes comandos que a hora da VM 192.168.1.40 está sincronizada com a máquina 192.168.1.39:

```
sudo systemctl start chrony
chronyc sources
chronyc tracking
date
```

Instalação do servidor WEB

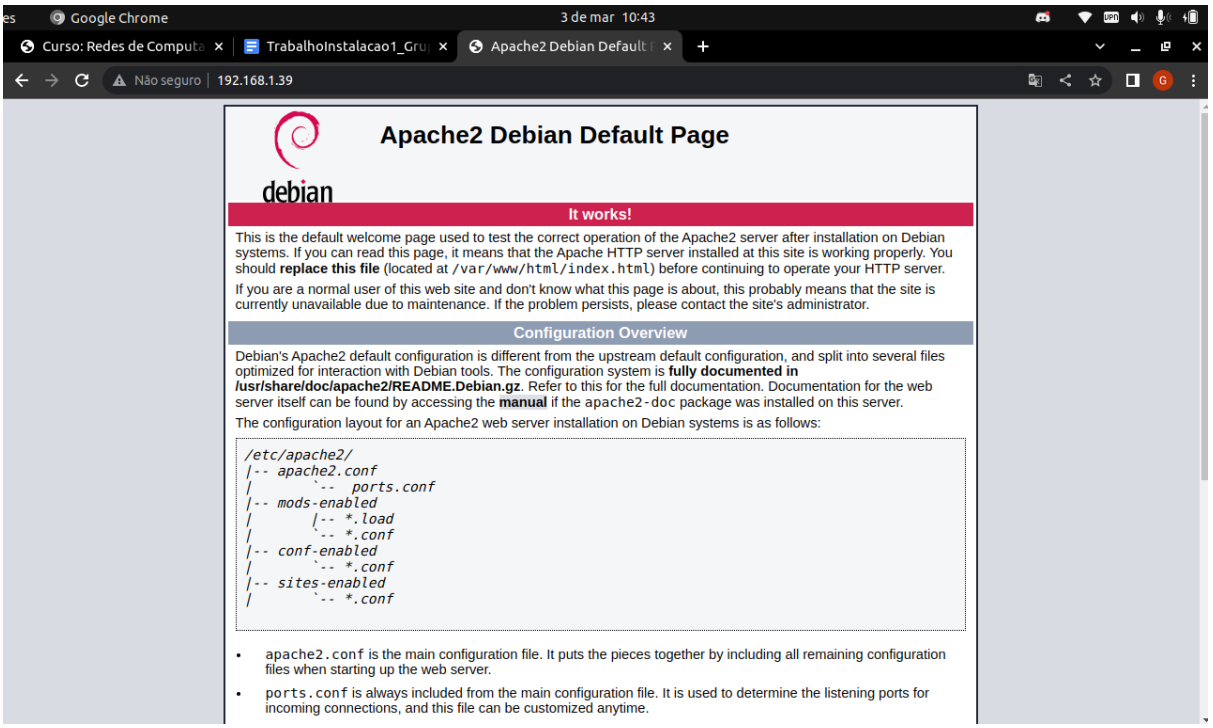
A VM adota para as instalações abaixo foi a **192.168.1.39.**

Instalação do apache

Em primeiro lugar, foi instalado o apache 2 com os comandos:

```
sudo apt install apache2
```

Testamos a instalação do serviço acessando um browser de uma máquina conectada a VM com a seguinte url: http://192.168.1.39, a saída foi:



Inserindo página html no servidor web

Acessamos o diretório do servidor apache onde os arquivos html estão com o comando a seguir:

```
cd /var/www/html/
```

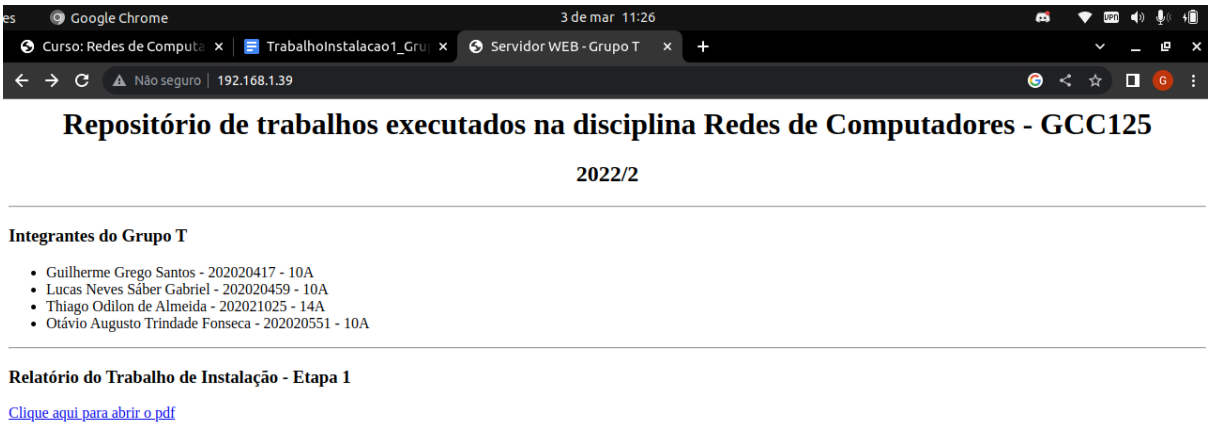
Com o seguinte comando, deletamos todos os arquivos da pasta com a finalidade de preparar o ambiente para nossos arquivos:

```
sudo rm *
```

Nossos arquivos são baixados do repositório https://github.com/GregoSX/Trabalho-Final-Redes.git, através dos comandos abaixo:

```
sudo
wget https://raw.githubusercontent.com/GregoSX/Trabalho-Final-Redes/main/index.html
```

Agora, ao acessar novamente o endereço http://192.168.1.39/ em um browser temos o seguinte resultado:



Inserindo acesso com criptografia (https) no servidor web

Em primeiro lugar, instalamos o pacote openssl na máquina de servidor web 192.168.1.39.

```
sudo apt install openssl
```

Após instalado habilitamos o ssl e rewrite do Apache.

```
sudo a2enmod ssl
```

```
sudo a2enmod rewrite
```

Utilizamos o editor de arquivo nano para fazer a configuração do Apache.

```
sudo nano /etc/apache2/apache2.conf
```

Adicionando o trecho de código abaixo:

```
[...]
<Directory /var/www/html>
        AllowOverride All
</Directory>
```

Em seguida foi criada a pasta para guardar o certificado ssl que será criado.

```
sudo mkdir /etc/apache2/certificate
cd /etc/apache2/certificate
```

Após entrar na pasta, criamos uma chave privada e o certificado ssl.

```
sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out
apache-certificate.crt -keyout
```

Depois disso, foi inserida as seguintes informações:

```
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:Minas Gerais
Locality Name (eg, city) []:Lavras
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Universidade Federal de
Lavras
Organizational Unit Name (eg, section) []:Departamento de Ciência da Computação
Common Name (e.g. server FQDN or YOUR name) []:192.168.1.39
Email Address []:.
```

E novamente utilizamos o nano para editar o arquivo 000-default.conf:

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

O arquivo conta com apenas as informações mostradas abaixo:

```
<VirtualHost *:80>
        [...]
</VirtualHost>
```

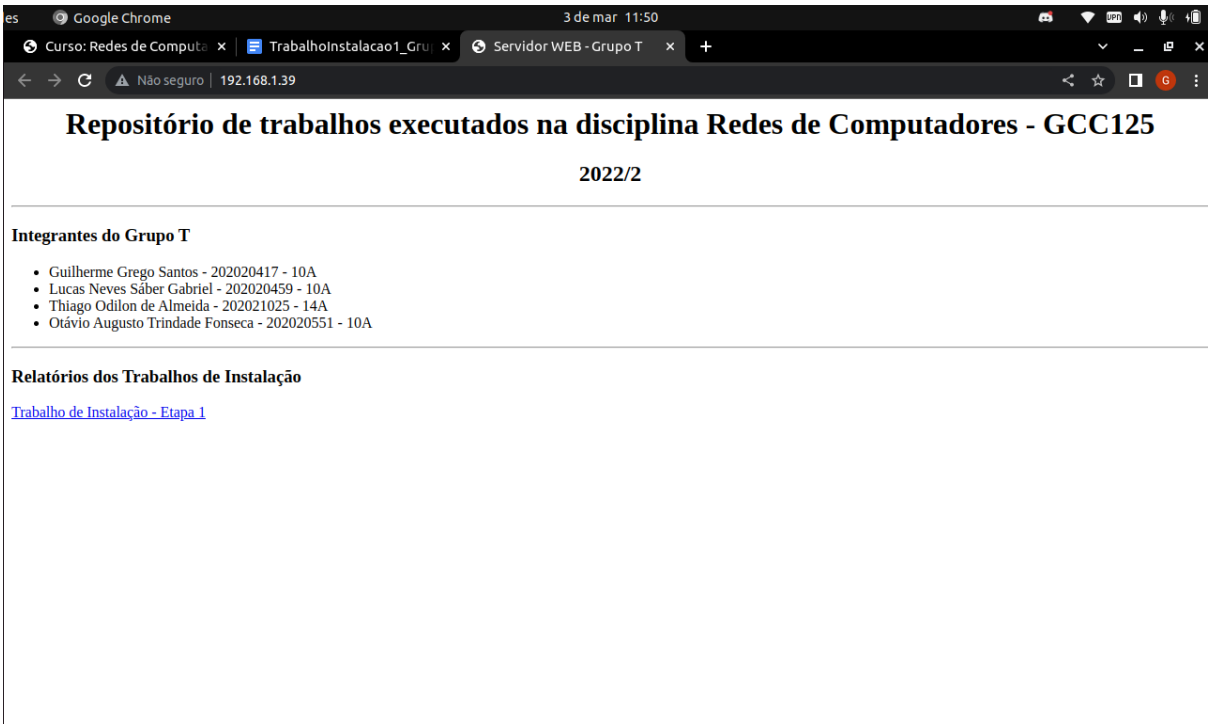
Adicionamos nele as seguintes configurações:

```
<VirtualHost *:443>
        [...]
        SSLEngine on
        SSLCertificateFile /etc/apache2/certificate/apache-certificate.crt
        SSLCertificateKeyFile /etc/apache2/certificate/apache.key
</VirtualHost>
```

Finalizado isso, reiniciamos o Apache:

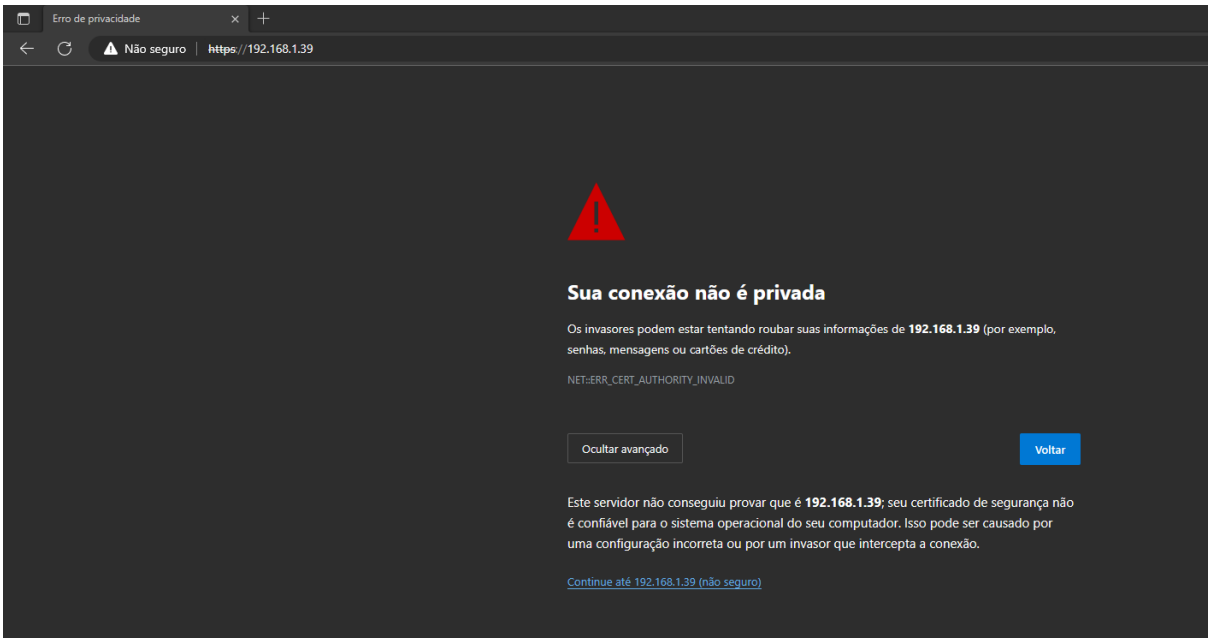
```
sudo systemctl restart apache2
```

Por fim, fizemos o teste com o acesso sem criptografia (https), pela url: http://192.168.1.39/, o resultado foi o seguinte:

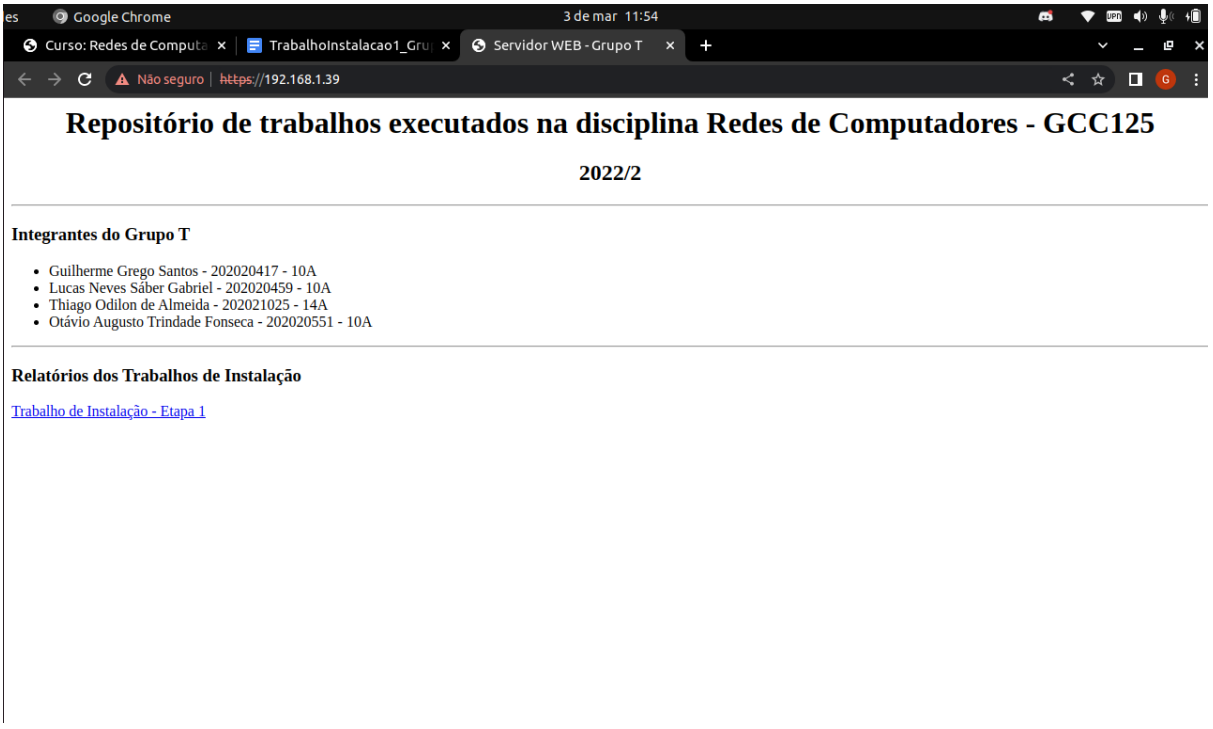


Agora, podemos acessar a página com criptografia (https) pela url https://192.168.1.39, Inicialmente é gerado uma aviso de segurança no

browser, pois o certificado ssl foi criado pelo próprio servidor, o que não garante segurança real aos usuários.



Depois de ignorar o aviso, temos o seguinte resultado:



Sendo assim, temos o servidor WEB instalado corretamente. Os relatórios podem ser acessados pela página web.