KPLABS Course

Certified Kubernetes Application Developer

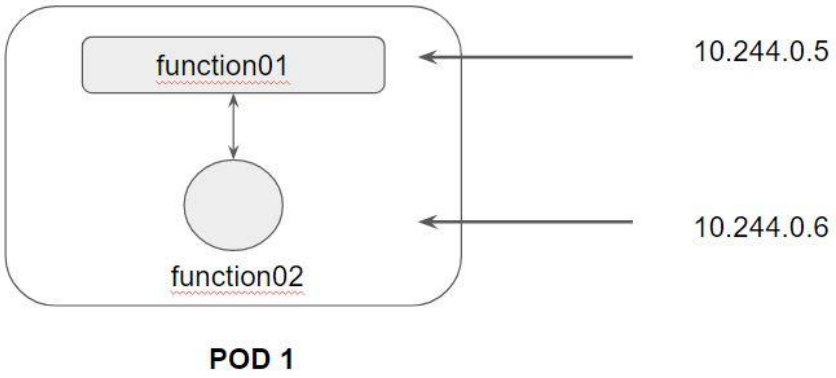
Services and Networking

ISSUED BY
Zeal Vora

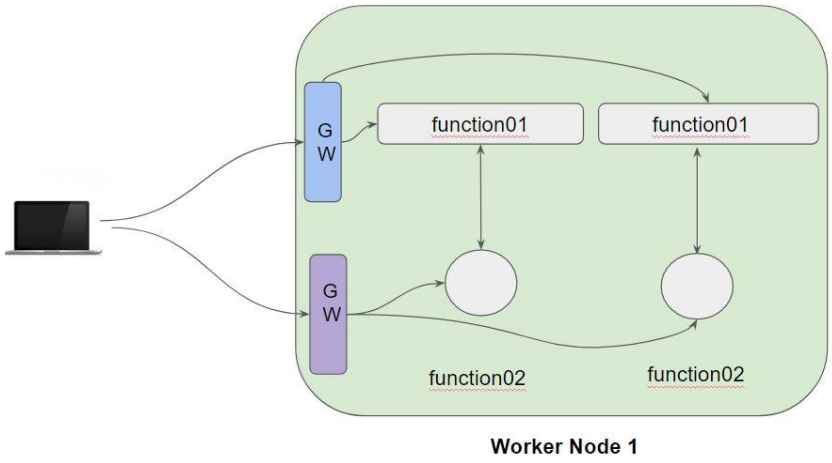
REPRESENTATIVE
instructors@kplabs.in

Module 1: Overview of Service

Whenever you create a Pod, the containers created will have Private IP addresses.



Following is a high-level diagram on the functionality of Service:



In a Kubernetes cluster, each Pod has an internal IP address.

Pods are generally ephemeral, they can come and go anytime.

We can make use of service which acts as a gateway and can get us connected with right set of pods.

Service is an abstract way of exposing application running in the pods as a network service.

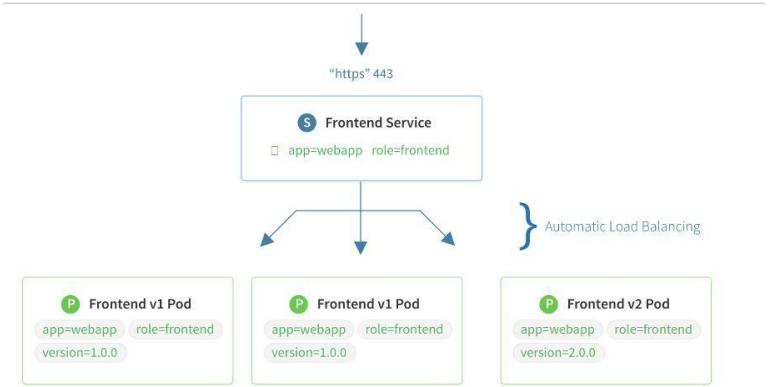
There are several types of Kubernetes Services which are available:

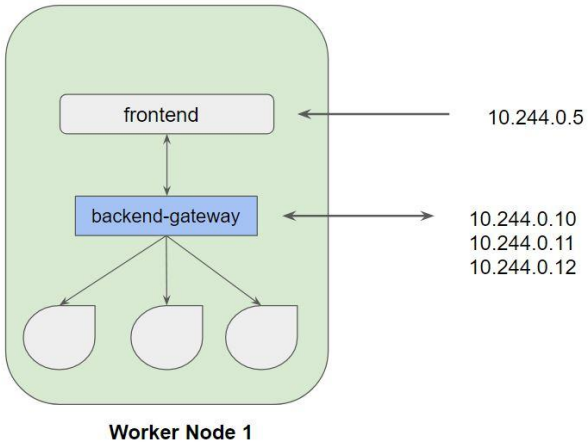
- NodePort
- ClusterIP
- LoadBalancer
- ExternalName

Module 2: Creating our first Service and Endpoint

Kubernetes Service can act as an abstraction which can provide a single IP address and DNS through which pods can be accessed.

Endpoints track the IP address of the objects that service can send traffic to.





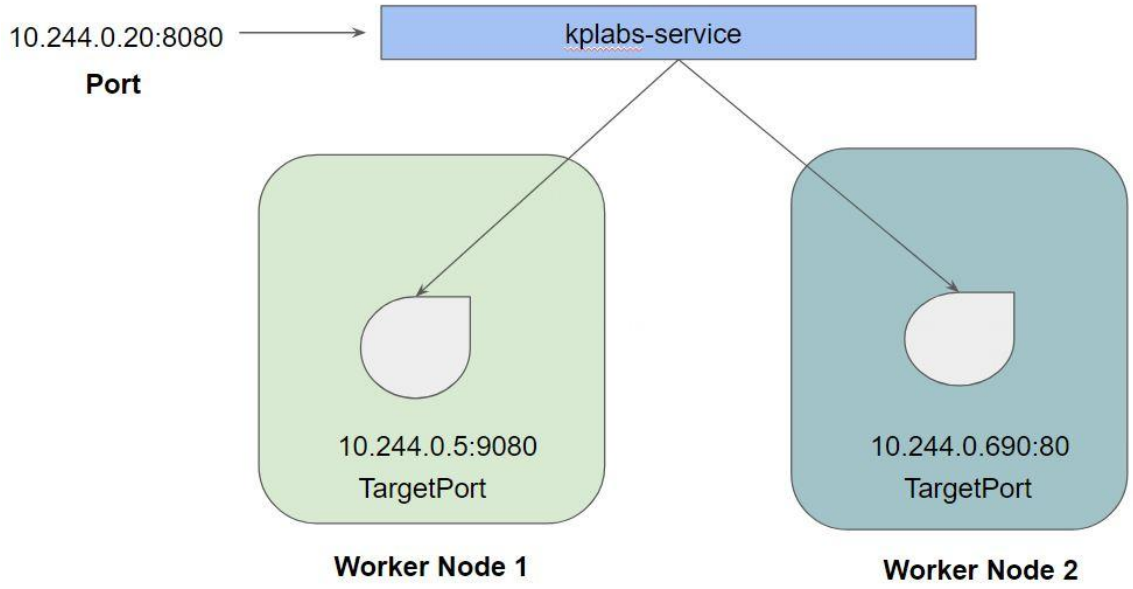
Module 3: Service Type - ClusterIP

Whenever the service type is ClusterIP, an internal cluster IP address is assigned to the service.

Since an internal cluster IP is assigned, it can only be reachable from within the cluster.

This is a default ServiceType.

Module 4: Port vs TargePort

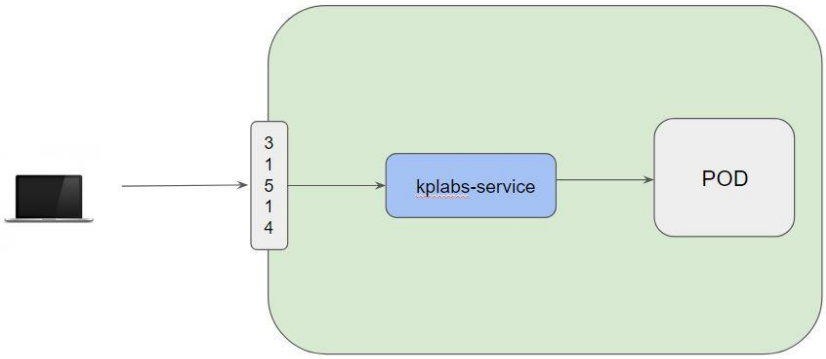


Module 5: Service Type - NodePort

From the name, we can identify that it has to do with opening a port on the nodes.

If the service type is NodePort, then Kubernetes will allocate a port (default: 30000-32767) on every worker node.

Each node will proxy that port into your service.



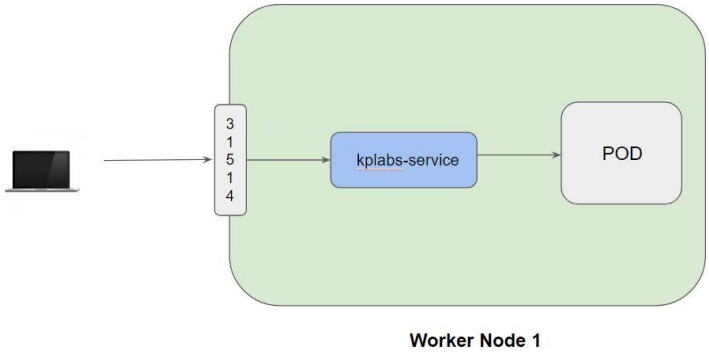
Module 6: Service Type - LoadBalancer

6.1 Challenges with NodePort

We know that NodePort ServiceType will assign a node in all the worker node which can forward the traffic to the underlying service.

Challenge in NodePort: We need to access it via IP/DNS:Port

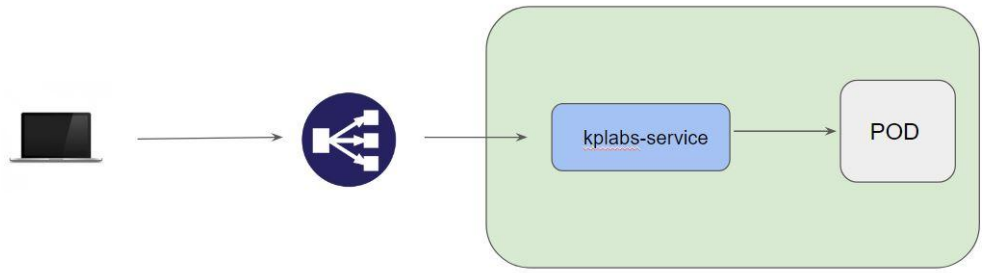
Example: google.com:31514



6.2 Understanding LoadBalancer Service Type

LoadBalancer Service Type will automatically deploy an external load balancer.

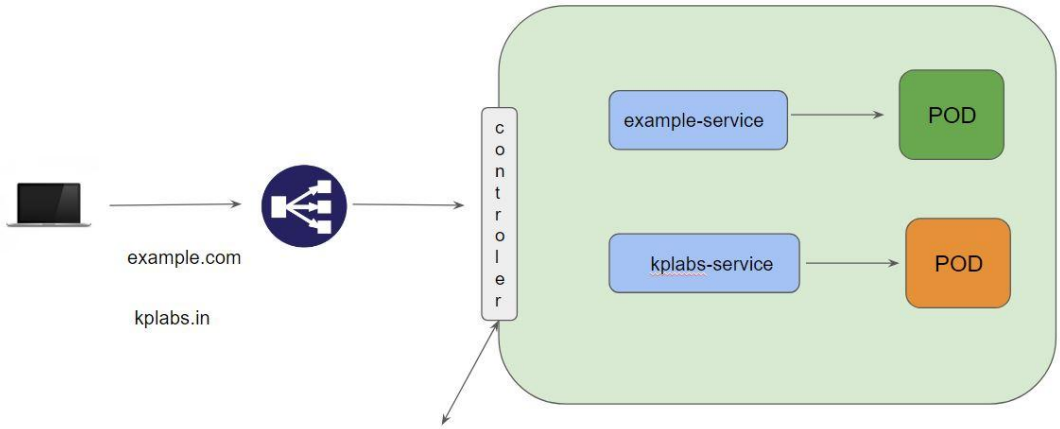
This load balancer takes care of routing requests to the underlying service.



6.3 Important Pointers

The overall implementation of LoadBalancer depends on your Cloud Provider.

If you plan to use it in bare-metal, then you will have to provide your own load balancer implementation.



Rule	Host	Service
1	example.com	example-service
2	kplabs.in	kplabs-service

Module 7: Network Policies

Understanding the Challenge:

By default, pods are non-isolated; they accept traffic from any source.

Example:

- Pod1 can communicate with Pod 2.
- Pod 1 in namespace DEV can communicate with Pod 3 in namespace Staging.

Overview of Network Policy

A network policy is a specification of how groups of pods are allowed to communicate with each other and other network endpoints.

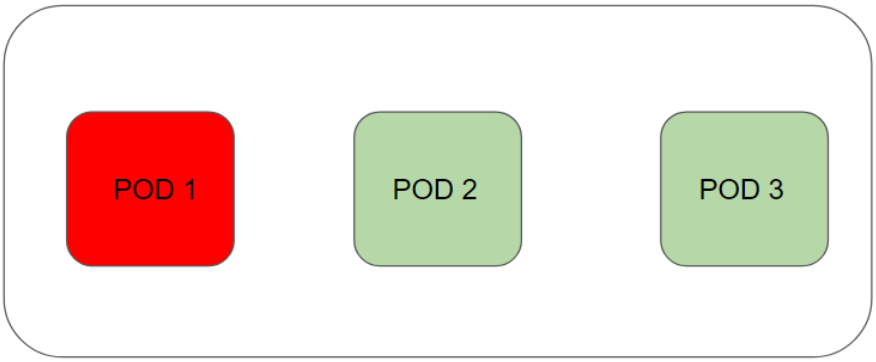
Example:

- POD 1 can only communicate with Pod 5 in the same namespace.
- POD 2 can only communicate with Pod 10 residing in namespace Security.
- No one should be able to communicate with Pod 3.

Use-Case 1 - Pod Compromise

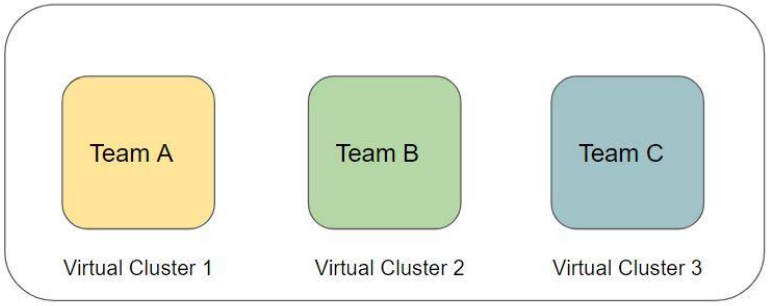
There is a POD named AppA which is compromised.

Security Policy says that the pod should be isolated so that no communication happens.



Module 8: Namespace

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



Following is the list of namespaces that are available in Kubernetes:

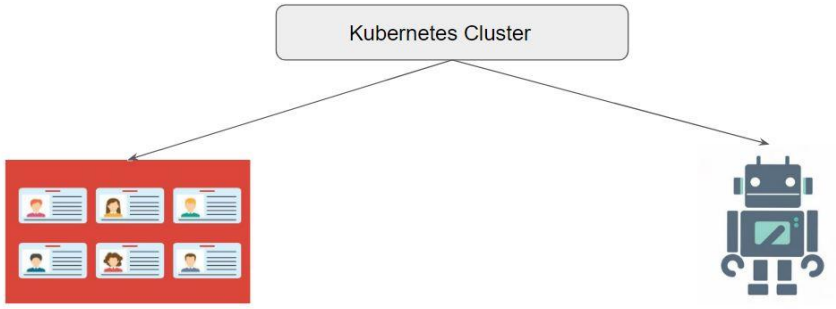
Namespace	Description
default	The default namespace for objects with no other namespace
kube-system	The namespace for objects created by the Kubernetes system
kube-public	This namespace is created automatically and is readable by all users. It contains information, like CA, that helps kubeadm join and authenticate worker nodes.
kube-node-release	The kube-node-lease namespace contains lease objects that are used by kubelet to determine node health.

Module 9: Service Accounts

13.1 Understanding Authentication

Kubernetes Clusters have two categories of users:

- Normal Users.
- Service Accounts

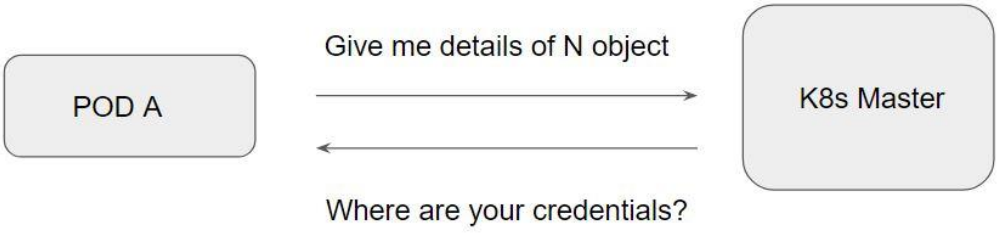


13.2 Understanding Service Accounts

Service Accounts allows the Pods to communicate with the API Server

Let's understand with a use-case:

Application within Pod A wants to retrieve an object within your K8S cluster.



Following diagram indicates the working of service account:

13.3 Important Pointer

Service Accounts are namespaced.

Default service account gets automatically created when you create a namespace

PODS are automatically mounted with the default service accounts.