**ASIST**

**Administração de Sistemas**

Sprint 2 – G033

A picture containing shape

Description automatically generated

1210816 - João Castro

1210913 - Pedro Mendes

1211523 - Martim Botelho

1191831 - Rui Gonçalves

ÍNDICE

[Alínea 1 3](#_Toc149511452)

[Alínea 2 4](#_Toc149511453)

[Alínea 3 6](#_Toc149511454)

[Alínea 4 8](#_Toc149511455)

[Alínea 5 13](#_Toc149511456)

[Alínea 6 16](#_Toc149511457)

[Alínea 7 18](#_Toc149511458)

[Alínea 8 20](#_Toc149511459)

# **Divisão - User Stories**

|  |  |
| --- | --- |
| **Alínea 1** | Ficou responsável pelo Pedro Mendes - 1210913 |
| **Alínea 2** | Ficou responsável pelo Martim Botelho - 1211523 |
| **Alínea 3** | Ficou responsável pelo Martim Botelho - 1211523 |
| **Alínea 4** | Ficou responsável pelo João Castro - 1210816 |
| **Alínea 5** | Ficou responsável pelo Rui Gonçalo - 1191831 |
| **Alínea 6** | Ficou responsável pelo Rui Gonçalo - 1191831 |
| **Alínea 7** | Ficou responsável pelo Pedro Mendes - 1210913 |
| **Alínea 8** | Ficou responsável pelo João Castro - 1210816 |

# **Alínea 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

Para começar, criei um servidor utilizando o Debian 11 –base system, configurando-o usando o nginx pois este pode receber varias conexões ao mesmo tempo, ou seja, comecei por usar estes dois comandos:

*Sudo apt install nodejs npm*

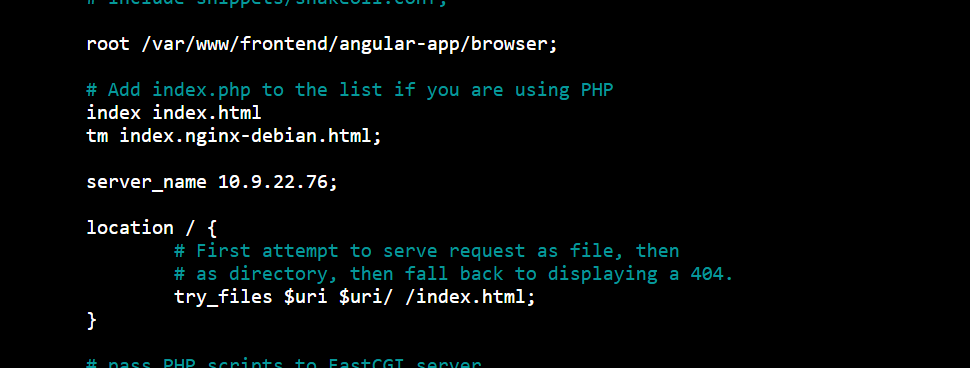
*Sudo apt install ginx*

De seguida, criei no modulo angular uma pasta com nome “dist” para armazenar o projeto angular que será enviado para o servidor.

Depois de o modulo estar criado, acedi a ficheiro de configuraçao usando o seguinte comando

*Sudo nano /etc/nginx/sites-available/default*

Dentro do ficheiro de configuração, caracterizei na root a localização do ficheiro index.html, como podemos observar nesta imagem:

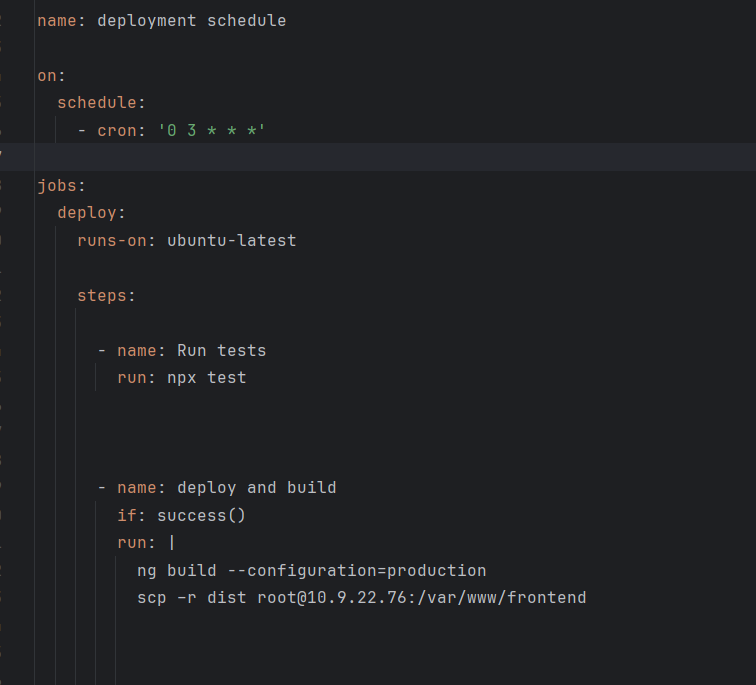


De seguida, apos ter configurado o meu servidor, com o comando *Scp –r dist* [*root@10.9.22.76:/var/www/*frontend](mailto:root@10.9.22.76:/var/www/frontend) é feito o deployment de um modulo, no meu caso foi o front-end do meu programa, na pasta /var/www/frontend

comecei por fazer o deployment de um dos módulos, neste caso, eu fiz o deployment do front-end do meu programa, na pasta /var/www/appAngular.

Por fim para tornar o deployment sistematico, validando de forma agendada com o plano de testes, criei um diretorio github/workflows no projeto, criando um ficheiro de configuraçao chamado deployment.yml.

Neste ficheiro defini um schedule para que o deployment seja agendado e seja executado automaticamente uma vez por dia, neste caso ás 3 da manha.



# **Alínea 2**

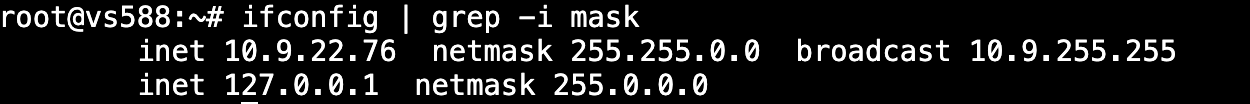
Como administrador do sistema quero que apenas os clientes da rede interna do DEI (cablada ou via VPN) possam aceder à solução.

# Primeiramente, temos de descobrir qual é o range dos IP’s que queremos permitir que acedam à solução. Para isso foi utilizado o comando ifconfig | grep -i mask

(Antes de instalar as net-tools com o comando sudo apt install net-tools e a iptables com o comando sudo apt install iptables)







Assim podemos concluir que o nosso range de IP’s será **10.9.0.1 – 10.9.255.254.**

Agora temos de utilizar os comandos iptables para alterar o comportamento da solução ao receber packages.

A screenshot of a computer

Description automatically generated

Utilizamos o -F para dar flush a todas as regras de todas as chains, -X para eliminar todas as chains definidas por users e o -P INPUT DROP para definir a política padrão da chain INPUT para drop.

Também adicionamos o uma regra para que todo o tráfego recebido pela porta 4000 (porta da aplicação Backend) seja bloqueado.

Por fim, definimos a regra que vai permitir que os IP’s do range definido anteriormente sejam tratados com a política ACCEPT.

Assim, apenas os clientes da rede interna do DEI (cablada ou via VPN) podem aceder à solução.

# **Alínea 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.

Para realizar esta user story tivemos de criar o ficheiro de texto que contém o CIDR que encontramos na última user story (10.9.0.0/16)

A black and white rectangle with a white line

Description automatically generated

Depois foi criado um script Bash para adicionar regras à iptables consoante os conteúdos do ficheiro **allowed\_ips.txt**.

A screen shot of a computer screen

Description automatically generated

Neste script começamos por definir o ficheiro onde se encontram os IP’s/CIDR’s que pretendemos adicionar à iptables e a interface de rede às quais estas se vão aplicar.

Para cada linha removemos quaisquer espaços em branco que possam estar no inicio ou no fim dessa linha. Depois verificamos se essa linha contém o carater **“/”** que nos indica se estamos na presença de um CIDR. Para cada opção adicionamos uma regra à iptables na chain INPUT, para a porta definida no início do script contendo o IP/CIDR lido na respetiva linha do ficheiro com a política definida em ACCEPT. Assim permitimos todo o tráfico proveniente desse IP/CIDR.

Por fim, damos echo a uma mensagem de sucesso da operação.

Depois da criação do script, foram-lhe associadas permissões de execução.



Por último, o script foi executado utilizando o comando ./update\_ips.sh.



Foi obtida a respetiva mensagem de sucesso.



Podemos confirmar a adição da regra à iptables com a utilização do comando iptables com a flag -L.

A computer screen with white text

Description automatically generated

# **Alínea 4**

Como administrador quero identificar e quantificar os riscos envolvidos na solução preconizada

Para compreender os riscos existentes na solução preconizada,decidi contruir uma matriz de risco para comparar tanto a probabilidade como o impacto identificando os riscos que são necessarios a nossa atençao, e os riscos cuja a probabilidade é mais reduzida.

|  |  |  |  |
| --- | --- | --- | --- |
| Probabilidade/Impacto | Nível 1 | Nível 2 | Nível 3 |
| Nível 1  (0%-33%) |  |  | Servidores do DEI serem alvos de ataques informáticos/irem abaixo |
| Nível 2  (34%-60%) |  | Erros no Planeamento da soluçao |  |
| Nível 3  (61%-100%) | Pequenos bugs na soluçao |  |  |

Conclusões:

Tendo em conta esta matriz de riscos podemos afirmar que estar dependente só serviços das maquinas do DEI não é uma boa opção pois estas podem sofrer ataques informáticos, ou simplesmente podem ir abaixo. Para resolver este problema era necessário investir numa cloud com uma melhor qualidade, apesar de hoje em dia, seja qual for a nossa cloud esta em risco de receber ataques informáticos.

# **Alínea 5**

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

**Proposta de MBCO - Sistema RobDroneGo**

Objetivo: Estabelecer um nível mínimo de operacionalidade para o sistema RobDroneGo, mantendo a gestão e execução das tarefas críticas durante e após uma interrupção.

Estratégias:

1. **Redundância e Alta Disponibilidade:** Implementar redundância geográfica e soluções de alta disponibilidade para componentes essenciais como base de dados e UI na cloud.
2. **Contingência e Recuperação:** Desenvolver planos de contingência abrangentes e procedimentos de recuperação detalhados para cenários de interrupção variados.
3. **Comunicação e Capacitação:** Estabelecer programas de comunicação e treino para assegurar a compreensão e preparação das equipas de resposta.
4. **Revisão e Melhoria:** Instituir revisões periódicas e um processo de melhoria contínuo para o MBCO, alinhando-o com a evolução das necessidades do negócio.

Implementação:

* Assegurar que o sistema de gestão de tarefas permaneça funcional com backups frequentes e *failovers* automáticos.
* Manter a comunicação contínua entre drones e a central de controlo, utilizando serviços de mensagens na nuvem.
* Garantir que as interfaces do utilizador estejam acessíveis, com o *frontend* hospedado em infraestruturas de cloud confiáveis.
* Testar e validar periodicamente os planos de recuperação para garantir uma resposta eficaz a interrupções.

Assim, com o MBCO proposto, o RobDroneGo estará preparado para enfrentar desafios operacionais, mantendo a satisfação do cliente e a integridade do serviço em situações de emergência. Este plano será apresentado aos *stakeholders* para aprovação e implementação subsequente.

# **Alínea 6**

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

**Estratégia de Salvaguarda de Informação**

A salvaguarda de informação, ou backup, é a duplicação de dados essenciais em locais alternativos para permitir a recuperação em caso de falhas. Esta recuperação pode exigir um período considerável, impactando as operações comerciais. Assim, é crucial avaliar três critérios para assegurar a eficácia de um plano de backup:

* **Objetivo de Ponto de Recuperação (OPR):** o intervalo de tempo máximo tolerável para a perda de dados;
* **Objetivo de Tempo de Recuperação (OTR)**: o período esperado para restaurar sistemas e infraestruturas;
* **Tempo de Recuperação de Trabalho (TRT):** o tempo necessário para restaurar e validar os dados.

Dentro do contexto do RobDroneGo, um sistema que integra uma frota de robôs e drones para execução de tarefas diversas, a consistência e a prontidão dos dados são cruciais. A estratégia de backup deve ser robusta para garantir que as informações sobre as tarefas atribuídas, o estado dos dispositivos, e o planeamento das rotas estejam sempre disponíveis e atualizadas.

A cópia será feita 1 vez por dia para que o RPO seja reduzido e de forma a não interferir com o negócio. Assim, estão programadas cópias para a hora de almoço (12h) onde se acredita que existirá uma menor alteração dos dados.

Para minimizar o POR, sugere-se que o backup seja efetuado com uma frequência horária, dada a natureza crítica e a rápida alteração dos dados neste sistema. Isto garante que, no máximo, apenas uma hora de dados de trabalho será perdida em caso de falha. O uso de snapshots incrementais pode ser uma forma eficiente de implementar esta estratégia sem sobrecarregar o sistema.

Quanto ao WRT, a estratégia deve focar na rapidez da recuperação. Uma abordagem seria utilizar um sistema de backup em tempo real ou quase em tempo real, como o uso de réplicas em hot standby. Com réplicas sincronizadas e prontas para entrar em ação, o sistema pode alternar para uma réplica atualizada quase instantaneamente em caso de falha, garantindo que o WRT seja mantido ao mínimo.

Para o sistema RobDroneGo, os backups devem ser automatizados através de scripts que possam ser executados em ambientes Windows ou Linux, dependendo da infraestrutura existente. Os scripts devem ser capazes de interagir com as bases de dados do sistema, que poderiam incluir SQL para dados estruturados e NoSQL para dados não estruturados e relacionados ao planeamento das rotas e estados dos dispositivos.

**Implementação**

Para efeitos de implementação, os seguintes passos foram realizados numa máquina com Windows Server 2019. Foi desenvolvido um script (MongoDB) que irá fazer a ligação à respetiva base de dados e vão fazer uma cópia dos ficheiros para uma localização definida na máquina.

O ficheiro tem o seguinte conteúdo:

A screenshot of a computer

Description automatically generated

Fazendo uma breve explicação do que acontece: a 1ª linha @echo off faz com que o Command Prompt não apareça quando o script é iniciado. As seguintes linhas, da 3 até à 10 vai usar o wmic (Windows Management Instrumentation Command-Line) para obter a data e, de seguida, é feita a separação em dia, mês e ano e o mesmo se faz com as horas usando o timestamp.

Depois de definido o nome do ficheiro através do prefixo backup\_mongodb, este é completado com a data e a hora do backup de forma a ser rápido de identificar para o caso de necessidade de reposição.

Para que as cópias de segurança possam ser feitas automaticamente usou-se o Programador de Tarefas do Windows e criou-se quatro tarefas programadas. Seguem-se os screenshots:

A screenshot of a computer

Description automatically generatedClicou-se em “Task Scheduler Library” e de seguida em “Create task...”.

De seguida, clicamos na aba Trigger e criamos um novo onde definimos o horário e quando a tarefa será executada (como dito mais acima, o backup será realizado às 12h):

A screenshot of a computer

Description automatically generated

Agora, em “Ações”, definimos o que irá correr nestes horários. Clicamos em Procurar e escolhemos o ficheiro .bat.

A screenshot of a computer program

Description automatically generated

No fim, o resultado é o seguinte:

A screenshot of a computer

Description automatically generated

# **Alínea 7**

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

Para realizar esta US, no inicio é necessário criar um diretório chamado “shared” usando o comando:

*Sudo mkdir shared*

Depois de criar este diretório é necessário criar um grupo de utilizadores:

*Sudo groupadd sharedGroup*

*De seguida, o objetivo é associar o diretório “shared” a este grupo de utilizadores que acabamos de criar. Neste comando é usado o “-R” pois este é recursivo, para que as alterações de grupo serem aplicadas a todos os ficheiros dentro do diretório “shared”*

*Sudo chgrp -R sharedGroup shared*

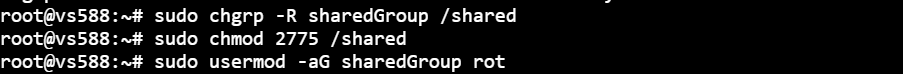
*Após a atribuição do diretório a este grupo de utilizadores, é necessário tratar das permissões do ficheiro “shared” usando este comando:*

*Sudo chmod -R 2775 shared*

*Neste comando, este presente novamente o -R pois as alterações, como acima descrito, devem ser aplicadas de forma recursiva, enquanto o “2775”, o numero 2 define permissões SGID significando que os ficheiros que forem criados dentro deste diretório em especifico terão como herdeiro o diretório pai, enquanto o primeiro 7 refere- se as permissões de leitura, escrita e execução do proprietário do ficheiro. O segundo 7 relaciona se com as permissões do grupo. Por fim o numero 5 atribui as permissões de leitura e execução para os restantes utilizadores.*

*Para terminar, basta adicionar os utilizadores do nosso sistema para o grupo sharedGroup para estes conseguirem ter acesso a esse diretório. Para isto usa-se o comando:*

*Sudo usermod -aG sharedGroup nomeDoUtilizador*



# **Alínea 8**

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

Para realizarmo esta US estámos a verificar no componente de login os utilizadores que tem acessos incorretos. *A screenshot of a computer program

Description automatically generated*

Se um utilizador não conseguir fazer login, o serviço manda um pedido post ao backend.

No backend depois temos uma route que recebe esse pedido e depois adiciona esse email ao ficheiro /var/auth.log. A computer screen shot of a program code

Description automatically generated

A screen shot of a computer

Description automatically generated

Aqui temos o ficheiro de texto com todos os logins onde houve falhar de autenticação.

O que fazemos agora é ter outra route que lê o seguinte ficheiro e retorna os Ids que estejam presentes no mesmo 3 ou mais vezes.

Aqui estámos a devolver todos os emais que apareçam no auth.log 3 ou mais vezes,

Por fim o UI vai fazer um pedido GET ao backend e irá receber os mesmos para mostrar ao administrador.

A screen shot of a computer program

Description automatically generated

*Aqui podemos ver o que se recebe na UI, como podemos ver estamos apenas a receber o email dos utilizadores que tiveram 3 ou mais falhas na autenticação.*

A screenshot of a computer

Description automatically generated