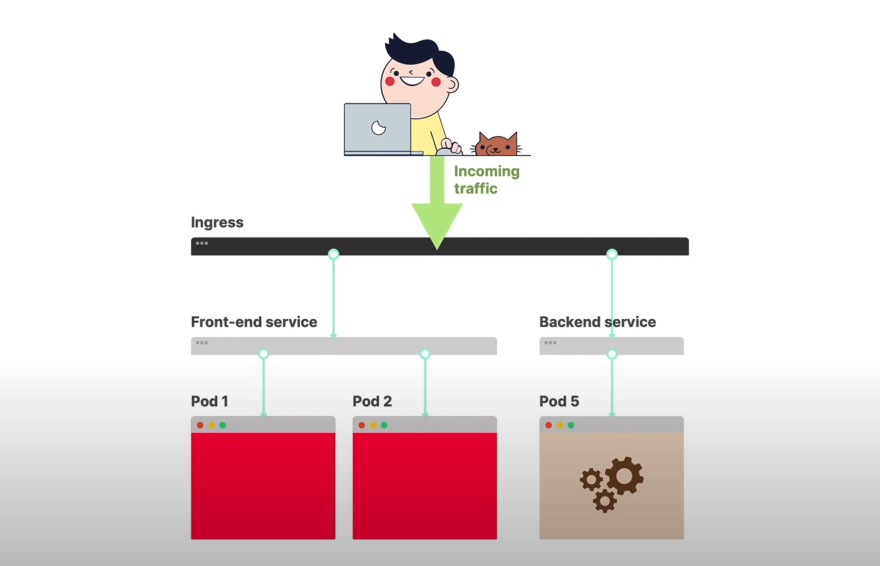
**Operating High Traffic Websites on Kubernetes**

**How can we scale automatically our resources in Kubernetes when we have high traffic on websites?**

**How do we distribute the traffic based on the requests coming in?**

In order to distribute the traffic between pods, we need to use a load balancer for the frontend and the backend know as **front-end service** and **backend service** respectively. The load balancer sends the traffic to the pods using their IP addresses. On top of that to distribute the traffic between both the services using an external load balancer which is a deployment and a service know as **Ingress**.

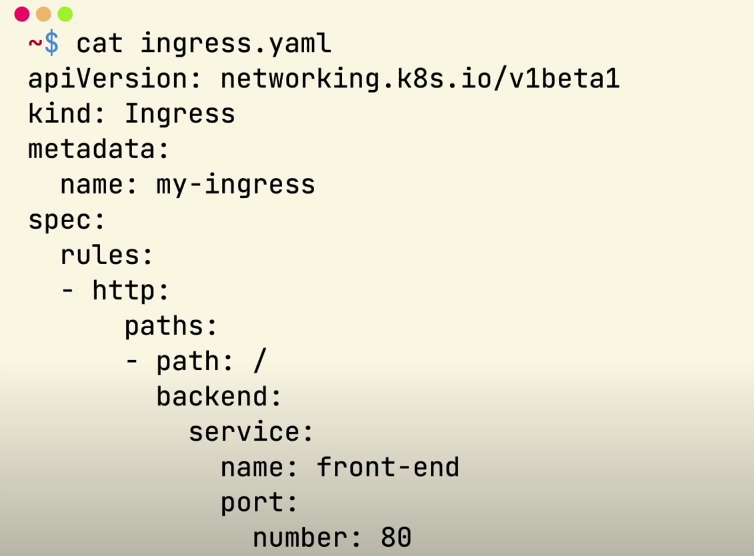


**Load Balancer**

A Load balancer is automatically provisioned. It works only if there is a cloud provider and it is for the Layer4 services only as it supports TCP and UDP. It does not work with HTTP requests. You can forward the requests based on the paths while keeping a low latency. You can apply policies such as authentication and great limiting.

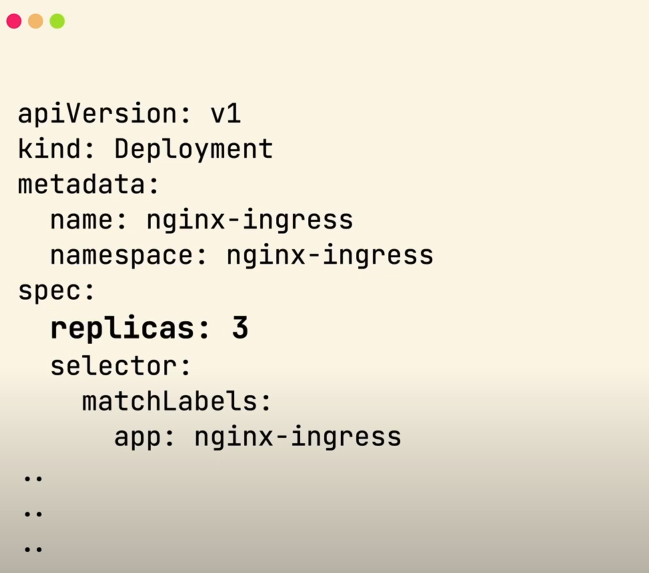
**How do we communicate with the cluster? How do we create these resources?**

Ingress itself is a programmable route. In-order to communicate and create multiple Ingress controllers you have to use a tool **kubectl** which is a command line tool. To create the resources such as replicas of the Ingress a configuration file has to be written.



**What if there were tons of traffic?**

For this scenario you will need to create multiple replicas of the Ingress to balance the traffic. You will have to set the number of replicas manually depending on the traffic.



**How do you set the number of replicas automatically depending upon the traffic?**

To set the number of Ingress controllers automatically, you will have to use a resource in Kubernetes called Horizontal Pod Autoscaler. You will query the scale-up or scale-down the metrics accordingly. The Autoscaler will increase the number of replicas whenever an instance is added or it reduces the number of requests. You will need to use an open-source project called Prometheus which is a great monitoring system to collect the metrics from pods and then stores them. In-order to automatically scale things based on requests coming in you will need to use another open-source project called KEDA (Kubernetes Event Driven Autoscaling). It allows to us to drive the auto-scaling of the Kubernetes workload based on the number of events.