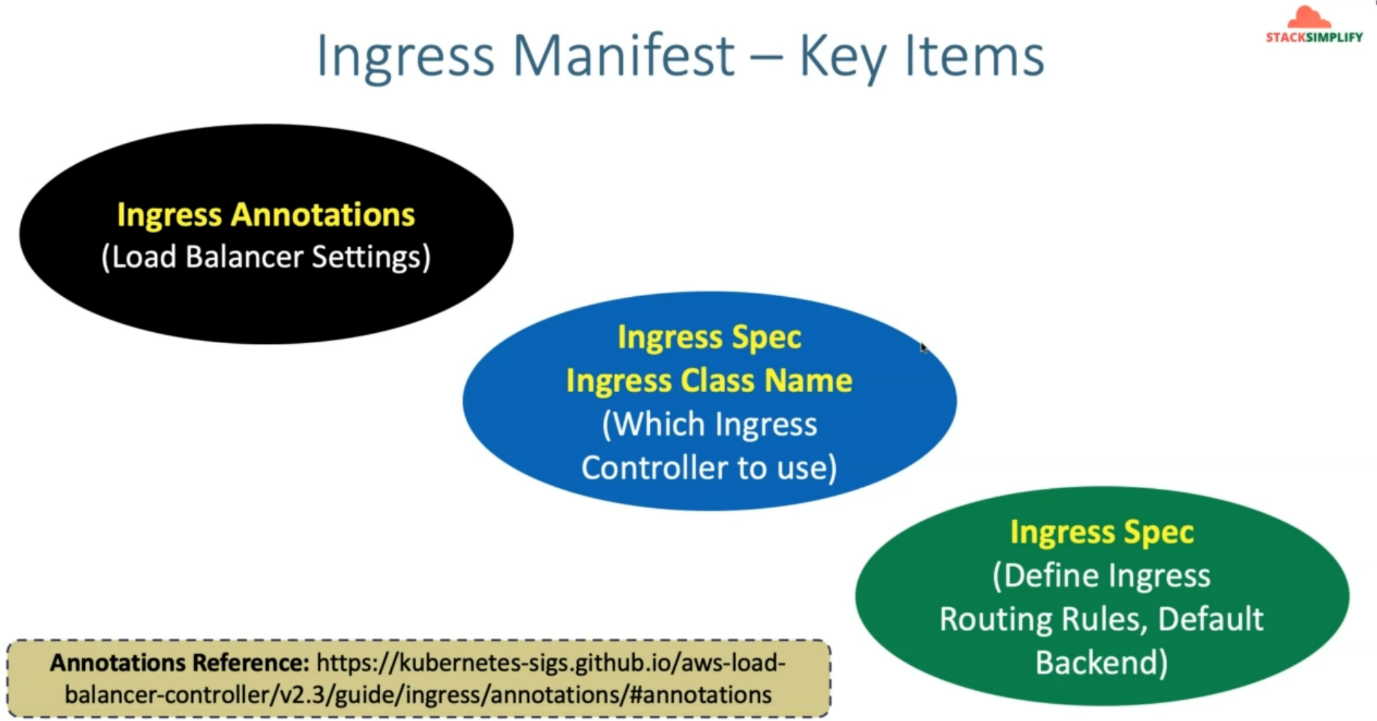
**1. Introduction to Ingress Basics**

--- **note** – the first thing we need to know about ingress is, what are the key items involved in the ingress service manifest.

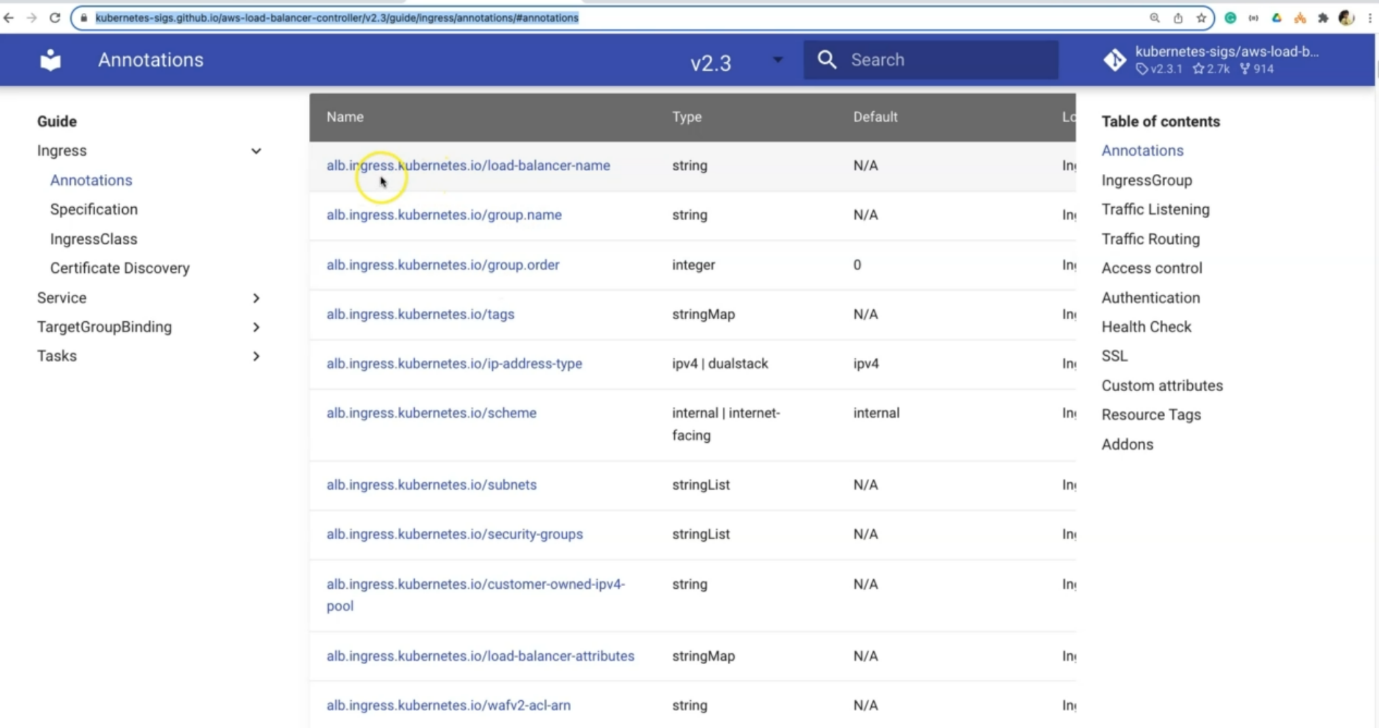


--- **ingress annotations** – ingress annotations are nothing but load balancer settings. Which we are using to provision is ingress service.

--- **ingress spec** – inside ingress, ingress spec the first item we are going to discuss is ingress class name to which this ingress service needs to be associated or belongs to.

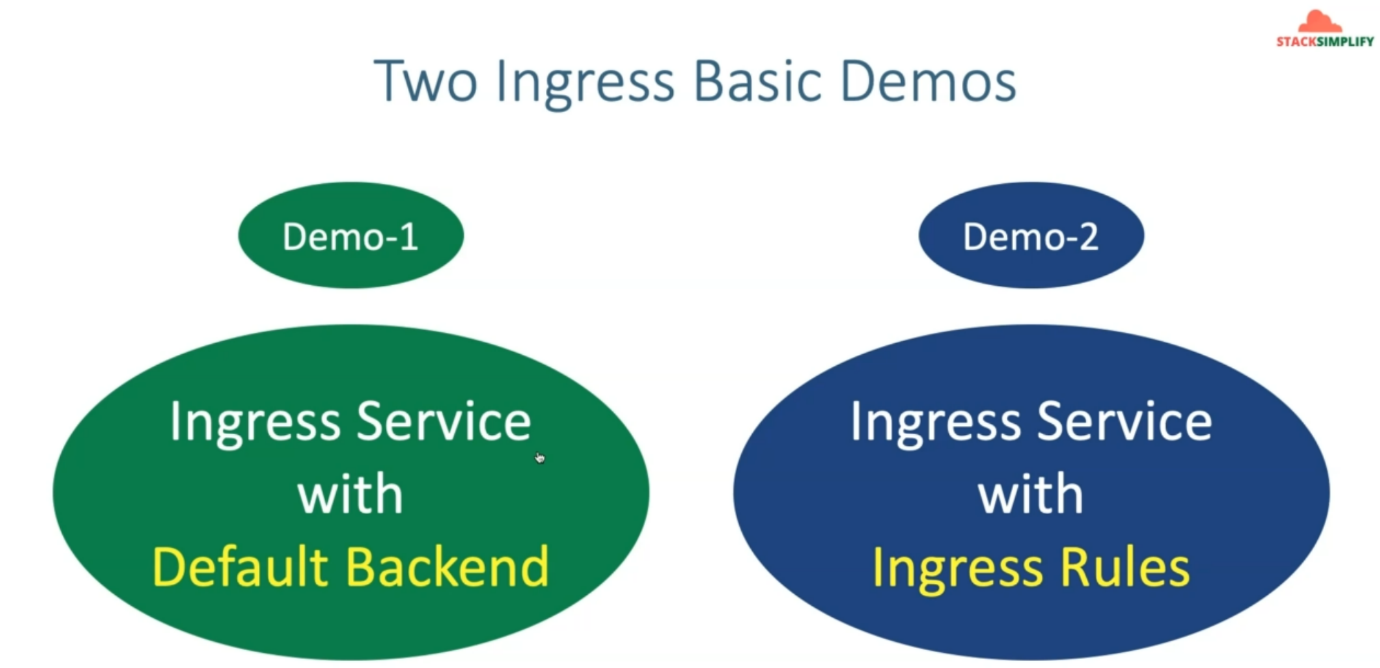
--- **note** – in ingress spec, you can additionally define ingress routing rules, default backend.

--- **note** – there are 30 plus aws kubernetes load balancer ingress annotations available as on today.



--- these are the settings, if you want to specifically deploy your subnet into particular subnet then you can do that using these annotations. Same goes to security group.

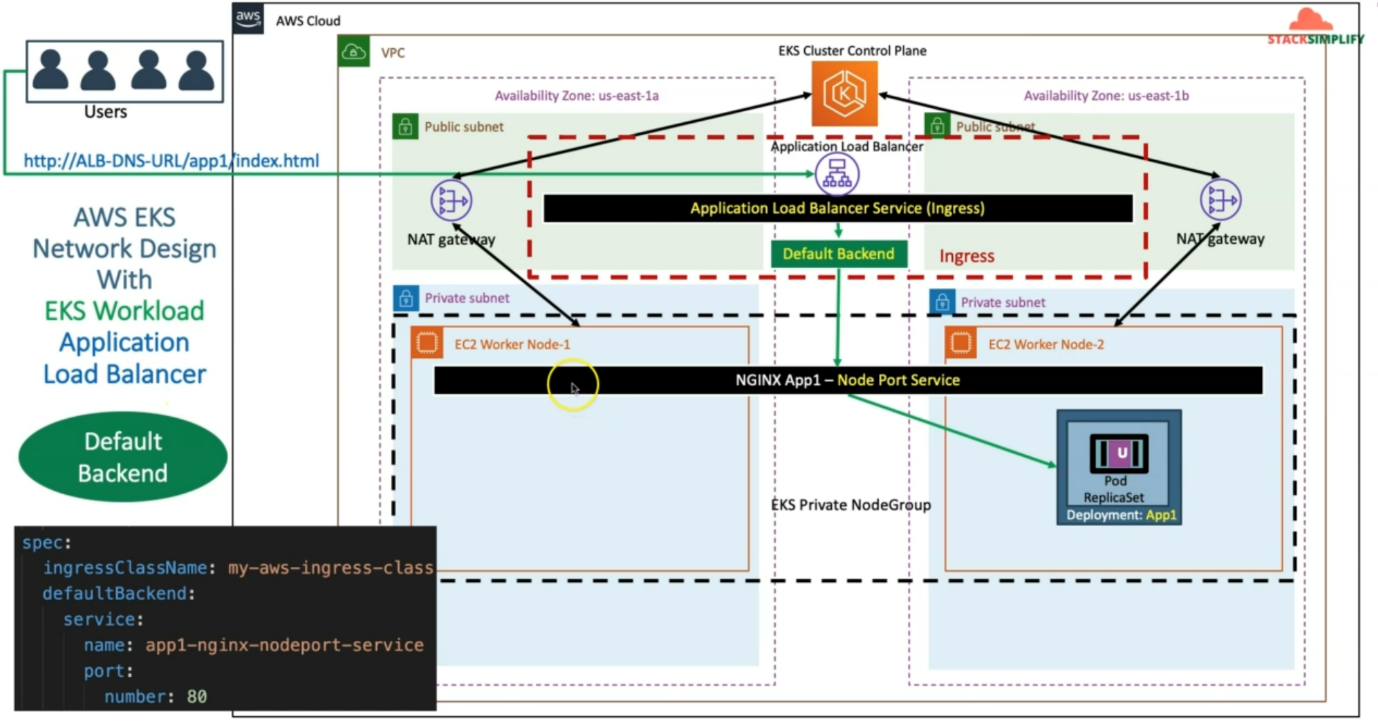
**Two ingress basic demos**



--- we are going to do 2 demos as part of this demo section

1. Ingress service default backend
2. Ingress service with ingress rules.

**Network design for default backend**



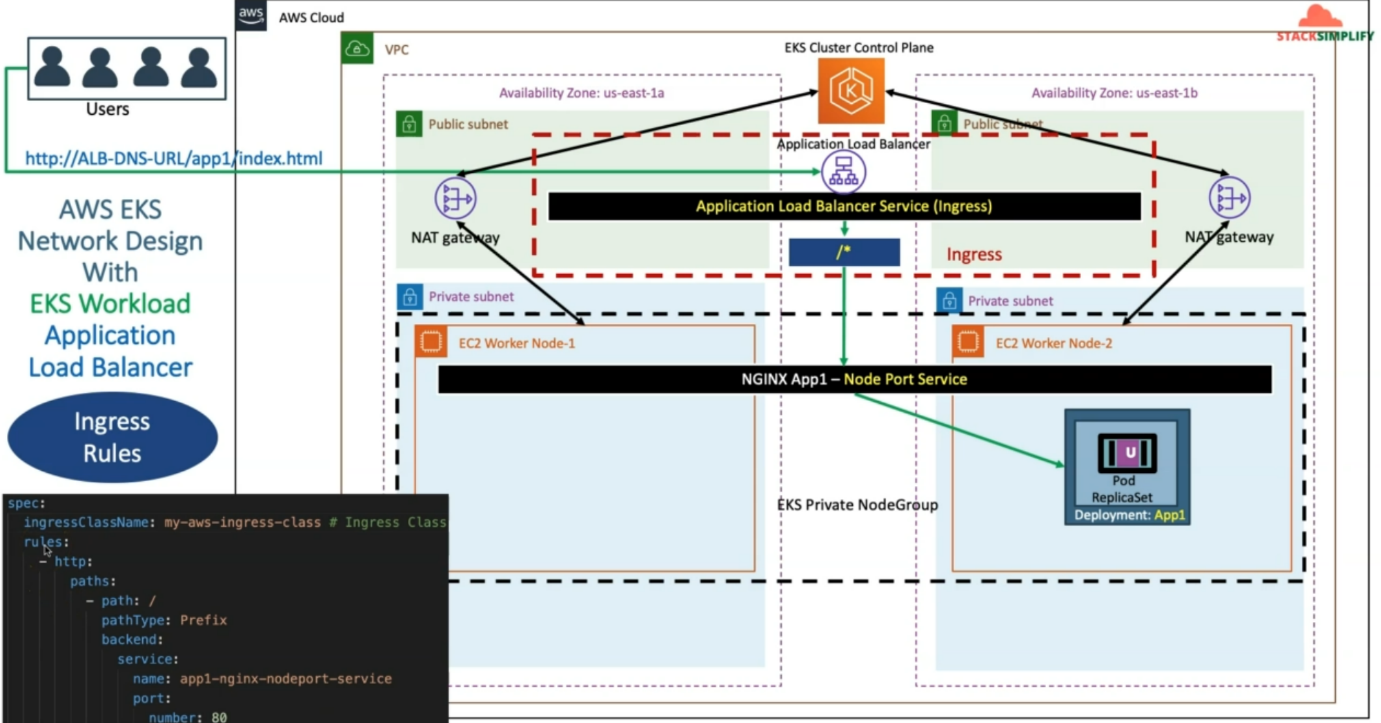
--- in aws cloud, we have created the EKS cluster and when we have created EKS cluster using kubectl. So, it also created a VPC and also created a public and private subnet and also the Nat Gateway and from private subnet the outbound communication goes via Nat Gateway to our EKS cluster control plane.

--- additionally, if we see here, we have created two worker nodes in the private subnet we call it as a private node group.

--- we deployed our simple application nginx app1 deployment in our private node group and which will create the pods for us and this respective deployment will be fronted with nginx app1 node port service. this node port service will be associated to the ingress resource.

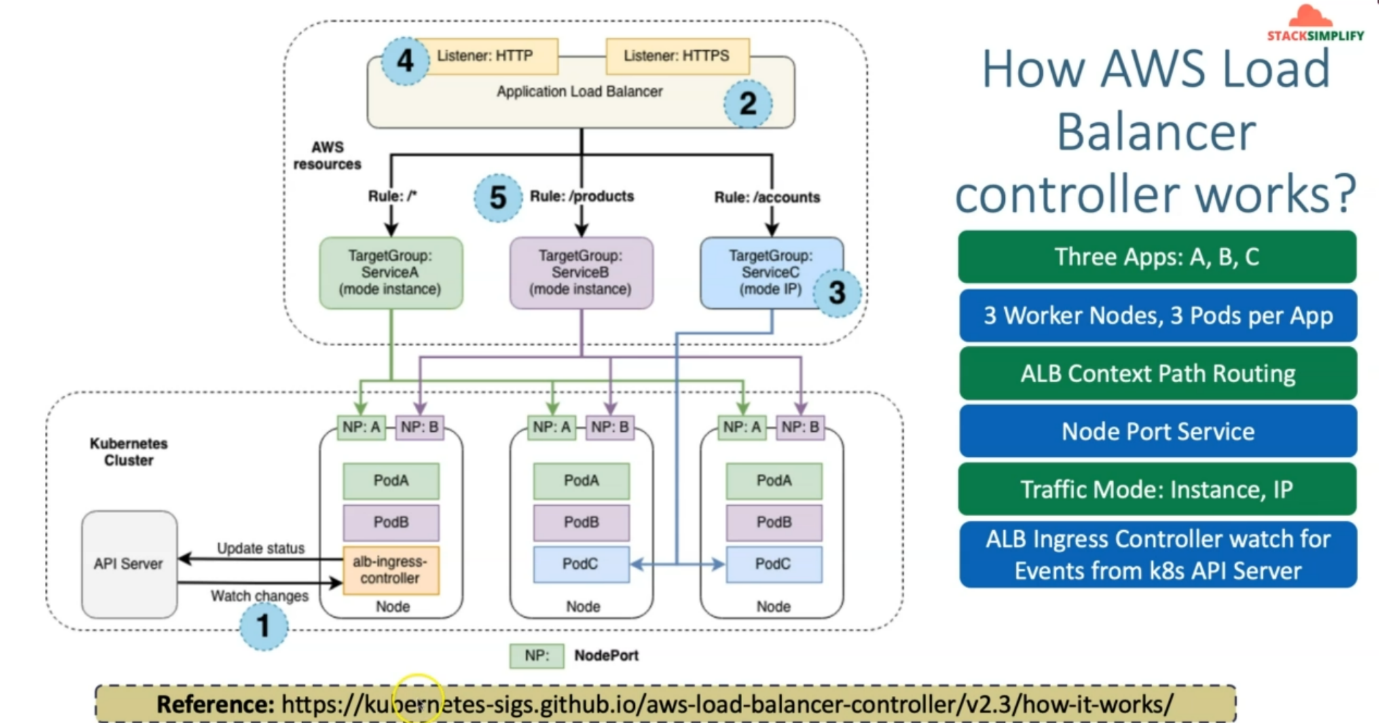
--- So, you will deploy ingress resource and inside the ingress resource default backend you are going to associate this node port service and whenever a user accesses your ALB DNS URL <https://ALB-DNS-URL/app1/index.html>. it is going to get the request to this application load balancer and inside the default backend you have the reference to you are node port service which is in your Kubernetes cluster, and from there it will go to your respective pod.

**Network design for ingress rules**



--- if you see here, you can see in the ingress spec from spec only. This is also path-based routing.

**How load balancer controller works…?**



--- these are aws list of resources, and here you can see this is the Kubernetes cluster and this is the Kubernetes API server and you already deployed your ALB ingress controller and you see you have the kubernetes API server and also you can see the worker nodes here.

--- You have three nodes here and if we see here, we have three apps A, B, C. let's consider you have three worker nodes and three pods per app here.

--- These three pods have ALB context-based routing implemented, which means the pod A. whenever you have the **/\*,** then the request will go to pod A. when you have the **/products**, the request will go to Pod B and when you have **/account**, the request will go to pod C.

--- let's go to the next level, which is nod port service, so for every application, you'll have its own port service, pod A have node port service, pod B have node port service, pod c don’t have node port service. We will discuss this later.

--- whenever the request comes to /\*, so it will go to that respective target group in the application load balancer. From there it will go to the respective node port service. From node port service the request will go to the pod A which are present in the other worker nodes as well. Same goes to the pod B as well.

--- whatever you are seeing the pod A and pod B is using traffic mode instance IP. Which means

Applications are using that traffic more instant type, so which means the traffic more instant type

is nothing but you are going to use the Northport service, which means these nodes are registered

to your target group in the load balancer and those nodes related node port service will be used to send the traffic to your pod.

--- when coming to traffic mode: instance IP, the third application, where the rule is /\*. So here directly in this target group, this pod related IP is present.

--- So, when you go to your load balancer and verify for this target group, it is pod A, pod B. You will find that your nodes are registered there, but here in pod c, you will find the Pod IP is associated with it. We will use this in fargate way.

--- **note** – here comes the important thing, when you deploy ingress service. This ingress service will check for the kubernetes API server for any ingress service is deployed for me. If it is deployed then it will take and deploy aws services.

--- the ingress service deploys application load balancer with listeners HTTP and HTTPS. It will create all the rules for us.

--- **note** – we will see the traffic mode instance ip during the far gateway.