

Kubernetes For DevOps

Lev Epshtein



kubernetes

What is Kubernetes?

“Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.”

Kubernetes

Kubernetes has a number of features. It can be thought of as:

- a container platform
- a microservices platform
- a portable cloud platform and a lot more.
- Kubernetes provides a container-centric management environment. It orchestrates computing, networking, and storage infrastructure on behalf of user workloads. This provides much of the simplicity of Platform as a Service (PaaS) with the flexibility of Infrastructure as a Service (IaaS), and enables portability across infrastructure providers.
- K8S Allows developers / system operators to cut to the cord and truly run a container-centric dev / microservice environment



Kubernetes

Core Concepts & Components

Kubernetes Concepts

Understanding K8S system and abstraction

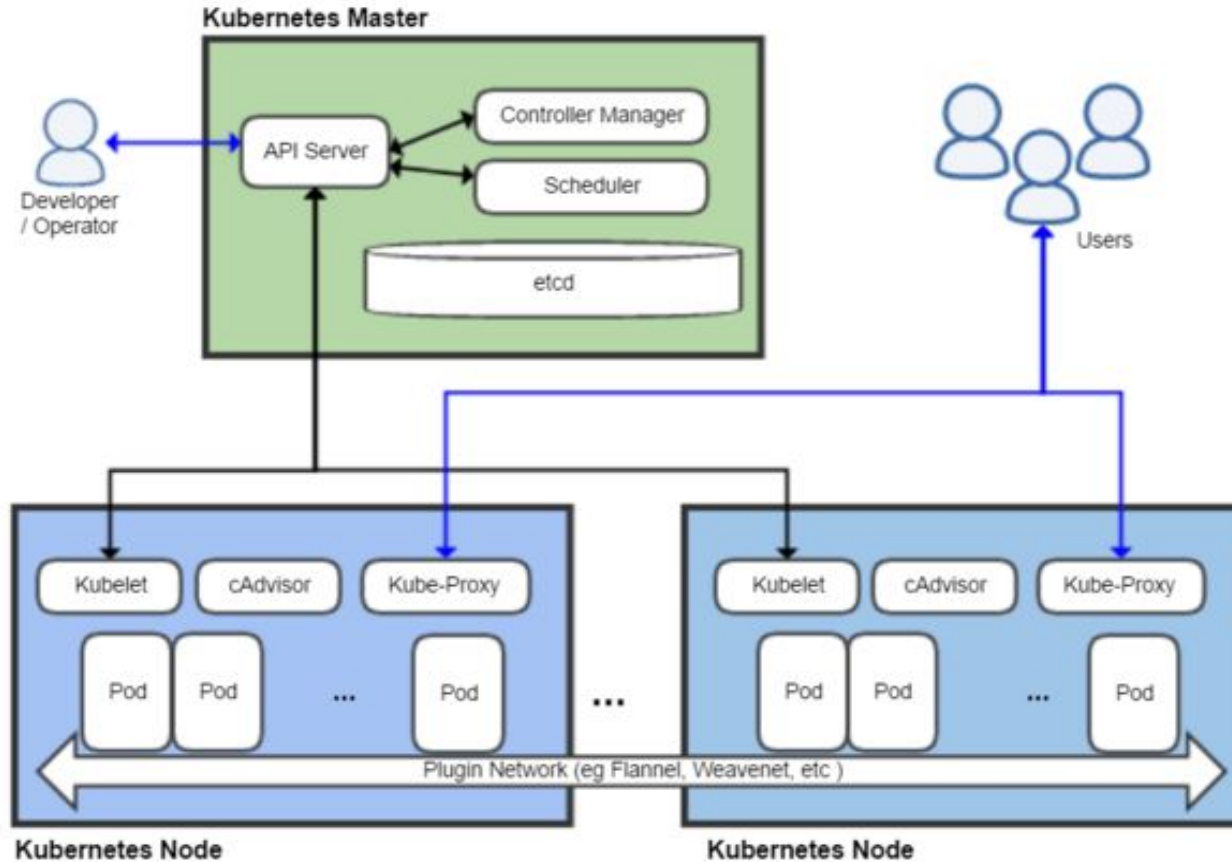
To work with Kubernetes, we use Kubernetes API objects to describe our cluster's desired state:

- what applications or other workloads we want to run
- What container images they use, the number of replicas, what network and disk resources we want to make available.
- Setting our desired state by creating objects using the Kubernetes API (typically via the command-line interface - `kubectl`)

Once we've set our desired state,

Kubernetes Control Plane works to make the cluster's current state match the desired state

Kubernetes



K8S: Master Components

Master components provide the cluster's control plane

- **Kube-apiserver** - Validates and configures data for the api objects which include pods, services, replicationcontrollers, and others.
- **Kube-controller-manager** - is a an application control loop that watches the shared state of the cluster through the apiserver and makes changes attempting to move the current state towards the desired state.
- **Kube-scheduler** - its job is to take pods that aren't bound to a node, and assign them one along with hardware/software/policy constraints
- **etcd** - highly-available key value store used as Kubernetes' backing store for all cluster data

Non Master Kubernetes Components

- Each individual non-master node in our cluster runs two processes:
 - **Kubelet** - which communicates with the Kubernetes Master.
 - **Kube-proxy** - A network proxy which reflects Kubernetes networking services on each node.



microk8s Installation

Installation

<https://kubernetes.io/docs/setup/>

During the course we will work with

<https://microk8s.io/>

<https://snapcraft.io/install/microk8s/rhel>

MicroK8s brings up Kubernetes as a number of different services run through `systemd`.

microk8s.enable dns ingress storage helm

Kubernetes Basics

To work with **Kubernetes objects**—whether to create, modify, or delete them—you'll need to use the **Kubernetes API**. When you use the `kubectl` command-line interface.

<https://kubernetes.io/docs/concepts/overview/working-with-objects/kubernetes-objects/>

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

Kubernetes Building Blocks

Basic Kubernetes Object

- ❑ POD
- ❑ SERVICE
- ❑ VOLUME
- ❑ NAMESPACE

Controllers

- ❑ Deployment
- ❑ StatefulSet
- ❑ DaemonSet
- ❑ JOB
- ❑ ReplicaSet

Kubernetes Namespaces

Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called namespaces.

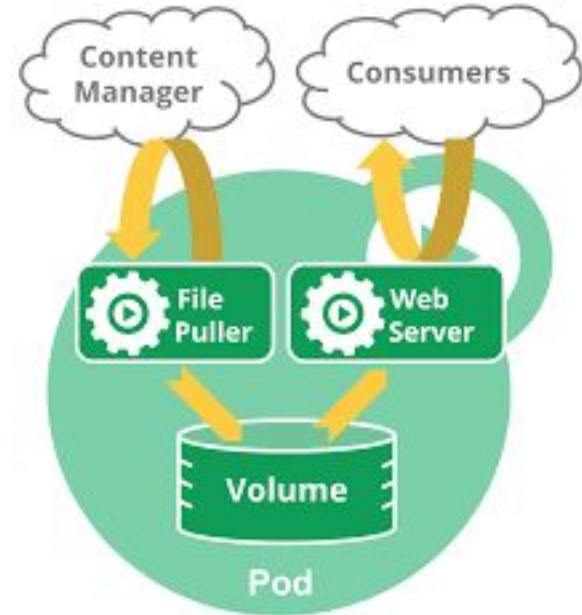
Names of resources need to be unique within a namespace, but not across namespaces.

Namespaces can not be nested inside one another and each Kubernetes resource can only be in one namespace.

Namespaces are a way to divide cluster resources between multiple users

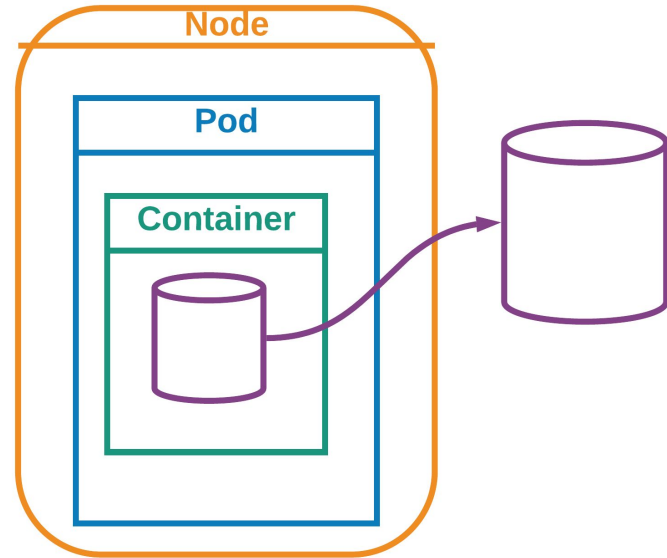
Kubernetes POD

- Basic building blocks.
 - Pods that run a single container
 - Pods that run multiple containers that need to work together



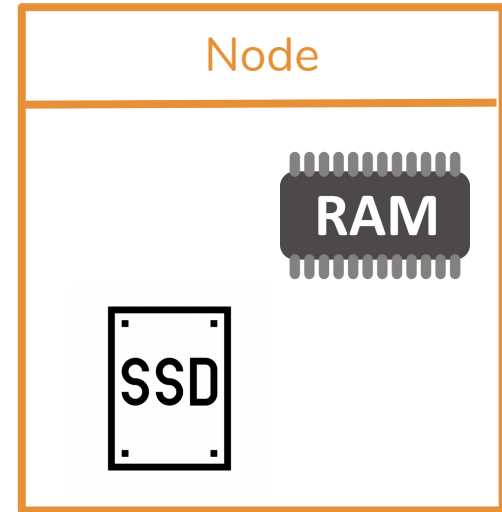
Kubernetes Volumes

- Persistent Volume for save state.
- When running a container together in a Pod - share files between those containers.

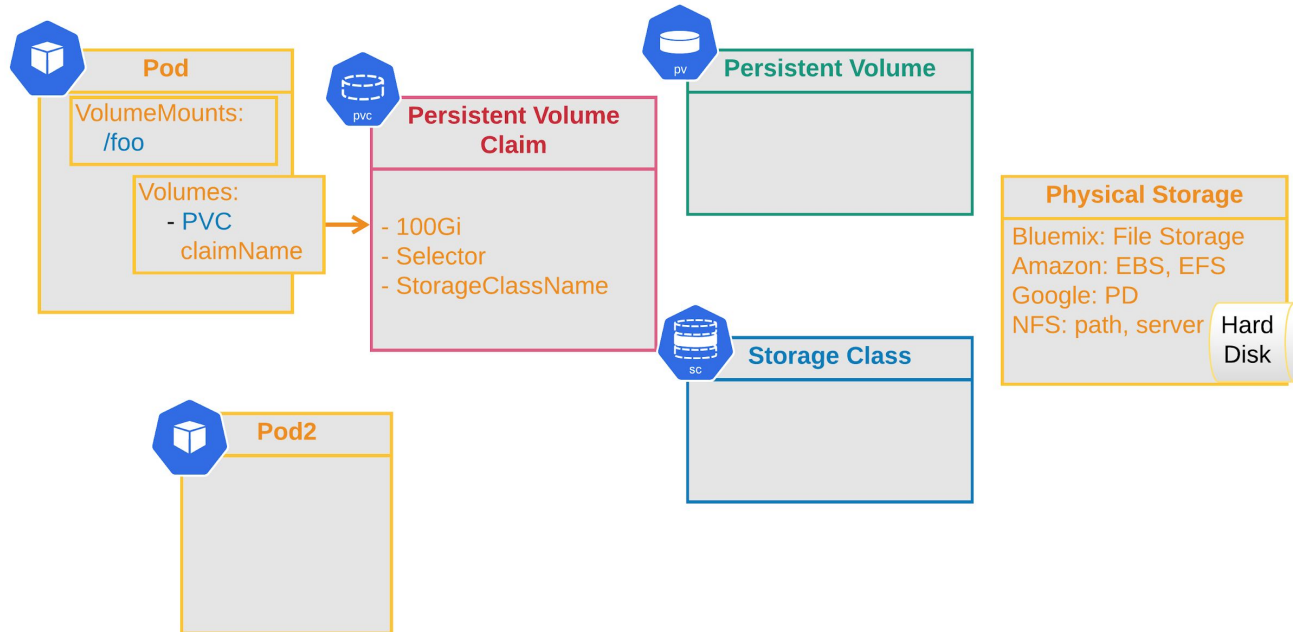


Kubernetes Volumes

- emptydir
- hostPath
- Cloud Volumes
- NFS
- Persistent Volume Claim
PVC

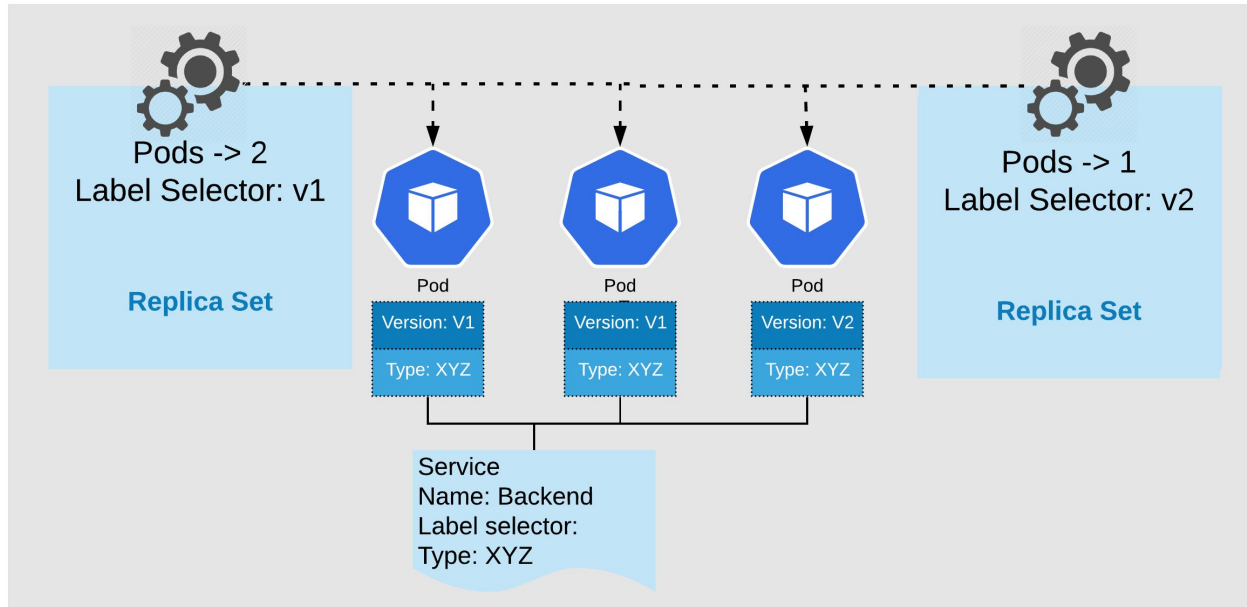


Kubernetes Volumes



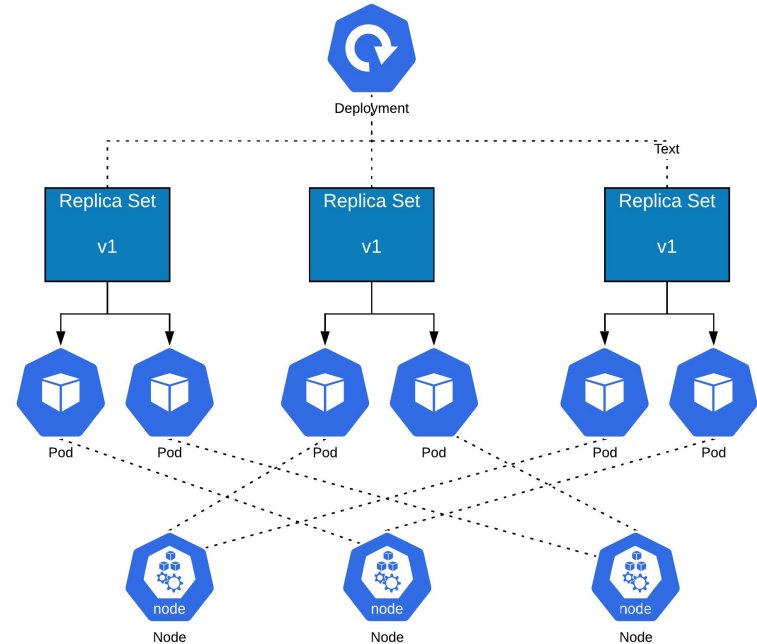
ReplicaSets

- ReplicaSet ensure how many replica of pod should be running. It can be considered as a replacement of replication controller.



Kubernetes Deployments

A controller that provides declarative updates for Pods and ReplicaSets. We describe a desired state in a Deployment object, and the Deployment controller changes the actual state to the desired state at a controlled rate.



Deployment Use - Case

- **Create a Deployment:**

A deployment is created. Once that is done the ReplicaSet automatically creates Pods in the background.

- **Update Deployment:**

A new ReplicaSet is created and the Deployment is updated. Each new ReplicaSet updates the revision of Deployment.

- **Rollback Deployment:**

Used in case when the current state of the deployment is not stable.
Only the container image gets updated.

- **Scale Deployment:**

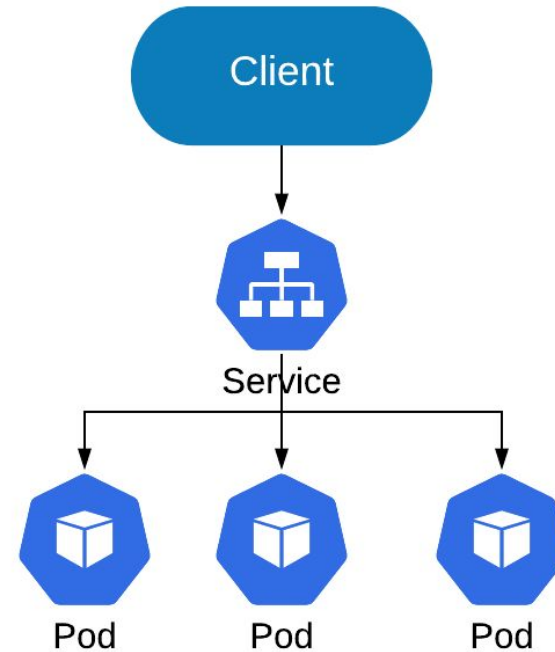
Each and every deployment can be scaled up or scaled down based on the requirement.

- **Pause the Deployment:**

Pause the deployment to apply multiple fixes and then the deployment can be resumed.

Kubernetes Services

- ClusterIP
- NodePort
- LoadBalancer

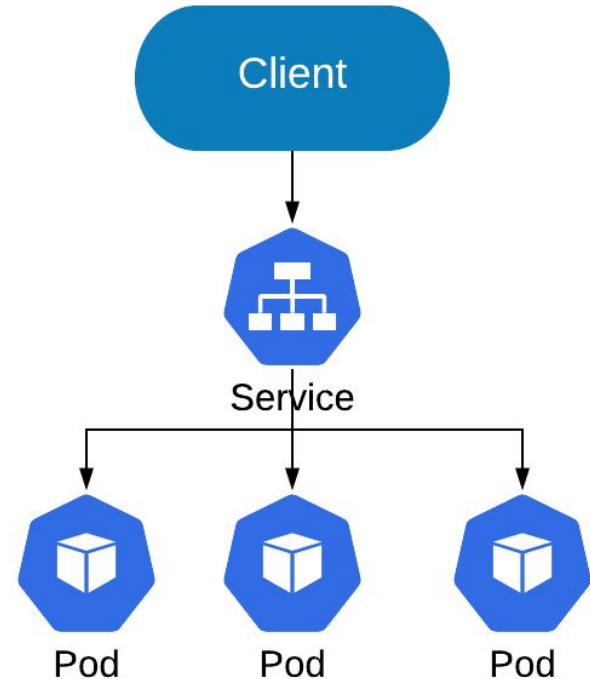


ClusterIP Service

ClusterIP

- Service **inside your cluster** that other apps inside your cluster can access.
- You can access it using k8s proxy:

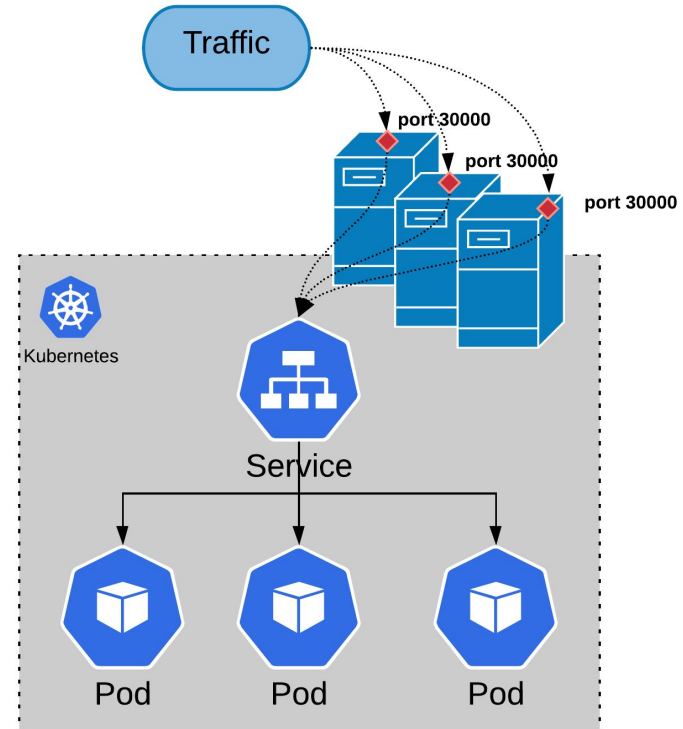
```
$ kubectl proxy --port=8080
```



NodePort Service

NodePort

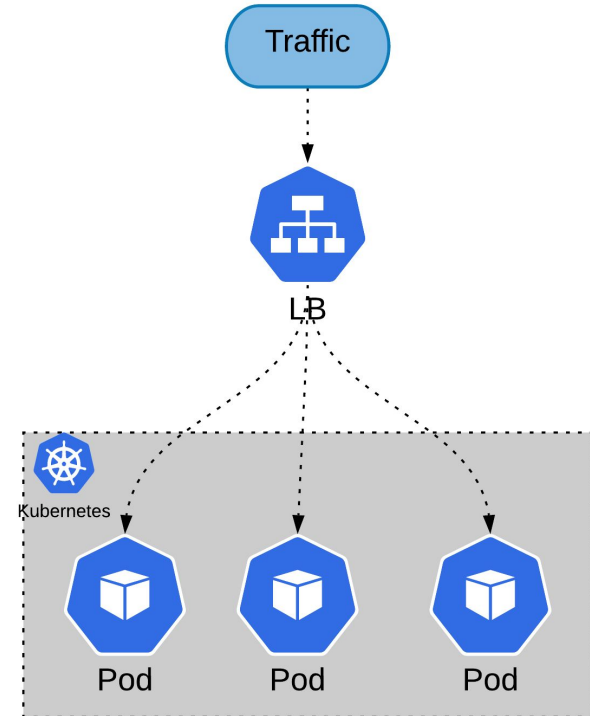
- Specific port on all the Nodes (the VMs) is open. (> 30000)
- Any traffic that is sent to this port is forwarded to the service.



LoadBalancer Service

LoadBalancer

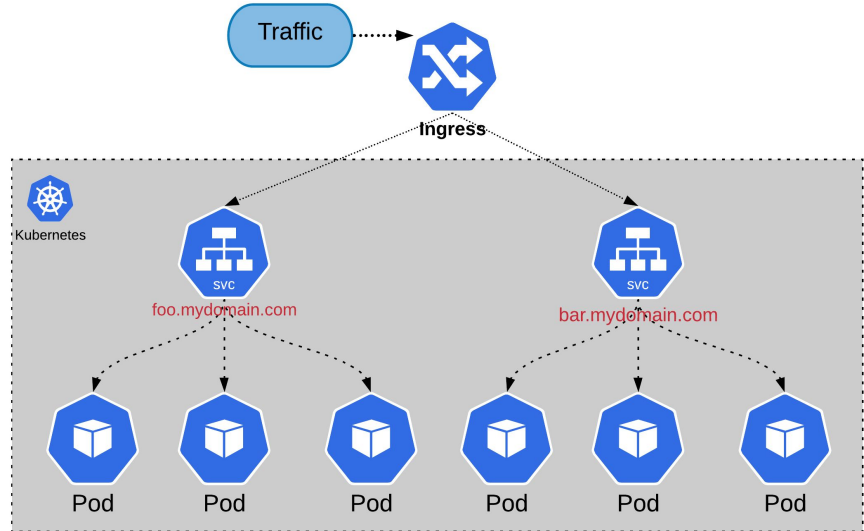
- External using network load balancer with a public IP.
- Standard way to expose a service to the internet.



Ingress

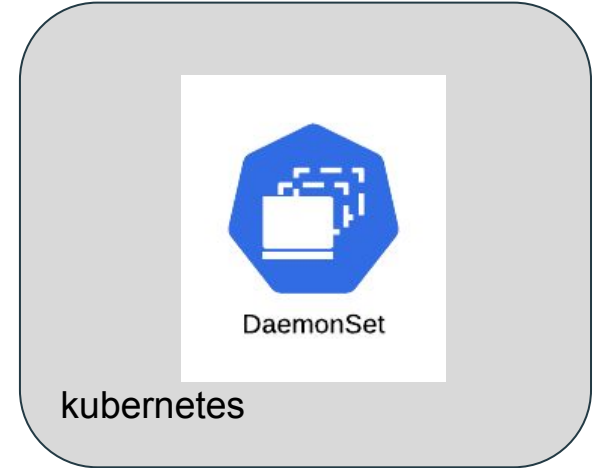
Ingress

- Ingress is a load balancer that will front multiple different pods.
- Traffic is routed via http URI or dns names.
- In GKE Ingress controller will spin up a HTTP(S) Load Balancer.



Config map

Kubernetes DaemonSet



Kubernetes JOB

Labels and Selectors

Kubernetes Cheat Sheet

What is Kubernetes?

Kubernetes is a platform for managing containerized workloads. Kubernetes orchestrates computing, networking and storage to provide a seamless portability across infrastructure providers.

Viewing Resource Information

Nodes

```
$ kubectl get no
$ kubectl get no -o wide
$ kubectl describe no
$ kubectl get no -o yaml
$ kubectl get node --selector=[label_name]
$ kubectl get nodes -o
  jsonpath='{.items[*].status.addresses
  [?(@.type="ExternalIP")].address}'
$ kubectl top node [node_name]
```

Pods

```
$ kubectl get po
$ kubectl get po -o wide
$ kubectl describe po
$ kubectl get po --show-labels
$ kubectl get po -l app=nginx
$ kubectl get po -o yaml
$ kubectl get pod [pod_name] -o yaml
--export
$ kubectl get pod [pod_name] -o yaml
--export > nameoffile.yaml
$ kubectl get pods --field-selector
status.phase=Running
```

Namespaces

```
$ kubectl get ns
$ kubectl get ns -o yaml
$ kubectl describe ns
```

Deployments

```
$ kubectl get deploy
$ kubectl describe deploy
$ kubectl get deploy -o wide
$ kubectl get deploy -o yaml
```

Services

```
$ kubectl get svc
$ kubectl describe svc
$ kubectl get svc -o wide
$ kubectl get svc -o yaml
$ kubectl get svc --show-labels
```

DaemonSets

```
$ kubectl get ds
$ kubectl get ds --all-namespaces
$ kubectl describe ds [daemonset_name] -n
[namespace_name]
$ kubectl get ds [ds_name] -n [ns_name] -o
yaml
```

Events

```
$ kubectl get events
$ kubectl get events -n kube-system
$ kubectl get events -w
```

Logs

```
$ kubectl logs [pod_name]
$ kubectl logs --since=1h [pod_name]
$ kubectl logs --tail=20 [pod_name]
$ kubectl logs -f -c [container_name]
[pod_name]
$ kubectl logs [pod_name] > pod.log
```

Service Accounts

```
$ kubectl get sa
$ kubectl get sa -o yaml
$ kubectl get serviceaccounts default -o
yaml > ./sa.yaml
$ kubectl replace serviceaccount default -f
./sa.yaml
```

ReplicaSets

```
$ kubectl get rs
$ kubectl describe rs
$ kubectl get rs -o wide
$ kubectl get rs -o yaml
```

Roles

```
$ kubectl get roles --all-namespaces
$ kubectl get roles --all-namespaces -o yaml
```

Secrets

```
$ kubectl get secrets
$ kubectl get secrets --all-namespaces
$ kubectl get secrets -o yaml
```

ConfigMaps

```
$ kubectl get cm
$ kubectl get cm --all-namespaces
$ kubectl get cm --all-namespaces -o yaml
```

Ingress

```
$ kubectl get ing
$ kubectl get ing --all-namespaces
```

PersistentVolume

```
$ kubectl get pv
$ kubectl describe pv
```

PersistentVolumeClaim

```
$ kubectl get pvc
$ kubectl describe pvc
```



Linux Academy

<http://linuxacademy.com>