

## Kubernetes

**Pod** : Collection of Containers also, it is the smallest unit in terms in Kubernetes

**API Resources:** All the Kubernetes related Resources

Derling : Copy of a Re...

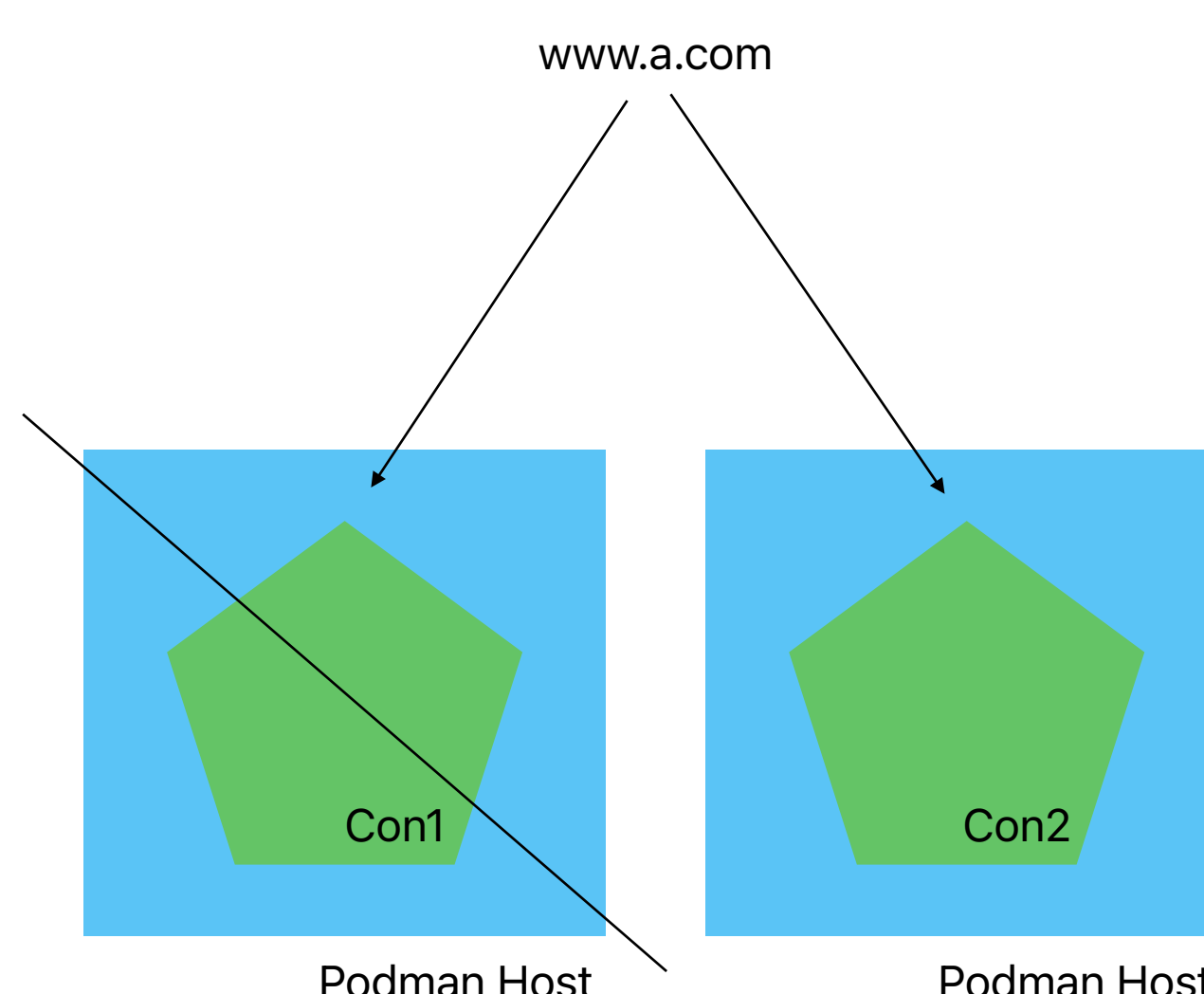
Node : A Server

**Cluster:** The whole Environment includes all nodes, all end resources,

## CONTAINERIZATION

## KIIRERNETE

11



www.a.com

Desired State: 3 Replicas  
Current State: 3 Replicas

node

pod

can.

API Resource  
**Replication Controller (RC)**

2



Apache Container --- disk: sat  
MySQL Container --- disk: ssd



```
API Resource
Scheduler

apache
nodedselector = disk:sata      replicas = 3

MySQL
nodedselector = disk:ssd
```

3



admin: redha

A: 5  
B: 10  
C: 15

**Authentication:** Who can login  
**Authorisation:** What a logged in Person can do

## API Resource

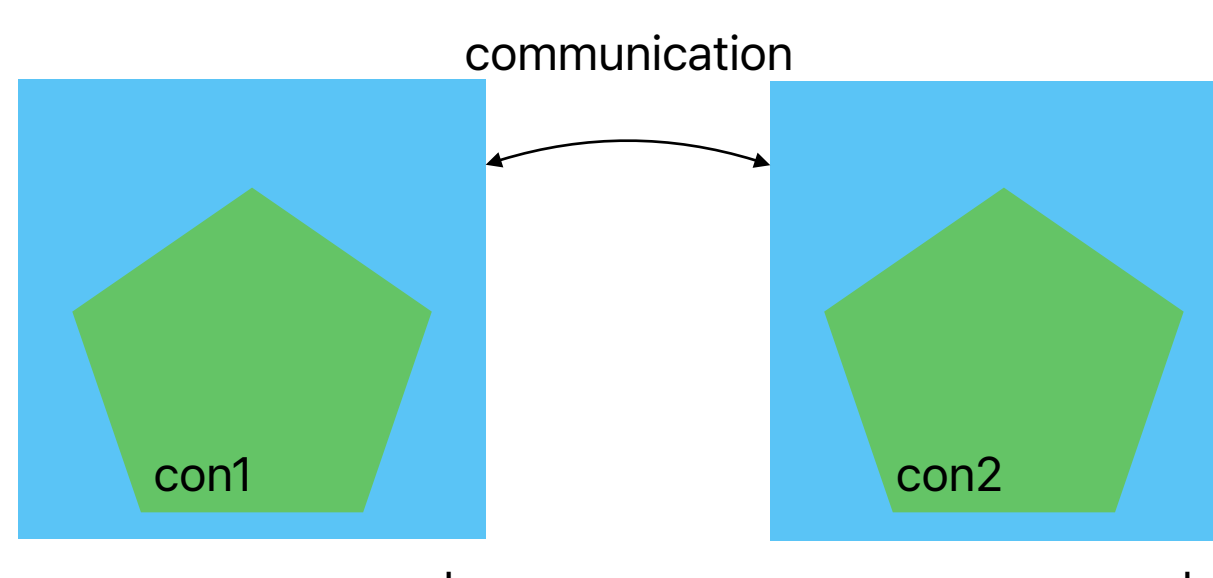
1

We do not have any control data storage devices.

## API Resource

A database that stores all information regarding the cluster configurations and states etc

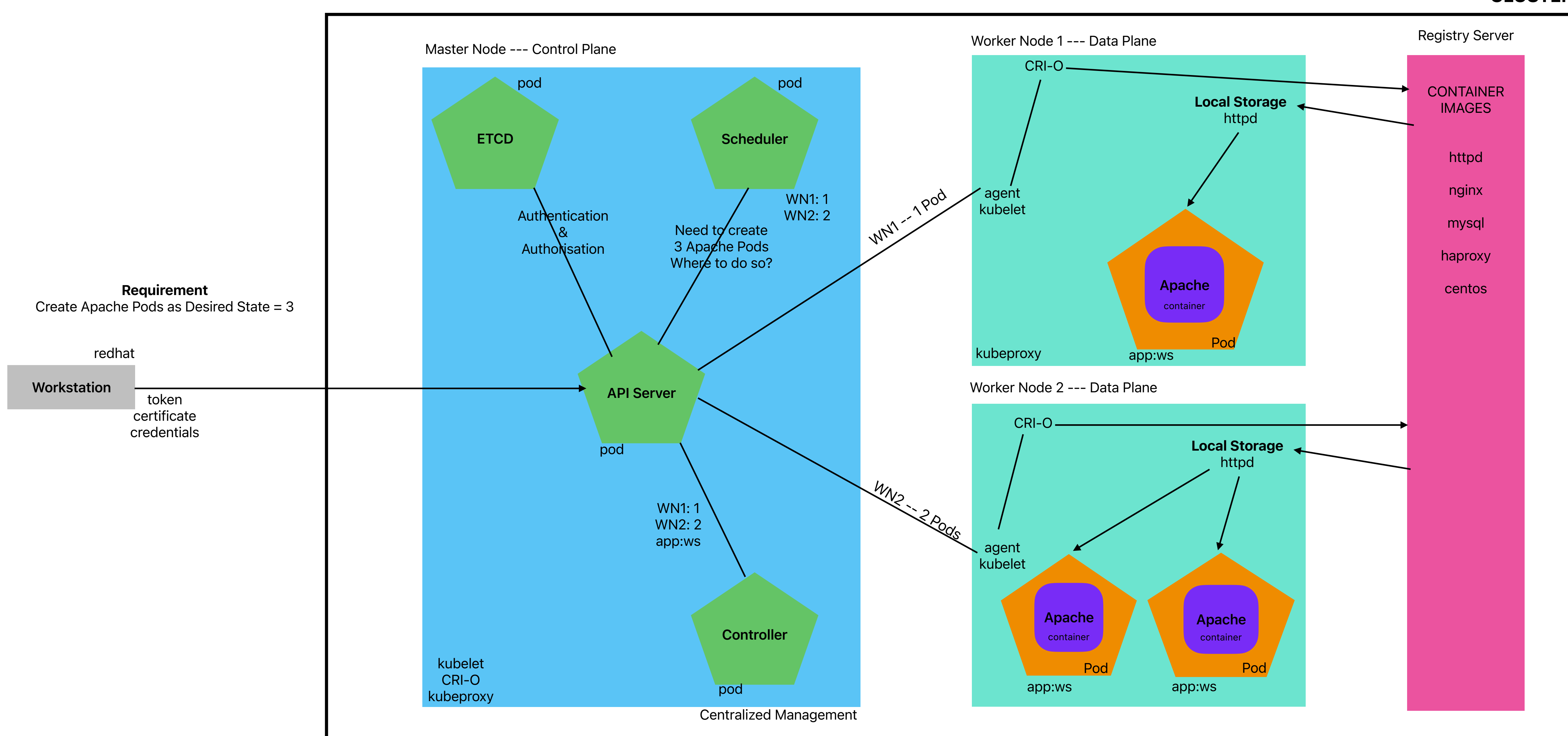
5



It gives you SDN (Software Defined Networking)

## KUBERNETES ARCHITECTURE

**CLUSTER**



## OpenShift

Developer --- index.php

1. Need to create a Containerfile / Dockerfile, to set all parameters according to his/her requirements
2. Create an Image using the Dockerfile.
3. Upload that Image to a Registry Server.
4. Need to create a Deployment which will create the RC/RS which will create the Pods
5. Need to setup the Networking

1. Need to create a Containerfile / Dockerfile, to set all parameters according to his/her requirements
2. The Developer can send his/her Dockerfile to OpenShift.
3. Automatically create an Image using the Dockerfile.
4. Automatically upload that Image to a Registry Server.
5. Automatically create a Deployment which will create the RC/RS which will create the Pods
6. Automatically setup the Networking

1. Developer has to upload the content onto a GitHub Repository
2. Provide OpenShift with that repository.
3. OpenShift will analyze the content and find the index file and accordingly use a builder image
4. OpenShift will create a Dockerfile from the builder
5. Automatically create an image using the Dockerfile.
6. Automatically upload that image to a Registry Server.
7. Automatically create a Deployment which will create the RC/RS which will create the Pods
8. Automatically setup the Networking

S21

Kubernetes	OpenShift
<p>Caas</p>	<p>PaaS and S2I</p>
<p>OpenSource</p>	<p>Enterprise OpenSource -- okd</p>
<p>Installed on all OS CentOS, RHEL, Ubuntu</p>	<p>Master Node has to be mandatorily rhelcoreos</p> <ol style="list-style-type: none"> <li>1. lightweight</li> <li>2. updates a single package</li> <li>3. immutable</li> </ol>
	<p>Worker Nodes can be rhelcoreos / rhel</p>
<p>No Preconfigured Console No Preconfigured Monitoring Tool No Preconfigured Logging Tool No Preconfigured Ingress Controller No Preconfigured Registry Server</p>	<p>Has Preconfigured Console Has Preconfigured Monitoring Tool Preconfigured Logging Tool Preconfigured Ingress Controller Preconfigured Registry Server</p>
	<p>WebGUI Prometheus, Grafana ELK Router Pods Internal OpenShift Regist</p>
<p>Containers run with Root privilege</p>	<p>Containers run with random user</p>

**GITHUB:** [dtq2468](#)