

Choosing an AWS compute service



Choosing an AWS compute service: AWS Decision Guide

Copyright © 2025 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

Decision guide **1**

Introduction 2

Understand 2

Consider 7

Choose 13

Use 14

Amazon EC2 14

Container services 17

Serverless 21

On-premises and hybrid 22

Cost and savings optimization 23

Elastic Load Balancing 26

Explore 26

Document history **28**

Choosing an AWS compute service

Taking the first step

Purpose	Help determine which AWS compute service is the best fit for your organization.
Last updated	February 17, 2025
Covered services	<ul style="list-style-type: none">• Amazon EC2• Amazon EC2 Spot Instances• Amazon EC2 Auto Scaling• AWS Batch• AWS Elastic Beanstalk• Amazon Elastic Container Service• Amazon ECS Anywhere• Amazon Elastic Container Registry• Amazon Elastic Kubernetes Service• Amazon EKS Anywhere• AWS Fargate• AWS Lambda• AWS Local Zones• AWS Dedicated Local Zones• AWS Outposts• AWS Wavelength• Savings Plans• AWS Compute Optimizer• EC2 Image Builder• Elastic Load Balancing• Amazon Lightsail

Introduction

AWS compute services are designed to meet the varied demands of modern applications, from small-scale projects to enterprise-grade solutions. These services provide scalable computing power that helps you to build, deploy, and manage applications.

To get the most from your investment in these services, it's important to choose the right services for the right task or use case, whether it involves processing simple web app requests or running complex, data-intensive algorithms.

You can, for example, use:

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) to launch and manage virtual servers
- [AWS Elastic Beanstalk](#) to quickly deploy and manage web applications with minimal management of the infrastructure that runs those applications
- [AWS Lambda](#) to run code without having to provision or manage servers
- [Amazon Elastic Container Service \(Amazon ECS\)](#) or [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#) to run and manage [containers](#)
- [AWS Fargate](#) to run containers on AWS managed compute, or [AWS Batch](#) to process large volumes of data in parallel

You can also use multiple types of compute solutions in a single workload, as each one has its own advantages.

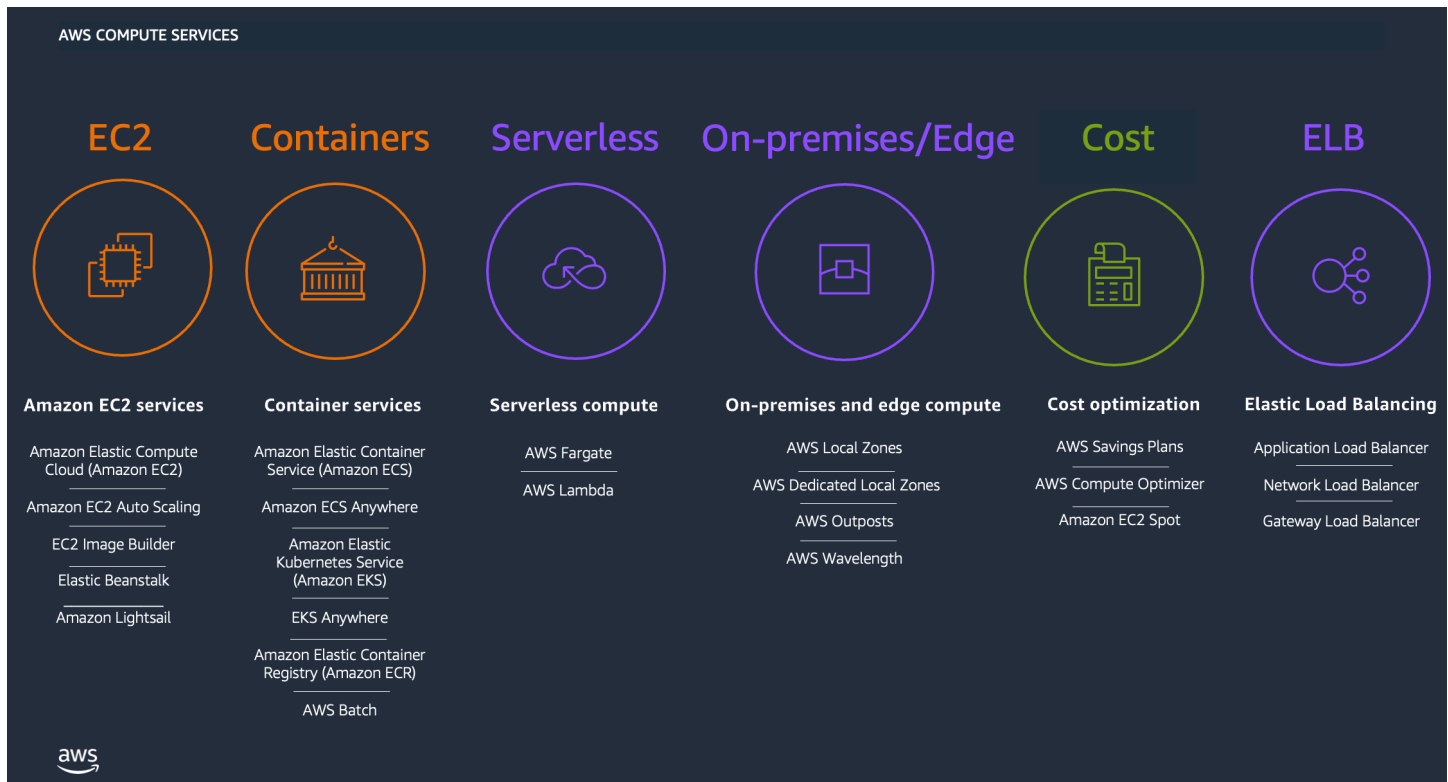
This guide will help you select the AWS compute services and tools that are the best fit for your needs and your organization.

[This two minute excerpt is from a presentation at re:Invent 2023 by Lorenzo Winfrey, an AWS Senior Business Development Specialist. He provides a quick overview of AWS compute application use cases.](#)

Understand

AWS compute services provide secure and resizable compute capacity in the cloud. AWS offers a range of compute services to meet various application requirements. These include Amazon EC2 for resizable virtual servers, Elastic Beanstalk for deploying web applications on your choice of a number of popular platforms, AWS Lambda for serverless computing, Amazon ECS and EKS for container orchestration, and AWS Fargate for serverless containers.

Furthermore, AWS Batch facilitates batch computing. AWS hybrid and edge services such as [AWS Local Zones](#) and [AWS Outposts](#) bring AWS infrastructure and services to metropolitan areas, on-premises locations, and edge sites, addressing requirements for low latency, digital sovereignty, and local data processing. Additionally, Amazon EC2 Auto Scaling automatically adjusts capacity. These services cater to different workload needs, from basic virtual machines (VMs) to fully managed serverless and container solutions.



Amazon EC2 services

Amazon EC2 offers a wide range of instance types tailored to different workloads and applications. You can choose from a range of configurations, including different combinations of CPU, memory, storage, and networking capacity. EC2 provides a large variety of compute capacity options optimized for various workloads: general purpose, compute optimized, memory optimized, storage optimize, accelerated computing, and high-performance computing.

- [Amazon EC2](#): Amazon EC2 provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud, including AWS Graviton4 instances for improved price-performance and lower latency in compute-intensive applications, Trn2 instances for high-performance AI/ML training and inference, and P5eN instances optimized for deep learning (DL), high-performance computing (HPC) workloads, and generative AI applications.

- [Amazon EC2 Auto Scaling](#): Amazon EC2 Auto Scaling helps you maintain application availability and lets you automatically add or remove Amazon EC2 instances by using scaling policies that you define.
- [EC2 Image Builder](#): EC2 Image Builder is a fully managed AWS service that helps you to automate the creation, management, and deployment of customized, secure, and up-to-date server images.
- [AWS Elastic Beanstalk](#): AWS Elastic Beanstalk supports applications developed in Python, Node.js, Java, .NET, PHP, Ruby, and GO. Elastic Beanstalk also supports Docker platforms. With Docker containers you can choose your own programming language and application dependencies that may not be supported by the other Elastic Beanstalk platforms.
- [Amazon Lightsail](#): Amazon Lightsail provides an easy way to build web applications providing instances, container services, managed databases, content delivery network distributions, load balancers, SSD-based storage, and DNS management.

Container services

AWS container compute options are designed to help you deploy, manage, and scale containerized applications efficiently. Amazon ECS, for example, allows you to run and manage Docker containers at scale, handling cluster management and orchestration of containers. Amazon EKS provides a fully managed Kubernetes service, simplifying the deployment, management, and scaling of containerized applications using Kubernetes.

You have the option of running containers on EC2 instances that you manage, or you can run them on Fargate on AWS managed compute. Additionally, AWS offers Amazon Elastic Container Registry (Amazon ECR), a fully managed Docker container registry.

- [Amazon ECS](#): Amazon ECS is a fully managed container orchestration service that helps you easily deploy, manage, and scale containerized applications.
- [Amazon ECS Anywhere](#): Amazon ECS Anywhere provides support for registering an *external instance* such as an on-premises server or VM, to your Amazon ECS cluster.
- [Amazon EKS](#): Amazon EKS is a managed service that helps you easily deploy, manage, and scale containerized applications using Kubernetes on AWS infrastructure. Amazon EKS removes the need to install, operate, and maintain your own Kubernetes control plane on AWS.
- [Amazon EKS Anywhere](#): Amazon EKS Anywhere is a container management software built by AWS that makes it easier to run and manage Kubernetes clusters on-premises and at the edge.

- [Amazon ECR](#): Amazon ECR is an AWS managed container image registry service that is secure, scalable, and reliable.
- [AWS Batch](#): AWS Batch is a fully managed batch computing service that plans, schedules, and runs your containerized batch machine learning, simulation, and analytics workloads across the full range of AWS compute offerings, such as Amazon ECS, Amazon EKS, Fargate, and Spot or On-Demand Instances.

Serverless compute

AWS offers serverless compute options, including AWS Lambda and AWS Fargate, which allow you to run your workloads without provisioning or managing servers. As a result, developers can focus on writing code by shifting as much management of the underlying infrastructure resources to AWS.

- [AWS Fargate](#): AWS Fargate is a technology that you can use with Amazon ECS to run containers without having to manage servers or clusters of Amazon EC2 instances.
- [AWS Lambda](#): Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging.

On-premises and edge compute

AWS provides hybrid and edge compute options that allow you to extend AWS infrastructure and services to your premises and the edge. These edge and hybrid compute options provide flexibility and scalability for a wide range of use cases across different network environments.

- [AWS Local Zones](#): AWS Local Zones places compute, storage, database, and other select AWS resources close to large population and industry centers. You can use Local Zones to provide your users with low-latency access to your applications.
- [AWS Dedicated Local Zones](#): AWS Dedicated Local Zones are a type of AWS Infrastructure that is fully managed by AWS, built for exclusive use by a customer or community, and placed in a customer-specified location or data center to comply with regulatory requirements.
- [AWS Outposts](#): AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises.

- [AWS Wavelength](#): AWS Wavelength helps developers to build applications that deliver ultra-low latencies to mobile devices and end users. Wavelength deploys standard AWS compute and storage services to the edge of communications service providers' (CSPs') 5G networks.

Cost optimization

AWS provides cost optimization services that allow you to reduce your AWS costs by committing to a usage level and generating recommendations to reduce the cost of your workloads.

- [Savings Plans](#): Savings Plans is a flexible pricing model that can help you reduce your bill compared to On-Demand prices, in exchange for a one- or three-year hourly spend commitment.
- [AWS Compute Optimizer](#): AWS Compute Optimizer provides artificial intelligence and machine learning-based analytics to help you right size your workloads, reduce costs, and improve the performance of your workloads.
- [Amazon EC2 Spot Instances](#): Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity at a significant discount to On-Demand pricing, allowing you to lower your Amazon EC2 costs significantly.

Elastic Load Balancing

[Elastic Load Balancing \(ELB\)](#) automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones.

- [Application Load Balancer](#): An Application Load Balancer functions at the application layer, the seventh layer of the OSI model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action.
- [Network Load Balancer](#): A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the target group for the default rule.
- [Gateway Load Balancers](#): Gateway Load Balancers help you to deploy, scale, and manage virtual appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems.

Consider

Here are some key factors to consider when choosing an AWS compute service. Choosing the right AWS compute service involves balancing these factors to match your specific workload needs, technical requirements, and business objectives, to help you optimize for performance, cost, and ease of management.

Workload type and requirements

Workload planning involves understanding the operational patterns and technical demands of your applications. For instance, **batch processing jobs**, which often involve running a series of tasks or a program on a large volume of data, require robust compute capacity, which can be scaled down upon job completion to manage costs effectively.

Web applications, on the other hand, demand high availability and consistent performance to serve end-user requests at any scale. These workloads typically need a compute service that can dynamically adjust to fluctuating traffic patterns, ensuring smooth user experiences during demand surges and cost savings during quieter periods. Elastic Beanstalk is specifically designed for running web applications. It provides the runtime on EC2 instances so you just need to deploy your application code.

Machine learning models introduce a different set of requirements, with distinct phases for training and inference. Training is computationally intensive and often requires specialized hardware accelerators such as GPUs or custom chips for a limited duration but at a high-performance level. Inference, however, might call for a highly available environment that can quickly respond to prediction requests, often benefitting from optimized compute services that support auto-scaling and low-latency processing.

Choosing the right compute service entails matching these workload characteristics with the specific capabilities of each service to achieve operational efficiency, performance optimization, and cost management.

Performance needs

Performance encompasses the compute power—CPU and GPU capabilities—for processing, the memory for data caching and operations, storage I/O for data throughput, and network bandwidth for data transfer.

High-performance applications, such as those involving complex calculations or data-intensive processing, might require robust and fast CPUs or GPUs, found in compute-optimized or

GPU-based EC2 instances. In contrast, **memory-intensive applications**, like those running large databases or in-memory caches, necessitate memory-optimized instances with a higher memory-to-CPU ratio.

Furthermore, applications with **heavy input/output operations**, such as high-traffic web applications or big data processing systems, need storage-optimized instances with high I/O throughput. Lastly, network performance is crucial for **distributed systems and applications** that require rapid data transfer across instances or services.

Optimizing for performance means choosing services that not only meet your current demands but can also scale with your application's growth, ensuring consistent, high-quality user experiences without overspending on resources.

Scalability

Scalability is a critical criterion in the selection of AWS compute services, as it defines the ability of the system to handle growth and manage increased demand. Effective scalability ensures that your application can accommodate more users, handle more transactions, and process larger datasets without degrading performance.

The scalability of a service can be **vertical**, allowing you to scale up by adding more power to your existing infrastructure (like increasing the CPU or memory of an instance), or **horizontal**, helping you to scale out by adding more instances to handle the load. Horizontal scalability is essential for dynamic workloads with fluctuating demands.

Choosing a service that can automatically adjust its scale, like AWS's Auto Scaling or serverless offerings, can provide the flexibility to seamlessly manage workload spikes and lulls. This not only maintains performance levels but also optimizes costs, as you only pay for the resources you use. Scalability considerations help you make sure that your infrastructure is resilient, cost-effective, and capable of supporting your application as it evolves.

Management overhead

Management overhead refers to the amount of effort and resources required to manage and maintain your computing infrastructure. In AWS, this can range from hands-on management of virtual servers to AWS managed services.

For instance, **managing** EC2 instances involves responsibility for setup, scaling, patching, and securing servers. This can be resource-intensive, requiring a dedicated operations team. However, for use cases where granular control over the compute environment is necessary, the overhead is often justified.

On the other hand, AWS offers services that **abstract** away much of the infrastructure management. Serverless computing options like AWS Lambda run code in response to events without the need to manage servers, reducing the operational burden. Managed container services help to optimize deployment and scaling of containers, providing a middle ground between control and convenience. Elastic Beanstalk provides managed updates for OS patches, runtime and framework on EC2 instances it manages so customers can just run their application code.

Balancing management overhead involves assessing your organization's operational capacity and expertise, the need for control versus convenience, and the overall cost of ownership, including the hidden costs of operational labor. Selecting a service that aligns with your management capabilities ensures operational efficiency and allows you to focus on innovation and building applications.

Cost optimization

Cost optimization is a key consideration when using AWS compute services, as it helps you make sure that you're obtaining the most economical solution for your specific needs without sacrificing performance and scalability. Consider the following to save money on compute on AWS.

1. Selecting the right instance

AWS has more than 750 instance types, with most of these instances built on the [AWS Nitro System](#). Each instance type provides a choice of processor, storage, networking, operating system, and size, so you can choose the instance configuration that best fits your specific workload and budget.

[AWS Graviton-based Instances](#) are designed to deliver the best price performance for your cloud workloads running in Amazon EC2. EC2 instances powered by AWS Graviton processors deliver up to 40% better price performance than comparable non-Graviton-based instances.

[Accelerated computing instances](#) such as AWS Trainium-based EC2 Trn1 instances and AWS Inferentia-based Inf2 instances are designed to deliver high performance at the lowest cost in EC2 for your machine learning training and deployment needs.

2. Choosing the right purchase plans

You have several purchase models to choose from to maximize saving.

[On-Demand Instances](#) let you pay for compute capacity by the hour or second with no long-term commitments. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware.

[Savings Plans](#) is a flexible pricing model that can help you reduce your bill by up to 72% compared to On-Demand prices, in exchange for a one- or three-year hourly spend commitment.

[Amazon EC2 Spot Instances](#) let you take advantage of unused EC2 capacity in the AWS Cloud and are available at a discount of up to 90% compared to On-Demand prices.

3. Right size your workloads

With AWS, you scale up or down as demand fluctuates. The following tools help you to right size your workload.

Amazon EC2 Auto Scaling helps you maintain application availability and helps you to automatically add or remove EC2 instances by using scaling policies that you define.

Compute Optimizer provides artificial intelligence and machine learning-based analytics to help you right size your workloads and reduce costs by up to 25%. [AWS Trusted Advisor](#) can help you identify unused resources and opportunities to lower your costs.

Latency and throughput

Latency and throughput are crucial performance metrics that influence the responsiveness and data handling capacity of your applications. **Low latency** is essential for time-sensitive applications, ensuring that user interactions or transactions are processed rapidly. **High throughput** is necessary for applications that need to process large volumes of data quickly, such as video streaming services or big data analysis platforms.

When selecting AWS compute services, consider the **geographical distribution** of your user base and the location of AWS data centers. Proximity to users can drastically reduce latency. Additionally, services that offer edge computing capabilities can further minimize latency by processing data closer to the source.

For throughput, you'll need to evaluate **network bandwidth** and the **I/O capacity** of the compute service, ensuring that the service can handle peak data loads efficiently. The right

selection can prevent bottlenecks, ensuring data is processed, analyzed, and transferred without delays, thus maintaining optimal application performance.

Compliance and security

Compliance and security are critical factors when choosing AWS compute services, as they ensure that the infrastructure adheres to regulatory standards and protects data integrity. Compliance with industry-specific frameworks, such as [Health Insurance Portability and Accountability Act \(HIPAA\)](#) for healthcare, [General Data Protection Regulation \(GDPR\)](#) for data protection in the EU, or [AWS System and Organization Controls \(SOC\)](#) for service organizations, is non-negotiable for many businesses. AWS provides a suite of services designed to meet these rigorous standards, offering features like data encryption, identity and access management, and logging and monitoring.

Security encompasses not just compliance but also the broader protection of your infrastructure from unauthorized access and cyber threats. The chosen service should offer robust security features, including network security groups, firewalls, and options for private connectivity. Additionally, AWS regularly attains third-party validations for its security and compliance controls, providing an assurance that its infrastructure can support the necessary compliance and protect sensitive data, thereby maintaining trust and integrity. Choosing a compute service that aligns with these requirements is essential for risk management and maintaining customer trust.

AWS Nitro System provides enhanced security that continuously monitors, protects, and verifies the instance hardware and firmware. Virtualization resources are offloaded to dedicated hardware and software minimizing the attack surface. Nitro System's security model is locked down and prohibits administrative access, eliminating the possibility of human error and tampering. AWS Local Zones, AWS Dedicated Local Zones, AWS Outposts, and AWS Wavelength are all built on the same Nitro System that powers modern EC2 instances in the AWS Regions today.

Integration

Integration within AWS pertains to how compute services work in concert with other AWS offerings and third-party applications. It's crucial to select a compute service that seamlessly connects with databases, analytics, machine learning, or other services that form the backbone of your cloud architecture. This approach should allow for a cohesive workflow, where data and processes can move smoothly across services, facilitating automation and reducing the need for manual intervention.

The chosen compute service should offer APIs, SDKs, and integration points that align with your existing tools and practices, enabling straightforward implementation into your development pipeline. AWS breadth of services and its extensive [partner network](#) can enhance application capabilities and accelerate innovation. Additionally, the availability of pre-built integrations can save development time and resources. Therefore, evaluating the compatibility of the compute service with the broader use is essential for building a scalable, agile, and efficient cloud environment.

Reliability and availability

Reliability and availability are paramount when selecting AWS compute services, as they determine the resilience and uptime of your applications. Reliability ensures that the service can consistently perform its intended function correctly under specific conditions, while availability measures the proportion of time the service is operational and accessible.

In assessing these criteria, consider the service's track record for stability and its ability to recover from failures. Look for features such as redundancy, failover processes, and backup capabilities that ensure continuous operation. AWS services often come with Service Level Agreements (SLAs) that guarantee a certain percentage of uptime.

Moreover, the ability to deploy across multiple Availability Zones can mitigate the impact of outages, while the global spread of AWS Regions can help ensure that your application remains available to users worldwide even during regional disruptions. The choice of a compute service with high reliability and availability minimizes potential downtime, maintaining business continuity and safeguarding user experience.

Development and deployment experience

The development and deployment experience are important when choosing AWS compute services, as it directly impacts the efficiency and agility of your development team. Services that offer streamlined workflows, comprehensive documentation, and robust tooling can significantly reduce the time to market for new features and applications.

Consider whether the service integrates well with your existing CI/CD pipelines, supports your preferred development languages and frameworks, and provides easy-to-use SDKs and APIs for seamless application integration. Services that offer containerization and serverless computing can simplify deployment and management, allowing developers to focus more on writing code and less on infrastructure concerns.

Moreover, the availability of detailed monitoring, logging, and debugging tools within the AWS Cloud can enhance the development experience, enabling quick identification and resolution

of issues. Choosing a compute service that aligns with your team's skills and workflows can foster a more productive and innovative development environment, ultimately driving business success.

Choose

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

The following table highlights which services are optimized for which circumstances.

Compute category	What is it optimized for?	Compute services
Amazon EC2	Providing scalable high-performance computing resources for CPU-intensive workloads.	Amazon EC2 Amazon EC2 Auto Scaling EC2 Image Builder AWS Elastic Beanstalk Amazon Lightsail
Container services	Helping your teams focus on building applications rather than the runtime environment or managing a control plane.	AWS Batch Amazon ECS Amazon ECS Anywhere Amazon EKS Amazon EKS Anywhere AWS Elastic Beanstalk Amazon ECR
Serverless compute	Minimizing your AWS management overhead, allowing you to focus on	AWS Fargate AWS Lambda

Compute category	What is it optimized for?	Compute services
	implementing your business logic.	
On-premises and edge compute	Allowing you to run familiar AWS interfaces to your premises and the edge, providing lower latency and local data processing needs.	AWS Local Zones AWS Dedicated Local Zones AWS Outposts AWS Wavelength
Cost and savings optimization	Helping you reduce your AWS costs for your workloads.	AWS Savings Plan AWS Compute Optimizer Amazon EC2 Spot Instances
Elastic Load Balancing	Increasing the availability and fault tolerance of your applications.	Application Load Balancer Network Load Balancer Gateway Load Balancer

Use

You should now have a clear understanding of each AWS compute 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 compute services, we have provided a pathway to learn how each of the services work. The following section provides links to in-depth documentation, hands-on tutorials, and resources to get you started.

Amazon EC2

Amazon EC2

- **Tutorial: Get started with Amazon EC2 Linux instances**

Use this tutorial to get started with Amazon EC2. You'll learn how to launch, connect to, and use a Linux instance.

[Use the tutorial](#)

- **Tutorial: Get started with Amazon EC2 Windows instances**

Use this tutorial to get started with Amazon EC2. You'll learn how to launch, connect to, and use a Windows instance.

[Use the tutorial](#)

- **Amazon EC2 instance types**

This guide will give you an overview of the various families of EC2 instance types and discuss the appropriate application for each family.

[Explore the guide](#)

- **Get Started with Amazon EC2 Graviton instances**

This guide will help you get started with Amazon EC2 Graviton instances and provides steps-by-step instructions to migrate your workload to Graviton.

[Explore the guide](#)

Amazon EC2 Auto Scaling

- **Get started with Amazon EC2 Auto Scaling**

In this tutorial, you will setup an Auto Scaling group, terminate your instance and verify the instance was removed from service and replaced.

[Use the tutorial](#)

- **Tutorial: Scale the size of your Auto Scaling Group**

In this tutorial, you will learn how to scale your Auto Scaling group using either manual scaling, scheduled scaling, dynamic scaling, or predictive scaling.

[Use the tutorial](#)

- **Amazon EC2 Auto Scaling FAQs**

Dive deep into the intricacies of EC2 Auto Scaling by reviewing the FAQ.

[Explore the FAQ](#)

EC2 Image Builder

- **Get started with EC2 Image Builder**

This guide will help you set up your environment and create an automated image pipeline for the first time.

[Use the tutorial](#)

- **Building golden images using Amazon EC2 Image Builder workshop**

This workshop will guide you through creating an EC2 Image Builder pipeline and then developing your own custom components.

[Use the workshop](#)

- **Implementing up-to-date images with automated EC2 Image Builder pipelines**

This post demonstrates how to automatically keep your base or standard images current, incorporating patches and any other changes using EC2 Image Builder pipelines.

[Read the blog](#)

AWS Elastic Beanstalk

- **Get started with AWS Elastic Beanstalk**

This tutorial walks you through creating, exploring, updating, and deleting an Elastic Beanstalk application. It takes less than an hour to complete.

[Use the tutorial](#)

- **Security best practices for Elastic Beanstalk**

This guide provides you with general guidelines for securing your Elastic Beanstalk application.

[Read the guide](#)

- **Creating an ECS managed Docker environment with the Elastic Beanstalk console**

In this tutorial, you'll implement an ECS managed Docker environment that uses two containers.

[Use the tutorial](#)

Amazon Lightsail

- **Launch a Linux virtual machine with Amazon Lightsail**

In this tutorial, you create an Amazon Linux instance in Amazon Lightsail in seconds. After the instance is up and running, you connect to it via SSH within the Lightsail console using the browser-based SSH terminal.

[Use the tutorial](#)

Container services

AWS Batch

- **Getting started with AWS Batch – Amazon EC2**

In this tutorial, you will set up an AWS Batch compute environment using Amazon EC2 orchestration.

[Use the tutorial](#)

- **Getting started with AWS Batch - Fargate**

In this tutorial, you will set up an AWS Batch compute environment using AWS Fargate.

[Use the tutorial](#)

- **Getting started with AWS Batch - Amazon EKS**

In this tutorial, you will set up an AWS Batch compute environment using Amazon EKS.

[Use the tutorial](#)

- **AWS Batch Deep Dive workshop**

This workshop provides a deep dive into the basic concepts and use of AWS Batch.

[Use the workshop](#)

Amazon Elastic Container Service

- **Getting started with Amazon ECS**

We provide 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**

Explore more than a dozen tutorials on how to perform common tasks - including the creation of clusters and VPCs.

[Get started with the tutorials](#)

- **What's new and what's next with Amazon ECS**

Learn what's new since the launch of Amazon ECS Anywhere, new features of AWS Fargate, and a look ahead at the exciting enhancements to Amazon ECS.

[Watch the video](#)

- **Amazon ECS deployment**

This guide offers an overview of Amazon ECS deployment options on AWS and shows how it can be used to manage a simple containerized application.

[Explore the guide](#)

- **Amazon ECS workshop**

This workshop is designed to educate those that might not be familiar with AWS Fargate, Amazon ECS, and 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 application, and delete your resources to avoid charges.

[Get started with the tutorial](#)

Amazon ECS Anywhere

- **Amazon ECS Anywhere FAQs**

Answers to frequently-asked questions about Amazon ECS Anywhere.

[Read the FAQ](#)

- **Registering an external instance with Amazon ECS Anywhere**

Amazon ECS Anywhere provides support for registering an external instance such as an on-premises server or VM, to your Amazon ECS cluster. Here's how to use that support.

[Explore the guide](#)

- **Getting Started with Amazon ECS Anywhere**

Amazon ECS Anywhere provides consistent tooling and APIs for all container-based applications and the same Amazon ECS experience for cluster management, workload scheduling, and monitoring both in the cloud and on customer-managed infrastructure. This blog details how and why you might want to use it.

[Read the blog](#)

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 it can be used to manage a general containerized application.

[Explore the guide](#)

- **Amazon EKS cluster deployment**

Use this guide to create an Amazon EKS cluster.

[Explore the guide](#)

- **Deploy a Kubernetes application**

Learn how to deploy a containerized application onto a Kubernetes cluster managed by Amazon EKS.

[Expolre the guide](#)

- **Amazon EKS workshop**

Explore practical exercises to learn about Amazon Elastic Kubernetes Service.

[Visit the workshop](#)

Amazon EKS Anywhere

- **Getting started with Amazon EKS Anywhere**

This guide helps you get started with Amazon EKS Anywhere, container management software built by AWS that makes it easier to run and manage Kubernetes clusters on-premises and at the edge.

[Explore the guide](#)

- **Amazon EKS Anywhere FAQs**

Get answers to your frequently asked questions about Amazon EKS.

[Read the FAQs](#)

- **Running Hybrid Container workloads with Amazon EKS Anywhere**

This whitepaper provides cloud engineers and architects best practices for operating Amazon EKS Anywhere on customer-managed infrastructure.

[Read the whitepaper](#)

Amazon ECR

- **Getting started with Amazon ECR**

This guide explains how to use Amazon ECR, an AWS managed container image registry service that is secure, scalable, and reliable.

[Explore the guide](#)

- **Amazon ECR FAQs**

Answers to frequently-asked questions about Amazon ECR.

[Read the FAQs](#)

Serverless

AWS Fargate

- **AWS Fargate for Amazon ECS**

Understand 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](#)

- **Learn how to create an Amazon ECS Linux task for the Fargate launch type**

Get started with Amazon ECS on 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**

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

[Get started with the tutorial](#)

AWS Lambda

- **What is AWS Lambda?**

Learn more about AWS Lambda, a compute service that lets you run code without provisioning or managing servers.

[Explore the guide](#)

- **Guide to AWS Lambda Pricing**

Explore and understand AWS Lambda pricing. You are charged based on the number of requests for your functions and the duration it takes for your code to start.

[Visit the page](#)

- **Using AWS Lambda with other services**

Explore common use cases and learn how invocation works. Navigate a table that covers the services that work with Lambda and how it can be invoked from that service.

[Explore the guide](#)

On-premises and hybrid

AWS Outposts

- **Get started with AWS Outposts**

This guide will demonstrate how to order AWS Outposts and launch an Amazon EC2 instance on your on-premises network.

[Explore the guide](#)

- **Planning an AWS Outposts implementation**

In this free AWS Skill Builder course, you learn about implementation planning for AWS Outposts, including security responsibilities, facilities requirements, networking requirements, and where to deploy your code.

[Start the course](#)

- **AWS Outposts Rack FAQs**

Dive deep into AWS Outposts Rack by reviewing the FAQ.

[Explore the FAQ](#)

- **AWS Outposts Server FAQs**

Dive deep into AWS Outposts Servers by reviewing the FAQ.

[Explore the FAQ](#)

AWS Local Zones

- **Get started with AWS Local Zones**

This guide will walk you through the steps to enable a Local Zone through the Amazon EC2 console and create resources in the Local Zone subnet.

[Explore the guide](#)

- **Connectivity options for Local Zones**

This guide discusses the various ways to connect users and applications to resources running in a Local Zone.

[Use the workshop](#)

- **Deploying Network Functions in AWS Local Zones for Edge Compute Telco use cases**

This workshop will teach you about AWS Local Zones and the benefits of distributed network functions and applications.

[Use the workshop](#)

- **Deploying your first 5G enabled application with AWS Wavelength**

In this blog, you will walk you through deploying one of the most common Wavelength use cases: machine learning inference.

[Read the blog](#)

Cost and savings optimization

AWS Savings Plans

- **Get started with AWS Savings Plans**

This guide will walk you through enabling your settings and permissions in Cost Explorer before using the AWS Billing and AWS Cost Management Console to view, analyze, and manage your Savings Plans.

[Explore the guide](#)

- **Blog post: Getting Started with AWS Savings Plans**

This post covers how you can use the AWS Cost Management product suite to purchase, manage, and monitor Savings Plans.

[Read the blog](#)

- **Savings Plans FAQ**

Dive deep in to the details of AWS Savings Plans FAQ by reviewing the FAQ.

[Explore the FAQ](#)

- **AWS cost management decision guide**

Get what you need to decide how best to optimize AWS costs.

[Read the guide](#)

Amazon EC2 Spot Instances

- **Best practices for Amazon EC2 Spot Instances**

This guide will cover the best practices to have the best experience using Amazon EC2 Spot Instances.

[Explore the guide](#)

- **Work with Spot Instances**

This guide will cover the details on use and work with Spot Instances.

[Explore the guide](#)

- **Amazon EC2 Spot Instances workshop**

This set of workshops are designed to get you familiar with Amazon EC2 Spot Instances and how to use them in different scenarios, highlighting best practices to follow when using Spot Instances.

[Use the workshops](#)

Amazon EC2 Auto Scaling

- **Get started with Amazon EC2 Auto Scaling**

In this tutorial, you will setup an Auto Scaling group, terminate your instance and verify that the instance was removed from service and replaced.

[Use the tutorial](#)

- **Tutorial: Increase or decrease compute capacity of your application with scaling**

In this tutorial, you will learn how to scale your Auto Scaling group using either manual scaling, scheduled scaling, dynamic scaling, or predictive scaling.

[Use the tutorial](#)

- **Amazon EC2 Auto Scaling FAQs**

Dive deep into the intricacies of EC2 Auto Scaling by reviewing the FAQ.

[Explore the FAQ](#)

AWS Compute Optimizer

- **Get started with AWS Compute Optimizer**

This guide will walk you through opting into AWS Compute Optimizer.

[Explore the guide](#)

- **AWS Compute Optimizer workshop**

The goal of this lab is to use AWS Compute Optimizer to gain insights into rightsizing recommendations to optimize your migrated environment on AWS.

[Use the workshop](#)

- **AWS Compute Optimizer FAQs**

Dive deep in to the details of AWS Compute Optimizer FAQ by reviewing the FAQ.

[Explore the FAQ](#)

Elastic Load Balancing

Elastic Load Balancing

- **Get started with Application Load Balancers**

This tutorial provides a hands-on introduction to Application Load Balancers through the AWS Management Console, a web-based interface.

[Use the tutorial](#)

- **Getting started with Network Load Balancers**

This tutorial provides a hands-on introduction to Network Load Balancers through the AWS Management Console, a web-based interface.

[Use the tutorial](#)

- **Getting started with Gateway Load Balancers**

In this tutorial, you'll implement an inspection system using a Gateway Load Balancer and a Gateway Load Balancer endpoint.

[Use the tutorial](#)

Explore

- **Architecture diagrams**

Explore reference architecture diagrams for compute on AWS.

[Explore architecture diagrams](#)

- **Whitepapers**

Explore whitepapers to help you get started and learn best practices for compute services and use cases.

[Explore whitepapers](#)

- **AWS Solutions**

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

[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 content about Elastic Beanstalk	February 17, 2025
Initial publication	Guide first published.	June 24, 2024