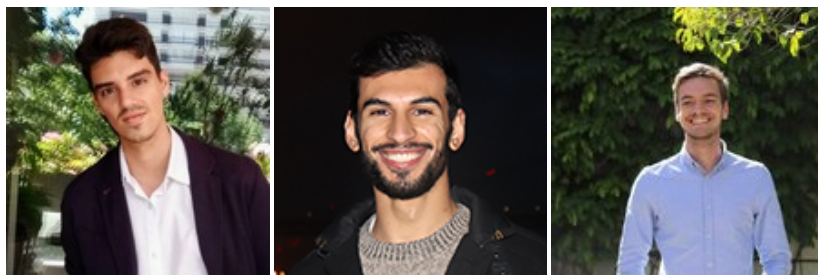


Project README

AGISIT 20201-2022

Authors

Team A43



Number	Name	Username	Email
81525	Carlos Antunes	https://git.rnl.tecnico.ulisboa.pt/ist181525	mailto:carlos.tejedor@tecnico.ulisboa.p
89451	Guilherme Areias	https://git.rnl.tecnico.ulisboa.pt/ist189541	mailto:guilherme.areias@tecnicoulisboapt
90531	Pedro Rodrigues	https://git.rnl.tecnico.ulisboa.pt/ist190531	mailto:pedronevesrodrigues@tecnico.ulisboa.pt

Module Leaders

- Designing the Architecture of the solution - Pedro.
- Configuration of the Monitoring system - Pedro.
- Definition of the Compute nodes capabilities - Guilherme.
- Networking and Metrics details - Carlos.
- Provisioning of the Infrastructure - Guilherme.
- Deployment of the Applications in the Containers - Carlos.
- Load Balancing strategy - Pedro.

Pre-Requisites

- Our project will be based on Deploying the Browser-based Calculator as a Microservices Achitecture deployed to a Google Cloud using Kubernetes Engine. For testing, we have first made a setup using minikube enablig us to deploy the application in our local machines.

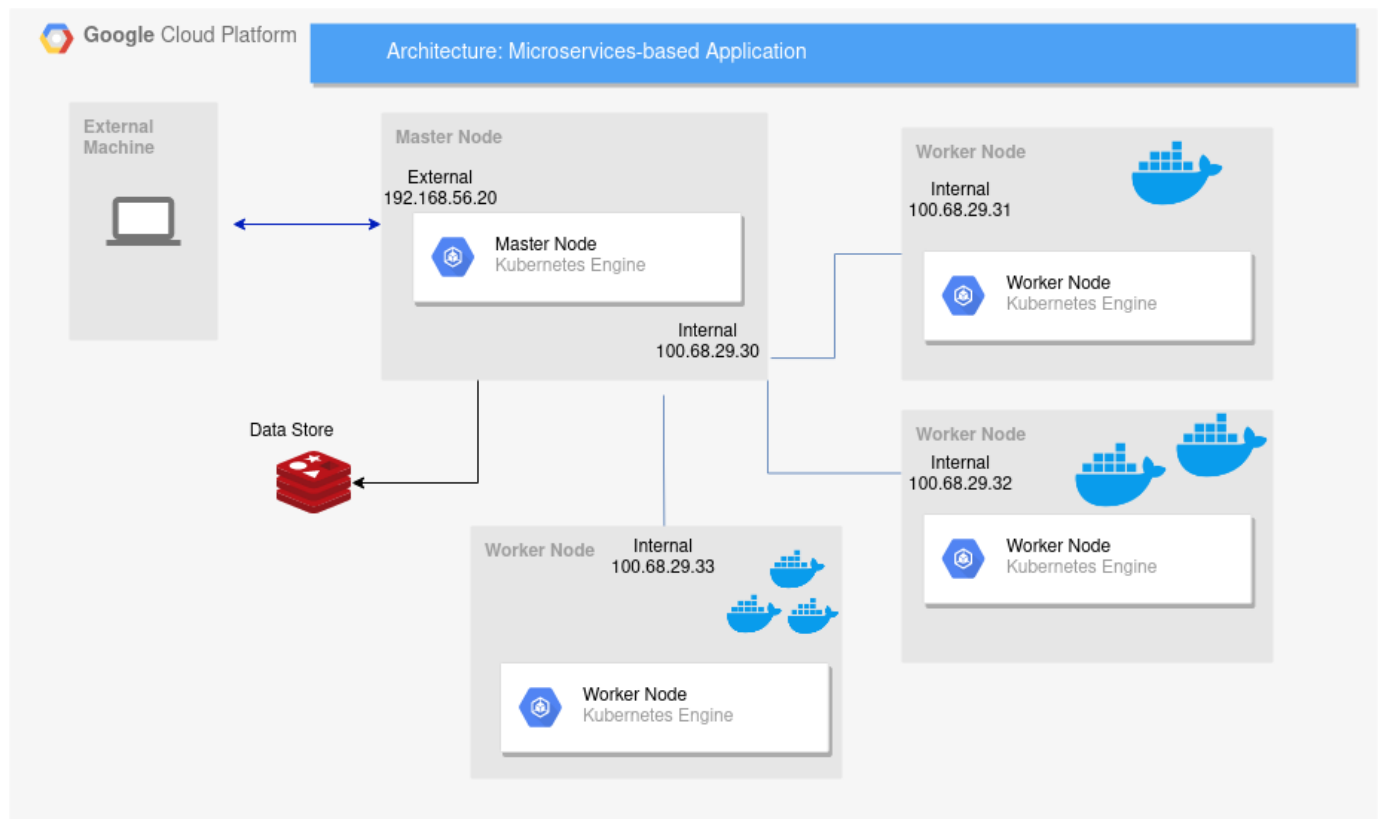
Project Files

- The main configuration file is the Vagrantfile. It builds and configures two virtual machines, a master node and a worker node.
- The bootstrap.sh contains instructions to install all dependencies needed on that VM, and sets the environment variables needed.
- The host_ip.sh configures the hosts file for the internal network.
- The hosts_ssh.sh enables the password authentication factor when trying to access that VM, via ssh.

Deployment

- The initial deployment is initialized by running the command 'vagrant up' in the directory 'team-43A/labs/project/project'. This command, runs the Vagrant file, wich contains the instructions to setup the working environment.
- When the working environment is created and running, run the command 'vagrant ssh mgmt'. This command, establishes a ssh connection between an external machine (the machine running this command) and the Master Node (mgmt).

Solution



- Via a ssh session, a external machine establishes a connection to the Master Node, of minikube, a local variation of the Kubernetes Orchestration.
- The Master Node has an external IP and an internal IP.
- In each Worker Node can have as many Docker Containers as the Master Node schedules to it. Each Container runs Microservices.
- Every Worker Node has its own internal IP each is used to communicate bewteen the Master Node as well as other Worker Nodes.
- The Data Store, where the cluster changes get stored in the key value store.

Implementation Options

(Brief description of the Options)

Final Remarks

- At the moment we desided to work with the Minikube orchestration, wich is local Kubernetes. Whitch runs a single-node Kubernetes cluster on a personal computer. After words, we will run a version of the project on the Google Cloud Platform, using the Kubernetes orchestration.