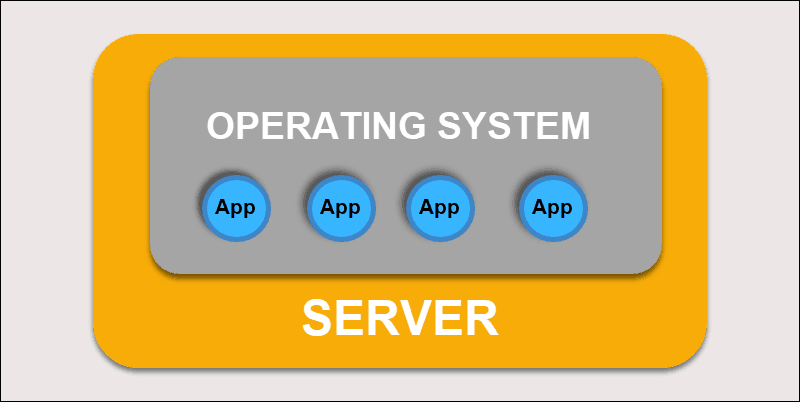
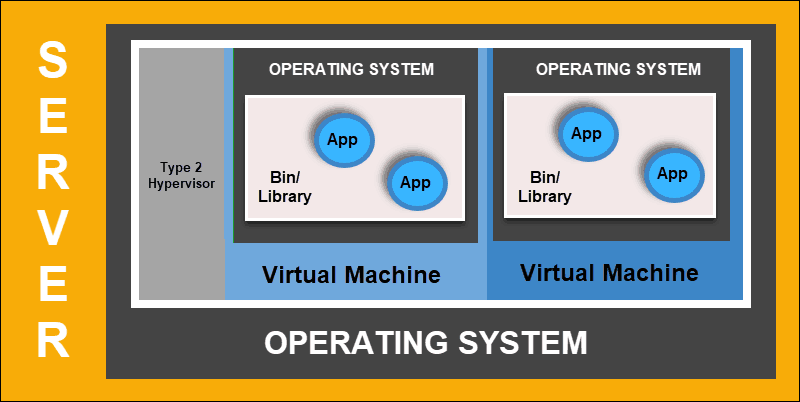
Conceptos básicos

1. Despliegue tradicional



Un servidor tiene un SO dentro del cual se ejecutan varias aplicaciones

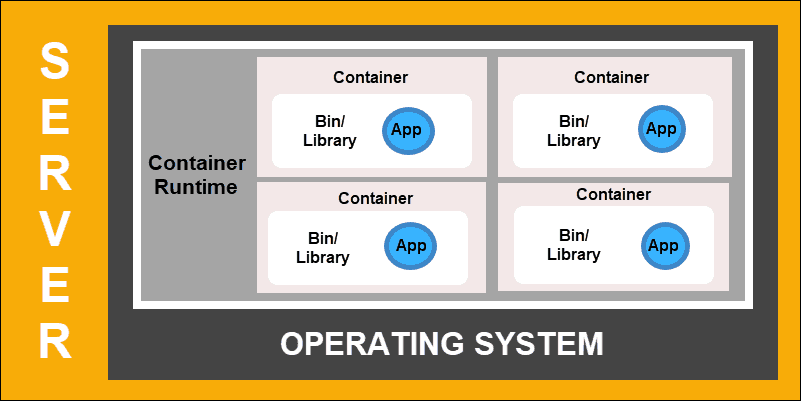
1. Despliegue virtualizado



Un servidor tiene un SO dentro del cual se ejecuta una aplicación de virtualización que ejecuta una o varias máquinas virtuales que a su vez consisten en diferentes SOs ejecutando distintas aplicaciones.

La ventajas que esto representa son entre otras: facilidad en la administración, ahorro de recursos (memoria, discos, etc)

1. Despliegue en contenedores



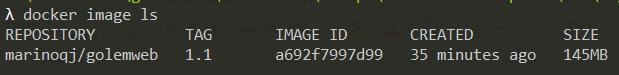
Los contenedores son un paso más en la virtualización pues en lugar de ejecutarse un SO por cada máquina virtual, se ejecuta el mínimo kernel necesario para que corra una aplicación determinada. Esto se produce dentro de un entorno de ejecución que ejecuta varios contenedores al mismo tiempo. Esto supone una reducción todavía mayor de los recursos necesarios, al mismo tiempo que se incrementan las facilidades de administración (aunque al mismo tiempo surgen otras necesidades diferentes)

Docker básico: utilizando contenedores

1. El conjunto mínimo necesario para que corra una aplicación se llama imagen

Para ver las imágenes que tenemos en nuestro repositorio local

docker image ls



1. Podemos utilizar imágenes ya creadas, o crear nuestras propias imágenes. Para ello se utilizar un fichero que se llama Dockerfile. Un ejemplo sería:

FROM httpd:2.4

COPY ./golemdr/ /usr/local/apache2/htdocs/

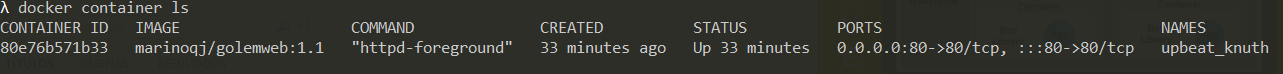
Esto básicamente quiere decir: baja la imagen ya creada del servidor apache (versión 2.4) y le copias en el directorio correpondiente el contenido que va a forma el site.

Para crear la imagen se utiliza el comando:

docker build -t marinoqj/golemweb:1.1 . (ojo al punto final que sino no funciona)

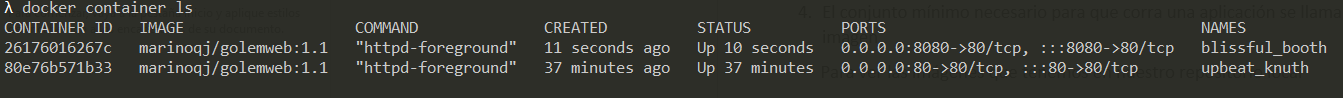
1. Para poder utilizar una imagen hay que “arrancarla”.
2. Un contenedor es una instancia en ejecución de una imagen

docker container ls



1. Se pueden tener corriendo varios contenedores de la misma imagen siempre que no colisionen en puertos, nombre o cualquier otra cosa. Para crear un contenedor

docker run -d -p 8080:80 marinoqj/golemweb:1.1



Ahora tenemos dos contenedores de la misma imagen corriendo en distintos puertos.

1. El container runtime siempre ejecuta imágenes que tiene en el repositorio local (podríamos llamarlo entorno de desarrollo). Cuando no existe una imagen en el repositorio local se baja del remoto. Para compartir y distribuir imágenes se utiliza un repositorio remoto.
2. Para subir imágenes al repositorio remoto utilizamos

docker login (normalmente el repositorio requiere autenticación)

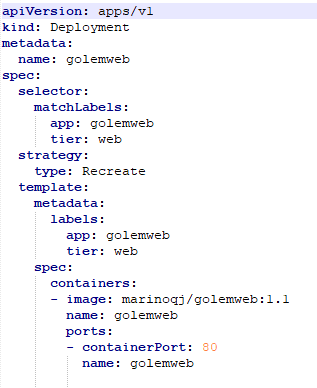
docker push marinoqj/golemweb:1.1

Kubernetes (k8s) básico: gestionando contenedores

1. Kubernetes es una capa que está por encima de los contenedores y que facilita su gestión en aspectos tales como el networking, redundancia, tolerancia a fallos, etc
2. Tiene varíos objetos propios que “encapsulan” los contenedores. Entre ellos:

* Deployment: es la entidad que configura los contenedores.
* Pod: es una instancia de un deployment
* Service: es un objeto que permite que los pods se comuniquen entre si mediante la gestión del networking

1. Para generar un deployment se utiliza un fichero de configuración escrito en yaml



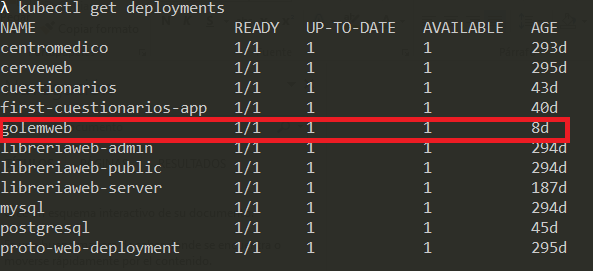
En él se especifica la imagen que vamos a utilizar para crear los contenedores dentro de k8s, entre otras cosas

1. Para desplegar un deployment y que se genere el pod correspondiente se ejecuta:

kubectl apply -f golemweb-deployment.yaml

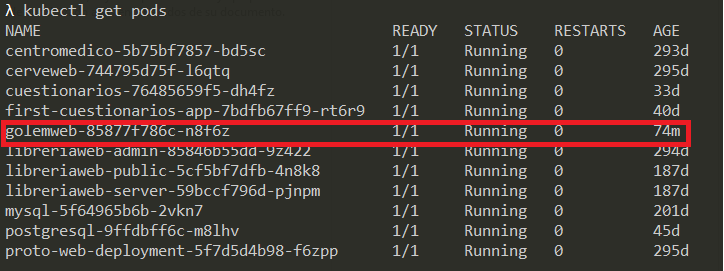
Para ver el deployment que se genera:

kubectl get deployments

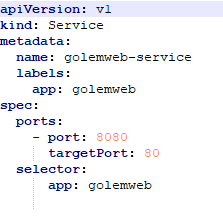


1. Mientras que para ver el pod

kubectl get pods



1. Para que desde fuera del nodo de kubernetes se pueda acceder al pod (o desde dentro por otros pods) hay que crear un servicio. También se configura con un yaml

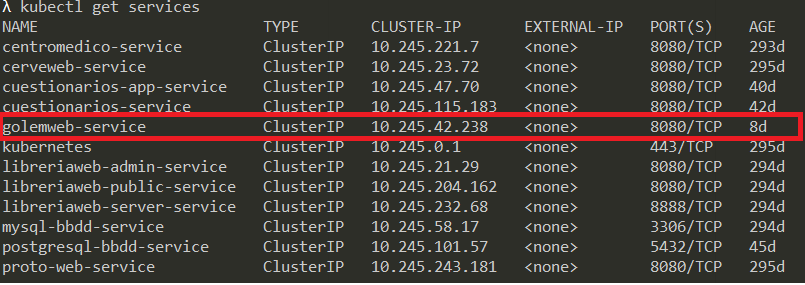


1. Y del mismo modo que el deployment hay que aplicar la configuración de dicho fichero

kubectl apply -f golemweb-service.yaml

1. Para ver el servicio creado

kubectl get services



Y de una forma muy muy resumida, este sería el proceso para desplegar un contenerdor en un nodo de k8s

 Y esto es, to esto es to, esto es todo amigos!!!