1

Outline

- Introduction to Kubernetes
- Kubernetes Architecture
- Start a Single node cluster locally with Docker Desktop for Windows
- Using the Kubernetes dashboard
- Basic Commands of Kubectl & Docker

2

Introduction to Kubernetes

3

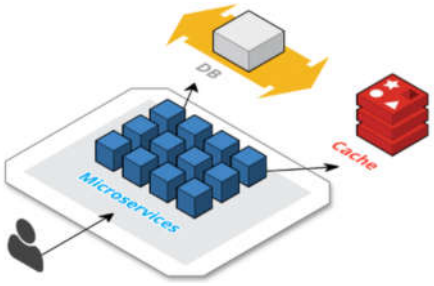
What is Kubernetes ?

- **Kubernetes is an open source system for managing containerized applications across multiple hosts.**
  - It provides basic mechanisms for deployment, maintenance, and scaling of applications.
- **Kubernetes builds upon a decade and a half of experience at Google running production workloads at scale using a system called Borg.**
  - Donated to the Cloud Native Computing Foundation (CNCF).
  - Hit the first production-grade version v1.0.1 in **July 2015**.

4

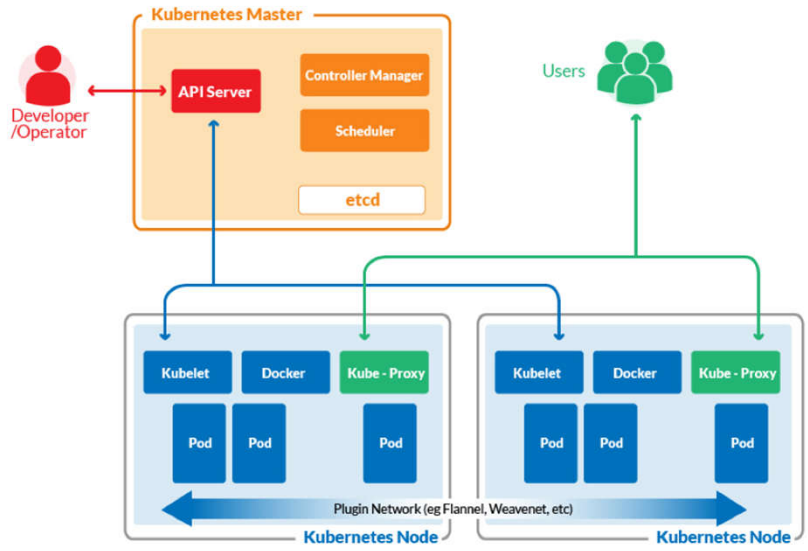
What problems does Kubernetes solve ?

- **The trend from monoliths to the Microservices increased the usage of containers**
  - Managing hundreds of containers across different plaforms using shell scipts and self-made tools becomes complex and even impossible.
- **Kubernetes provides a way to manage hundreds of containers. It provides**
  - **Code deployment with** no down times
  - **High availability** dynamic scheduling of workloads
  - **Scalability** (or high performance)
  - **Disaster recovery** (backup and restore)



5

Kubernetes Architecture



6

Kubernetes Installation Options

- **Single Node Local Cluster**
  - Use Case : Developer sandbox
  - Alternatives :
    - DockerDesktop for Windws and Moc
    - Minikube (Locally or via Katacoda or Play with Kubernetes)
- **Multi-Cluster Node**
  - Use Case : Customized Kubernetes installation
  - Alternatives :
    - Kops installer. Provides provisiong and Orchestration itself
    - Kubespray installer. Gives flexiblity but requires to know Ansible
    - Kubeadm installer. Kubeadm sets up a minimal viable cluster.
- **Kubernetes as A Service**
  - Use case: Hybrid and multi-cloud installations
  - Alternatives:
    - AWS
    - Azure
    - GCP

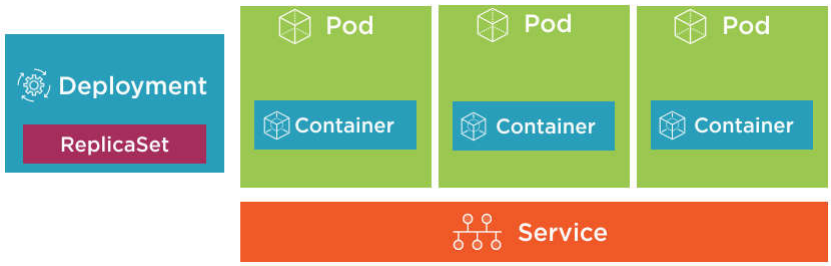
7

Kubernetes Concepts

8

Workload Concepts, Recap

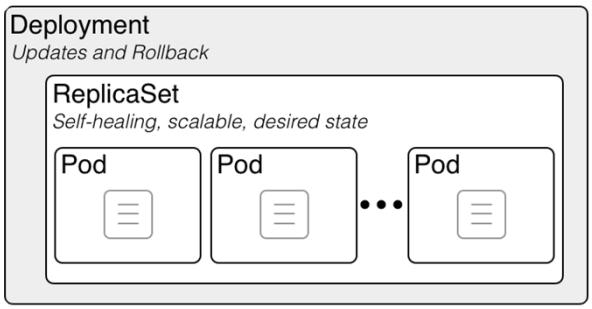
- A pod is the smallest unit of work or management resource within Kubernetes.
- ReplicaSet ensures that a specified number of Pod replicas are running at any given time.
- Deployment is used to change the current state to the desired state.
- Services provide a method of exposing and consuming Pod network accessible resources.



9

Deployment Objects

- **Pods** are the basic, atomically deployable unit
- ReplicaSet are responsible for achieving and reconciling **the desired state** of an application service.
- The **Deployment** augments a ReplicaSet by providing rolling update and rollback functionality on top of it.



10

Deployment Manifest Example

- **Required fields**
  - **apiVersion** - Which version of the Kubernetes API you're using to create this object
  - **kind** - What kind of object you want to create
  - **metadata** - Data that helps uniquely identify the object, including a name string, UID, and optional namespace
  - **spec** - What state you desire for the object. The precise format of the object spec is different for every Kubernetes object

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:                    # Replicaset
    matchLabels:
      app: nginx
  replicas: 3
  template:                    # Pod Template
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.19.8
        ports:
        - containerPort: 80
```


11

Service Types

1. **ClusterIP (default)**
   - **A virtual IP address is allocated for the service**
   - **This IP address is reachable only from within the cluster (nodes and pods)**
   - **Perfect for internal communication, within the cluster**
2. **NodePort**
   - **NodePort services extend the ClusterIP service.**
   - **Exposes a port on every node's IP.**
   - **Port can either be statically defined, or dynamically taken from a range between 30000-32767.**
3. **LoadBalancer**
   - **LoadBalancer services extend NodePort**
   - **Works in conjunction with an external system to map a cluster external IP to the exposed service (typically a cloud load balancer, e.g. ELB on AWS, GLB on GCE ...)**

12

Core Concepts

- **Namespace**
  - A logical cluster or environment. Primary method of dividing a cluster or scoping access.
- **Label**
  - Key-value pairs that are used to identify, describe and group together related sets of objects. Labels have a strict syntax and available character set.
- **Selector**
  - Selectors use labels to filter or select objects.

13

Labels and Selectors

Deployment configuration:

```
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:1.7.9
        ports:
        - containerPort: 80
```

Deployment

ReplicaSet

Pod

Service configuration:

```
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80
    targetPort: 9376
```


14

Kubectl Cheat Sheet

https://kubernetes.io/docs/reference/kubectl/cheatsheet/

15

Kubectl command examples

- **Getting Information about Cluster**
  - `kubectl version` Prints the client and server versions.
  - `kubectl cluster-info` Prints information about the control plane and add-ons.
  - `kubectl config get-contexts` Displays the list of cluster contexts
- **Getting information about resources**
  - `kubectl get nodes/pods/deployements/secrets` Prints information about resources
  - `kubectl describe nodes/pods/deployements/secrets` Prints detailed information about resources
- **Creating/Updating a Resource from Manifest**
  - `kubectl create/apply -f my-nginx-deployment.yaml` Creates/Updates resources described in my-nginx-deployment.yaml
  - `kubectl delete -f f my-nginx-deployment.yaml` Deletes resources described in my-nginx-deployment.yaml
- **Editing resources**
  - `kubectl edit deployment my-nginx` Opens NotePad (on the editor configured in EDITOR ou KUBE_EDITOR env variable) with the current state of the resource. After editing and saving the resource will be updated.
- **Accessing Pod Container Logs**
  - `kubectl logs etcd-docker-desktop –n kube-system` Prints the log of the etcd pod

16

Docker Cheat Sheet

https://www.docker.com/sites/default/files/d8/2019-09/docker-cheat-sheet.pdf/

17

Building and Shipping an Image to a registry

- **Build an image from the Dockerfile in the current directory and tag the image**
  - `docker build -t myimage:1.0 .`
- **List all images that are locally stored with the Docker Engine**

  `docker image ls`
- **Ship an image to DockerHub.com (An account is required)**
  - `docker push myrepo/myimage:1.0`
- **How to create a local Docker Registy**
  - `docker run -p 5000:5000 --restart=always--name myregistry -v $(pwd):/var/lib/registry registry:2`
    [https://docs.docker.com/registry/deploying/]
- **How to ship an image to a local docker registry**
  - `docker image tag my-image localhost:5000/my-image`
  - `docker push localhost:5000/my-image`

18