

UNIVERSIDADE DO MINHO
MESTRADO EM ENGENHARIA INFORMÁTICA

APLICAÇÕES E SERVIÇOS DE COMPUTAÇÃO EM NUVEM

**Automatização do Processo de Instalação,
Configuração, Monitorização e Avaliação
da Aplicação Ghost**

Grupo 39



Daniel Xavier
PG50310

Diogo Rebelo
PG50327

Henrique
Alvelos
PG50414

João Cerquido
PG50469

Júlio Gonçalves
PG50527

Repositório GitHub

14 de janeiro de 2023

Conteúdo

1	Introdução	3
2	Ghost	3
3	Arquitetura e Componentes	4
4	Ferramentas e Abordagem utilizadas para a Instalação e Configuração	5
4.1	MySQL Deployment	5
4.2	Ghost Deployment	5
5	Abordagem escolhida para o mecanismo de replicação	6
6	Abordagem escolhida para o mecanismo de persistência de dados	6
7	Ferramentas de Monitorização, Métricas e Visualização	7
8	Conclusão	10

1 Introdução

No âmbito da UC de Aplicações e Serviços de Computação em Nuvem, foi-nos proposta a tarefa de automatizar o processo de instalação, configuração, monitorização e avaliação da aplicação *Ghost*. Estes processos enumerados são fundamentais para garantir a eficiência e a qualidade do sistema, sendo que com a sua automatização é possível garantir que a aplicação seja instalada e configurada corretamente, que seja monitorada constantemente para detetar problemas e que seja avaliada regularmente para garantir que está a satisfazer as necessidades do utilizador. Uma das principais vantagens da automatização é a redução dos erros humanos, pois a maioria dos erros ocorrem devido a procedimentos manuais. Além disso, a automatização permite a otimização dos processos, o que resulta num melhor desempenho do sistema. Para além disso, existe um conjunto de objetivos que se pretendem ver concretizados no final do desenvolvimento do presente trabalho, nomeadamente:

- Compreender detalhadamente as diferentes fases do processo de automatização subjacente;
- Utilização de várias ferramentas, nomeadamente *Ansible*, na automatização no serviço GKE da *GCP*;
- Utilização de ferramentas de monitorização que permitam a visualização da instalação da aplicação *Ghost*, sendo possível analisar os resultados produzidos;
- Adquirir competências de avaliação experimental, através da avaliação dos resultados gerados pela monitorização da aplicação;
- Analisar os conceitos de escalabilidade, resiliência e desempenho, através do estudo da aplicação, perante aumento de carga;
- Desenvolvimento de *skills* transversais, nomeadamente, características específicas do ambiente de desenvolvimento da solução encontrada pelo grupo, características inerentes ao *deployment* da aplicação, capacidade de discussão, perante a existência de diferentes arquiteturas, no contexto do desenvolvimento de aplicações na *Cloud*.

2 Ghost



Figura 1: Ghost.org

Ghost.org é uma plataforma de *blogging* baseada em nuvem, que permite aos usuários criar e gerenciar os seus próprios blogs de forma fácil e rápida. Com o Ghost, os usuários podem escrever *posts*, adicionar imagens e vídeos e até personalizar o *layout* do blog.

Além disso, a plataforma oferece integrações com ferramentas de marketing, permitindo que os utilizadores promovam e optimizem os seus *blogs* para os vários motores de busca. É uma plataforma em nuvem que oferece flexibilidade e escalabilidade, tornando possível gerir os blogs em qualquer lugar e em qualquer momento.

A existência de várias vantagens a nível de escalabilidade e flexibilidade na implementação tornam esta plataforma um candidato vantajoso para a construção e gestão de aplicações na *Cloud*, onde essas mesmas características, em contextos específicos, podem ser críticas.

3 Arquitetura e Componentes

Definir a arquitetura num sistema distribuído como este é fundamental por diversas razões porque terá muito impacto na escalabilidade, disponibilidade, resiliência, desempenho e segurança do mesmo. Assim é importante ter uma arquitetura pensada e bem estruturada para garantir que a aplicação possa lidar com estes fatores. Pelo que a arquitetura do Ghost instalada num cluster do Google Cloud inclui os seguintes componentes:

Componentes	Descrição
Nós do cluster	Instâncias da VM que executam os pods da aplicação. Podem ser escalados de acordo com as necessidades da aplicação.
Pods	Unidades de execução do Kubernetes que contêm um ou mais containers.
Containers	Unidades de software que contêm a aplicação Ghost e os seus componentes dependentes. São executados dentro dos pods e podem ser facilmente escalados e movidos entre os nós do cluster.
Persistência de dados	A aplicação pode usar um banco de dados para armazenar dados permanentes, como posts de blog e configurações. Neste caso é usado o MySQL.
Load Balancer	Distribui o tráfego da aplicação entre os pods disponíveis, garantindo a disponibilidade e escalabilidade da aplicação.
Gestão de Configuração	É usado o Ansible para garantir que as configurações da aplicação sejam consistentes em todos os pods.
Monitorização	Foram utilizadas as ferramentas do Google Cloud.

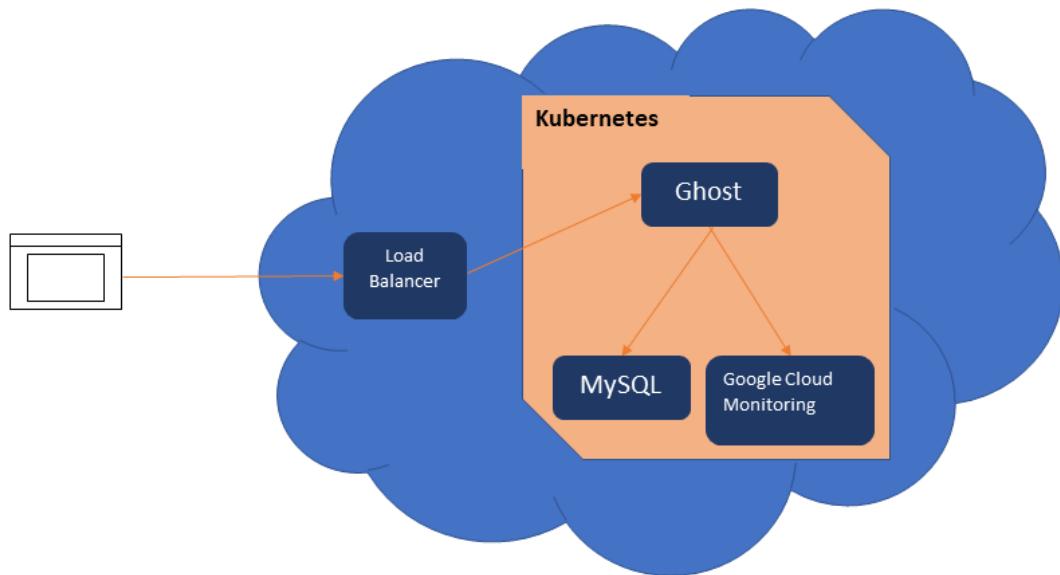


Figura 2: Arquitetura da aplicação

4 Ferramentas e Abordagem utilizadas para a Instalação e Configuração

A abordagem utilizada para a instalação e configuração automática da aplicação Ghost no serviço Kubernetes da Google Cloud Platform foi o uso de Ansible Playbooks.

Através desta ferramenta é possível escrever scripts para automatizar tarefas de instalação e configuração, incluindo a criação de recursos do Kubernetes, como pods e serviços.

Deste modo, a instalação e configuração do nosso serviço Ghost pode-se repartir em 2 fases:

- **Criação de um Cluster:** Através de um playbook *create-gke-cluster.yml* previamente fornecido, criamos um cluster com dois nodos;
- **Deployment da Aplicação:** Através do playbook *deploy-ghost.yml* são criados 2 pods e 2 serviços, para a aplicação Ghost e para a aplicação MySQL que vai fornecer serviços de armazenamento de dados à aplicação Ghost.

4.1 MySQL Deployment

O processo de deployment do serviço MySQL visa dar suporte e garantir as funcionalidades da nossa aplicação Ghost com um sistema de base de dados.

O nosso playbook cria um ponto de controle de volume (PVC) para armazenar os dados do MySQL de forma persistente, um serviço para expor a base de dados futuramente e um deployment para gestão dos pods do MySQL. Tanto o IP deste serviço como o nome do Pod a ser utilizado são guardados para estabelecermos a conexão ao deployment do Ghost.

4.2 Ghost Deployment

Começamos por expor o serviço Ghost para que possa ser acedido externamente utilizando as portas 80 (cluster) e 2368 (container) para expor o serviço. Antes de prosseguir com o deployment, esperamos que o LoadBalancer nos forneça um IP atualizado. Com o IP atualizado, passamos para o normal deployment da aplicação Ghost.

Por fim, é adicionado automaticamente um *admin* com as credenciais de acesso fornecidas no enunciado pelo que após inseridas o mesmo tem acesso à seguinte dashboard:

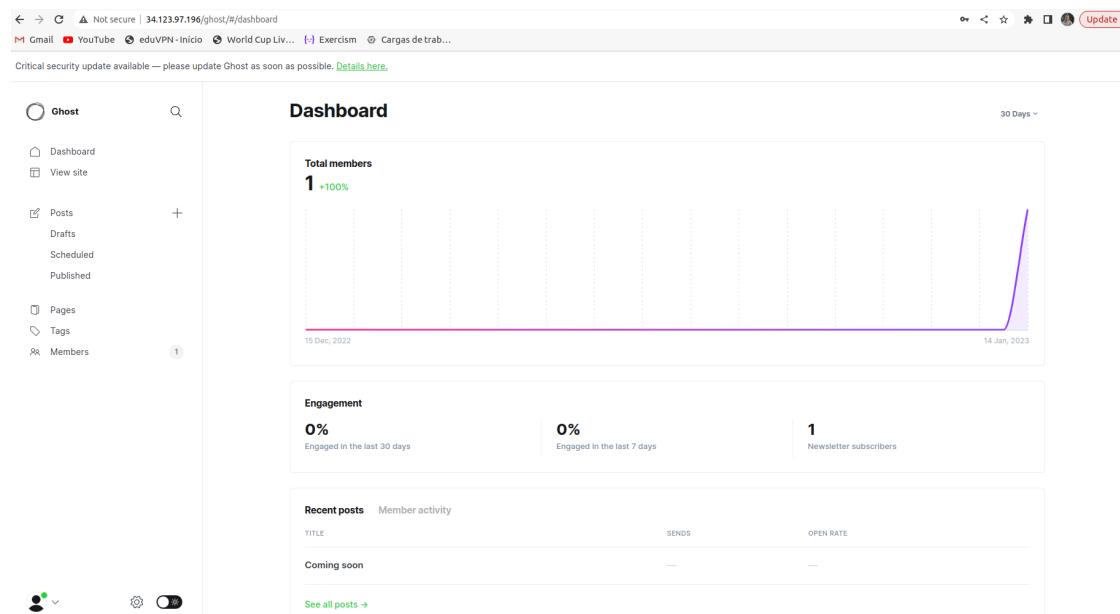


Figura 3: Dashboard do Admin

5 Abordagem escolhida para o mecanismo de replicação

A abordagem escolhida para o mecanismo de replicação é o uso de um LoadBalancer. O LoadBalancer permite redirecionar e gerir a distribuição do tráfego entre os diferentes pods do sistema. Ele é responsável por redirecionar as solicitações dos utilizadores para os pods disponíveis e garantir que cada pod não seja sobre-carregado com trabalho.

Deste modo, com o uso de um LoadBalancer, permitimos que o nosso serviço seja escalável, ou seja, que possa acompanhar o crescimento do número de utilizadores e aumentar o número de pods conforme necessário. Além disso, o LoadBalancer garante a alta disponibilidade do sistema, pois ele redireciona as solicitações para outros pods quando um falha.

6 Abordagem escolhida para o mecanismo de persistência de dados

De modo a garantir a persistência dos dados quando é executado através do ansible o playbook *undeploy_ghost.yml*, este apenas termina a execução da instalação do Ghost pelo que é necessário passar a flag(*-e delete_data=true* para remover todos os dados persistentes da instalação. Assim quando colocada a flag é executada a role *full_undeploy* que remove o PVC do MySQL bem como os PV associados ao mesmo. Sem a flag apenas remove os pods, deployments e serviços.

7 Ferramentas de Monitorização, Métricas e Visualização

Com o intuito de aplicar ferramentas de monitorização que permitissem observar a instalação da aplicação Ghost, exploramos a ferramenta de monitorização disponibilizada pela plataforma Google Cloud.

A monitorização é fundamental ao projeto uma vez por que verifica os seguintes domínios da disponibilidade e do desempenho uma vez que monitora a saúde e permite identificar e resolver problemas rapidamente, garantindo que a aplicação esteja sempre disponível para os utilizadores e monitora o uso de recursos como CPU, memória, rede e I/O de disco permite identificar gargalos de desempenho e tomar medidas para melhorar o desempenho da aplicação.

A escolha da dashboard, na figura seguinte, foi devidamente analisada e discutida pelo grupo para conter as métricas de monitorização mais importantes que explicaremos de seguida.

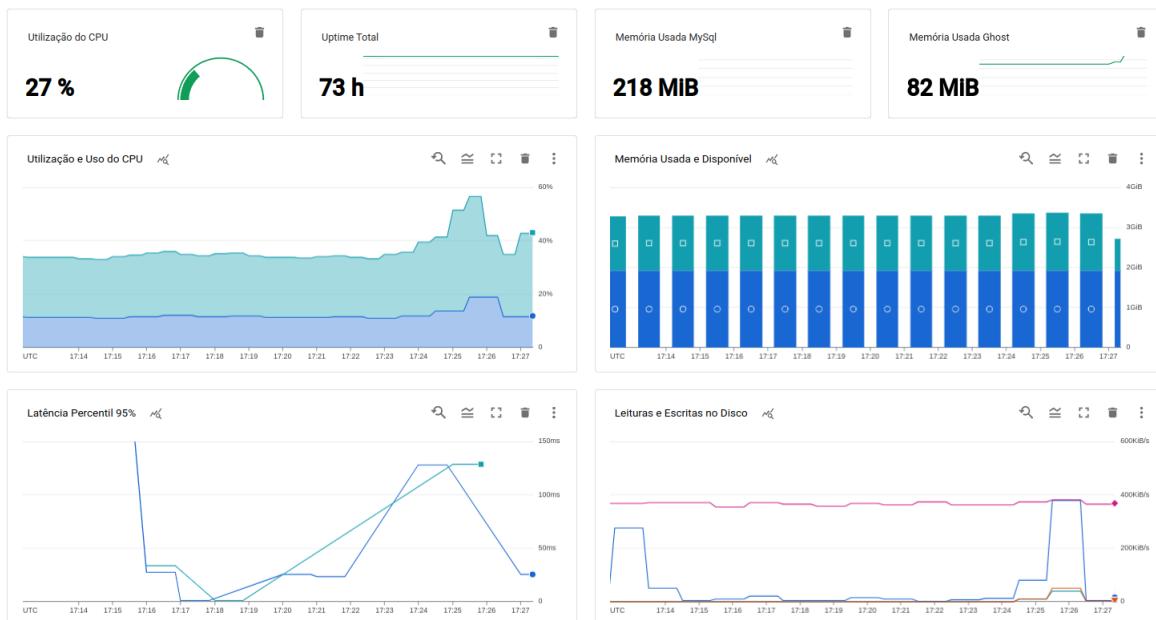


Figura 4: Dashboard de Monitorização

Assim segue uma explicação detalhada das métricas e gráficos escolhidos: Como métrica primordial colocamos o CPU e o seu desempenho. Assim no primeiro gráfico combinamos duas métricas essenciais:

- Utilização do CPU (a verde): refere-se à percentagem do CPU que está a ser utilizado em relação à capacidade total de CPU.
- Uso do CPU (a azul): refere-se à quantidade de tempo que o CPU está a ser usado para executar tarefas.

Por outras palavras, a "utilização da CPU" mede o quanto ocupado está o processador, enquanto o "uso da CPU" mede quanto tempo ele está a ser usado para realizar tarefas.

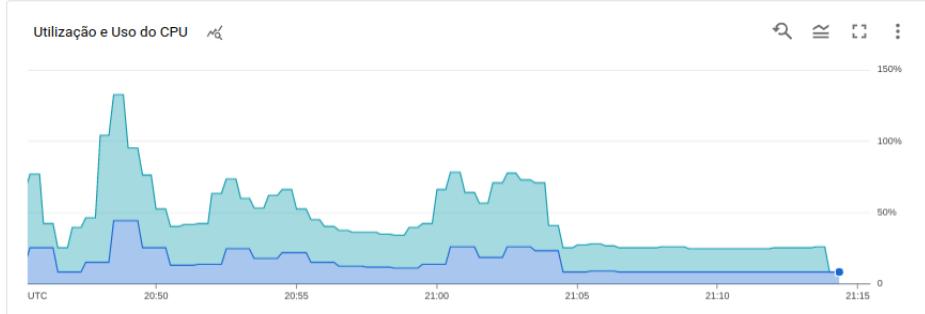


Figura 5: Uso e Utilização do CPU

Para evidenciar a métrica de utilização de CPU colocamos a mesma no topo bem como o uptime, tempo desde que a aplicação está disponível.

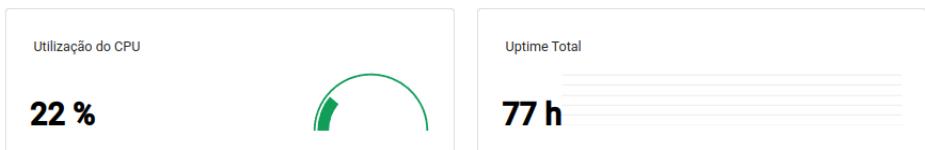


Figura 6: Utilização do CPU e *Uptime*

Outra métrica primordial, a par do CPU, é a Memória. Assim para fazer a monitorização combinamos as métricas Memória Usada (a verde) e Memória Disponível (a azul). Realizar a monitorização da memória usada e disponível é importante por várias razões como a identificação de gargalos, ou seja, se a memória disponível estiver a diminuir, pode ser sinal de que os recursos estão a ser subutilizados, fazendo com que o sistema fique lento ou travado. Outra razão é para fazer uma correta otimização dos recursos e escalá-los de modo a melhorar o desempenho da aplicação. Permite também uma prevenção de falhas ou seja se a memória disponível for insuficiente, pode ocorrer uma falha no sistema, causando a interrupção do funcionamento da aplicação.

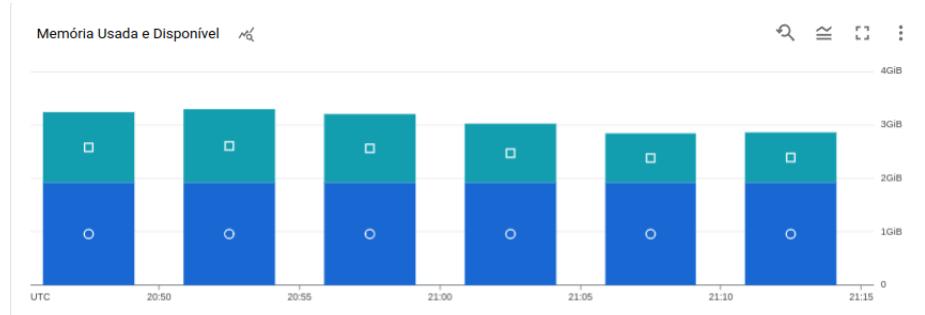


Figura 7: Memória Usada e Disponível

Contudo, como é importante verificar o uso da memória por parte do MySQL e do Ghost de forma separada colocamos evidenciados esses valores.

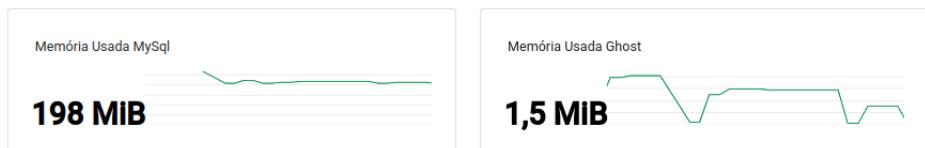


Figura 8: Memória usada pelo MySQL e Ghost

Outro ponto fundamental da monitorização é as leituras e escritas, uma vez que é esta comunicação com o disco que causa mais perda na performance. Assim incluímos mais um gráfico que inclui estes dados para todas as instâncias.

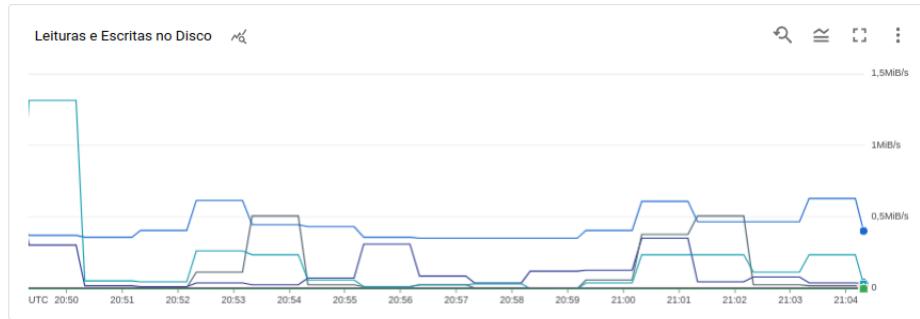


Figura 9: Leituras e escritas ao Disco

Por fim decidimos monitorizar a latência, isto é, o tempo que leva para que uma ação ocorra após um pedido ser feito. Esta métrica também é fundamental uma vez que latência elevada pode indicar problemas de rede ou de sobrecarga de sistemas que podem afetar a disponibilidade da aplicação e pode afetar a velocidade com que as solicitações são processadas e, consequentemente, afetar o desempenho da aplicação pelo que uma latência reduzida é fundamental para o utilizador ter a melhor experiência possível.

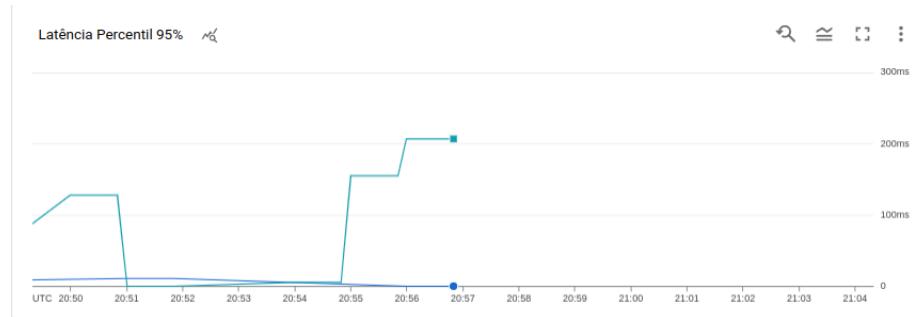


Figura 10: Latência

8 Conclusão

Na sua globalidade estamos bastante contentes com o trabalho desenvolvido e pensamos que atingimos os objetivos propostos neste trabalho seguindo as técnicas e os padrões de desenvolvimento neste que foi o nosso primeiro contacto Aplicações e Serviços de Computação em Nuvem, primordial à nossa formação enquanto Engenheiros Informáticos uma vez que esta área é o futuro.

Esta tarefa de instalação e configuração automática da aplicação e implementação permitiu-nos assim "meter a mão na massa" e colocar em prática o que foi lecionado nas aulas práticas.

Embora pensamos ter cumprido grande parte os objetivos propostos, em todos os projetos de grande dimensão e que envolvem bastante trabalho e rigor, há sempre pontos em que apresentamos mais dificuldades, e este projeto não foi exceção.

Pelo que os nossos principais pontos fracos e dificuldades recaíram principalmente na parte inicial do projeto relativamente a como arquiteturar e desenvolver a solução, resolvendo erros com pouca informação. Todavia, apresentamos uma melhor facilidade a trabalhar como grupo, discutindo profusas vezes as ideias apresentadas mas mantendo o espírito de equipa como base do progresso do trabalho.

Contudo, poderíamos estender o trabalho e implementar futuramente novas funcionalidades como o serviço de *mail* e implementar a avaliação experimental e escalabilidade e resiliência à aplicação.