Kubernetes Resources

Pod

The smallest deployable unit in K8s. A Pod is a group of 1+ containers scheduled on the same node with shared storage and networking.

pod.yml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: devops
  labels:
    env: prod
spec:
  # Volumes for the pod. Different
  # types available (here: emptyDir)
  volumes:
    - name: cache-volume
      emptyDir: {}
  containers:
    - image: tfogo/devops
      name: devops
      # Where to mount volumes
      volumeMounts:
        - mountPath: /cache
          name: cache-volume
      # Which ports to expose
      ports:
        - containerPort: 80
          name: http
          protocol: TCP
      resources:
        # Request minimum resources
        requests:
          cpu: 500m
          memory: 128Mi
      # How K8s tests pod's liveness
      livenessProbe:
        httpGet:
          path: /healthy
          port: 80
        initialDelaySeconds: 5
        timeoutSeconds: 1
        periodSeconds: 10
        failureThreshold: 3
```

Deployment

Allows for declarative updates.

deployment.yml

```yaml
apiVersion: v1
kind: Deployment
metadata:
  name: devops
  labels:
    env: prod
spec:
  replicas: 10
  # Update strategy used by this deployment
  strategy:
    rollingUpdate:
      # Temporarily increased limit for pod
      # replicas during an update (here 12).
      maxSurge: 20%
      # Number of pods that can be
      # unavailable during an update
      maxUnavailable: 0
  selector:
    # Matching pods are in the deployment
    matchLabels:
      app: devops
  template:
    metadata:
      name: devops
      # Make sure this matches matchLabels
      labels:
        app: devops
    spec:
      # Define a configMap as a volume
      volumes:
        - name: config-volume
          configMap:
            name: my-config
      containers:
        - image: tfogo/devops
          name: devops
          ports:
            - containerPort: 80
          # Mount configMap volume
          volumeMounts:
            - name: config-volume
              mountPath: /config
          # Set env variable from a secret
          env:
            - name: SECRET_TOKEN
              valueFrom:
                secretKeyRef:
                  name: mysecret
                  key: token
```

Service

An abstraction which defines a way to access a set of Pods.

service.yml

```yaml
kind: Service
apiVersion: v1
metadata:
  name: devops
spec:
  selector:
    # Pods with this label will be routed
    # to by this service
    app: devops
  ports:
  - protocol: TCP
    # Port exposed on the Service Cluster IP
    port: 80
    # Exposed port on the Pod
    targetPort: 80
  # Service type can be ClusterIP (default),
  # NodePort, LoadBalancer, or ExternalName
  type: NodePort
```

Other Resources

CONFIGMAP

Allows you to decouple configuration artifacts and image content.

CRONJOB

Creates a job that runs at a regular interval.

DAEMONSET

Ensures that some Nodes run a copy of a Pod.

JOB

Creates one or more Pods and ensures a number of them terminate successfully.

REPLICASET

Ensures a specific # of Pod replicas are run at any time. Should be set up as a Deployment.

STATEFULSET

Manages deployment and scaling of a set of Pods. Provides ordering and uniqueness of Pods.