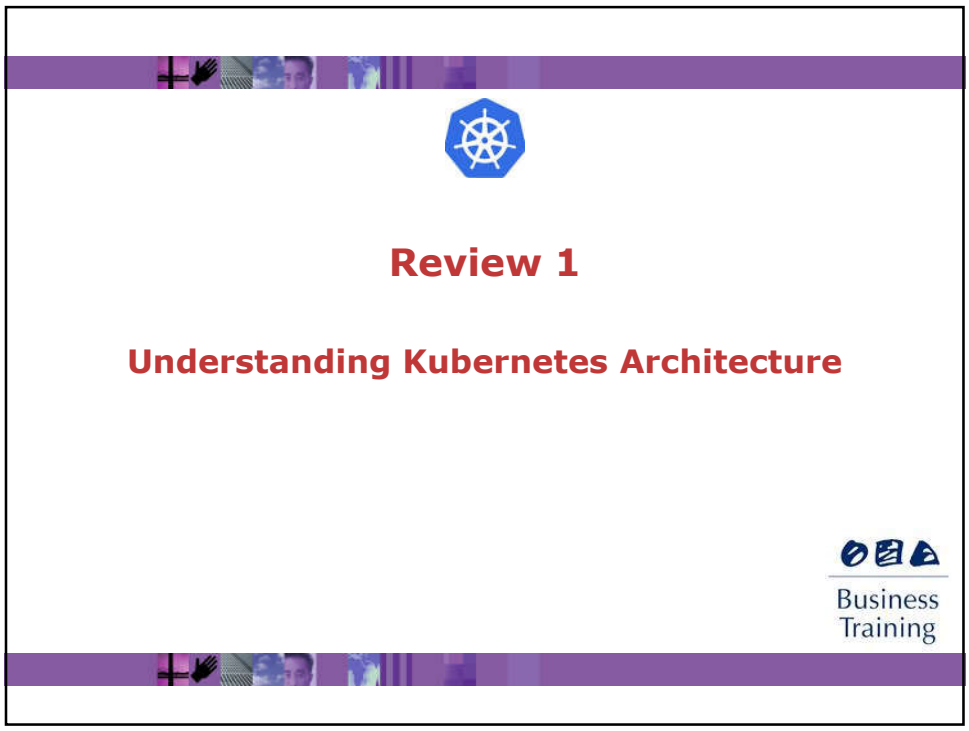
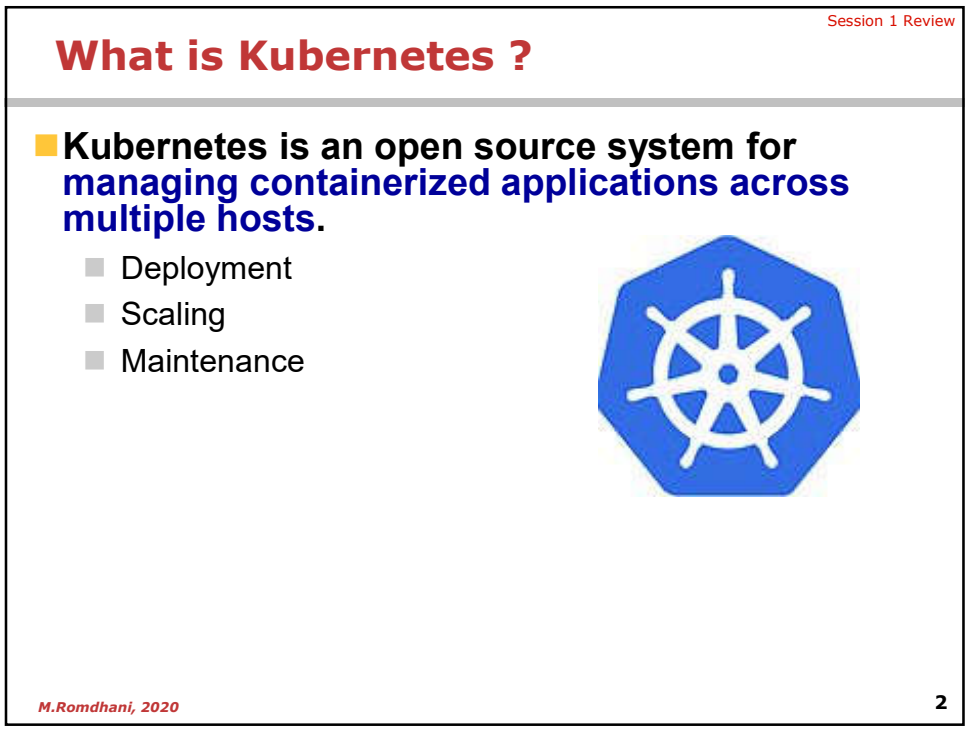

Review 1
Understanding Kubernetes Architecture
Business
Training

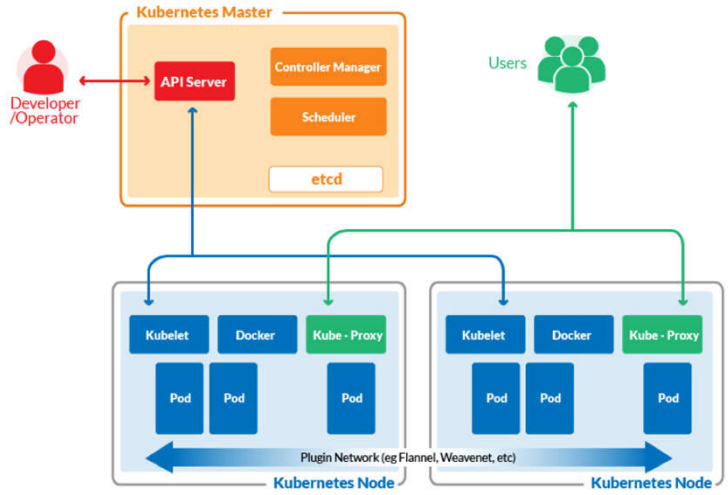
1


Session 1 Review
What is Kubernetes ?
Kubernetes is an open source system for managing containerized applications across multiple hosts.
Deployment
Scaling
Maintenance
M.Romdhani, 2020
2

2

Session 1 Review

Kubernetes Architecture

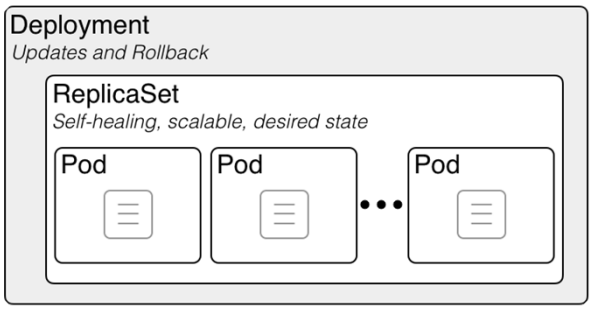


3

Session 1 Review

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.

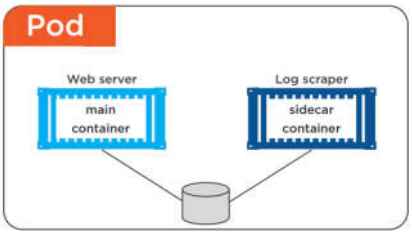


4

Session 1 Review

What is a Pod ?

- **A Pod is the basic building block of Kubernetes–the smallest and simplest unit in the Kubernetes object model that you create or deploy.**
  - A Kubernetes pod is a group of containers that are deployed together on the same host and share storage and networking resources. it's very common to have a group of containers work together to produce an artifact or process a set of work.
- **Containers within a Pod share an IP address and port space, and can find each other via localhost.**
- **Pods aren't intended to be treated as durable entities. They are ephemeral.**



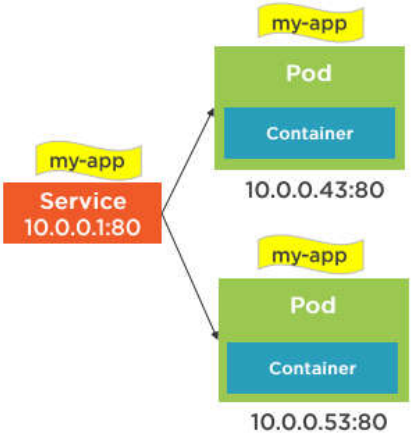
- **Pods serve as unit of deployment, horizontal scaling, and replication.**

5

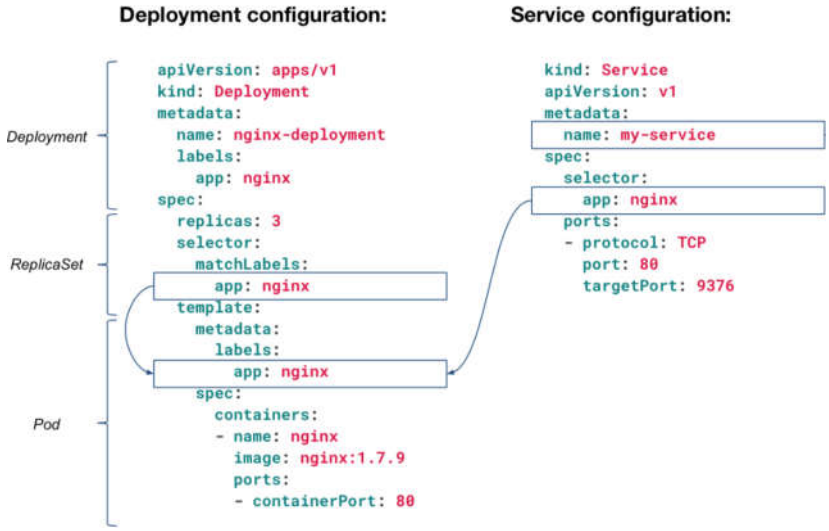
Session 1 Review

What is a Service ?

- **A service is an abstract way to expose an application running on a set of Pods as a network service.**
  - It acts as the unified method of accessing replicated pods.
- **Services provide important features that are standardized across the cluster:**
  - **Services abstract Pod IP Addresses from consumers**
  - **Load-balancing between Pods**
  - **Rely on labels to associate service with a Pod**
  - **Node's kube-proxy creates virtual IP for services**
  - **Services are not ephemeral**



Labels and Selectors



7

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

8

Kubernets manifests examples

- **The required fields in the .yaml file:**
  - **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

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-nginx
  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
```


9

Vocabulary/Concepts

- **Cluster**
- **Control Plane**
- **Orchestration/scheduling**
- **Sel Healing**
- **Desired State**
- **Scalability**
- **Workload**
- **Labels and Selectors**

10