**Docs:**

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.18/#deployment-v1-apps>

**NAMESPACE:**

**root@ip-172-31-41-82:~# kubectl get namespace**

**NAME STATUS AGE**

**default Active 7d**

**kube-public Active 7d**

**kube-system Active 7d**

**root@ip-172-31-41-82:~# kubectl create namespace test**

**namespace/test created**

**root@ip-172-31-41-82:~# kubectl get pods --all-namespaces**

**NAMESPACE NAME READY STATUS RESTARTS AGE**

**damok8s firstpod 1/1 Running 2 6d23h**

**default deploy1-5f744cdc9b-67dfc 1/1 Running 2 7d**

**kube-system coredns-bb49df795-c8qxw 1/1 Running 2 7d**

**kube-system coredns-bb49df795-pt2g6 1/1 Running 2 7d**

**kube-system etcd-ip-172-31-41-82 1/1 Running 2 7d**

**kube-system kube-apiserver-ip-172-31-41-82 1/1 Running 2 7d**

**kube-system kube-controller-manager-ip-172-31-41-82 1/1 Running 2 7d**

**kube-system kube-flannel-ds-amd64-cv99q 1/1 Running 2 7d**

**kube-system kube-flannel-ds-amd64-xfdww 1/1 Running 2 7d**

**kube-system kube-proxy-b8pnm 1/1 Running 2 7d**

**kube-system kube-proxy-jhdcb 1/1 Running 2 7d**

**kube-system kube-scheduler-ip-172-31-41-82 1/1 Running 2 7d**

**root@ip-172-31-41-82:~# kubectl get pods --all-namespaces -o wide**

**NAMESPACE NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE**

**damok8s firstpod 1/1 Running 2 6d23h 10.244.1.15 ip-172-31-4-141 <none>**

**default deploy1-5f744cdc9b-67dfc 1/1 Running 2 7d 10.244.1.14 ip-172-31-4-141 <none>**

**kube-system coredns-bb49df795-c8qxw 1/1 Running 2 7d 10.244.1.13 ip-172-31-4-141 <none>**

**kube-system coredns-bb49df795-pt2g6 1/1 Running 2 7d 10.244.1.16 ip-172-31-4-141 <none>**

**kube-system etcd-ip-172-31-41-82 1/1 Running 2 7d 172.31.41.82 ip-172-31-41-82 <none>**

**kube-system kube-apiserver-ip-172-31-41-82 1/1 Running 2 7d 172.31.41.82 ip-172-31-41-82 <none>**

**kube-system kube-controller-manager-ip-172-31-41-82 1/1 Running 2 7d 172.31.41.82 ip-172-31-41-82 <none>**

**kube-system kube-flannel-ds-amd64-cv99q 1/1 Running 2 7d 172.31.4.141 ip-172-31-4-141 <none>**

**kube-system kube-flannel-ds-amd64-xfdww 1/1 Running 2 7d 172.31.41.82 ip-172-31-41-82 <none>**

**kube-system kube-proxy-b8pnm 1/1 Running 2 7d 172.31.4.141 ip-172-31-4-141 <none>**

**kube-system kube-proxy-jhdcb 1/1 Running 2 7d 172.31.41.82 ip-172-31-41-82 <none>**

**kube-system kube-scheduler-ip-172-31-41-82 1/1 Running 2 7d 172.31.41.82 ip-172-31-41-82 <none>**

**POD**:

Pods can be created in two ways:

1. Declarative method
2. Iterative method

Kubectl create deployment pod2 –image=nginx

apiVersion: v1

kind: Pod

metadata:

name: memory-demo-2

namespace: damok8s

spec:

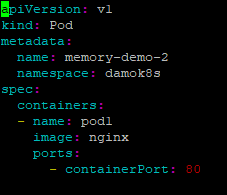
containers:

- name: pod1

image: nginx

ports:

- containerPort: 80



By default pod is not exposed to outer world. You can check in browser with port 80.

**LOGIN TO POD**

kubectl exec -it secondpod sh

# curl http://localhost:80

<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>

<style>

PODS CAN BE EXPOSED BY SERVICES

REPLICASET:

controllers/frontend.yaml

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: frontend

labels:

app: guestbook

tier: frontend

spec:

# modify replicas according to your case

replicas: 3

selector:

matchLabels:

tier: frontend

template:

metadata:

labels:

tier: frontend

spec:

containers:

- name: php-redis

image: gcr.io/google\_samples/gb-frontend:v3

root@ip-172-31-41-82:~# kubectl get rs

NAME DESIRED CURRENT READY AGE

deploy1-5f744cdc9b 1 1 1 8d

replicaset-example 1 1 1 3m56s

**DEPLOYMENTS**:

Mainly for rolling update suppose if you have changed the updated image . Once new pods started to respond it will automatically delete the old pods. Deployment will also control replicasets. So we don’t want replicasets.

controllers/nginx-deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.14.2

ports:

- containerPort: 80

Just give wrong image which doesn’t exist. Kubernetes will run only with already existing image not with the new image as image doesn’t exist at all

kubectl set image deploy deployment-example nginx=nginx

kubectl scale deployment deployment-example --replicas=2

**Daemnoset:**

**apiVersion: apps/v1**

**kind: DaemonSet**

**metadata:**

**name: mynode-exporter**

**labels:**

**app: mynode-exporter**

**spec:**

**selector:**

**matchLabels:**

**app: mynode-exporter**

**template:**

**metadata:**

**labels:**

**app: mynode-exporter**

**spec:**

**containers:**

**- name: mynode-exporter**

**image: prom/node-exporter:v0.18.1**

**ports:**

**- containerPort: 9100**

**hostPort: 9100**

**protocol: TCP**