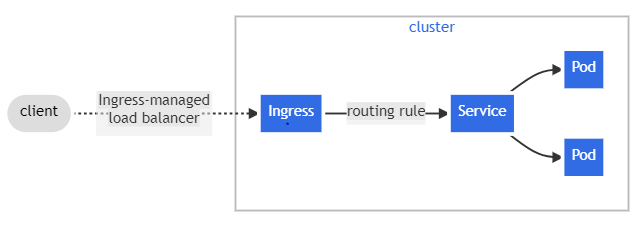
Here we can use different type of load balancers for integrating with kubernetes cluster like nginx ingress, istio etc. Here we have integrated nginx ingress load balancer to control external traffic to the application.

We have deployed a 2048 game application in kubernetes cluster

Ingress:

Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource. The Ingress concept lets you map traffic to different backends based on rules you define via the Kubernetes API. An API object that manages external access to the services in a cluster, typically HTTP. Ingress may provide load balancing, SSL termination and name-based virtual hosting.



Ingress Controllers:

An Ingress controller is a specialized load balancer for Kubernetes (and other containerized) environments. Kubernetes is the de facto standard for managing containerized applications. For many enterprises, moving production workloads into Kubernetes brings additional challenges and complexities around application traffic management. An Ingress controller abstracts away the complexity of Kubernetes application traffic routing and provides a bridge between Kubernetes services and external ones.

Ingress Resources:

An ingress is a collection of rules to allow inbound connections to the Kubernetes cluster services.It can be configured to give services externally-reachable urls, load balance traffic, terminate SSL, offer name based virtual hosting etc. Users request ingress by POSTing the Ingress resource to the Api server.

Procedure:

* Create aks cluster in azure cloud and set storage account when using cloud shell or we can access through installing azure account extension in vs code
* Now, Merge with the Kubernetes cluster using Az aks get credentials Command
* Now write 3 manifest files consisting of deployment group,service, ingress rules.
* Now deploy those in pods using kubectl apply –f “name” command.
* Now, we can install ingress controller by deploying it in cluster through official github of ingress.
* Now check whether the nginx ingress is deployed or not.
* Now, we have to use ingress controller external ip to access the application. If the application is running we can see it is successfully deployed.

