Anexos

Apéndice A

Despliegue de Kubernetes con Ansible

Uno de los objetivos del trabajo era el despliegue de Kubernetes en múltiples máquinas. Para ello, se decidió hacer uso de herramientas de automatización, para simplificar dicho trabajo. Aunque ya existen herramientas que permiten desplegar Kubernetes de manera automática (como kops), dichas herramientas están orientadas a complejos despliegues de producción, especialmente en nubes públicas como AWS, OpenStack o GCE, y escapan a la escala de este trabajo. Por tanto, se decidió la creación de un conjunto de manifiestos Ansible, mucho más sencillos que las soluciones mencionadas, pero mucho más adecuados para los despliegues a realizar (despliegues parciales y con una configuración bastantes estática).

Apéndice B

Plugin de red

La filosofía de Kubernetes consiste en dar una IP a cada pod, de tal forma que cada clúster contiene una subred interna (normalmente del rango 10.0.0.0/8, para evitar la falta de IPs), sin embargo, Kubernetes no hace esa tarea, sino que la delega a lo que denomina un "plugin de red"[1]. Esos plugins utilizan una interfaz denominada *Container Networking Interface*, comúnmente abreviada como CNI. Dicha interfaz especifica la forma de detallar conexiones de red y relaciones entre contendores, de forma que se puedan crear redes virtuales sobre la red física que permitan interconectar contenedores en distintos *hosts* físicos.

Existen gran cantidad de plugins de red, algunos motores de redes creados previamente y otros creados especificamente para su uso con Kubernetes. Algunos de los ejemplos son Calico, Flannel, Romana, Weave…No todos los plugins soportan todas las funcionalidades de red que ofrece Kubernetes, y algunas (Como Ingress o Network Policies) solo están en algunos de los plugins. Para este trabajo, se han valorado 2 de estos plugins: Flannel y Weave. Flannel se considero por venderse como una solución sencilla, desarrollado por CoreOS (Una distribución de Linux orientada a la containerización) y nativa a Kubernetes, mientras que Weave se ofrece como una solución compleja y potente, capaz de crear redes complejas y adaptables, no creada para Kubernetes pero que se integra con Docker.

Tras unos pocos experimentos, instalando y configurando ambas soluciones con varias situaciones, se ha decidido instalar Weave tanto en el *Edge* como en la nube. Esta elección se ha hecho en base a que ofrece la capacidad de crear configuraciones complejas, y a la capacidad de interconectar distintos clusters.

La instalación de este plugin se ha realizado con el despliegue Kubernetes ofrecido por el propio WeaveWorks, con el manifiesto `https://github.com/weaveworks/weave/releases/download/latest_release/weave-daemonset-k8s-1.6.yaml`

[1] Existe un plugin integrado, denominado kubenet; pero es sencillo, no soporta configuraciones complejas y Kubernetes recomienda el uso de alguno de los otros existentes

El funcionamiento de este plugin de red se basa en el módulo de Linux *iptables*. Weave crea un pod (denominado "weave-net-…") en cada host del clúster (incluido el maestro), dichos pods estan conformados por dos contenedores, *weave* y *weave-npc*. El primero se encarga de establecer la conexión con el clúster a traves del host, mientras que el segundo se encarga de la gestión de la red *per se*. Para la gestión de la red, Weave asocia una subred a cada host, y crea reglas iptables en todos los host, de tal forma que pueda conectar dichas subredes internas entre todos los *host*, de esa forma, contenedores en máquinas distintas pueden conectarse de manera transparente.