

Examen 3

Nombre Estudiante: Emmanuel Bolaños Ocampo
Código Estudiante: A00309828

Repositorio

<https://github.com/emmanuel8664/sd-exam3/blob/master/A00309828/Informe.md>

Universidad ICESI

Curso: Sistemas Distribuidos

Docente: Daniel Barragán C.

Tema: Automatización de infraestructura (Vagrant+Chef)

Correo: daniel.barragan at correo.icesi.edu.co

Objetivos

Realizar de forma autónoma el aprovisionamiento automático de infraestructura

Diagnosticar y ejecutar de forma autónoma las acciones necesarias para lograr infraestructuras estables

Prerrequisitos

Docker

Docker-Compose

Contenedores: consul, consul-template, registrator, load balancer (nginx, haproxy)

Descripción

Deberá realizar el aprovisionamiento de un ambiente compuesto por los siguientes elementos: un servidor web con capacidad de escalar a N instancias (puede

emplear apache+php o crear un servicio web con el lenguaje de su preferencia), un balanceador de carga para redireccionar las peticiones a los servidores web.

Tenga en cuenta:

Para el aprovisionamiento deberá usar docker-compose

Emplear una herramienta de descubrimiento de servicio (zookeeper, consul, etcd) que permita registrar automáticamente las nuevas instancias de servidores web. Las tecnologías de descubrimiento de servicio se componen de agentes y un servidor ó clúster de servidores. Los agentes envían información al clúster acerca de los servicios que se ejecutan en las instancias. El servidor registran los servicios que son anunciados por los agentes para ser consultados por los clientes ú otros servicios. Para evitar ejecutar mas de un servicio por contenedor (agente de consul y servicio web) emplee la aplicación dockerizada registrator (ó una tecnología similar) para registrar los nuevos contenedores ante el servidor de descubrimiento de servicio. Para actualizar la configuración de los archivos de configuración del balanceador de carga y reiniciar el servicio emplee la aplicación consul-template. Consul-template consulta al servidor de consul el estado de los servicios y ante un cambio en ellos, a partir de plantillas, crea nuevamente los archivos de configuración.

Consigne los comandos de linux necesarios para el aprovisionamiento de los servicios solicitados. En este punto no debe incluir archivos tipo Dockerfile solo se requiere que usted identifique los comandos o acciones que debe automatizar

```
docker run -d --name=consul -p 8500:8500 consul
docker run -d \
  --name=registrator \
  --net=host \
  --volume=/var/run/docker.sock:/tmp/docker.sock \
  gliderlabs/registrator:latest \
  -internal \
  consul://localhost:8500
docker logs registrator
curl localhost:8500/v1/catalog/services
```

```

from flask import Flask
from redis import Redis
import os
import socket

app = Flask(__name__)
redis = Redis(host='redis', port=6379)
host = socket.gethostname()

@app.route('/')
def hello():
    count = redis.incr('hits')
    return 'Hello World! I have been seen %s times. My Host name is %s\n\n' % (count ,host)

if __name__ == "__main__":
    app.run(host="0.0.0.0", debug=True)

```

Escriba los archivos Dockerfile para cada uno de los servicios solicitados junto con los archivos fuente necesarios. Tenga en cuenta consultar buenas prácticas para la elaboración de archivos Dockerfile DockerFile Haproxy

```

FROM centos:latest

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

#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/

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

WORKDIR /etc/haproxy
ADD haproxy.ctmpl .

```

DockerFile Web

```

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

```

Escriba el archivo docker-compose.yml necesario para el despliegue de la infraestructura. No emplee configuraciones deprecated. Incluya un diagrama general de los componentes empleados

```
version: '2'
services:
  consul:
    image: consul
    container_name: consul
    ports:
      - 8500:8500

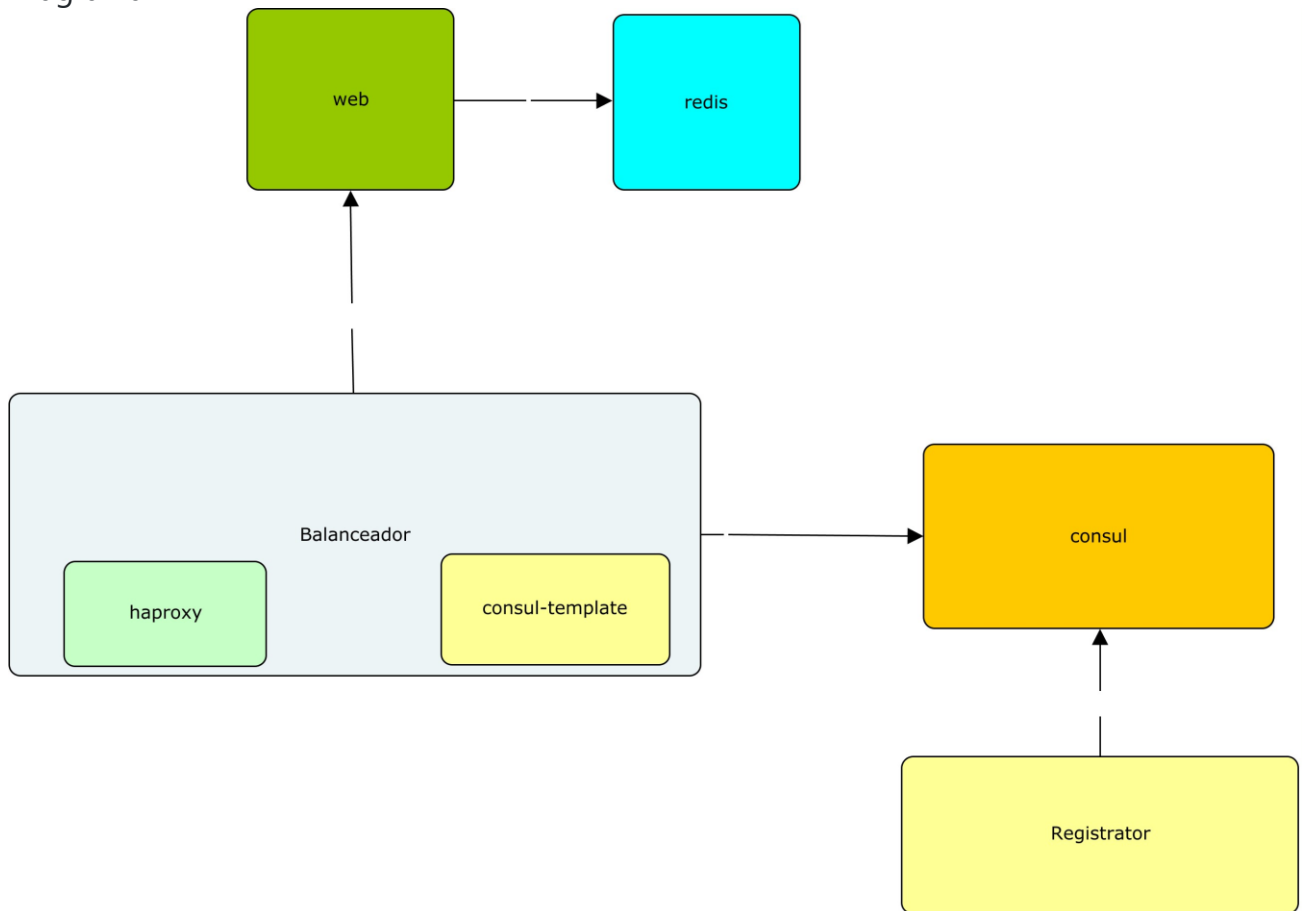
  registrator:
    image: gliderlabs/registrator:latest
    container_name: registrator
    links:
      - consul:consul
    volumes:
      - /var/run/docker.sock:/tmp/docker.sock
    command: -internal consul://consul:8500

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

  web:
    build: web/.
    labels:
      - "SERVICE_NAME=Python"
    ports:
      - "5000"

  redis:
    image: "redis:alpine"
    container_name: redis
    ports:
      - "6379"
```

Diagrama



Incluya evidencias que muestran el funcionamiento de lo solicitado

Name	Command	State
a00309828_web_1	python app.py	Up 0.0.0.0:32801->5000/tcp
consul	docker-entrypoint.sh agent ...	Up 8300/tcp, 8301/tcp, 8301/udp, 8302/tcp, 8...
haproxy	consul-template -consul-ad ...	Up 0.0.0.0:80->80/tcp
redis	docker-entrypoint.sh redis ...	Up 0.0.0.0:32802->6379/tcp
registrador	/bin/registrator -internal ...	Up

← → ↻ ⓘ localhost

Aplicaciones Para acceder rápidamente, coloca tus favoritos aquí en la barra de favoritos

Hello World! I have been seen 1 times. My Host name is 13c750108f5c

Name	Command	State	
a00309828_web_1	python app.py	Up	0.0.0.0:32801->5000/tcp
a00309828_web_2	python app.py	Up	0.0.0.0:32804->5000/tcp
a00309828_web_3	python app.py	Up	0.0.0.0:32803->5000/tcp
consul	docker-entrypoint.sh agent ...	Up	8300/tcp, 8301/tcp, 8301/udp, 8302/tcp, 8
haproxy	consul-template -consul-ad ...	Up	0.0.0.0:80->80/tcp
redis	docker-entrypoint.sh redis ...	Up	0.0.0.0:32802->6379/tcp
registrador	/bin/registrator -internal ...	Up	
backend nodes			
balance	roundrobin		
server	cdde7be54c72:a00309828_web_1:5000	172.24.0.2:5000	
server	cdde7be54c72:a00309828_web_2:5000	172.24.0.8:5000	
server	cdde7be54c72:a00309828_web_3:5000	172.24.0.7:5000	

Name	Command	State	
a00309828_web_1	python app.py	Up	0.0.0.0:32801->5000/tcp
a00309828_web_2	python app.py	Up	0.0.0.0:32804->5000/tcp
a00309828_web_3	python app.py	Up	0.0.0.0:32803->5000/tcp
a00309828_web_4	python app.py	Up	0.0.0.0:32806->5000/tcp
a00309828_web_5	python app.py	Up	0.0.0.0:32807->5000/tcp
a00309828_web_6	python app.py	Up	0.0.0.0:32805->5000/tcp
consul	docker-entrypoint.sh agent ...	Up	8300/tcp, 8301/tcp, 8301/udp, 8302/tcp, 8
haproxy	consul-template -consul-ad ...	Up	0.0.0.0:80->80/tcp
redis	docker-entrypoint.sh redis ...	Up	0.0.0.0:32802->6379/tcp
registrador	/bin/registrator -internal ...	Up	
backend nodes			
balance	roundrobin		
server	cdde7be54c72:a00309828_web_1:5000	172.24.0.2:5000	
server	cdde7be54c72:a00309828_web_2:5000	172.24.0.8:5000	
server	cdde7be54c72:a00309828_web_3:5000	172.24.0.7:5000	
server	cdde7be54c72:a00309828_web_4:5000	172.24.0.10:5000	
server	cdde7be54c72:a00309828_web_5:5000	172.24.0.11:5000	
server	cdde7be54c72:a00309828_web_6:5000	172.24.0.9:5000	

Documento algunos de los problemas encontrados y las acciones efectuadas para su solución al aprovisionar la infraestructura y aplicaciones

Problema: comunicar haproxy con consul sin depender de la ip

Solucion:

Para que haproxy se actualice dinámicamente debe ejecutar consul-template en background. Se pensó ejecutarlo usando la directiva RUN de Dockerfile pero se obtuvieron problemas y además se dependía de la ip con la cual quedara el contenedor de consul. Por ello, se decidió ejecutar el comando necesario directamente desde docker-compose usando la directiva command

```
balanceador:
  build: haproxy/.
  container_name: haproxy
  labels:
    - "SERVICE_NAME=Balanceador"
  ports:
    - "80:80"
  links:
    - consul:consul
  command: consul-template -consul-addr=consul:8500 -template="/etc/haproxy/haproxy.ctmpl:/etc/h
```