

# Docker Clusters and Google Cloud Platform

## System Benchmarking and Deployment

2020/2021

The main goal of this guide is to understand how to configure and deploy docker container clusters running different services/applications on top of Google Cloud Platform (GCP).

Useful Docker documentation is available at:

- Docker Documentation - <https://docs.docker.com>
- Step by Step Example - <https://docs.docker.com/get-started/>
- Swarm Routing Mesh - <https://docs.docker.com/engine/swarm/ingress/>

### Setup

1. Create two Virtual Machines at GCP (small instances) with Ubuntu 20.04. Check the course slides for further info about this step.
2. Configure SSH keys and connect to these through your terminal.
3. Ensure that both VMs can communicate with each other through their internal IPs.
4. Configure GCP firewall to allow traffic on port 8080.
5. Install the needed Docker utilities in both VMs. Docker and Docker-Compose utilities can be installed with the following commands:

```
sudo apt-get install -y apt-transport-https ca-
certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/
ubuntu/gpg | sudo apt-key add -
sudo apt-key fingerprint 0EBFCD88
sudo add-apt-repository "deb [arch=amd64] https
://download.docker.com/linux/ubuntu $(
lsb_release -cs) stable"
sudo apt-get -y update
```

```

sudo apt-get -y install docker-ce
sudo curl -L https://github.com/docker/compose/
releases/download/1.16.1/docker-compose -
uname -s'-'uname -m' -o /usr/local/bin/docker
-compose
sudo chmod +x /usr/local/bin/docker-compose

```

6. Copy (with the *scp* command) the *docker.env* folder and its content, created at the last practical guide, to the VMs.

### Docker-Swarm and Compose Utilities

1. Setup a Swarm cluster in which one of the VMs is the master and the other is the worker (<https://docs.docker.com/engine/swarm/swarm-tutorial/create-swarm/>).
- The VMs internal network interfaces should be used for cluster nodes communication.
2. Understand the usage of the command *docker node* to check if the nodes have successfully joined the Swarm cluster.
3. At the master VM modify the docker compose YAML file so that the *web* service now contemplates two replicas.
4. At the master node launch the docker stack with the command *docker stack deploy*. Note that the compose configuration file only needs to be present at the master node. However, since we are not using a docker registry for the tomcat docker image, this image must be present (built) on both VMs.
5. Check that the docker containers are running on both VMs (swarm nodes) with the *docker ps* command.
6. Check the service status at the Master (VM1) with *docker service inspect --pretty "service\_name"*.
7. Check in your computer browser that the tomcat service is accessible through both VM IPs.
8. At the master node downscale the service to one replica (*sudo docker service scale "service\_name"=1*).
9. Check that the docker containers are running on only one VM (swarm nodes) with the *docker ps* command.
10. Check in your computer browser that the tomcat service is still accessible through both VM IPs.

**Learning outcomes** Experiment GCP and linux Docker containers configuration and deployment. Experiment Docker cluster deployment and configuration. Revise Docker configuration parameters and deployment/management commands.