

Kubernetes - Configuración y optimización de sistemas de cómputo

Francesc Folch Company

February 21, 2022

Abstract

1 Kubernetes (K8S)

1.1 Linux cgroups y namespaces

Los cgroups (grupos de control)[1] son una funcionalidad del kernel de Linux que permite a los procesos ser organizados y separados mediante una jerarquía grupal. Esto habilita la compartimentalización de recursos, ya que se puede limitar el uso de CPU y memoria a unos procesos de un mismo cgroup.

Los namespaces [2] son divisiones de los procesos para que puedan acceder a recursos globales de manera aislada y segura. Esto permite que, por ejemplo, un servidor con Kubernetes esté aislado del resto de procesos del sistema. Esto previene posibles riesgos de seguridad.

1.2 Arquitectura y componentes Kubernetes

Un clúster Kubernetes es un conjunto de nodos que ejecutan contenedores, esto es, aplicaciones de virtualización ligera [3]. Se compone de un plano de control (como mínimo) y de workers, que son las máquinas que realizan el trabajo. Lo bueno de un clúster Kubernetes es que los contenedores no están ligados a una sola máquina, si no que son repartidos a lo largo del clúster, abstrayendo la capa "física".

Los componentes son los siguientes:

- Plano de control: Se asignan las tareas y se coordinan los nodos para mantener un estado deseado.

- **Nodos:** Son las máquinas que realizan el trabajo que se les ha asignado desde el plano de control.
- **Pod:** Es el conjunto de contenedores desplegados en un solo nodo.
- **Labels:** Son pares de clave/valor que se asignan a un pod y describen atributos.
- **Servicio:** Es la manera de ofrecer una aplicación corriendo en un conjunto de Pods como un servicio en la red. Lo que unifica un conjunto de Pods de una misma finalidad conjunta.
- **ReplicaSets:** Es una API que permite correr varias instancias de un mismo Pod, para aumentar la disponibilidad de dicho Pod, por si uno falla.
- **Deployments:** En un despliegue se especifican los Pods y sus ReplicaSets, esto sirve para definir o actualizar ReplicaSets, llegando al estado deseado por el Deployment.
- **Replication Controller:** Si se han creado demasiadas réplicas de un Pod, y algunas están "ociosas", el Replication Controller las terminará. Por otro lado, si hay muy pocos Pods, este iniciará más. También reemplazará los pods que hayan fallado y hayan caído.
- **Volumen:** Es un sistema de almacenamiento de información y datos accesible por los contenedores de un Pod, estos datos son persistentes aunque los contenedores se reinicien.
- **Namespace:** Se utiliza la funcionalidad de namespaces en Linux para separar diferentes clusters corriendo en un mismo servidor.

1.3 Networking y acceso externo

Existen diferentes tipos de comunicaciones entre los componentes de un clúster[4]. Principalmente 4:

1. Contenedor-Contenedor: Se resuelve con los Pods, con comunicaciones en localhost.
2. Pod-Pod
3. Pod-Servicio
4. Exterior-Servicio

Cada pod recibe una IP propia, y si se quieren comunicar varios pods en un mismo nodo no hará falta NAT, mientras que si son de distintos nodos, se aplicará NAT.

Los servicios internos tendrán una IP virtual proporcionado por kube-proxy y los servicios externos tendrán una IP accesible desde fuera del cluster.

2 Creación de un cluster Kubernetes

2.1 Con kubeadm

Para crear un cluster kubernetes primero he creado tres máquinas virtuales CentOS 7 llamadas k8smaster, k8sworker1 y k8sworker2. He usado VirtualBox y Vagrant para administrar y lanzar fácilmente cada una de las máquinas. He creado 2 adaptadores de red para cada una: un NAT para acceder por ssh a las máquinas y http a k8smaster; y un Host Only para la comunicación interna entre ellos. Estas IP son estáticas, y se especifica en el VagrantFile.

Todas las máquinas tienen 2 CPUs, 2048MB de RAM y 10GB de almacenamiento.

Tras la configuración básica se ha seguido la guía para la instalación de Kubernetes[5]. Después de iniciar kubeadm nos mostrará el siguiente comando para poder acceder al master desde cualquier worker:

```
kubeadm join 192.168.56.10:6443 --token q0jxgr.f0766vuojlq30zi8 \
--discovery-token-ca-cert-hash \
sha256:a95ce758c13fc48aba9a3cad92e9dae219df42d3daed767893cc91b4ce661a
```

Después se instala la red calico y ejecutamos el comando anterior en cada nodo para unirse al cluster.

Tras unir todos los nodos al control-plane, se creará el deployment nginx, especificando el número de réplicas.

Con `kubectl get pods -o wide` podemos ver el estado de los pods y a que nodo se ha asignado cada uno. Finalmente, para poder hacer uso de nginx desde el exterior se ha de crear un servicio, linkeandolo al deployment.

2.2 Con ansible

Visto el proceso "manual" desplegando uno por uno cada nodo, ahora veremos el proceso automático, que permitirá desplegar numerosos nodos de manera sencilla y veloz. Solo hará falta un nodo extra, que será el controlador de ansible, y que se comunicará por ssh con el resto de máquinas que formarán parte del cluster Kubernetes.

El único requisito es que cada máquina tenga habilitada una conexión ssh, y que tengan una IP estática. De esta manera, ansible leerá los archivos yaml para provisionar a cada nodo, diferenciándolos en másters o workers. El ejemplo utilizado [6] es un repositorio de github donde se encuentran todos los archivos necesarios para provisionar un master y dos nodos en este caso, especificando el nombre de cada uno en el archivo hosts.

Una vez lanzados los tres yaml con el comando `ansible-playbook` tendremos el cluster creado y podremos desplegar el servicio nginx desde el master de igual manera que hicimos con kubeadm.

References

- [1] cgroups(7). *Linux Programmer's Manual*.
`https://www.man7.org/linux/man-pages/man7/cgroups.7.html`;
[Consulta: 28 de enero de 2022]
- [2] namespaces(7). *Linux Programmer's Manual*.
`https://man7.org/linux/man-pages/man7/namespaces.7.html`; [Con-
sulta: 28 de enero de 2022]
- [3] What is a Kubernetes cluster?. *Red Hat*.
`https://www.redhat.com/en/topics/containers/
what-is-a-kubernetes-cluster`; [Consulta: 12 de febrero de 2022]
- [4] Cluster Networking. *kubernetes*.
`https://kubernetes.io/docs/concepts/cluster-administration/
networking/`; [Consulta: 12 de febrero de 2022]
- [5] Create Kubernetes Cluster Using Kubectl — How To Setup Kubernetes Cluster On Linux Networking. *YouTube*.
`https://www.youtube.com/watch?v=cAZ5nkLfL6M`; [Consulta: 12 de
febrero de 2022]
- [6] Ansible k8s setup. *GitHub*.
`https://github.com/networknexus/ansible-k8s-setup`; [Consulta: 12
de febrero de 2022]