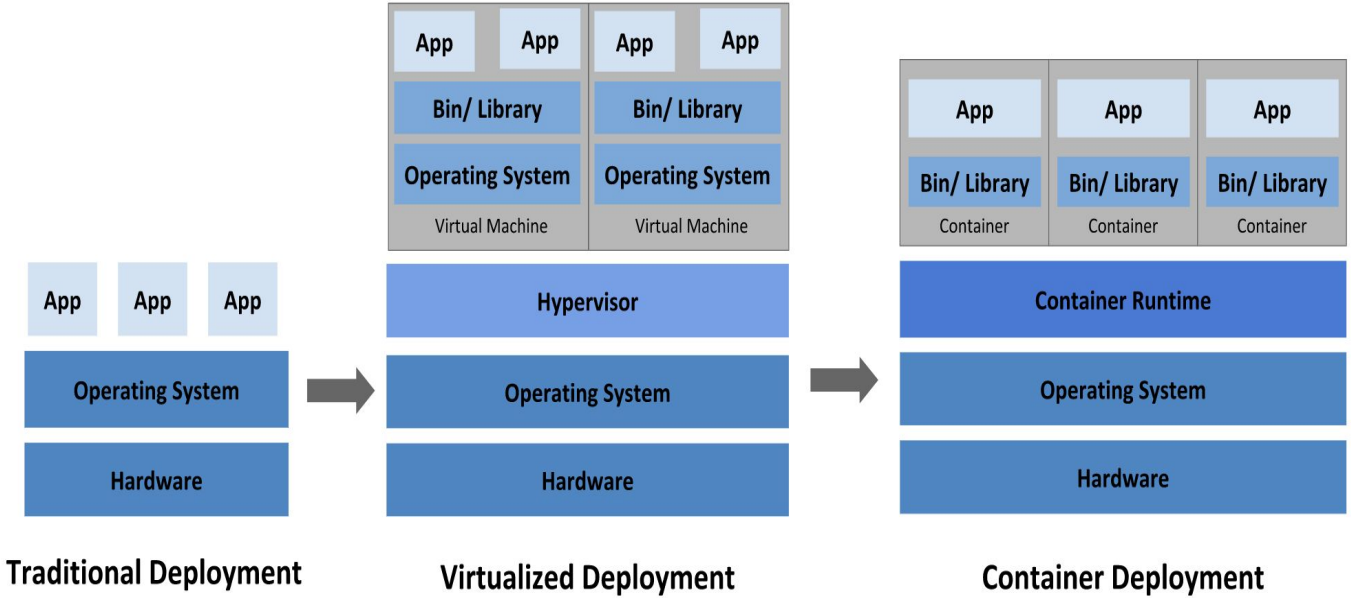
Tech Talk on Kubernetes

By Shivani Singhal

Why kubernetes?

Let's take a look at why Kubernetes is so useful by going back in time.



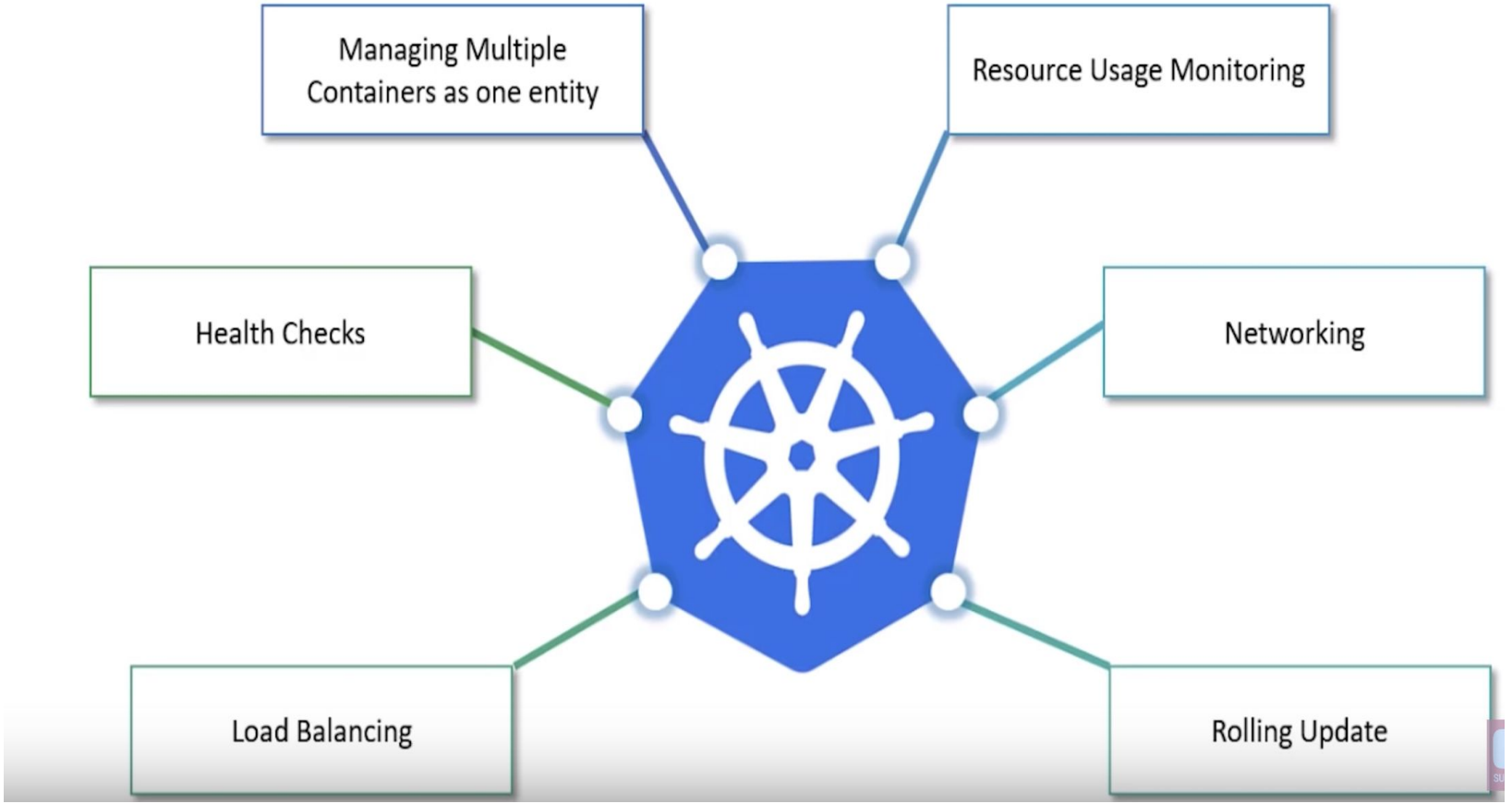
What is Kubernetes?

"Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure."

http://kubernetes.io/docs/whatisk8s/

Kubernetes Features

Kubernetes Features



What does Kubernetes do?

- Provide a runtime environment for Docker containers
- Scale and load balance docker containers
- Abstract away the infrastructure containers run on
- Monitor/health check containers
- Declarative definition for running containers
- Update containers (also rolling updates)
- Storage mounting (allow abstracting infrastructure)
- Service discovery and exposure
- Labelling and selection of any kind of object (we‘ll get to this)

What does kubernetes not do?

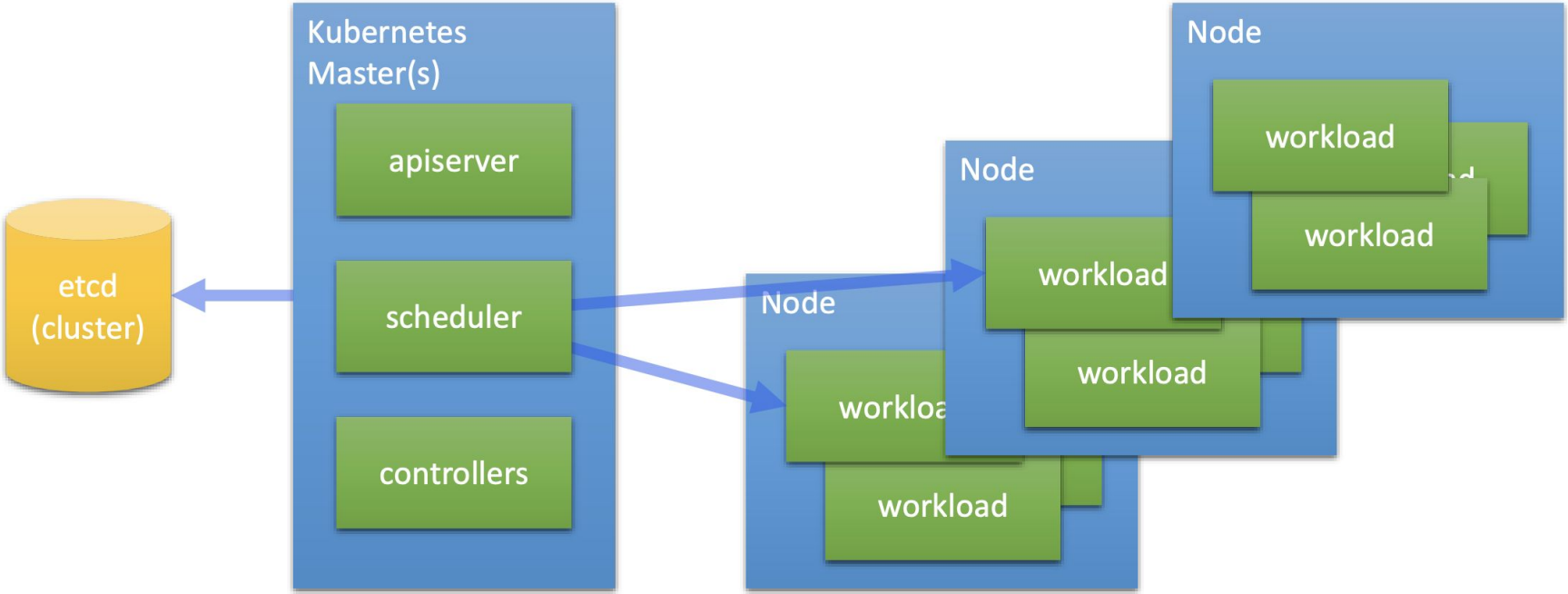
- Provide any additional services than just Docker
- Compile or build your source code (needs images)
- Provide real orchestration, relies on containers working independently
- Mandate or provide any kind of special configuration language
  - You're free to do whatever you want
  - But: You have to find out a way yourself

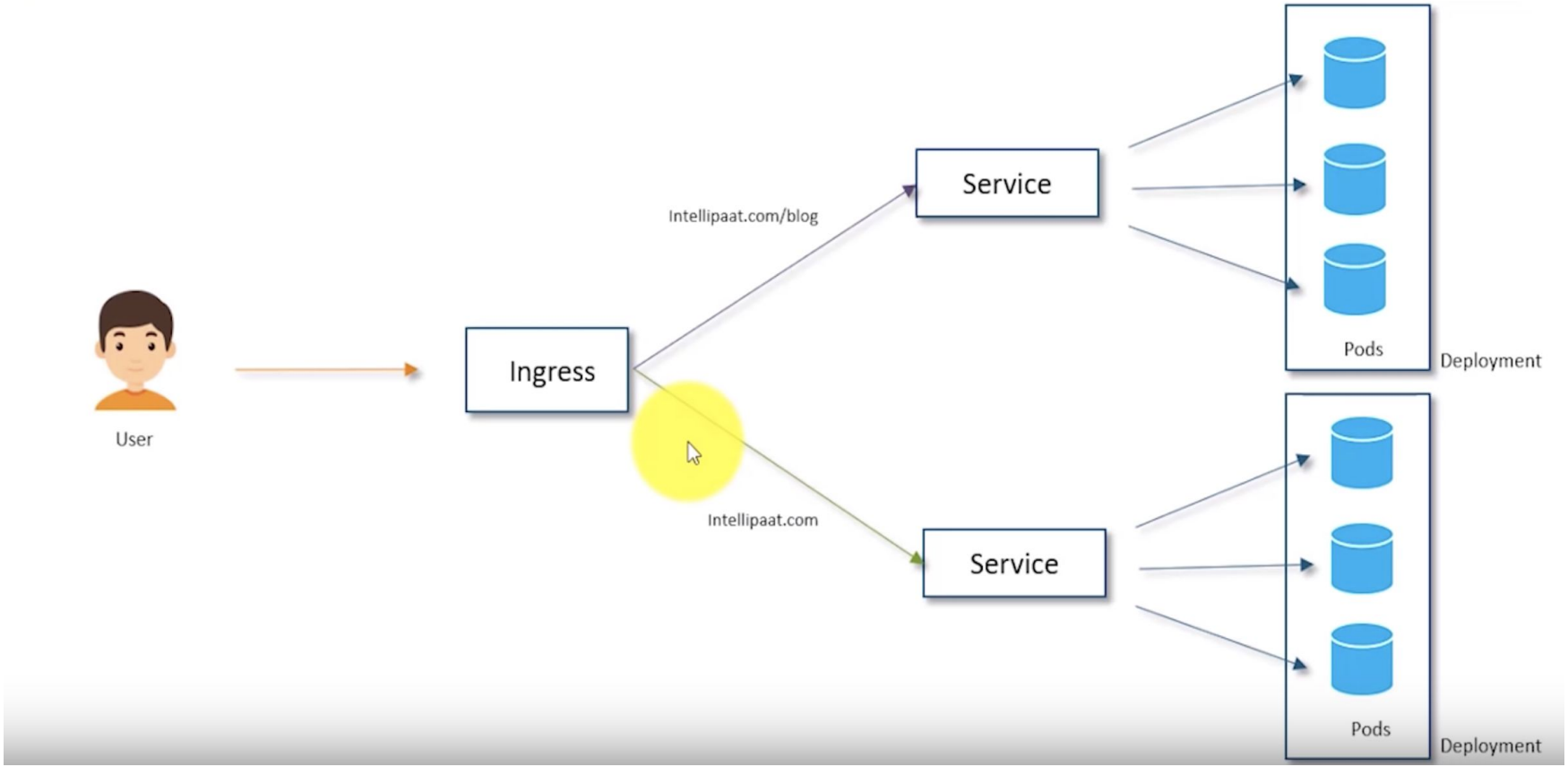
Kubernetes and Docker

- Kubernetes adds functionality to Docker
- Manages a set of Docker Hosts, forming a Cluster
- Takes care of Container scheduling
- Supervises Docker containers
- Kubernetes is **replacement/alternative** for Docker Swarm

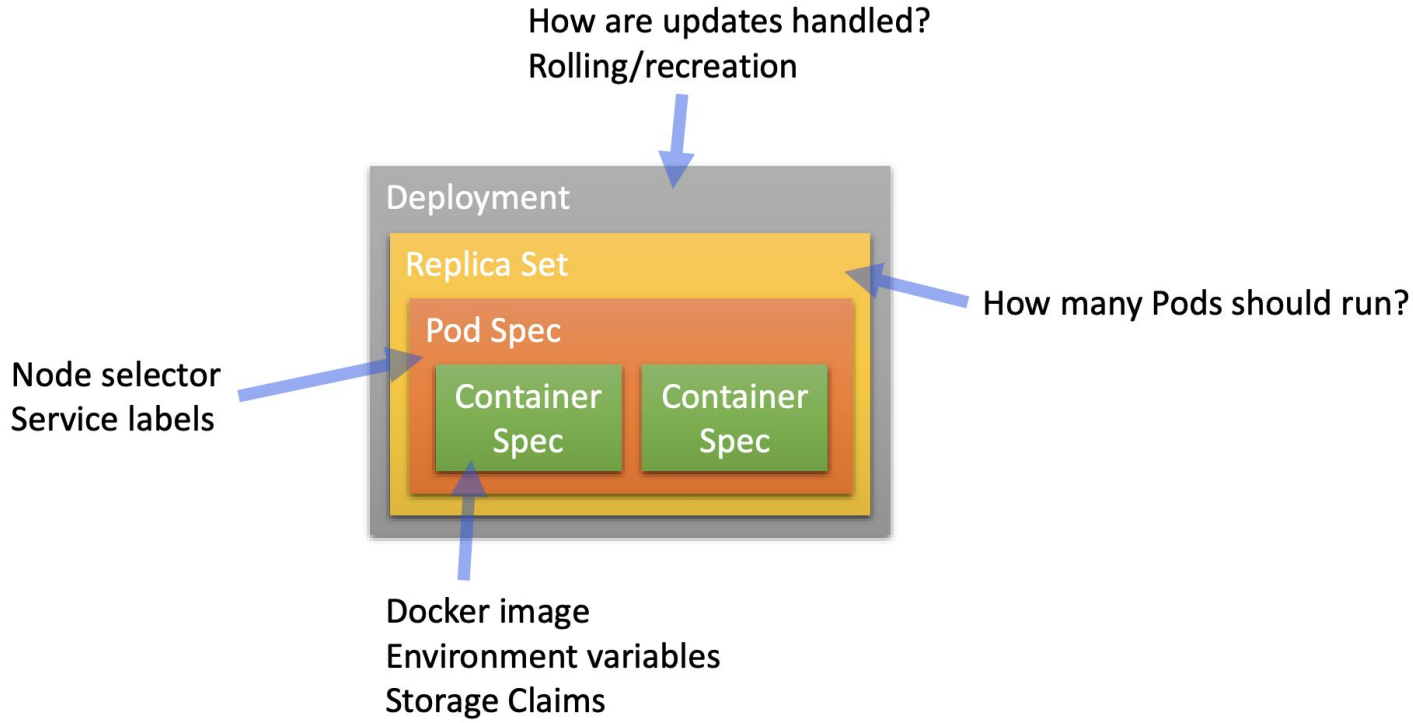
Deployment Architecture

All blue boxes are Docker Hosts (VMs)
Kubernetes Components are also running as stateless containers!

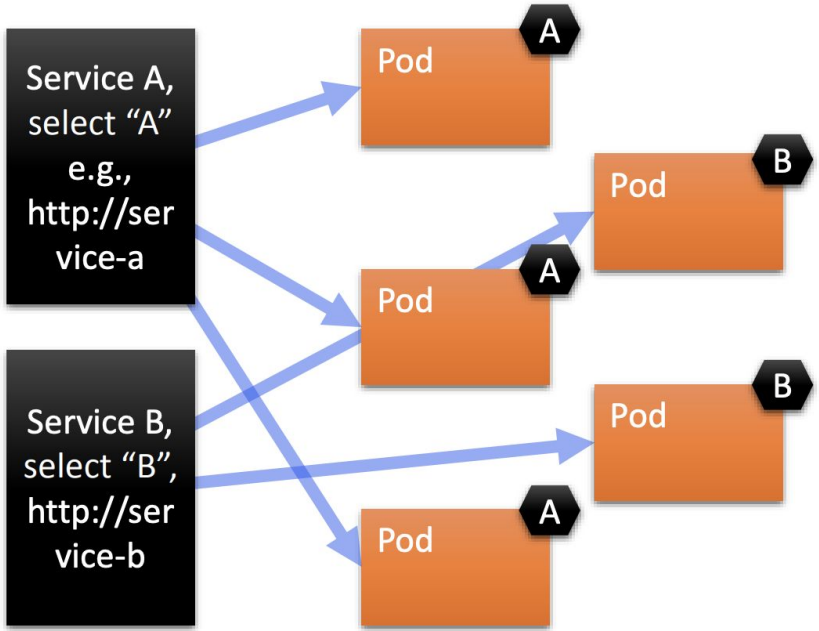



User
Ingress
Intellipaat.com/blog
Intellipaat.com
Service
Service
Pods
Deployment
Pods
Deployment

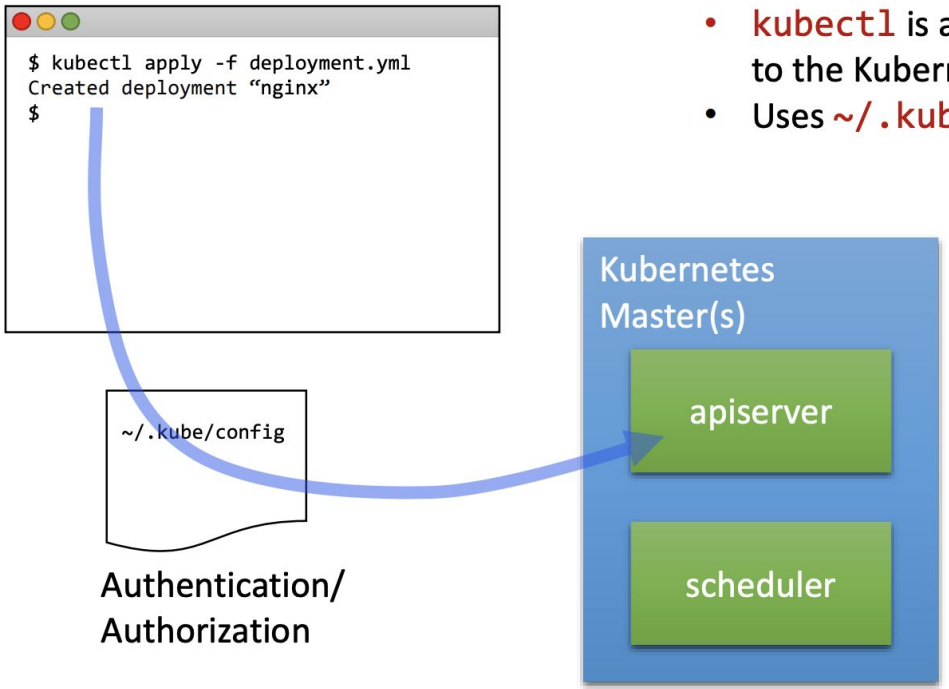
Abstractions (1) - “Boxes in boxes”



Abstractions (2) - Services



Working with kubectl



- `kubectl` is a convenient way to talk to the Kubernetes API
- Uses `~/.kube/config` for AuthN/Z

DEMO

Steps to launch a deployment using an image path

- docker-machine ls
- docker-machine ip shivani-k8s-demo
- Minikube delete
- minikube start --vm-driver="virtualbox" --insecure-registry="docker-machine IP":80
- kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.10
- kubectl expose deployment hello-minikube --type=NodePort --port=8080
- kubectl get pod
- minikube service hello-minikube --url

Delete all resources

- Minikube delete svc hello-minikube
- Minikube delete deploy/hello-minikube
- Minikube delete

Subdividing your cluster using Kubernetes namespaces

- Kubectl get namespaces
- kubectl create -f https://k8s.io/examples/admin/namespace-dev.json
- kubectl create -f https://k8s.io/examples/admin/namespace-prod.json
- kubectl get namespaces --show-labels
- kubectl config view
- kubectl config current-context
- kubectl config set-context dev --namespace=development --cluster=minikube --user=minikube
- kubectl config set-context prod --namespace=production --cluster=minikube --user=minikube
- kubectl config use-context dev
- kubectl run snowflake --image=k8s.gcr.io/serve_hostname --replicas=2
- kubectl get deployment
- kubectl config use-context prod
- kubectl run cattle --image=k8s.gcr.io/serve_hostname --replicas=5
- kubectl get deployment
- kubectl get pods -l run=cattle

Reference Links

- https://kubernetes.io/docs/concepts/
- https://kubernetes.io/docs/tasks/administer-cluster/namespaces/

Thank You