

Aplicações e Serviços de Computação em Nuvem

Trabalho Prático

2023/2024

Informações Gerais

- Cada grupo é constituído por **5 elementos**. A inscrição é feita via a plataforma *blackboard*.
- O trabalho prático possui **3 momentos distintos de avaliação**. Dois *checkpoints* intermédios, com avaliação automática, e uma entrega final com avaliação presencial. Os mesmos são descritos na Secção **Avaliação**.
- Uma *FAQ* com respostas a questões e outra informação importante para a realização do trabalho, será mantida e atualizada na secção “Discussão” da organização ASCN-UM, acessível a todos os alunos.
- Os repositórios de cada grupo serão também alojados nesta organização.

Enunciado

O trabalho prático pretende exercitar os conhecimentos adquiridos na cadeira de forma a automatizar a instalação, configuração, monitorização e avaliação da aplicação **Laravel.io** (<https://github.com/laravelio/laravel.io>).

Tarefas

Tarefa 1 – Instalação e Configuração Automática da Aplicação

Objetivo: A primeira tarefa consiste em utilizar a ferramenta **Ansible** para automatizar a instalação e configuração da aplicação Laravel.io no serviço **Google Kubernetes Engine (GKE)** da Google Cloud.

Requisitos (*i.e.*, para além da utilização de Ansible e do serviço GKE):

- Os diferentes componentes da aplicação Laravel.io devem executar, quando possível, em *pods* distintos.
- A instalação e configuração da aplicação, bem como dos serviços da plataforma Google Cloud, devem ser efetuadas de forma automática no menor número de passos manuais possíveis.
- Deve ser possível parar a aplicação e voltar a executar a mesma (p.ex., devido a uma tarefa de manutenção programada) sem que sejam perdidos dados críticos (p.ex., dados de utilizadores).
- **É obrigatório** respeitar a estrutura e funcionalidades dos *playbooks* Ansible disponibilizados (ver Secção de **Requisitos de Desenvolvimento**).

Resultado: Após instalada e configurada, a aplicação Laravel.io deve ser acessível a partir do exterior através de um *browser*.

Tarefa 2 – Exploração e otimização da aplicação

Objetivo: A segunda tarefa tem como objetivo compreender e otimizar o desempenho, escalabilidade e resiliência da aplicação. É pretendido que o grupo responda às seguintes questões:

1. Considerando a instalação base proposta pelo grupo para a Tarefa 1:
 - a. Para um número crescente de clientes, que componentes da aplicação poderão constituir um gargalo de desempenho?
 - b. Qual o desempenho da aplicação perante diferentes números de clientes e cargas de trabalho?
 - c. Que componentes da aplicação poderão constituir um ponto único de falha?
2. Com base nas respostas dadas para as questões anteriores:
 - a. Que otimizações de distribuição/replicação de carga podem ser aplicadas à instalação base?
 - b. Qual o impacto das otimizações propostas no desempenho e/ou resiliência da aplicação?

Considerações:

- As questões anteriores devem ser discutidas no relatório final do trabalho prático, utilizando os conhecimentos adquiridos na cadeira.
- O grupo deve aplicar mecanismos de distribuição/replicação de carga que permitam otimizar a escalabilidade e/ou resiliência da instalação base.
Nota: Para reduzir a complexidade desta tarefa, o grupo deve apenas focar-se num **único componente** da aplicação (p.ex., servidor aplicacional, base de dados), justificando a escolha.
- É sugerido que o grupo explore a ferramenta de monitorização disponibilizada pela plataforma Google Cloud e considere diferentes métricas de monitorização (p.ex., CPU, RAM, I/O).
- Devem ser considerados testes experimentais para avaliar diferentes funcionalidades e/ou componentes da aplicação, tendo em conta a interação típica dos utilizadores com a mesma.

As soluções de otimização, avaliação e monitorização propostas nesta tarefa devem ser **incluídas nos playbooks Ansible** para poderem ser executadas e reproduzidas de forma automática.

Avaliação

O trabalho prático inclui 3 fases de avaliação, as quais são descritas em seguida:

Checkpoint #1 (2 valores): Este *checkpoint* pretende avaliar a correta especificação e funcionamento da imagem *Docker* para a camada aplicacional do **Laravel.io**. **Data limite:** 17/11/2023

Checkpoint #2 (5 valores): Este *checkpoint* pretende validar que, após instalada e configurada, a aplicação **Laravel.io** funciona corretamente e é acessível a partir do exterior. **Data limite:** 07/12/2023

Considerações:

- Os *checkpoints* serão avaliados automaticamente, através do **repositório GitHub** do grupo.
- Para estes checkpoints não é necessária a entrega de relatório.

Avaliação Final (13 valores): A avaliação final considerará todas as componentes do trabalho prático que não foram avaliadas de forma automática nos *checkpoints*. **Data limite:** 29/12/2023

Considerações:

- O trabalho prático é **concluído com a entrega de um relatório** via a plataforma *blackboard*.
- **Todos os scripts e ficheiros de configuração** usados na realização do trabalho têm de ser disponibilizados através do **repositório GitHub do grupo**.
- É **obrigatório** que o relatório identifique o número do grupo e todos os seus elementos na capa.
- O prazo limite para a **entrega do trabalho é 29 de dezembro de 2023**. Após esta data **não serão considerados** relatórios ou novos *commits* no repositório GitHub.
- A apresentação e discussão do trabalho decorre na semana entre 8 e 12 de Janeiro de 2024.

Requisitos do Relatório

O relatório apresentado deve conter a seguinte informação. De notar que a ordem pela qual cada um dos pontos em baixo é apresentada no relatório fica ao critério do grupo.

- Uma descrição breve da arquitetura e principais componentes da aplicação **Laravel.io**.
- Identificação das ferramentas e abordagem utilizadas para a instalação e configuração automática da aplicação.
- Uma discussão crítica sobre as questões colocadas na Tarefa 2.
- Ferramentas de monitorização, métricas e visualizações escolhidas, justificando a sua escolha.
- Ferramentas de avaliação e testes desenvolvidos, justificando a sua escolha.
- Apresentação e análise dos resultados da avaliação experimental.
- Reflexão final sobre o trabalho apontando os principais pontos fortes e os pontos a melhorar.

O relatório não necessita de incluir os scripts de aprovisionamento (estes são disponibilizados no repositório GitHub) e deve focar-se nas principais decisões tomadas para resolver os pontos especificados neste enunciado. Embora não exista um limite máximo de páginas, um relatório de 10 páginas deverá ser suficiente para responder ao enunciado.

Mais ainda, **todos os elementos do grupo** devem estar preparados para fazer uma breve apresentação do relatório que contempla uma demonstração de 5 minutos. Esta demonstração deverá ser realizada utilizando a Google Cloud e, por uma questão de otimização de tempo, cada grupo deve garantir que o cluster de *Kubernetes* GKE é criado e encontra-se utilizável antes da apresentação.

Requisitos de Desenvolvimento

Todos os scripts e ficheiros de configuração usados na realização do trabalho têm de ser disponibilizados através de um **repositório GitHub** na organização <https://github.com/ASCN-UM>.

De modo a obter acesso à organização, cada aluno deverá preencher, até ao dia **6 de Outubro**, o formulário *GitHub Information* disponibilizado na plataforma *blackboard*, indicando o email associado com a sua conta GitHub. Na secção *Conteúdo* da *blackboard* serão disponibilizados slides para auxiliar no uso da organização.

Uma **FAQ** será mantida e atualizada na secção “Discussão” da organização ASCN-UM, acessível a todos os alunos. Esta FAQ irá conter informação sobre:

- Detalhes sobre a avaliação para cada *checkpoint*.
- Estrutura do código base que cada grupo deve utilizar para procedermos à sua avaliação.
- Acesso à plataforma Google Cloud.

Finalmente, a equipa docente irá disponibilizar um conjunto de *playbooks* Ansible base que os grupos devem utilizar. A disponibilização dos mesmos será feita posteriormente e notificada via a plataforma *blackboard*.