What is Amazon EKS?

Amazon Elastic Kubernetes Service (Amazon EKS) is 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. Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications. Amazon EKS:

* Runs and scales the Kubernetes control plane across multiple AWS Availability Zones to ensure high availability.
* Automatically scales control plane instances based on load, detects and replaces unhealthy control plane instances, and it provides automated version updates and patching for them.
* Is integrated with many AWS services to provide scalability and security for your applications, including the following capabilities:
  + Amazon ECR for container images
  + Elastic Load Balancing for load distribution
  + IAM for authentication
  + Amazon VPC for isolation
* Runs up-to-date versions of the open-source Kubernetes software, so you can use all of the existing plugins and tooling from the Kubernetes community. Applications that are running on Amazon EKS are fully compatible with applications running on any standard Kubernetes environment, no matter whether they're running in on-premises data centers or public clouds. This means that you can easily migrate any standard Kubernetes application to Amazon EKS without any code modification.

## Amazon EKS control plane architecture

Amazon EKS runs a single tenant Kubernetes control plane for each cluster. The control plane infrastructure isn't shared across clusters or AWS accounts. The control plane consists of at least two API server instances and three etcd instances that run across three Availability Zones within an AWS Region. Amazon EKS:

* Actively monitors the load on control plane instances and automatically scales them to ensure high performance.
* Automatically detects and replaces unhealthy control plane instances, restarting them across the Availability Zones within the AWS Region as needed.
* Leverages the architecture of AWS Regions in order to maintain high availability. Because of this, Amazon EKS is able to offer an [SLA for API server endpoint availability](http://aws.amazon.com/eks/sla).

Amazon EKS uses Amazon VPC network policies to restrict traffic between control plane components to within a single cluster. Control plane components for a cluster can't view or receive communication from other clusters or other AWS accounts, except as authorized with Kubernetes RBAC policies. This secure and highly available configuration makes Amazon EKS reliable and recommended for production workloads.

# What is Amazon Elastic Container Service?

Amazon Elastic Container Service (Amazon ECS) is a highly scalable and fast container management service. You can use it to run, stop, and manage containers on a cluster. With Amazon ECS, your containers are defined in a task definition that you use to run an individual task or task within a service. In this context, a service is a configuration that you can use to run and maintain a specified number of tasks simultaneously in a cluster. You can run your tasks and services on a serverless infrastructure that's managed by AWS Fargate. Alternatively, for more control over your infrastructure, you can run your tasks and services on a cluster of Amazon EC2 instances that you manage.

Amazon ECS provides the following features:

* A serverless option with AWS Fargate. With AWS Fargate, you don't need to manage servers, handle capacity planning, or isolate container workloads for security. Fargate handles the infrastructure management aspects of your workload for you. You can schedule the placement of your containers across your cluster based on your resource needs, isolation policies, and availability requirements.
* Integration with AWS Identity and Access Management (IAM). You can assign granular permissions for each of your containers. This allows for a high level of isolation when building your applications. In other words, you can launch your containers with the security and compliance levels that you've come to expect from AWS.
* AWS managed container orchestration. As a fully managed service, Amazon ECS comes with AWS configuration and operational best practices built-in. This also means that you don't need to manage control plane, nodes, or add-ons. It's integrated with both Alexa Web Information Service and third-party tools, such as Amazon Elastic Container Registry and Docker. This integration makes it easier for teams to focus on building the applications, not the environment.
* Continuous integration and continuous deployment (CI/CD). This is a common process for microservice architectures that are based on Docker containers. You can create a CI/CD pipeline that takes the following actions:
  + Monitors changes to a source code repository
  + Builds a new Docker image from that source
  + Pushes the image to an image repository such as Amazon ECR or Docker Hub
  + Updates your Amazon ECS services to use the new image in your application
* Support for service discovery. This is a key component of most distributed systems and service-oriented architectures. With service discovery, your microservice components are automatically discovered as they're created and terminated on a given infrastructure.
* Support for sending your container instance log information to CloudWatch Logs. After you send this information to Amazon CloudWatch, you can view the logs from your container instances in one convenient location. This prevents your container logs from taking up disk space on your container instances.

The AWS container services team maintains a public roadmap on GitHub. The roadmap contains information about what the teams are working on and enables AWS customers to provide direct feedback. For more information, see [AWS Containers Roadmap](https://github.com/aws/containers-roadmap) on the GitHub website.

## Launch types

There are two models that you can use to run your containers:

* Fargate launch type - This is a serverless pay-as-you-go option. You can run containers without needing to manage your infrastructure.
* EC2 launch type - Configure and deploy EC2 instances in your cluster to run your containers.

The Fargate launch type is suitable for the following workloads:

* Large workloads that need to be optimized for low overhead
* Small workloads that have occasional burst
* Tiny workloads
* Batch workloads

The EC2 launch type is suitable for the following workloads:

* Workloads that require consistently high CPU core and memory usage
* Large workloads that need to be optimized for price
* Your applications need to access persistent storage
* You must directly manage your infrastructure

## Access Amazon ECS

You can create, access, and manage your Amazon ECS resources using any of the following interfaces:

* **AWS Management Console** — Provides a web interface that you can use to access your Amazon ECS resources.
* **AWS Command Line Interface (AWS CLI)** — Provides commands for a broad set of AWS services, including Amazon ECS. It's supported on Windows, Mac, and Linux. For more information, see [AWS Command Line Interface](https://aws.amazon.com/cli/).
* **AWS SDKs** — Provides language-specific APIs and takes care of many of the connection details. These include calculating signatures, handling request retries, and error handling. For more information, see [AWS SDKs](http://aws.amazon.com/tools/#SDKs).
* **AWS Copilot** — Provides an open-source tool for developers to build, release, and operate production ready containerized applications on Amazon ECS. For more information, see [AWS Copilot](https://github.com/aws/copilot-cli) on the GitHub website.
* **Amazon ECS CLI** — Provides a command line interface for you to run your applications on Amazon ECS and AWS Fargate using the Docker Compose file format. You can quickly provision resources, push and pull images using Amazon Elastic Container Registry, and monitor running applications on Amazon ECS or Fargate. You can also test containers that are running locally along with containers in the Cloud within the CLI. For more information, see [Amazon ECS CLI](https://github.com/aws/amazon-ecs-cli) on the GitHub website.
* **AWS CDK** — Provides an open-source software development framework that you can use to model and provision your cloud application resources using familiar programming languages. The AWS CDK provisions your resources in a safe, repeatable manner through AWS CloudFormation. For more information, see [Getting started with Amazon ECS using the AWS CDK](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/tutorial-ecs-web-server-cdk.html).

## Pricing

Amazon ECS pricing is dependent on whether you use AWS Fargate or Amazon EC2 infrastructure to host your containerized workloads. When using Amazon ECS on AWS Outposts, the pricing follows the same model that's used when you use Amazon EC2 directly. For more information, see [Amazon ECS Pricing](https://aws.amazon.com/ecs/pricing).

Amazon ECS and Fargate also offer Savings Plans that provide significant savings based on your AWS usage. For more information, see the [*Savings Plans User Guide*](https://docs.aws.amazon.com/savingsplans/latest/userguide/).

To view your bill, go to the **Billing and Cost Management Dashboard** in the [AWS Billing and Cost Management console](https://console.aws.amazon.com/billing/). Your bill contains links to usage reports that provide additional details about your bill. To learn more about AWS account billing, see [AWS Account Billing](https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/).

If you have questions concerning AWS billing, accounts, and events, [contact AWS Support](https://aws.amazon.com/contact-us/).

Trusted Advisor is a service that you can use to help optimize the costs, security, and performance of your AWS environment. For more information about Trusted Advisor, see [AWS Trusted Advisor](https://aws.amazon.com/premiumsupport/trustedadvisor/).

## How does Amazon EKS work?


    How Amazon EKS works
   

Getting started with Amazon EKS is easy:

1. Create an Amazon EKS cluster in the AWS Management Console or with the AWS CLI or one of the AWS SDKs.
2. Launch managed or self-managed Amazon EC2 nodes, or deploy your workloads to AWS Fargate.
3. When your cluster is ready, you can configure your favorite Kubernetes tools, such as kubectl, to communicate with your cluster.
4. Deploy and manage workloads on your Amazon EKS cluster the same way that you would with any other Kubernetes environment. You can also view information about your workloads using the AWS Management Console.

To create your first cluster and its associated resources, see [Getting started with Amazon EKS](https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html). To learn about other Kubernetes deployment options, see [Deployment options](https://docs.aws.amazon.com/eks/latest/userguide/eks-deployment-options.html).

## Pricing

An Amazon EKS cluster consists of a control plane and the Amazon EC2 or AWS Fargate compute that you run pods on. For more information about pricing for the control plane, see [Amazon EKS pricing](http://aws.amazon.com/eks/pricing). Both Amazon EC2 and Fargate provide:

* **On-Demand Instances** – Pay for the instances that you use by the second, with no long-term commitments or upfront payments. For more information, see [Amazon EC2 On-Demand Pricing](http://aws.amazon.com/ec2/pricing/on-demand/) and [AWS Fargate Pricing](http://aws.amazon.com/fargate/pricing/).
* **Savings Plans** – You can reduce your costs by making a commitment to a consistent amount of usage, in USD per hour, for a term of 1 or 3 years. For more information, see [Pricing with Savings Plans](http://aws.amazon.com/savingsplans/pricing/).

### What is Amazon RDS?

Amazon Relational Database Service (RDS) is a managed SQL database service provided by Amazon Web Services (AWS). Amazon RDS supports an array of database engines to store and organize data. It also helps with [relational database](https://www.techtarget.com/searchdatamanagement/definition/relational-database) management tasks, such as data migration, backup, recovery and patching.

Amazon RDS facilitates the deployment and maintenance of relational databases in the cloud. A cloud administrator uses Amazon RDS to set up, operate, manage and scale a relational instance of a [cloud database](https://www.techtarget.com/searchcloudcomputing/definition/cloud-database). Amazon RDS is not itself a database; it is a service used to manage relational databases.

### How does Amazon RDS work?

Databases are used to store large quantities of data that applications can draw on to help them perform various functions. A relational database uses tables to store data. It is called relational because it organizes data points with defined relationships.

Administrators control Amazon RDS with the [AWS Management Console](https://www.techtarget.com/searchaws/definition/AWS-Management-Console), Amazon RDS [API](https://www.techtarget.com/searchapparchitecture/definition/application-program-interface-API) calls or the [AWS Command Line Interface](https://www.techtarget.com/searchaws/definition/AWS-Command-Line-Interface). They use these interfaces to deploy database instances to which users can apply specific settings.

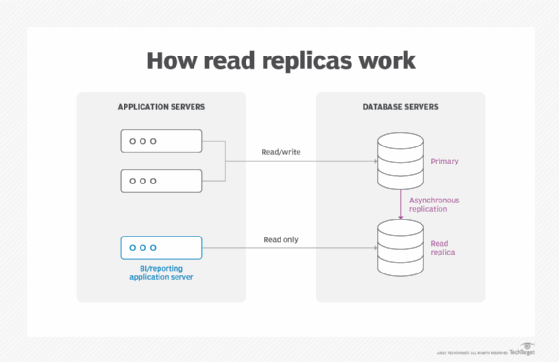
Amazon provides several instance types with different combinations of resources, such as CPU, memory, storage options and networking capacity. Each type comes in a variety of sizes to suit the needs of different workloads.

RDS users can use [AWS Identity and Access Management](https://www.techtarget.com/searchaws/definition/Amazon-Web-Services-AWS-Identity-and-Access-Management-IAM) to define and set permissions for who can access an RDS database.

### Amazon RDS features

Amazon RDS features include the following:

**Replication.**RDS uses the Replication feature to create read replicas. These are read-only copies of database instances that applications use without altering the original production database. Administrators can also enable automatic [failover](https://www.techtarget.com/searchstorage/definition/failover) across multiple availability zones through RDS Multi-AZ deployment and with synchronous data replication.



TECHTARGET

See how read replicas create read-only snapshots of database instances to help with load balancing.

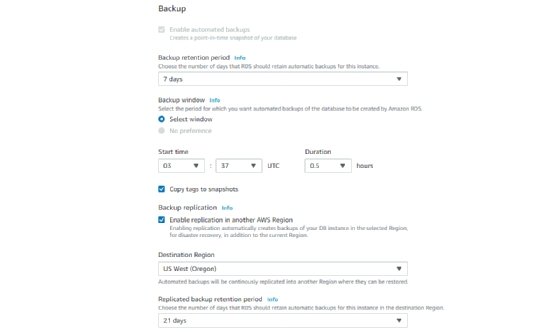
**Storage.**RDS provides three types of storage:

* **General-purpose solid-state drive (**[**SSD**](https://www.techtarget.com/searchstorage/definition/SSD-solid-state-drive)**).** Amazon recommends this storage as the default choice.
* **Provisioned input-output operations per second (**[**IOPS**](https://www.techtarget.com/searchstorage/definition/IOPS-input-output-operations-per-second)**).** SSD storage for I/O-intensive workloads.
* **Magnetic.** A lower cost option.

**Monitoring.**The Amazon CloudWatch service enables managed monitoring. It lets users view capacity and I/O metrics.

**Patching.**RDS provides patches for whichever database engine the user chooses.

**Backups.** Another feature is failure detection and recovery. RDS provides managed instance backups with transaction logs to enable point-in-time recovery. Users pick a retention period and restore databases to any time during that period. They also can manually take snapshots of instances that remain until they are manually deleted.

RDS lets users specify the time and duration of the backup processes. They also can choose how long to retain backups and snapshots.

**Incremental billing.** Users pay a monthly fee for the instances they launch.

**Encryption.**RDS uses public key encryption to secure automated backups, read replicas, data snapshots and other data stored at rest.

### What are the benefits and drawbacks of Amazon RDS?

There are several pros and cons to using Amazon RDS.

#### Benefits

The main benefit of Amazon RDS is that it helps organizations deal with the complexity of managing large relational databases. Other benefits include the following:

* **Ease of use.**Admins don't need to learn specific database management tools. They also can manage multiple database instances using the management console. RDS is compatible with database engines that users may already be familiar with, such as [MySQL](https://www.techtarget.com/searchoracle/definition/MySQL) and [Oracle](https://www.techtarget.com/searchoracle/definition/Oracle) And it automates manual backup and recovery processes.
* **Cost-effectiveness.**According to AWS, customers only pay for what they use. Also, the time spent maintaining instances is reduced, because maintenance tasks, such as backups and patching, are automated.
* The use of read replicas routes read-heavy traffic away from the main database instance, reducing the workload on that one instance.
* RDS splits up compute and storage so admins can [scale](https://www.techtarget.com/searchdatacenter/definition/scalability) them independently.

#### Drawbacks

Some downsides of using Amazon RDS include the following:

* **Lack of root access.** Because it is a managed service, users do not have root access to the server running RDS. RDS restricts access for certain procedures to those with advanced privileges.
* **Downtime.** Systems must go offline for some patching and scaling procedures. The timing on these processes varies. With scaling, compute resources need a few minutes downtime on average.

### Amazon RDS database instances

A database administrator can create, configure, manage and delete an Amazon RDS instance, along with the resources it uses. An Amazon RDS instance is a cloud database environment. Admins can also spin up many databases or [schemas](https://searchsqlserver.techtarget.com/definition/schema); how many depends on the database used.

Amazon RDS limits each customer to a total of 40 database instances per account. AWS imposes further limitations for Oracle and SQL Server instances. With those database instances, a user generally can only have up to 10.

### Amazon RDS database engines

An AWS customer can spin up six types of database engines within Amazon RDS:

1. **Amazon Aurora**is a proprietary AWS relational database engine. [Amazon Aurora](https://www.techtarget.com/searchaws/definition/Amazon-Aurora) is compatible with MySQL and PostgreSQL.
2. **RDS for MariaDB** is compatible with [MariaDB](https://www.techtarget.com/searchdatamanagement/definition/MariaDB), an open source relational database management system ([RDBMS](https://www.techtarget.com/searchdatamanagement/definition/RDBMS-relational-database-management-system)) that's an offshoot of MySQL.
3. **RDS for MySQL**is compatible with the MySQL open source RDBMS.
4. **RDS for Oracle Database** is compatible with several editions of Oracle Database, including bring-your-own-license and license-included versions.
5. **RDS for PostgreSQL** is compatible with [PostgreSQL](https://www.techtarget.com/whatis/definition/PostgreSQL) open source object-RDBMS.
6. **RDS for SQL Server** is compatible with [Microsoft SQL Server](https://searchsqlserver.techtarget.com/definition/SQL-Server), an RDBMS.

Amazon RDS adds support for major and minor versions of database engines over time. It is designed to allow admins to specify an engine version when they create a database instance. In most cases, Amazon RDS can support developer code, applications and tools that are already in use with existing databases.

AWS provides other database services, including the following:

* [Amazon DynamoDB](https://www.techtarget.com/searchaws/definition/Amazon-Dynamo-Database-DDB) key-value and document database for NoSQL databases;
* [Amazon Neptune](https://www.techtarget.com/searchaws/definition/Amazon-Neptune) for graph databases; and
* [AWS Database Migration Service](https://www.techtarget.com/searchaws/definition/AWS-Database-Migration-Service-AWS-DMS) to ease database transfers and transformations.

### Amazon RDS use cases

Amazon RDS' scalability, security and availability make it useful for a variety of applications. Some possible uses include the following:

**Online retailing.** These applications manage complex databases that track inventories, transactions and pricing.

**Mobile and online gaming.**RDS supports developers that need to continuously update these applications and users who need high availability.

**Travel applications.**Applications like Airbnb take advantage of RDS' ability to simplify time-consuming database administration tasks and automate database replication. Mobile apps like Airbnb also take advantage of RDS' scalable storage capability.

**Streaming applications.**Applications like Netflix take advantage of RDS' storage scalability as well, and availability of Amazon RDS, which allows them to handle high demand daily.

**Finance applications.**These applications, like other mobile applications, can use RDS to simplify administrative database tasks and save time and money.

Business-to-business reporting company Enlyft said 6,096 companies were using Amazon RDS in 2021, including The American Red Cross, Penguin Random House and Zendesk. Amazon also reported in 2021 that Airbnb, Intuit and the U.S. Department of Veterans Affairs are among the organizations that use RDS to support their data workloads.

### Amazon Aurora and Amazon Aurora Serverless vs. Amazon RDS

Amazon Aurora is a database engine created by Amazon. RDS is a service used to manage database engines and instances, including Amazon Aurora databases.

Amazon Aurora Serverless can also be used to [manage instances](https://aws.amazon.com/rds/aurora/serverless/) of Amazon Aurora. Its automation features relieve developers from having to manually launch servers and manage database capacity.

With RDS, servers must be manually scaled, which can result in significant downtime. Aurora Serverless' automatic scaling capability enables faster deployment with little to no downtime.

One downside of Aurora Serverless is it only works with Amazon Aurora, MySQL and PostgreSQL. RDS is compatible with six database engines.

Organizations should [assess the two database management options](https://www.techtarget.com/searchcloudcomputing/answer/When-should-I-use-Amazon-RDS-vs-Aurora-Serverless). Both have advantages and limitations, depending on the type of application they serve. Aurora Serverless is considered suitable for applications with unpredictable and steep spikes in usage, which need more efficient capacity adjustments. Amazon RDS is suitable for more predictable applications because capacity adjustments take more time in RDS than in Aurora Serverless.

### The takeaway

Amazon RDS helps organizations handle relational database management tasks such as migration, backup, recovery and patching. Some of the main features of Amazon RDS are replication, high performance storage and failure detection.

One of the biggest advantages of Amazon RDS is its ease of use. It lets administrators manage multiple database instances without having to learn other database management tools.

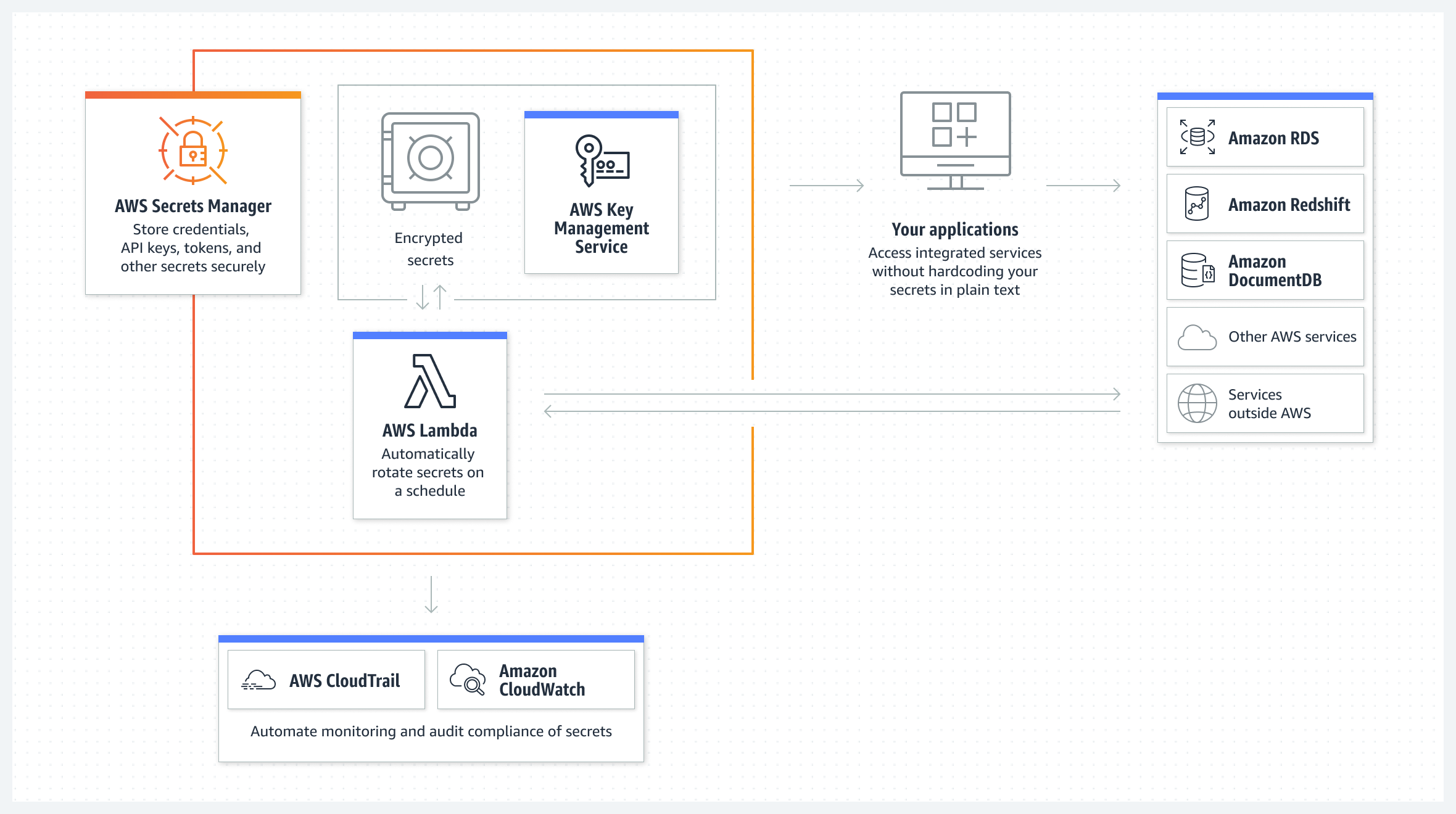
These features enable RDS to help organizations cut costs that come from time-consuming database administration tasks and manage the hidden costs that come with using high-performance storage in AWS. Learn more about [ways to manage AWS costs](https://www.techtarget.com/searchcloudcomputing/tip/Get-a-handle-on-AWS-costs-with-these-tips-and-tools).

Q.what is cloudwatch in AWS?

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), IT managers, and product owners. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, and optimize resource utilization. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events. You get a unified view of operational health and gain complete visibility of your AWS resources, applications, and services running on AWS and on-premises. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.

**Q-What is AWS Secrets Manager?**

AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. Users and applications retrieve secrets with a call to Secrets Manager APIs, eliminating the need to hardcode sensitive information in plain text. Secrets Manager offers secret rotation with built-in integration for Amazon Relational Database Service (Amazon RDS), Amazon Redshift, and Amazon DocumentDB. Also, the service is extensible to other types of secrets, including API keys and OAuth tokens. In addition, Secrets Manager enables you to control access to secrets using fine-grained permissions and audit secret rotation centrally for resources in the AWS Cloud, third-party services, and on-premises. (<https://aws.amazon.com/secrets-manager/>)



**Q-What is Elastic Search in AWS?**

Elasticsearch is a distributed search and analytics engine built on Apache Lucene. Since its release in 2010, Elasticsearch has quickly become the most popular search engine and is commonly used for log analytics, full-text search, security intelligence, business analytics, and operational intelligence use cases.

## How does Elasticsearch work?

You can send data in the form of JSON documents to Elasticsearch using the API or ingestion tools such as [Logstash](https://aws.amazon.com/opensearch-service/the-elk-stack/logstash/" \t "_blank) and [Amazon Kinesis Firehose.](https://aws.amazon.com/kinesis/data-firehose/) Elasticsearch automatically stores the original document and adds a searchable reference to the document in the cluster’s index. You can then search and retrieve the document using the Elasticsearch API. You can also use [Kibana](https://aws.amazon.com/opensearch-service/the-elk-stack/kibana/" \t "_self), a visualization tool, with Elasticsearch to visualize your data and build interactive dashboards.

You can run Apache 2.0 licensed Elasticsearch versions (up until version 7.10.2 & Kibana 7.10.2) on-premises, on Amazon EC2, or on Amazon OpenSearch Service. With on-premises or Amazon EC2 deployments, you are responsible for installing Elasticsearch and other necessary software, provisioning infrastructure, and managing the cluster. Amazon OpenSearch Service, on the other hand, is a fully managed service, so you don’t have to worry about time-consuming cluster management tasks such as hardware provisioning, software patching, failure recovery, backups, and monitoring.

## Elasticsearch benefits

### Fast time-to-value

Elasticsearch offers simple REST based APIs, a simple HTTP interface, and uses schema-free JSON documents, making it easy to get started and quickly build applications for a variety of use-cases.

### High performance

The distributed nature of Elasticsearch enables it to process large volumes of data in parallel, quickly finding the best matches for your queries.

### Complimentary tooling and plugins

Elasticsearch comes integrated with Kibana, a popular visualization and reporting tool. It also offers integration with Beats and Logstash, while enable you to easily transform source data and load it into your Elasticsearch cluster. You can also use a number of open-source Elasticsearch plugins such as language analyzers and suggesters to add rich functionality to your applications.

### Near real-time operations

Elasticsearch operations such as reading or writing data usually take less than a second to complete. This lets you use Elasticsearch for near real-time use cases such as application monitoring and anomaly detection.

### Easy application development

Elasticsearch provides support for various languages including Java, Python, PHP, JavaScript, Node.js, Ruby, and many more.

**Q-What is** Amazon Kinesis?

Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information. Amazon Kinesis offers key capabilities to cost-effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application. With Amazon Kinesis, you can ingest real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications. Amazon Kinesis enables you to process and analyze data as it arrives and respond instantly instead of having to wait until all your data is collected before the processing can begin.

# What is Amazon VPC?

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

## Features

The following features help you configure a VPC to provide the connectivity that your applications need:

**Virtual private clouds (VPC)**

A [VPC](https://docs.aws.amazon.com/vpc/latest/userguide/configure-your-vpc.html) is a virtual network that closely resembles a traditional network that you'd operate in your own data center. After you create a VPC, you can add subnets.

**Subnets**

A [subnet](https://docs.aws.amazon.com/vpc/latest/userguide/configure-subnets.html) is a range of IP addresses in your VPC. A subnet must reside in a single Availability Zone. After you add subnets, you can deploy AWS resources in your VPC.

**IP addressing**

You can assign IPv4 addresses and IPv6 addresses to your VPCs and subnets. You can also bring your public IPv4 and IPv6 GUA addresses to AWS and allocate them to resources in your VPC, such as EC2 instances, NAT gateways, and Network Load Balancers.

**Routing**

Use [route tables](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html) to determine where network traffic from your subnet or gateway is directed.

**Gateways and endpoints**

A [gateway](https://docs.aws.amazon.com/vpc/latest/userguide/extend-intro.html) connects your VPC to another network. For example, use an [internet gateway](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html) to connect your VPC to the internet. Use a [VPC endpoint](https://docs.aws.amazon.com/vpc/latest/privatelink/privatelink-access-aws-services.html) to connect to AWS services privately, without the use of an internet gateway or NAT device.

**Peering connections**

Use a [VPC peering connection](https://docs.aws.amazon.com/vpc/latest/peering/) to route traffic between the resources in two VPCs.

**Traffic Mirroring**

[Copy network traffic](https://docs.aws.amazon.com/vpc/latest/mirroring/) from network interfaces and send it to security and monitoring appliances for deep packet inspection.

**Transit gateways**

Use a [transit gateway](https://docs.aws.amazon.com/vpc/latest/userguide/extend-tgw.html), which acts as a central hub, to route traffic between your VPCs, VPN connections, and AWS Direct Connect connections.

**VPC Flow Logs**

A [flow log](https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html) captures information about the IP traffic going to and from network interfaces in your VPC.

**VPN connections**

Connect your VPCs to your on-premises networks using [AWS Virtual Private Network (AWS VPN)](https://docs.aws.amazon.com/vpc/latest/userguide/vpn-connections.html).

## Getting started with Amazon VPC

Your AWS account includes a [default VPC](https://docs.aws.amazon.com/vpc/latest/userguide/default-vpc.html) in each AWS Region. Your default VPCs are configured such that you can immediately start launching and connecting to EC2 instances. For more information, see [Get started with Amazon VPC](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-getting-started.html).

You can choose to create additional VPCs with the subnets, IP addresses, gateways and routing that you need. For more information, see [Create a VPC](https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#Create-VPC).

## Working with Amazon VPC

You can create and manage your VPCs using any of the following interfaces:

* **AWS Management Console** — Provides a web interface that you can use to access your VPCs.
* **AWS Command Line Interface (AWS CLI)** — Provides commands for a broad set of AWS services, including Amazon VPC, and is supported on Windows, Mac, and Linux. For more information, see [AWS Command Line Interface](https://aws.amazon.com/cli/).
* **AWS SDKs** — Provides language-specific APIs and takes care of many of the connection details, such as calculating signatures, handling request retries, and error handling. For more information, see [AWS SDKs](http://aws.amazon.com/tools/#SDKs).
* **Query API** — Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access Amazon VPC, but it requires that your application handle low-level details such as generating the hash to sign the request, and error handling. For more information, see [Amazon VPC actions](https://docs.aws.amazon.com/AWSEC2/latest/APIReference/OperationList-query-vpc.html) in the Amazon EC2 API Reference.

## Pricing for Amazon VPC

There's no additional charge for using a VPC. There are charges for some VPC components, such as NAT gateways, Reachability Analyzer, and traffic mirroring. For more information, see [Amazon VPC Pricing](http://aws.amazon.com/vpc/pricing/).

# What is Amazon Route 53?

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking.

If you choose to use Route 53 for all three functions, be sure to follow the order below:

**1. Register domain names**

Your website needs a name, such as example.com. Route 53 lets you register a name for your website or web application, known as a *domain name*.

* For an overview, see [How domain registration works](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/welcome-domain-registration.html).
* For a procedure, see [Registering a new domain](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-register.html).
* For a tutorial that takes you through registering a domain and creating a simple website in an Amazon S3 bucket, see [Getting started with Amazon Route 53](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/getting-started.html).

**2. Route internet traffic to the resources for your domain**

When a user opens a web browser and enters your domain name (example.com) or subdomain name (acme.example.com) in the address bar, Route 53 helps connect the browser with your website or web application.

* For an overview, see [How internet traffic is routed to your website or web application](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/welcome-dns-service.html).
* For procedures, see [Configuring Amazon Route 53 as your DNS service](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-configuring.html).
* For a procedure on how to route email to Amazon WorkMail, see [Routing traffic to Amazon WorkMail](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-to-workmail.html).

**3. Check the health of your resources**

Route 53 sends automated requests over the internet to a resource, such as a web server, to verify that it's reachable, available, and functional. You also can choose to receive notifications when a resource becomes unavailable and choose to route internet traffic away from unhealthy resources.