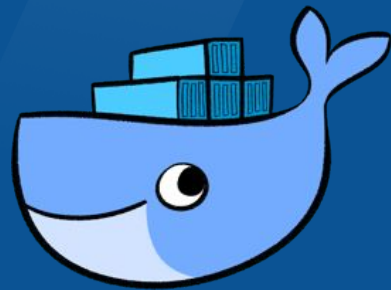


Kubernetes

101

Camilo Rivera

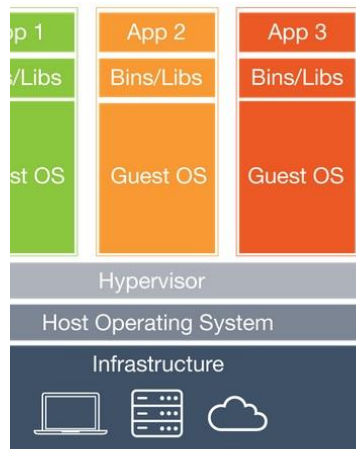


AGENDA

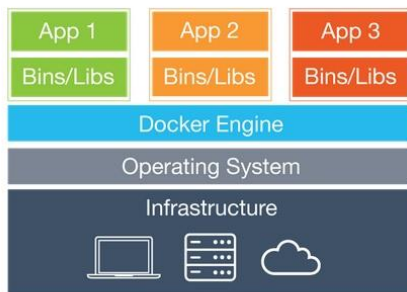
- VM vs Containers
- Images vs container
- Run/stop/remove containers
- What Happens When We Run
- Kubernetes
- Pods
- Deployments
- Services



VM vs Containers



Virtual Machines

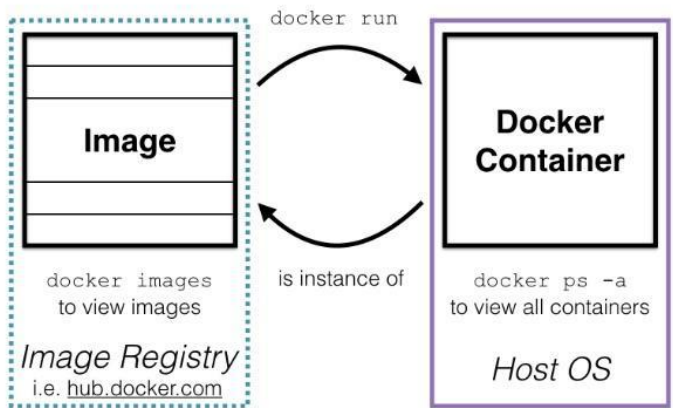


Containers



Containers are much more portable and efficient, even though they have similar resource isolation

Images vs Containers



- An image is the application we want to run
- A container is an instance of that image running as a process
- You can have many containers running of the same image
- Image registry : hub.docker.com

Run / Stop / Remove containers



Run

- `docker container run --publish 80:80 nginx`
- `docker container run --publish 80:80 --detach nginx`



Stop

`docker container stop <container
ids>`



Remove

`docker container rm < container
ids>`

What Happens When We Run

- Looks for that image locally in image cache, doesn't find anything ?
- Then Looks in remote image repository (defaults to docker Hub)
- Download the latest version (nginx:latest)
- Creates new container based on that image and prepares to start
- Give it an virtual IP on a private network inside docker engine
- Opens up port 80 on host and forwards to port 80 in container
- Starts container by using the CMD in the image Dockerfile

Kubernetes

Is an open-source system for automating deployment, scaling, and management of containerized applications.



Pods

A Pod is the basic building block of Kubernetes—the smallest and simplest unit in the Kubernetes object model that you create or deploy. A Pod represents a running process on your cluster.



Deployments

A Deployment Controller can create and manage multiple Pods for you, handling replication and rollout and providing self-healing capabilities at cluster scope. For example, if a Node fails, the Controller might automatically replace the Pod by scheduling an identical replacement on a different Node.



Services

A Kubernetes Service is an abstraction which defines a logical set of Pods and a policy by which to access them - sometimes called a micro-service. The set of Pods targeted by a Service is (usually) determined by a Label Selector (see below for why you might want a Service without a selector).



K8 Cheat Sheet

- `kubectl apply -f deployments.yaml`
- `kubectl logs <Pods name>`
- `kubectl get (nodes/services/deployments/pods/namespaces)`
- `kubectl describe (nodes/services/deployments/pods/namespaces) <resource name>`
- `kubectl port-forward --namespace default svc/my-service 3333:80`

