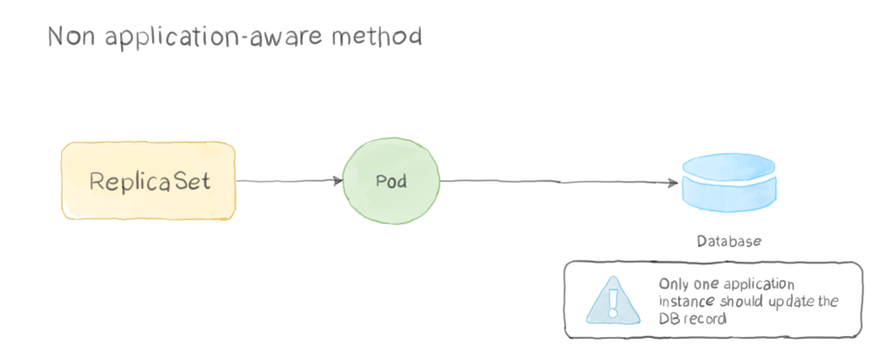
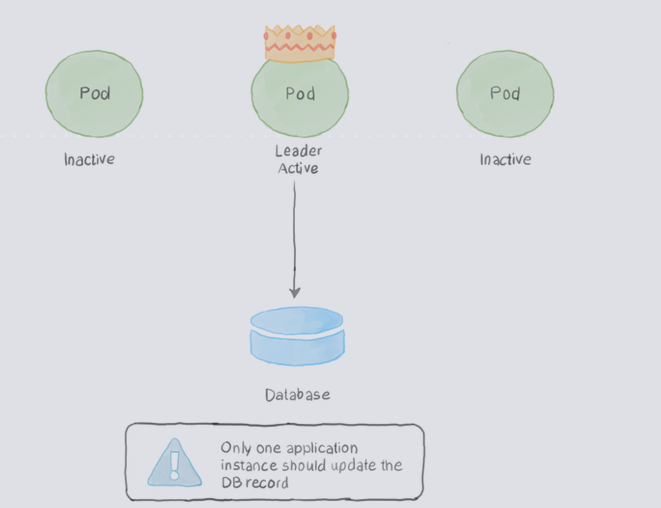
**Implementing the Singleton Pattern in Kubernetes**

This is primarily for Developer where they use ETCD API.

In a Kubernetes cluster, the default behaviour is to run and maintain several versions (replicas) of the application for high availability. A web application that runs on only one Nginx instance is vulnerable to downtime if this Nginx went down or got restarted. However, sometimes this may not best-serve your environment needs. In the microservices architecture, an application may be running on more than one component. If the application is hosted on Kubernetes, some of those components may need to follow the Singleton pattern when they run.

For example, a web application that needs to consume a message from a message queue in a sequential manner should not have more than one instance running at a time. Let’s see how we can implement the Singleton pattern in Kubernetes using the two methods that we described earlier: from within the application and from outside the application.





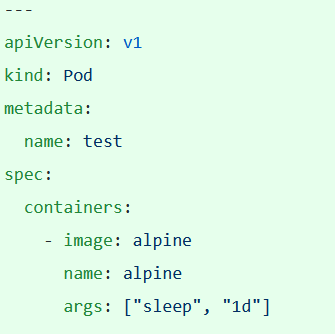
## Singleton Service

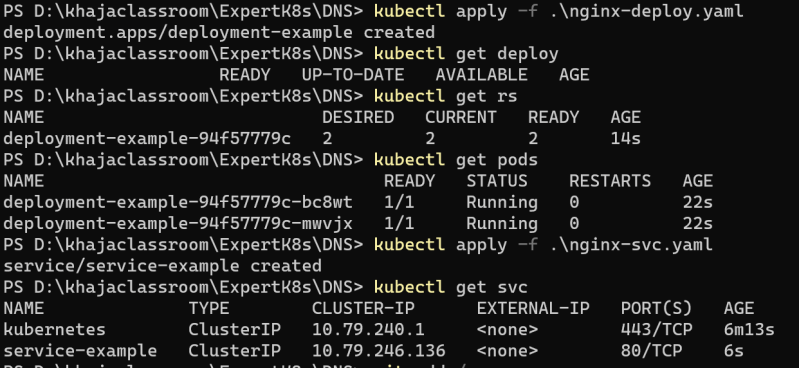
* Overview: Singleton Pattern ensures only one instance of an application is active at a time and is highly available
* Problem:
  + In some cases, only one instance of the service is allowed to run
* Solution:
  + Out of Application Locking:
    - Stateful set or replica set with replicas = 1
  + In-Application Locking
    - In a distribute environment one way to control the service instance count is through a distributed lock
    - We can implement leader election using kubernetes api’s. For example apache camel has a K8s connector that also provides leader election and Singleton capabilities
    - This connector access the etcd api directly and k8s api to use Config Maps to acquire a distributed lock [Refer Here](https://github.com/apache/camel/tree/main/components/camel-kubernetes/src/main/java/org/apache/camel/component/kubernetes/cluster)
    - [Refer Here](https://kubernetes.io/blog/2016/01/simple-leader-election-with-kubernetes/) for sample leader election in k8s.

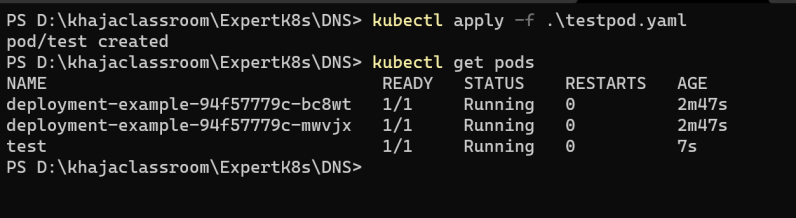
## DNS For Services and Pods

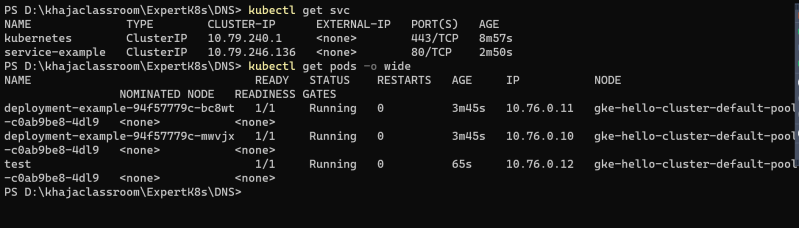






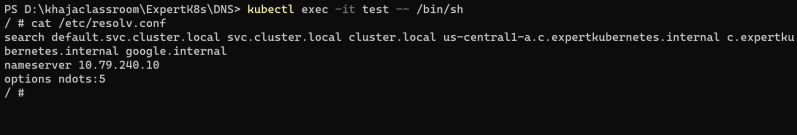


Lets create the test pod 

Make a note of service names and ip addresses of pods 

Now login into test pod

kubectl exec -it test -- /bin/sh

Look into resolve.conf 

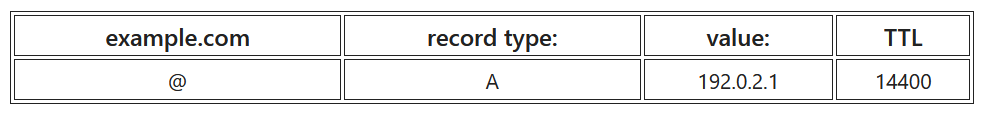
What objects in k8s gets DNS records Services and Pods

## DNS A record Sample: Google.com = 8.8.8.8

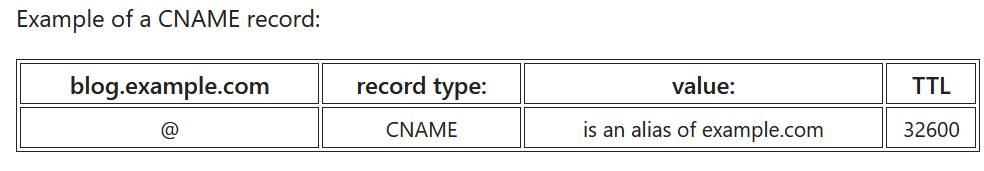
The ‘A’ stands for ‘address’ and this is the most fundamental type of [DNS](https://www.cloudflare.com/learning/dns/what-is-dns/) record: it indicates the [IP address](https://www.cloudflare.com/learning/dns/glossary/what-is-my-ip-address/) of a given [domain](https://www.cloudflare.com/learning/dns/glossary/what-is-a-domain-name/). For example, if you pull the DNS records of cloudflare.com, the A record currently returns an IP address of: 104.17.210.9.

A record only hold IPv4 addresses. If a website has an IPv6 address, it will instead use an ‘AAAA’ record.

Here is an example of an A record:

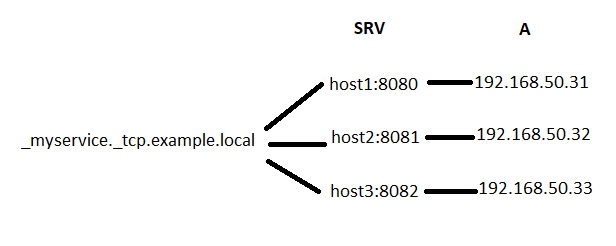


DNS CNAME (Here alias name is also specified like- fb.com, facebook.com)



## What is a DNS SRV record?

The [DNS](https://www.cloudflare.com/learning/dns/what-is-dns/) ‘service’ (SRV) record specifies a host and port for specific services such as voice over IP (VoIP), instant messaging, and so on. Most other [DNS records](https://www.cloudflare.com/learning/dns/dns-records/) only specify a server or an [IP address](https://www.cloudflare.com/learning/dns/glossary/what-is-my-ip-address/), but SRV records include a port at that IP address as well. Some Internet protocols require the use of SRV records in order to function.



* K8s Services:
  + A/AAAA records
    - Normal Services are assigned a DNS A record depending on the IP family of the service for the name in the form of my-svc.my-namespace.svc.cluster-domain.example
  + SRV records:
    - SRV records are created for each named ports that are part of normal or Headless Services. For each name port the SRV record would have the form \_my-port-name.\_my-port-protocol.my-svc.my-namespace.svc.cluster.local
* K8S Pods
  + A/AAAA records:
    - Pod has the following dns resolution
  + pod-ip-address.my-namespace.pod.cluster.local
* In kubernetes in addition to DNS records k8s create environmental variables with service information