

Google Compute Engine

Introducción

En este práctico repetiremos lo realizado en los prácticos de Docker Machine y Docker Swarm (sugerimos repasarlos antes de continuar), pero en este caso alojaremos las machines y los nodos en la nube de Google a través de su IaaS [Google Compute Engine](#).

Requisitos previos

Todo lo necesario para los laboratorios de Docker, Docker Engine y Docker Machine.

Instalar la última versión de [Google Cloud SDK Tools](#).

Descargar usando:

```
wget  
https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-  
cloud-sdk-204.0.0-linux-x86_64.tar.gz
```

Descomprimirlo usando:

```
tar xvzf google-cloud-sdk-204.0.0-linux-x86_64.tar.gz
```

Instalar ejecutando:

```
./google-cloud-sdk/install.sh
```

Crear un proyecto y conectarlo a Gcloud

Ingresa a la consola de administración de [Google Cloud Platform](#) ya que es necesario crear un proyecto (se puede utilizar el de la práctica anterior) y además activar el servicio de Compute Engine (nos solicitará datos de tarjeta de crédito ya que es necesario activar la facturación de recursos).

Inicializar Gcloud utilizando:

```
gcloud init
```

Nos solicitará datos sobre la cuenta a utilizar, el proyecto a emplear y la zona por defecto donde se crearán las instancias.

Setear que Gcloud pueda acceder a nuestra cuenta con:

```
gcloud auth application-default login
```

Creación de las machines

En primer lugar crearemos las machines en GCE

```
docker-machine create mgr-1 -d google -google-machine-type n1-standard-1 --google-tags myswarm --google-project id-de-proyecto
```

```
docker-machine create mgr-2 -d google -google-machine-type n1-standard-1 --google-tags myswarm --google-project id-de-proyecto
```

```
docker-machine create wkr-1 -d google -google-machine-type n1-standard-1 --google-tags myswarm --google-project id-de-proyecto
```

```
docker-machine create wkr-2 -d google -google-machine-type n1-standard-1 --google-tags myswarm --google-project id-de-proyecto
```

Verificamos que las machines están creadas con:

```
docker-machine ls
```

Podemos verificar además las instancias de VM en GCE con:

```
gcloud compute instances list
```

Creación del swarm

Ahora crearemos un Swarm con dos nodos manager y dos nodos worker.

Ingresamos y configuramos el nodo mgr-1 (debemos anotar la IP interna del nodo mgr-1 de la salida de **docker-machine ls**, en este caso es 10.128.0.2):

```
docker-machine ssh mgr-1
```

```
sudo docker swarm init --advertise-addr 10.128.0.2 --listen-addr 10.128.0.2:2377
```

Anotamos el comando para agregar otro manager

```
sudo docker swarm join-token manager
```

Anotamos el comando para agregar workers

```
sudo docker swarm join-token worker
```

Salimos del nodo mgr-1

```
exit
```

Ingresamos al nodo mgr-2 y lo configuramos como manager:

```
docker-machine ssh mgr-2
```

```
sudo docker swarm join --token SWMTKN-1-1xti3u0ok4vetke04gacuetdrmkq-fn73gysll0z5wlyiaivtz5-63g0zyfomgz7hpx9lgai0j2mi 10.128.0.2:2377
```

```
exit
```

Ingresamos al nodo wkr-1 y lo configuramos como worker:

```
docker-machine ssh wkr-1
```

```
sudo docker swarm join --token SWMTKN-1-1xti3u0ok4vetke04gacuetdrmkq-  
fn73gyslloz5wlyiaivtz5-8xfg7dwgba5pizsfp06vl3n3b 10.128.0.2:2377
```

```
exit
```

Ingresamos al nodo wkr-2 y lo configuramos como worker:

```
docker-machine ssh wkr-2
```

```
sudo docker swarm join --token SWMTKN-1-1xti3u0ok4vetke04gacuetdrmkq-  
fn73gyslloz5wlyiaivtz5-8xfg7dwgba5pizsfp06vl3n3b 10.128.0.2:2377
```

```
exit
```

Ingresamos nuevamente al nodo mgr-1 y vemos el estado del swarm con:

```
sudo docker node ls
```

Creamos una red overlay para el swarm

Crearemos una red overlay para utilizarla en los servicios del cluster. Ingresar al nodo mgr-1 y ejecutar:

```
sudo docker network create --driver overlay nw1
```

Podemos verificar la creación de la red mediante el comando:

```
sudo docker network ls
```

Creamos un servicio

Ahora crearemos un servicio basado en la imagen nginx y con 6 réplicas.

En el nodo mgr-1 ejecutar el comando:

```
sudo docker service create --replicas 6 --network nw1 -p 80:80/tcp --  
name nginx --detach=false nginx
```

Podemos verificar la creación y el estado del servicio con:

```
sudo docker service ls
```

Podemos verificar además la distribución de las réplicas usando:

```
sudo docker service ps nginx
```

Recursos

- [Docker Docs - Create machines on Google Compute Engine](#)
- [Google Compute Engine Documentation](#)

Elaborar un informe de entrega grupal con las salidas de pantalla de todas las actividades comprendidas en el presente práctico con fecha límite 30/06/2019.