## Cloud Computing Report Assignment 3



|  |  |
| --- | --- |
| **Name:** | Nikodijevic Mihajlo  Stefkovic Jano  Palic Kristjan |
| **Student number:** | 01646292  11743161  11905873 |
| **E-Mail Address:** | [a01646292@unet.univie.ac.at](mailto:a01646292@unet.univie.ac.at)  [a11743161@unet.univie.ac.at](mailto:a11743161@unet.univie.ac.at)  [a11905873@unet.univie.ac.at](mailto:a11905873@unet.univie.ac.at) |
| **Datum:** | 26.12.2019 |

### Google Cloud Platform Pricing Calculator

At the first glance, I got very confused and this calculator used to be very complicated

**Google Compute Engine**

With all specifications I came up with following pricing for my app with GCE  
  
1 instance (n1-standard-1, regular, Frankfurt, 1 year commitment) = 28.12$ / Month  
  
+ 1 Local SSD of 375GB = 50.99$ / Month  
+ Persistent Disk of 512GB Regional Standard in Frankfurt (49.15$) = 100.15$/Month  
+ Load Balancing (1 Rule, 10GB) 22.00$ = 122.15$  
  
To recap: This is a costs for 1 average instance = 122.15$  
We would need 5 instances, what gives me 326.12$ / Month  
  
If I understood this correctly, this GCE is exactly the same as traditional instance renting, with additional functionalities, and that’s why it’s so expensive.

### Google Kubernetes Engine

With all specifications I came up with following pricing for my app with GKE:  
  
5 nodes (standard,375GB SSD, Frankfurt, 1 year commitment) = 254.97$ / Month  
+ Persistent Disk of 512GB Regional Standard in Frankfurt (49.15$) = 271.54$  
+ Load Balancing 1 Rule, 10GB (22.00$) = 301.54$ / Month  
  
Conclusion part 1  
It is not clear to me what are we comparing here between GCE and GKE, because it seems to be very similar(pricing seems to be very strange in my opinion).  
However, I got the architecture difference, what should be the main point in the lecture and think that Kubernetes would be better solution for our SaaS from Assignment 1, because it stays with cloud computing principles and containerization.

Kubernetes – Development and Deployment part 2

1. Starting with Kubernetes on my local machine (Windows 10) was so painful that I almost gave on this assignment. Then, I found one good article which helped me configure my minikube cluster properly with Hyper-V.  
  
2. After that painful experience I started creating deployment files for my services.  
I found “**kompose**” tool(<https://github.com/kubernetes/kompose>) which generates our files automatically if we had correctly deployment using docker-compose from the Assignment 1, which I luckily had.  
  
3. Another thing I did before trying to deploy was publishing my services to docker hub, because it was suggested in forum as easier solution. That’s why I published also my code on Github to be public, to be easier to connect with docker hub.  
  
4. Kompose tool has generated all the necessary files for me and I run into more trouble after that.

Things that I needed to adapt there were:

1. apiVersion – I changed it to apps/v1

2. Adding label and selector part – This part was not provided automatically☹  
3. Images names were changed to use images from docker hub

5. Fixing communication between microservices on local machine. As mentioned by Mr. Jan in forum, standard communication as in docker doesn’t work in Kubernetes using http://<SERVICE\_NAME>:<SERVICE\_PORT>/ endpoint. I found this way for access: http:// <SERVICE\_NAME>.default.svc.cluster.local:<SERVICE\_PORT>/   
Update: this way is not necessary, docker like way works too on GKE. 😊

Kubernetes Setup

Cluster

default-master version

default-pool – 4 nodes – 15GB – 4vCPUs

general-purpose  
n1-standard-1  
  
When I uploaded my deployment yaml files to GKE and ran deployment command, not all microservices could start. Then, I found out that I was running everything on single node and that only 4/8 are running. Then, I extended my cluster to 4 nodes and it ran automatically all 8😊  
  
The thing that I didn’t found out easily is how to start my camera agents to stream on GKE, because I couldn’t find endpoint to which my cluster is exposed. ¯\\_(ツ)\_/¯  
Than I started over cli my cameras and everything worked well as in assignment 1 with docker and locally on minikube cluster.

Scaling:  
All services that I ran on GKE are single, so no replication implemented in the files that I uploaded. But I experimented with replication and load balancer locally via minikube. Collector service is very convenient for scaling because it’s stateless and in the same time very big bottleneck in our surv (kubectl scale –replicas=4 deployment/collector)  
Load balancer exposing:  
kubectl expose deployment collector –type=LoadBalancer –name collector-lb …

----------------------------------------------------------------------------------------------------------------

Conclusion

The Assignment 2 was not my favorite part of IT, because it digs more in DevOps and low-level things which I don’t like. I also lost a lot more time on this task 2 comparing to task 1, even if no programming was needed to accomplish this task.  
Kubernetes is however very important for todays development, and I am very thankful that we learned a lot about it.

In my opinion, we should have some tutorium for basics to get started in such advanced things, because diving into this huge ocean like kubernetes alone at home is very painful. Some of my colleagues gave up on this assignment. One more time, thanks for the good help via forum. That helped a lot again. 😊