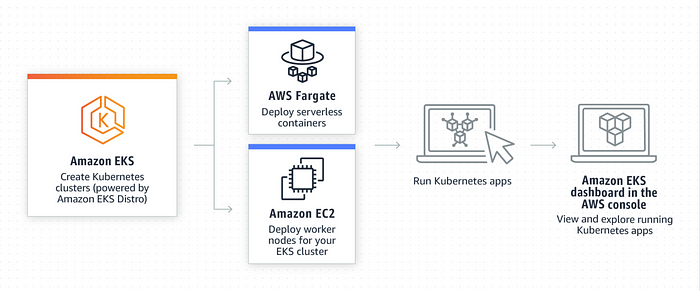
**Creating an Amazon EKS Cluster with AWS CLI**



Run Amazon EKS in the cloud

**EKS (Elastic Kubernetes Service)** is a managed Kubernetes service provided by Amazon Web Services (AWS). It allows users to easily deploy, manage, and scale containerized applications using Kubernetes on AWS infrastructure. Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications.

Advantages of EKS over running Kubernetes on our own cloud infrastructure:

***Managed Service:****EKS is a fully managed service, meaning that AWS takes care of the underlying infrastructure, such as server provisioning, scaling, and patching, allowing users to focus on their applications.*

***Scalability:****EKS enables easy scaling of Kubernetes clusters to accommodate varying workloads. It integrates with other AWS services like Auto Scaling Groups, Elastic Load Balancing, and AWS Fargate, making it simple to scale applications based on demand.*

***Security and Compliance:****EKS offers strong security features, including integration with AWS Identity and Access Management (IAM), Amazon Virtual Private Cloud (VPC), and AWS Security Groups. It also provides compliance with various industry standards.*

***High Availability:****EKS is designed to provide high availability for Kubernetes clusters. It runs across multiple availability zones, ensuring that applications remain resilient and accessible even in the event of failures.*

***Integration with AWS Services:****EKS seamlessly integrates with other AWS services, such as Amazon Elastic Container Registry (ECR), AWS CloudFormation, AWS Identity and Access Management (IAM), and AWS App Mesh, allowing users to leverage the full capabilities of the AWS ecosystem.*

Now, let’s look at some useful commands for eksctl, which is a command-line tool used for managing EKS clusters:

1. **Create a new EKS cluster:**

**eksctl create cluster --name my-eks-cluster --region us-west-2 --node-type t2.micro --nodes 3**

This command creates a new EKS cluster named “my-eks-cluster” in the “us-west-2” region with worker nodes of type “t2.micro” and 3 nodes.

**2. Delete an EKS cluster:**

**eksctl delete cluster --name my-eks-cluster --region us-west-2**

This command deletes the EKS cluster named “my-eks-cluster” in the “us-west-2” region.

**3. Scale the number of nodes in an EKS cluster:**

**eksctl scale nodegroup --cluster my-eks-cluster --nodes 5 --name my-nodegroup**

This command scales the worker nodes in the node group named “my-nodegroup” of the cluster “my-eks-cluster” to 5 instances.

**4. Update the Kubernetes version of an EKS cluster:**

**eksctl update cluster --name my-eks-cluster --region us-west-2 --kubernetes-version 1.24**

This command updates the Kubernetes version of the EKS cluster “my-eks-cluster” in the “us-west-2” region to version 1.24.

**5.. View cluster details:**

**eksctl get cluster --name my-eks-cluster --region us-west-2**

This command provides information about the EKS cluster “my-eks-cluster” in the “us-west-2” region, such as status, version, and node groups.

***Remember to replace “my-eks-cluster,” “us-west-2,” “t2.micro,” and other parameters with your desired values when using these commands.***

In conclusion, EKS provides a managed and scalable platform for running Kubernetes workloads. It offers benefits such as reduced operational overhead, seamless integration with other AWS services, and simplified management through the eksctl command-line tool. These advantages make EKS a powerful option for deploying containerized applications on AWS.