Macintosh HD:Users:alexandre:Desktop:LOGOS_DEPARTAMENTOS_PAPEL_DE_CARTA:CIENCIAS_ULISBOA_INFORMATICA_CMYK.eps



**Relatório de Situação do Projeto**(2ª Avaliação Periódica)

# Grupo 05

|  |  |  |
| --- | --- | --- |
|  | Nome | Número |
| 1 | André Silva | 58644 |
| 2 | Bernardo Caldas | 58650 |
| 3 | David Carreira | 58629 |
| 4 | Martim Emauz | 58668 |
| 5 | Miguel Martins | 58661 |
| 6 | Tomás Alves | 58633 |

EM FALTA

INSERIR LEGENDAS DAS IMAGENS **NO FIM**

VERIFICAR SE OS FICHEIROS DOS SCRIPTS E INSTRUÇÕES SÃO OS MAIS RECENTES

FORMATAR TEXTO

# Implementação da arquitetura distribuída

< Apresente nesta seção o diagrama atualizado da arquitetura do sistema e descreva-o detalhando as instâncias de VM e configurações de rede. >

## Implementação dos balanceadores de carga

1. **Configuração de sub-redes**

Para começar, criámos uma sub-rede “extnet” para que os balanceadores de carga comuniquem com os clientes, de forma a isolar a comunicação externa e interna.

Esta foi configurada com um intervalo de endereços IPv4 **10.10.0.0/20**, e na mesma região em que são configuradas as máquinas virtuais. Adicionalmente, ativámos a opção que define que as *VMs* na sub-rede não têm IPs externos atribuídos, para que estes sejam atribuídos dinamicamente através do *Google Compute Engine* (GCE):

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

1. **Criação de instâncias**

Criadas as redes, seguiu-se a criação das VMs responsáveis pelo balanceamento da carga. Criámos instâncias com nomes “load-balancer-X”, em que X corresponde ao número da ordem de criação dos balanceadores.

Olhando para o exemplo referido nos [slides](file:///C:\Users\Utilizador\Downloads\config_servers_PTI.pdf) disponibilizados relativamente aos Balanceadores, e através de pesquisa, utilizámos valores equivalentes aos sugeridos pelos docentes.

A **região escolhida** foi a mais barata, que nos permitisse ter proximidade ao público-alvo (Portugal) e máquinas disponíveis. Por isso, escolhemos criar as sub-redes e as *VMs* na Holanda.

Uma imagem com captura de ecrã, texto, file, preto

Descrição gerada automaticamente

No que toca à configuração das máquinas, seguimos as diretrizes dadas nos exemplos dos docentes, no que toca à quantidade de processadores virtuais (*vCPU*) e de memória. Além disso, feita alguma pesquisa, verificámos que era adequada a configuração sugerida para o tipo de carga de trabalho que esperamos enfrentar com a execução do serviço [1]. Desta forma, os balanceadores suportarão tráfego TCP e HTTP com até **4000 conexões por segundo**.

Dito isto, estabelecemos a configuração da máquina do tipo E2-micro, que disponibiliza um máximo de **2 *vCPU***, com **1 *core***, e **1GB de memória**.

O **custo** torna-se **reduzido**, tanto pela escolha de região, como da configuração das máquinas:

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

O **sistema foi operativo** escolhido foi o ***Ubuntu***, com a versão 20.04 LTS, pois era a versão estável mais recente disponível equivalente à versão 18.04 referida pelos docentes. Trata-se de uma distribuição que suporta todo o software necessário para a implementação do nosso projeto, além de todos estarmos familiarizados com o seu funcionamento. O tamanho do disco é de **10GB**, um valor que além de recomendado pelos docentes, será mais do que suficiente para hospedar todo o código e software necessário para a sua implementação:

Uma imagem com texto, captura de ecrã, Tipo de letra, software

Descrição gerada automaticamente

Definimos que as máquinas virtuais terão acesso a **todas as *APIs*** da *Cloud*:

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Nas **definições avançadas**, e na aba de **Redes**, definimos o tipo de interfaces de rede como gVNIC, a interface de rede virtual da Compute Engine:

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

As interfaces de rede implementam as sub-redes default, e extnet. A sub-rede default estabelecerá a comunicação entre os balanceadores de carga e os servidores web. A sub-rede extnet dará palco à comunicação entre os balanceadores e os pedidos provenientes dos clientes, bem como a comunicação VRRP entre ambos os balanceadores:

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Na interface de rede ***extnet***, configurámos o IP interno de ambas as máquinas para ser dinâmico. Este será o endereço que comunica com os servidores-web, direcionando o tráfego adequadamente entre as várias instâncias de servidores em vigor, através do algoritmo “***leastconn***”. O IP externo será um IP estático reservado, que será associado à máquina designada como mestre. Inicialmente, será associado ao balanceador de carga 1.

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

Na interface da rede ***default***, as máquinas terão IPs internos dinâmicos automaticamente designados, mas não terão IPs externos:

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

1. **Configuração de máquinas virtuais**

Após a sua inicialização, ambas as máquinas foram configuradas para utilizar o ***HAProxy*** e o ***Keepalived***. Estes serão responsáveis pela propriedade de tolerância a falhas do sistema de balanceamento de carga, e pela distribuição do fluxo de processos **de forma justa** entre as várias instâncias de servidores web.

O ficheiro **/etc/haproxy/haproxy.cfg** define os parâmetros da configuração do *HAProxy*. Este será parecido nas duas instâncias, na sua estrutura e parâmetros:

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Após alguma pesquisa, definimos que o algoritmo de balanceamento da carga mais adequado para o nosso sistema seria o ***leastconn***. Este distribui os pedidos pelas instâncias com menos ligações existentes, garantindo que o fluxo é distribuído de forma **equitativa** [2] [3]. No que toca aos valores máximos para os períodos de inatividade das várias conexões (**timeout**), fizemos alguma pesquisa no que toca à sua otimização [4] [5], e consideramos os valores sugeridos pelos docentes ideais.

O ficheiro **/etc/keepalived/keepalived.conf** estabelece a relação entre as máquinas e o seu papel no caso de uma falha. A linha “**state**” define o estado da máquina mestre como “MASTER” e o estado da máquina de reserva como “BACKUP”. O parâmetro “**priority**” estabelece a importância relativa das instâncias. A instância mestre terá um valor de 100, enquanto a outra terá um valor igual a 50.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

O ***script*** **etc/keepalived/takeover.sh** define as ações a cumprir se uma máquina deixar de verificar que a outra está funcional. Este envia comandos à *Google Cloud* para alterar a configuração da máquina alheia, removendo a associação desta ao IP externo, e associa-o a si mesma.

## Implementação dos web servers

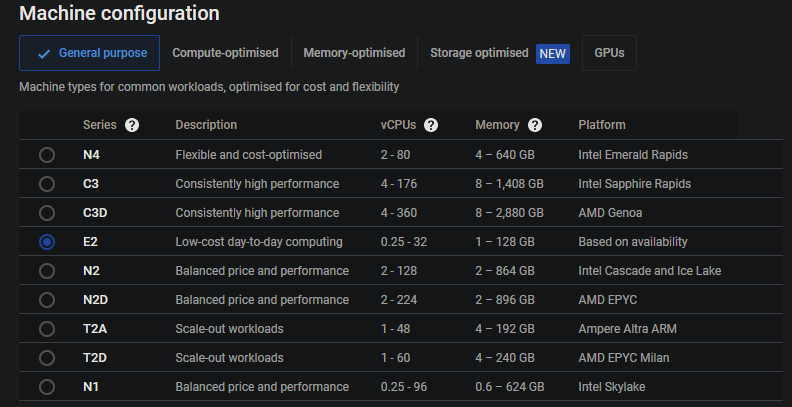
Criámos instâncias com o nome “servidor-X”, em que X corresponde ao índice do servidor sob a ordem da sua criação.

A **região escolhida** foi a mesma do que no caso dos balanceadores de carga, para garantir consistência:

Uma imagem com captura de ecrã, texto, file, preto

Descrição gerada automaticamente

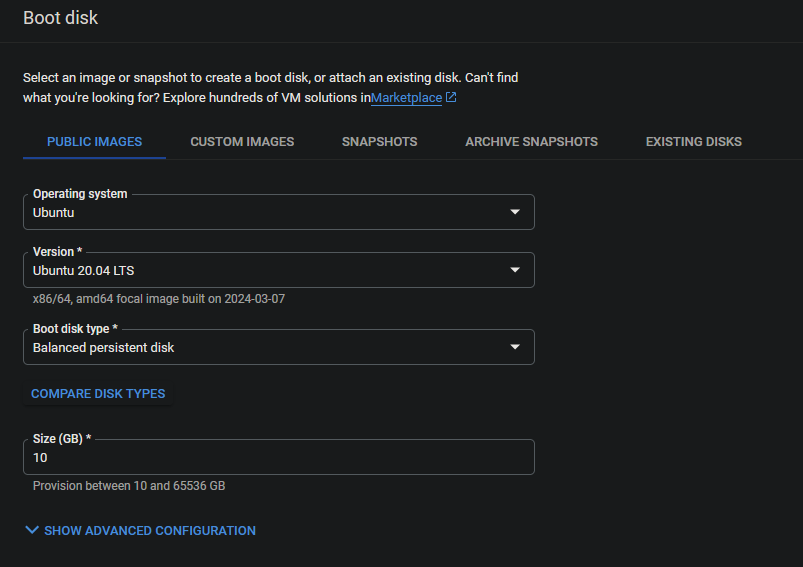
A **configuração das máquinas** no que toca à quantidade de vCPU e memória, foi idêntica à anteriormente referida:



Uma imagem com captura de ecrã, texto, Tipo de letra, file

Descrição gerada automaticamente

O **sistema operativo** escolhido foi o mesmo do das máquinas dos balanceadores de carga:



Nas opções da **firewall**, ativámos as opções para permitir comunicação HTTP e HTTPS.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Nas **opções avançadas**, definimos, na aba de Redes, a configuração de performance de rede como *gVNIC* e a interface de rede como *default*:

Uma imagem com texto, captura de ecrã, Tipo de letra, file

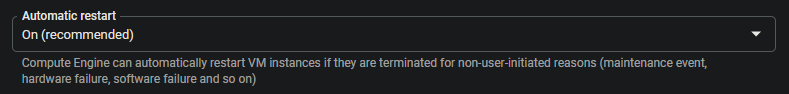
Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Na sub-rede default, as máquinas virtuais apenas terão a si associadas um IP externo cada uma.

Adicionalmente, verificámos a não-omissão da opção de *“auto-restart”* das máquinas, que é definida automaticamente.



Por fim, elaborámos um [script](file:///C:\Users\Utilizador\Downloads\script_init_server.txt) para automatizar e otimizar a inicialização dos servidores, que executa a parte das [instruções](file:///C:\Users\Utilizador\Downloads\instruções_web_servers.txt) necessárias para ter os servidores prontos. Em detalhe, este:

* Inicializa o modo *sudo*, para realizar as operações no sistema sem problemas com permissões de acesso, execução ou leitura;
* Atualiza a função “apt-get” do Linux, que nos deu problemas na instalação do apache nas primeiras tentativas, pois não encontra 4 recursos na instalação do mesmo;
* Instala o apache2, e emite a resposta “Y”, quando a instalação pergunta se deve proceder;
* Ativa e inicializa os serviços do apache2;
* Atualiza os conteúdos do ficheiro /var/www/html/index.html, Introduzindo “Servidor X” no título e num parágrafo na página HTML.Com recurso aos meta dados da própria instância, conseguimos recolher o nome dado à instância na GCP e, consequentemente, armazenar o número correspondente ao índice do servidor em questão. Com este número, podemos atualizar automaticamente os dados da página HTML referida, sem que ninguém tenha que o fazer manualmente.

## Implementação das bases de dados

< Detalhe a implementação dos balanceadores das bases de dados. >

# Escalabilidade

< Explique os mecanismos de escalabilidade implementados.>

# Tolerância a Faltas

< Explique os mecanismos de tolerância a falta implementados.>