

## Laboratório 3

**Objetivo:** Ganhar prática e consolidar os conceitos fundamentais do serviço Cloud Storage. Utilização da API do Cloud Storage e revisão do middleware gRPC.

- 1) Crie 2 *buckets* regionais usando a interface web do GCP: 1 na Europa (zona belgica) e 1 na Ásia (zona australia).
- 2) Pretende-se verificar o efeito da *cache* nos Blob públicos. Nesta alínea é usada a interface web e a ferramenta *gsutil* já existente na *cloud shell*.

a) Aceda à Google *Cloud Shell*.

b) Na *Cloud Shell* faça *download* da seguinte imagem do site nasa.org:

```
wget https://eoimages.gsfc.nasa.gov/images/imagerecords/144000/144875/BlackMarble_2012_01deg_geo.tif
```

c) Renomeie o ficheiro para *world.tif*.

```
ren BlackMarble_2012_01deg_geo.tif world.tif
```

d) Faça upload do ficheiro para o *bucket* da Ásia usando a ferramenta *gsutil* (<https://cloud.google.com/storage/docs/gsutil/commands/cp/>).

```
gsutil cp world.tif gs://...
```

e) Usando a interface web altere o blob para ter acesso público e sem *cache* (Metadata com: *Cache-control = no-cache*)

f) Realize 10 acessos ao *blob* e registre o tempo médio usando o seguinte comando:

```
for n in {1..10}; do time curl -o world2.tif
```

```
https://storage.googleapis.com/cn-lab3-asia/world.tif; done
```

g) Ative a *cache* do *blob* (Metadata com: *Cache-control = public*).

h) Realize novamente os 10 acessos ao *blob* e registre o tempo médio.

- 3) Para instalação local em computadores pessoais estão disponíveis as ferramentas de linha de comandos do Google Cloud SDK (*gcloud* e *gsutil*). Instale e configure o *Google Cloud SDK* no seu computador, seguindo as instruções seguintes:

Windows (<https://cloud.google.com/sdk/docs/quickstart-windows>)

MacOS (<https://cloud.google.com/sdk/docs/quickstart-macos>)

Linux (<https://cloud.google.com/sdk/docs/quickstart-debian-ubuntu>).

Um dos primeiros passos de configuração é dar consentimento para acesso aos recursos existentes na sua conta Google GCP e no projeto que já criou anteriormente. Este passo é feito através do protocolo de autorização OAuth 2.0. A autorização de acesso aos recursos é feita no *browser*, após se autenticar na conta Google usada na criação do projeto.

- 4) Execute o comando `gsutil perfdiag <bucket>` (<https://cloud.google.com/storage/docs/gsutil/commands/perfdiag>), na *cloud shell* da GCP e no seu computador, usando como teste o bucket na Europa e na Ásia. Observe e analise os resultados do *throughput* de escrita e leitura.  
A opção `-o <file>` guarda os resultados no ficheiro indicado.
- 5) Considere o projeto *maven CNStorageBase*, disponível em anexo ao enunciado, que possibilita algumas operações básicas de acesso ao serviço *Storage*: listagem de *buckets* de um projeto, criação e remoção de *buckets* e *upload/download* de *blobs*.
  - a) Execute o programa configurando a variável de ambiente para o ficheiro com a chave de *service account*  
  
GOOGLE\_APPLICATION\_CREDENTIALS=<pathname do ficheiro json com chave>
  - b) Acrescente outras operações para consolidar conceitos, especialmente a operação para colocar um *blob* com permissões de acesso público.
- 6) Considere o seguinte contrato (*forum.proto*) de um serviço de fórum de subscrição e publicação de mensagens, existente no projeto *maven ClientForumFileShare*, em anexo ao enunciado.

```
service Forum {  
  // subscribe a topic  
  rpc TopicSubscribe(SubscribeUnSubscribe) returns (stream ForumMessage);  
  //unsubscribe a topic  
  rpc TopicUnSubscribe(SubscribeUnSubscribe) returns (google.protobuf.Empty);  
  // get all topics in server  
  rpc GetAllTopics(google.protobuf.Empty) returns (ExistingTopics);  
  // Send a message to a topic  
  rpc MessagePublish (ForumMessage) returns (google.protobuf.Empty);  
}  
  
message SubscribeUnSubscribe {  
  string usrName = 1;  
  string topicName = 2;  
}  
  
message ExistingTopics {  
  // list of topic names  
  repeated string topicName = 1;  
}  
  
message ForumMessage {  
  string fromUser = 1;  
  string topicName = 2;  
  string txtMsg = 3;  
}
```

Pretende-se realizar um **cliente, em gRPC, do serviço Forum** que possibilite a partilha de links para ficheiros (*blobs*) alojados no serviço *Google Cloud Storage*.

As mensagens enviadas para o *Forum* podem representar apenas texto, ou um texto com o nome de um *bucket* e de um *blob* com acesso público, onde a aplicação cliente escreveu o ficheiro que pretende partilhar. As mensagens que o cliente envia para um tópico têm o formato `<texto>;<bucket>;<blob>` onde:

- i) `<texto>` é obrigatório e representa a mensagem a enviar para o tópico
- ii) `;<bucket>;<blob>` é opcional e representa o nome do *bucket* onde é escrito o *blob*

O cliente quando recebe uma mensagem de um tópico que subscreveu, analisa-a segundo este formato e, caso esteja presente o nome de um *bucket* e de um *blob*, faz *download* do mesmo para um ficheiro local.

Tenha em conta que a aplicação cliente:

- Deve usar o serviço *Forum* disponível no endereço **35.214.60.17** porto **8000** em execução numa VM da *Google Cloud Platform*;
- Acede diretamente ao *Google Cloud Storage* e gere 1 *bucket* e vários *blobs* que devem ter acesso público para poderem ser acedidos por diferentes utilizadores do fórum;
  - a) Considere o projeto *maven* em anexo, o qual contém a definição do contrato no ficheiro `forum.proto`. Crie um projeto no IntelliJ a partir do projeto *maven* e utilize-o como base para desenvolver o código de cliente.
    - i) Comece por fazer *'package'* para gerar os *stubs*;
    - ii) Desenvolva o cliente com as funcionalidades anteriormente descritas.
  - b) **Desafio**: Considere que quem publica uma mensagem indicando um *blob* garante que o mesmo tem *metadata* com um tempo limite em que o mesmo fica público. Ao fim desse tempo o publicador retira o acesso público ao *blob*. Desta forma quem recebe uma mensagem com link para um *blob* tem de fazer *download* do mesmo enquanto ele está público.