AWS ECS VS EKS: WHAT'S THE DIFFERENCE? HOW TO CHOOSE?

The increased popularity of containerized applications has illustrated the need for proper container orchestration platforms to support applications at scale.

Containers need to be managed throughout their lifecycle, and many products have been created to fulfill this need. These container orchestration products range from open-source solutions such as Kubernetes and Rancher to provider-specific implementations such as:

- Amazon Elastic Container Service (ECS)
- Azure Kubernetes Service (AKS)
- Elastic Kubernetes Service (EKS)

All these different platforms come with their unique advantages and disadvantages. Amazon itself offers an extensive array of container management services and associated tools like the ECS mentioned above, EKS, AWS Fargate, and the newest option, EKS Anywhere.

AWS users need to evaluate these solutions carefully before selecting the right container management platform for their needs—and we're here to help!

(This tutorial is part of our AWS Guide. Use the right-hand menu to navigate.)

How container management works

A container is a lightweight, stand-alone, portable, and executable package that includes everything required to run an application from the application itself to all the configurations, dependencies, system libraries, etc. This containerization greatly simplifies the development and deployment of applications. However, we need the following things to run containers:

- An executable environment with CPU, RAM, and storage resources
- Networking to communicate between containers, other services, and the wider internet
- Storage and databases
- Caching, APIs, other external services
- Monitoring from metrics, application and system logs, security

While containers encapsulate the application itself, the container management or an orchestration platform provides the rest of the above facilities required throughout the lifecycle of the container.

ECS and EKS are the primary offerings by AWS that aim to provide this container management functionality. In the following sections, we will see what exactly these two offerings bring to the table.

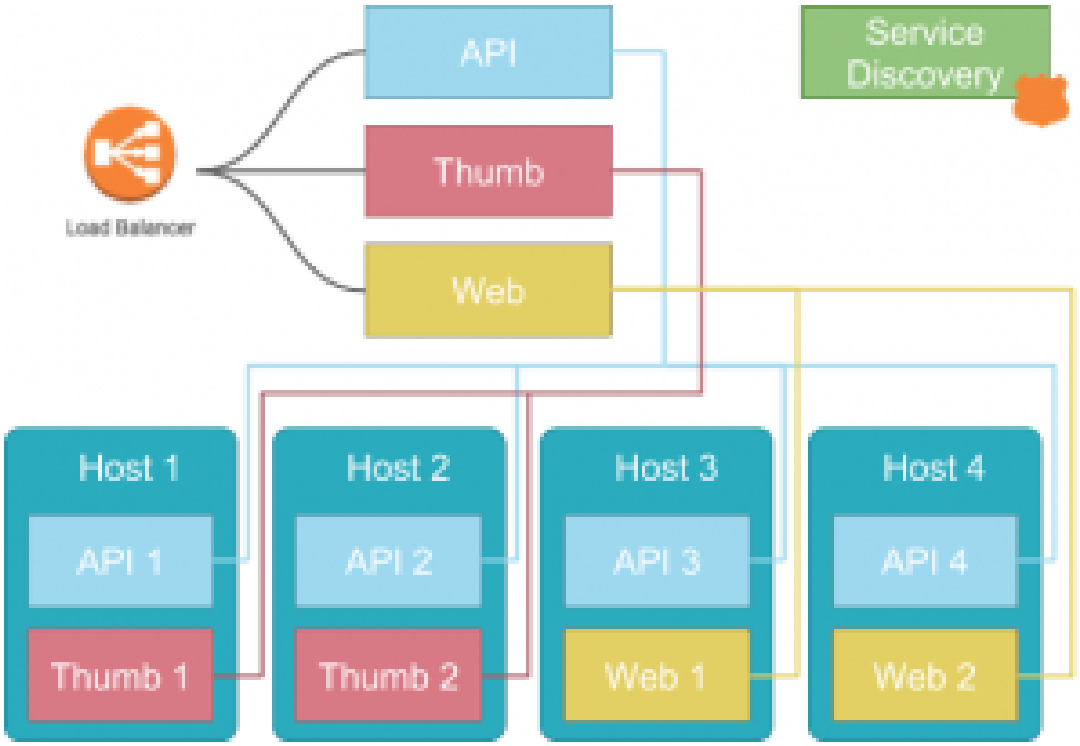
What is Amazon Elastic Container Service (ECS)?

The Elastic Container Service can be construed as a simplified version of Kubernetes—but that's misleading. The Elastic Container Service is an AWS-opinionated, fully managed container orchestration service. ECS is built with simplicity in mind without sacrificing management features. It easily integrates with AWS services such as AWS Application/Network load balancers and CloudWatch.

Amazon Elastic Container Service uses its scheduler to determine:

- Where a container is run
- The number of copies started
- How resources are allocated

As shown in the following image, ECS follows a simple, easily understood model. Each application in your stack (API, Thumb, Web) is defined as a service in ECS and schedules (runs) tasks (instances) on one or more underlying hosts that meet the resource requirements defined for each service.



This model is relatively simple to understand and implement for containerized workloads as it closely resembles a

traditional server-based implementation. Thus migrating applications to ECS becomes a simple task that only requires the containerized application, pushing the container image to the Amazon Elastic Container Repository (ECR) and then defining the service to run the image in ECS.

Most teams can easily adapt to such a workflow. ECS also provides simple yet functional management and monitoring tools that suit most needs.

What is Elastic Kubernetes Service (EKS)?

The Elastic Kubernetes Service is essentially a fully managed Kubernetes Cluster. The primary difference between ECS and EKS is how they handle services such as networking and support components.

- ECS relies on AWS-provided services like ALB, Route 53, etc.,
- EKS handles all these mechanisms internally, just as in any old Kubernetes cluster.

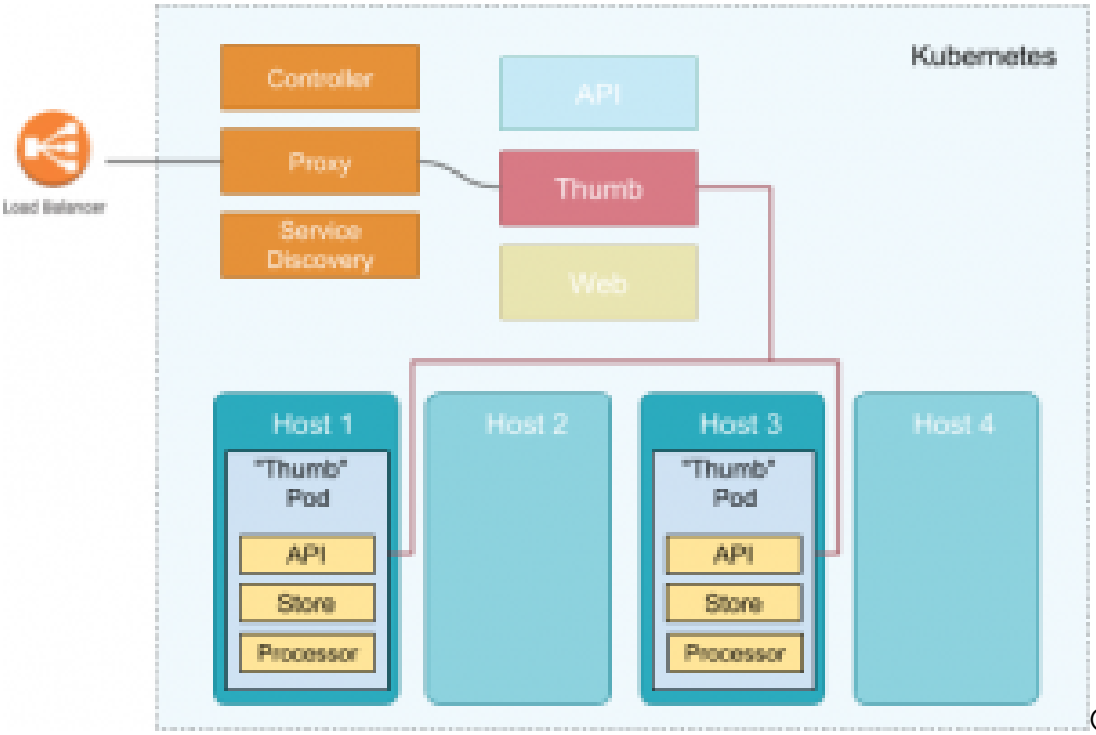
The Elastic Kubernetes Service provides all the features and flexibility of Kubernetes while leveraging the managed nature of the service. However, all these advantages come with the increased complexity of the overall application architecture.

EKS introduces the Kubernetes concept of Pods to deploy and manage containers while ECS directly uses individual containers to deploy them. Pods can contain either one or more containers with a shared resource pool and provide far more flexibility and fine-grained control over components within a service. The below image shows that all the services (ex: proxy, service discovery) that need to run containers are within the Kubernetes cluster.

Let's assume that our Thumb service was a combination of three separate components:

- A microservice API
- An image processor
- A storage engine

Kubernetes allows us to run these three separate components as distinct containers within a single Pod that makes up the Thumb service.



Containers within the pods run collocated with one another. Furthermore, they have easy access to each other and can share resources like storage without relying on complex configurations or external services. All these facts make it possible for users to create more complex applications architectures with EKS.

Additionally, EKS enables users to tap into the wider Kubernetes echo-system and use add-ons like:

- The networking policy provider Calico
- Romana Layer 3 networking solution
- CoreDNS flexible DNS service
- Many other third-party add-ons and integrations

Since EKS is based on Kubernetes, users have the flexibility to move their workloads between different Kubernetes clusters without being vendor-locked into a specific provider or platform.

What is Fargate? How does it affect all this?

Even with managed services, servers still exist, and users can decide which types of compute options to use with ECS or EKS.

AWS Fargate is a serverless, pay-as-you-go compute engine that allows you to focus on building applications without managing servers. This means that AWS will take over the management of the underlying server without requiring users to create a server, install software, and keep it up to date. With AWS Fargate, you only need to create a cluster and add a workload—then, AWS will automatically add pre-configured servers to match your workload requirements.

Fargate is the better solution in most cases. It will not cost more than self-managed servers and, most of the time, saves costs due to only charging for the exact usage. Therefore, users do not have to worry about the unused capacity like in self-managed servers, which requires manually shutting down to save costs.

However, here some notable exceptions of Fargate:

- **Fargate cannot be used in highly regulated environments with strict security and**

compliance requirements. The reason is that users lose access to the underlying servers, which they might need control over to meet those stringent regulatory requirements. Additionally, Fargate does not support "dedicated tenancy" hosting requirements.
- **ECS and Fargate only support AWS VPC networking mode,** which may not be suitable when deep control over the networking layer is required.
- **Fargate automatically allocates resources depending on the workload with limited control over the exact mechanism.** This automatic resource allocation can lead to unexpected cost increases, especially in R&D environments where many workloads are tested. Therefore, self-managed servers with capacity limitations will be a better solution for these kinds of scenarios.

What about EKS Anywhere?

EKS Anywhere extends the functionality of EKS by allowing users to create and operate Kubernetes clusters on customer-managed infrastructure with default configurations. It provides the necessary tools to manage the cluster using the EKS console.

EKS Anywhere is built upon the Amazon EKS Distro and provides all the necessary and up-to-date software which resides on your infrastructure. Moreover, it provides a far more reliable Kubernetes platform compared to a fully self-managed Kubernetes cluster.

EKS Anywhere is also an excellent option to power hybrid cloud architecture while maintaining operational consistency between the cloud and on-premises. Besides, EKS Anywhere provides the ideal solution to keep data with on-premises infrastructure where data sovereignty is a primary concern. It leverages AWS to manage the application architecture and delivery.

Choosing ECS vs EKS: which is right for you?

EKS is undoubtedly the more powerful platform. However, it does not mean EKS is the de facto choice for any workload. ECS is still suitable for many workloads with its simplicity and feature set.

When to use ECS

- ECS is much simpler to get started with a lower learning curve. Small organizations or teams with limited resources will find ECS the better option to manage their container workloads compared to the overhead associated with Kubernetes.
- Tighter AWS integrations allow users to use already familiar resources like ALB, NLB, Route 53, etc., to manage the application architectures. It helps them to get the application up and running quickly.
- ECS can be the stepping stone for Kubernetes. Rather than adapting EKS at once, users can use ECS to implement a containerization strategy and move its workloads into a managed service with less up-front investment.

When to use EKS

On the other hand, ECS can sometimes be too simple with limited configuration options. This is where EKS shines. It offers far more features and integrations to build and manage workloads at any scale.

- Pods may not be required for many workloads. However, Pods offer unparalleled control over

pod placement and resource sharing. This can be invaluable when dealing with most service-based architectures.
- EKS offers far more flexibility when managing the underlying resources with the flexibility to run on EC2, Fargate, and even on-premise via EKS Anywhere.
- EKS provides the ability to use any public and private container repositories.
- Monitoring and management tools of ECS are limited to the ones provided by AWS. While they are sufficient for most use cases, EKS allows greater management and monitoring capabilities both via built-in Kubernetes tools and readily available external integrations.

All in all, the choice of the platform comes down to specific user needs. Both options have their pros and cons, and any of them can be the right choice depending on the workload.

To sum up, it's better to go with EKS if you are familiar with Kubernetes and want to get the flexibility and features it provides. On the other hand, you can try ECS first if you are just starting up with containers or want a simpler solution.

Related reading

- BMC DevOps Blog
- AWS ECS vs AWS Lambda: What's The Difference & How To Choose
- AWS vs Azure vs GCP: Comparing The Big 3 Cloud Platforms
- Kubernetes Deployments Fully Explained
- The Spring Framework Beginner's Guide: Features, Architecture & Getting Started
- Explained: Monitoring & Telemetry in DevOp