- Docs »
- Kubernetes Fundamentals »
- Launching Pods

•

Deploying Pods

Life of a pod

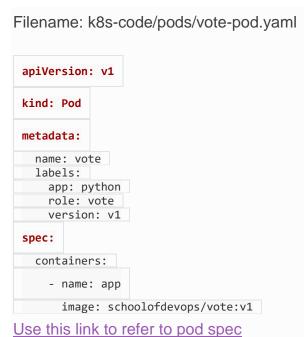
- Pending : in progress
- Running
- · Succeeded : successfully exited
- Failed
- Unknown

Resource Configs

Each entity created with kubernetes is a resource including pod, service, deployments, replication controller etc. Resources can be defined as YAML or JSON. Here is the syntax to create a YAML specification.

AKMS => Resource Configs Specs apiVersion: v1 kind: metadata: spec: Spec Schema: https://kubernetes.io/docs/user-guide/pods/multi-container/ To list supported version of apis kubectl api-versions **Writing Pod Spec** Lets now create the Pod config by adding the kind and specs to schme given in the file vote-pod.yaml as follows. Filename: k8s-code/pods/vote-pod.yaml apiVersion: kind: Pod metadata: spec:

Lets edit this and add the pod specs



Launching and operating a Pod

To launch a monitoring screen to see whats being launched, use the following command in a new terminal window where kubectl is configured.

```
watch -n 1 kubectl get pods,deploy,rs,svc
kubectl Syntax:
```

```
kubectl apply --help
 kubectl apply -f FILE
To Launch pod using configs above,
 kubectl apply -f vote-pod.yaml
To view pods
 kubectl get pods
 kubectl get po -o wide
 kubectl get pods vote
To get detailed info
 kubectl describe pods vote
[Output:]
                vote
 Name:
                default
 Namespace:
                kube-3/192.168.0.80
 Node:
```

Start Time:

Tue, 07 Feb 2017 16:16:40 +0000

```
app=voting
Labels:
               Running
Status:
IP:
               10.40.0.2
Controllers:
                <none>
Containers:
 vote:
   Container ID:
                       docker://48304b35b9457d627b341e424228a725d05c2ed97cc9970bbff32a1b365d9a5d
                       schoolofdevops/vote:latest
   Image:
                       docker-pullable://schoolofdevops/vote@sha256:3d89bfc1993d4630a58b831a6d44ef73d2be76a7862153e02e7a7c0cf2936731
    Image ID:
    Port:
                       80/TCP
    State:
                       Running
                       Tue, 07 Feb 2017 16:16:52 +0000
     Started:
    Ready:
                       True
    Restart Count:
                       0
   Volume Mounts:
     /var/run/secrets/kubernetes.io/serviceaccount from default-token-2n6j1 (ro)
    Environment Variables:
                                <none>
Conditions:
 Type
               Status
 Initialized
               True
 Ready
               True
 PodScheduled True
Volumes:
 default-token-2n6j1:
    Type:
               Secret (a volume populated by a Secret)
```

SecretNa	ame: default-toker	n-2n6j1					
QoS Class:	BestEffort						
Tolerations	: <none></none>						
Events:							
FirstSeen	LastSeen	Count	From	SubObjectPath	Туре	Reason	Message
21s	21s	1	{default-scheduler }		Normal	Scheduled	Successfully
assigned vo	te to kube-3						
20s	20s	1	{kubelet kube-3}	<pre>spec.containers{vote}</pre>	Normal	Pulling	pulling image
"schoolofde	ops/vote:latest"						
10s	10s	1	{kubelet kube-3}	<pre>spec.containers{vote}</pre>	Normal	Pulled	Successfully
pulled image	e "schoolofdevops,	/vote:lates	st"				
9s	9s	1	{kubelet kube-3}	<pre>spec.containers{vote}</pre>	Normal	Created	Created contained
with docker	id 48304b35b945;	Security:	[seccomp=unconfined]				
9s	9s	1	{kubelet kube-3}	<pre>spec.containers{vote}</pre>	Normal	Started	Started contained
with docker	id 48304b35b945						
ommands	to operate the p	od					
kubectl logs	s vote						

kubectl exec -it vote sh

Inside the container in a pod



Attach a Volume to the Pod

Lets create a pod for database and attach a volume to it. To achieve this we will need to

- create a volumes definition
- attach volume to container using VolumeMounts property

Local host volumes are of two types:

- * emptyDir
- * hostPath

We will pick hostPath. Refer to this doc to read more about hostPath.

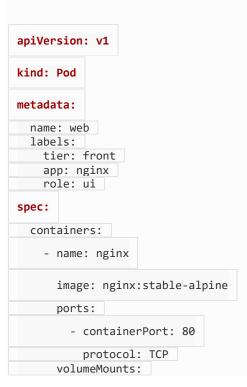
```
File: db-pod.yaml
 apiVersion: v1
 kind: Pod
 metadata:
   name: db
   labels:
     app: postgres
     role: database
     tier: back
 spec:
   containers:
     - name: db
       image: postgres:9.4
       ports:
         - containerPort: 5432
       volumeMounts:
       - name: db-data
         mountPath: /var/lib/postgresql/data
   volumes:
   - name: db-data
     hostPath:
       path: /var/lib/pgdata
      type: DirectoryOrCreate
To create this pod,
 kubectl apply -f db-pod.yaml
```

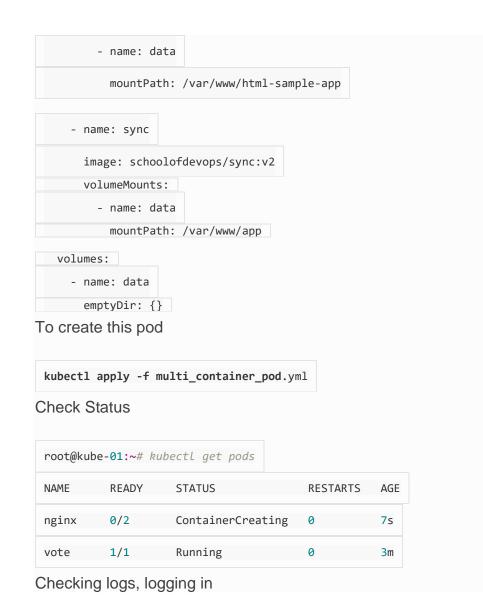


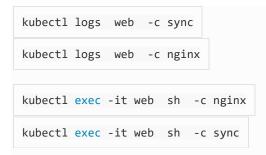
Exercise: Examine /var/lib/pgdata on the systems to check if the directory is been created and if the data is present.

Creating Multi Container Pods

file: multi_container_pod.yml







Observe whats common and whats isolated in two containers running inside the same pod using the following commands,

shared

hostname ifconfig

isolated



Exercise

Create a pod definition for redis and deploy.

Reading List

- PodSpec: https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.9/#pod-v1-core
- Managing Volumes with Kubernetes: https://kubernetes.io/docs/concepts/storage/volumes/
- Node Selectors, Affinity: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/