Relatório do Sprint 2 de ASIST

Turma 3DB _ Grupo 007
1211148 _ Joana Perpétuo
1211016 _ João Rodrigues
1221693 _ Hugo Silva
1201551 _ António Martingo

Distribuição de User Stories:

- **1211148 (Joana Perpétuo) –** 6 & 7
- **1211016 (João Rodrigues) –** 1 & 3
- **1201551 (António Martingo) –** 5 & 8
- **1221693 (Hugo Silva) –** 2 & 4

1. Como administrador do sistema quero que o deployment de um dos módulos do RFP numa VM do DEI seja sistemático, validando de forma agendada com o plano de testes.

O módulo escolhido para ser deployed foi o módulo de visualização. Esta escolha foi feita, uma vez que o módulo será acedido apenas pelas máquinas internas ao DEI, fará sentido ser aquele em que os utilizadores têm uma interface para interagir e não um módulo de backend que não seria possível ser posteriormente acedido pelo exterior

Para que o módulo possa ser deployed na máquina virtual do DEI, é primeiramente necessário que haja uma forma de o obter do repositório automaticamente. Para isso, é preciso criar uma chave na VM.

ssh-keygen -t rsa -b 4096 -C "name@email.com"

eval "$(ssh-agent -s)"

ssh-add ~/.ssh/id_rsa

E posteriormente adicionar esta chave ssh ao repositório.

Agora, é possível clonar o repositório na VM, o próximo passo é instalar as dependências do módulo; no caso, estamos a usar o módulo de Visualização.

apt update

apt install nodejs

apt install npm

npm install -g @angular/cli

Desta forma, a VM está pronta para receber e correr o projeto. Agora é necessário criar um script que verifica se há alterações no projeto e caso haja, serão transferidas as atualizações e corridos os scripts de build, testes e run.

```
  GNU nano 7.2                                                  deploy.sh
#!/bin/bash

export PATH=/usr/bin/git:/usr/bin/npm:/usr/local/bin/node:$PATH
export NODE_HOME=$(dirname $(which node))

local_directory="/home/lapr5_g07"
vizualization_module="/home/lapr5_g07/Vizualization"
bitbucket_repo="git@bitbucket.org:hugo_silva2901/lapr5_g07.git"

log_file="/home/cron_script.log"

perform_actions(){
        echo "Performing actions at $(date)" >> "$log_file"
        cd "$vizualization_module" || { echo "Error: Unable to change directory." >> "$log_file"; exit 1; }
        npm install >> "$log_file" 2>&1
        ng test >> "$log_file" 2>&1
        node --max_old_space_size=4096 ./node_modules/@angular/cli/bin/ng serve --host 0.0.0.0 "$log_file" 2>&1
}

if [ ! -d "$local_directory" ]; then
        git clone "$bitbucket_repo" "$local_directory" >> "$log_file" 2>&1
        perform_actions
else
        local_last_commit=$(git rev-parse HEAD)
        remote_last_commit=$(git ls-remote --heads "$bitbucket_repo" | cut -f 1)
        if [ "$local_last_commit" != "$remote_last_commit" ]; then
                git -C "$local_directory" pull >> "$log_file" 2>&1
                perform_actions
        else
                echo "Local directory is up-to-date. No action needed." >> "$log_file"
        fi
fi
```

Com isto, apenas será necessário acrescentar ao ficheiro ***/etc/crontab***, através do comando crontab -e a seguinte linha:

```
  GNU nano 7.2                                          /tmp/crontab.Qsud91/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# 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
0 5 * * * /home/deploy.sh
```

O script será executado todos os dias às 5:00AM.

2. Como administrador do sistema quero que apenas os clientes da rede interna do DEI (cablada ou via VPN) possam aceder à solução.

Começamos por correr este comando ***sudo apt-get install iptables-persistent*** para permitir que as ipatbles possam ser persistidas.
A seguir usamos o comando ***sudo iptables -A INPUT -s 10.8.0.0/16 -j ACCEPT*** *e* ***sudo iptables -A INPUT -s 10.4.0.0/16 -j ACCEPT*** de maneira a permitir tantos os alunos do dei e os professores a terem acesso.

De seguida fizemos ***sudo iptables –L*** Para verificar se as regras foram adicionadas corretamente, você pode listar todas as regras do iptables.

Agora a partir do vpn do dei o deinet podemos aceder.

```
root@vs750:/# sudo iptables  --list
# Warning: iptables-legacy tables present, use iptables-legacy to see them
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  10.8.0.0/16          anywhere
ACCEPT     all  --  10.4.0.0/16          anywhere
```

3. Como administrador do sistema quero que os clientes indicados na user story 2 possam ser definidos pela simples alteração de um ficheiro de texto.

Como foi referido na User story anterior, as iptables agora serão persistidas num ficheiro. Para definir quem pode aceder à solução disponibilizada pela VM basta editar o ficheiro de texto da localização: ***/etc/iptables/rules.v4.***

```
  GNU nano 7.2                                        /etc/iptables/rules.v4
# Generated by iptables-save v1.8.9 (nf_tables) on Fri Nov 24 12:09:46 2023
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -s 10.8.0.0/16 -j ACCEPT
-A INPUT -s 10.4.0.0/16 -j ACCEPT
COMMIT
# Completed on Fri Nov 24 12:09:46 2023
# Generated by iptables-save v1.8.9 (nf_tables) on Fri Nov 24 12:09:46 2023
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A PREROUTING -p tcp -m tcp --dport 2222 -j REDIRECT --to-ports 22
COMMIT
# Completed on Fri Nov 24 12:09:46 2023
```

Alterando as linhas da primeira secção (*filter) define-se a firewall, podendo permitir ou recusar o tráfego externo, definindo que redes podem aceder à solução.

4. Como administrador do sistema quero identificar e quantificar os riscos envolvidos na solução preconizada.

Os riscos nesta solução são.

Risco de falha do firewall: Se o iptables falhar por algum motivo, isso pode permitir o acesso não autorizado à solução. Para mitigar este risco, é importante monitorizar regularmente o estado do iptables e corrigir quaisquer problemas que possam surgir. Se o iptables falhar por algum motivo, isso pode permitir o acesso não autorizado à solução. As falhas do firewall podem ser causadas por várias razões, incluindo:

> **Vulnerabilidades de software**: As vulnerabilidades causadas por dispositivos de rede configurados incorretamente podem causar estragos, mas as vulnerabilidades de firmware representam um risco maior2. Existem falhas em qualquer programa de software que os atacantes podem explorar; isso é verdade para os aplicativos de firewall, bem como para qualquer outro software.
>
> **Problemas de desempenho**: Como os firewalls frequentemente contêm hardware de rede que é mais lento do que os tubos de internet aos quais estão conectados, a incorporação de um firewall na arquitetura de rede pode resultar em grandes gargalos quando ocorrem picos de tráfego.

Risco de configuração incorreta: Se a lista de permissões de IP ou a autenticação VPN forem configuradas incorretamente, isso pode permitir o acesso não autorizado à solução. Para mitigar este risco, é importante testar cuidadosamente todas as configurações e corrigir quaisquer erros

Risco de perda de dados: Se os dados sensíveis não forem encriptados corretamente, isso pode levar à perda de dados. Para mitigar este risco, é importante usar métodos de encriptação fortes e testar regularmente a segurança dos dados.

Risco de acesso não autorizado: Se a autenticação de dois fatores (2FA) for comprometida, isso pode permitir o acesso não autorizado à solução. Para mitigar este risco, é importante usar um provedor de 2FA confiável e educar os usuários sobre a importância da segurança das suas credenciais de 2FA. Mas isso ás vezes é uma segurança falsa. Em que autenticação de dois fatores proporciona um nível de segurança, mas geralmente é exagerada. Embora a 2FA possa tornar mais difícil para um atacante obter acesso a uma conta, ela não é infalível. Por exemplo, um atacante que tenha acesso físico ao dispositivo usado para a 2FA (como um telefone celular) pode ser capaz de contornar essa proteção.

Risco de ataques de rede: Mesmo com todas estas medidas de segurança em vigor, ainda existe o risco de ataques de rede, como ataques DDoS. Para mitigar este risco, é importante ter medidas de segurança adicionais em vigor, como um sistema de prevenção de intrusões (IPS). Os ataques podem incluir **ataques cibernéticos**: Se as políticas de segurança não forem seguidas, isso pode deixar a sua solução vulnerável a uma variedade de ataques cibernéticos, como ataques de negação de serviço (DoS) ou ransomware.

5. Como administrador do sistema quero que seja definido o MBCO (Minimum Business Continuity Objective) a propor aos stakeholders.

6. Como administrador do sistema quero que seja proposta, justificada e implementada uma estratégia de cópia de segurança que minimize o RPO (Recovery Point Objective) e o WRT (Work Recovery Time).

- O **Recovery Point Objective (RPO)** é uma medida que estabelece a quantidade máxima de dados que uma organização está disposta a perder em caso de interrupção ou falha. Em termos mais simples, é o ponto até o qual uma empresa está disposta a "voltar no tempo" para recuperar os seus dados.
- O **Work Recovery Time (WRT)** é o período que uma organização leva para recuperar completamente os seus sistemas, aplicações e dados, além de realizar testes para garantir que tudo esteja a funcionar conforme o esperado.

Durante a pesquisa pela melhor abordagem, surgiu a estratégia **3-2-1**. Esta estratégia consiste em implementar:

- Pelo menos três cópias dos dados (a original e mais dois backups);
- Dois tipos de armazenamento para estes dados;
- Um destes armazenamentos precisa de estar localizado fora da organização, num ambiente seguro.

Fonte: Conheça a estratégia de backup 3-2-1 e saiba como aplicá-la (artbackup.com.br)

Comecemos então pelo primeiro backup, para isso vamos utilizar a ferramenta **Rsync**:

apt install rsync

De seguida podemos executar o comando abaixo para copiar todos os diretórios e ficheiros do sistema Linux, excluindo alguns que não são necessários, para **/mnt/backup3:**

rsync -aAXv / --exclude={"/dev/*","/proc/*","/sys/*","/tmp/*","/run/*","/mnt/*","/media/*","/lost+found"} /mnt/backup3

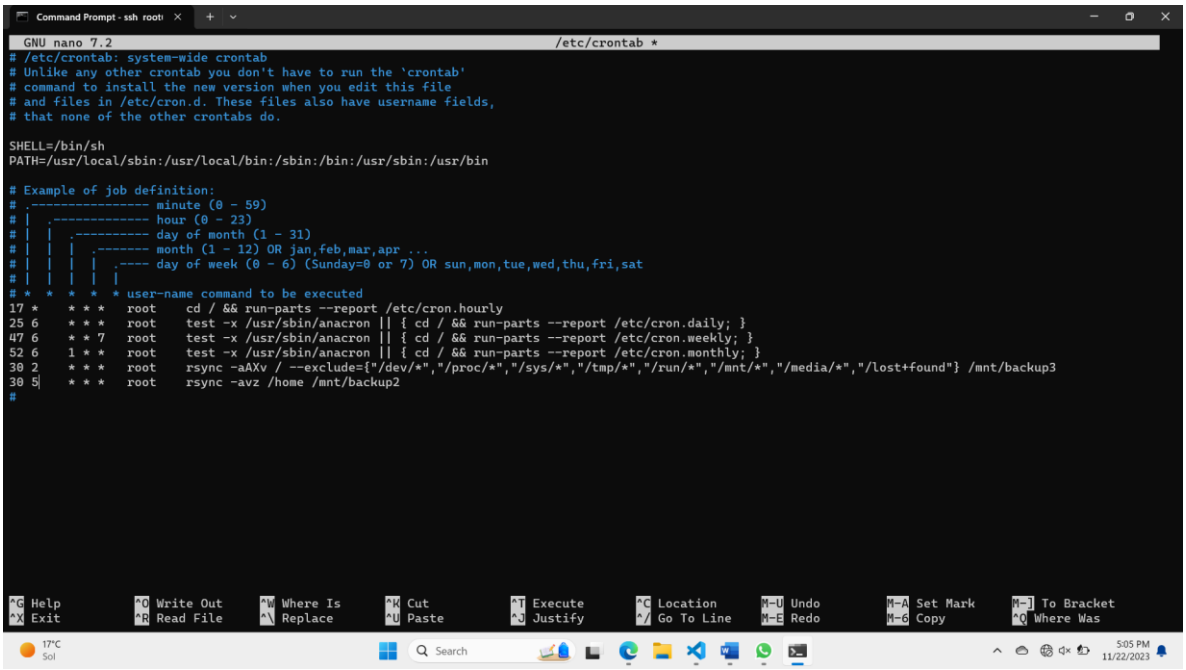
Agora podemos ainda agendar backups diários com a ferramenta **crontab,** editando o ficheiro **/etc/crontab:**

30 2 * * * root rsync -aAXv / --exclude={"/dev/*","/proc/*","/sys/*","/tmp/*","/run/*","/mnt/*","/media/*","/lost+found"} /mnt/backup3

Ou seja, será feito um backup todos os dias, às 2h30, de todo o sistema e colocado em /mnt/backup3 (desde que o servidor esteja ligado).

Para efetuar o segundo backup basta repetir o processo, mas colocar uma pasta de destino diferente, por exemplo **/mnt/backup2** e, para garantir que o primeiro já terminou antes de começar o segundo, o segundo backup começará 3 horas depois, às 5h30. Neste caso iremos efetuar apenas o backup do diretório **/home,** dado que são menos ficheiros, com este comando:

rsync -avz /home /mnt/backup2



Por fim, precisamos fazer o terceiro backup, mas numa localização fora do sistema. Para fazer este backup externo podemos utilizar um serviço de armazenamento em nuvem, como por exemplo o **Duplicity**:

apt install duplicity

O Duplicity permite fazer backups criptografados e incrementais, o que significa que apenas as alterações feitas depois do último backup serão armazenadas, para além de fornecer uma série de outras informações úteis a cada backup, como por exemplo o tempo de início e de fim do backup, a sua duração, o tamanho dos ficheiros, diferenças encontradas, erros etc. Se fizermos um backup duma pasta original diferente para a mesma pasta de destino também faz as verificações necessárias. Dado que ambas as máquinas precisam de estar na mesma rede, o nosso servidor remoto será então criado nos servidores virtuais do DEI (**Debian 11 Bullseye (base system) - All VNETs**). Neste servidor vamos criar o diretório **/mnt/Backup** que será onde vai ficar localizado o backup. Ambas as máquinas precisam de ter o ssh funcional para ser permitido o backup como também, posteriormente, a recuperação. Se preferirmos, podemos definir pares de chaves SSH (são usadas para autenticação e criptografia em conexões SSH) para facilitar o processo de autenticação e torná-lo mais seguro. Para isso é necessário executar os seguintes comandos na máquina do nosso sistema:

ssh-keygen -t rsa

ssh-copy-id root@[ip do servidor remoto]

Esta chave encontra-se no ficheiro **/root/.ssh/authorized_keys.** Feito isto, se agora quisermos aceder ao servidor que armazena o backup por SSH, estando na máquina do sistema, não pede a senha de login.

```
root@vs750:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:zXw/RX3BP017nBWevZmcWj2MF5HLdEtQGpwQku1TGBA root@vs750
The key's randomart image is:
+---[RSA 3072]----+
|         E=+*o==o|
|         ..o ++=X|
|          . ..+*#|
|         + o  +O&|
|        S + o. @=|
|           . .+..|
|             .o  |
|               . |
|                 |
+----[SHA256]-----+
```

```
root@vs750:~# ssh-copy-id root@10.9.24.28
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
```

Agora podemos fazer o backup com o seguinte comando, no terminal da máquina do sistema:
duplicity [Caminho/Original] scp://root@[ip do servidor remoto]//[Caminho/Destino]

```
root@vs750:~# duplicity /home scp://root@10.9.24.28//mnt/Backup
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: none
GnuPG passphrase for decryption:
Retype passphrase for decryption to confirm:
No signatures found, switching to full backup.
12345
--------------[ Backup Statistics ]--------------
StartTime 1700701732.72 (Thu Nov 23 01:08:52 2023)
EndTime 1700701867.57 (Thu Nov 23 01:11:07 2023)
ElapsedTime 134.85 (2 minutes 14.85 seconds)
SourceFiles 31296
SourceFileSize 634317725 (605 MB)
NewFiles 31296
NewFileSize 634317725 (605 MB)
DeletedFiles 0
ChangedFiles 0
ChangedFileSize 0 (0 bytes)
ChangedDeltaSize 0 (0 bytes)
DeltaEntries 31296
RawDeltaSize 618320938 (590 MB)
TotalDestinationSizeChange 215973296 (206 MB)
Errors 0
-------------------------------------------------
```

```
root@vs1052:/mnt/Backup# ls -a
.   duplicity-full.20231123T010807Z.manifest.gpg      duplicity-full.20231123T010807Z.vol2.difftar.gpg
..  duplicity-full.20231123T010807Z.vol1.difftar.gpg  duplicity-full-signatures.20231123T010807Z.sigtar.gpg
root@vs1052:/mnt/Backup#
```

Podemos ainda fazer mais backups para pastas diferentes, enquanto ainda houver espaço livre no servidor. Para verificar o espaço podemos executar estes comandos, ou equivalentes:

ssh root@[ip do servidor remoto] df -h

ssh root@[ip do servidor remoto] df -h | grep sda

```
root@vs750:~# ssh root@10.9.24.28 df -h
Filesystem                                                      Size  Used Avail Use% Mounted on
192.168.62.61:/volume3/vs_cloud/VS_7_1211148_1052/lxc/vs1052/rootfs  7.3T  3.6T  3.7T  50% /
none                                                            492K     0  492K   0% /dev
tmpfs                                                           7.8G     0  7.8G   0% /dev/shm
tmpfs                                                           3.2G   84K  3.2G   1% /run
tmpfs                                                           5.0M     0  5.0M   0% /run/lock
tmpfs                                                           4.0M     0  4.0M   0% /sys/fs/cgroup
tmpfs                                                           1.6G     0  1.6G   0% /run/user/0
root@vs750:~#
```

Para restaurar os backups remotos basta inverter os argumentos no comando para fazer os backups:

duplicity scp://root@[ip do servidor remoto]//[Caminho/Destino] [Caminho/Original2]

Para efeitos de precaução, não se deve restaurar os arquivos no mesmo local, dado que corremos o risco de restaurar um arquivo de configuração no local errado. Deve-se, portanto, criar um diretório vazio como destino. Para tornar este processo mais seguro era possível ainda restringir a comunicação do servidor remoto com o exterior, com a exceção da máquina do sistema.

A ferramenta Duplicity não faz o agendamento de backups, no entanto podemos agendar backups diários como foi feito anteriormente, no ficheiro **/etc/crontab.**

Consulta: https://youtu.be/-emhC92FPyE?si=vD5Vu7E3j9xE67RJ

Para efeitos de demonstração e de testagem das funcionalidades, foi mantida localmente uma cópia maior e uma cópia parcial, e uma cópia parcial num local remoto, porém, se tivéssemos mais tempo, faríamos duas cópias parciais locais e pelo menos uma cópia total num local remoto, e fariamos o agendamento diário da cópia total.

7. Como administrador do sistema quero definir uma pasta pública para todos os utilizadores registados no sistema.

Para a execução desta US foi utilizada a ferramenta **Samba.**

O SAMBA é um servidor e conjunto de ferramentas que permite que máquinas Linux e Windows se comuniquem entre si, compartilhando serviços (arquivos, diretório, impressão) através do protocolo SMB (Server Message Block)/CIFS (Common Internet File System), equivalentes à implementação NetBEUI no Windows.

Começamos por instalar o package:

apt install samba

Em seguida criamos uma pasta em **/home** com um nome à escolha, por exemplo **common:**

mkdir /home/common

Depois alteramos as permissões da pasta para que todos os utilizadores tenham permissões de leitura:

chmod -R 755 /home/common

O número 755 é uma representação de permissões no sistema de arquivos do Linux.

- 7 (ou seja, 4+2+1): O proprietário pode ler, escrever e executar
- 5 (ou seja, 4+0+1): Os membros do grupo podem ler e executar, mas não podem escrever
- 5 (ou seja, 4+0+1): Outros usuários podem ler e executar, mas não podem escrever

É um requesito do cliente que todos os utilizadores tenham permissões de leitura, mas que apenas os administradores do sistema tenham permissões de escrita. Para verificar quais são os utilizadores administradores do sistema (possuem permissões sudo) podemos analisar o ficheiro **/etc/sudoers:**

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

Podemos então criar um grupo chamado **admins** que contém o utilizador root, e que posteriormente pode vir a ter mais administradores caso estes sejam criados:

groupadd admins

usermod -a -G admins root

Agora alteramos o grupo da pasta para admins:

chgrp admins /srv/samba/public

chmod -R 775 /srv/samba/public

- 7 (ou seja, 4+2+1): O proprietário pode ler, escrever e executar
- 7 (ou seja, 4+2+1): Os membros do grupo podem ler, escrever e executar

- 5 (ou seja, 4+0+1): Outros usuários podem ler e executar, mas não podem escrever

Alteramos também o ficheiro **/etc/samba/smb.conf** para configurar o samba, acrescentando o seguinte:

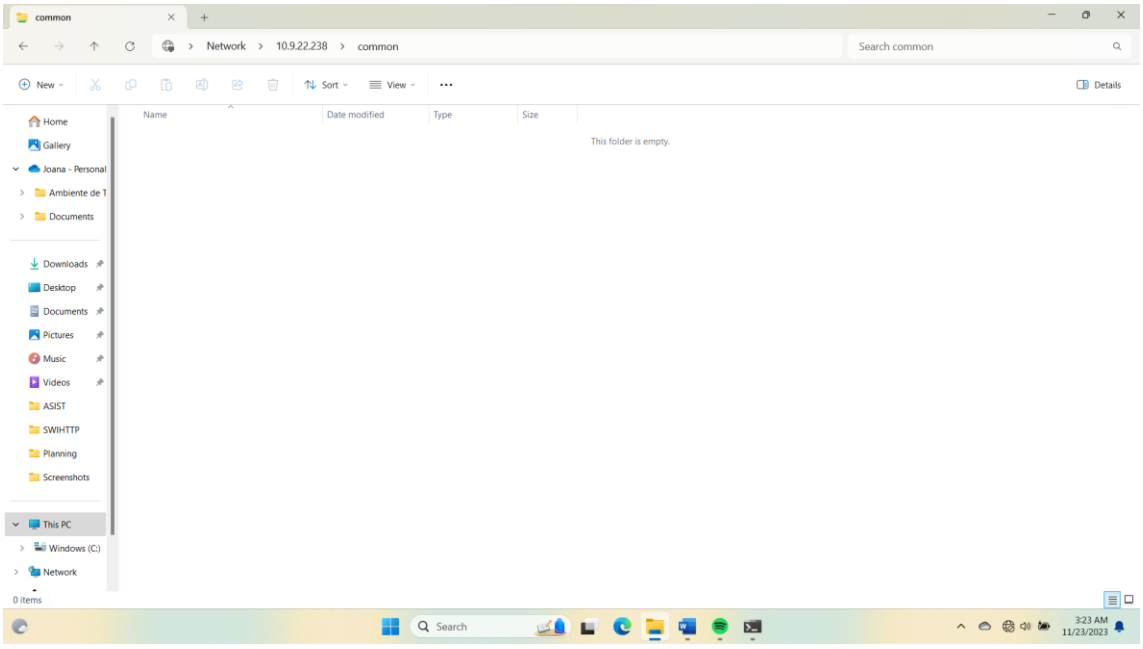
```
[common]
   path = /home/common
   read only = yes
   guest ok = yes
```

Como podemos observar, depois de executar o comando **smbclient -L //10.9.22.238,** a pasta common encontra-se entre as pastas que estão partilhadas.

```
root@vs750:/home# smbclient -L //10.9.22.238
Password for [WORKGROUP\root]:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        common          Disk
        IPC$            IPC       IPC Service (Samba 4.17.12-Debian)
        nobody          Disk      Home Directories
SMB1 disabled -- no workgroup available
root@vs750:/home#
```

Outra maneira de testar seria abrir o Explorador de Ficheiros do Windows e escrever **\\10.9.22.238\common,** e, depois de efetuar a autenticação, devemos poder aceder à pasta common porém sem poder criar ficheiros, editar os já existentes, etc:



Caso queiramos acrescentar uma descrição ou outros atributos devemos considerar a seguinte estrutura no ficheiro smb.conf:

[common]

comment = Public Shared Folder for all users

path = /home/common

browseable = yes

guest ok = yes

writable = yes

create mask = 0664

directory mask = 0775

force user = nobody

force group = nogroup

valid users = @admins

8. Como administrador do sistema quero obter os utilizadores com mais do que 3 acessos incorretos.