

Minikube

Minikube is a tool that makes it **easy to run Kubernetes locally**. **Minikube runs a single-node Kubernetes cluster inside a VM** (on Virtualbox for example) on your laptop for users looking to try out Kubernetes or develop with it day-to-day. Internally, minikube runs a single **Go** binary (named *localkube*), which runs all the kubernetes' components. The result is a local kubernetes endpoint that you can use with the *kubect!* client.

- Minikube supports **kubernetes features** such as DNS, Dashboards, CNI, NodePorts, ConfigMaps and Secrets and so on.
- The minikube VM runs Docker, but supports also **rkt** container engine.
- **Multiple version of kubernetes are supported**, from kubernetes 1.3.0 to 1.7.5.
- Possible to reuse the **minikube's built-in Docker daemon**; as this means you don't have to build a docker registry on your host machine and push the image into it – you can just build inside the same docker daemon as minikube which speeds up local experiments.
- Supports **persistent volumes** and **host folder mounting**.
- Kubernetes environment configurable.

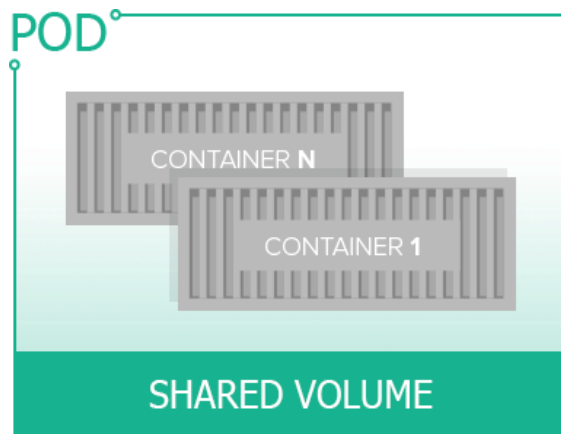
Pods

A pod is the smallest deployable entity in Kubernetes and it is important to understand the main principles around pods

- Containers **always** run inside a Pod.
- A pod usually has 1 container but can have more.
- Containers in the same pod are **guaranteed** to be located on the same machine and share resources

Why you might want to run more than one container in a pod

- If you have containers that are tightly coupled and share a lot of I/O then you may want them on the same machine and allow them to share resources.



Services

- Services allow you to expose your microservices (that run in pods) both internally and externally between the other running pods.
- The service object exposes a consistent IP and port for the microservice. Any call to that IP will be routed to one of the microservice's running Pods.
- The benefit of services is that while each and every pod is assigned an IP, Pods can be shut down (due to failure or rollout of new changes) and the pod's ip will no longer be available. Meanwhile the service IP and port will not change and always accessible to the other pods in your cluster.
- By default, the DNS service of Kubernetes exposes the service name to all pods so they can refer to it simply by its name.

Replication Controller is one of the key features of Kubernetes, which is responsible for managing the pod lifecycle. It is responsible for making sure that the specified number of pod replicas are running at any point of time.