81

Contenedor: no es una entidad. Son distintas tecnologias que trabajando en conjunto crean un contenedor.

Las 3 tecnologias son:

**CGroups:** Asignan a cada contenedor/proceso los recursos va a utilizar (memoria, disco, cpu). Limitan el uso de recursos del sistema operativo para cada contenedor.

**Chroot:** Nos permite que nuestro proceso/container tenga visibilidad sobre archivos donde tiene que trabajar y no acceder a otros recursos del sistema.

**Namespaces:** (son 7, aqui los mas importantes):

* Mount: Nos permite que nuestro proceso tenga una visibilidad reducida de los directorios. Esto permite que dos contenedores que trabajen sobre un sistema de archivos no se interfieran entre si.
* Networking: Permite que cada contenedor tenga su dirección IP, su tabla de rutas, su interfaz de red, y que no interfiera con otros contenedores.  
  Concepto de POD: Entidad atomica scheduleable - Entidad sobre la cual kubernetes va a ejecutar los contenedores. (se verá en detalle más adelante).
* PID o de proceso: si ejecutamos un ps cuando ejecutamos nuestro contenedor, vamos a ver que nuestro contenedor es el process id 1 y no vemos todo el resto de los procesos del SO. Esto es posible gracias al namespace de procesos. (se verá en detalle más adelante).

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente

**Nodo Master**

1. **API Server:** A lo que todo se conecta, los agentes, el CLI, el dashboard etc. Cuando se cae un nodo master es lo que se pierde. Se usa el algoritmo de ruft para algoritmo de elección.
2. **Scheduler:** Cuando se deben crear un job, un pod en máquinas específicas, el scheduler se encarga de asignar las tareas y administrar los flujos de trabajos, revisando siempre las restricciones y los recursos disponibles.
3. **Controller Manager:** Es un proceso que está en un ciclo de reconciliación constante buscando llegar al estado deseado con base al modelo declarativo con el que se le dan instrucciones a K8s.  
   ++Tipos de controller manager: ++

* Replica manager
* Deployment manager
* Service manager  
  …

1. **Etcd:** Key value store que permite que el cluster este altamente disponible.

**Componentes muy importantes que viven en los nodos:**

1. **Kubelet:** Agente de kubernetes, se conecta con el control play y le pregunta que recursos (pods, contenedores) debo correr al scheduler via API Server. Monitorea los pods constantemente para saber si están vivos, los recursos disponibles etc y le comunica constantemente al scheduler via API Server.
2. **Kube-proxy:** Se encarga de balancear el tráfico que corre en nuestros contenedores/servicios. Una vez llega una request se encarga de decidir a que pod y contenedor debe de ir.

**Nodos == Minions**

Todos los nodos y masters están conectados a una red física para poder hablarsen entre sí.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto, Aplicación

Descripción generada automáticamente

Recuerda que el modelo TCP/IP no se corresponde capa a capa con el modelo OSI, en el modelo TCP/IP la capa de red es la capa 2, la de transporte (UDP y TCP) capa 3 y la capa 4 la de usuario o aplicación.