Amazon EKS

Amazon Elastic Kubernetes Service

Case Study



Introduction

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed service that you can use to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane or nodes. Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications. Amazon EKS:

* Runs and scales the Kubernetes control plane across multiple AWS Availability Zones to ensure high availability.
* Automatically scales control plane instances based on load, detects and replaces unhealthy control plane instances, and it provides automated version updates and patching for them.
* Is integrated with many AWS services to provide scalability and security for your applications, including the following capabilities:
  + Amazon ECR for container images
  + Elastic Load Balancing for load distribution
  + IAM for authentication
  + Amazon VPC for isolation
* Runs up-to-date versions of the open-source Kubernetes software, so you can use all of the existing plugins and tooling from the Kubernetes community. Applications that are running on Amazon EKS are fully compatible with applications running on any standard Kubernetes environment, no matter whether they're running in on-premises data centers or public clouds. This means that you can easily migrate any standard Kubernetes application to Amazon EKS without any code modification.

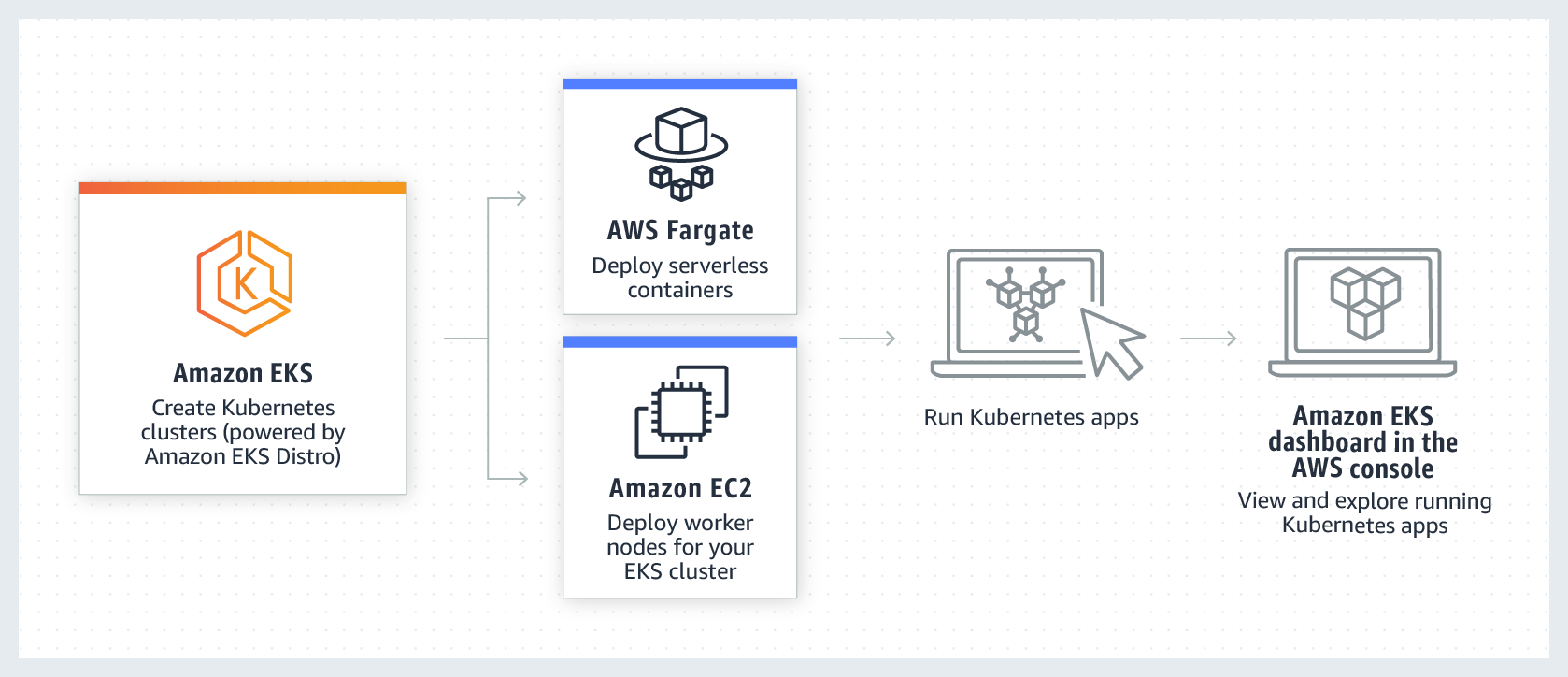
**How does Amazon EKS work?**


    How Amazon EKS works
   

Getting started with Amazon EKS is easy:

1. Create an Amazon EKS cluster in the AWS Management Console or with the AWS CLI or one of the AWS SDKs.
2. Launch managed or self-managed Amazon EC2 nodes, or deploy your workloads to AWS Fargate.
3. When your cluster is ready, you can configure your favorite Kubernetes tools, such as kubectl, to communicate with your cluster.
4. Deploy and manage workloads on your Amazon EKS cluster the same way that you would with any other Kubernetes environment. You can also view information about your workloads using the AWS Management Console.

Deploy application with amazon EKS in cloud.



Step 1: Create your Amazon EKS cluster and nodes

Create your Amazon EKS cluster with the following command. You can replace *my-cluster* with your own value. The name can contain only alphanumeric characters (case-sensitive) and hyphens. It must start with an alphabetic character and can't be longer than 100 characters. Replace *region-code* with any AWS Region that is supported by Amazon EKS. For a list of AWS Regions, see [Amazon EKS endpoints and quotas](https://docs.aws.amazon.com/general/latest/gr/eks.html) in the AWS General Reference guide.

eksctl create cluster --name my-cluster --region region-code –fargate

or

eksctl create cluster \

--name test-cluster \

--version 1.17 \

--region ap-south-1 \

--nodegroup-name linux-nodes \

--node-type t2.micro \

--nodes 2

**Step 2: View Kubernetes resources**

Get worker nodes list write following command

* **Kubectl get nodes**
* **Kubectl get ns**

1. View your cluster nodes.

**kubectl get nodes -o wide**

The example output is as follows.

1. View the workloads running on your cluster.

**kubectl get pods -A -o wide**

**Step 3: Delete your cluster and nodes**

After you've finished with the cluster and nodes that you created for this tutorial, you should clean up by deleting the cluster and nodes with the following command. If you want to do more with this cluster before you clean up, see [Next steps](https://docs.aws.amazon.com/eks/latest/userguide/getting-started-eksctl.html#gs-eksctl-next-steps).

**eksctl delete cluster --name my-cluster --region region-code**