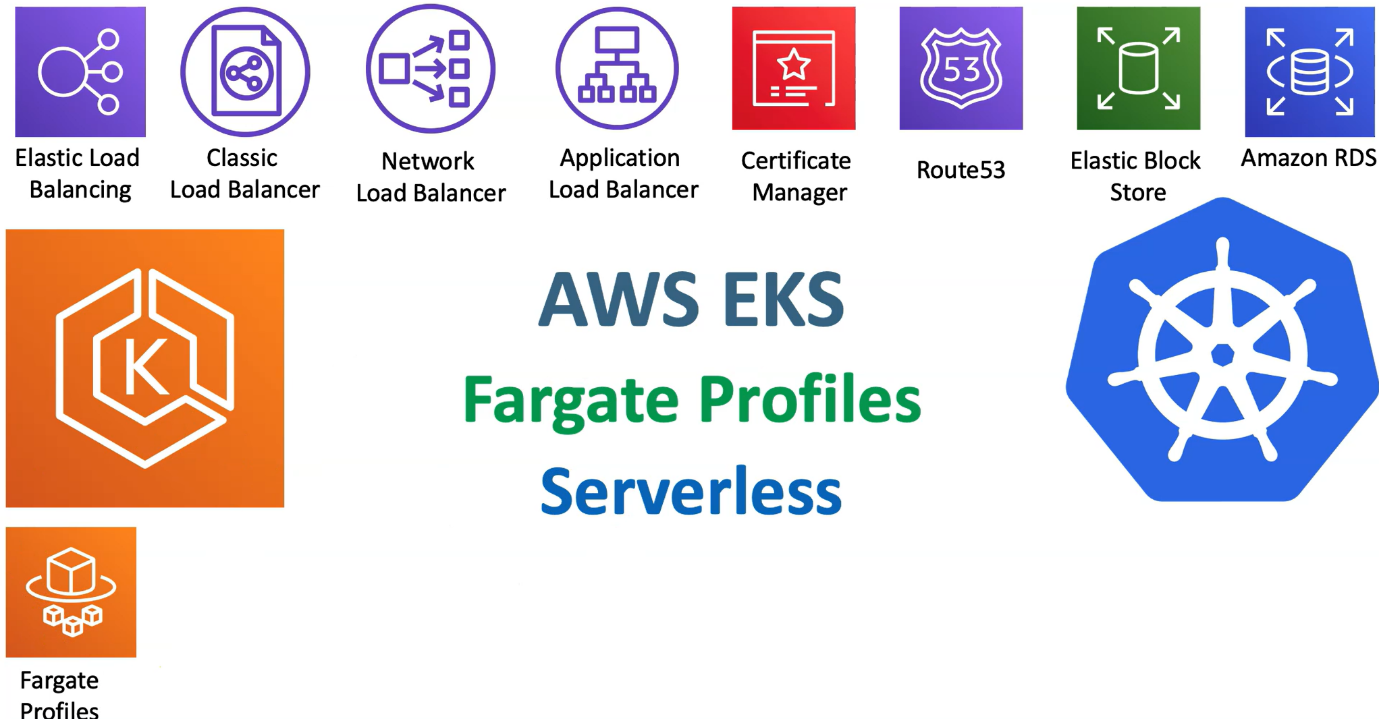
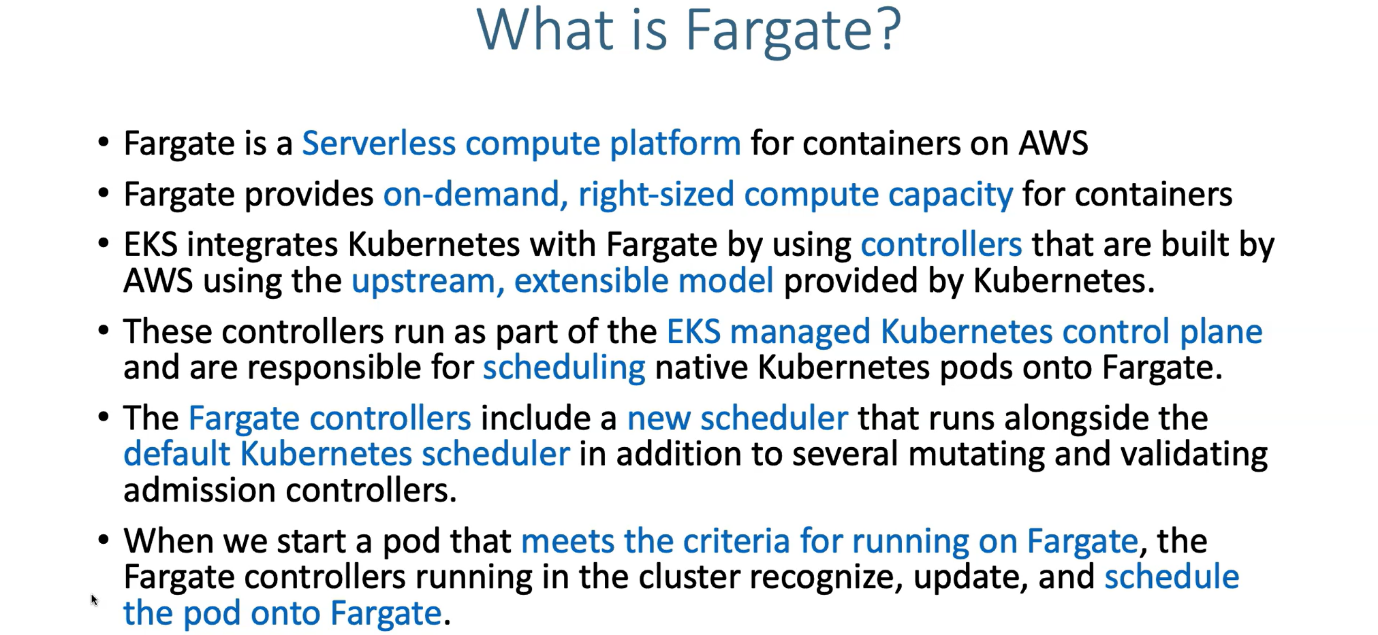
**1. EKS & Fargate Introduction**



--- In this lecture, we are going to get a very high-level introduction about, what fargate profiles and what is this fargate serverless.

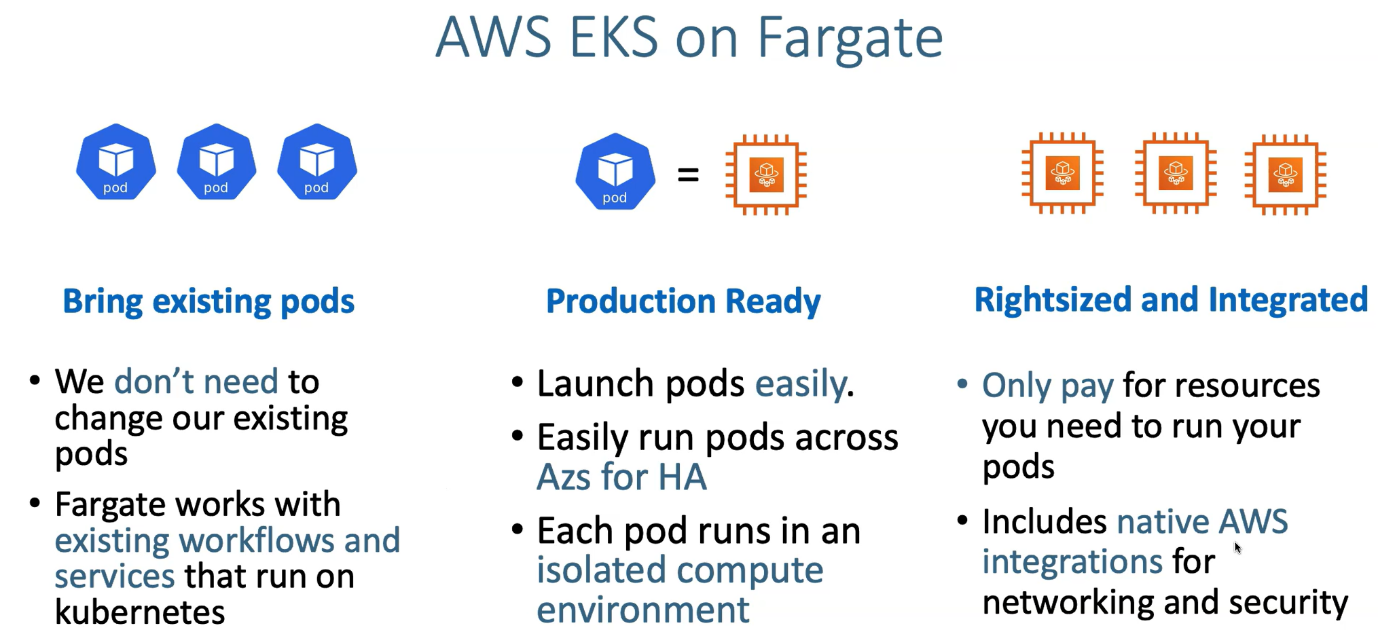
--- So, as part of this section, we are going to add our new feature to our EKS cluster, which is fargate profiles.

**What is the faregate?**



--- the key point in this slide is, we are going to have fargate controllers inside our **EKS** managed control plane, which is built using upstream extensible model provided by kubernetes for scheduling the pods on fargate.

**AWS EKS on fargate**



--- So, let's understand what we are going to get using fargate.

--- we can directly bring our existing pods and then directly deploy to fargate without any hassle. So, we don't need to change our existing pods.

--- the fargate is works with existing workflows and services that run on kubernetes. No change, you can straight away bring and deploy here.

--- but there is one condition. So, when we are running things on Fargate. There are some architectural decisions and then what it supports and then what it doesn't support also provided by **AWS** and that is a big list.

--- Those are called forget considerations. we also need to have a knowledge on those things. If we want to bring our applications to run on fargate.

--- next thing is there are production ready. So, now EKS fargate is production ready. we can launch pods easy to fargate and it runs pods across available zones for high availability and each pod runs in an isolate compute environment.

--- **note** - one pod is equal to one EC2 fargate instance in EKS. So, whenever we launch a pod, it is going to generate a EC2 fargate instance automatically in the background.

--- whenever you say **kubectl get nodes**, whenever you have deployed fargate pods then you'll get five EC2 fargate worker nodes available for you.

**Right sized and integrated**

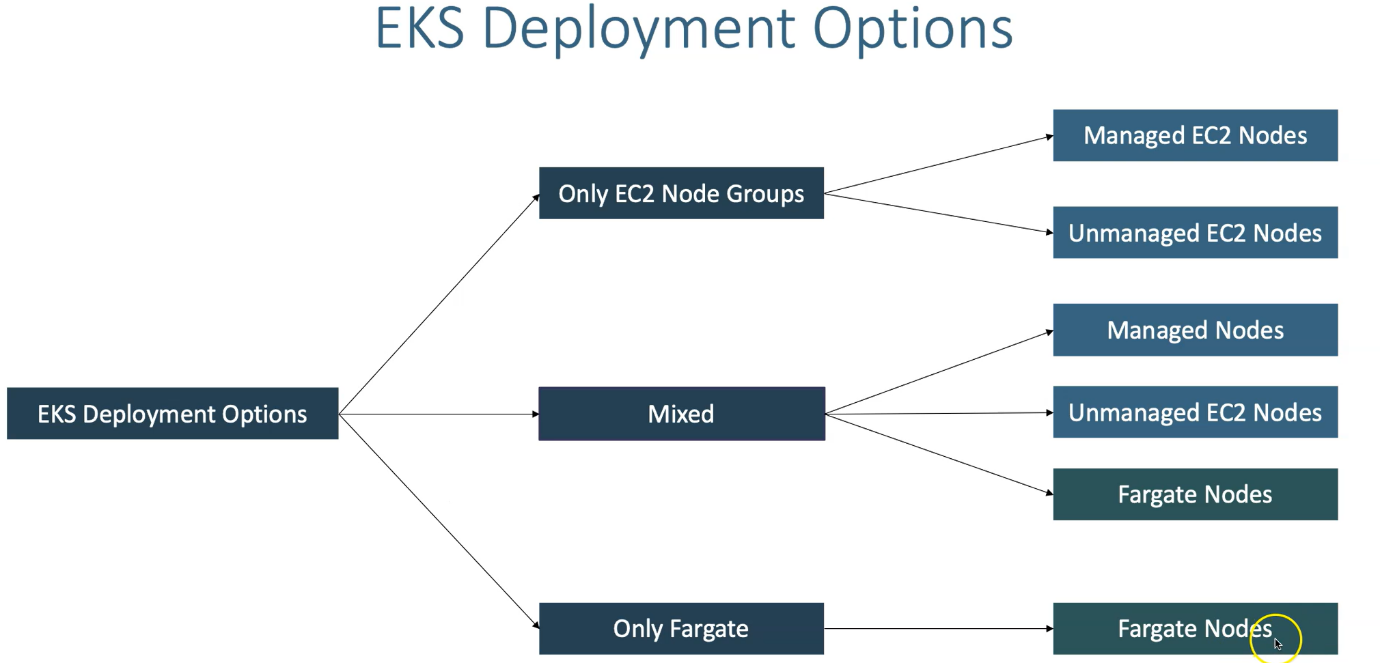
--- next thing is rightsized and then integrated. So, this is something like how much you need, use that much only.

--- if my pod needs these many resources then use that much only. So only pay for resources.

--- So, it also includes things like integrations for networking and then security.

--- So, networking and security is not compromised by using fargate on EKS. Moreover, your security is doubled only we can say, not reduced.

**EKS deployment options**



--- So, what are deployment options is available. If we take a look on a very high level.

--- we can have only EC2 node groups which are managed or unmanaged, we can have a mixed deployment with EC2 managed node groups with EC2 unmanaged node groups or only fargate cluster control plan with Fargate profiles can be created.

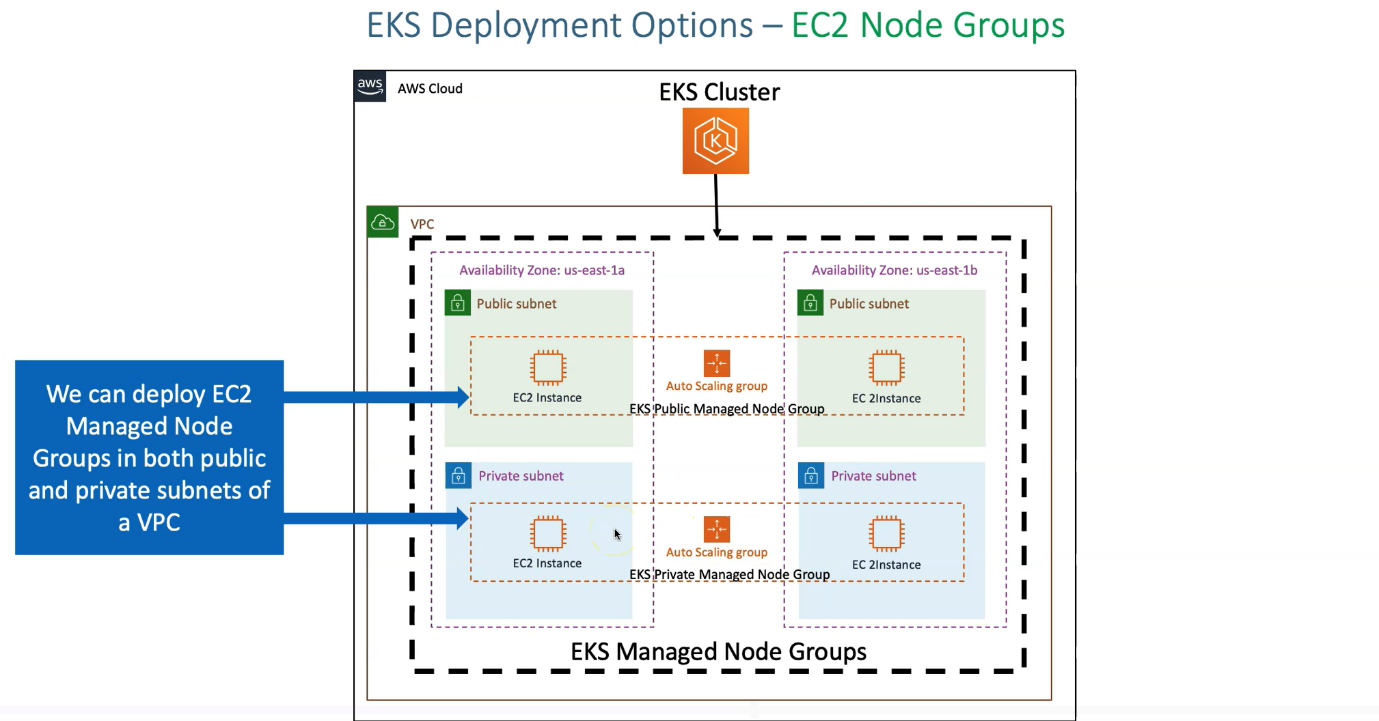
--- So, you can see EC2 node groups with manage EC2 and then unmanaged EC2 nodes.

--- our entire discourse will be dependent on managed EC2 and fargate profiles.

--- the next thing is Managed nodes and unmanaged nodes and fargate nodes in the mixer mode

--- forget you can only have fargate nodes. In simple terms. Fargate is Serverless.

**EKS Deployment options – EC2 Node Groups**



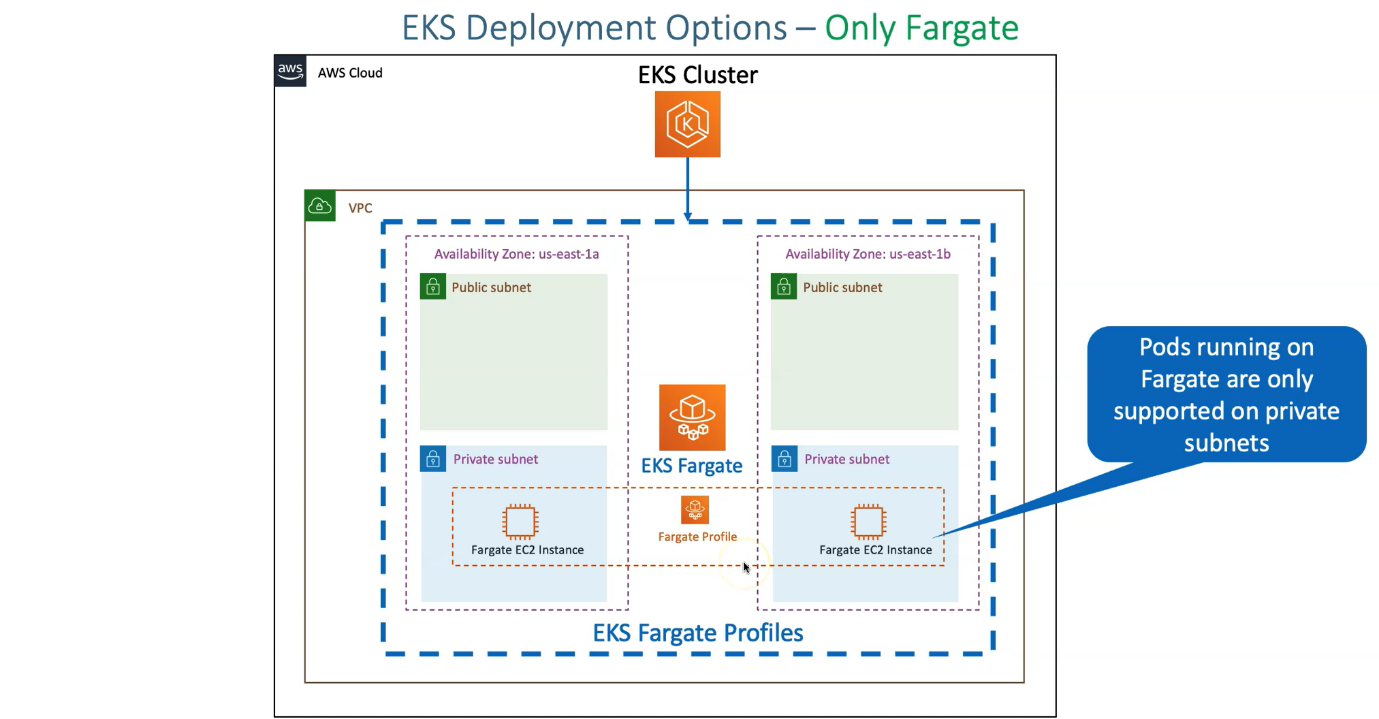
--- next thing is deployment options when you have EC2 node groups. primarily we are focusing on managed node groups only wherever I say, node groups. It is going to be managed node groups.

--- So, in an AWS cloud, whenever you create a cluster, it is also going to create a VPC and when we create a manager node group as part of that.

--- you can create your manager node group in public subnet or you can create in private subnet. So, this flexibility we have when we are dealing with EC2 instances.

--- So, you can have your manager node groups created in both the public and then private subnets and then handle that but when we go to fargate, you'll have a restriction there. we'll go there and then see it.

**EKS deployment options – only fargate**

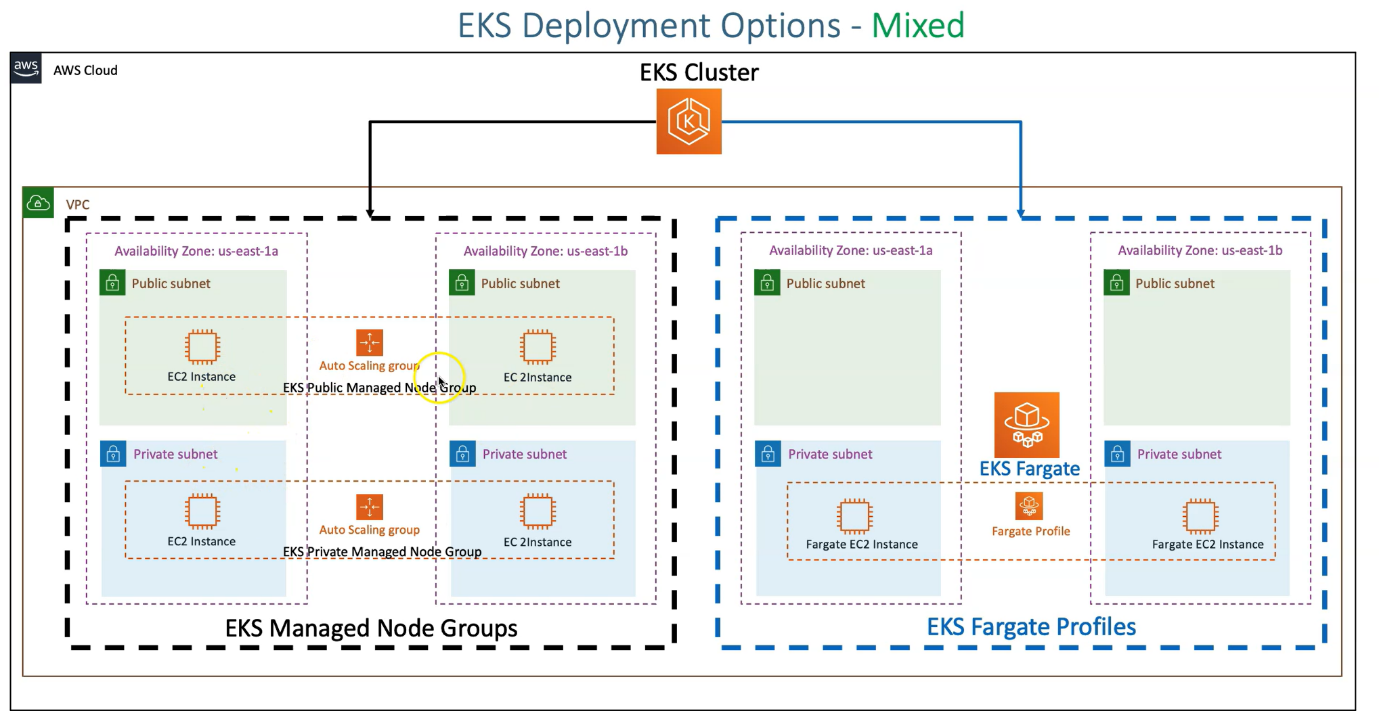


--- we are going to have our respective fargate profiles get created only inside private subnet. So, pods running on fargate only supported on private subnet.

--- In that case, whenever you have a EKS cluster and then whenever you want to deploy your loads on fargate, in your cluster at least one private subnet should be there to create your fargate profiles.

**EKS deployment options – Mixed**

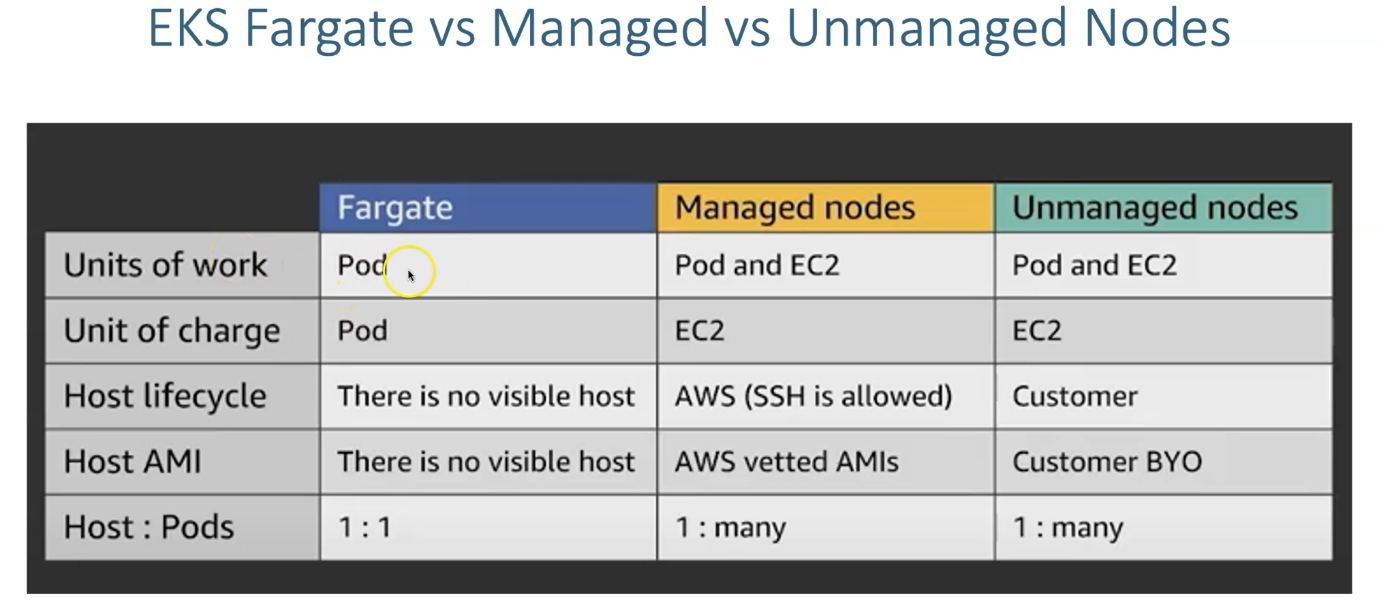
--- how would a mixer more deployment looks like?



--- EC2 instances in private and then public subnets and then you can create your public and private managed node groups there.

--- for fargate, you definitely need to force yourself to create your EKS fargate profiles in your private subnets.

**EKS fargate vs managed vs unmanaged nodes**



--- **units of work** - So, units of work for fargate is pod, whenever you deploy a pod, it is going to create a new underlaying fargate EC2 instance But it concerns about only pod but in managed nodes we first created EC2 instances and then redeploy our pods in them and in unmanaged nodes also. It is the same thing.

--- **unit of charge** - units of charge means like how the building happens. here in Fargate, it is going to be pod. So, when you are dealing with multi tenancy environments and then you want to charge for specific claims.

--- they have like few pods are running and then for them you want to charge. the charging perspective in the manager notes going to be based on the number of EC2 worker nodes. in unmanaged also, It's the same.

--- **Host Lifecycle** - there is no visibility of the host from forget perspective. The visibility in the sense, you will not be able to connect to that host, visible in the sense if you say **kubectl get nodes**. this fargate nodes also will be visible for us. Per pod, it is going to spend one node. ideally, we don't have access to that. what is happening inside is not our business?

--- in managing node, we'll have the ssh enabled. in unmanaged notes. the host life cycles is customer dependent. which means aws really don't care about what you are going to use and how you are going to handle your authentication or authorization about your host.

--- **host AMI** – In host AMI perspective for fargate, we really don't have any visibility about the host AMI in our forget because we are not even able to log in., we are not controlling that.

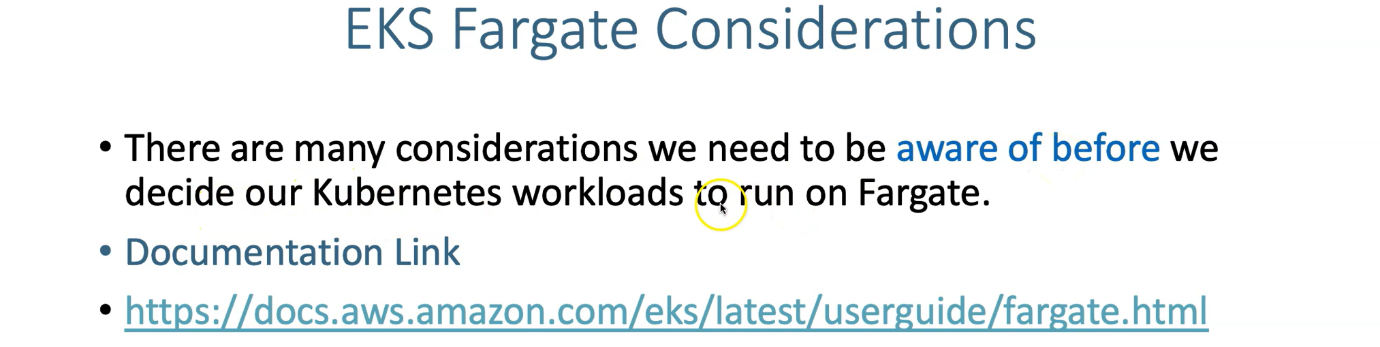
--- but in managed node, it is AWS vetted AMI’S will be provided for us and then they will be patched, auto-updated from AWS for us.

--- from unmanaged nodes perspective, customer, bring your own thing, which means including the OS also. you need to bring and then just use the EKS cluster control plane with that. lot of hard work we need to do.

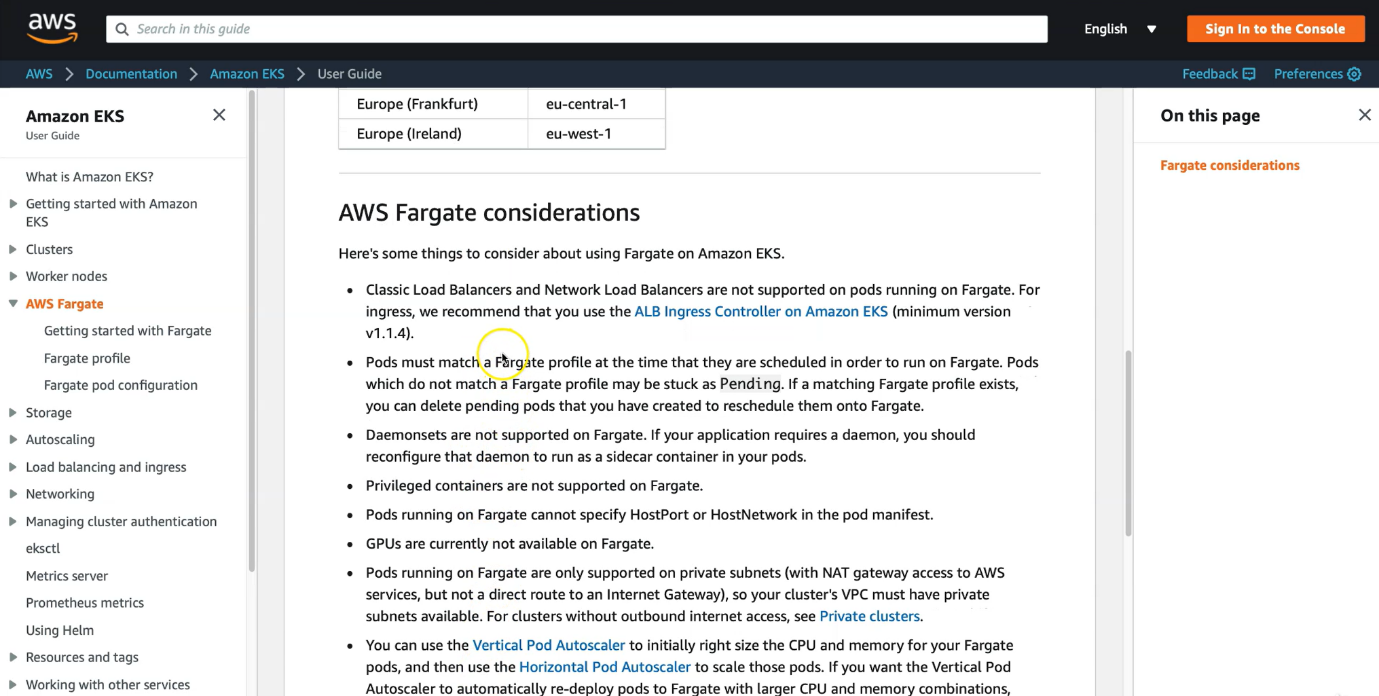
--- **host: pods** – host to pod relation. this is very, very important for us to make a note from fargate perspective. So, one fargate host is equal to one pod. So, if you spend 10 pods, then it is 10 fargate host.

--- managed nodes you can have two instances and then you can have many pods based on the C.P.U and then memory resources available in those nodes. It is the same case. One too many in unmanaged nodes.

**EKS fargate consideration**



--- use the above link to search in google.



--- read all the AWS fargate considerations.