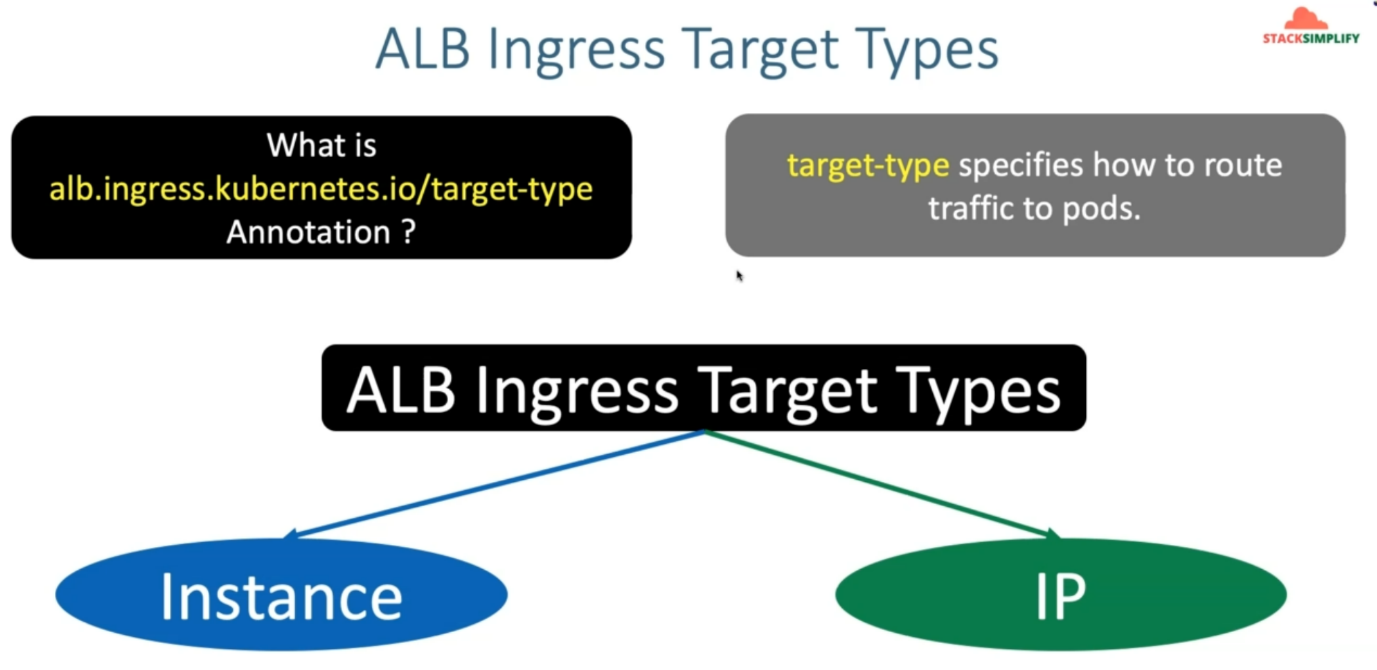
**1. Introduction to Ingress Target Type IP Mode**

--- in this section, we are going to learn about Kubernetes Ingress Concept name target type equal to IP, so let's understand what is **ALB** Ingress Kubernetes target type annotations.

**ALB ingress target types**

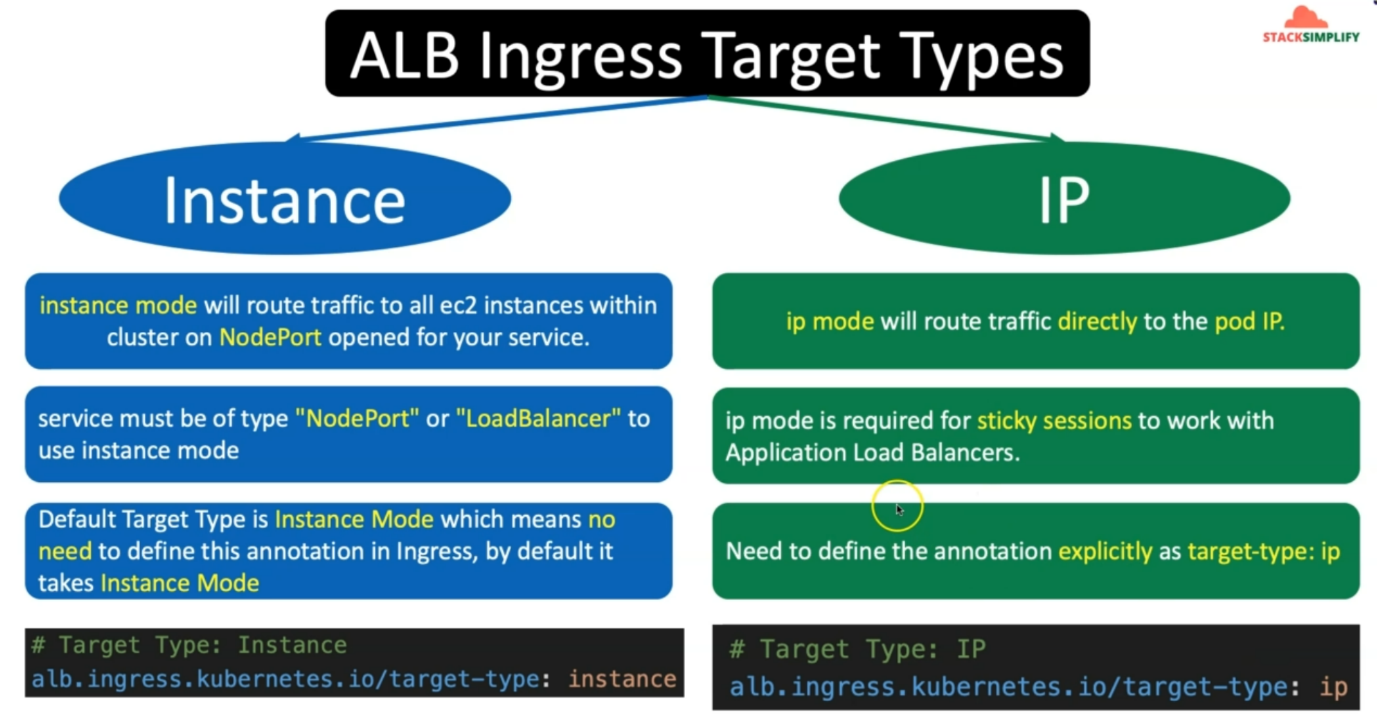


--- this target type annotations specifies, how to route traffic to your pods, which is nothing but your Kubernetes pods from your application load balancer.

--- we have ways to do that in our **ALB** ingress target types.

1. One is instance mode.
2. another one is IP mode

**instance mode**



--- let's understand the instance mode first and then we'll move to the IP mode.

--- Instance mode will route traffic to all EC2 instances within cluster on node port open for your service.

--- which means if you remember our previous examples also, we have created app1, app2, app3 deployments and we also created app1, app2 and app3 nodeport service.

--- So, the request coming to your application load balancer will go to that Kubernetes nodeport service and from that Kubernetes node port service, it is going to your application pod.

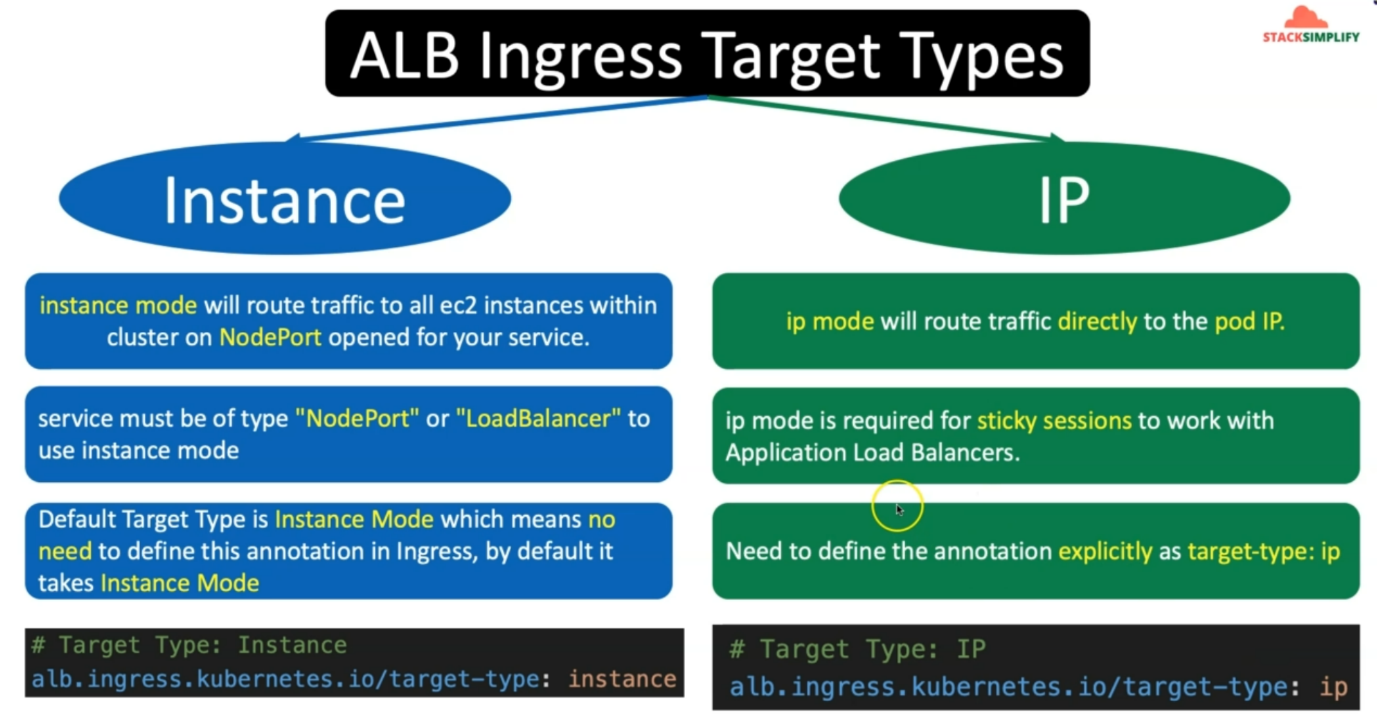
--- this Kubernetes NodePort Service will have your worker node IP colon port from your that respective node.

--- this service must be off type nodeport or load balancer to use the instance mode

--- We have implemented the nodeport service for our previous examples and default target type is instance mode in our ingress resources. So, which means we don't need to define this annotation in ingress by default it takes instance mode.

--- if you see our ingress template defined previous to this demo, nowhere, we have defined **alb.ingress.kubernetes.io/target-type: instance**. So, why we don't need to define that because the default target type is instance mode and we don't need to define this.

**IP mode**



--- now let's understand what is this target type IP mode, IP mode will route traffic directly to the pod, which means when the request comes from the internet to your application load balancer. from there, the request will directly go to your Kubernetes pod.

--- So, there is no middleman like Kubernetes NodePort Service involved here, and IP mode is required for sticky sessions to work with application load balancer.

--- So that is the core reason for implementing the IP mode, when you are using the ec2 worker nodes.

--- when you are using the Fargate way, definitely you only need to use IP mode.

--- you need to use IP mode only and need to define the annotation explicitly as target type equal to IP.