A Scalable Microservices-based Web Application in a Public Cloud

Capstone Project Report - Group A43

Management and Administration of IT Infrastructures and Services

Team nr.:	A43
81525:	Carlos Antunes
89451:	Guilherme Areias
90531:	Pedro Rodrigues

Information Systems and Computer Engineering
IST-ALAMEDA

2021/2022

Contents

List of Figures

Acronyms

IP	Internet Protocol
AGI	Management and Administration of IT Infrastructures and Services
GCP	Google Cloud Platform
K8S	Kubernetes Engine
VM	Virtual machine

Chapter 1

Introduction

The following project from Management and Administration of IT Infrastructures and Services Management and Administration of IT Infrastructures and Services (AGI), consists on the implementation, deployment and provision of a microservice based web application on a public cloud provider. The Browser-based Calculator as a Microservice Architecture was used as the base of the Project in conjunction with the Google Public Cloud Platform Google Cloud Platform (GCP) and the Kubernetes Engine Kubernetes Engine (K8S).

1.1 Video

Link of the video of the Project-Group-A43: https://www.youtube.com/watch?v=0maBBIlJq0I

Chapter 2

Implementation

The goal of this project is the deployment and provision of a microservice based web application on a public cloud provider. To accomplish this the project infrastructures are built using **Terraform**, with K8S and **Docker** behind the scenes as Terraform's providers, which are hosted on GCP. The monitoring services are responsibility of the **Grafana** and **Prometheus** software.

2.1 Implementation options

The Web Microservices-based Application has high availability with multiple replicas of each microservice, a Balancing system for the frontend and a DataStore backend.

2.2 Pre-requisites

Here are the Pre-requisites for the System:

PR01 Create or have a GCP account with billing plan enabled;

PR02 Have installed **Oracle VirtualBox** software on a working desktop;

PR03 Have at least 8 GB of free disk storage.

2.3 Instructions

2.3.1 Start and getting the mgmt up and running

I-1.01 Extract the Project by ’git clone https://git.rnl.tecnico.ulisboa.pt/AGISIT-21-22/team-43A/src/branch/main;

I-1.02 Open the terminal or cmd and change directory to ’/labs/project/’;

I-1.03 Run the command ‘vagrant up’ (this operation might take a while);

```
user@user-machine:~/project$ vagrant up
```

I-1.04 Run the command ‘vagrant global-status’ and check if the ‘mgmt-project’ state is ‘running’;

```
user@user-machine:~/project$ vagrant global-status
id          name           provider     state     directory
....
3d7f9f5     mgmt-project   virtualbox   running   ~/project
```

I-1.05 Establish a ssh connection to the ‘mgmt-project’, by running the command ‘vagrant ssh mgmt-project’;

```
user@user-machine:~/project$ vagrant ssh mgmt-project
Welcome to Ubuntu 20.04.3 LTS ....
....
vagrant@mgmt-project:~$
```

I-1.06 To give administrator permissions to docker, run the command ‘sudo usermod -aG docker ${USER}’. It is required to exit the Virtual machine (VM) and log-in again, so run ‘exit’ and repeat the command ‘vagrant ssh mgmt-project’.

```
vagrant@mgmt-project:~$ sudo usermod -aG docker ${USER}
vagrant@mgmt-project:~$ exit
user@user-machine:~/project$ vagrant ssh mgmt-project
....
```

```
vagrant@mgmt-project:~$
```

2.3.2 Google Cloud Platform

I-2.01 Create a new project name under GCP with the name **AGISIT-Project-A43**;

I-2.02 Go to **API and Services** and select **Dashboard**. Click on **Enable API and Services**. Search for **Kubernetes Engine API** and enable that API service;

I-2.03 Go to **IAM and Admin** and select **Service Accounts**. Click on the project **AGISIT-Project-A43**. In **Actions** select **Manage keys**. **Add** and **create new key**, and save the credentials on a **.json file**;

I-2.04 Copy the **.json** file to the project directory '/infrastructure'.

I-2.05 Go to **IAM and Admin** and select **IAM**. **Add Another Role** and on **Select a role** search for **Kubernetes Engine Admin**, and save;

2.3.3 Terraform

I-3.01 Run the command 'gcloud auth login' and login to the GCP account;

```
vagrant@mgmt-project:~$ gcloud auth login
```

I-3.02 On the VM 'mgmt-project' go inside the directory '/infrastructure' and create a new file with the extension **.tfvars**, as shown on (Figure 2.1).;

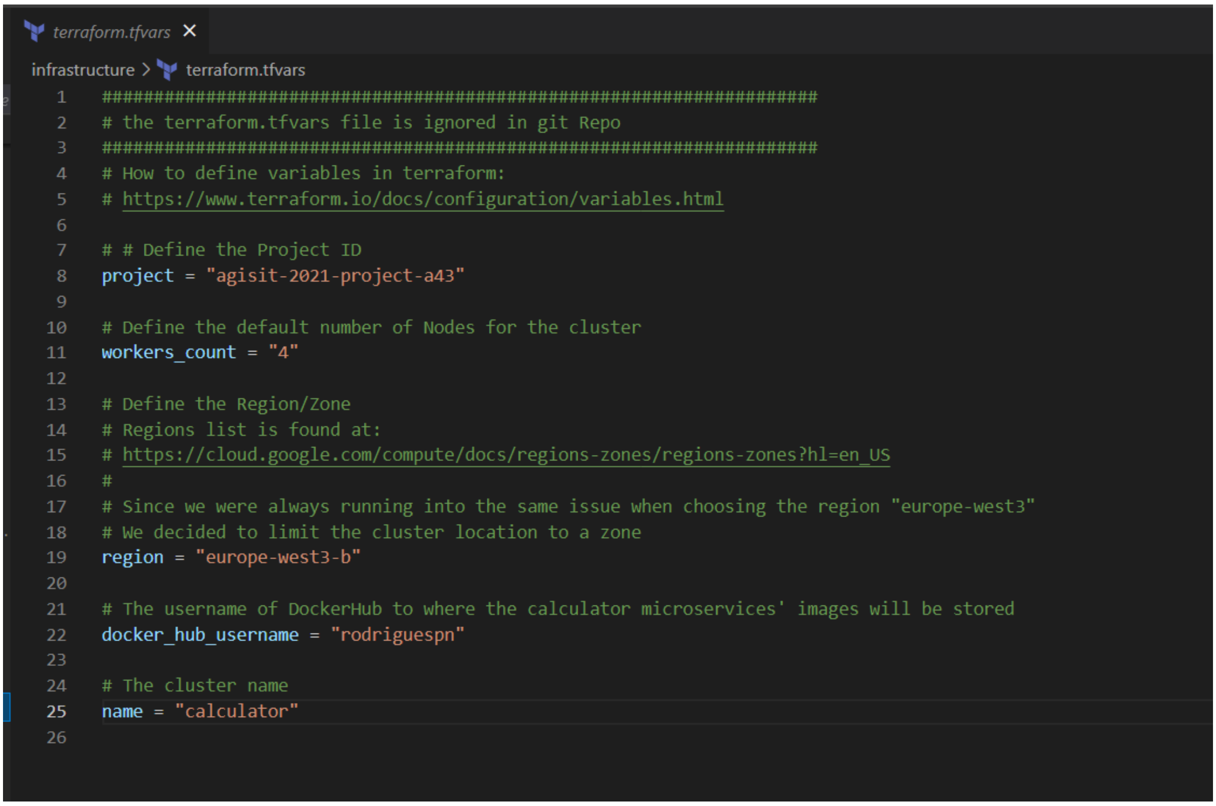
I-3.03 Run the command 'terraform init';

```
vagrant@mgmt-project:~/infrastructure terraform init
```

I-3.04 Followed by the commands 'terraform apply', when asked type 'yes'. The infrastructure will start to be built, it may take a while.

```
vagrant@mgmt-project:~/infrastructure terraform apply
....
Enter: yes
```

```
....
```



```
#####################################################################
# the terraform.tfvars file is ignored in git Repo
#####################################################################
# How to define variables in terraform:
# https://www.terraform.io/docs/configuration/variables.html

# # Define the Project ID
project = "agisit-2021-project-a43"

# Define the default number of Nodes for the cluster
workers_count = "4"

# Define the Region/Zone
# Regions list is found at:
# https://cloud.google.com/compute/docs/regions-zones/regions-zones?hl=en_US
#
# Since we were always running into the same issue when choosing the region "europe-west3"
# We decided to limit the cluster location to a zone
region = "europe-west3-b"

# The username of DockerHub to where the calculator microservices' images will be stored
docker_hub_username = "rodriguespn"

# The cluster name
name = "calculator"

```

Figure 2.1: Example of .tfvars file

2.3.4 Finishing the Experiment

I-4.01 Run the command ‘terraform destroy’. It may take a while;

```
vagrant@mgmt-project:~/infrastructure terraform destroy
```

I-4.02 Check on your GCP account if the resources were destroyed;

I-4.03 Exit the VM by running the command ‘exit’;

```
vagrant@mgmt-project:~$ exit
....
user@user-machine:~/project$
```

I-4.04 Clean all resources and destroy the VM ‘mgmt-project’ by running the commands ‘vagrant halt mgmt-project’, followed by ‘vagrant destroy mgmt-project’.

```
user@user-machine:~/project$ vagrant halt mgmt-project
....
user@user-machine:~/project$ vagrant destroy mgmt-project
....
```

Chapter 3

Methodology

3.1 Architecture

The Architecture of the Capstone Project consists of the following main services:

- **Frontend**
  - K8S **ingress** entry point that exposes inbound connections to reach the endpoints defined in the Backend services.
- **Backend services**
  - **VueCalc** microservice which renders the calculator UI.
  - **Express** microservice for addition and subtraction operations.
  - **Hapi** microservice for multiplication and division operations.
  - **Spring Boot** microservice to communicate with the Redis DataStore.
- **DataStore**
  - **Redis DataStore** to keep track of the calculator's operations history.
- **Monitoring**
  - **Grafana** service used for data visualization, data metrics and analytics.
  - **Prometheus** service used for software monitoring and as an alerting tool..

3.2 Diagram

The external user makes requests to the application. The **Frontend (ingress)** receives these requests and redirects them through the internal Internet Protocol (IP) to the designated service. The standard execution of these requests are channeled to **VueCalc** microservice. Which in turns makes requests to the other microservices. The **Spring Boot** microservice communicates to the **Redis DataStore**. The external user can have access to the monitoring service handled by the **Grafana** and **Prometheus** software. Diagram of the Architecture of the microservice based web application (Figure 3.1) created with https://www.draw.io and then exported as "jpg" crop format.

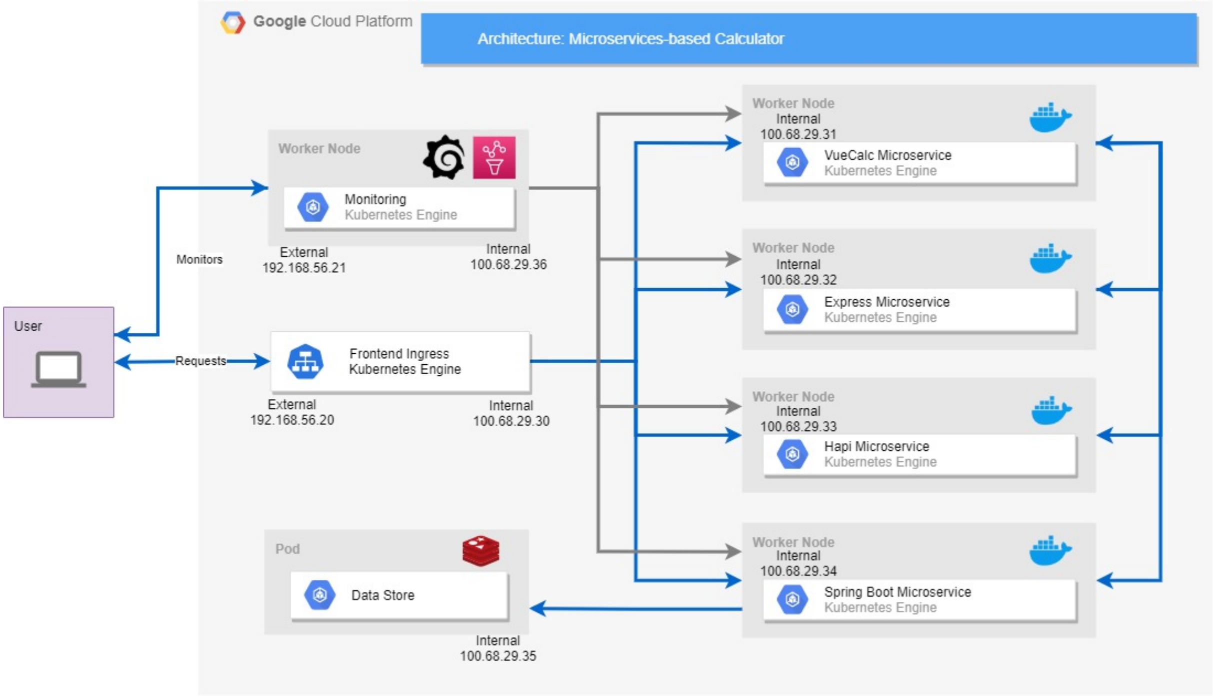


Figure 3.1: Report Diagram

Chapter 4

Sitography

- https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs
- https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/service
- https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/deployment
- https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/deployment
- https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/ingress
- https://github.com/khandelwal-arpit/kubernetes-starterkit