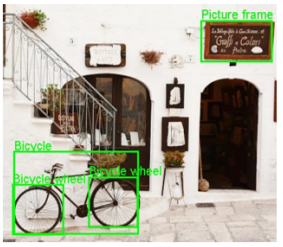
Trabalho final

Data limite para entrega do trabalho: **9 de junho 2022**
Para além do desenvolvimento que cada grupo realizará autonomamente fora das aulas, haverá aulas específicas, predominantemente nas aulas de 3 horas, para apoio à realização do trabalho.
Componentes a entregar:
1) Ficheiro ZIP com as componentes desenvolvidas, incluindo ficheiros README com informações sobre configurações, pressupostos de execução, teste ou outros.
2) Documento em formato PDF com descrição da solução: Diagramas de arquitetura, contratos e pressupostos entre as partes envolvidas, formatos de dados e mensagens envolvidos nas interações, bem como os aspectos relevantes da implementação e eventuais pontos de falha.

Objetivos: Saber planear e realizar um sistema para submissão e execução de tarefas de computação na nuvem, com requisitos de elasticidade, utilizando de forma integrada serviços da Google Cloud Platform para armazenamento, comunicação e computação, nomeadamente, Cloud Storage, Firestore, Pub/Sub, Compute Engine e Cloud Functions e Vision API.

1. Introdução

Desenvolva um sistema, designado *CN2122TF*, com o objetivo de detetar múltiplos objectos (por exemplo, bicicletas, cadeiras, quadros) em ficheiros de imagem (JPG, PNG, etc.) e gerar novas imagens com as zonas onde estão os objetos detetados. Associado à deteção de um objeto haverá um valor (entre 0 e 1) que corresponde ao grau de certeza do sistema sobre o nome do objeto.
O sistema deve ter elasticidade, aumentando ou diminuindo a sua capacidade de processamento de imagens.

As funcionalidades do sistema estão disponíveis para as aplicações cliente através de uma interface gRPC com as seguintes operações:

- Submissão de um ficheiro imagem para deteção de objetos. Esta operação recebe o conteúdo de um ficheiro imagem em *stream* de blocos, guardando o mesmo como um *blob* no serviço Cloud Storage. No final, a operação retorna um identificador do pedido (por exemplo, uma composição única entre o nome do *bucket* e do *blob*) que será usado posteriormente para obter o resultado da submissão.

- A partir de um identificador retornado na chamada à operação anterior deve ser possível obter:
  - a lista de nomes dos objetos encontrados na imagem;
  - a imagem original anotada com as zonas onde foram detectados objetos;

- Obter todos os nomes de ficheiros armazenados no sistema entre duas datas, que contêm um objeto com determinado nome e com um grau de certeza na deteção acima de *t* (por exemplo, imagens com quadros, com um grau de certeza acima de 0.6);

Todas as operações de submissão e posterior consulta são disponibilizadas através de um servidor gRPC, o qual funciona como a fachada do sistema, isto é, a aplicação cliente não usa serviços da plataforma GCP. Para aumentar a disponibilidade e balanceamento de carga do sistema devem existir várias réplicas do servidor gRPC, cada uma a executar-se numa VM de um *instance group*. Para obter (*lookup*) o endereço IP do servidor gRPC, a que se vai conectar, o cliente acede por HTTP a um URL pré-definido para obter a lista de endereços IP dos servidores gRPC, e escolhe um IP aleatoriamente. A arquitetura do sistema *CN2122TF* usa os seguintes serviços GCP:

- O serviço Cloud Storage armazena as imagens a processar e as imagens anotadas;
- O serviço Firestore guarda a informação relevante sobre o processamento de uma imagem, nomeadamente o identificador do pedido, data do processamento, metadados do *blob* no storage, os objetos detetados nas imagens, e outros que achar convenientes;
- O serviço Pub/Sub é usado para troca desacoplada de mensagens entre os componentes do sistema;
- O serviço Compute Engine é usado para alojar as máquinas virtuais (*instance group*) onde se executam as réplicas do servidor gRPC e as máquinas virtuais onde se executam as réplicas da aplicação (*Detect Objects App*) de deteção de objetos nas imagens;
- O serviço Vision API para detetar objetos nas imagens.

As diferentes interações entre os componentes do sistema são apresentadas na Figura 1.

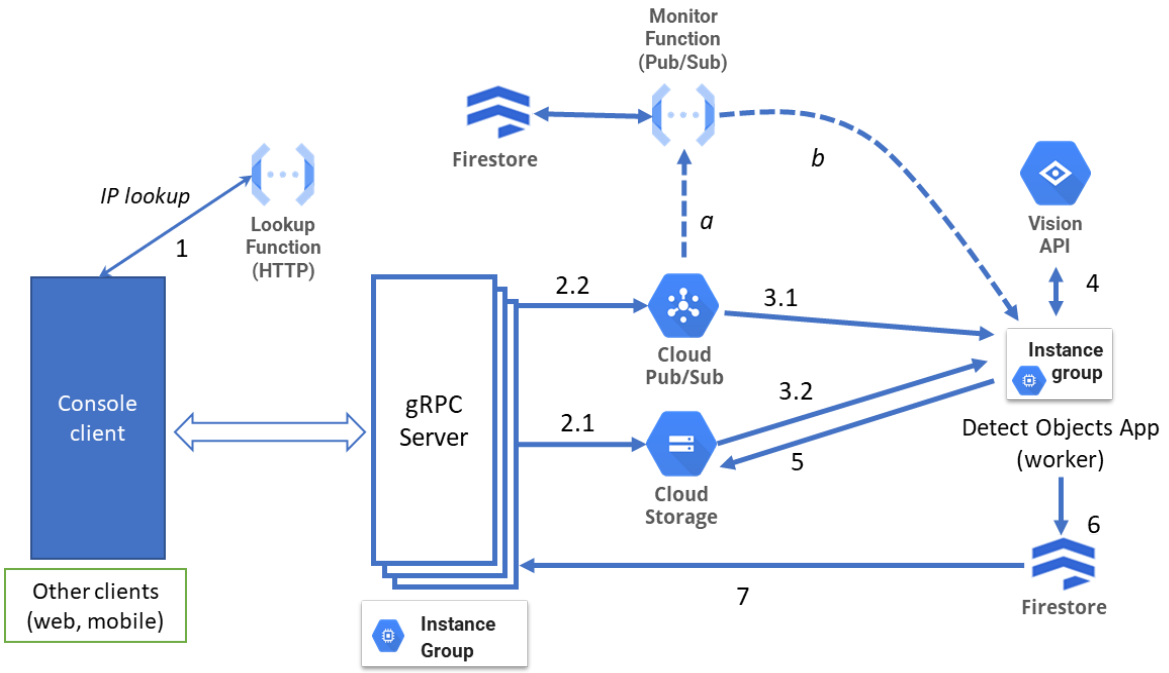


Figura 1: Componentes do *CN2122TF* e respetivas interações

2. Disponibilidade e elasticidade

Para que o sistema tenha maior disponibilidade e garantias de elasticidade devem ser considerados os seguintes requisitos:

- O serviço *Lookup Function,* usado pela aplicação cliente (1) para obtenção dos endereços IP dos servidores gRPC, deve ser desenvolvido como uma *Cloud Function*. O cliente escolhe um IP aleatoriamente e, em caso de falha de ligação ao servidor gRPC através do IP escolhido, tenta outro IP ou repete o processo de *lookup* para atualizar a lista dos IP e estabelecer uma nova ligação;

- Deve existir um *instance group* com réplicas do servidor gRPC cuja dimensão pode aumentar (máximo 3) ou diminuir (mínimo 1) manualmente através da consola web GCP;

- Deve existir um instance group com réplicas da aplicação *Detect Objects App* (*worker*) para deteção de características. O número de VM deste *instance group* é variável, em função do número de imagens submetidas a cada intervalo de monitorização. Consideremos como referência 3 pedidos por minuto, ou seja $ref = \frac{3\ pedidos}{60\ segundos} = 0,05$. Sendo k o número atual de VM em execução e o período de monitorização de 60 segundos:
  - se o número de pedidos por segundo for maior que x, o número de VM passa a $k + 1$;
  - se o número de pedidos por segundo for inferior a y, o número de VM passa a $k - 1$;
  - deve ser salvaguardado que no mínimo há 1 VM em execução e no máximo 4;
  - para efeitos de teste use os valores: $k = 1$; $x = ref + 0,02$; $y = ref - 0,02$, ou seja, o valor de referência de 3 pedidos por minuto $(0,05)$ com mais ou menos um *threshold* de $0,02$.

3. Fluxo de operações

Tendo em conta os números de sequência de ações, apresentados na Figura 1, a lista seguinte descreve cada uma das funcionalidades:

- Após a submissão de uma imagem, a mesma é guardada no Cloud Storage (2.1) e é retornado ao cliente gRPC um identificador único para posteriormente ser possível realizar as interrogações. De seguida, é enviado para um tópico Pub/Sub com nome *detectionworkers* o identificador do pedido, o nome do *bucket* e do *blob* para processamento de deteção de objetos (2.2);

- Associado ao tópico *detectionworkers* existem 2 subscrições: uma subscrição é consumida por vários *workers* (*work-queue pattern*) (3.1), a outra está ligada à função de monitorização (descrita abaixo). Um *worker* de análise de imagem recebe o nome da imagem a processar (3.1) e obtém o seu conteúdo do Cloud Storage (3.2), interagindo depois com o serviço Vision API (4);

- Após o processamento, a imagem anotada com os objetos detectados é escrita no Cloud Storage (5), sendo também guardado no Firestore (6) a informação relevante do pedido e do resultado da análise;

- Com base no descritor do pedido, a qualquer momento, as aplicações cliente podem pedir ao servidor gRPC informações sobre os ficheiros submetidos, tal como descrito na Secção 1. Para retornar essa informação o servidor gRPC consulta a base de dados Firestore (7).

- A *Monitor Function* representada na Figura 1 representa uma Cloud Function que contabiliza as mensagens enviadas para o tópico *detectionworkers*, sem afetar o processamento das mesmas pelos *workers*. Esta função implementa o algoritmo referido na Secção 2 (a) atuando sobre o número de VM do instance group *workers* (b).

4. Aspetos de implementação:

- A API de visão faz deteção de objetos em imagens, retornando, entre outras informações, uma string (nome do objeto), um indicador do grau de certeza da deteção (entre 0 e 1) e as coordenadas da zona retangular onde o objeto foi localizado na imagem. (https://cloud.google.com/vision/docs/object-localizer)

- Será posteriormente apresentado na aula um exemplo de código que, com base numa imagem presente no Cloud Storage, usa a API de visão para detetar objetos na imagem, criando no Cloud Storage uma nova imagem com as anotações detetadas.

5. Critérios de avaliação do trabalho:

- 30% - Qualidade do relatório, que permita a um leitor entender claramente a arquitetura e as decisões de interação entre as partes, evitando apresentar código, exceto se o mesmo ajudar a explicar detalhes relevantes. O relatório deve indicar os pressupostos assumidos, indicando eventuais comparações com outras decisões possíveis. Deve constar no relatório qual a(s) parte(s) onde cada elemento do grupo de alunos de CN teve mais ou menos responsabilidade.

- 60% - Operacionalidade, simplicidade e flexibilidade das soluções, nomeadamente na configuração e utilização da solução;
  - Nesta avaliação será ponderado o resultado da apresentação da funcionalidade da solução a toda a turma nas aulas da última semana de aulas. Para tal, será posteriormente estabelecido para cada grupo um calendário de apresentação, bem como um guião dos aspetos principais a demonstrar.

- 10% - Participação individual de cada elemento do grupo durante as aulas afetas à realização do trabalho, bem como na apresentação do trabalho à turma.

José Simão, Luís Assunção