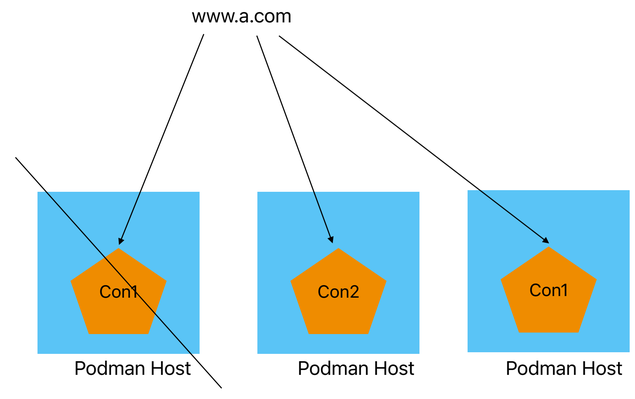
Introduction to Kubernetes

Orchestration Tool to manage Containers

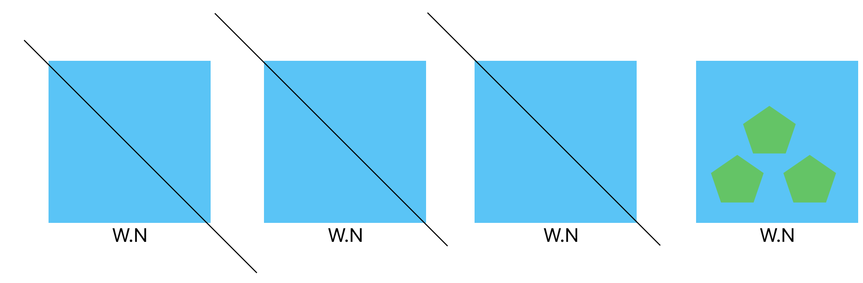
IMPORTANT TERMS
Pod: A collection of one or more containers
Node: Any Server/Host/Machine
API Resource: Set of Resources that define the Kubernetes Architecture
Cluster: Collection of all Nodes and all Resources
Worker Node: Where pods are created
Master Node: This is from where we manage the Kubernetes Cluster

Problem 1:



API Resource: **Replication Controller (RC) / Replica Set (RS)**

Desired Replica Value of Pods == 3
Current State == 3



Problem 2:

6 Podman Hosts

Whichever Podman Host has disk: ssd ----> Install MySQL
Whichever Podman Host has disk: sata ----> Install Httpd

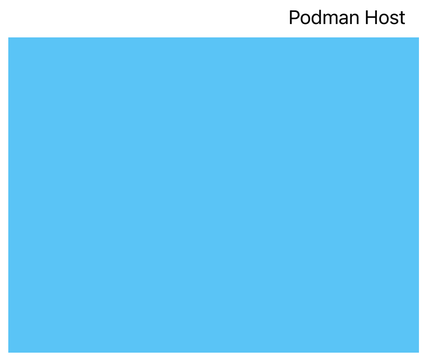
API Resource: **Scheduler**

Nodeselector:

httpd:
disk: sata

mysql:
disk:ssd

Problem 3:



A: 5 containers
B: 3 containers
C: 2 containers

API Resource: **API Server**

It Manages --

Authentication: Who can Login
Authorisation: After logging what can the Person do

A: 5 Pods
B: 3 Pods
C: 2 Pods

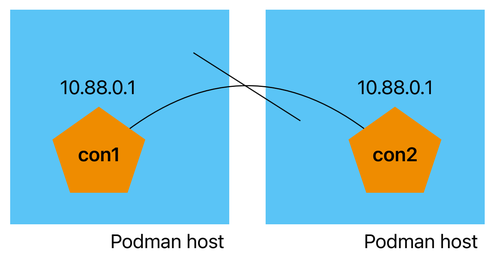
Problem 4:

In Podman, we do not a centralized database to store information

API Resource: **ETCD**

A Distributed Database which stores each and every information about your cluster.
Only the API Server can write into the ETCD.

Problem 5:



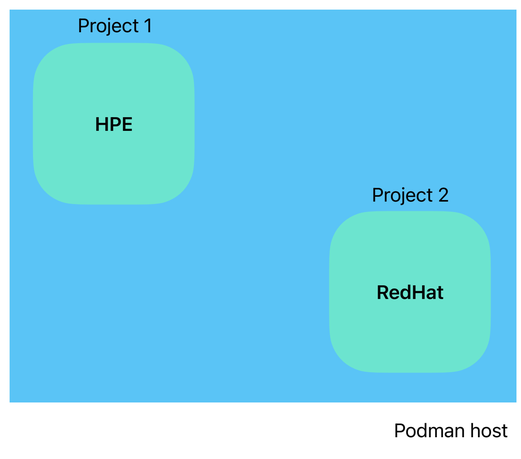
Kubernetes - Better Networking and Routing

SDN (Software Defined Networking)

With this all the pods across the cluster can be in the same network and can communicate.
This can be completely restricted as well.

Problem 6:

A --- Project 1
B --- Project 2
C --- Project 1
D --- Project 2

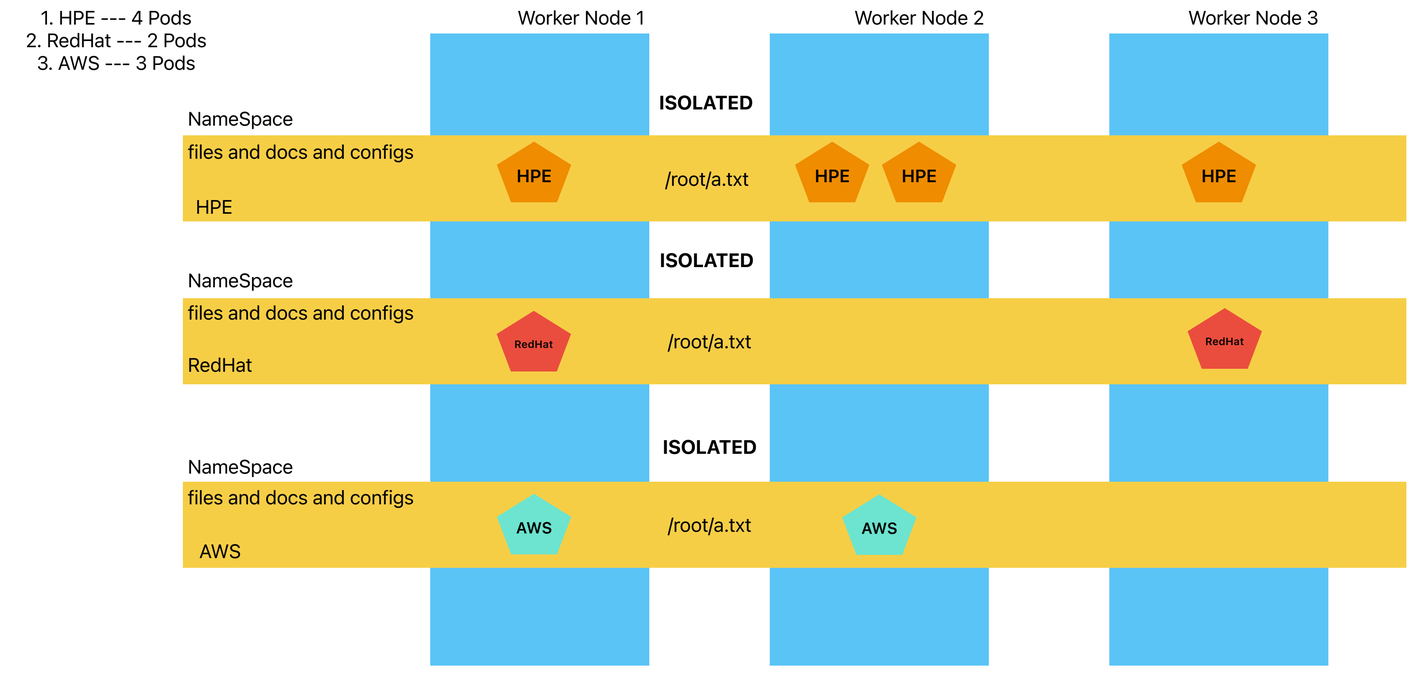


Namespace Isolation

A --- HPE Namespace
B --- RedHat Namespace
C --- AWS namespace

3 Projects

1. HPE --- 4 Pods
2. RedHat --- 2 Pods
3. AWS --- 3 Pods



Problem 7:

No Container Autoscaling
No Node Autoscaling

Container Autoscaling --- Manual / Automatic
Node Autoscaling --- Cluster Auto Scaler