

Service Name

AWS Cloud Map

Service Overview

AWS Cloud Map is a fully managed cloud resource discovery service that enables applications to automatically discover and connect to backend services and resources using logical names. It serves as a service registry that maps meaningful names to dynamic cloud resources like Amazon EC2 instances, Amazon ECS tasks, Amazon DynamoDB tables, Amazon SQS queues, and other AWS services. Applications can locate these resources through DNS queries or API calls, with Cloud Map only serving healthy resources. The service simplifies microservices architecture by providing a centralized way to track and connect to distributed application components as they dynamically change locations and endpoints.

Key Use Cases

- **Service discovery for microservices:** Enable microservices to automatically discover and connect to each other in dynamic containerized environments using Amazon ECS and Amazon EKS
- **Dynamic application resource mapping:** Map logical names to frequently changing cloud resources like load balancers, databases, and compute instances that scale up and down
- **Health-aware service routing:** Automatically route traffic only to healthy service instances by integrating with Amazon Route 53 health checks and custom health checkers
- **Cross-account service discovery:** Share service registries across multiple AWS accounts using AWS Resource Access Manager for centralized service management in multi-account architectures

Features

- **Namespace Management:** Create HTTP, private DNS, and public DNS namespaces to group and organize application services with different discovery methods
- **Service Registration:** Register and manage service instances with custom attributes for filtering and discovery using RegisterInstance API
- **Health Checking:** Automatic health monitoring using Amazon Route 53 health checks or custom health checkers to serve only healthy resources
- **Discovery Methods:** Support for both DNS-based discovery and API-based discovery using DiscoverInstances for flexible service location

- **Custom Attributes:** Assign custom metadata attributes to services and instances for advanced filtering and routing decisions
- **Service-Level Attributes:** Store and access service-level information like traffic weights and configuration directly at the service level
- **Cross-Account Sharing:** Share Cloud Map namespaces across AWS accounts using AWS Resource Access Manager for centralized service registries
- **Container Integration:** Deep integration with Amazon ECS and Amazon EKS for automatic service registration and deregistration

Value Proposition

AWS Cloud Map provides significant advantages over traditional service discovery solutions by offering a fully managed, highly available service that eliminates the operational overhead of maintaining service registries. Unlike static configuration files or custom-built discovery solutions, Cloud Map automatically handles service registration, health checking, and provides multiple discovery mechanisms (DNS and API) in a single service:

- The deep integration with AWS container services like ECS and EKS enables automatic service lifecycle management, reducing manual effort and potential errors
- Its support for cross-account sharing through AWS RAM makes it ideal for enterprise multi-account architectures, while the ability to use both DNS queries and API calls provides flexibility for different application patterns and requirements

Service Integration

AWS Cloud Map integrates seamlessly with core AWS services to provide comprehensive service discovery capabilities. It works natively with Amazon ECS and Amazon EKS for automatic container service registration and deregistration, and with AWS Fargate for serverless container deployments:

- The service leverages Amazon Route 53 for DNS-based discovery and health checking, automatically creating hosted zones for DNS namespaces
- Integration with AWS App Mesh enables advanced service mesh capabilities for microservices communication and traffic management
- Cloud Map also works with AWS Identity and Access Management (IAM) for fine-grained access control, AWS Resource Access Manager (RAM) for cross-account namespace sharing, and supports various AWS resources including Amazon EC2 instances, Amazon DynamoDB tables, Amazon S3 buckets, Amazon SQS queues, and Amazon API Gateway APIs as discoverable services

Onboarding and Offboarding

Getting started with AWS Cloud Map involves creating a namespace to group services, then creating services within that namespace, and finally registering service instances. Users can access Cloud Map through the AWS Management Console, AWS CLI, SDKs, or REST API:

- The process begins by choosing the appropriate namespace type (HTTP for API-only discovery, private DNS for VPC-based DNS queries, or public DNS for internet-accessible services), followed by creating service templates that define health check configurations and DNS settings
- Applications then register instances using the RegisterInstance API and discover services using either DNS queries or the DiscoverInstances API
- AWS provides comprehensive tutorials, code examples, and CLI-based getting started guides to help users implement service discovery patterns quickly and effectively

Getting Started Guide: <https://docs.aws.amazon.com/cloud-map/latest/dg/tutorials.html>

Data Removal

To offboard from AWS Cloud Map, customers must deregister all service instances using the DeregisterInstance API, delete all services using DeleteService, and finally delete namespaces using DeleteNamespace. When deleting DNS-based namespaces, the corresponding Amazon Route 53 hosted zones are automatically removed. All registered service data, custom attributes, and health check configurations are permanently deleted. The deregistration process should be performed in reverse order of creation to avoid dependencies. AWS provides CLI commands and API operations to automate the cleanup process for bulk operations.

Backup/Restore and Disaster Recovery

AWS Cloud Map does not provide traditional backup and restore capabilities as it serves as a dynamic service registry rather than a data store. The service configuration and registered instances are designed to be ephemeral and recreated automatically by applications. However, Cloud Map itself is a highly available, fully managed service that operates across multiple Availability Zones. For disaster recovery, customers should implement automated service registration as part of their application startup process, ensuring services can re-register themselves when deployed in recovery regions. Custom scripts or infrastructure-as-code templates can be used to recreate namespace and service configurations.

Backup Capabilities

AWS Cloud Map does not have traditional backup capabilities or integration with AWS Backup since it is a dynamic service discovery registry. The service data (namespaces,

services, and instances) is designed to be transient and automatically managed by applications:

- Instead of backups, customers should implement infrastructure-as-code using AWS CloudFormation or other tools to recreate Cloud Map configurations
- Service instance data is expected to be re-registered automatically by applications during startup, making traditional backup mechanisms unnecessary for this service type

Disaster Recovery

AWS Cloud Map does not directly integrate with AWS Elastic Disaster Recovery as it is not a traditional data service requiring replication. However, the service itself provides high availability by running across multiple Availability Zones within each region:

- For disaster recovery scenarios, customers should deploy Cloud Map resources in multiple regions and implement application logic to re-register services automatically in the recovery region
- The service configurations can be replicated across regions using infrastructure-as-code tools, while service instances should be dynamically registered by applications as part of the disaster recovery process

Recovery Objectives

AWS Cloud Map helps customers meet RPO and RTO objectives by providing rapid service discovery and registration capabilities during disaster recovery scenarios. Since service instance data is ephemeral and designed for dynamic registration, the RPO is effectively zero as no historical data needs to be recovered. The RTO is primarily determined by how quickly applications can re-register services in the recovery region and discover each other, typically within minutes. The service's high availability architecture and fast DNS propagation help minimize discovery-related delays during failover events, supporting aggressive RTO targets for microservices architectures.

Service Constraints

Service Quotas: <https://docs.aws.amazon.com/cloud-map/latest/dg/cloud-map-limits.html>

FAQ: <https://aws.amazon.com/cloud-map/faqs/>

Technical Requirements

Service Documentation: <https://docs.aws.amazon.com/cloud-map/latest/dg/what-is-cloud-map.html>

User Guide: <https://docs.aws.amazon.com/cloud-map/latest/dg/tutorials.html>

API Reference: <https://docs.aws.amazon.com/cloud-map/latest/api/Welcome.html>

Pricing Overview

Please see the AWS UK G Cloud 15 Pricing Document accompanying this service in the Digital Marketplace.

Public Pricing: <https://aws.amazon.com/cloud-map/pricing/>

Cost Optimization

AWS Cloud Map offers several cost optimization opportunities through its flexible discovery mechanisms and efficient resource management. Customers can reduce costs by choosing HTTP namespaces for API-only discovery, avoiding DNS-related charges when DNS discovery is not needed:

- Using appropriate DNS TTL values balances discovery freshness with query costs
- The service charges only for registered resources and API calls, so deregistering unused instances immediately reduces costs
- Implementing efficient service discovery patterns with caching and connection pooling minimizes API call frequency
- Cross-account namespace sharing eliminates duplicate registry costs in multi-account architectures, while service-level attributes reduce the need for additional discovery calls by providing metadata in single API responses

Savings Considerations

Cost savings with AWS Cloud Map come from eliminating the need to build and maintain custom service discovery infrastructure, reducing operational overhead and development time. The pay-per-use model means customers only pay for active registered resources and actual discovery calls, avoiding fixed infrastructure costs:

- Compared to running self-managed service registries on EC2 instances, Cloud Map eliminates compute, storage, and maintenance costs while providing higher availability
- The integration with container services reduces manual service management overhead
- DNS-based discovery can be more cost-effective than API calls for high-frequency lookups, while the free tier provides cost savings for smaller workloads with up to certain usage limits