1. Kubernetes Setup
   1. MiniKube
   2. Kubeadm –
      1. <https://kubernetes.io/docs/setup/independent/install-kubeadm/>
      2. If you are following along with your own setup, make sure to use the below command for setting up **Networking with Flannel**. The command in the Kubernetes Documentation web site seems to be having an issue.

kubectl create -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

* 1. KOPS
  2. Google Cloud Platform
  3. Play-with-k8s.com

# Kubernetes commands

* Kubectl version - To get kubectl version.
* Kubectl cluster-info - To get info about cluster
* Kubectl get nodes – To get info on nodes
* Kubectl get deployments – To get info on how many deployments are available.

1. Create pods
   1. Each pod has one container (recommended)
   2. Create pod with image from docker hub

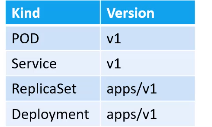
# Kubernetes commands

* Kubectl get pods
* Kubectl describe pods
* Kubectl run ngnix – deploys pod with container
* Kubectl run ngnix --image ngnix - Pulls image from dockerhub and creates container
  1. Pods with YAML:

Pod- definition file : YAML file description:

1.apiversion

2. kind



3.metadata

* Name:
* Labels:
  + App: myapp1
  + Type: front-end

4.**spec: This part tells about the type of object we are creating**

* Containers:
  + Name: nginx-container
  + Image: nginx

# Kubernetes commands

* + Kubectl get pods
  + Kubectl describe pod <pod-name>
  1. YAMLeditor**: use pycharm YAML editor.**

# **Jetbrains.com/pycharm**

1. Replication Controllers and Replicasets

Replication controller is old way, Replication is currently used.

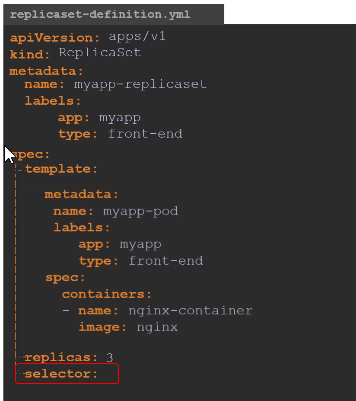
Replicaset-definition.yml

Template and replicas need to be under spec



# Kubernetes commands

* + Kubectl create –f replication\_definition.yml
  + Kubectl get replicationcontroller
  + Kubectl get replicaset
  + Kubectl delete replicaset <name> ---this deletes all underlying pods
  + Kubectl replace –f replicaset-definition.yml
  + Kubectl scale –replicas=6 –f replicaset-definition.yml
  + Kubectl scale –replicas=6 replicaset <replicaset-name>
  + Kubectl get pods – no. of pods running by the replication



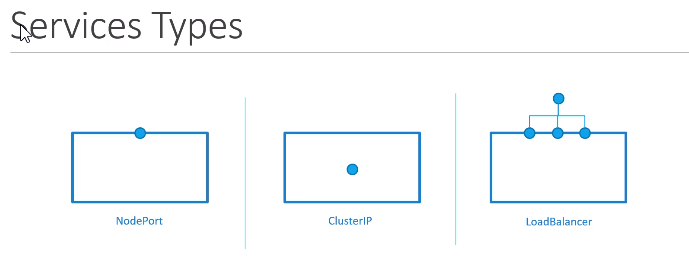
selector: This is the main difference between replicaset and replicaset controller.

This section will match labels to create new pod.

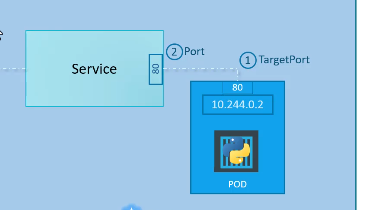
1. Networking between Pods:
   1. Each pod will get its won IP adress:
   2. All containers/PODs can communicate to one another without NAT
   3. All nodes can communicate with all containers and vice-versa without NAT
   4. Networks available are :
      1. Cisco
      2. **Flannel - used more**
      3. VMWare- NSX
      4. Cilium
      5. calico

* Flannel creates a IP address to access all the pods. Flannel will create a routing for all pods to communicate

1. Services:
   1. Enable connectivity between pods/ frontend to backend and web connectivity(external communication) also.
   2. Service is object just like pod/replicaset
      1. Nodeport service: Maps port on Node to Port on POD
      2. CusterIP: virtual IP is created
      3. Loadbalancer



* 1. Nodeport service:



* 1. Service-definition.yml

apiversion: v1

kind: Service

metadata:

name: myapp-service

spec:

Type: Nodeport

ports:

-targetPort: 80

port: 80

nodeport: 30008 (Range:30000-32767)

selector:

\*\*pull labels from pod-definition.yml to indicate the pods to be mapped.

app: myapp

type: front-end