Elastic Kubernetes service (AWS)

1. Can I just have Fargate, I don’t need EKS?

* - It took me a while before this lightbulb turned on. Fargate is nothing more than a type of Kubernetes worker node. EKS is nothing more than a Kubernetes control plane + etcd node.
* A Kubernetes cluster needs both control nodes and worker nodes, so Fargate worker nodes need EKS control nodes.
* You cannot make a peanut butter and jelly sandwich without peanut butter AND jelly.

1. what is Fargate?

* - A Kubernetes worker node that hosts a single pod. When the pod is deleted, this worker node is deleted. The worker node is an EC2 instance.
* If you have no pods then you have no charges for worker nodes.

1. Fargate is just a worker node type, is there another type?

* Yup, these are two types of worker nodes in EKS: Fargate and Node Groups

1. Can I have both Fargate and Node Group workers in the same EKS cluster?

* Yes, and in my opinion if you want to run Fargate you should also have at least one Node Group defined (see question 9, 10, and 11 for why).

1. what is a Node Group worker?

* These are “normal, regular, vanilla, plain ‘ol” Kubernetes worker nodes based on EC2 instances. The Kubernetes Scheduler will cram as many pods onto these VMs until CPU, Memory, or Network Interfaces resources are exhausted. When you create a Node Group you select:
* Name
* Node IAM Role
* -AMI Type (linux x64, x64+GPU or ARM)
* Instance Type (ie: t3.medium)
* Disk Size (for the node’s local storage)
* You pay for these EC2 instances regardless of their utilization.

1. Why not just run all the pods on Fargate?

* A couple of reasons with the one with the blinking warning lights is first:
* Fargate should not be used for stateful pods. If you need to write to a PVC and care - - about keeping that data around, you’ll need a Node Group.
* Fargate cannot be used for daemonsets.
* Fargate is not available everywhere EKS is. For instance, EKS is available in GovCloud East/West but Fargate is not.
* Kubernetes upgrades (to the kubelet on the worker) cannot be done unless the pod is recreated.

1. How do I place a pod on a Fargate worker node?

* You need to define a Fargate Profile which is just a list of namespaces. In the AWS Console these can be created by navigating to EKS > Pick your cluster > Compute > Add Fargate Profile.
* Once the Fargate Profile is created, when you create a pod, if the namespace of the pod matches one of the namespaces in the Fargate Profile, then it will result in a Fargate worker node being created and the pod placed on the worker.

1. Does the namespace need to exist before I create a Fargate Profile which references the namespace?

* Nope. Just know that the placement only is determined when the pod is created.

1. How do I choose whether a pod goes to Fargate or a Node Group if I have both?

* If an EKS cluster has both Fargate Profiles and Node Groups, the Fargate Profile is evaluated first. If the pod’s namespace matches in the Fargate Profile it will wind up on a Fargate worker, otherwise it will be created on a Node Group worker.

1. What happens if I’m only using Fargate but deploy a pod to a namespace not in the Fargate Profile?

* Sadness. Your pod will be stuck in a pending state. Forever. Stuck in a perpetual Yay, welcome to Ground Hog Day, 2020 edition.

1. Can I make all pods always go to Fargate regardless of the namespace?

* As far as I can tell, no. You have to define the list of namespaces that you want to leverage Fargate in the Fargate Profile

1. Can I use my own AMI?

* Another great question with a few answers:
* For the Control Plane – No, this is controlled by AWS
* For Fargate workers – No, this is controlled by AWS
* For Node Group workers – Yes, but a complicated yes. When you create a Node Group via the AWS Console, the default behavior is to not use Launch Templates so you only have the 3 AMI options (linux x64, linux x64+GPU, linux ARM). If you opt to use Launch Templates you can select your own full list of AMIs. Launch Templates will also allow you to leverage spot instances, IAM instance profiles, tenancy and a dozen other - - configurations. If you use Launch Templates then you’ll need to maintain them going forward.

1. Does autoscaling work?

* A couple things to unpack here as there is both Node Autoscaling and (Horizontal) Pod Autoscaling.
* Does Horizontal Pod Autoscaling work on Fargate? Yes. On Node Groups? Yes. For each you need to make sure to enable Metrics Server.
* For node autoscaling, if you are using Fargate for the pod, just add another pod, you’ll get another transparent EC2 worker node. If you are using Node Groups, you set the:
* Minimum number of nodes
* Maximum number of nodes
* Desired number of nodes – This is the number of nodes the Node Group will initially create
* As you deploy or remove pods the number of nodes will scale within the guardrails defined above.

1. Can I deploy EKS to a Dedicated VPC?

* No

1. Can I run my own CNI like Calico?

* On Fargate, no.
* On Node Groups yes, but you’ll need to create your and manage your own AMIs. Using a CNI other than the Amazon VPC CNI plug-in may also mean you are responsible for debugging any pod networking issues that crop up.

1. How do I upgrade Kubernetes on EKS?

* Performing an “Upgrade of Kubernetes” is done in multiple steps. You start by upgrading the Control Plane first. In the AWS Console, this is done by click in the big blue Update Now on the Clusters page.
* One gotcha is you cannot upgrade two or more minor version at a time. If your control plane is at 1.17 and you have any Node Groups or Fargate workers running 1.16, you have to upgrade these to 1.17 before upgrading the control plane to 1.18.

1. How do I upgrade Kubernetes on Node Groups

* Once the Control Plane is upgraded to a newer version, the Node Groups can be updated.
* If you aren’t using your own Launch Templates you’ll be prompted in the AWS Console under EKS > Pick your cluster > Compute > Node Groups > AMI release version if an AMI with a newer kubelet version is available. Clicking Update nowwill result in:
* New workers will be create using the newer AMI
* Once all the new worker nodes are healthy, the older nodes will be drained and status will be Ready,SchedulingDisabled.
* After the pods are moved off of the old workers, the underlying EC2 instances are terminated and the upgrade is complete
* If you are maintaining your own AMIs you’ll need to create AMIs with the newer kubelet version. It may be easiest to just create a new Node Group with the new AMI, then delete the old Node Group once the new ones are healthy.

1. How do I upgrade Kubernetes on Fargate?

* Once the control plane has been upgraded to the newer version, any newly created fargate pods will deploy EC2 instances with AMIs based on the version the master has.
* To upgrade existing pods, there is no automation to do this. The pods needs to be destroyed and recreated. To do this without downtime for the app itself, Kubernetes Deployments should be leveraged which can do rolling upgrades.

1. How are the kubelet certificates rotated?

* This depends on which nodes we are looking at:
* Control plane nodes – Certificate rotation for the master nodes is managed by AWS.
* Fargate nodes – This is baked into the AMI, recreating the pod will result in a new EC2 instance with a newer kubelet + certificate
* Node Group nodes – The kubelet certificate is configured to rotate in the /etc/kubernetes/kubelet/kubelet-config.json configured in the AMI

1. Can Fargate nodes exist in a private subnet?

* Yes, also Node Groups can be either public or private subnets. You may want to leverage more than 1 Node Group and add labels to keep some apps private versus public facing.

1. Do Node Group VMs cost more than the EC2 VMs of the same instance type? No, whatever the cost of the EC2 instance is is what you pay for.
2. Is the version of Kubernetes automatically upgraded for me when new versions come out? No, an admin needs to upgrade the control plane and then any workers. AWS will publish new AMIs but the admin is required to consume these upstream changes.
3. Can I use something other than the AWS Console to spin up clusters? No, there is an excellent CLI tool called eksctl which will you can use to completely manage the bootstrap and lifecycle of the deployment. You can also follow this blog post to use Terraform to bootstrap and maintain the deployments. As a side note: pick one tool, don’t deploy with Terraform and then use the AWS Console to make changes, your next terraform apply will overwrite or be blocked by the manual changes.