

# Choosing an AWS container service



---

# Choosing an AWS container service: AWS Decision Guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

---

# Table of Contents

<b>Decision guide</b> .....	<b>1</b>
Introduction .....	1
Understand .....	2
Consider .....	5
Choose .....	6
Use .....	7
Capacity .....	8
Orchestration .....	10
Vertical solutions .....	12
Tools and services with container support .....	16
On-premises .....	18
Explore .....	20
<b>Document history</b> .....	<b>21</b>

# Choosing an AWS container service

## Taking the first step

<b>Purpose</b>	Help determine which AWS container service is the best fit for your organization.
<b>Last updated</b>	April 5, 2024
<b>Covered services</b>	<ul style="list-style-type: none"><li>• <a href="#">Amazon EC2</a></li><li>• <a href="#">Amazon ECR</a></li><li>• <a href="#">Amazon ECS</a></li><li>• <a href="#">Amazon EKS</a></li><li>• <a href="#">Amazon Lightsail</a></li><li>• <a href="#">AWS App Runner</a></li><li>• <a href="#">AWS Batch</a></li><li>• <a href="#">AWS Copilot</a></li><li>• <a href="#">AWS Fargate</a></li><li>• <a href="#">AWS Lambda</a></li><li>• <a href="#">AWS Outposts</a></li><li>• <a href="#">Red Hat OpenShift Service on AWS (ROSA)</a></li></ul>

## Introduction

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

Containers provide a discrete reproducible compute environment. They also provide a way to simplify packaging and dependency management. From the orchestration of very large multi-cluster estates to web applications—or even testing your work and doing a proof of concept on your laptop—they are a great way to get started and build software to deploy in the cloud.

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

[This three and a half 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:** Through a consistent runtime environment that can run on any platform that supports the container runtime.
- **Scalability:** By being lightweight and easily replicated, containers can make it easier to scale applications up or down, based on demand.
- **Consistency:** Containers provide a consistent runtime environment, ensuring that the application runs the same way in all environments.
- **Efficiency:** Containers are lightweight and use fewer resources than traditional virtual machines.
- **Isolation:** Containers provide process-level isolation, meaning that applications running in separate containers cannot interfere with each other improving security and reliability.
- **Agility:** Containers allow you to package and deploy applications quickly and easily, reducing the time it takes to bring new features or applications to market.

# Options available to run containers on AWS



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

- **Compute capacity layer**, where your containers actually run. This layer consists of:
  - **Amazon Elastic Compute Cloud (Amazon EC2) instances:** These 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 that allows you to run containers without managing the underlying infrastructure. It removes the need to provision and manage EC2 instances. Fargate can be used 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.
- **Orchestration layer**, to schedule and scale 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. It supports Docker containers and allows you 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 that makes it easy to 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 that allows you to 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.
- **Vertical solutions layer,** 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 that enables you to 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 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 Docker container registry that allows you to store, manage, and deploy Docker container images. It is designed to provide secure and scalable storage for your container images and make it easier to provision containers with the desired images.

### A note on operating models

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 that you need 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 may choose the cloud to reduce operational cost by standardizing on managed services that shift the operational burden to AWS. Higher levels of abstraction allow developers and operators to focus on their own unique value-add activities, instead of undifferentiated tasks.

Building with containers on AWS uses services with higher levels of abstraction to shift the operational overhead of maintaining infrastructure to AWS.

### Workload characteristics

You can more easily make architecture choices by understanding your workload patterns. This may include web applications, API-based microservices, event-driven applications, streaming and messaging, data pipelines, IT automations, and more. Some workloads will perform better or be more cost effective in a one compute environment versus another type.

### Application portability

Many customers want to ensure that their applications can run in—and be easily 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 that allows you to more readily port differentiating business logic between compute services.

Applications built using some technologies might run more effectively on some compute services versus 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 may 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 simplest path, allowing you to deploy your application on the internet without managing or customizing the underlying infrastructure.
- **Amazon ECS** is a good choice if you need a little more control over the network and security configurations without sacrificing scale or features.
- **Amazon EKS** allows you to use the 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-managed AWS virtual machines or AWS managed compute.	<a href="#">AWS Fargate</a> <a href="#">Amazon EC2</a> <a href="#">AWS Outposts</a>
Orchestration	Use when you need to be able to deploy and manage up to thousands of containers.	<a href="#">Amazon ECS</a> <a href="#">Amazon EKS</a> <a href="#">Red Hat OpenShift Service on AWS</a>
Vertical solutions	Use when you or your team don't have a lot of experience with containers or infrastructure.	<a href="#">AWS App Runner</a> <a href="#">AWS Lambda</a> <a href="#">Amazon Lightsail</a> <a href="#">AWS Batch</a>
Tools and services with container support	Use for supporting your container operations.	<a href="#">AWS Copilot</a> <a href="#">Amazon ECR</a> <a href="#">AWS Cloud Map</a>
On-premises	Use these services for flexibility in where you run your container-based applications.	<a href="#">Amazon ECS Anywhere</a> <a href="#">Amazon EKS Anywhere</a>

## 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 of 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?



### What is AWS Outposts?

Access the complete set of AWS Outposts technical documentation.

[Explore the guides](#)



### **AWS Outposts servers pricing**

Get details on the pricing of AWS Outposts servers.

[Get details on pricing](#)

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

[Explore the guide](#)

## **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 Workshop

Use this workshop to become familiar with AWS Fargate, Amazon ECS, and possibly even Docker container workflow.

[Explore the workshop](#)

### Deploy Docker containers on Amazon ECS

Learn how to run a Docker-enabled 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 Quick Start Reference Deployment

Using a Quick Start reference deployment guide, we provide step-by-step instructions for deploying Amazon EKS clusters.

[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?

This video looks at when you might use Red Hat OpenShift over standard Kubernetes and explores 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](#)



## Deploy a web app using AWS App Runner

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

[Use the tutorial](#)

## 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](#)

## 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



## 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.



provisioning and automatic scaling, and logging.

[Explore the guide](#)



### 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](#)

[Explore the guides](#)

## 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.



### 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](#)

[Explore the guide](#)

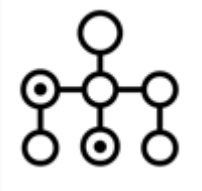


**Amazon Lightsail resource center**

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

[Visit the resource center](#)

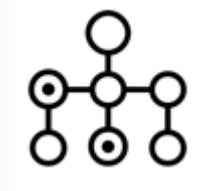
AWS Batch



**What is AWS Batch**

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

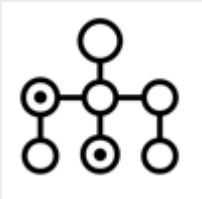
[Explore the guide](#)



**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

### AWS Copilot



#### Getting started with Amazon ECS using AWS Copilot

Get started with Amazon ECS using AWS Copilot by deploying an Amazon ECS application.

[Explore the guide](#)



#### AWS Copilot CLI documentation

Learn how to use the AWS Copilot CLI, a tool for developers to build, release, and operate production-ready containerized applications on AWS App Runner and Amazon ECS on AWS Fargate.

[Explore the documentation](#)



#### Introduction to Amazon ECS using AWS Copilot CLI

Learn how to deploy your application to Amazon ECS using AWS Copilot.

[Watch the video](#)

## 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](#)

## AWS Cloud Map



## AWS Cloud Map documentation

Use the AWS Cloud Map developer guide, API reference and AWS Cloud Map (service discovery) in the AWS CLI Reference to get the most from this service.

[Explore the guides](#)



## AWS Cloud Map FAQs

Get answers to frequently asked questions about AWS Cloud Map.

[Explore the FAQs](#)

## What is AWS Cloud Map?

Learn how you can use AWS Cloud Map to create and maintain a map of backend services and resources for your applications.

[Explore the guide](#)

## 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.



## Amazon ECS Anywhere pricing

Use this pricing guide to understand Amazon ECS Anywhere pricing, which is based on a model where you are charged based on the amount of time the instances you have registered to an Amazon ECS cluster are

[Explore the guide](#)

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](#)

## Explore

### For your role

- [Developers](#)
- [Solution Architects](#)
- [Professional development](#)
- [Startups](#)
- [Decision makers](#)

### Architecture diagrams

Explore reference architecture diagrams for containers on AWS.

[Explore architecture diagrams](#)

### For an introduction to

- [Docker](#)
- [Kubernetes](#)
- [Breaking a monolith into microservices](#)

### Whitepapers

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

[Explore whitepapers](#)

### For a video about

- [Containers from the couch](#)
- [How to containerize anything!](#)
- [Building a container CI/CD pipeline](#)
- [Building a container app with AWS CDK](#)

### 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
<a href="#">Guide updated</a>	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
<a href="#">Initial publication</a>	Guide first published.	April 26, 2023