Kubernetes-LoadBalancer

Here is a summary of Load Balancer and Service Discovery in Kubernetes, along with an example:

What is Load Balancing?

Load balancing is a technique used to distribute network traffic across multiple servers to improve responsiveness, reliability, and scalability. In the context of Kubernetes, load balancing allows you to expose your application to the outside world by assigning a single IP address that can be accessed from anywhere.

How does Service Discovery work in Kubernetes?

In Kubernetes, Service Discovery is a mechanism that enables communication between services running within the cluster. When a pod (a collection of containers) is created and exposed as a service, Kubernetes creates an entry in the DNS system that allows other pods to communicate with it using its name.

Types of Load Balancers:

There are two main types of load balancers:

1. **Layer 4**: This type of load balancer operates at the transport layer (TCP/UDP) and directs incoming traffic to a specific node in the cluster.
2. **Layer 7**: Also known as an application delivery controller, this type of load balancer can inspect HTTP headers and make decisions based on that information.

Service Types:

Kubernetes provides several service types for different use cases:

1. **ClusterIP**: This is the default service type, which exposes a service to other pods within the cluster.
2. **NodePort**: Exposes the service at a specific port on each node in the cluster.
3. **LoadBalancer**: This service type uses an external load balancer to route traffic to a service.

Example:

Let's say you have a pod called `web` that exposes a web server running on port 80. You create a Service object named `http-service` with the following specification:

```
 apiVersion: v1
kind: Service
metadata:
  name: http-service
spec:
  selector:
    app: web
  ports:
  - name: http
    port: 80
    targetPort: 80
  type: LoadBalancer
```

In this example:

- The service selects the pod labeled `app=web`.
- It exposes port 80 on each node in the cluster.
- The load balancer directs incoming traffic to one of the nodes running the web service.

How it works:

1. When a request is made to the external IP address of the LoadBalancer, it's routed through the load balancer to a random node in the cluster.
2. The node receives the request and sends it to the `web` pod.
3. The web pod responds to the request, and the response is sent back to the load balancer.
4. The load balancer directs the response back to the client's IP address.

This is a basic example of how Load Balancing works in Kubernetes with Service Discovery. There are many other features and configurations that can be used depending on your specific use case.

Curated by Brajesh Kumar