Kubernetes Minikube Installation.

* Some Important Points.
* What is Kubernets objects.
* It represents as JSON or YAML files.
* You create these and then push them to the Kubernetes API with kubectl
* Kubernetes uses objects to represent the state of your cluster.
* Which containerized applications are running
* The policies around how those application behave, such as restart policy, upgrade and fault tolerance.
* Once you create the Object, the Kubernetes system will constantly work to ensure that objects exists and maintain’s clustere’s desired state.
* Every Kubernetes object include two nested fields that govern the object config the object spec and the object status.
* The spec which we provide describe your desired state for the object, the characteristics that you want the object to have.
* The status describes the actual state of the object and is supplied and updated by the Kubernetes system.
* All object are identified by a unique name and a UID.
* The basic Kubernetes objects include :
* Pod
* Service
* Volume
* Namespace
* Replicaset
* Secrets
* ConfigMaps
* Deployments
* Jobs
* Demonsets.
* Relationship between these object.
* Pod manages containers.
* Replicaset manage pods
* Service expose pod process to the outside world.
* Configmaps and secrets helps you configure pods.
* State of the Object.
* Replica
* Image
* Name
* Port
* Volume
* Startup
* Detached (default)
* Kubernetes Object Management
* The Kubernetes command line tool supports several different ways to create and manage Kubernetes Object

Management Technique Operates on Recommended Environment

Imperative command live object development project

Declarative object configuration individual files(yalm / json) production

* Declarative is about describing what you are trying to achieve without instructing how to do it.
* Imperative , explicitly tells “how to accomplish it”.
* Fundamentals of Pods.
* When a pod gets created it is scheduled to run on a node in your cluster.
* The pod remains on that node until the process is terminated, the pod object is deleted, the pod is evicted for lack of resources or the node fails.
* If a pod is scheduled to a node that fails or if the scheduling operation itself fails the pod is deleted.
* If a node dies the pod scheduled to that node are scheduled deletion after a timeout period
* A given pod (UID) is not rescheduled to a new node, instead it will be replaced by an identical pod, with even the same name if desired , but with a new UID.
* Volume in a pod will exists as long as that pod (with that UID) exists if that pod is deleted for any reason, volume is also destroyed and created as new on new pod
* A container can create and manage multiple pods , handling replication rollout and providing self healing capabilities.
* Kubernetes configuration.
* All-in one single node installation
* With all in one, all the master and worker components are installed on a single node this is very useful for learning, development, and testing . This type should not be used in production . Minikube is one such example, and we are going to explore it .
* Single Node etcd, single master and multi worker installation
* In this setup, we have a single master node, which also runs a single node etcd instance. Multiple worker nodes are connected to the master.
* Single node etcd, multi master and multi worker installation.
* In this setup we have multiple master nodes which works in on High Availability mode, but we have a single node etcd instance. Multiple worker node are connected to the master node.