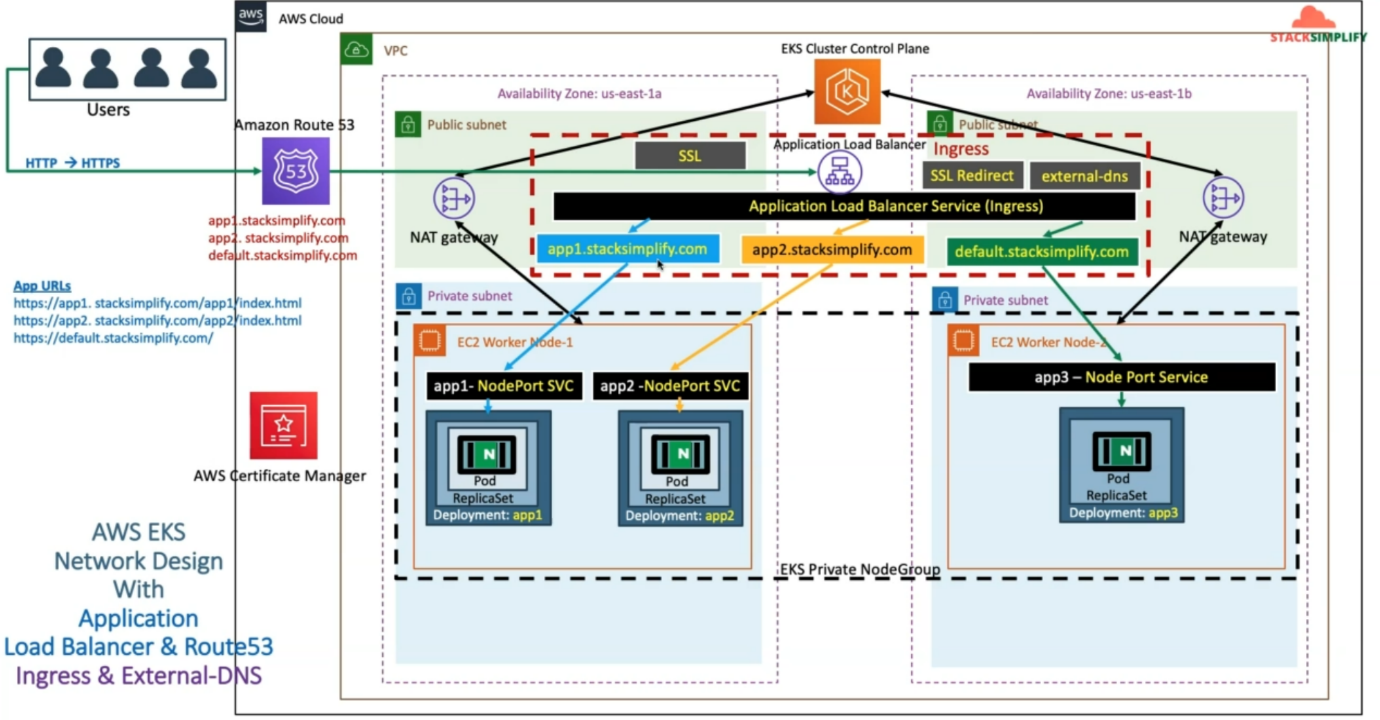
**1. Introduction to Ingress Name Based Virtual Host Routing**

--- in this section, we are going to implement another important ingress concept called name based virtual host or host-based routing and we are going to implement this concept on AWS application

load balancer.

**Network design**



--- So, let's understand the network designed for the same thing. As usual, we are going to use our three applications app1, app2, app3 and their 3 Kubernetes deployments and nodeport services are going to be as these.

---the ingress is the core thing. Inside Ingress, we are going to make the changes for our rules. So, we are going to define rules with the host header-based routing.

--- So, we are going to define two rules.

1. One is **ap1.stacksimplify.com**, which means that the host equal to ap1.stacksimplify.com. route the request to app1 node Port Service, from there it will go to that respective pod
2. When the host equal to **ap2.stacksimplify.com**, go to app2 nodeport service and from there it will go to equilent app2 pod
3. for **default**.**stacksimplify.com.** this is not rule**.** So, we will directly define this related to default backend with external DNS related annotation and anything you access with the **default**.**stacksimplify.com**. it will go to App three nodeport services, and from there it will go to App3 deployment later

--- So, next thing here is whenever you deploy this ingress resource, when you define these **ap1.stacksimplify.com**, **ap2.stacksimplify.com.** whenever you deploy this ingress rules and already if you have the external DNS related pods running in your EKS cluster, so then it is going to DNS register those domain names, whatever you have specified in host.

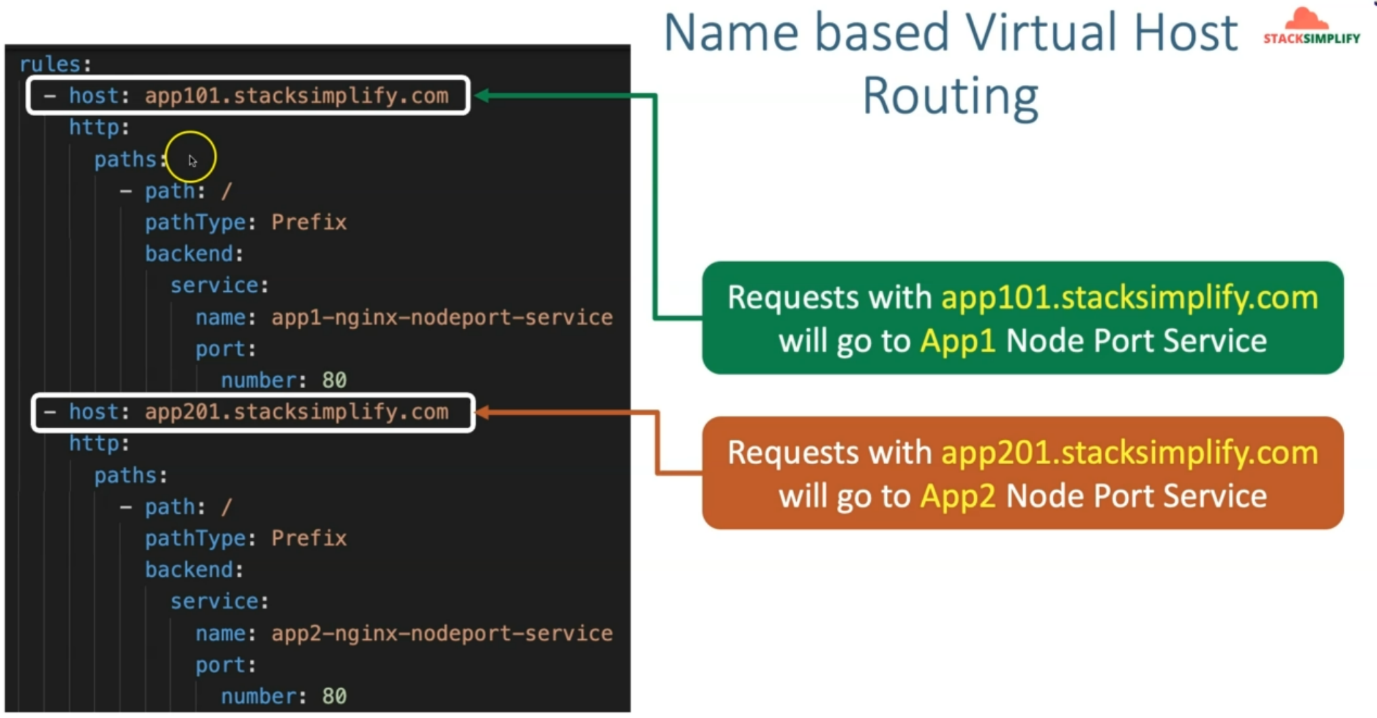
--- in addition to whatever you have specified in your external DNS related annotations. So, which means in the host rules also, you have specified **ap1.stacksimplify.com** and **ap1.stacksimplify.com.**

So, these two also will get DNS register in your Route53 using external DNS.

--- whenever user accesses [**https://ap1.stacksimplify.com/app1/index.html**](https://ap1.stacksimplify.com/app1/index.html). So, the requests will come to our ingress and then go to the app1 NodePort Service and that respective Pod, in same line whenever the user accesses [**https://ap2.stacksimplify.com/app1/index.html**](https://ap2.stacksimplify.com/app1/index.html). So it comes to app2 rule in ingress Load Balancer and from there it will go to app2 node port service and reach the app2 related nginx pod.

--- **note** - whenever the user accesses with **default**.**stacksimplify.com**. the requests will go to the default backend. So, this is called the host header-based routing or name based virtual host routing in ingress.

**Template for name based virtual host routing**

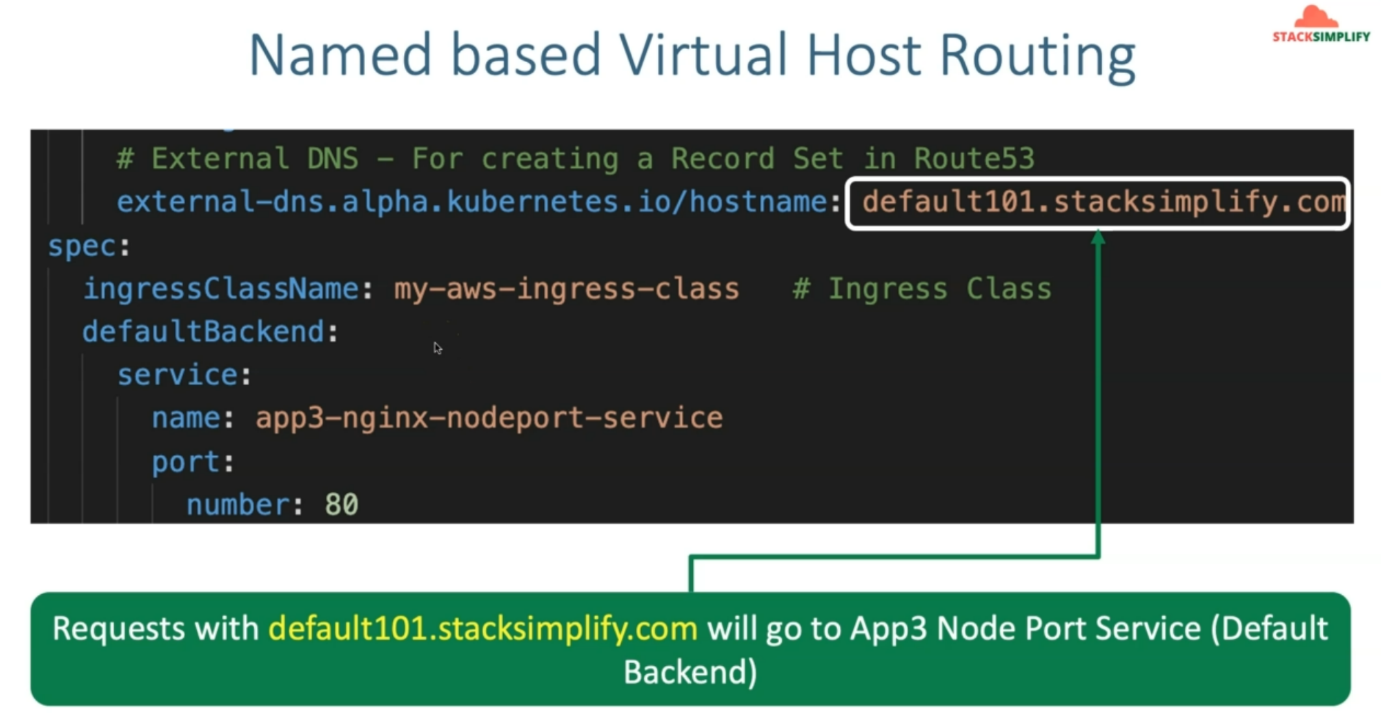


--- let's see the templates, how it is going to look for us. So, these are the rules, whatever we have discussed so far.

--- So earlier, the rules used to start with **http** and under http used to have paths and you can define multiple paths, that is called context path-based routing.

--- the host is replacing with **ap1.stacksimplify.com**, the request will goes to the app1-nginx-nodeport-service.

**For default backend**



--- for the default, we can directly use our external DNS related annotation with **default**.**stacksimplify.com** goes to our default backend, which is app3 nodeport service.