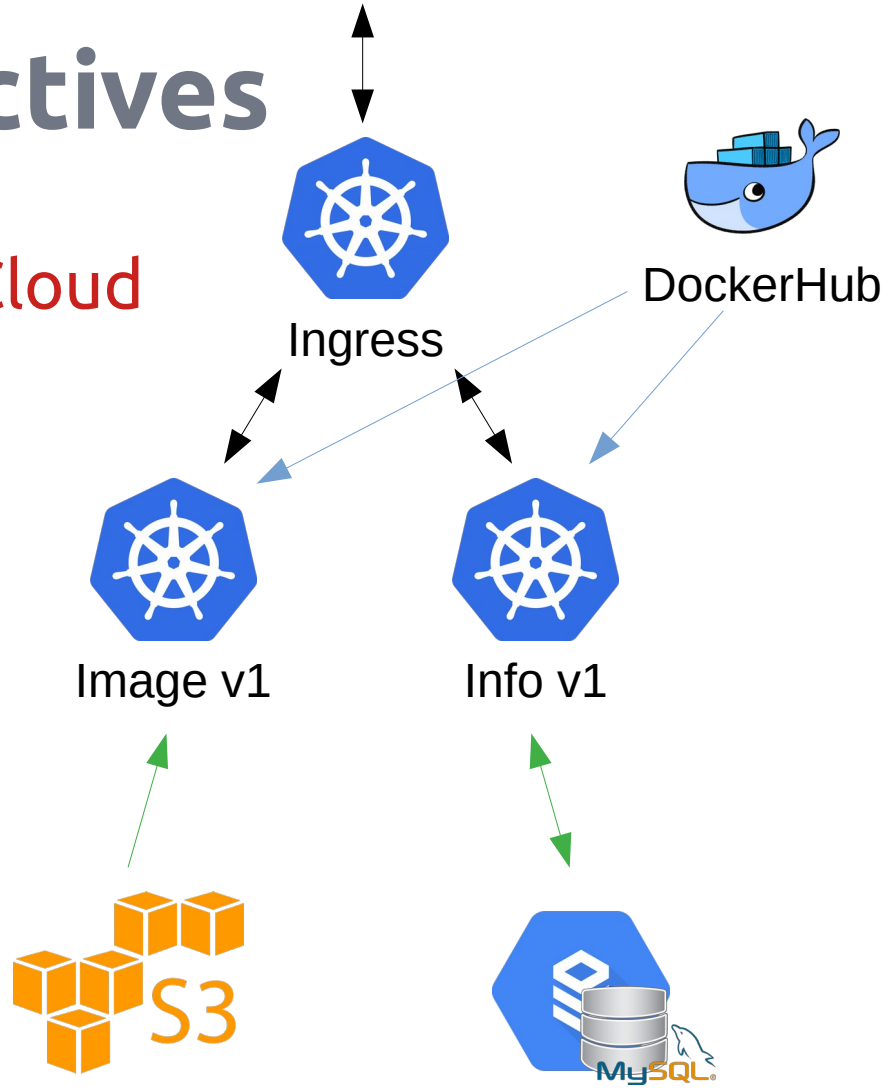
CLOUD COMPUTING PROJECT

WATCHES
WEBSERVICES

PART II

Part II - Objectives

- Deploy Watch Service in Google Cloud
- Leverage PaaS
  - GCloud Kubernetes
  - GCloud SQL
  - Docker Images repository
  - NoSQL DB (AWS S3)
- Automate the process
  - Code → Deployment



Microservice - Image info v1

- New Image (proxy) microservice
  - One endpoint /image/v1/get/{sku}
    - Will transfer an image from a NoSQL repo (AWS S3) to the user
      - Fixed mime type (image/png)
      - No image processing (ATM ;-)
    - Set proper caching headers and ETag
      - Lifetime 1h
  - OpenAPI description in git: `image_openapi_v1.yaml`

Watches Images

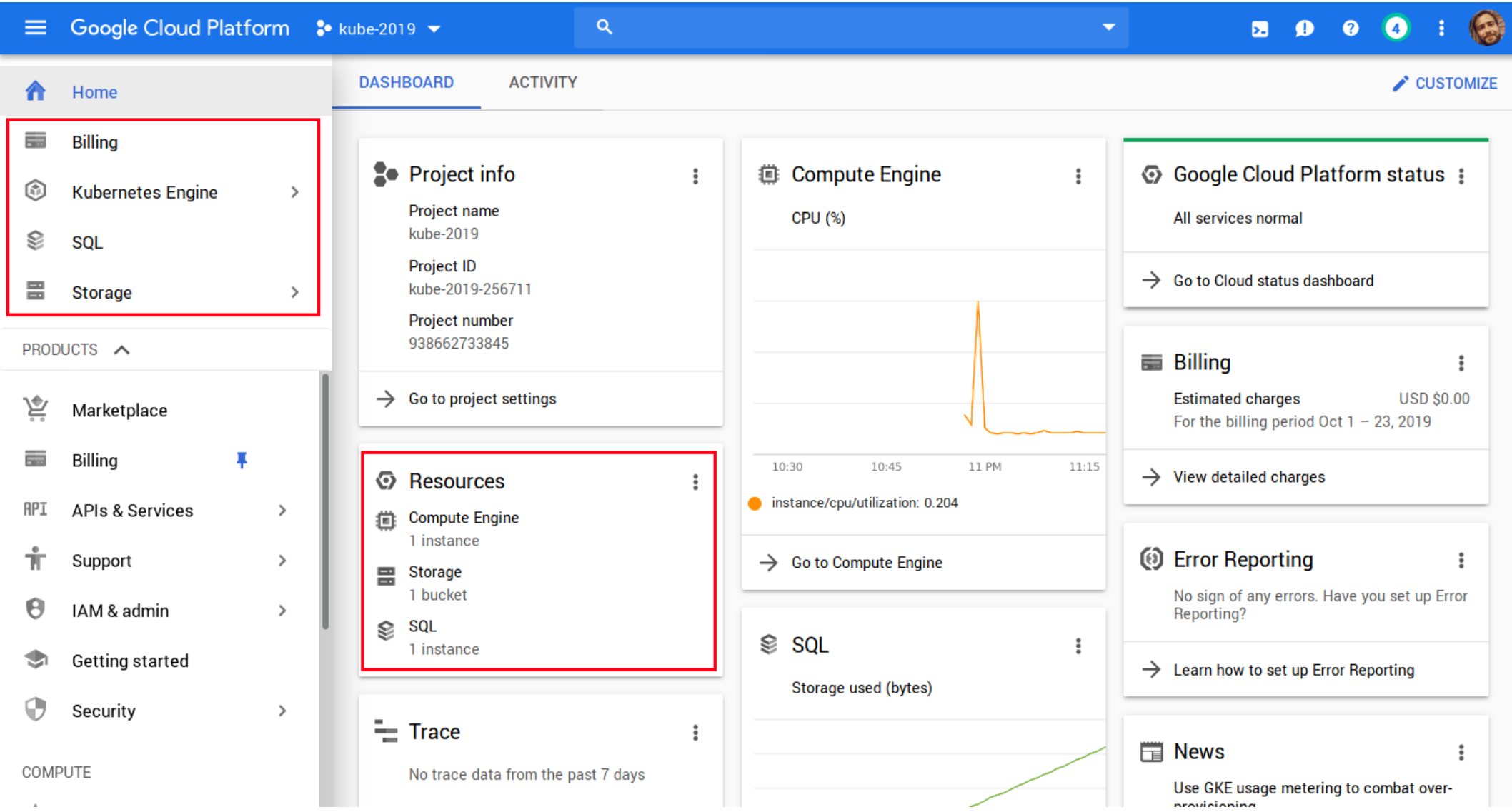
- The images are stored in a AWS S3 bucket (Ireland)
  - Publicly accessible
    - You can access them from GKE
  - Image format is transparent PNG (max 1024x1024)
    - `https://s3-eu-west-1.amazonaws.com/cloudcomputing-2018/project1/images/<SKU>.png`
      - `https://s3-eu-west-1.amazonaws.com/cloudcomputing-2018/project1/images/CAC1111.BA0850.png`
      - `https://s3-eu-west-1.amazonaws.com/cloudcomputing-2018/project1/images/CV201AP.FC6429.png`
  - Not all images will be available ! (~ 50% of SKUs have images)
    - Check for 200 or 404 HTTP codes

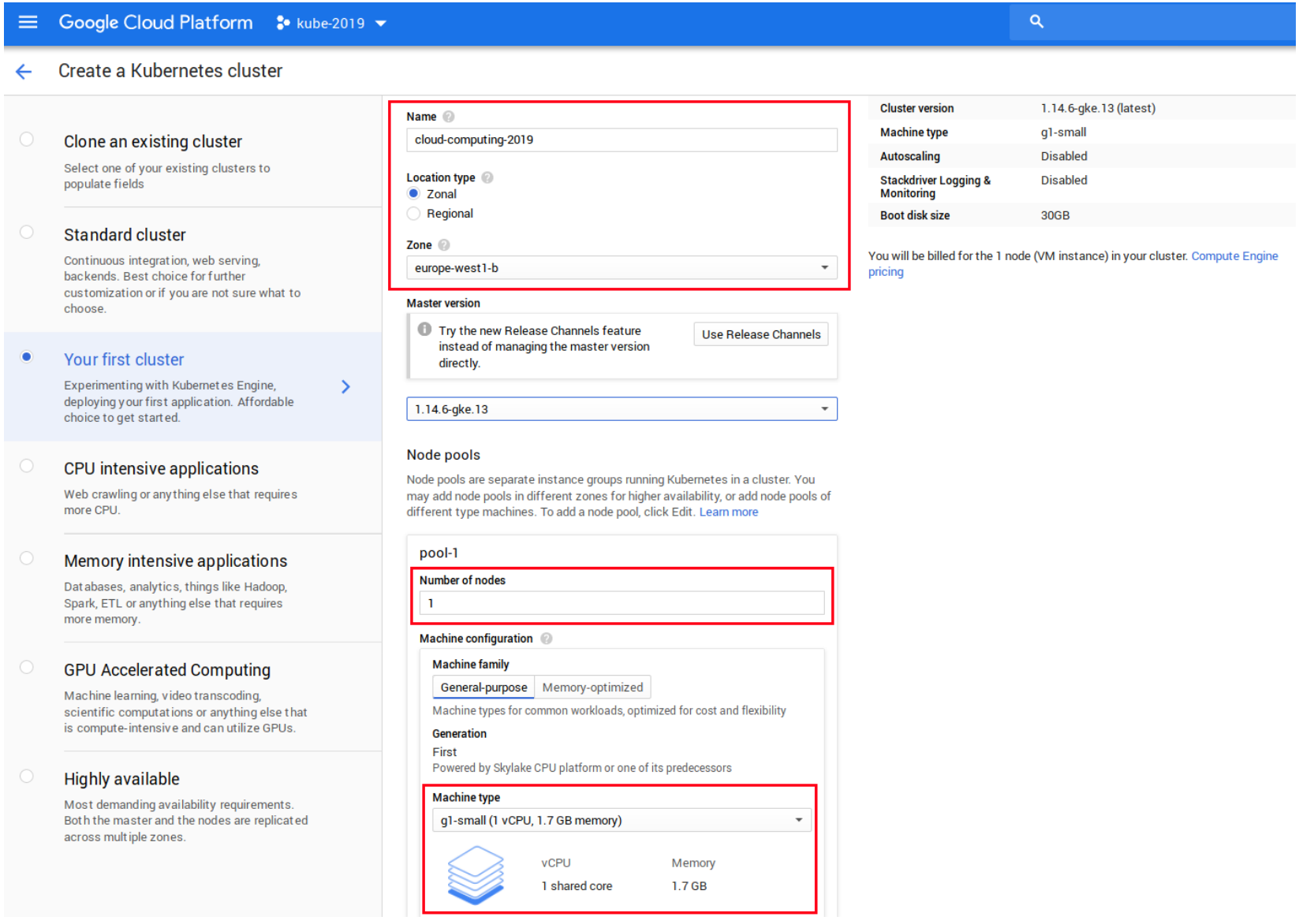
DockerHub

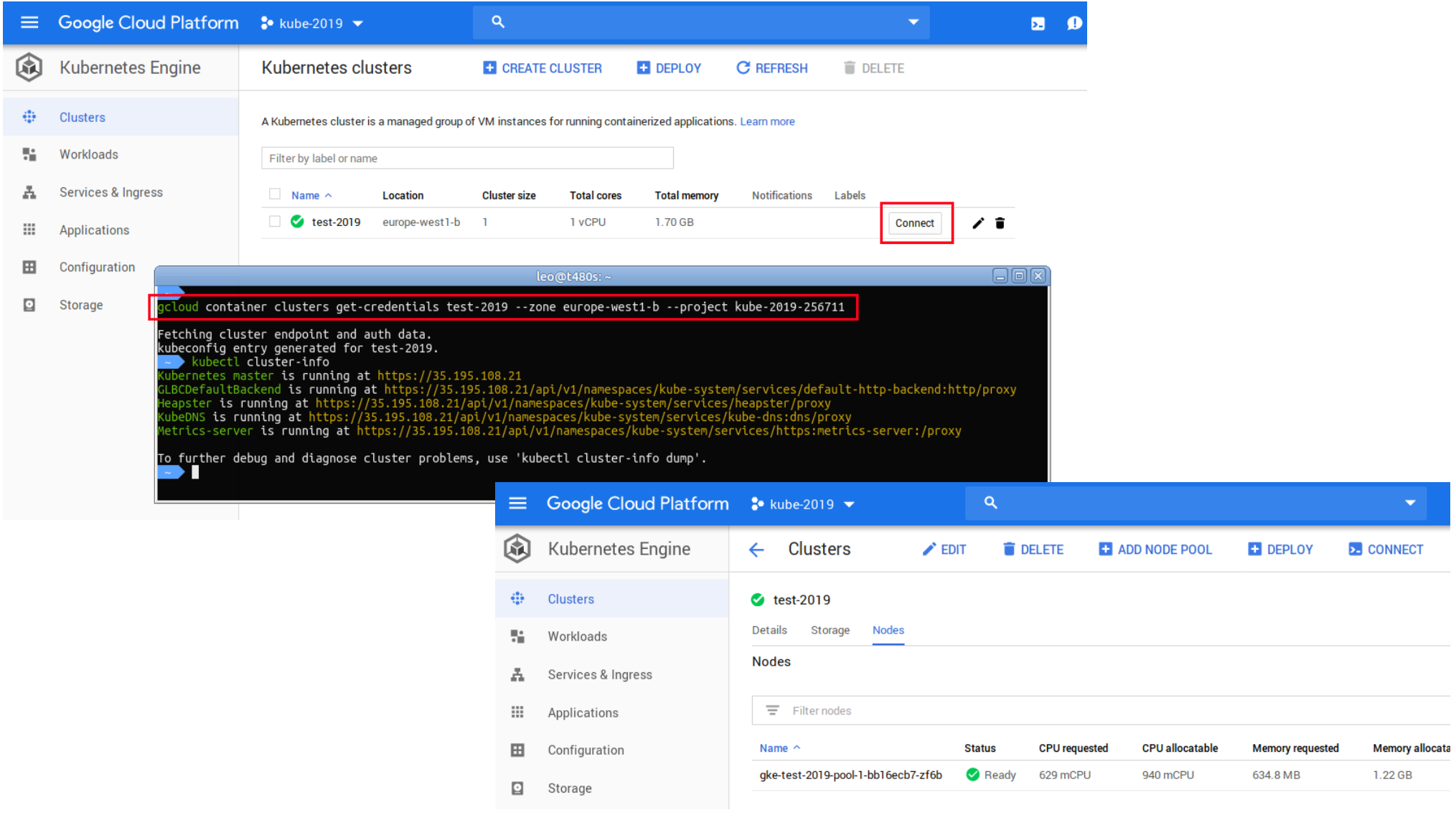
- Docker/Kubernetes/AWS can use any Docker images repositories
  - DockerHub is the default
    - If you publish your images (publicly) on DockerHub, you will not have to manage complex access to a particular repo (and no authentication to manage)
    - Create a free account
- Microservices images to publish
  - Infos v1
  - Images v1
- Publishing updated images should be part of your build process

Kubernetes - GKE

- Create a Google Cloud Platform account
  - https://cloud.google.com/compute/
  - 300$ / 1 year free credits
  - Create a Kubernetes cluster in Europe West (close to AWS S3 for images)
  - Link your account to kubectl
    - https://cloud.google.com/kubernetes-engine/docs/quickstart
    - $ gcloud container clusters get-credentials <cluster> --zone europe-west1-b --project <project>
    - Then, use kubectl as you will use it locally with minikube !
- Kubernetes also exists on AWS but is expensive (~200$ / month with mininal config) and much complicated to deploy
  - Kubernetes is a technology from Google and is better integrated in GCE


Google Cloud Platform
kube-2019
Home
Billing
Kubernetes Engine
SQL
Storage
PRODUCTS
Marketplace
Billing
APIs & Services
Support
IAM & admin
Getting started
Security
COMPUTE
DASHBOARD
ACTIVITY
CUSTOMIZE
Project info
Project name
kube-2019
Project ID
kube-2019-256711
Project number
938662733845
Go to project settings
Resources
Compute Engine
1 instance
Storage
1 bucket
SQL
1 instance
Trace
No trace data from the past 7 days
Compute Engine
CPU (%)
10:30
10:45
11 PM
11:15
instance/cpu/utilization: 0.204
Go to Compute Engine
SQL
Storage used (bytes)
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
Billing
Estimated charges
USD $0.00
For the billing period Oct 1 – 23, 2019
View detailed charges
Error Reporting
No sign of any errors. Have you set up Error Reporting?
Learn how to set up Error Reporting
News
Use GKE usage metering to combat over-


Google Cloud Platform
kube-2019
Create a Kubernetes cluster
Clone an existing cluster
Select one of your existing clusters to populate fields
Standard cluster
Continuous integration, web serving, backends. Best choice for further customization or if you are not sure what to choose.
Your first cluster
Experimenting with Kubernetes Engine, deploying your first application. Affordable choice to get started.
CPU intensive applications
Web crawling or anything else that requires more CPU.
Memory intensive applications
Databases, analytics, things like Hadoop, Spark, ETL or anything else that requires more memory.
GPU Accelerated Computing
Machine learning, video transcoding, scientific computations or anything else that is compute-intensive and can utilize GPUs.
Highly available
Most demanding availability requirements. Both the master and the nodes are replicated across multiple zones.
Name
cloud-computing-2019
Location type
Zonal
Regional
Zone
europe-west1-b
Master version
Try the new Release Channels feature instead of managing the master version directly.
Use Release Channels
1.14.6-gke.13
Node pools
Node pools are separate instance groups running Kubernetes in a cluster. You may add node pools in different zones for higher availability, or add node pools of different type machines. To add a node pool, click Edit. Learn more
pool-1
Number of nodes
1
Machine configuration
Machine family
General-purpose
Memory-optimized
Machine types for common workloads, optimized for cost and flexibility
Generation
First
Powered by Skylake CPU platform or one of its predecessors
Machine type
g1-small (1 vCPU, 1.7 GB memory)
vCPU
1 shared core
Memory
1.7 GB
Cluster version
1.14.6-gke.13 (latest)
Machine type
g1-small
Autoscaling
Disabled
Stackdriver Logging & Monitoring
Disabled
Boot disk size
30GB
You will be billed for the 1 node (VM instance) in your cluster. Compute Engine pricing


Google Cloud Platform
kube-2019
Kubernetes Engine
Kubernetes clusters
CREATE CLUSTER
DEPLOY
REFRESH
DELETE
Clusters
Workloads
Services & Ingress
Applications
Configuration
Storage
A Kubernetes cluster is a managed group of VM instances for running containerized applications. Learn more
Filter by label or name
Name
Location
Cluster size
Total cores
Total memory
Notifications
Labels
test-2019
europe-west1-b
1
1 vCPU
1.70 GB
Connect
leo@t480s: ~
gcloud container clusters get-credentials test-2019 --zone europe-west1-b --project kube-2019-256711
Fetching cluster endpoint and auth data.
kubeconfig entry generated for test-2019.
kubectl cluster-info
Kubernetes master is running at https://35.195.108.21
GLBCDefaultBackend is running at https://35.195.108.21/api/v1/namespaces/kube-system/services/default-http-backend:http/proxy
Heapster is running at https://35.195.108.21/api/v1/namespaces/kube-system/services/heapster/proxy
KubeDNS is running at https://35.195.108.21/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://35.195.108.21/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
Google Cloud Platform
kube-2019
Kubernetes Engine
Clusters
EDIT
DELETE
ADD NODE POOL
DEPLOY
CONNECT
Clusters
Workloads
Services & Ingress
Applications
Configuration
Storage
test-2019
Details
Storage
Nodes
Nodes
Filter nodes
Name
Status
CPU requested
CPU allocatable
Memory requested
Memory allocata
gke-test-2019-pool-1-bb16ecb7-zf6b
Ready
629 mCPU
940 mCPU
634.8 MB
1.22 GB

SQL Database

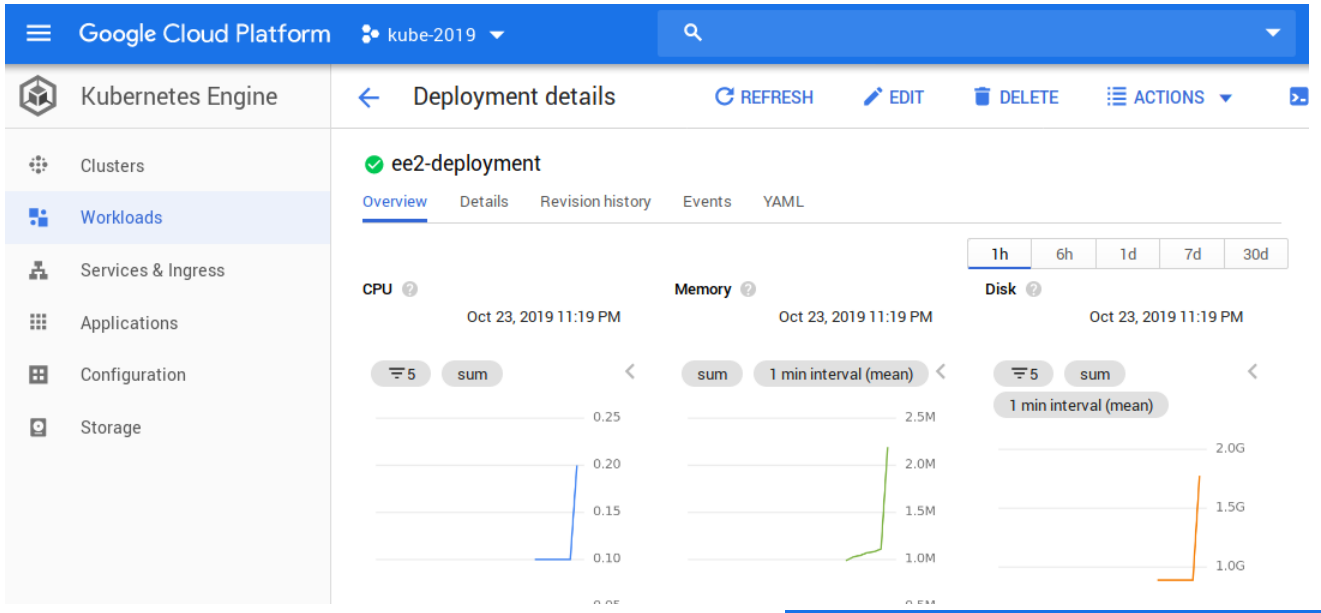
- The SQL database will be used by info services v1
  - Deploy the same database as in part I
- The product to use is Google Cloud SQL
  - Managed MySQL
    - See doc below to access it from Kubernetes pods
      - https://cloud.google.com/sql/docs/mysql/connect-kubernetes-engine

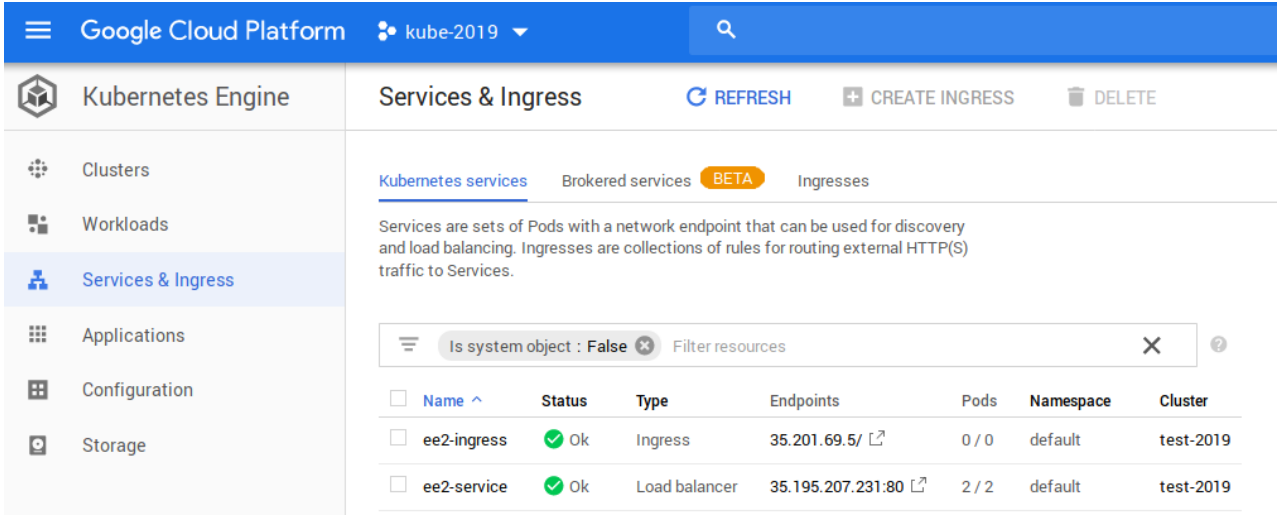
Kubernetes Architecture

- Objects to create (at the end all have to be described in a single yaml file)
  - **Deployments**
    - One deployment for each docker image (→ 2)
    - 2 replicas (pods)
  - **Services**
    - One service for each image deployment (→ 2)
    - Service will load balance between the pods of the corresponding deployment
  - **Ingress**
    - Will route to the right service depending on the path

Kubernetes - Ingress controller

- Use ingress controller to route to your microservices
  - /info/v1/* → Service info
  - /image/v1/* → Service image
- Use HTTP (HTTPS not mandatory)
- https://cloud.google.com/kubernetes-engine/docs/concepts/ingress
- https://cloud.google.com/kubernetes-engine/docs/tutorials/http-balancer
- Note: you should wait a few (10-15) minutes before the ingress controller is really deployed


Google Cloud Platform
kube-2019
Kubernetes Engine
Deployment details
REFRESH
EDIT
DELETE
ACTIONS
Clusters
Workloads
Services & Ingress
Applications
Configuration
Storage
ee2-deployment
Overview
Details
Revision history
Events
YAML
1h
6h
1d
7d
30d
CPU
Oct 23, 2019 11:19 PM
Memory
Oct 23, 2019 11:19 PM
Disk
Oct 23, 2019 11:19 PM
5
sum
sum
1 min interval (mean)
5
sum
1 min interval (mean)
0.25
0.20
0.15
0.10
2.5M
2.0M
1.5M
1.0M
2.0G
1.5G
1.0G


Google Cloud Platform
kube-2019
Kubernetes Engine
Services & Ingress
REFRESH
CREATE INGRESS
DELETE
Clusters
Workloads
Services & Ingress
Applications
Configuration
Storage
Kubernetes services
Brokered services
BETA
Ingresses
Services are sets of Pods with a network endpoint that can be used for discovery and load balancing. Ingresses are collections of rules for routing external HTTP(S) traffic to Services.
Is system object : False
Filter resources
Name
Status
Type
Endpoints
Pods
Namespace
Cluster
ee2-ingress
Ok
Ingress
35.201.69.5/
0 / 0
default
test-2019
ee2-service
Ok
Load balancer
35.195.207.231:80
2 / 2
default
test-2019

Deliverables

- `/info/`
  - Move here all files from project part I
- `/image/`
  - Write the new service (follow project part I guidelines):
    - Dockerfile, server.py, requirements.txt, run.sh
- `/README`
  - DNS:port to test your API !
  - Additional indications, specially if something is not working as expected
- `/build.sh`
  - Rebuild the 2 docker service images
- `/deploy.sh`
  - Publish/update the 2 docker images to DockerHub
  - 2 instances for each services (replicas: 2)
  - Kubernetes rolling upgrade
- `/all.yaml`
  - Deployments + Services + Ingress
  - Multiple object descriptions can be grouped in one file, use '\n---\n' as separator

Delay - Grading

- Delay: 4 weeks
  - Deadline: 2019-11-20T23:59:59+02:00
- Grading
  - Participation
    - 1 point
  - Endpoint with working APIs for Info & Image
    - 3 points
  - Scripts to automate build & rolling upgrade
    - 2 points

Documentation

- Minikube
  - https://github.com/kubernetes/minikube
  - https://kubernetes.io/docs/tasks/tools/install-minikube/
  - https://kubernetes.io/docs/setup/learning-environment/minikube/
- Kubernetes Cheat Sheet
  - https://linuxacademy.com/blog/containers/kubernetes-cheat-sheet/