**Pod:** The basic building block in Kubernetes. The name is the same idea as a pod of whales, one of more containers– or better said, a running process on the cluster —  running on a given system which can be scaled up or down

**Service:** A consistent network access point for a pod. What this means is that regardless of the underlying containers’ state – restarts, etc, a service will always provide access to the given pod/application.

**ReplicaSet:** Ensures a given number of pods are running in the cluster at a given time

**Deployment:** Suited for more ephemeral applications like web applications, anything that doesn’t have persistent qualities. It Provides declarative updates for Pods and ReplicaSets by describing a desired state the deployment in turn acts on

**Stateful Set:** Suited for stateful applications like databases or even monitoring systems like Prometheus. Similar to a deployment in that it manages the deployment and scaling of a set of pods, but differs in that it ensures ordinal (things happen in a given sequence and are numbered) and uniqueness of pods. Overall, it  ensures that all resources in the stateful set have a sticky identity across restarts.

**Labels:** When creating Kubernetes objects, one can give any number of labels to each object which are discoverable across the system. For instance, one could create a database with the label “customer\_database” and another application pod could reference it by simply using that name “customer\_database” in the selector value of the manifest file.

**Namespaces:** Virtual clusters. Provides a means to separate applications in the cluster.

**Primary Kubernetes components**

The master components are where all the work is done to run the cluster; the minion is where containers are run. These components are all interchangeable and can be run in various configurations.

**Kubernetes API Server:** This is the main point of interaction with the Kubernetes control plane providing the Kubernetes API. Its backing store is Etcd

**Etcd:** A key value database providing where Kubernetes stores information about the cluster.

**Kube Controller Manager:** A master component that runs controllers for Kubernetes

**Kube Scheduler**: Schedules pods on machines intelligently based on numerous factors such as node health, resources, affinity, etc

**Kube DNS:** The Kubernetes internal DNS server. Any pod or service in Kubernetes ends up with an DNS entry in KubeDNS. This makes it possible to refer to refer to pods and services by name within Kubernetes.

**Minion components**

**Kubelet:** Runs containers on each node

**Kube Proxy:** Provides networking (iptables) for pods and services

And for all of these, one uses the kubectl utility or interacts programmatically using the Kubernetes API with the cluster.