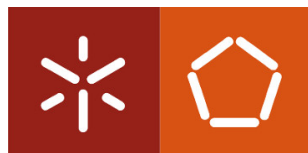


UNIVERSIDADE DO MINHO

ESCOLA DE ENGENHARIA



# Aplicações e Serviços de Computação em Nuvem

Mestrado em Engenharia Informática

## Instalação automatizada, configuração, monitorização e avaliação da aplicação Ghost

Trabalho Prático - Grupo 31

<https://github.com/FranciscoReisIzquierdo/Projeto-de-ASCN.git>

Francisco Novo - [PG50374]  
Francisco Izquierdo - [PG50384]  
João Carvalho - [PG51243]  
Mário Correia - [PG51246]  
Tiago Ribeiro - [PG50779]

Janeiro, 2023

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Instalação e Configuração Automática da Aplicação</b>	<b>3</b>
2.1	<i>Deployment</i> da aplicação Ghost . . . . .	3
2.1.1	<i>Deploy</i> do MySQL . . . . .	3
2.1.2	<i>Deploy</i> do Ghost . . . . .	4
2.2	<i>Undeployment</i> da aplicação Ghost . . . . .	5
<b>3</b>	<b>Funcionalidades da aplicação Ghost</b>	<b>6</b>
<b>4</b>	<b>Monitorização</b>	<b>8</b>
<b>5</b>	<b>Avaliação experimental</b>	<b>9</b>
<b>6</b>	<b>Conclusão</b>	<b>10</b>

# 1. Introdução

O trabalho prático da unidade curricular de Aplicações e Serviços de Computação em Nuvem tem como objetivo aplicar os conhecimentos adquiridos durante o decorrer do semestre para automatizar o processo de instalação, configuração, monitorização e avaliação da aplicação *Ghost*. Esta aplicação é uma plataforma *open-source* de *blogs*, permitindo aos seus utilizadores publicar conteúdos num *blog*.

O trabalho prático está dividido em várias tarefas distintas:

- Tarefa Base: utilizar a ferramenta *Ansible* para automatizar a instalação e configuração da aplicação *Ghost* no serviço *Google Kubernetes Engine (GKE)* da *Google Cloud*. Esta é a tarefa principal do trabalho prático, servindo como base para a implementação das tarefas seguintes.
- Tarefa 1: Monitorização - aplicação de ferramentas de monitorização que permitam observar a instalação da aplicação *Ghost*.
- Tarefa 2: Avaliação Experimental - demonstração do método para validar experimentalmente a instalação e configuração da aplicação *Ghost*.
- Tarefa 3: Escalabilidade e Resiliência - dotar a aplicação de mecanismos de replicação que permitam melhorar o desempenho e/ou resiliência da instalação da aplicação *Ghost*.

O desenvolvimento deste trabalho prático consiste também na criação de 2 *playbooks* que têm como função: instalar, configurar e executar todos os componentes da aplicação *Ghost* no *cluster* GKE criado e efetuar a desinstalação destes componentes, com a opção de não eliminar os dados presentes na base de dados.

## 2. Instalação e Configuração Automática da Aplicação

A primeira tarefa deste trabalho prático consiste em utilizar a ferramenta Ansible para automatizar a instalação e configuração da aplicação Ghost no serviço Google Kubernetes Engine (GKE) da Google Cloud.

### 2.1 *Deployment* da aplicação Ghost

Para instalar, configurar e executar a aplicação Ghost no cluster GKE, é utilizado o play-book *deploy-ghost.yml*. Este *playbook* consiste em três *plays*, que são conjuntos de instruções que são executadas de forma sequencial.

A primeira *play*, chama o role *create\_namespace*. É criado um *namespace* no cluster onde o Ghost e os seus componentes serão implementados. Esta *play* utiliza a função *create\_namespace*, que contém tarefas para criar o mesmo, sendo que este tem o nome "*namespace-ghost*". Todo o processo de *deployment* irá ocorrer dentro deste *namespace*.

A segunda *play*, cria um secret no *namespace* que contém os detalhes de ligação à base de dados MySQL que o Ghost irá utilizar. Esta *play* chama o role *create\_secret*, que utilizando o ficheiro *secret.yml*, que é um ficheiro encriptado pelo Ansible Vault, cria o *secret* no *namespace*. Este contém as várias palavras-passe e dados sensíveis utilizados ao longo do *deployment*.

A terceira *play*, implementa a base de dados MySQL e a aplicação Ghost no *namespace* criado na primeira *play*. Esta *play* chama os roles *deploy\_mysql* e *deploy-ghost*, que contém tarefas para implementar o MySQL e o Ghost, respetivamente.

#### 2.1.1 *Deploy* do MySQL

No role de *deploy* do *MySQL*, este é utilizado para implementar uma base de dados MySQL no *cluster* Kubernetes. O role consiste em três tarefas:

Inicialmente é criado um serviço no *cluster* que expõe a base de dados MySQL a outros componentes no *cluster*. Esta tarefa utiliza o módulo *kubernetes.core.k8s* para criar o serviço usando um ficheiro de configuração chamado *definition*. O serviço é criado no "*namespace-ghost*" e tem o nome "*mysql-service*", que também será o nome DNS do serviço. O serviço escuta na porta especificada na variável "*mysql-port*" e expõe-a internamente no *cluster*.

Depois, este role cria uma *Persistent Volume Claim* (PVC) no *cluster*, que é utilizada para armazenar os dados da base de dados MySQL. A PVC é criada no espaço de nome especificado na variável *namespace* e tem o nome "*mysql-pv-claim*". É configurada para utilizar a classe de armazenamento standard e solicita um armazenamento de 20GB.

Por fim, é criado um objeto Deployment no *cluster* que é utilizado para implementar a base de dados *MySQL* num *pod*. O Deployment é criado com o nome *"mysql-deployment"*. Este utiliza a imagem *mysql* presente no repositório do *Docker* e expõe a porta especificada na variável *"mysql\_port"*. O Deployment também inclui as variáveis de ambiente necessárias para ligar à base de dados e utiliza o *secret* *"mysql-secret"* para recuperar a palavra-passe de *root* e a palavra-passe do utilizador. O Deployment monta a PVC, criada na segunda *task*, no *pod* e utiliza-a para armazenar os dados da base de dados *MySQL*.

### 2.1.2 Deploy do Ghost

O role de *deploy* do *MySQL* é utilizado para implementar a aplicação Ghost, que é uma plataforma de *blogging* de código aberto popular, num *cluster* Kubernetes. O role consiste em oito tarefas:

Para iniciar é criado um endereço no *Google Cloud Platform (GCP)* que será utilizado para expor a aplicação Ghost ao exterior. Esta tarefa utiliza o módulo *google.cloud.gcp-compute-address* para criar o endereço usando o nome *"address-ghost"* e a região *"us-central1"*. O endereço é criado no projeto especificado na variável *"gcp\_project"* e os detalhes de autenticação são especificados na variável *"gcp\_auth\_kind"* e no ficheiro *"gcp\_cred\_file"*. O endereço é guardado como uma variável para ser utilizado nas *tasks* seguintes.

Depois, são atualizadas duas variáveis que são utilizadas nas *tasks* seguintes: *"ghost\_ip"* com o endereço criado na primeira tarefa e *"ghost\_port"* com o valor da porta onde é exposta a aplicação, que neste caso será 80.

De seguida, é criado um serviço no *cluster* que expõe a aplicação Ghost ao exterior. O serviço é criado no espaço de nome especificado na variável *"namespace"* e tem o nome *"ghost-service"*, que também será o nome DNS do serviço. O serviço é do tipo LoadBalancer e usa o endereço do endereço GCP criado na primeira tarefa. O serviço escuta na porta especificada na variável *"g\_port"* internamente no *cluster* e expõe a porta 80.

É criado um objeto Deployment no *cluster* que é utilizado para implementar a aplicação Ghost num *pod*. O Deployment é criado no espaço de nome especificado na variável *"namespace"* e tem o nome *"ghost"*. O Deployment utiliza a imagem da aplicação Ghost especificada na variável *"ghost\_version"*, que neste caso será a 5.14.1 tal como aconselhado pelos docentes, e escuta a porta especificada na variável *"g\_port"*. O Deployment também inclui as variáveis de ambiente necessárias para ligar à base de dados *MySQL* e utiliza o serviço *"mysql-service"* para se ligar à base de dados. O Deployment também inclui a variável *"url"*, que é definida a partir do endereço do endereço GCP criado na primeira tarefa e permite aceder à aplicação através do browser. Além disto também é configurado o serviço de email, que utiliza o *Mailgun*, um serviço de email gratuito, que vai efetuar o envio de emails a partir do protocolo SMTP.

Nas duas *tasks* seguintes é obtido o nome do *pod* do *MySQL*, que irá ser utilizado numa tarefa posterior e é realizado um período de pausa de 60 segundos, para permitir a inicialização de todos os componentes da aplicação. O módulo *kubernetes.core.k8s\_info* é utilizado para obter o nome do *pod MySQL* que é utilizado para criar um administrador do *blog* Ghost na aplicação.

Após a espera é inicializada a *task* seguinte que consiste na criação de um administrador de blog, sendo este criado na base de dados a partir de uma query MySQL. Com o módulo *kubernetes.core.k8s\_exec* é acedido o *pod* do MySQL e executada a query com os dados do administrador a introduzir na base de dados. Esta query é repetida até ser bem sucedida e a entrada do utilizador estiver presente na base de dados.

Na última *task* são criadas as *dashboards* utilizadas para a monitorização da aplicação Ghost. Estas são criadas a partir de um ficheiro de configuração denominado *dashboards.json*.

## 2.2 *Undeployment* da aplicação Ghost

O *undeployment* da aplicação *Ghost* é feito através do *playbook* *undeploy-ghost.yml* e tem como função eliminar a instalação atual da aplicação *Ghost*. Este *playbook* pode ser executado com a *flag* *delete\_data*.

A primeira play chama o role *undeploy\_ghost* que trata de eliminar os serviços e *Pods* do MySQL e do Ghost, tal como o *secret* e as *dashboards* criadas. Estes componentes são eliminados através de comandos da ferramenta *kubectl*.

A segunda play só é executada quando a *flag* é definida como *true* e elimina o *persistent storage* criado anteriormente para armazenar os dados do MySQL e o *namespace*. Se a *flag delete\_data* não for definida ou o seu valor for *false* este procedimento não é realizado.

### 3. Funcionalidades da aplicação Ghost

Neste capítulo são demonstradas algumas das funcionalidades presentes neste *deployment* da aplicação Ghost.

Depois de efetuada a instalação e configuração da aplicação Ghost, é possível aceder à mesma através de um browser. Ao aceder ao caminho `"/ghost"` é possível encontrar a página de autenticação do administrador de blog. A imagem seguinte mostra a página com que um utilizador se depara ao aceder ao Ghost e a página de *login* do administrador.

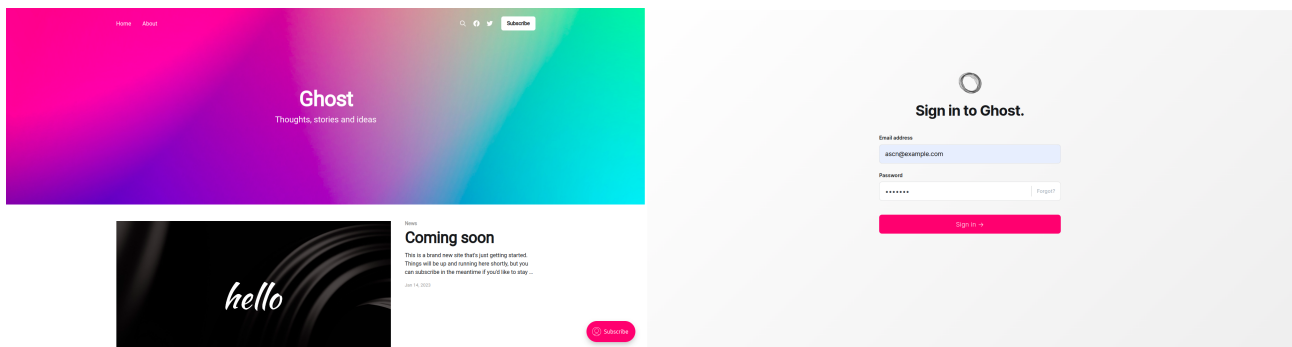


Figura 3.1: Página inicial do Ghost e página de *login* do administrador

Depois de o administrador efetuar o *login*, o administrador encontra uma página onde pode ver métricas acerca dos utilizadores que acederam à aplicação, criar uma novo *post*, ver a lista de *posts* ou até visualizar os utilizadores registados.

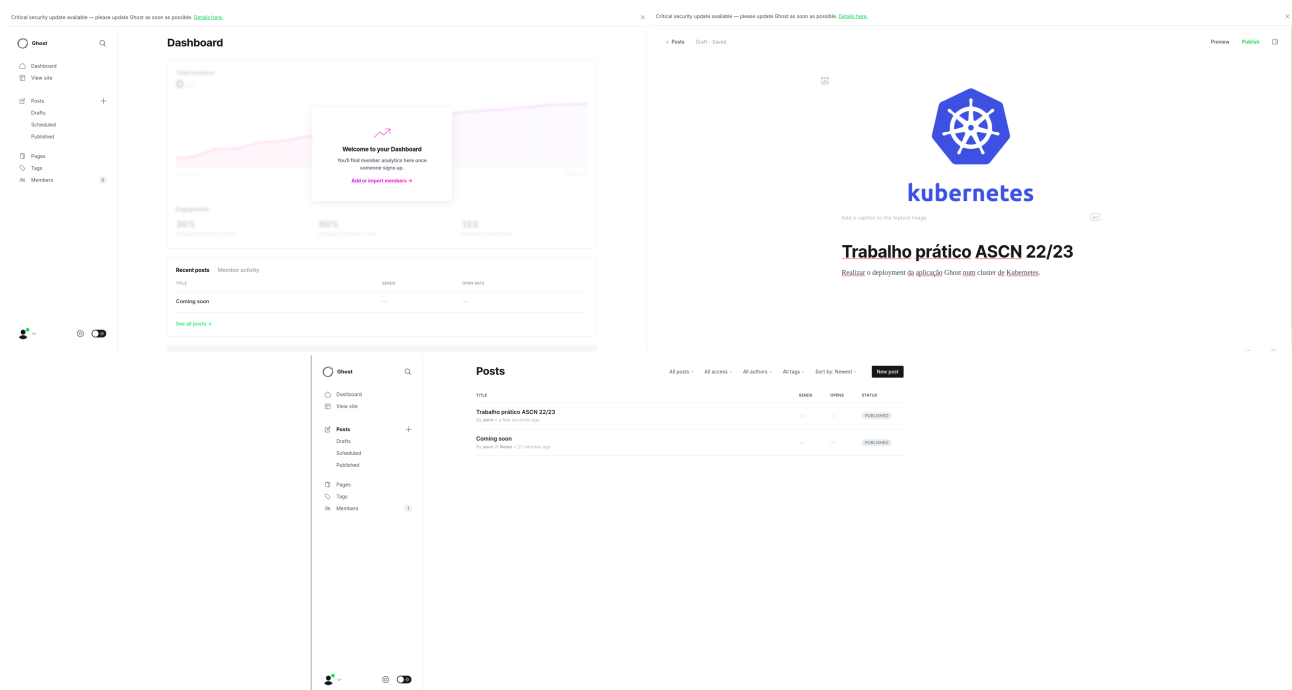


Figura 3.2: Página principal do administrador e criação de um post

Um utilizador que se quer registar prime o botão *Subscribe* da página inicial e efetua o registo da sua conta, utilizando o seu nome e o endereço de email. Depois irá receber um email para confirmar a criação da sua conta.

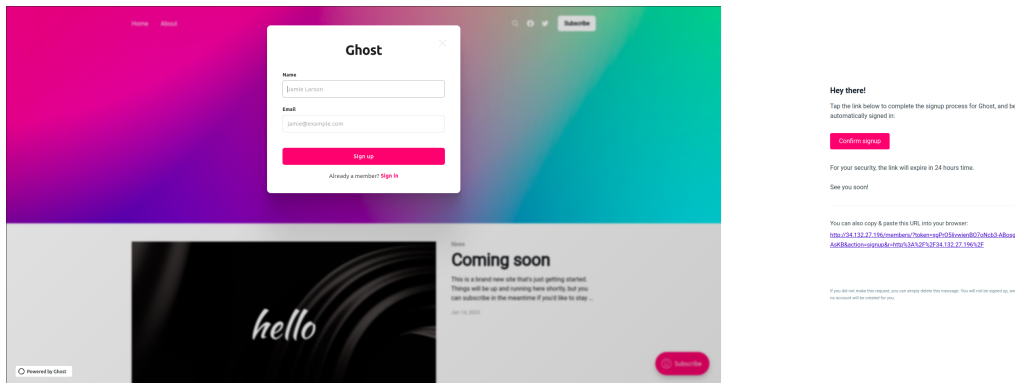


Figura 3.3: Página de registo de um utilizador e email recebido

Depois de confirmada a criação da conta, o utilizador pode começar a visualizar os *posts* publicados.

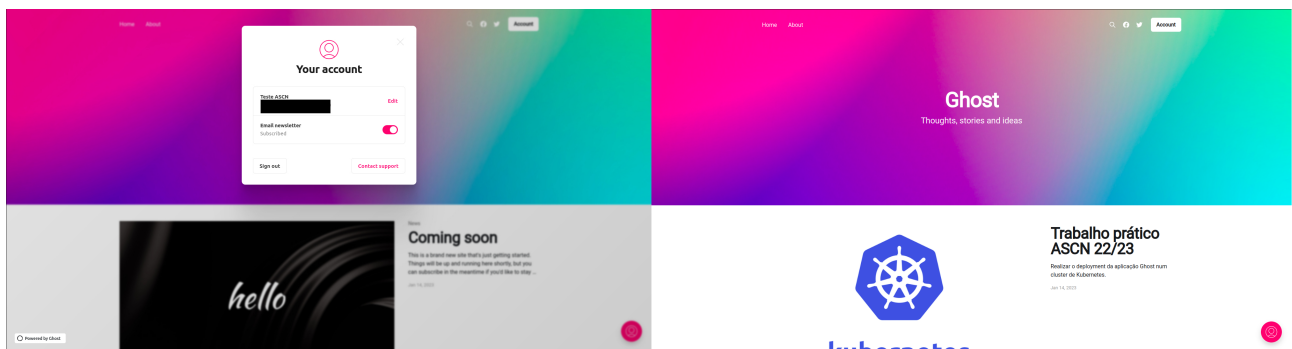


Figura 3.4: Página da conta de um utilizador e página principal com o novo *post*

A *dashboard* e as métricas a que o administrador tem acesso, são atualizadas à medida que novos utilizadores se juntam ao sistema.

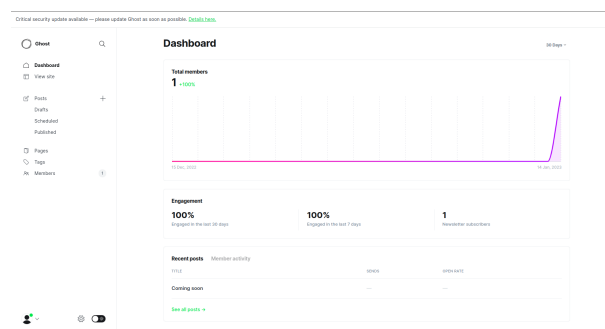


Figura 3.5: *Dashboard* e métricas do administrador atualizadas

Ao efetuar o *deployment*, seguido de um *undeployment* da aplicação sem a eliminação dos dados, todos os dados são mantidos e é possível aceder às contas de utilizador e visualizar os *posts* efetuados no *deployment* antigo.



## 4. Monitorização

Ao executar o *playbook* de deployment da aplicação Ghost são criadas várias *dashboards*, que podem ser visualizadas a partir do Google Cloud, na aba de Monitoramento.

As *dashboards* criadas permitem visualizar dados acerca do CPU, I/O e utilização de memória acerca do *pod* do Ghost. Estas contêm métricas como a utilização do CPU, número de bytes recebido e transmitidos, número de reinicializações e uso de memória. Todas estas métricas são importantes e permitem efetuar uma avaliação geral da performance do sistema de uma forma rápida e interativa.



Figura 4.1: *Dashboards* criadas

## 5. Avaliação experimental

Foram executados vários testes automáticos para avaliar a performance da instalação da aplicação Ghost.

Em primeiro lugar foi criado um grupo de 300 threads que com um delay de 300 ms enviam um pedido *HTTP GET* à página inicial da aplicação.

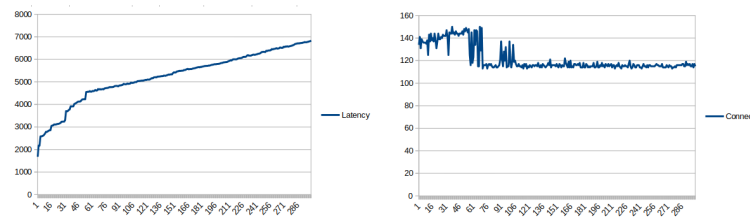


Figura 5.1: Teste 1

Todas as mensagens foram efetuadas com sucesso, com um tempo médio de 121 ms.

O mesmo teste foi executado para a página de login do administrador.

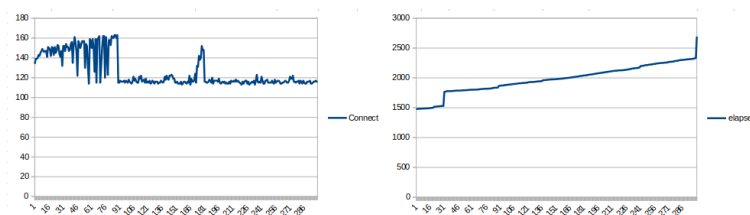


Figura 5.2: Teste 2

Todos os pedidos foram efetuados com sucesso com um tempo médio de 1973 ms, sendo que o último pedido foi tratado aos 2692 ms. Neste teste, os pedidos tiveram de ser redirecionados para o caminho específico (*HTTP 301 Moved Permanently*).

Por último foi efetuado o teste na página do *post* existente por defeito.

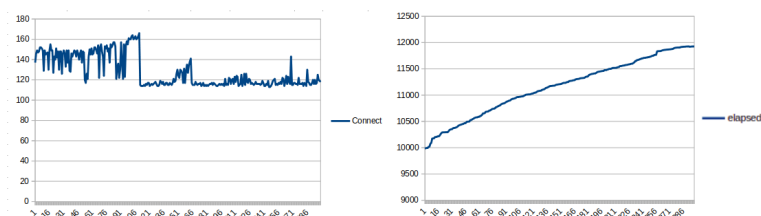


Figura 5.3: Teste 3

Todos os pedidos foram tratados com sucesso com um tempo médio de 11141 ms. Tal como no teste anterior os pedidos tiveram de ser redirecionados. Comparativamente ao último teste, este teve um tempo maior, pois a página onde foi realizado o pedido *HTTP GET* tem mais conteúdo. O pedido que mais tempo demorou a tratar durou 11926 ms.

## 6. Conclusão

Dando como concluído este trabalho prático, podemos agora refletir sobre o trabalho desenvolvido. Sentimos que, à exceção da Tarefa 3 de Escalabilidade e Resiliência, o projeto foi implementado com sucesso, permitindo-nos efetuar automaticamente o processo de instalação, configuração, monitorização e avaliação da aplicação *Ghost* bem como adquirir conhecimentos sobre a aplicação *Ghost*, a ferramenta *Ansible* e sobre o serviço *Google Kubernetes Engine (GKE)* da Google Cloud.

Analisando o trabalho, podemos concluir que a utilização do *Ansible* para automatização elimina a necessidade de efetuar estes passos manualmente, o que torna a instalação mais rápida, eficiente, menos propensa a erros e mais consistente. Podemos também reter que o serviço *Google Cloud* disponibiliza várias ferramentas bastante úteis tais como a ferramenta de monitorização com a qual podemos visualizar vários *dashboards* que contêm dados acerca de várias métricas relacionadas com o deployment da aplicação *Ghost*.