

SPRINT 3

Administração de Sistemas

Miguel Oliveira 1211281

Rodrigo Castro 1220636

Rodrigo Cardoso 1221083

Mário Ribeiro 1221019

Instituto Superior de Engenharia do Porto

Índice

Us01 [Miguel Oliveira 1211281]	1
Us02 [Rodrigo Castro 1220636]	4
Us03 [Mário Ribeiro 1221019]	6
Us04 [Mário Ribeiro 1221019]	9
Us05 [Miguel Oliveira 1211281]	11
Us06 [Mário Ribeiro 1221019]	15
Us07 [Rodrigo Cardoso 1221083]	17
Us08 [Rodrigo Castro 1220636]	22
Us09 [Miguel Oliveira 1211281]	26
Us10 [Rodrigo Cardoso 1221083]	29
Us11 [Rodrigo Cardoso 1221083]	34
Us12 [Rodrigo Castro 1220636]	42

Índice de figuras

Figura 1 - script Us03	7
Figura 2 - /etc/crontab	8
Figura 3 - script Us04	10
Figura 4 - etc/crontab db	10
Figura 5 - Exemplo de alerta ao iniciar sessão	14
Figura 6 - US06 script.....	15
Figura 7 - US06 crontab.....	16
Figura 8 - Configuração haproxy	26
Figura 9 - Monitorização de desempenho	27
Figura 10 - Output de criação da chave de acesso SSH.....	30
Figura 11 - Output da chave pública SSH da conta "adminsyst".....	31
Figura 12 - Conteúdo adicionado no ficheiro sshd_config.....	32
Figura 13 - Proibição do acesso direto de contas pela root.....	32
Figura 14 - Teste de acesso SSH á conta do adminsyst	33
Figura 15 - Painel Shares em File and Storage Services	35
Figura 16 - Opção SMB Share - Quick da partilha de ficheiros	35
Figura 17 - Pasta Partilhada PublicShare.....	36
Figura 18 - Menu das permissões da pasta partilhada	37
Figura 19 - Confirmação da criação da pasta partilhada.....	38
Figura 20 - Teste de execução na máquina interna	39
Figura 21 - Acesso á pasta partilhada na máquina interna	39
Figura 22 - Acesso à pasta partilhada a partir de uma máquina remota presente na mesma rede	40
Figura 23 - Confirmação do acesso à pasta partilhada a partir de uma máquina remota	40
Figura 24 backup_db.sh	42
Figura 25 Exemplo de execução do backup_db.sh	43

Us01 [Miguel Oliveira 1211281]

Objetivo

O Plano de Recuperação de Desastres (DRP) tem como principal objetivo minimizar os efeitos de desastres imprevisíveis na organização, reduzindo o tempo de inatividade e as perdas de dados. Através deste plano, é possível estabelecer estratégias para a recuperação eficaz e eficiente em casos de incidentes como incêndios, inundações, falhas de energia, sabotagem, avarias e outros imprevistos que possam afetar a operação.

Objetivos Específicos

- Reduzir ao máximo o tempo de inatividade e a perda de dados em caso de desastre.
- Estabelecer planos de emergência claros e eficientes.
- Otimizar a reposição dos serviços afetados.
- Definir cenários de desastre e os processos de recuperação correspondentes.

Planeamento Inicial e Comitê Executivo de DRP

Foi constituído um comitê executivo para planejar, atualizar e supervisionar a implementação do DRP. Este comitê também tem a função de apoiar as equipas de projeto em assuntos relacionados ao DRP. Os gerentes de projeto devem colaborar com o comitê para concluir o planeamento detalhado e realizar entrevistas para avaliar a segurança e elaborar a análise de impacto no negócio. Além disso, será implementado um programa de formação sobre DRP para a gestão e peças-chave do projeto.

Funções da Equipa de Recuperação de Desastres

As funções necessárias para a implementação e gestão do DRP são as seguintes:

- Administrador: Responsável pela supervisão geral do DRP, garantindo que todas as partes envolvidas cumpram as diretrizes estabelecidas e coordenando as ações em caso de desastre.
- Técnico do Sistema de Monitorização: Responsável pela monitorização contínua dos sistemas e infraestrutura da organização, identificando e alertando para potenciais riscos ou falhas que possam ativar o plano de recuperação.
- Chefe do Comitê Executivo de DRP: Responsável pela liderança do comitê executivo, coordenando as atividades de planeamento e atualizações do DRP, assim como a comunicação com todas as partes envolvidas.

- **Chefe da Equipa de Resposta a Incidentes:** Responsável por coordenar as respostas a incidentes em caso de desastre, organizando a equipa de recuperação e assegurando a execução eficiente do plano de recuperação.

Armazenamento da Documentação do Plano

O plano será armazenado em formato físico e digital em múltiplas localizações seguras dentro da organização. Todos os membros da equipa terão acesso a uma cópia digital, sendo que as cópias físicas estarão disponíveis apenas para a equipa de recuperação após desastre.

Estratégia de Backup

Para garantir a integridade dos dados, será implementada uma estratégia de backup “3,2,1”, ou seja:

- 3 cópias dos dados serão realizadas;
- 2 formatos diferentes serão utilizados para o backup;
- 1 das cópias será armazenada em um local diferente, seja na cloud, em servidor remoto ou em disco externo.

Ativadores do Plano

O DRP será ativado automaticamente quando ocorrer uma das seguintes situações:

- Incapacitação da internet;
- Inundação do edifício;
- Avaria ou perda de dados na base de dados;
- Avaria nos servidores;
- Falha de energia.

Plano de Recuperação

Em caso de desastre, a equipa será dividida em duas subequipes, que trabalharão em turnos de 4 horas. As ações a tomar variam consoante o tipo de desastre:

- **Desastres Naturais:** A equipa de resposta a incidentes deverá contactar as autoridades competentes para controlar a situação e permitir o diagnóstico do problema.
- **Falhas de Hardware:** A equipa deve diagnosticar o problema e, se possível, reparar o equipamento. Caso contrário, será necessária a substituição da infraestrutura.
- **Falhas de Software:** A equipa de TI deverá diagnosticar e corrigir o bug, ou reverter o sistema para a última versão estável.

Recuperação de Dados

- Para as bases de dados, a equipa deverá restaurar os dados a partir do backup mais recente disponível.

Relatório Pós-Desastre

Após o fim do evento de desastre, deverá ser elaborado um relatório detalhado que inclua:

- Informação sobre a emergência, incluindo notificações e ações tomadas;
- Lista das pessoas notificadas da emergência;
- Relatório das ações realizadas pela equipa de recuperação;
- Resultados obtidos com as ações implementadas;
- Lições aprendidas com o incidente.

Atualização e Melhoria do DRP

Após cada incidente, o DRP deverá ser revisto e atualizado com base no formulário pós-desastre. As alterações propostas serão avaliadas para corrigir possíveis lacunas e garantir a eficácia do plano. A melhoria contínua do DRP é essencial para garantir uma resposta robusta a futuros desastres, sendo os testes periódicos uma prática recomendada.

Us02 [Rodrigo Castro 1220636]

1. Introdução Este relatório tem como objetivo apresentar as justificações para as alterações propostas na infraestrutura da organização, com o intuito de garantir um MTD (Maximum Tolerable Downtime) de 20 minutos. O MTD é o tempo máximo aceitável para que um sistema ou serviço fique indisponível sem causar impactos significativos às operações da organização.

2. Contexto e Importância O MTD de 20 minutos exige que a organização esteja preparada para responder rapidamente a falhas, minimizar o tempo de inatividade e garantir a continuidade do negócio. Para cumprir com este requisito, é necessário adotar uma série de medidas de natureza técnica e operacional. Estas mudanças são cruciais para assegurar a resiliência da infraestrutura, prevenindo prejuízos financeiros, perdas de dados e danos à reputação da organização.

3. Justificação para as Mudanças

3.1. Implementação de Soluções de Alta Disponibilidade (HA)

- **Motivação:** Reduzir a probabilidade de interrupções no serviço e garantir que, em caso de falha, um sistema secundário entra em operação automaticamente.
- **Mudança Proposta:** Implementar clusters de servidores redundantes, distribuição de carga (load balancing) e soluções de failover automático.
- **Impacto no MTD:** Caso ocorra uma falha em um servidor, outro assume as operações imediatamente, minimizando o tempo de inatividade.

3.2. Melhorias nos Mecanismos de Recuperação de Desastres (Disaster Recovery)

- **Motivação:** Assegurar a capacidade de restaurar rapidamente os sistemas em caso de falhas catastróficas, como desastres naturais ou ataques informáticos.
- **Mudança Proposta:** Implementar um site de recuperação secundário (disaster recovery site) geograficamente distante e sincronizado em tempo real com o principal.
- **Impacto no MTD:** A possibilidade de transferir as operações para o site de recuperação assegura que o tempo de inatividade seja inferior a 20 minutos.

3.3. Automação dos Processos de Recuperação

- **Motivação:** Reduzir o tempo necessário para executar processos de recuperação manual, que podem ser demorados e suscetíveis a erros humanos.
- **Mudança Proposta:** Adotar scripts automatizados e soluções de orquestração para automatizar o failover de sistemas e a restauração de serviços.
- **Impacto no MTD:** A automação reduz o tempo de resposta e possibilita a recuperação automática, reduzindo o tempo total de inatividade.

3.4. Melhoria na monitorização e na resposta a Incidentes

- **Motivação:** Identificar e corrigir falhas antes que estas se tornem críticas, evitando a indisponibilidade dos serviços.
- **Mudança Proposta:** Adotar ferramentas de monitorização em tempo real que utilizem alertas proativos baseados em anomalias e métricas de desempenho.
- **Impacto no MTD:** A deteção precoce de falhas permite agir rapidamente e evitar que o tempo de inatividade ultrapasse o limite de 20 minutos.

3.5. Reforço da Redundância na Infraestrutura Crítica

- **Motivação:** Evitar pontos únicos de falha (Single Points of Failure) que possam causar a indisponibilidade de serviços críticos.
- **Mudança Proposta:** Implementar conexões de rede duplas e armazenamento replicado.
- **Impacto no MTD:** A eliminação de pontos únicos de falha assegura que a falha de um componente não resulte na interrupção total do sistema.

4. Benefícios Esperados

- **Continuidade do Negócio:** Minimizar o impacto de falhas nos serviços críticos, garantindo operações ininterruptas.
- **Conformidade:** Cumprimento das obrigações contratuais e regulamentares em relação à continuidade de serviço.
- **Satisfação dos Clientes:** Reduzir as interrupções percebidas pelos utilizadores, garantindo um serviço mais estável e confiável.
- **Redução de Prejuízos:** Prevenir perdas financeiras resultantes de paragens prolongadas dos serviços.

5. Conclusão Para garantir o MTD de 20 minutos, é imprescindível implementar mudanças na infraestrutura, incluindo soluções de alta disponibilidade, melhoria nos mecanismos de recuperação de desastres, automação dos processos de recuperação, reforço do monitoramento e eliminação de pontos únicos de falha. Estas ações visam assegurar a continuidade dos serviços críticos e proteger a reputação e a sustentabilidade financeira da organização.

Us03 [Mário Ribeiro 1221019]

Nesta User Story é pedido que como administrador de sistemas seja possível a realização de uma cópia de segurança da(s) DB(s) para um ambiente de Cloud através de um script.

A nossa base de dados de produção atualmente encontra-se implementada nos servidores virtuais do DEI, pelo que o servidor utilizado para realizar esta US será também um servidor virtual do DEI.

Idealmente esta cópia de segurança da base de dados deveria ser feita num servidor localizado fisicamente distante do servidor original, precisamente por questões de segurança. Colocando isto por outras palavras, se, na zona que tem o servidor da base de dados, e que por sua vez, teria também o servidor que possui e faz as cópias de segurança acontece algo inesperado, como por exemplo, se alguém obtiver acesso à sala onde estão localizados os servidores e danificar os equipamentos, todo o conteúdo será perdido. Desta forma, a cópia de segurança consegue mitigar apenas um conjunto específico de problemas, mas não oferece proteção total ou próxima disso. Por isso, é essencial que a cópia de segurança da base de dados seja armazenada e realizada em um servidor localizado fisicamente distante do servidor em produção.

Como estamos a desenvolver um projeto num contexto académico não existirá um grande problema em fazermos desta forma, no entanto é de referir que um bom administrador de sistemas tem de pensar em todos os casos possíveis e que em qualquer outro contexto seria um erro enorme deixar isto passar.

Dado este contexto, vamos agora apresentar como esta US foi implementada. No sprint anterior, já tínhamos uma solução para realizar cópias de segurança da base de dados, mas essa abordagem não cumpria os requisitos definidos para esta US. Atualmente, temos uma cópia de segurança incremental durante os dias da semana e total ao domingo (todas estas ocorrem pelas 4:30 da manhã). Agora, passaremos a realizar uma cópia de segurança total diariamente, seguindo o padrão exigido pela própria US. Foi definido que o formato dos backups deverá ser “<db_name>_yyyymmdd” onde “db_name” corresponderá ao nome da base de dados, “yyyy” ao ano de realização da cópia, “mm” ao mês de realização da cópia e “dd” ao dia de realização da cópia.

Para implementar esta US, começamos pela criação do script necessário para atender aos requisitos estabelecidos. Assim, foi criado o script no caminho “/etc/daily_database_backup.sh” utilizando o comando touch. De seguida, ajustaram-se as permissões para garantir que apenas o utilizador que criou o script (neste caso, o utilizador

root) pudesse lê-lo, escrevê-lo e executá-lo. Para isso, utilizou-se o comando “**chmod 700 /etc/daily_database_backup.sh**”.

Depois foi escrito o código necessário no script, código este que poderá ser observado na seguinte imagem.

```
#!/bin/bash
db_backup_dir=/etc/daily_database_backup
db_dump_dir=/etc/daily_database_dump
db_backup_logs_dir=/etc/daily_database_backup/logs

mkdir -p $db_backup_logs_dir

today_date=$(date +%Y%m%d)
db_log_file=$db_backup_logs_dir/sem5g5isep_$today_date.log

mkdir -p $db_dump_dir
chmod -R 700 $db_dump_dir

echo "Iniciando mongodump..." >> "$db_log_file"
mongodump --uri="mongodb+srv://rodrigocastro2004:projetoIaprsen5@sem5g5isep.gtdw4.mongodb.net/?retryWrites=true&majority=&appName=sem5g5isep" --out="$db_dump_dir" 2>&1 | tee "$db_log_file"

if [ $? -eq 0 ]; then
    mkdir -p $db_backup_dir
    chmod -R 700 $db_backup_dir
    tar -czvf "$db_backup_dir/sem5g5isep_$today_date.tgz" -C "$db_dump_dir" . >> "$db_log_file" 2>&1
    if [ $? -eq 0 ]; then
        echo "✅ O backup foi realizado com sucesso." >> $db_log_file
        rm -rf $db_dump_dir
    else
        echo "❌ O backup falhou. Ocorreu um erro ao realizar o comando tar." >> $db_log_file
    fi
else
    echo "❌ O backup falhou. Ocorreu um erro no mongodump." >> $db_log_file
fi
```

Figura 1 - script Us03

Primeiramente, definimos variáveis contendo os caminhos das pastas necessárias para a realização do dump da base de dados e do backup. Estas pastas são criadas ao longo do script caso ainda não existam, utilizando o comando “**mkdir -p /path/to/(nome)**”. Ao criar estes caminhos de pastas, também são alteradas as permissões para que apenas o *owner* do script consiga ler, escrever e executar coisas nessas pastas, sendo que para isso utiliza-se o comando “**chmod -R 700 /path/to/(nome)**”.

Para garantir o formato necessário para o nome dos backups, foi criada uma variável que contem essa informação, permitindo sua reutilização sempre que necessário.

O script também realiza o dump da base de dados para um diretório específico, com os outputs gerados por este comando sendo registados num ficheiro de logs correspondente. Caso o dump seja concluído com sucesso, os ficheiros gerados são copiados para o backup. Caso contrário, é registado nos logs um aviso sobre o ocorrido, e o programa é encerrado.

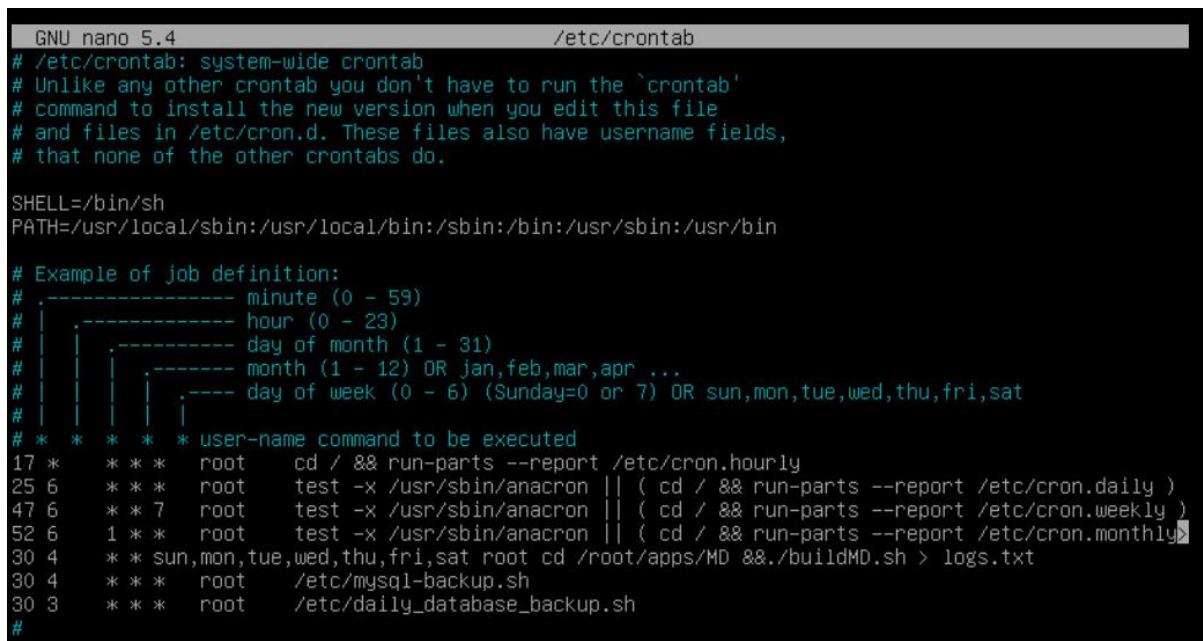
Caso a cópia de segurança realizada com o comando tar seja bem-sucedida, uma mensagem de sucesso é registada no ficheiro de logs, caso contrário, é registada uma mensagem de erro. Quando a cópia de segurança é concluída com sucesso, a pasta que contém o dump da base de dados é removida, pois não será mais necessária.

O comando tar é utilizado com os seguintes parâmetros:

- '-c': Cria um novo ficheiro.
- '-z': Utiliza a compressão gzip.
- '-v': Ativa a saída detalhada (verbose), exibindo o progresso do comando na consola enquanto o ficheiro é criado.
- '-p': Preserva as permissões dos ficheiros e pastas na pasta comprimida.
- '-f': Especifica o nome do ficheiro comprimido a ser criado, incluindo o caminho onde será armazenado.

Após a criação do script, foi utilizado o comando "**bash -n /etc/daily_database_backup.sh**" para verificar se não havia erros de sintaxe. De seguida, o script foi executado para garantir que tudo estava a funcionar conforme o esperado.

Agora, é necessário que o script seja executado diariamente para garantir uma cópia de segurança diária. Com isso, a cópia de segurança terá um *Recovery Point Objective* (RPO) de 24h, o que significa que a perda de dados pode ser de no máximo 24 horas. Essa perda de dados pode ocorrer entre as 3h30min da madrugada, momento em que o backup da base de dados é realizado. Para agendar a execução diária do script, foi editado o ficheiro "**/etc/crontab**" e adicionada a última linha antes do símbolo "#", conforme mostrado na imagem seguinte.



```
GNU nano 5.4 /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
30 4 * * sun,mon,tue,wed,thu,fri,sat root cd /root/apps/MD &&./buildMD.sh > logs.txt
30 4 * * * root /etc/mysql-backup.sh
30 3 * * * root /etc/daily_database_backup.sh
#
```

Figura 2 - /etc/crontab

Assim, podemos ver que o script criado será executado diariamente às 3h30min da madrugada, e a sua execução será iniciada pelo utilizador root.

Us04 [Mário Ribeiro 1221019]

Nesta User Story é pedido que os backups realizados na User Story anterior (US3), sejam guardados conforme uma política de retenção que mantenha 1 backup por mês para os últimos 12 meses, 1 backup por semana para as últimas 4 semanas e 1 backup por dia para os últimos 7 dias. Esta abordagem assegurará uma gestão eficiente dos backups, otimizando desta forma o uso de espaço em disco e garantindo a disponibilidade das cópias mais relevantes.

O primeiro passo para atender à User Story foi a criação do script “*save_database_backup.sh*”, responsável por gerir os backups gerados pelo sistema. Este script executa as seguintes tarefas principais:

Movimentação de Backups: O script move o backup diário realizado para o diretório de backups salvos, garantindo que os arquivos sejam arquivados de acordo com o cronograma de retenção.

Exclusão de Arquivos Antigos: O script também é responsável pela limpeza dos backups antigos:

- Diários com mais de 7 dias;

- Mensais com mais de 1 ano;

- Semanais com mais de 1 mês.

Estes serão removidos, de acordo com as políticas definidas.

Para além disso, logs são gerados a cada execução, registando se o backup foi movido com sucesso ou não. Esse registo ajuda a monitorizar a integridade do processo e a detetar problemas rapidamente.

Para garantir que o script seja executado de forma automática e dentro do cronograma necessário, configuramos o *crontab* para agendar a execução do script com a seguinte periodicidade:

```
#!/bin/bash

# Definir diretórios e formatos de arquivos
source_directory="/etc/daily_database_backup"
destination_directory="/etc/saved_database_backups"
current_date=$(date +%Y%m%d)
file_format="sem5g55isep_${current_date}.tgz"
log_directory="${destination_directory}/logs"
mkdir -p "${log_directory}"

# Arquivo de log
log_file="${log_directory}/move_file_log_${current_date}.txt"

# Excluir backups diários com mais de 7 dias
find "${source_directory}" -name "sem5g55isep*.tgz" -type f -mtime +7 -exec rm -f {} \;
echo "$(date +%Y-%m-%d %H:%M:%S) - Backups diários mais antigos que 7 dias excluídos." >> "${log_file}"

# Excluir backups mensais mais antigos que 1 ano
find "${destination_directory}" -name "sem5g55isep*.tgz" -type f -mtime +365 -exec rm -f {} \;
echo "$(date +%Y-%m-%d %H:%M:%S) - Backups mensais mais antigos que 1 ano excluídos." >> "${log_file}"

# Excluir backups semanais mais antigos que 1 mês
find "${destination_directory}" -name "sem5g55isep*.tgz" -type f -mtime +30 -exec rm -f {} \;
echo "$(date +%Y-%m-%d %H:%M:%S) - Backups semanais mais antigos que 1 mês excluídos." >> "${log_file}"

# Verificar se o arquivo de backup diário existe
if [ -e "${source_directory}/${file_format}" ]; then
    # Mover o backup diário para o diretório de backups salvos
    mv "${source_directory}/${file_format}" "${destination_directory}/"
    echo "$(date +%Y-%m-%d %H:%M:%S) - Arquivo movido com sucesso." >> "${log_file}"
else
    echo "$(date +%Y-%m-%d %H:%M:%S) - Arquivo não encontrado: ${source_directory}/${file_format}" >> "${log_file}"
fi
```

Figura 3 - script Us04

Mensalmente: No primeiro dia de cada mês, às 5:30 da manhã, o script será executado para mover os backups mensais.

Semanalmente: Todo domingo, às 5:30 da manhã, o script será executado para lidar com os backups semanais.

Diariamente: O script também é executado todos os dias, às 5:30 da manhã, para mover os backups diários e realizar a limpeza.

Desta forma, o cronograma de 1 backup por mês, 1 backup por semana, e 1 backup por dia na última semana é devidamente cumprido (últimas 3 linhas antes do #).

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
30 4 * * sun,mon,tue,wed,thu,fri,sat root cd /root/apps/MD && ./buildMD.sh > logs.txt
30 4 * * * root /etc/mysql-backup.sh
30 3 * * * root /etc/daily_database_backup.sh
#Diario
30 5 * * * root /etc/save_database_backup.sh
#Mensal
30 5 1 * * root /etc/save_database_backup.sh
#Semanal
30 5 * * 0 root /etc/save_database_backup.sh
#
```

Figura 4 - etc/crontab db

Us05 [Miguel Oliveira 1211281]

Descrição:

Como administrador de sistemas, pretendo que o processo de cópia de segurança da base de dados seja registado nos logs do sistema Linux de forma adequada e que, em caso de falha crítica, o administrador seja notificado automaticamente ao aceder à consola.

Implementação:

Para garantir o correto acompanhamento dos processos críticos, como a cópia de segurança da base de dados, foi implementado um sistema de monitorização através de logs no Linux. Os registos permitem identificar falhas rapidamente, facilitando o diagnóstico e a resolução dos problemas, bem como detetar situações anómalas que podem, a longo prazo, comprometer o desempenho do sistema.

1. Instalação do Sistema de Logs

Na máquina virtual em questão, não existia previamente um sistema de registo de eventos configurado. Foi então instalado o **RSYSLOG**, um serviço de registo amplamente utilizado no Linux:

```
sudo apt install rsyslog
```

2. Configuração do RSYSLOG

O sistema de registos no Linux utiliza os atributos **facility** (origem do evento) e **severity** (grau de gravidade) para classificar os logs. A configuração é armazenada no ficheiro `/etc/rsyslog.conf`.

Por padrão, o **RSYSLOG** já escreve os eventos da **facility cron** (tarefas agendadas) no ficheiro `/var/log/cron.log`. Para melhorar a organização e o controlo das mensagens geradas durante o processo de backup, foram definidos três ficheiros distintos:

- `/var/log/cron_info.log`: Para registos com severidade *info* e superior.
- `/var/log/cron_emerg.log`: Para registos críticos com severidade *emerg*.
- `/var/log/special_cron_emerg.log`: Para armazenar temporariamente eventos críticos e alertar o administrador no login.

Os ficheiros foram criados utilizando o comando:

```
touch /var/log/cron_info.log /var/log/cron_emerg.log /var/log/special_cron_emerg.log
```

Foram então adicionadas as seguintes regras ao ficheiro `/etc/rsyslog.conf`:

```
cron.info    /var/log/cron_info.log
cron.emerg   /var/log/cron_emerg.log
cron.emerg   /var/log/special_cron_emerg.log
```

Estas configurações asseguram que:

- Os eventos informativos e superiores são registados em **cron_info.log**.
- As falhas graves são registadas simultaneamente em **cron_emerg.log** e **special_cron_emerg.log**.

Para aplicar as alterações, o serviço **RSYSLOG** foi reiniciado:

```
systemctl restart rsyslog
```

3. Alteração do Script de Backup

O script responsável pela cópia de segurança da base de dados (`/etc/daily_database_backup.sh`) foi atualizado para registar os eventos diretamente no sistema de logs. Para tal, foi utilizado o comando **logger** com a severidade apropriada em cada etapa do processo:

- Para um backup bem-sucedido:
`logger -p cron.info -t "BackupDB" "Backup da base de dados concluído com sucesso."`
- Para uma falha crítica:
`logger -p cron.emerg -t "BackupDB" "Falha grave no processo de backup!"`

A opção **-p** define a prioridade (facility e severity), enquanto **-t** atribui uma etiqueta identificativa ao evento.

4. Alerta Automático no Login

Para notificar o administrador no momento do login, foi criada uma script auxiliar chamada **/etc/check_critical_logs.sh**. Esta script verifica o ficheiro **special_cron_emerg.log** e apresenta o alerta com as falhas registadas:

```
#!/bin/bash
if [ -s /var/log/special_cron_emerg.log ]; then
    echo "⚠ ALERTA: Foram detetadas falhas críticas no processo de backup! ⚠ "
    cat /var/log/special_cron_emerg.log
    > /var/log/special_cron_emerg.log # Limpar o ficheiro após apresentação
fi
```

As permissões da script foram ajustadas para que apenas o utilizador root a possa executar:

```
chmod 700 /etc/check_critical_logs.sh
```

Por fim, a execução da script foi integrada no ficheiro **/etc/profile**. Isto garante que a verificação é realizada sempre que o administrador acede à consola:

```
if [ $(id -u) -eq 0 ]; then
    /etc/check_critical_logs.sh
fi
```

5. Testes e Validação

Para validar a implementação, foram simulados eventos de falha crítica com o comando:

```
logger -p cron.emerg "Simulação de falha crítica no backup."
```

Após sair da consola e voltar a iniciar sessão, verificou-se que o alerta foi exibido corretamente, contendo a quantidade de falhas e as mensagens registadas. O ficheiro **special_cron_emerg.log** foi limpo automaticamente, garantindo que o aviso não se repetia desnecessariamente.


```
miguel — ssh root@uvm055.dei.isep.ipp.pt — root@uvm055.dei.isep.ipp.pt — ssh root@uvm055.dei...
[root@uvm055:~# logger -p cron.emerg "Simulação de falha crítica no backup."
Message from syslogd@uvm055 at Dec 14 09:50:47 ...
root: Simulação de falha crítica no backup.
root@uvm055:~#
Broadcast message from systemd-journald@uvm055 (Sat 2024-12-14 09:50:47 GMT):

root[710101]: Simulação de falha crítica no backup.

[root@uvm055:~# logout
Connection to uvm055.dei.isep.ipp.pt closed.
> ssh root@uvm055.dei.isep.ipp.pt
[root@uvm055.dei.isep.ipp.pt's password:
Bem-vindo,root ao sistema Linux do grupo 55
Data e Hora: Saturday, 14 de December de 2024, 09:50:59
Desejamos-lhe uma boa estadia!

"⚠ALERTA: Foram detetadas falhas críticas no processo de backup! ⚠
Dec 14 09:50:47 uvm055 root: Simulação de falha crítica no backup.
root@uvm055:~#
```

Figura 5 - Exemplo de alerta ao iniciar sessão

Conclusão:

Com esta configuração, o processo de backup da base de dados é registado de forma organizada e eficaz no sistema de logs do Linux. Em caso de falha grave, o administrador é alertado automaticamente ao iniciar sessão, facilitando a deteção e resolução de problemas.

Us06 [Mário Ribeiro 1221019]

A User Story (US) pede que os backups realizados no sistema da base de dados não ultrapassem 7 dias de vida, exceção feita para as cópias mensais e anuais, que devem ser preservadas. O objetivo principal é garantir que os backups diários sejam removidos após este período.

Para atender a esta demanda, foi desenvolvido um script em *bash* denominado “*remove_old_backups.sh*”. Este script é responsável por encontrar e remover todos os arquivos de backup com mais de 7 dias de idade, garantindo que o armazenamento não seja sobrecarregado com backups antigos desnecessários.

Este script utiliza o comando *find* para localizar os arquivos de backup do tipo *sem5g55isep_* que foram modificados há mais de 7 dias. Os arquivos identificados são então removidos utilizando o comando *rm -f*, garantindo que os backups mais antigos sejam eliminados automaticamente, evitando o acúmulo de arquivos desnecessários e otimizando o uso do armazenamento.

```
#!/bin/bash

# Definir a pasta de backups e o número de dias
backup_folder="/etc/daily_database_backup"
days_threshold=7

# Função para verificar se um arquivo é semanal (domingo) ou mensal (dia 1)
is_special_backup() {
    local filename="$1"

    # Extrair a data do nome do arquivo (assumindo formato sem5g55isep_YYYYMMDD)
    local file_date=$(echo "$filename" | grep -oE "[0-9]{8}")

    # Validar se a data foi extraída corretamente
    if [[ -z "$file_date" ]]; then
        return 1 # Não é um backup válido
    fi

    # Verificar se é mensal (dia 1 do mês)
    if [[ "${file_date:6:2}" == "01" ]]; then
        return 0
    fi

    # Verificar se é semanal (domingo)
    local day_of_week=$(date -d "$file_date" +%u)
    if [[ "$day_of_week" == "7" ]]; then
        return 0
    fi

    return 1
}

# Remover backups antigos (exceto semanais e mensais)
find "$backup_folder" -type f -name "sem5g55isep*" -mtime +$days_threshold | while read -r backup_file; do
    if ! is_special_backup "${basename "$backup_file"}"; then
        rm -f "$backup_file"
    fi
done

# Registro no log
echo "$(date '+%Y-%m-%d %H:%M:%S') - Limpeza de backups concluída. Backups antigos (exceto semanais/mensais) foram removidos." >> /var/log/remove_old_backups.log
```

Figura 6 - US06 script

Para garantir que o script seja executado automaticamente, o agendamento foi feito via *crontab*. O script *remove_old_backups.sh* foi configurado para rodar todos os dias às 5:30 da manhã, usando a seguinte entrada no arquivo de configuração do *crontab* (última linha antes do #):

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
30 4 * * sun,mon,tue,wed,thu,fri,sat root cd /root/apps/MD && ./buildMD.sh > logs.txt
30 4 * * * root /etc/mysql-backup.sh
30 3 * * * root /etc/daily_database_backup.sh
#Diario
30 5 * * * root /etc/save_database_backup.sh
#Mensal
30 5 1 * * root /etc/save_database_backup.sh
#Semanal
30 5 * * 0 root /etc/save_database_backup.sh
30 5 * * * root /etc/remove_old_backups.sh
#
```

Figura 7 - US06 crontab

Us07 [Rodrigo Cardoso 1221083]

No âmbito desta *User Story*, como administrador da organização, quero que seja apresentado uma *Business Impact Analysis* (BIA) da solução final, onde é possível adaptar com os riscos já citados no Sprint anterior.

Identificação e Avaliação dos Riscos

Durante o desenvolvimento da solução final, foram identificados os seguintes riscos no Sprint anterior, classificados com base na sua **possibilidade, impacto e risco** associado:

Ameaça	Possibilidade	Impacto	Risco
Avaria do Sistema	1 (Improvável)	4 (Catastrófico)	4 (Médio)
Avaria dos Servidores	2 (Remoto)	3 (Crítico)	6 (Médio)
Falhas de autenticação (Acesso indevido)	3 (Ocasional)	3 (Crítico)	9 (Sério)
Perda da Base de Dados	1 (Improvável)	4 (Catastrófico)	4 (Médio)
<i>Leak</i> de Dados Sensíveis	2 (Remoto)	3 (Crítico)	6 (Médio)
Vulnerabilidades do Sistema	2 (Remoto)	4 (Catastrófico)	8 (Sério)

Impacto nos Processos de Negócio

Os riscos identificados possuem **implicações diretas** na continuidade e eficiência dos processos na aplicação do *System Appointment and Resource Management* (SARM), onde impacta operações críticas, alocação de recursos e experiência de utilizador. Serão apresentados os seguintes impactos dos processos de cada risco/ameaça:

1. Avaria do Sistema:

- Afeta a capacidade do sistema de gerir todos os dados da aplicação e de alocar recursos de forma eficiente.
- Resulta em atrasos no atendimento, perda de produtividade e insatisfação dos utilizadores.
- Pode causar falhas em notificações e atualizações críticas, como confirmações de alteração de dados ou pedidos de remoção de dados pessoais.

2. Avaria dos Servidores

- Reduz a disponibilidade do sistema, impossibilitando o acesso a registos de alocação de cirurgias.
- Compromete a capacidade e reprogramação dinâmica de recursos, como a troca de horários ou salas de cirurgias.
- Impacta diretamente na experiência com os pacientes e a coordenação interna entre os administradores e os profissionais de saúde.

3. Falhas de autenticação (Acesso indevido)

- Pode permitir que utilizadores não autorizados possam aceder a informações sensíveis, como dados pessoais dos pacientes e detalhes dos recursos agendados.
- Compromete a integridade do sistema, assim permitindo alterações indesejadas em agendamentos e alocação de recursos.
- Abala a confiança dos *stakeholders* no sistema.

4. Perda da Base de Dados

- Elimina dados essenciais para a gestão de cirurgias, histórico de pacientes e dados de alocação.
- Paralisa operações, onde impossibilita-se novos agendamentos.
- Viola regulamentações de proteção de dados e continuidade de registos médicos, onde poderá ser acarretado processos legais.

5. Leak de Dados Sensíveis

- Exposição de informação confidencial, como registos de saúde dos pacientes, compromete a privacidade e pode levar a consequências legais e financeiras.
- Danifica a reputação da organização e mina a confiança dos utilizadores.
- Requer esforços substanciais para mitigar os danos e reconquistar a credibilidade.

6. Vulnerabilidades do Sistema

- Exploração de falhas permite ciberataques que comprometem a integridade, disponibilidade e confidencialidade do sistema.
- Paralisa operações críticas, incluindo a gestão de horários, salas, médicos e pacientes.
- Aumenta os custos de manutenção e recuperação, além de prejudicar a experiência geral dos utilizadores.

Proposta de Mitigação e Adaptação

1. Avaria do Sistema

Adaptação

- Implementar monitoramento contínuo de performance para identificar anomalias antes que ocorram falhas.

Mitigação

- Desenvolver e testar regularmente um Disaster Recover Plan (DRP) para reduzir o tempo de inatividade e restaurar os serviços rapidamente.

2. Avaria dos Servidores

Adaptação

- Adotar uma infraestrutura híbrida ou baseada em cloud para aumentar a resiliência e escalabilidade.

Mitigação

- Realizar cópias de segurança (backups) regularmente de todas as informações críticas e a configuração de tratamento de erros automáticos para manter o sistema operacional em caso de falhas.

3. Falhas de autenticação (Acesso indevido)

Adaptação:

- Implementar a Autenticação de Multifator (MFA) para garantir que apenas utilizadores autorizados possam aceder o sistema.

Mitigação:

- Definir políticas rigorosas de palavra-passe, onde exige-se complexidade e mudanças periódicas.

4. Perda de Base de Dados

Adaptação:

- Configurar backups incrementais automáticos e armazenar cópias de segurança em locais distintos para maior resiliência.

Mitigação:

- Realizar simulações regulares de recuperação de dados para validar a eficácia do plano de contingência.

5. *Leak* de Dados Sensíveis

Adaptação:

- Adotar algoritmos de criptografia para todos os dados pessoais/sensíveis em repouso ou em trânsito.

Mitigação:

- Configurar sistemas de prevenção de violação de dados (DLP – *Data Loss Prevention*).

6. Vulnerabilidades do Sistema

Adaptação:

- Implementar um ciclo contínuo de atualizações de segurança e realizar testes de penetração regulares.

Mitigação:

- Adotar um processo formal de gestão de *patches* (soluções rápidas) para corrigir vulnerabilidades de forma eficiente.

As adaptações e medidas propostas **fortalecem a resiliência** da aplicação *System Appointment and Resource Management* (SARM), o que faz **reduzir a probabilidade de interrupções e aumentar a proteção da integridade dos dados e dos processos operacionais**. Essas iniciativas contribuem para a continuidade do negócio para a confiança dos utilizadores e stakeholders na solução.

Us08 [Rodrigo Castro 1220636]

1. Objetivo da Gestão de Acessos no Sistema

Na nossa aplicação, a **gestão de acessos** será implementada com o objetivo de garantir que apenas utilizadores autorizados possam aceder aos recursos e funcionalidades adequados, com base nas permissões atribuídas a cada utilizador. Isto permitirá proteger os dados sensíveis, garantir a integridade do sistema e assegurar a conformidade com normas de segurança, como o RGPD.

2. Medidas a implementar

1. Autenticação de Utilizadores

Para garantir que apenas utilizadores legítimos acessem ao sistema, vamos implementar as seguintes medidas:

- **Login Seguro:**
 - Os utilizadores terão de se autenticar com um nome de utilizador (e-mail) e uma palavra-passe forte.
 - Será implementada a obrigatoriedade de uma palavra-passe forte, com requisitos de comprimento mínimo (8 caracteres) e o uso de letras maiúsculas, minúsculas, números e caracteres especiais.
 - Caso um utilizador introduza uma palavra-passe errada mais de 5 vezes, a conta será temporariamente bloqueada durante **15 minutos** e o administrador de sistema irá ser notificado para prevenir ataques de força bruta.
- **Autenticação Multifator (MFA):**
 - Para contas de administrador ou para funções críticas, implementar o **MFA (Multifactor Authentication)**.
 - O MFA será feito através de um código enviado por e-mail ou SMS para o utilizador, além da palavra-passe.
- **Recuperação de Palavra-Passe:**
 - Caso o utilizador se esqueça da palavra-passe, ele poderá solicitar uma recuperação.
 - Será enviado um link de recuperação para o e-mail do utilizador, com validade de **24 horas**, permitindo-lhe redefinir a palavra-passe.

2. Autorização de Utilizadores

Para controlar o que os utilizadores podem fazer no sistema,

Role-Based Access Control (RBAC), ou seja, controlo de acessos baseado em papéis.

- **Papéis Definidos no Sistema:**
 - **Administrador:** Acesso total ao sistema e capacidade de gerir todos os utilizadores.
 - **Médico:** Pode criar, editar e visualizar pedidos de operações e aceder a registos médicos dos seus pacientes.
 - **Enfermeiro:** Pode visualizar e atualizar informações operacionais relacionadas com os pacientes atribuídos.
 - **Paciente:** Apenas pode aceder ao seu próprio registo médico e visualizar marcações de consultas.
- **Implementação de Permissões:**
 - Cada papel terá permissões claramente definidas. Por exemplo, apenas o administrador pode criar novas contas de utilizador ou redefinir o papel de um utilizador.
 - O sistema verificará o papel do utilizador em cada solicitação de API. Utilizando **claims no JWT** (JSON Web Token), as permissões do utilizador serão incorporadas no token de autenticação, para que cada solicitação seja validada de forma automática.
- **Princípio do Menor Privilégio (PoLP):**
 - Cada utilizador só terá acesso às funcionalidades que realmente precisa para executar as suas tarefas.

3. Controlo de Sessões

Para controlar as sessões de utilizadores sistema, serão aplicadas as seguintes medidas:

- **Expiração de Sessões Inativas:**
 - Se o utilizador estiver inativo por **30 minutos**, a sessão será encerrada automaticamente.
 - Isso será implementado controlando o JWT (JSON Web Token) do utilizador, incluindo um campo de "expiração" que determina a validade do token.

4. Auditoria e Monitorização

A monitorização e registo de acessos ao sistema é essencial para identificar atividades anómalas e garantir a conformidade com regulamentos de segurança.

- **Registo de Logs de Acesso:**

- Todas as tentativas de login (com sucesso ou falha) serão registadas no sistema.
- Para cada tentativa, o sistema registará a data, hora, endereço IP e o utilizador associado.
- Este registo permitirá identificar ataques de força bruta ou atividades suspeitas.

- **Registo de Atividades Sensíveis:**

- Quando um utilizador alterar os seus dados pessoais ou quando um administrador alterar permissões de utilizador, será registada uma entrada no log.
- Assim, será possível rastrear quem fez alterações críticas no sistema.

5. Preocupações de Segurança

Para proteger o sistema contra ataques e abusos, vamos considerar as seguintes preocupações de segurança:

- **Proteção Contra Ataques de Força Bruta:**

- Implementar o bloqueio temporário de contas após múltiplas tentativas falhadas.
- Adicionar **CAPTCHAs** após várias tentativas de login falhadas para garantir que os pedidos são de humanos e não de bots.

- **Proteção de Palavras-Passe:**

- As palavras-passe dos utilizadores serão armazenadas de forma segura, usando algoritmos de hash.
- O sistema nunca armazenará palavras-passe em texto simples.

6. Controlo de Acesso ao Hardware

Para proteger os equipamentos e recursos críticos que suportam a nossa aplicação, como os servidores, serão adotadas as seguintes medidas de controlo de acesso ao hardware:

- **Acesso Restrito**

- O acesso aos servidores será exclusivo para o administrador do sistema, sendo ele o único a possuir as credenciais root de cada servidor.

- **Acesso Remoto Seguro**

- O acesso remoto aos servidores será feito exclusivamente através de SSH

- Para aceder via SSH, será necessária a autenticação com palavra-passe
- **VPN Obrigatória**
 - Para reforçar a segurança, o acesso remoto aos servidores só será possível através de uma VPN (Virtual Private Network) do DEI, o que significa que apenas dispositivos conectados à rede interna da VPN terão permissão de acesso.
 - Isto impede que conexões externas e não autorizadas sejam aceites, adicionando uma camada extra de proteção ao sistema.

3. Conclusão

A gestão de acessos no sistema garante a **proteção de dados sensíveis**, a **integridade do sistema** e a **conformidade com o RGPD**. Para isso, serão implementados mecanismos de **autenticação forte**, **RBAC (controlo de acessos por papéis)** e **Princípio do Menor Privilégio (PoLP)**. Sessões inativas serão encerradas automaticamente, e a **auditoria de acessos** permitirá rastrear ações críticas. O acesso aos servidores será restrito ao administrador, com acesso remoto apenas via **SSH e VPN**. Estas medidas garantem um sistema mais seguro e controlado.

Us09 [Miguel Oliveira 1211281]

Objetivo

Na User Story 9, o objetivo principal foi implementar um sistema de clustering para garantir resiliência, escalabilidade e alta disponibilidade da aplicação. Até ao sprint anterior, a aplicação operava apenas num único servidor (vs584). Neste sprint, expandimos a infraestrutura para incluir mais um servidor principal (vs373) e introduzimos o servidor vs442 como gestor do cluster. No vs442, instalámos e configurámos o HAProxy para realizar o balanceamento de carga e a gestão de failover.

Configuração do HAProxy

1. Instalação do HAProxy

No servidor gestor (vs442), instalámos o HAProxy utilizando os seguintes comandos:

```
sudo apt-get update && sudo apt-get install haproxy
```

2. Configuração do HAProxy

Editámos o ficheiro principal de configuração (haproxy.cfg) com o comando:

```
nano /etc/haproxy/haproxy.cfg
```

A configuração finalizada ficou assim:

```
# Frontend Configuration
frontend http_proxy_frontend
    bind 10.9.21.186:80
    stats enable
    mode http
    option httpclose
    option forwardfor

    acl not_enough_capacity nbsrv(http_proxy_backend) le 1
    use_backend http_proxy_backend_backup if not_enough_capacity

    default_backend http_proxy_backend

# Backend Configuration
backend http_proxy_backend
    balance roundrobin
    server server1 10.9.22.72:2226 check
    server server2 10.9.21.117:2226 check

backend http_proxy_backend_backup
    balance roundrobin
    option allbackups
    server backup_server1 10.9.22.72:2226 check backup
    server backup_server2 10.9.21.117:2226 check backup
```

Figura 8 - Configuração haproxy

Ressalvas na Configuração

Inicialmente, considerámos a possibilidade de incluir um terceiro servidor de backup. Contudo, como o ambiente atual inclui apenas dois servidores principais (vs584 e vs373), optámos por uma arquitetura simplificada, onde ambos os servidores são utilizados para balanceamento de carga, sem a necessidade de backups adicionais.

Validação e Aplicação da Configuração

Para garantir que a configuração do HAProxy era válida, utilizámos o seguinte comando:

```
haproxy -c -f /etc/haproxy/haproxy.cfg
```

Depois de validar a configuração, reiniciámos o HAProxy para aplicar as alterações:

```
service haproxy restart
```

Monitorização do Desempenho

Ativámos a interface de monitorização do HAProxy, acessível através do caminho `/haproxy?stats`. Observámos os seguintes comportamentos:

- Ambos os servidores principais (vs584 e vs373) estavam operacionais no backend principal (`http_proxy_backend`).
- O tráfego foi distribuído de forma equilibrada entre os dois servidores, conforme configurado.

HAProxy version 2.2.9-2+deb11u6, released 2023/12/23

Statistics Report for pid 2067

> General process information

pid = 2067 (process #1, nproc = 1, nthread = 2)
uptime = 0d 1h31m57s
system limits: memmax = unlimited; ulimit-n = 1048576
maxsock = 1048576; maxconn = 524288; maxpipes = 0
current conn = 2; current pipes = 0; conn rate = 2/sec; bit rate = 0.000 kbps
Running tasks: 1/17; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
Note: "NOLEBYDRAIN" = UP with load-balancing disabled.

Display option:
• Scope:
• Hide "DOWN" servers
• Refresh now
• CSV export
• JSON export (schema)

External resources:
• [HAProxy site](#)
• [Upstream v2.2](#)
• [Online manual](#)

http_proxy_frontend																			
Queue				Session rate				Sessions				Bytes				Denied			
Cur	Max	Limit		Cur	Max	Limit		Cur	Max	Limit		In	Out	Req	Resp	Req	Conn	Resp	Errors
Frontend	2	6	-	2	3			524 286	25			8 080	340 744	0	0	5			
http_proxy_backend																			
Queue				Session rate				Sessions				Bytes				Denied			
Cur	Max	Limit		Cur	Max	Limit		Cur	Max	Limit		In	Out	Req	Resp	Req	Conn	Resp	Errors
server1	0	0	-	0	1			0	1			790	214	0	0	0	0	0	0
server2	0	0	-	0	1			0	1			872	514	0	0	0	0	0	0
Backend	0	0	-	0	2			52 427	4			1 662	728	0	0	0	0	0	0
http_proxy_backend_backup																			
Queue				Session rate				Sessions				Bytes				Denied			
Cur	Max	Limit		Cur	Max	Limit		Cur	Max	Limit		In	Out	Req	Resp	Req	Conn	Resp	Errors
backup_server1	0	0	-	0	0			0	0			0	0	0	0	0	0	0	0
backup_server2	0	0	-	0	0			0	0			0	0	0	0	0	0	0	0
Backend	0	0	-	0	0			0	0			0	0	0	0	0	0	0	0

Figura 9 - Monitorização de desempenho

Testes de Balanceamento de Carga

Executámos múltiplos pedidos ao servidor utilizando o seguinte comando:

```
while true; do curl 10.9.21.166:80; sleep 1; done
```

Ao monitorizar a interface de estatísticas, verificámos que o tráfego era distribuído alternadamente entre o servidor `server1` (vs584) e o servidor `server2` (vs373), confirmando o funcionamento correto do balanceamento de carga.

Testes de Failover

Para testar a capacidade de failover, desativámos temporariamente o servidor `server1` (vs584). Durante o teste:

1. Monitorizámos a interface de estatísticas para observar o comportamento do HAProxy.
2. Confirmámos que todo o tráfego era automaticamente redirecionado para o servidor `server2` (vs373), sem interrupções no serviço.

Conclusão

A implementação do clustering, com a integração do HAProxy, proporcionou uma solução robusta e eficiente para a aplicação. A arquitetura final inclui:

1. Um servidor gestor de cluster (vs442), responsável pelo balanceamento de carga e pela gestão de falhas.
2. Dois servidores principais (vs584 e vs373), garantindo escalabilidade e resiliência.

Com esta configuração, o sistema oferece:

- **Distribuição eficiente do tráfego** entre os servidores ativos.
- **Failover automático**, assegurando a continuidade do serviço em situações de falha.
- **Monitorização eficaz** através da interface de estatísticas do HAProxy.

Desta forma, a infraestrutura foi projetada para maximizar o desempenho, a resiliência e a disponibilidade da aplicação, proporcionando uma experiência consistente e fiável para os utilizadores, mesmo perante eventos imprevistos.

Us10 [Rodrigo Cardoso 1221083]

No âmbito desta *User Story*, propõe-se que seja configurado o acesso à máquina virtual pelo administrador possa conectar via SSH utilizando exclusivamente certificados de chave pública/privada, com a desativação de uso de palavras-passe para autenticação.

Criação de uma conta com permissões de administrador

Em vez de usar a *root* da máquina para não gerar efeitos indesejados, foi criado uma conta com **permissões de administrador** na máquina virtual, com o nome de “adminsyst” e com o seguinte comando:

```
adduser adminsyst
```

Foi implementado uma password teste, sendo que será descartada mais adiante. Foram configuradas as permissões adequadas para a nova conta criada:

```
usermod -aG sudo adminsyst
```

Este comando adiciona o utilizador “adminsyst” ao grupo “sudo”, permitindo que ele execute **comandos administrativos** com o uso do **sudo**. A opção **-aG** assegura que o utilizador seja incluído no grupo adicionalmente aos grupos aos quais já pertence.

Com essa conta criada procede-se à configuração do acesso dessa conta pelo certificado de chave pública/privada.

Geração e Configuração de Chaves Públicas e Privadas no SSH

Será gerado um novo **par de chaves pública/privada** no SSH com o seguinte comando:

```
ssh-keygen -t rsa -b 4096
```

Com esse comando, será criado um par de chaves pública e privada (para descriptar a chave pública e obter acesso ao recurso). Esse par de chaves será do tipo **RSA** (Rivest-Shamir-Adleman) e com um tamanho de **4096 bits** (512 bytes) para uma segurança mais reforçada no acesso via SSH.

Esse foi o output após a criação do certificado:

```
The key fingerprint is:
SHA256:neDhf9huRyDTT/6X6UmLQe6JT4iSF73DoDGfmUnuXGI
The key's randomart image is:
+---[RSA 4096]-----+
|
|      o  .
|    o ++.o .
|   oS+o+ *
|  O.Xo= +
| + Eo*o= oo|
| * oo* *o=|
| o oo*.+.|
+---[SHA256]-----+
```

Figura 10 - Output de criação da chave de acesso SSH

Agora será transferido esse mesmo par de chaves para o acesso SSH da conta recém-criada:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub adminsys@10.9.10.55
```

O certificado de chave pública foi extraído do ficheiro **id_rsa.pub** e armazenado no ficheiro **authorized_keys** dentro do ambiente da outra conta. Foi necessário a autenticação pela palavra-passe temporária para validar a entrada dos pares de chave pública/privada no sistema.

Agora será verificado se o certificado foi realmente inserido dentro da conta de administrador. Com isso entrou-se no ambiente de trabalho da conta:

```
su - adminsys
```

Dentro do ambiente de trabalho, foi verificado a existência do certificado com o seguinte comando:

```
cat ~/.ssh/authorized_keys
```

Ao executar esse comando, deverá aparecer conteúdo do tipo “*ssh-rsa <chave>*”. O output gerado está apresentado na seguinte imagem:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQCq+bfbDbObYv4a69NmGmGeIn8jpVRSWUY9Jg2rvlQgjdjEFGvixXM51zaFM4ZThA7aBD4G7H+VuiczVF1L
B0R0YWo7GhmdRcCyDvDi4a3bydZoPcbHeBF+I4Sb2m1bANopgHE6MgjAlO8dKvrv60Y0jIp17DnYxu4Pgckjtz2vuE/2hhpyo6jLP6kmzKIMmtmNMVT4xffl
dteleKxp2FYIm5u03xVh2E6vwXL39Z3cdDw6P2q8KwuRKUZ00MhqPNr256TaaOmaETk0rVM6gW1nN0XeSrv5rg4AYekiYw932MzF7NtAU99/v6f8Bn0Mr1Ky
S6WJE8jfyM+f5UAd1f09tcm4KOCYFbAhiK6ocRQwOg1YpvuLKWMEvsK50aSyOSQnQqJ89k0T3rhCwyv0kjKfDlw6qQdNe8H7a5gy1QAzDAs+/2Inw3RM2w
tsL/xKrPsnYgkmlZF7qzxt3eTjbYNoGet/UcJlHkix8kxduuIb8FyGBqajUjyGqj1gccxWJC9khAdS4L9moF1djIWHFBJUv9bHLI+a2bZzERdl+TxkgLcQX
+bRH4ZruUxotS5zbGevS2QbddlN1I0jI4ZLxDePu4/iBjF0+wt1T8T6v004fi8Vxc5s2caLY0j79/EtjHsIRmni0PT6R4kMiY5SEZN2Qe+0BBTCZqXV+TKnL
RQ== root@uvm055
```

Figura 11 - Output da chave pública SSH da conta "adminsys"

Nesse caso, foi validado com sucesso a **presença da chave pública** dentro do ambiente do “adminsys”.

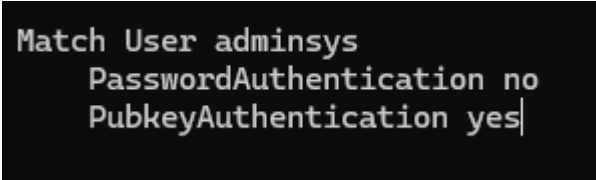
Configuração do acesso SSH

Com o certificado criado, iremos proceder à configuração do acesso SSH da conta do administrador. Foi efetuada a autenticação pela conta do **administrador** “adminsyst”. Logo após a autorização, foi efetuado o seguinte comando para configurar o acesso SSH:

```
sudo nano /etc/ssh/sshd_config
```

Foi efetuado a manipulação do ficheiro no caso de ativação e desativação de acessos SSH.

Foi adicionado o seguinte conteúdo ao ficheiro:

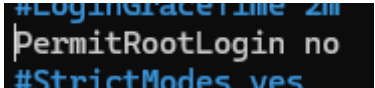


```
Match User adminsyst
    PasswordAuthentication no
    PubkeyAuthentication yes
```

Figura 12 - Conteúdo adicionado no ficheiro sshd_config

O conteúdo presente apresenta as **configurações de acesso** por SSH apenas no utilizador selecionado. Foi removida o acesso à conta por **palavra-passe** a adicionada a autenticação por **chave pública**.

Também foi proibido o **acesso direto** das contas pela **root**, para efeitos de validação do acesso pela chave pública:



```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
```

Figura 13 - Proibição do acesso
direto de contas pela root

Com essas configurações adicionadas, efetuou-se o **reinício** do SSH, para guardar as alterações feitas no ficheiro de configuração com o comando:

```
sudo systemctl restart sshd
```

Teste de acesso SSH pela conta de administrador

Foi efetuado a tentativa de acesso via SSH pelo administrador através da *root* (onde está presente o conteúdo da chave pública) e esse foi o output mostrado:

```
root@uvm055:~# ssh -i ~/.ssh/id_rsa adminsys@10.9.10.55
Bem-vindo,root ao sistema Linux do grupo 55
Data e Hora: Sunday, 22 de December de 2024, 20:10:55
Desejamos-lhe uma boa estadia!

adminsys@uvm055:~$ |
```

Esse output valida que não foi necessário o uso da palavra-passe para efetuar a autenticação à conta, sendo assim usado o certificado de chave pública presente no conteúdo do ficheiro **id_rsa**.

A implementação de acesso SSH utilizando exclusivamente certificados garante um **método confiável** e eficiente de autenticação. Esta configuração elimina a dependência de palavras-passe, reduzindo a probabilidade de acessos não autorizados.

Além disso, a utilização de boas práticas na configuração do servidor SSH assegura um ambiente devidamente preparado para **administração remota**. Este processo promove uma **gestão remota funcional** e alinhada com as normas modernas de acesso a sistemas.

Us11 [Rodrigo Cardoso 1221083]

No âmbito desta *User Story*, propõe-se a criação de uma pasta pública de ficheiros, com o objetivo de agilizar a partilha de ficheiros entre diferentes equipas. Essa mesma pasta terá o formato SMB/CIFS ou NFS.

No nosso caso, irá ser usado o formato **SMB/CIFS** da pasta pública, com o uso da máquina virtual **Windows**.

Instalação da ferramenta File Server

Para partilhar os ficheiros de forma pública, será necessário instalar o File Server, que corresponde a uma *feature* da seguinte forma:

1. Abrir o **Server Manager** e depois clicar em “**Add Roles or Features**”
2. Selecionar **File and Storage Services** e depois **File and iSCSI Services**
3. Ativar a *feature* **File Server**

No caso do nosso grupo, o File Server já está instalado e não será necessário o processo de instalação e ativação.

Criação e ativação da partilha de ficheiros

Com a ferramenta presente na máquina virtual, será efetuado a partilha de ficheiros para o acesso público das equipas.

No **Server Manager**, navegou-se para **File and Storage Services** e após isso foi acedido ao painel **Shares**. Esse é o menu apresentado pelo painel selecionado:

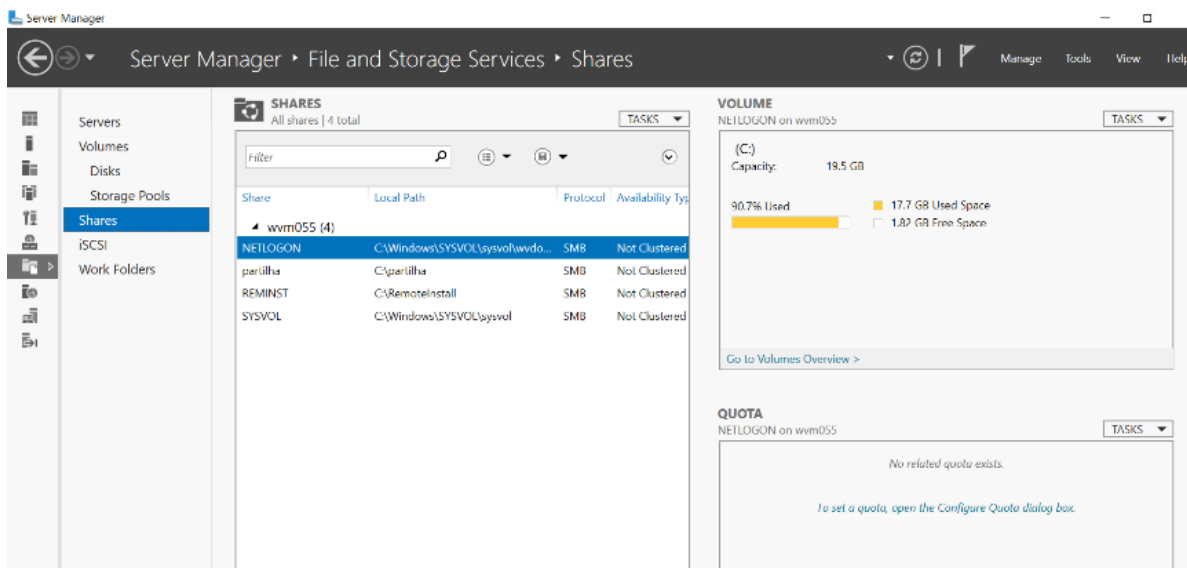


Figura 15 - Painel Shares em File and Storage Services

Agora seleciona-se o nome da máquina (**wvm055**) com o botão direito do rato e clica-se em **New Share**.

Como perfil de partilha foi selecionada a opção **SMB Share – Quick** pois é a mais simples e apropriada para as partilhas públicas de ficheiros.

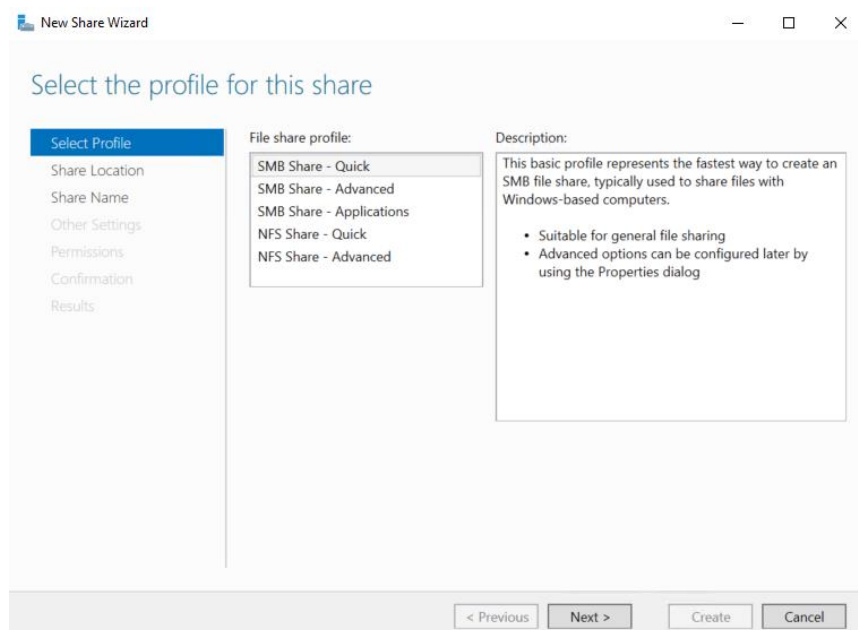


Figura 16 - Opção SMB Share - Quick da partilha de ficheiros

Os ficheiros que serão partilhados por todas as equipas estarão numa pasta partilhada chamada **“PublicShare”**.

Foi adicionada como uma pasta customizada dentro do disco C: da máquina virtual.

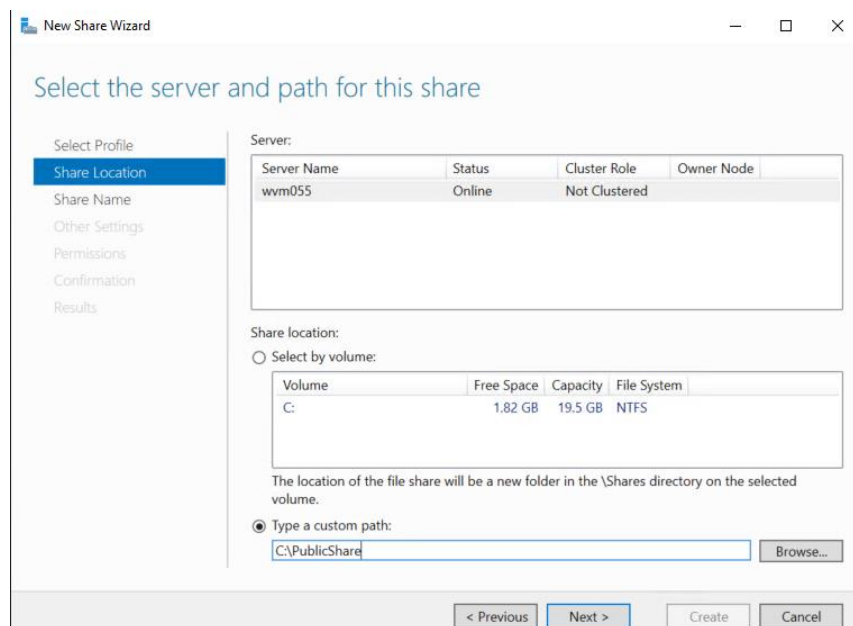


Figura 17 - Pasta Partilhada PublicShare

Após esse momento, e quando clicar em **Next**, será criado a pasta partilhada dentro da máquina virtual.

Foi avançado até ao menu das permissões. No caso do nosso grupo, as permissões foram definidas da seguinte forma:

- Os utilizadores (todos os membros das equipas) terão apenas permissões de **leitura e execução**, pois em um contexto de uma aplicação de agendamento de cirurgias, num caso de manipulação de ficheiros sem a autorização de um superior, poderá ocorrer o risco de corromper a base de dados ligada aos ficheiros partilhados e perda dos próprios dados.

- Os chefes e secretários das equipas (no nosso caso o grupo wgrupo1) irão obter permissões de **escrita, leitura e execução**, porque correspondem aos superiores dos membros da equipa, e assim poderão potencializar a eficiência comunicativa entre as várias equipas.

- O administrador irá ter **controlo total** da pasta partilhada, pois corresponde à entidade máxima responsável pelo sistema.

- Todos os utilizadores, incluindo os chefes, secretário e o administrador terão **controlo total** sobre a partilha dos ficheiros públicos, mas sempre com uma atenção redobrada em partilhas a utilizadores não pertencentes ao sistema.

Assim fica o menu das permissões com todos os pontos definidos no sistema:

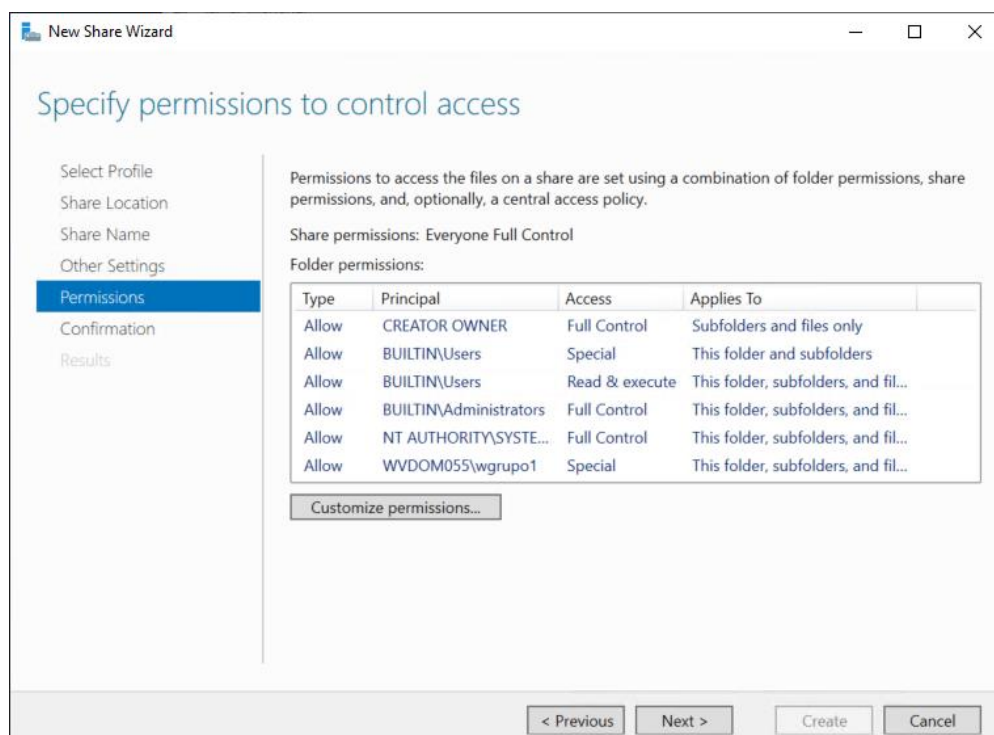


Figura 18 - Menu das permissões da pasta partilhada

Para atribuir as permissões adequadas, foi clicado em **Customize permissions**. Caso é preciso modificar as permissões do diretório clica-se num grupo que está presente na tabela e altera-se as permissões necessárias. Para adicionar uma *entry* (permissões a um grupo diferente) basta seleccionar a opção **Add** e assim seleccionar as permissões adequadas.

Agora será apresentada a confirmação da criação e atribuição das permissões dos ficheiros públicos presentes na tal pasta criada.

Foi apresentada a confirmação da criação da pasta partilhada:

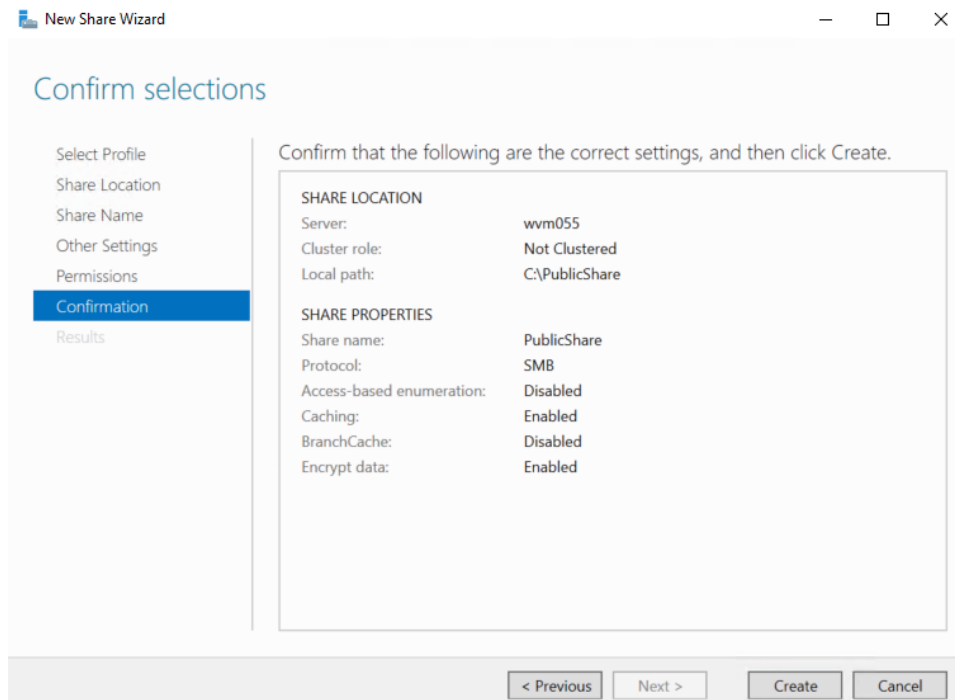


Figura 19 - Confirmação da criação da pasta partilhada

Neste painel de confirmação aparece os seguintes atributos e os seus valores:

- O nome do servidor, que é wvm055.
- Não está atribuído cargos de clusters, sendo que o nosso grupo suponha que o sistema funciona de forma interna e que não seja necessária a presença de sistemas externos.
- A pasta local de partilha que será a **pasta criada** (PublicShare).
- O protocolo usado: **Samba (SMB)**.
- Não é suposto ter acesso baseado em enumeração ligado, pois todos utilizadores no sistema irão ter **acesso** aos ficheiros presentes no diretório.
- Foi ativada o modo de **Cache de partilha** para os utilizadores conseguirem partilhar os conteúdos em modo offline (fora da rede local) mas não será necessário o download dessa mesma cache (**BranchCache**).
- Os dados serão encriptados de modo a aumentar a segurança do sistema e assim evitar o risco de vulnerabilidade e exposição de dados pessoais.

Foi clicado em **Create** e assim a pasta está pronta para partilhar ficheiros públicos com o uso do protocolo **SMB/CIFS**.

Testes de acesso à pasta partilhada

Foi efetuada a tentativa de acesso à pasta partilhada. O formato de acesso à pasta partilha é do seguinte formato:

\\nome do servidor\nome da partilha

Na máquina virtual (local no requisito dos ficheiros partilhados) foi executado o seguinte comando no menu Run (**Win + R** no teclado):

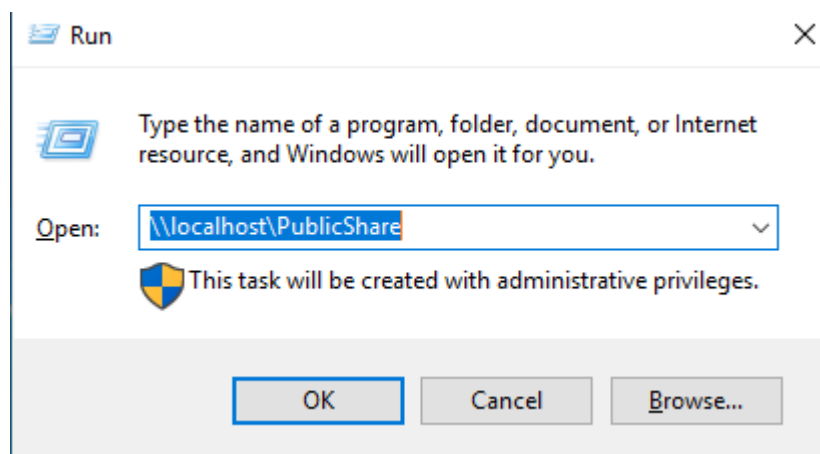


Figura 20 - Teste de execução na máquina interna

Foi verificado um acesso validado ao **recurso**, o que confirma o acesso de todos os utilizadores a partir da máquina local.

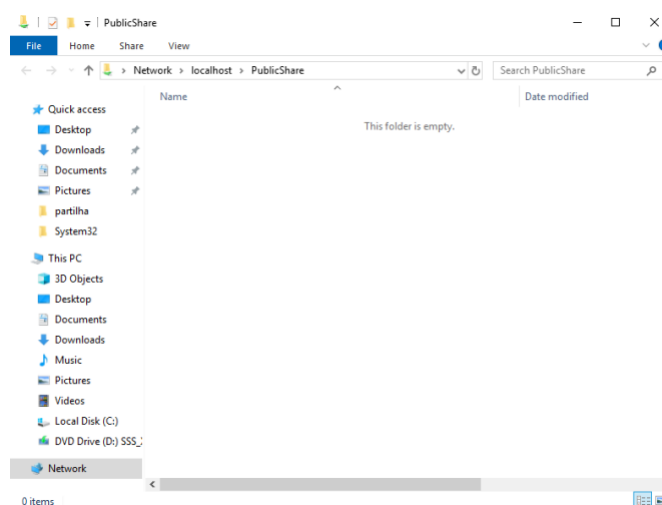


Figura 21 - Acesso á pasta partilhada na máquina interna

Agora será testado o acesso à pasta partilhada numa máquina presente na mesma rede que a máquina virtual.

Foi efetuado o mesmo comando, mas com o nome do servidor em vez de *localhost* com o menu **Run**:

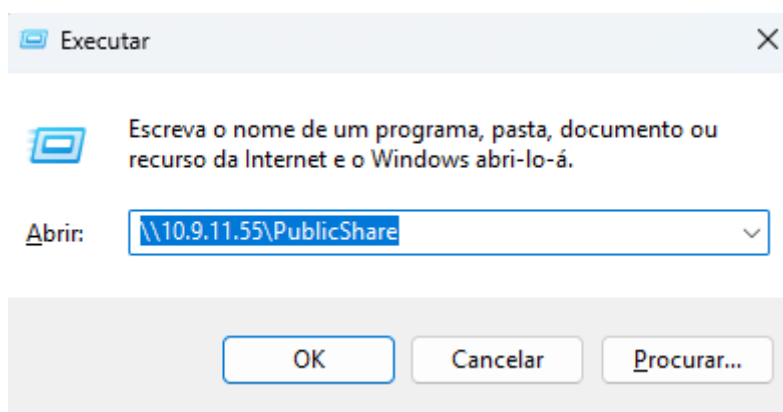


Figura 22 - Acesso à pasta partilhada a partir de uma máquina remota presente na mesma rede

Foram pedidas as **credenciais** no acesso ao recurso (no nosso grupo foi efetuado a autenticação como **administrador**).

Após a autorização, foi validado com **sucesso** o acesso à pasta partilhada. Logo, as máquinas presentes na mesma rede conseguem aceder aos ficheiros presentes na pasta.

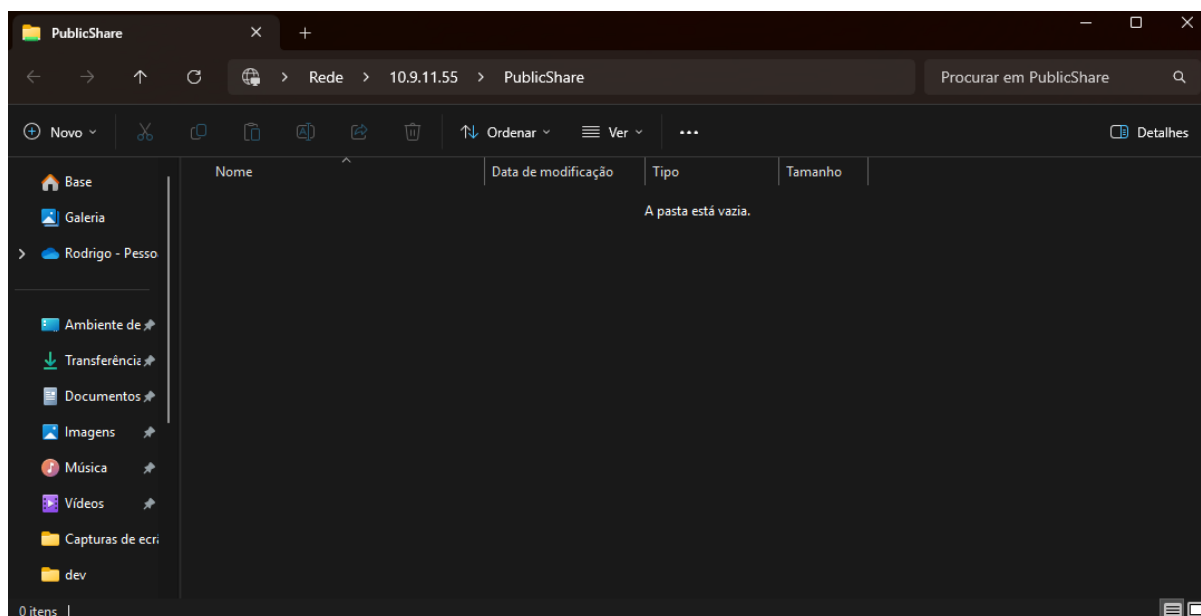


Figura 23 - Confirmação do acesso à pasta partilhada a partir de uma máquina remota

A criação da partilha pública no formato SMB demonstrou ser uma solução **eficiente** para atender à **necessidade de colaboração entre as equipas**. Com a partilha devidamente configurada e acessível, foi possível estabelecer um ambiente que **simplifica** o fluxo de trabalho e promove a troca de informações de forma **ágil**.

Por fim, a implementação de boas práticas de segurança e monitorização será essencial para garantir que a partilha continue a atender às necessidades organizacionais de forma sustentável e segura.

Us12 [Rodrigo Castro 1220636]

```
GNU nano 5.4 backup_db.sh
#!/bin/bash
#!/bin/bash

# Path to the backup directory and MongoDB URI
MONGO_URI="mongodb+srv://rodrigocastro2004:projetolaprsem5@sem5g55isep.gtdw4.mongodb.net/?retryWrites=true&w=majority&appName=sem5g55isep"
BACKUP_DIR="/etc/daily_database_backup/sem5_backup_20241213/test"

# Restore the backup
echo "🔧 Initializing restore of backup..."
mongorestore --drop --uri="$MONGO_URI" --dir="$BACKUP_DIR"

# Check if the restoration was successful
if [ $? -eq 0 ]; then
    echo "✅ Backup restored successfully."
else
    echo "❌ Error restoring the backup."
    exit 1
fi

# Database name (adjust as needed)
DB_NAME="sem5g55isep"

# Collections to validate
COLLECTIONS=("allergycatalogs" "medicalconditions" "medicalrecords" "roles" "users")

# Validating the existence of collections in the database
echo "🔍 Validating the existence of collections in the database..."

for COLLECTION in "${COLLECTIONS[@]}; do
    # Checking the collection using Mongo shell
    mongosh "$MONGO_URI/$DB_NAME" --eval "
        if (db.getCollectionNames().indexOf('$COLLECTION') != -1) {
            print('✅ The collection $COLLECTION was restored successfully.');"
        } else {
            print('❌ The collection $COLLECTION was not restored or does not exist in the database.');"
        }
    "
done
```

Figura 24 backup_db.sh

Explicação script:

O script realiza a restauração de um backup de base de dados MongoDB. Inicialmente, define o caminho do diretório de backup e a URI de ligação ao MongoDB. Usando o comando `mongorestore`, restaura a base de dados e substitui as coleções existentes com o parâmetro `--drop`. Após a restauração, o script verifica se a operação foi bem-sucedida e exibe uma mensagem de sucesso ou erro. Em seguida, valida a existência de coleções específicas na base de dados restaurada, utilizando `mongosh`, e imprime mensagens a indicar se cada coleção foi restaurada com sucesso ou não.

```
root@uvm855:~# ./backup_db.sh /etc/daily_database_backup/sens_backup_20241213/test
[+] Initializing restore of backup...
2024-12-13T12:26:28.632+0000 preparing collections to restore from
2024-12-13T12:26:28.633+0000 reading metadata for roles.roles from /etc/daily_database_backup/sens_backup_20241213/test/roles/roles.metadata.json
2024-12-13T12:26:28.634+0000 reading metadata for users.users from /etc/daily_database_backup/sens_backup_20241213/test/users/users.metadata.json
2024-12-13T12:26:28.637+0000 reading metadata for allergycatalogs.allergycatalogs from /etc/daily_database_backup/sens_backup_20241213/test/allergycatalogs/allergycatalogs.metadata.json
2024-12-13T12:26:28.638+0000 reading metadata for medicalconditions.medicalconditions from /etc/daily_database_backup/sens_backup_20241213/test/medicalconditions/medicalconditions.metadata.json
2024-12-13T12:26:28.639+0000 reading metadata for medicalrecords.medicalrecords from /etc/daily_database_backup/sens_backup_20241213/test/medicalrecords/medicalrecords.metadata.json
2024-12-13T12:26:28.677+0000 dropping collection medicalrecords.medicalrecords before restoring
2024-12-13T12:26:28.740+0000 dropping collection users.users before restoring
2024-12-13T12:26:28.855+0000 dropping collection allergycatalogs.allergycatalogs before restoring
2024-12-13T12:26:28.940+0000 dropping collection medicalconditions.medicalconditions before restoring
2024-12-13T12:26:28.959+0000 restoring medicalconditions.medicalconditions from /etc/daily_database_backup/sens_backup_20241213/test/medicalconditions/medicalconditions.bson
2024-12-13T12:26:28.980+0000 finished restoring medicalconditions.medicalconditions (8 documents, 0 failures)
2024-12-13T12:26:28.983+0000 finished restoring medicalconditions.medicalconditions (18 documents, 0 failures)
2024-12-13T12:26:28.986+0000 dropping collection roles.roles before restoring
2024-12-13T12:26:28.987+0000 restoring users.users from /etc/daily_database_backup/sens_backup_20241213/test/users/users.bson
2024-12-13T12:26:28.975+0000 restoring allergycatalogs.allergycatalogs from /etc/daily_database_backup/sens_backup_20241213/test/allergycatalogs/allergycatalogs.bson
2024-12-13T12:26:28.988+0000 finished restoring users.users (2 documents, 0 failures)
2024-12-13T12:26:28.989+0000 restoring roles.roles from /etc/daily_database_backup/sens_backup_20241213/test/roles/roles.bson
2024-12-13T12:26:28.991+0000 finished restoring roles.roles (4 documents, 0 failures)
2024-12-13T12:26:28.991+0000 restoring indexes for collection roles.roles from metadata
2024-12-13T12:26:28.992+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"domainId.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"domainId", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.993+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"name.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"name", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.994+0000 restoring indexes for collection users.users from metadata
2024-12-13T12:26:28.995+0000 restoring indexes for collection allergycatalogs.allergycatalogs from metadata
2024-12-13T12:26:28.996+0000 restoring indexes for collection medicalconditions.medicalconditions from metadata
2024-12-13T12:26:28.997+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"firstHame.1", "v":2}, Key:primitive.D[primitive.E{key:"firstHame", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.998+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"domainId.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"domainId", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.999+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"domainId.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"domainId", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.999+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"name.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"name", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.999+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"email.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"email", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.999+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"name.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"name", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.999+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"domainId.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"domainId", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.999+0000 restoring indexes for collection medicalrecords.medicalrecords from metadata
2024-12-13T12:26:28.999+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"patientId.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"patientId", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.999+0000 index: &ids.IndexDocument[Options:primitive.M{"background":true, "name":"id.1", "unique":true, "v":2}, Key:primitive.D[primitive.E{key:"id", Value:1}], PartialFilterExpression:primitive.D(nil)]
2024-12-13T12:26:28.999+0000 31 document(s) restored successfully. 0 document(s) failed to restore.
[+] Backup restored successfully.
[+] Validating the existence of collections in the database...
[+] The collection allergycatalogs was restored successfully.
[+] The collection medicalconditions was restored successfully.
[+] The collection medicalrecords was restored successfully.
[+] The collection roles was restored successfully.
[+] The collection users was restored successfully.
root@uvm855:~#
```

Figura 25 Exemplo de execução do backup_db.sh