AWS Decision Guide

# Choosing an AWS container service

# Choosing an AWS container service: AWS Decision Guide

# Table of Contents

# Choosing an AWS container service

**Taking the first step**

| | |
|---|---|
| **Purpose** | Determine which AWS container service is the best fit for your organization. |
| **Last updated** | May 16, 2025 |
| **Covered services** | <ul><li>[Amazon EC2](#)</li><li>[Amazon ECR](#)</li><li>[Amazon ECS](#)</li><li>[Amazon EKS](#)</li><li>[Amazon Lightsail](#)</li><li>[AWS App Runner](#)</li><li>[AWS Batch](#)</li><li>[AWS Fargate](#)</li><li>[AWS Lambda](#)</li><li>[AWS Outposts](#)</li><li>[Red Hat OpenShift Service on AWS (ROSA)](#)</li></ul> |

# Introduction

Containers are a key component of modern application development. They are the standard for organizing compute resources, and managing the content of your application deployments.

Containers provide a discrete reproducible compute environment for building software to deploy in the cloud. They also simplify packaging and dependency management. You can use them for everything from orchestrating web applications or very large multi-cluster estates to testing your work and doing a proof of concept on your laptop.

This decision guide helps you get started and choose the right AWS container service for your modern application development.

*This 3½-minute excerpt is from an 11-minute presentation at re:Invent 2023 by Umar Saeed, an AWS senior manager and solutions architect. He provides a quick overview of AWS container choices.*

# Understand

Containers offer a number of advantages for packaging, deploying, and running applications:

- **Portability:** Benefit from a consistent runtime environment that can run on any platform that supports the container runtime.

- **Scalability:** Scale applications up or down, based on demand, with lightweight and easily replicated containers.

- **Consistency:** Ensure that the application runs the same way in all environments with a consistent runtime environment.

- **Efficiency:** Use fewer resources than traditional virtual machines with lightweight containers.

- **Isolation:** Improve security and reliability with containers' process-level isolation, with which applications running in separate containers cannot interfere with each other, improving security and reliability.

- **Agility:** Reduce the time that it takes to bring new features or applications to market by quickly packaging and deploying applications.

You can think about the universe of AWS container services in three distinct layers:

- The **Compute capacity layer** is where your containers actually run. This layer consists of:

  - **Amazon Elastic Compute Cloud (Amazon EC2) instances:** These instances provide the underlying compute capacity for running containers. You can choose from a wide range of instance types and sizes to match your application requirements. EC2 instances can be used as the compute layer for both Amazon ECS and Amazon EKS.

  - **AWS Fargate:** Fargate is a serverless compute engine for containers with which you can run containers without managing the underlying infrastructure. It removes the need to provision and manage EC2 instances. You can use Fargate with both Amazon ECS and Amazon EKS.

  - **AWS Outposts:** AWS Outposts is a fully managed service that extends AWS infrastructure and services to your on-premises or hybrid environment. With AWS Outposts, you can run containers on AWS infrastructure deployed in your own data center.

- The **orchestration layer** schedules and scales your environment. This layer includes:

  - **Amazon Elastic Container Service (Amazon ECS):** Amazon ECS is a fully managed container orchestration service that simplifies the deployment, management, and scaling of containerized applications. You can use Amazon ECS to define tasks and services, handle service discovery, and manage the lifecycle of containers.

- **Amazon Elastic Kubernetes Service (Amazon EKS):** Amazon EKS is a managed Kubernetes service with which you can deploy, manage, and scale containerized applications using Kubernetes. It provides a highly available and secure Kubernetes control plane.

- **Red Hat OpenShift Service on AWS (ROSA):** ROSA is a fully managed service with which you can deploy and run Red Hat OpenShift clusters on AWS infrastructure. OpenShift is a popular enterprise-grade Kubernetes platform that extends the capabilities of Kubernetes with additional features and tools for building, deploying, and managing containerized applications.

- The **Vertical solutions layer** is a set of vertical integration services that provide higher-level and bundled services that simplify the process of deploying and managing applications. The AWS services in this layer are:

  - **AWS App Runner:** AWS App Runner is a fully managed service designed to simplify the deployment and use of containerized web applications and APIs. You provide your container image, and App Runner automatically builds, deploys, and scales your application. It handles the provisioning of the underlying compute resources, load balancing, and automatic scaling based on incoming traffic.

  - **Amazon Lightsail:** Amazon Lightsail is a cloud platform that offers pre-configured virtual private servers (instances) and other resources for running applications. It provides pre-defined configurations for quickly provisioning compute instances, databases, storage, and networking resources. Lightsail supports running containerized applications by provisioning instances with Docker pre-installed, aimed at easing the deployment and management of your containers.

  - **AWS Batch:** AWS Batch is a fully managed service with which you can run batch computing workloads on AWS. It dynamically provisions the optimal compute resources based on the volume and specific resource requirements of the batch jobs that you submit. It automatically handles job scheduling, resource provisioning, and scaling based on the workload requirements.

  - **Amazon Elastic Container Registry (Amazon ECR):** Amazon ECR is a fully managed container registry with which you can store, manage, and deploy container images. It is designed to provide secure and scalable storage for your container images and simplify provisioning containers with the desired images.

> ⓘ **Note**
>
> AWS provides a variety of ways to deploy and run containers. One of the first considerations is your preference for either a serverless operational model or a Kubernetes operation model. In practice, most customers use both to varying degrees.

> The choice of operating model is explored in-depth in the [Choosing a modern application strategy decision guide](#), which is a useful resource for anyone who wants to explore this question further. In addition, the [Containers and Serverless Recommendation Guide](#) takes you through the choices to make when choosing your operating model.

# Consider

It's important to choose a container service that aligns to your application requirements and operational preferences. The following section outlines some of the key criteria to consider when choosing a container service, as well as supporting tools and services.

Managed service and operation overhead

Organizations might choose to build on managed container services to reduce their operational overhead. This approach allows developers and operators to focus on their unique activities that add value, instead of on undifferentiated tasks.

Workload characteristics

Understanding your workload patterns can help you make architecture choices. Workload patterns can include web applications, API-based microservices, event-driven applications, streaming and messaging, data pipelines, IT automations, and more. Some workloads perform better or are more cost effective in one compute environment versus another type.

Application portability

Many customers want to ensure that their applications can run in—and be migrated or moved to—a different environment. It's important for them to be able to preserve choice, or run an application both on premises and in the cloud.

We recommend building competency in software architectures and build packaging with which you can readily port differentiating business logic between compute services.

Applications built using some technologies might run more effectively on some compute services rather than others.

Organization size and skills

The skills of your organization are a major factor when deciding which container services you use. The approach you take can require some investment in DevOps and Site Reliability

Engineer (SRE) teams. Building out an automated pipeline to deploy applications is common for most modern application development.

Some choices elevate the amount of management you need to do. For example, some organizations have skills and resources to run and manage a Kubernetes implementation, because they invest in strong SRE teams to manage Kubernetes clusters and find value in the associated skill portability. These teams handle frequent cluster upgrades. For example, Kubernetes has three major releases a year, and deprecates old versions.

Organization size is a key factor, as smaller organizations might have a more limited IT team made up of people fulfilling multiple roles, while larger enterprises can support hundreds of workloads in production at once.

Ease of deployment

Different AWS container services meet unique needs in terms of deployment complexity. Here's how each service is optimized for its own role:

- **AWS App Runner** offers the most straightforward path for you to deploy your application on the internet without managing or customizing the underlying infrastructure.

- **Amazon ECS** is a good choice if you need more control over the network and security configurations without sacrificing scale or features.

- **Amazon EKS** provides flexibility and control over application deployment and orchestration provided by Kubernetes technology.

# Choose

Now that you know the criteria by which you are evaluating your container options, you are ready to choose which AWS container services might be a good fit for your organizational requirements.

The following table highlights which services are optimized for which circumstances. Use the table to help determine which container services and tools are.

| Containers category | When would you use it? | Services |
|---|---|---|
| Capacity | Use when you want to run your containers on self- | AWS Fargate <br><br> Amazon EC2 |

| Containers category | When would you use it? | Services |
|---|---|---|
|  | managed or AWS managed compute. | AWS Outposts |
| Orchestration | Use when you need the capacity to deploy and manage up to thousands of containers. | Amazon ECS<br><br>Amazon EKS<br><br>Red Hat OpenShift Service on AWS |
| Vertical solutions | Use when you or your team don't have a lot of experience with containers or infrastructure. | AWS App Runner<br><br>AWS Lambda<br><br>Amazon Lightsail<br><br>AWS Batch |
| Tools and services with container support | Use for supporting your container operations. | Amazon ECR |
| On-premises | Use these services for flexibility in where you run your container-based applications. | Amazon ECS Anywhere<br><br>Amazon EKS Anywhere<br><br>Amazon EKS Hybrid Nodes |

# Use

You should now have a clear understanding of each AWS container service (and the supporting AWS tools and services) and which one might be the best fit for your organization and use case.

To explore how to use and learn more about each of the available AWS container services, we have provided a pathway to explore how each of the services work. The following section provides links to in-depth documentation, hands-on tutorials, and resources to get you started.

# Capacity

Amazon EC2

- **What is Amazon EC2?**

  Get an overview of Amazon EC2. This guide not only provides an introduction to the service, but also covers how to get started using it and then provides in-depth descriptions of key features and how to use them.

  [Explore the guides](#)

- **Amazon EC2 instance types**

  When you launch an EC2 instance, the instance type that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities, and is grouped in an instance family based on these capabilities. This guide walks you through EC2 instance types.

  [Explore the guide](#)

- **Amazon EC2 Auto Scaling with EC2 Spot Instances**

  Learn how to create a stateless, fault-tolerant workload using Amazon EC2 Auto Scaling with launch templates to request Amazon EC2 Spot Instances.

  [Get started with the tutorial](#)

AWS Fargate

- **Getting started with AWS Fargate**

  This guide explains the basics of AWS Fargate, a technology that you can use with Amazon ECS to run containers without having to manage servers or clusters of Amazon EC2 instances.

  [Explore the guide](#)

- **Getting started with the console using Linux containers on AWS Fargate**

  Get started with Amazon ECS on AWS Fargate by using the Fargate launch type for your tasks in the Regions where Amazon ECS supports AWS Fargate.

  [Explore the guide](#)

- **Creating a cluster with a Fargate Linux task using the AWS CLI**

  Set up a cluster, register a task definition, run a Linux task, and perform other common scenarios in Amazon ECS with the AWS CLI.

  [Explore the guide](#)

AWS Outposts

- **Getting started with AWS Outposts**

  Access the complete set of AWS Outposts technical documentation.

  [Explore the guides](#)
- **What is AWS Outposts?**

  Get an introduction to this fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises.

  [Explore the guide](#)
- **AWS Outposts servers pricing**

  Get details on the pricing of AWS Outposts servers.

  [Get details on pricing](#)

# Orchestration

Amazon ECS

- **Getting started with Amazon ECS**

  Get an introduction to the tools available to access Amazon ECS and introductory step-by-step procedures to run containers.

  [Explore the guide](#)
- **Tutorials for Amazon ECS**

  Learn how to perform common tasks—including the creation of clusters and VPCs—when using Amazon ECS.

[Get started with the tutorials](#)

- **Amazon ECS Immersion Day**

  This workshop expands on your foundational understanding of containers and provides practical experience scaling, monitoring, and managing container workflows using Amazon ECS and AWS Fargate.

  [Explore the workshop](#)

- **Deploy containers on Amazon ECS**

  Learn how to run a sample application on an Amazon ECS cluster behind a load balancer, test the sample application, and delete your resources to avoid charges.

  [Explore the guide](#)

Amazon EKS

- **Getting started with Amazon EKS**

  Learn more about Amazon EKS, 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.

  [Explore the guide](#)

- **Amazon EKS deployment**

  Explore Amazon EKS deployment options on AWS and learn how to use it to manage a general containerized application.

  [Explore the guide](#)

- **Amazon EKS workshop**

  Explore practical exercises to learn about Amazon EKS.

  [Visit the workshop](#)

Red Hat OpenShift Service on AWS

- **What is Red Hat OpenShift Service on AWS?**

  Learn how to use this managed service to build, scale, and deploy containerized applications with the Red Hat OpenShift enterprise Kubernetes platform on AWS.

  Explore the guide

- **Getting started with Red Hat OpenShift Service on AWS**

  Learn how to get started using Red Hat OpenShift Service on AWS (ROSA).

  Explore the guide

- **Why would you use ROSA?**

  Watch a video to learn when to use Red Hat OpenShift over standard Kubernetes and explore ROSA in depth.

  Watch the video

# Vertical solutions

AWS App Runner

- **What is AWS App Runner?**

  Learn when to use this service to deploy from source code or a container image directly to a scalable and secure web application in the AWS Cloud.

  Explore the guide

- **Getting started with AWS App Runner**

  Use this tutorial to configure the source code and deployment, the service build, and the service runtime to deploy your application to AWS App Runner.

  Use the tutorial

- **Deploy a web app using AWS App Runner**

  Follow these step-by-step instructions to deploy a containerized web application using AWS App Runner.

[Use the tutorial](#)

## AWS Lambda

- **What is AWS Lambda?**

  Learn how to use Lambda to run your code on a high-availability compute infrastructure and perform all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging.

  [Explore the guide](#)

- **AWS Lambda documentation**

  Work with AWS Lambda documentation to understand how you can use this service to run code without provisioning or managing servers and only pay for the compute time that you consume.

  [Explore the guides](#)

- **Working with Lambda container images locally**

  Learn how you can use a deployment package to deploy your function code to Lambda. Lambda supports two types of deployment packages: container images and `.zip` file archives.

  [Explore the guide](#)

## Amazon Lightsail

- **What is Amazon Lightsail?**

  Get the full story on Amazon Lightsail, including what it does and how you can benefit from it. This guide also includes step-by-step guidance to help you get started using Lightsail and then configure it to meet your needs.

  [Explore the guide](#)

- **Creating Lightsail container service images**

Learn how to create a container image on your local machine using a Dockerfile. You can then push it to your Lightsail container service to deploy it.

[Explore the guide](#)

- **Amazon Lightsail resource center**

Explore Lightsail tutorials, videos, and links to core concept documentation.

[Visit the resource center](#)

AWS Batch

- **What is AWS Batch?**

Learn how to use AWS Batch to run batch computing workloads in the AWS Cloud.

[Explore the guide](#)

- **Best practices for AWS Batch**

Consider this guidance on how to run and optimize your workloads when using AWS Batch.

[Explore the guide](#)

- **AWS Batch workshops center**

Use these workshops, organized in a progressive manner from beginner to advanced, to explore and learn AWS Batch.

[Explore the workshops](#)

# Tools and services with container support

Amazon ECR

- **Amazon ECR documentation**

Use the Amazon ECR documentation to explore the best ways to use this fully managed container registry.

[Explore the guides](#)

- **What is Amazon Elastic Container Registry (Amazon ECR)?**

  A guide to getting started with—and using—Amazon ECR.

  [Explore the guide](#)

- **Amazon ECR in Multi-Account and Multi-Region Architectures**

  Explore key considerations for Amazon ECR architectures that span across AWS accounts and AWS Regions, and architectures related to hypothetical customer use cases.

  [Read the blog post](#)

# On-premises

Amazon ECS Anywhere

- **What is Amazon ECS Anywhere?**

  Learn how Amazon ECS Anywhere provides support for registering an *external instance*, such as an on-premises server or virtual machine (VM), to your Amazon ECS cluster.

  [Explore the guide](#)

- **Amazon ECS Anywhere pricing**

  Use this pricing guide to understand Amazon ECS Anywhere pricing, which is based on a model in which you are charged based on the amount of time the instances you have registered to an Amazon ECS cluster are connected to the ECS control plane, rounded up to the nearest second.

  [Explore the pricing guide](#)

- **Amazon ECS Anywhere FAQs**

  Get answers to frequently asked questions about Amazon ECS Anywhere.

  [Explore the FAQs](#)

Amazon EKS Anywhere

- **Amazon EKS Anywhere documentation**

Use the documentation to understand the use of and best practices for Amazon EKS Anywhere.

Read the documentation

- **Amazon EKS Anywhere pricing**

Use this pricing guide to understand Amazon EKS Anywhere pricing.

Explore the pricing guide

- **Amazon EKS Anywhere FAQs**

Get answers to frequently asked questions about Amazon EKS Anywhere.

Explore the FAQs

Amazon EKS Hybrid Nodes

- **Amazon EKS Hybrid Nodes overview**

Use the documentation to understand the use of and best practices for Amazon EKS Hybrid Nodes.

Read the documentation

- **Amazon EKS Hybrid Nodes pricing**

Use the pricing guide to understand Amazon EKS Hybrid Nodes pricing.

Explore the pricing guide

- **A deep dive into Amazon EKS Hybrid Nodes**

Explore how you can use your existing on-premises and edge infrastructure as nodes in Amazon EKS clusters.

Read the blog post

# Explore

- **For your role**

- [Developers](#)

- [Solution Architects](#)

- [Professional development](#)

- [Startups](#)

- [Decision makers](#)

- **For an introduction**

  - [Docker](#)

  - [Kubernetes](#)

  - [Breaking a monolith into microservices](#)

- **For a video**

  - [Containers from the couch](#)

  - [How to containerize anything!](#)

  - [Building a container CI/CD pipeline](#)

  - [Building a container app with AWS CDK](#)

- **Architecture diagrams**

  Explore reference architecture diagrams for containers on AWS.

  [Explore architecture diagrams](#)

- **Whitepapers**

  Explore whitepapers to help you get started and learn best practices.

  [Explore whitepapers](#)

- **AWS solutions**

  Explore vetted solutions and architectural guidance for common use cases for containers.

  [Explore solutions](#)

# Document history

The following table describes the important changes to this decision guide. For notifications about updates to this guide, you can subscribe to an RSS feed.

| Change | Description | Date |
| --- | --- | --- |
| Guide updated | Added Amazon EKS Hybrid Nodes. Removed AWS Copilot and AWS Cloud Map. Numerous editorial updates throughout. | May 16, 2025 |
| Guide updated | Added AWS Copilot, AWS Batch, and AWS Outposts. Changed capacity, orchestration, and provisioning to compute capacity, orchestration, and vertical solutions. Numerous editorial changes throughout. | April 5, 2024 |
| Initial publication | Guide first published. | April 26, 2023 |