

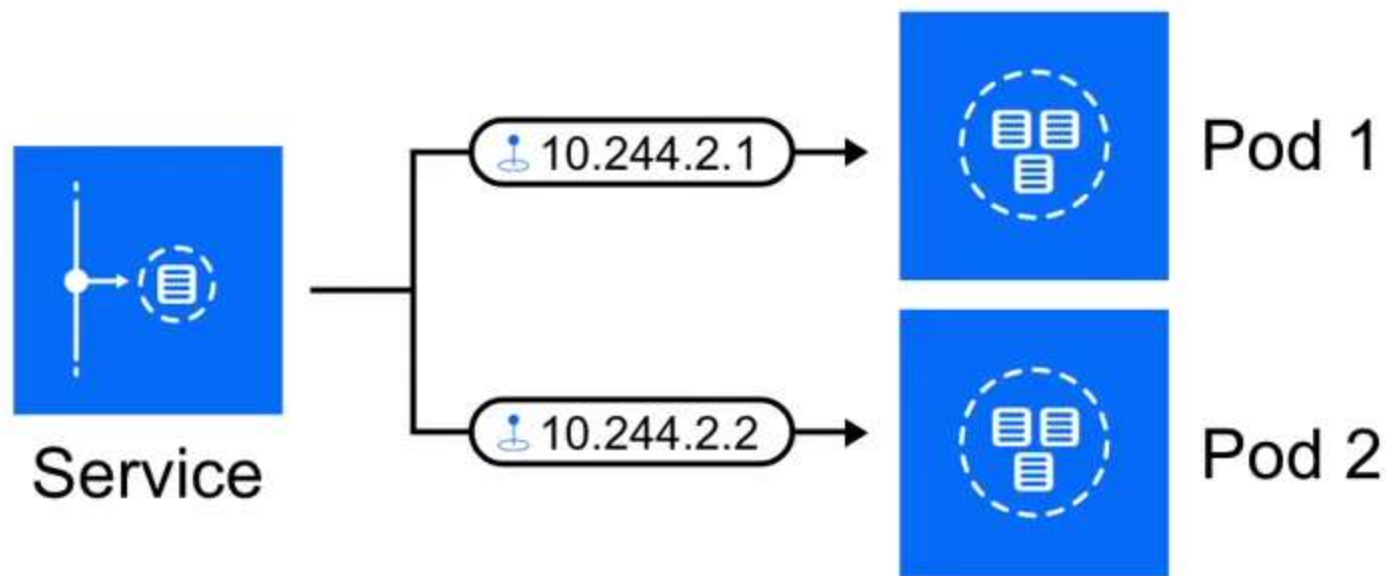
# Kubernetes Services

Cheat sheets, Practice Exams and Flash cards 🖱️ [www.examprompro.co/kcna](http://www.examprompro.co/kcna)

Kubernetes Service allows you to attach a **static IP address** and DNS name for a set of pods

A K8s Service **allows you to persist an address** for a pod even if it dies.

K8s Service also acts a load balancer



A Pod without a service will have a dynamic IP address.  
So when the pod dies so does the IP address.

Kubernetes Services have the following **service types**:

- **ClusterIP** – default, randomly forward traffic to any pod set with target port
- **Headless** – send traffic to very specific pod, when you have stateful pods eg. Database
- **NodePort** – external service, allows you to use worker node IP address
- **LoadBalancer** – similar to nodeport except leverages Cloud Service Provider's (CSPs) load balancer
- **ExternalName** — a special service that does not have selectors and uses DNS names instead

# Kubernetes Services – Traffic Policies

Cheat sheets, Practice Exams and Flash cards 🖱️ [www.examprompro.co/kcna](http://www.examprompro.co/kcna)

K8s Service allows you set a Traffic policies to determine how ingress traffic is routed.

There are 2 types of Traffic policies

## 1. External Traffic Policy

how traffic from external sources is routed and has two valid values:

- Cluster – route external traffic to all ready endpoints
- Local - only route to ready node-local endpoints

## 2. Internal Traffic Policy

how traffic from internal sources is routed (has the same two values as External)



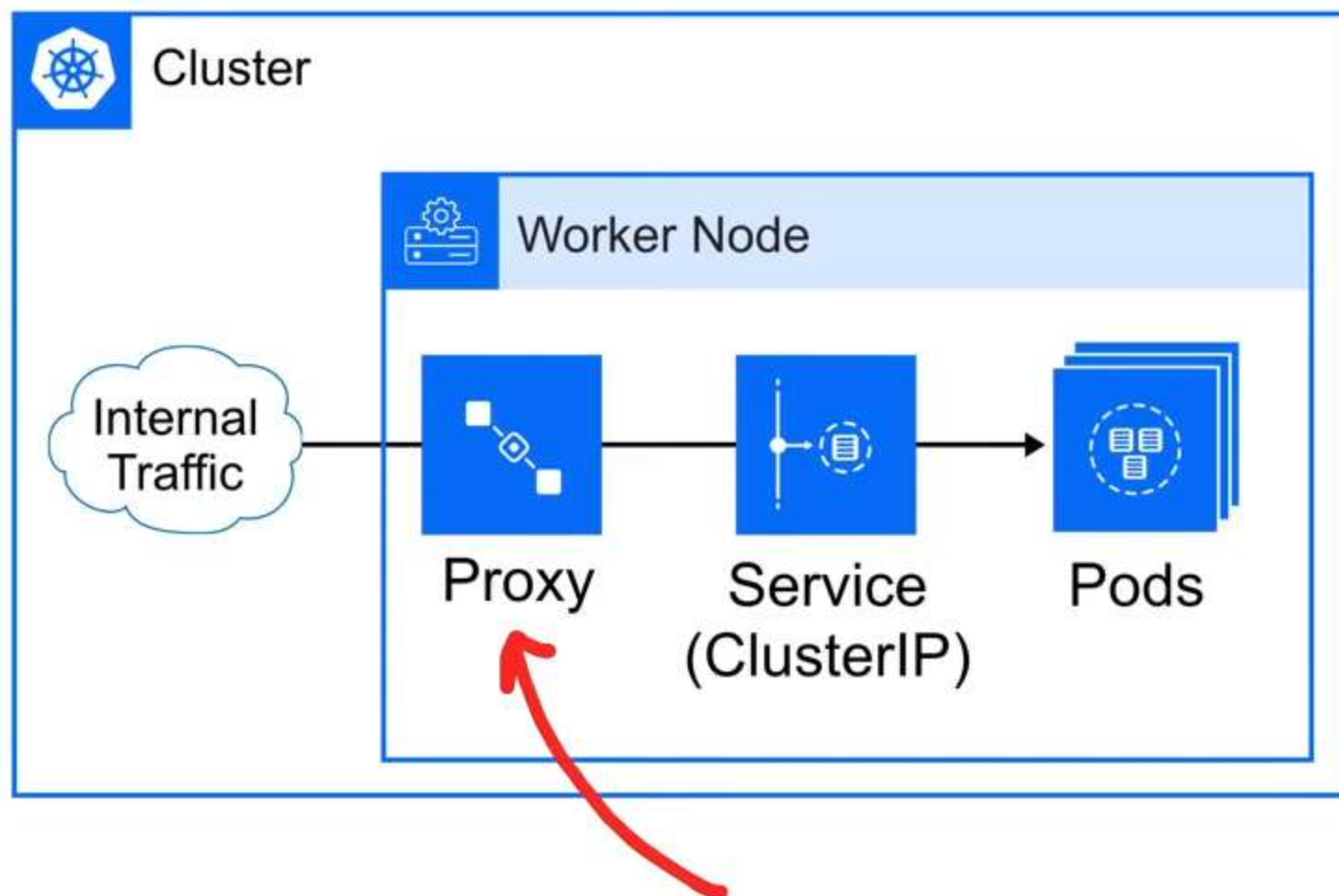
If the traffic policy is Local and there are no node-local endpoints, then kube-proxy does not forward any traffic for the relevant Service



# ClusterIP

Cheat sheets, Practice Exams and Flash cards 🖱️ [www.examprompro.co/kcna](http://www.examprompro.co/kcna)

**ClusterIP** is the default service type for a K8 service.  
It is used for internal traffic. External traffic will not reach the service.  
Traffic will be randomly distributed to any targeted pods.



If you **omit type** it will default to ClusterIP

```
apiVersion: v1
kind: Service
metadata:
  name: my-internal-service
spec:
  selector:
    app: my-app
  type: ClusterIP
  ports:
    - name: http
      port: 80
      targetPort: 80
      protocol: TCP
```

Traffic originating from within the cluster will pass through the Node's **Kubernetes Proxy** and then onto to Kubernetes Service



A service can span multiple worker nodes for cross-node pods.

When to use ClusterIP:

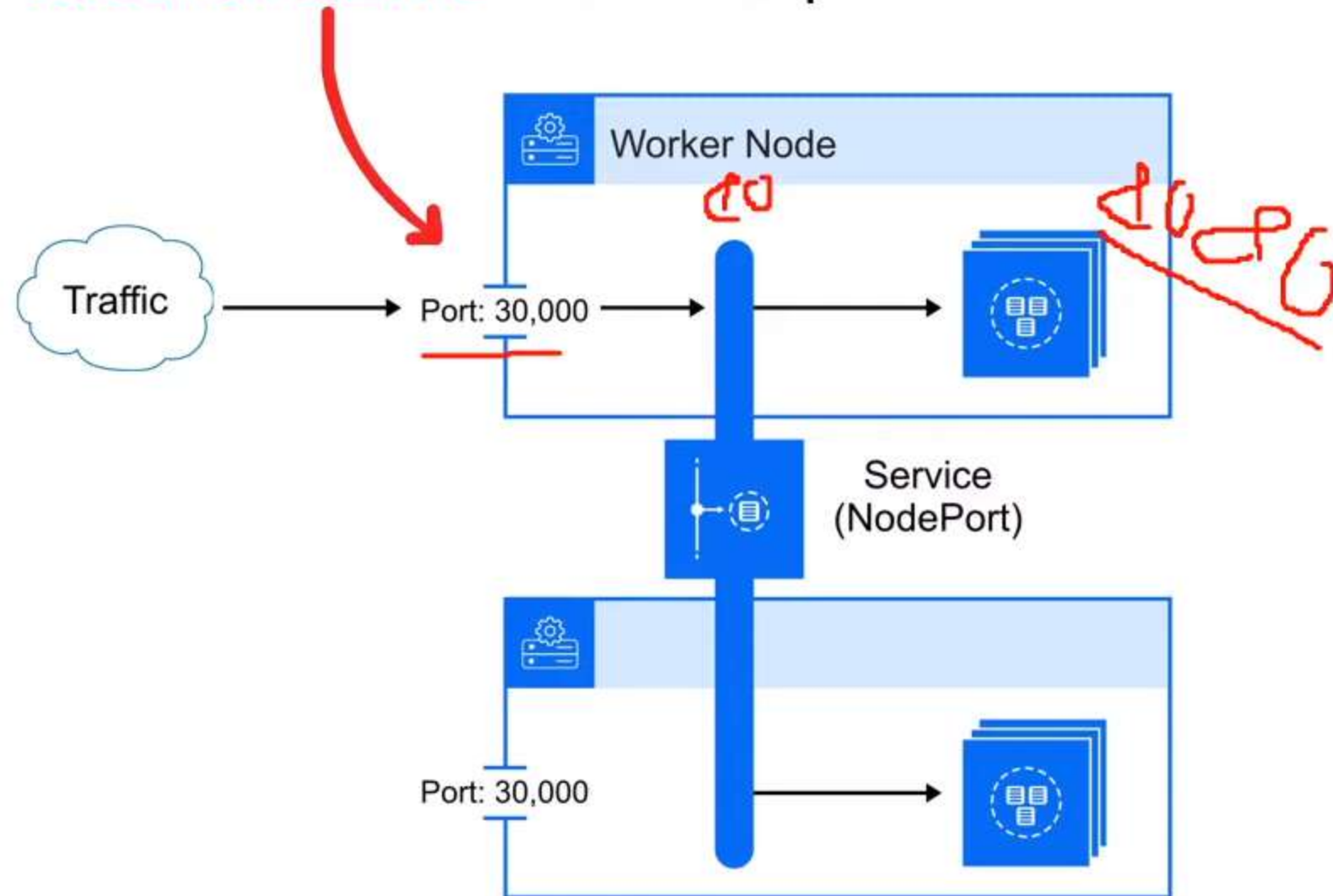
- Debugging
- Testing
- Internal traffic
- Internal Dashboards

# NodePort

Cheat sheets, Practice Exams and Flash cards 🖱️ [www.examprompro.co/kcna](http://www.examprompro.co/kcna)

**NodePort** allows you to expose a port for Virtual Machines running pods that the Service is managing.

**External traffic** can reach pods



Set **type** to NodePort

Set **nodePort**

```
apiVersion: v1
kind: Service
metadata:
  name: my-nodeport-service
spec:
  selector:
    app: my-app
  type: NodePort
  ports:
    - name: http
      port: 80
      targetPort: 80
      nodePort: 30036
      protocol: TCP
```



There is no external load balancer so NodePort is intended for a single Kubernetes Service and for non production workloads.



# NodePort

Cheat sheets, Practice Exams and Flash cards 🖱️ [www.examprompro.co/kcna](http://www.examprompro.co/kcna)

## Port

Port exposes the Kubernetes service on the specified port within the cluster.  
Other pods within the cluster can communicate with this server on the specified port.

## TargetPort

Target Port is the port on which the service will send requests to, that your pod will be listening on. Your application in the container will need to be listening on this port also.

## NodePort

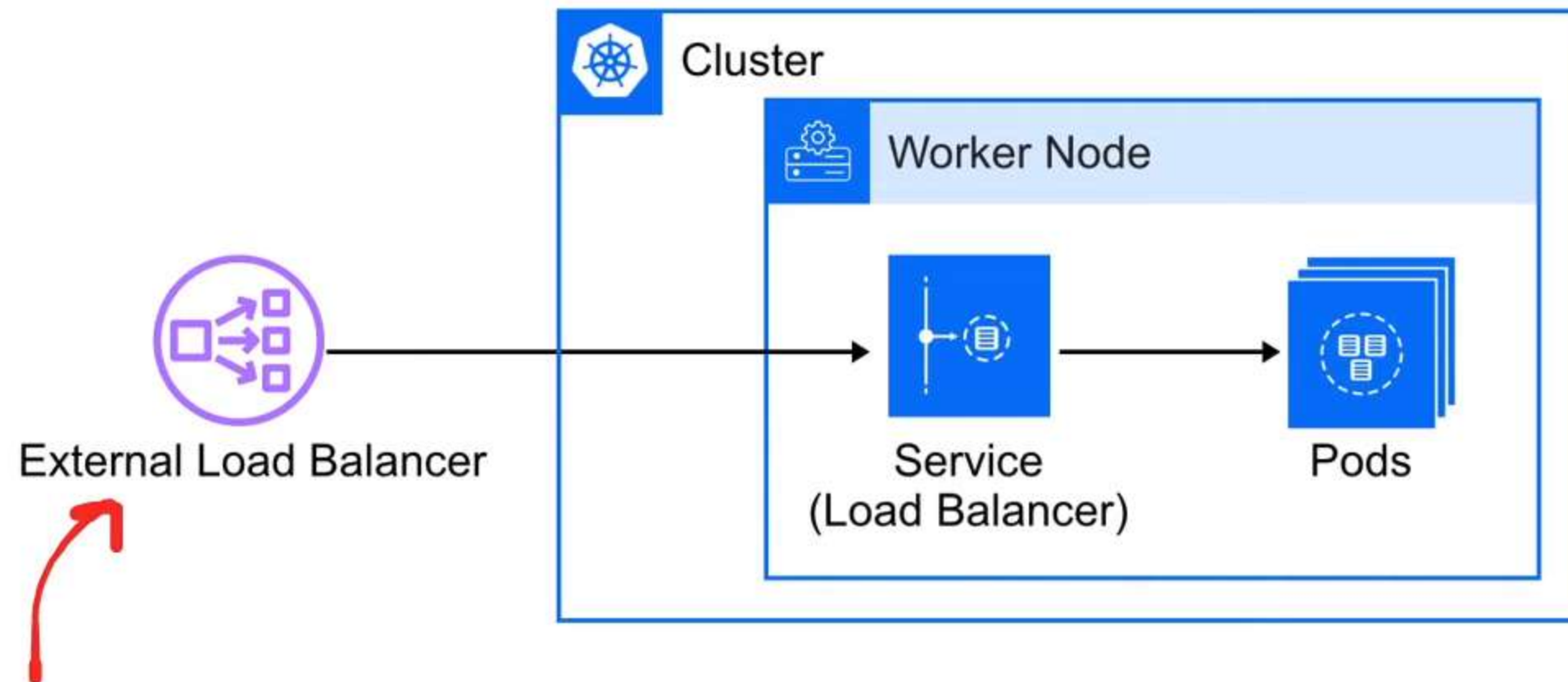
NodePort exposes a service externally to the cluster by means of the target nodes IP address and the NodePort. NodePort is the default setting if the port field is not specified.

```
apiVersion: v1
kind: Service
metadata:
  name: my-nodeport-service
spec:
  selector:
    app: my-app
  type: NodePort
  ports:
    - name: http
      port: 80
      targetPort: 80
      nodePort: 30036
      protocol: TCP
```

# Load Balancer

Cheat sheets, Practice Exams and Flash cards  [www.examprompro.co/kcna](http://www.examprompro.co/kcna)

A Load Balancer service type allows you to use an External Load Balancer.  
The External Load Balancer handles the routings and traffic distribution logic



An external load balancer would be from a **managed third-party cloud service**.  
Eg. AWS Network Load Balancer (NLB)



Load Balancer type is well suited for production workloads.  
Generally its recommended to use Kubernetes Ingress

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
  clusterIP: 10.0.171.239
  type: LoadBalancer
status:
  loadBalancer:
    ingress:
      - ip: 192.0.2.127
```

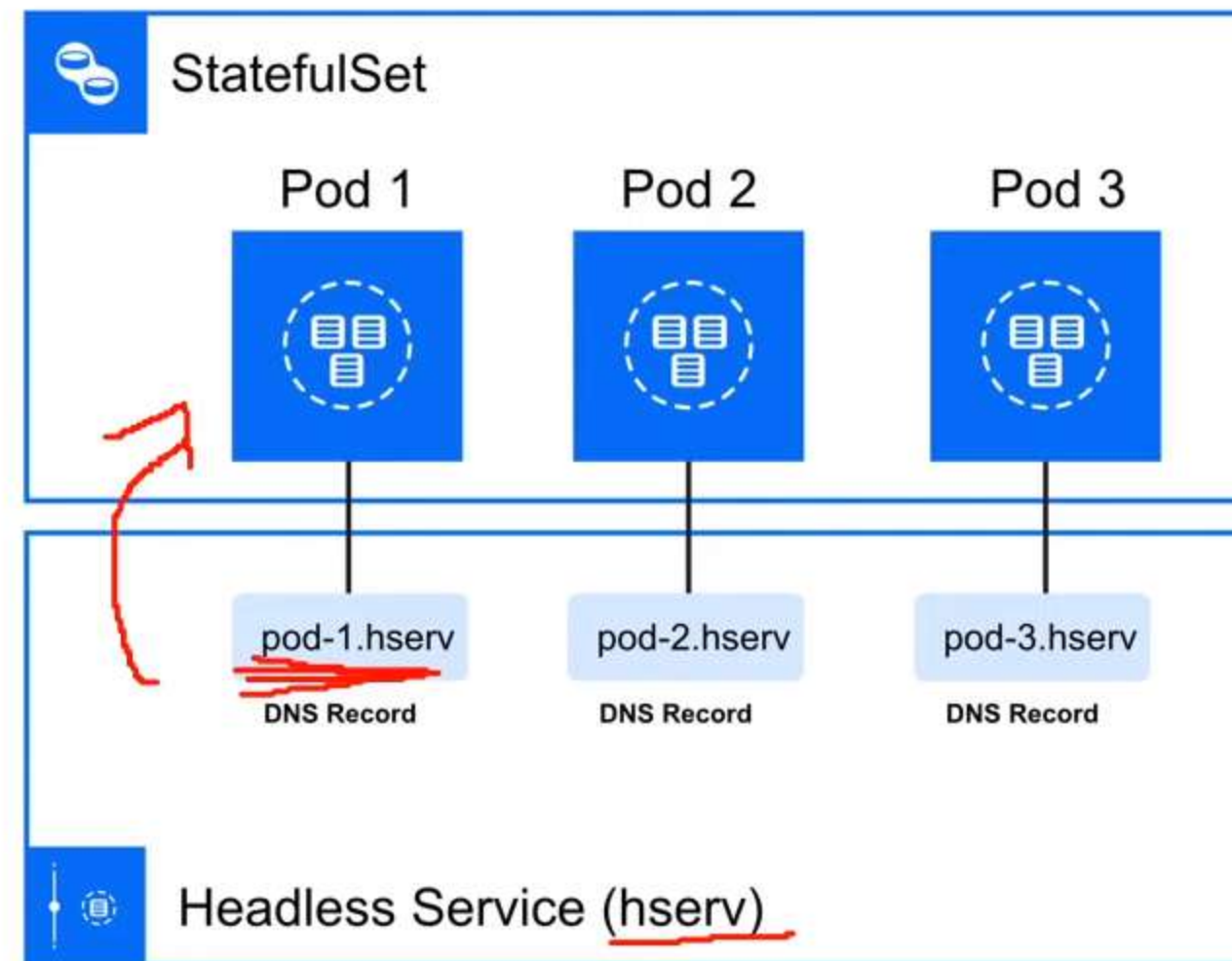


# Headless

Cheat sheets, Practice Exams and Flash cards 🖱️ [www.examprompro.co/kcna](http://www.examprompro.co/kcna)

- A headless service is a service with no ClusterIP address
- A headless service does not provide load balancing or proxying

Headless are useful when you are dealing with a Stateful application (reads and writes) and you need writes to go to a specific pod.



```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
    - port: 80
      name: web
  clusterIP: None
  selector:
    app: nginx
```

Headless Service is needed to manage network identity of the stateful pods by assigning a DNS Record to each pod so you can route traffic to a DNS Hostname.

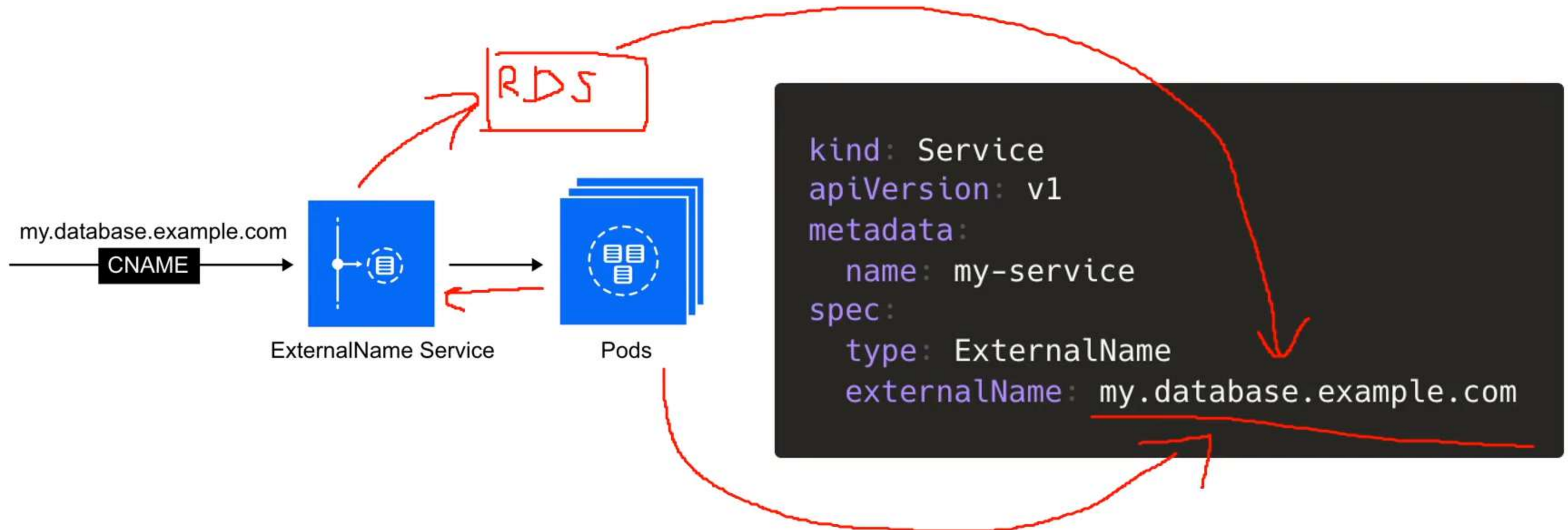
# ExternalName

Cheat sheets, Practice Exams and Flash cards 🖱️ [www.exampopro.co/kcna](http://www.exampopro.co/kcna)

ExternalName Services are the same as ClusterIP Service with the exception of instead returning a StaticIP it returns a CNAME record.

## What is a CNAME record?

Canonical Name (CName) record is a DNS record that maps one domain name (an alias) to another name (canonical name)






# KubeCTL Expose Command

Cheat sheets, Practice Exams and Flash cards 🖱️ [www.examprompro.co/kcna](http://www.examprompro.co/kcna)

Expose is used to quickly create Kubernetes Services for a **deployment**



```
kubectl expose deployment my-app --type=NodePort \  
--name=my-service --port=80 --targetport=8080 --nodeport=3000
```

# BusyBox

Cheat sheets, Practice Exams and Flash cards 🖱️ [www.examprompro.co/kcna](http://www.examprompro.co/kcna)



**BusyBox** combines tiny versions of many common UNIX utilities into a single small executable



It's The Swiss Army knife of Embedded Linux, as the single executable replaces basic functions of more than **300 common commands**

BusyBox can be used to **interactively debug Services** to ensure they are working.



```
kubectl run -it --rm --restart=Never busybox --image=gcr.io/google-containers/busybox sh
```