ASIST - Sprint 2

Grupo 33

**Realizado por:**

Pedro Sousa -1201428

Gonçalo Boa-Nova - 1190616

José Silva - 1190772

Diogo Carvalho - 1200611

Diogo Ribeiro – 1210792

**Índice**

**User story 1 3**

**User story 2 5**

**User story 3 6**

**User story 4 8**

**User story 5 10**

**User story 6 11**

**User story 1**

**US – 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 fazer o Deployment, em primeiro lugar optamos por criar uma Virtual Machine no DEI de Ubuntu. Criamos um user “lapr” com destino a fazer o Deployment do programa, para isso colocamos esse user presente no ficheiro *authorized\_files* para que o bitbucket tivesse acesso à Virtual Machine como forma de login.

Através da seguinte pipeline foi possível fazer o Deployment na dada VM:

Uma imagem com texto

Descrição gerada automaticamente

Análise do *script* da pipeline:

- cd dddnetcore 🡪 serve para entrar no módulo do projeto a que vamos dar *deploy*

- dotnet publish -o ./publish –configuration Release 🡪 com este comando compilamos o código e posteriormente esses ficheiros vão ser transferidos para um servidor “nginx”

- pipe 🡪 o comando pipe tem como variáveis as informações de login (nome do user: lapr (não necessita de password porque a autenticação é feita com SSH), nome do servidor a que vai aceder, o caminho dentro da VM onde vão estar os ficheiros do projeto (REMOTE\_PATH), indicação do destino dos ficheiros em que os ficheiros compilados vão estar guardados (LOCAL\_PATH) e a porta do servidor)

- apt-get updates && apt-get -qq install ssh openssh-client 🡪 instalação do ssh

- ssh -i ~/.ssh/config [lapr@csgate-ssh,dei.isep.ipp.pt](mailto:lapr@csgate-ssh,dei.isep.ipp.pt) -p 10812 sh /home/lapr/restart\_service 🡪comando para executar um script remotamente, script que reinicia o serviço de dotnet na VM

Uma imagem com texto

Descrição gerada automaticamente

Na imagem acima é apresentada a configuração do serviço de dotnet.

Working Directory – local onde estão os ficheiros da aplicação dotnet

ExecStart - executa a aplicação dotnet a partir do ficheiro .dll dentro da directory na linha acima especificada

Seguidamente apresenta-se a configuração do servidor “nginx”.

Uma imagem com texto

Descrição gerada automaticamente

O servidor ouve pedidos na porta 80 (*http*) e converte esses pedidos com proxy para a porta correta do server dotnet.

**User story 2**

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

Para permitir aceder á solução apenas clientes que estão na rede interna do DEI é necessário usar o comando “iptables” para aceder ao sistema de controlo de tráfego “Netfilter”.

Os comandos utilizados foram:

iptables -A INPUT -s (ip da rede do DEI) -p tcp -dport 80 --j ACCEPT

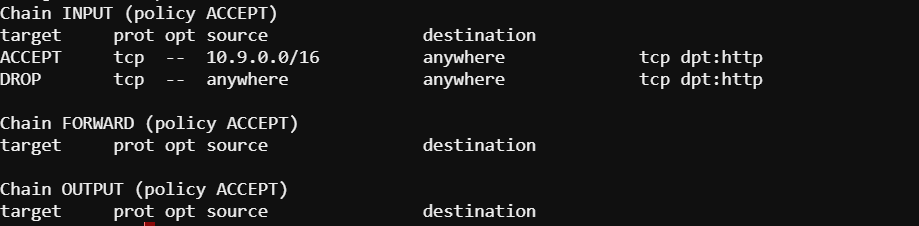
Este comando serve para permitir que os utilizadores da rede interna do DEI possam fazer um request http ao servidor em questão.

iptables -A INPUT -p tcp -dport 80 --j DROP

Este comando serve para negar acesso a todos os clientes que tentem aceder á solução.

(assumindo que a rede do DEI 10.9.0.0)

Resultado da iptable:



**User story 3**

**US – 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 definir os clientes através de um ficheiro é necessário um ficheiro de texto para definir os ips que podem aceder á solução, um batch file para ler os ips do ficheiro de texto e usar os respetivos ips com o comando “iptables”, este batch file deve ser executado de pouco em pouco tempo para que seja feita a atualização, para isso é feita uma alteração do ficheiro em /etc/crontab.

Ficheiro para definir ips: ips.txt

Código do batch file:

#!/bin/bash

filename=ips.txt #nome do ficheiro

iptables -F #dar reset ás iptables

while read line;do #enquanto o ficheiro tiver linhas

iptables -A INPUT -s $line -p tcp –dport 80 -j ACCEPT #usar a linha do ficheiro no comando

done<$filename

iptables -A INPUT -p tcp –dport 80 -j DROP #rejeitar qualquer outro ip

No ficheiro crontab adicionar a linha:

\*/5 \* \* \* \* root (path para o ficheiro batch)

Esta linha fará com que o ficheiro batch seja executado a cada 5 minutos, criando assim um sistema de atualização dos ips.

Ips.txt:



Batch file:

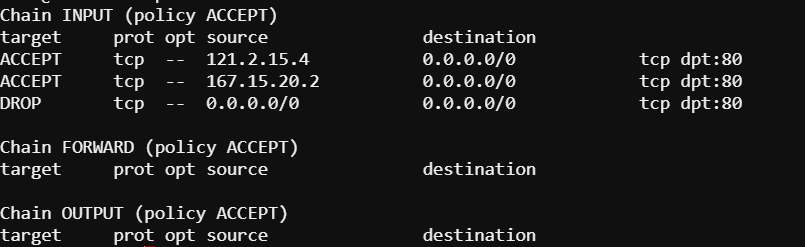
Uma imagem com texto

Descrição gerada automaticamente

/etc/crontab:



Iptables(resultado):



**User story 4**

**US – 4** – Como administrador quero identificar e quantificar os riscos envolvidos na solução preconizada.

**Matriz de Riscos**

|  |  |  |  |
| --- | --- | --- | --- |
| Probabilidade\Gravidade | **Tolerável** | **Moderado** | **Catastrófico** |
| **Improvável** | 1 | 4 | 7 |
| **Ocasional** | 2 | 5 | 8 |
| **Provável** | 3 | 6 | 9 |

A tabela seguinte representa riscos com a sua devida justificação, avaliados através da matriz de riscos:

|  |  |  |  |
| --- | --- | --- | --- |
| **Componente** | **Desastre** | **Nível de Risco** | **Justificação** |
| SPA | Serviço DEI inacessível | 8 | Devido a um grande congestionamento na utilização da Cloud do DEI, os serviços podem ficar lentos ou até inacessíveis. |
| SPA | Interrupção do serviço | 3 | Devido a manutenções semanais à infraestrutura da Cloud do DEI, os serviços ficam indisponíveis, sendo necessário reiniciá-los. |
| Servidor (.Net) | Serviço DEI inacessível | 8 | Devido a um grande congestionamento na utilização da Cloud do DEI, os serviços podem ficar lentos ou até inacessíveis. |
| Servidor (.Net) | Interrupção do serviço | 3 | Devido a manutenções semanais à infraestrutura da Cloud do DEI, os serviços ficam indisponíveis, sendo necessário reiniciá-los. |
| Base de Dados (SQL) | Serviço DEI inacessível | 8 | Devido a um grande congestionamento na utilização da Cloud do DEI, os serviços podem ficar lentos ou até inacessíveis. |
| Base de Dados (SQL) | Interrupção do serviço | 3 | Devido a manutenções semanais à infraestrutura da Cloud do DEI, os serviços ficam indisponíveis, sendo necessário reiniciá-los. |
| Servidor (Node.js) | Serviço DEI inacessível | 5 | Devido a um grande congestionamento na utilização da Cloud do DEI, os serviços podem ficar lentos ou até inacessíveis. |
| Servidor (Node.js) | Interrupção do serviço | 3 | Devido a manutenções semanais à infraestrutura da Cloud do DEI, os serviços ficam indisponíveis, sendo necessário reiniciá-los. |
| Base de Dados (Mongo) | Serviço DEI inacessível | 5 | Devido a um grande congestionamento na utilização da Cloud do DEI, os serviços podem ficar lentos ou até inacessíveis. |
| Base de Dados (Mongo) | Interrupção do serviço | 3 | Devido a manutenções semanais à infraestrutura da Cloud do DEI, os serviços ficam indisponíveis, sendo necessário reiniciá-los. |
| Planeamento | Serviço DEI inacessível | 5 | Devido a um grande congestionamento na utilização da Cloud do DEI, os serviços podem ficar lentos ou até inacessíveis. |
| Planeamento | Interrupção do serviço | 3 | Devido a manutenções semanais à infraestrutura da Cloud do DEI, os serviços ficam indisponíveis, sendo necessário reiniciá-los. |

**User story 5**

**US – 5** – Como administrador quero que seja definido o **Mininum Business Continuity Objective (MBCO)** a propor aos stakeholders.

|  |  |
| --- | --- |
| **Módulo de Negócio** | **MBCO** |
| Gestão de Armazéns | * Continuar a operar as cargas e descargas possíveis nos armazéns |
| Logística | * Período máximo de inoperacionalidade de 2 horas * Período máximo de disrupção de 5 horas |
| Planeamento de Distribuição | * Todas as entregas têm um período máximo de atraso de 1 dia * Os camiões finalizam as entregas que estiverem a efetuar |

**User story 6**

**US – 6** – Como administrador quero que seja proposta e implementada uma estratégia de segurança para minimizar o **Recovery Point Objective (RPO)** (tempo máximo de perda de dados aceite) e o **Work Recovery Time (WRT)** (tempo necessário para repor os dados e aplicações e testá-los).

**Estratégia de cópia de segurança**

**Backups**

Foi definido pelo Grupo 33 que os servidores de armazenamento de dados devem ter prioridade sobre os servidores API. Isto devido ao seu baixo RPO, requerendo backups recorrentes para a menor perda de dados possível e visto que os servidores API, que apresentam raras alterações, satisfazem-se com novos backups apenas em momentos de alteração de código. Assim, cada servidor deve ter um servidor de “backup” para Load Balancing, implementado na Cloud do DEI de modo a manter a consistência da infraestrutura. Para servidores de armazenamento de dados, deve existir adicionalmente uma Base de Dados e mais um servidor de backup para o anterior.

**Base de Dados** - Backup integral em dois dias da semana (Segunda e Sexta) e três backups incrementais diários (8 em 8 horas).

**API** - Backup Integral sempre que exista uma alteração.

**Processo de mitigação de dados**

Numa situação de desastre, deve existir, para além dos servidores de backup para Load Balancing, um servidor responsável pelo Load balancing do tráfego recebido. Este servidor tem a responsabilidade de garantir que o tráfego recebido seja enviado apenas para servidores que estão operacionais ou com menos conexões. Cada servidor de armazenamento de dados, deverá ter a função de enviar recorrentemente as suas alterações para a Base de Dados que irá agregar e registar as alterações. Finalmente, é usada a estratégia de backup para a Base de Dados.