

# Parcial 3 - Daniel Gutierrez - A00320176

Repositorio: <https://github.com/dgutierrez1/sd-exam3/tree/A00320176/add-solution>

## Comandos de linux necesarios para el aprovisionamiento de los servicios solicitados

Se deben utilizar diferentes comandos dependiendo del microservicio que se este configurando

Comando	Descripcion
<code>pip install -r requirements.txt</code>	Se utilizo para la instalacion de los paquetes necesarios para el funcionamiento de la aplicacion de Flask. Los paquetes estaban especificados en el archivo requirements.txt
<code>firewall-cmd --zone=public --add-port=80/tcp --permanent</code>	Se usa para abrir el puerto especificado
<code>yum -y install wget &amp;&amp; yum -y install unzip &amp;&amp; yum -y install</code>	Se usa para instalar los paquetes necesarios en el contenedor

El uso de Docker y Docker-compose disminuye sustancialmente el uso de otros comandos que serian necesarios en la configuracion de la maquina normalmente.

## Escriba los archivos Dockerfile para cada uno de los servicios solicitados

### Haproxy

```
#Instalacion
RUN yum -y install wget && yum -y install unzip && yum -y install haproxy
```

```

#Configuracion de puertos
EXPOSE 80 8080

#Instalacion de consul-template
ENV CONSUL_TEMPLATE_VERSION 0.19.3
ADD https://releases.hashicorp.com/consul-template/${CONSUL_TEMPLATE_VERSION}/consul-template_${CONSUL_TEMPLATE_VERSION}_SHA256SUMS /tmp/
ADD https://releases.hashicorp.com/consul-template/${CONSUL_TEMPLATE_VERSION}/consul-template_${CONSUL_TEMPLATE_VERSION}_linux_amd64.zip /tmp/

#Configuracion de consul-template
RUN cd /tmp && \
    sha256sum -c consul-template_${CONSUL_TEMPLATE_VERSION}_SHA256SUMS 2>&1 |
grep OK && \
    unzip consul-template_${CONSUL_TEMPLATE_VERSION}_linux_amd64.zip && \
    mv consul-template /bin/consul-template && \
    rm -rf /tmp

# Copiar el archivo de configuracion del haproxy
WORKDIR /etc/haproxy
ADD haproxy.ctmpl .

```

## Web

```

FROM python:3.4-alpine
ADD . /code
WORKDIR /code
#EXPOSE 5000
RUN pip install -r requirements.txt
CMD ["python", "app.py"]

```

## Docker-compose

```

docker-compose.yml
version: '2'

services:

  consul:
    image: troyfontaine/consul-server
    container_name: consul
    ports:
      - 8500:8500
    command: "-bind=0.0.0.0 -server -bootstrap"

  registrator:
    image: "gliderlabs/registrator:latest"

```

```

container_name: registrator
command: "-internal consul://consul:8500"
links:
  - consul:consul
volumes:
  - /var/run/docker.sock:/tmp/docker.sock

balancer:
  build: ./haproxy
  container_name: balancer
  ports:
    - 80:80
  links:
    - consul:consul
  command: consul-template -consul-addr=consul:8500 -
template="/etc/haproxy/haproxy.ctmpl:/etc/haproxy/haproxy.cfg:/usr/sbin/haproxy
-f /etc/haproxy/haproxy.cfg -D -p
/var/run/haproxy.pid"
  environment:
    - SERVICE_NAME=Balanceador

redis:
  image: "redis:alpine"

web:
  build: ./web
  ports:
    - 5000
  volumes:
    - ./web:/code
  environment:
    - SERVICE_NAME=Web

```

## Evidencias funcionamiento

---

Escalando los contenedores

web

```
root@ubuntu-xenial:/home/ubuntu/sd-exam3/A00320176# docker-compose scale web=10
WARNING: The scale command is deprecated. Use the up command with the --scale flag instead.
Starting a00320176_web_1 ...
Starting a00320176_web_1 ... done
Creating a00320176_web_2 ...
Creating a00320176_web_3 ...
Creating a00320176_web_4 ...
Creating a00320176_web_5 ...
Creating a00320176_web_6 ...
Creating a00320176_web_7 ...
Creating a00320176_web_8 ...
Creating a00320176_web_9 ...
Creating a00320176_web_10 ...
Creating a00320176_web_2 ... done
Creating a00320176_web_3 ... done
Creating a00320176_web_4 ... done
Creating a00320176_web_5 ... done
Creating a00320176_web_6 ... done
Creating a00320176_web_7 ... done
Creating a00320176_web_8 ... done
Creating a00320176_web_9 ... done
Creating a00320176_web_10 ... done
```

La lista de los contenedores

corriendo

```
root@ubuntu-xenial:/home/ubuntu/sd-exam3/A00320176# docker-compose ps

```

Name	Command	State	
a00320176_redis_1	docker-entrypoint.sh redis ...	Up	6379/tcp
a00320176_web_1	python app.py	Up	0.0.0.0:32794->5000/tcp
a00320176_web_10	python app.py	Up	0.0.0.0:32801->5000/tcp
a00320176_web_2	python app.py	Up	0.0.0.0:32803->5000/tcp
a00320176_web_3	python app.py	Up	0.0.0.0:32795->5000/tcp
a00320176_web_4	python app.py	Up	0.0.0.0:32798->5000/tcp
a00320176_web_5	python app.py	Up	0.0.0.0:32796->5000/tcp
a00320176_web_6	python app.py	Up	0.0.0.0:32797->5000/tcp
a00320176_web_7	python app.py	Up	0.0.0.0:32799->5000/tcp
a00320176_web_8	python app.py	Up	0.0.0.0:32802->5000/tcp
a00320176_web_9	python app.py	Up	0.0.0.0:32800->5000/tcp
balancer	consul-template -consul-ad ...	Up	0.0.0.0:80->80/tcp, 8080/tc
consul	/bin/consul agent -server ...	Up	8300/tcp, 8301/tcp, 8301/ud
registrator	/bin/registrator -internal ...	Up	

El balanceador direcciona las peticiones a los diferentes contenedores disponibles



Esta pagina ha sido vista 22 veces. El nombre del host es 525e8dad7933

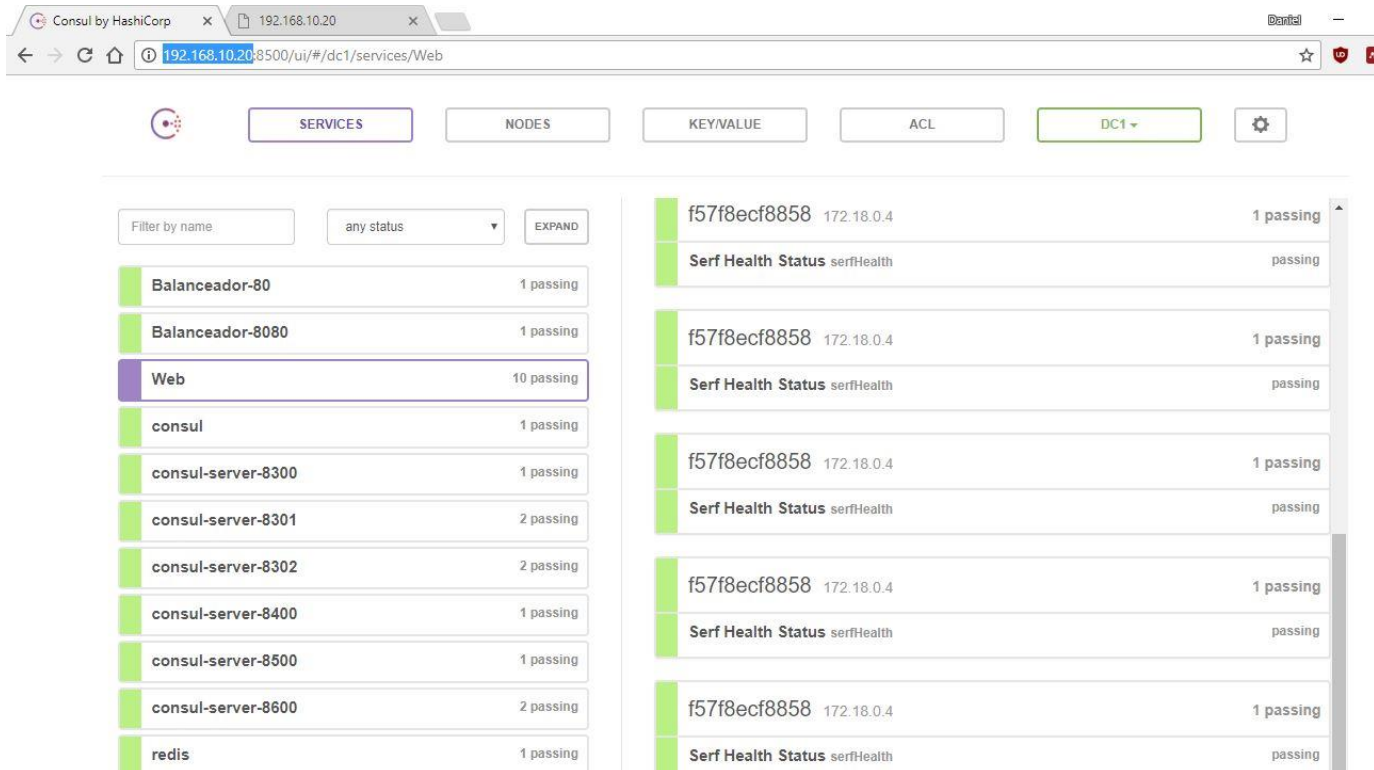


Esta pagina ha sido vista 24 veces. El nombre del host es 084fb5e30fc7



Esta pagina ha sido vista 25 veces. El nombre del host es 0a8c329a6fff

## Evidencia en Consul



## Problemas encontrados y las acciones efectuadas para su solución

Problema	Solucion
Poder evidenciar efectivamente que el balanceador esta direccionando a diferentes nodos.	En la aplicacion web se uso una libreria de Python con la que se puede obtener el nombre del host y mostrarlo al usuario
Errores en el balanceador usando NGINX	Inicialmente se intento usar NGINX pero en la configuración con el template hubo muchos errores, la mayoría durante la creacion del contenedor. El consul template tambien genero errores porque el repositorio de donde se estaba obtenido había cambiado su estructura. Al final se

Problema	Solucion
	utilizó HAPROXY y se pudo configurar el balanceador fácilmente.