Microservices

Build highly available microservices to power applications of any size and scale.

Get Started with
Microservices on AWS

| What are Microservices? | Benefits | Microservices on AWS | Getting Started |

What are Microservices?

Microservices are an architectural and organizational approach to software development where software is composed of small independent services that communicate over well-defined APIs. These services are owned by small, self-contained teams.

Microservices architectures make applications easier to scale and faster to develop, enabling innovation and accelerating time-to-market for new features.

Monolithic vs. Microservices Architecture

improving a monolithic application's features becomes more complex as the code base grows. This complexity limits experimentation and makes it difficult to implement new ideas. Monolithic architectures add risk for application availability because many dependent and tightly coupled processes increase the impact of a single process failure.

Breaking a monolithic application into microservices

With a microservices architecture, an application is built as independent components that run each application process as a service. These services communicate via a well-defined interface using lightweight APIs. Services are built for business capabilities and each service performs a single function. Because they are independently run, each service can be updated, deployed, and scaled to meet demand for specific functions of an application.

Characteristics of Microservices

Autonomous

Each component service in a microservices architecture can be developed, deployed, operated, and scaled without affecting the functioning of other services. Services do not need to share any of their code or implementation with other services. Any communication between individual components happens via well-defined APIs.

Specialized

Each service is designed for a set of capabilities and focuses on solving a specific problem. If developers contribute more code to a service over time and the service becomes complex, it can be broken into smaller services.

Microservices foster an organization of small, independent teams that take ownership of their services. Teams act within a small and well understood context, and are empowered to work more independently and more quickly. This shortens development cycle times. You benefit significantly from the aggregate throughput of the organization.
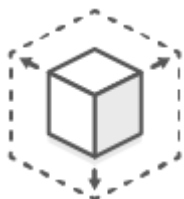
Flexible Scaling

Microservices allow each service to be independently scaled to meet demand for the application feature it supports. This enables teams to right-size infrastructure needs, accurately measure the cost of a feature, and maintain availability if a service experiences a spike in demand.

Easy Deployment

Microservices enable continuous integration and continuous delivery, making it easy to try out new ideas and to roll back if something doesn't work. The low cost of failure enables experimentation, makes it easier to update code, and accelerates time-to-market for new features.

Technological Freedom

Microservices architectures don't follow a "one size fits all" approach. Teams have the freedom to choose the best tool to solve their specific problems. As a consequence, teams building microservices can choose the best tool for each job.

Reusable Code

Dividing software into small, well-defined modules enables teams to use functions for multiple purposes. A service written for a certain function can be used as a building block for another feature. This allows an application to bootstrap off itself, as developers can create new capabilities without writing code from scratch.

The Most Complete Platform for Microservices

AWS has integrated building blocks that support any application architecture, regardless of scale, load, or complexity.

Compute

Processing power for microservices.

Containers

Amazon Elastic Container Service

A highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances.

Learn More »

Serverless

AWS Lambda

AWS Lambda lets you run code without provisioning or managing servers. Just upload your code and Lambda manages everything that is required to run and scale your code with high availability.

Learn more »

Using Amazon ECS, Coursera can now deploy software changes in minutes instead of hours in a resource-isolated environment.

Learn More »

Localytics used AWS Lambda to build microservices that allowed their development teams to build custom analytics without central support. Learn More »

Storage & Databases

Scalable, durable, and secure data storage.

Caching

Amazon ElastiCache

Amazon ElastiCache improves service performance by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

Learn More »

Object Storage

Amazon S3

Amazon S3 provides developers and IT teams highly reliable, secure, and scalable object storage for all of their data, large or small. Learn More »

NoSQL Databases

Amazon DynamoDB

A fully managed, fast, and flexible NoSQL database service for all applications that need consistent, single-digit, millisecond latency at any scale. Learn more »

Relational Databases

Amazon RDS

Easily setup, operate, and scale a relational database in the cloud. Choose from six familiar database engines, including Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB. Learn more »

Amazon Aurora

A relational database engine that combines the speed and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases. Deliver up to 5x the throughput of standard MySQL running on the same hardware. Learn more »

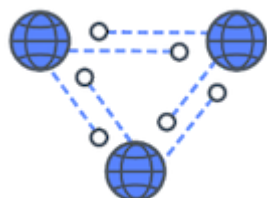remind | Remind reduced application response times 200% by building a PaaS for microservices on Amazon ECS.
Learn More »

Networking services with high-throughput and sub-millisecond latency.

Service Discovery
AWS Cloud Map

AWS Cloud Map is service discovery for all your cloud resources. With Cloud Map, you can define custom names for your application resources, and it maintains the updated location of these dynamically changing resources.

Learn more »

Service Mesh
AWS App Mesh

AWS App Mesh makes it easy to monitor and control microservices running on AWS. App Mesh standardizes how your microservices communicate, giving you end-to-end visibility and helping to ensure high-availability for your applications.

Learn more »

Elastic Load Balancing
Application Load Balancer

The Application Load Balancer load balances HTTP and HTTPS traffic at the application layer (level 7) providing advanced request routing that is targeted at the delivery of modern application architectures, including microservices and containers.

Learn More »

Network Load Balancer

The Network Load Balancer offers high performance load balancing that operates at the network connection layer (level 4) and allows you to route connections to microservices based on IP protocol data. The Network Load Balancer can handle millions of requests per second while maintaining ultra-low latencies.

Learn more »

API Proxy
Amazon API Gateway

Amazon API Gateway offers a comprehensive platform for API management. Amazon API Gateway allows you to process

DNS
Amazon Route 53

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service that effectively connects

Learn more »

Learn more »

Airtime provides its social experience to customers faster, more reliably, and with no lag time after redesigning its app as microservices running on AWS. Learn More »

Messaging

Publish and coordinate inter-process communications.

Message Publishing & Subscription

Amazon Simple Notification Service (Amazon SNS)

Amazon SNS is a fully managed pub/sub messaging service that makes it easy to decouple and scale microservices, distributed systems, and serverless applications.

Learn more »

Message Queueing

Amazon Simple Queue Service (Amazon SQS)

Amazon SQS is a fully managed message queuing service that makes it easy to decouple and scale microservices, distributed systems, and serverless applications.

Learn more »

Lyft uses AWS to move faster as a company and manage its exponential growth, leveraging AWS products to support more than 100 microservices that enhance every element of its customers' experience. Learn More »

Monitor service performance and resource utilization. Trace across complex architectures for troubleshooting and optimization.

API Monitoring
AWS CloudTrail

With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your infrastructure. CloudTrail event history simplifies security analysis, resource change tracking, and troubleshooting.

Learn more »

Application and Resource Monitoring
Amazon CloudWatch

Use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes across your running services and AWS resources.

Learn more »

Distributed Tracing
AWS X-Ray

Get an end-to-end view of requests as they travel through your application and see a map of your application's underlying components. As a set of microservices works together to handle a request, AWS X-Ray can provide a centralized view of logs, allowing you to monitor and troubleshoot complex interactions. Learn more »

By using microservices hosted on Amazon ECS, Shippable has been able to focus on delivering features to its customers and has sped feature deployment times from once a week to multiple per day. Learn More »

Manage the code lifecycle from commit to run.

Container Image Repository

Amazon Elastic Container Registry (Amazon ECR)

A fully managed Docker container registry that you can use to store, manage, and deploy Docker container images. Amazon ECR is integrated with Amazon Elastic Container Service (Amazon ECS), simplifying development to production workflow for containers. Learn more »

Continuous Delivery

AWS Developer Tools

AWS Developer Tools is a set of services that enable developers and IT operations professionals practicing DevOps to rapidly and safely deliver software. Together, these services help you securely store and version control your application's source code, and automatically build, test, and deploy your application to AWS or your on-premises environment. Learn more »

GILT

Gilt moved from an on-premises data center to AWS to leverage the speed and efficiency of a cloud-based microservices infrastructure.

Learn More »

Get Started

Start building microservices today with these resources.

AWS Serverless Multi-
Tier Architectures

Go Microservices

Node.js Microservices

Java Microservices

Workshops

Zombie Serverless
Microservices

Containerized
Microservices: Break the
Monolith!

Training: Running
Container-Enabled
Microservices on AWS

Microservices on
Amazon Elastic
Container Service

Building Scalable
Applications and
Microservices

Run Containerized
Microservices with
Amazon ECS and ALB

Using Amazon API
Gateway with
microservices deployed
on Amazon ECS

Service Discovery: An
Amazon ECS Reference
Architecture

Amazon S3

Amazon RDS

Amazon Route 53

Elastic Load Balancing

Amazon SNS

Amazon SQS

AWS CloudTrail

Amazon CloudWatch

AWS X-Ray

Amazon ECR

Developer Tools

Get Started with AWS

Learn how to start
using AWS in minutes

What's New with AWS

Learn about the
latest products,
services, and more

AWS Training & Certification

Free self-paced, online training wherever,
whenever you want. Boost your career with 600+
digital courses built by AWS Experts »

What Is AWS?

What Is Cloud Computing?

AWS Inclusion, Diversity & Equity

What Is DevOps?

What Is a Container?

What Is a Data Lake?

What is Generative AI?

AWS Cloud Security

What's New

Blogs

Press Releases

AWS Solutions Library

Architecture Center

Product and Technical FAQs

Analyst Reports

AWS Partners

.NET on AWS

Python on AWS

Java on AWS

PHP on AWS

JavaScript on AWS

Help

Contact Us

Get Expert Help

File a Support Ticket

AWS re:Post

Knowledge Center

AWS Support Overview

Legal

AWS Careers

Language

عربي |

Bahasa Indonesia |

Português |
Tiếng Việt |
Türkçe |
Русский |
ไทย |
日本語 |
한국어 |
中文 (简体) |
中文 (繁體)