AWS global infrastructure

AWS has done a similar thing with how the AWS global infrastructure is set up. It's not good enough to have one giant data center where all of the resources are. If something were to happen to that data center, like a power outage or a natural disaster, everyone's applications would go down all at once. You need high availability and fault tolerance.

Turns out, it's not even good enough to have two giant data centers. Instead, AWS operates in all sorts of different areas around the world called Regions. We're going to talk about this in depth in upcoming videos. In the meantime, I'm gonna sit here and relax, because I know my business is highly available regardless of any parades blocking the street.

Each Region can be connected to each other Region through a high speed fiber network, controlled by AWS.

Each Region is isolated from every other Region in the sense that absolutely no data goes in or out of your environment in that Region without you explicitly granting permission for that data to be moved. This is a critical security conversation to have.

For example:- Any data stored in the Frankfurt Region never leaves the Frankfurt Region, or data in the London region never leaves London, or Sydney never leaves Sydney, unless you explicitly, with the right credentials and permissions, request the data be exported.

Regional data sovereignty is part of the critical design of AWS Regions.

So four key factors to choose a Region: Compliance, proximity, feature availability, and pricing.

Number one, compliance. Before any of the other factors, you must first look at your compliance requirements. Do you have a requirement your data must live in UK boundaries? Then you should choose the London Region, full stop.

Number two, proximity. How close you are to your customer base is a major factor because speed of light, still the law of the universe. If most of your customers live in Singapore, consider running out of the Singapore Region. Now you certainly can run out of Virginia, but the time it takes for the information to be sent, or latency, between the US and Singapore is always going to be a factor. Now we may be developing quantum computing, but quantum networking, still some ways away. The time it takes light to travel around the world is always a consideration. Locating close to your customer base, usually the right call.

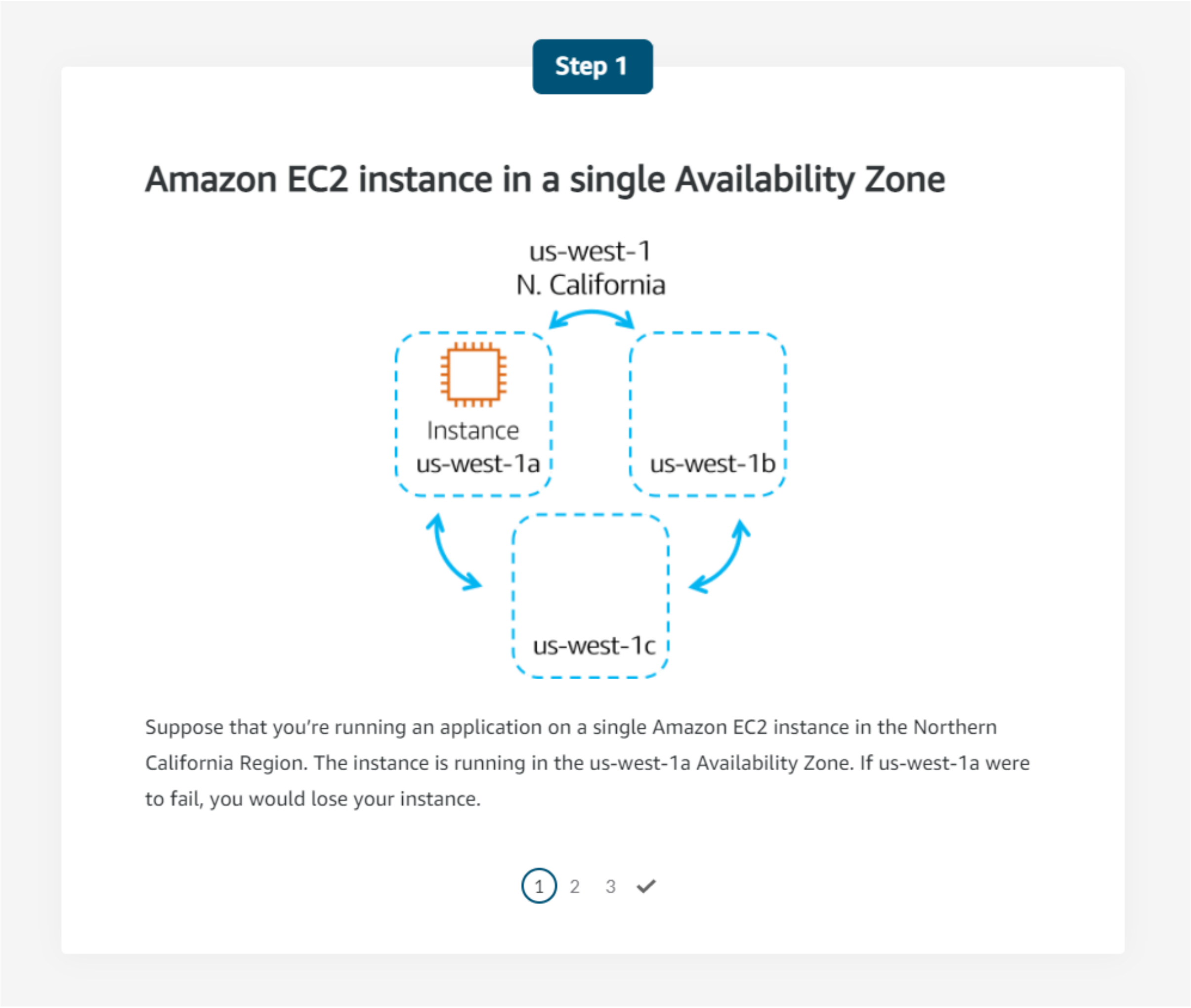
Number three, feature availability. Sometimes the closest Region may not have all the AWS features you want. So let's say your developers wanted to play with Amazon Braket, our new quantum computing platform. Well then, they have to run in the Regions that already have the hardware installed. Eventually, can we expect it to be in every Region? Well, yeah, that's a good expectation, but if you wanna use it today, then that might be your deciding factor.

Number four, pricing. Even when the hardware is equal one Region to the next, some locations are just more expensive to operate in, for example, Brazil. Now Brazil's tax structure is such that it costs AWS significantly more to operate the exact same services there than in many other countries. The exact same workload in Sao Paulo might be, let's say, 50% more expensive to run than out of Oregon in the United States. Price can be determined by many factors, so AWS has very transparent granular pricing.

Availability Zones

AWS calls a single data center or a group of data centers, an Availability Zone or AZ. Each Availability Zone is one or more discrete data centers with redundant power, networking, and connectivity. When you launch an Amazon EC2 instance, it launches a virtual machine on a physical hardware that is installed in an Availability Zone. This means each AWS Region consists of multiple isolated and physically separate Availability Zones within a geographic Region.

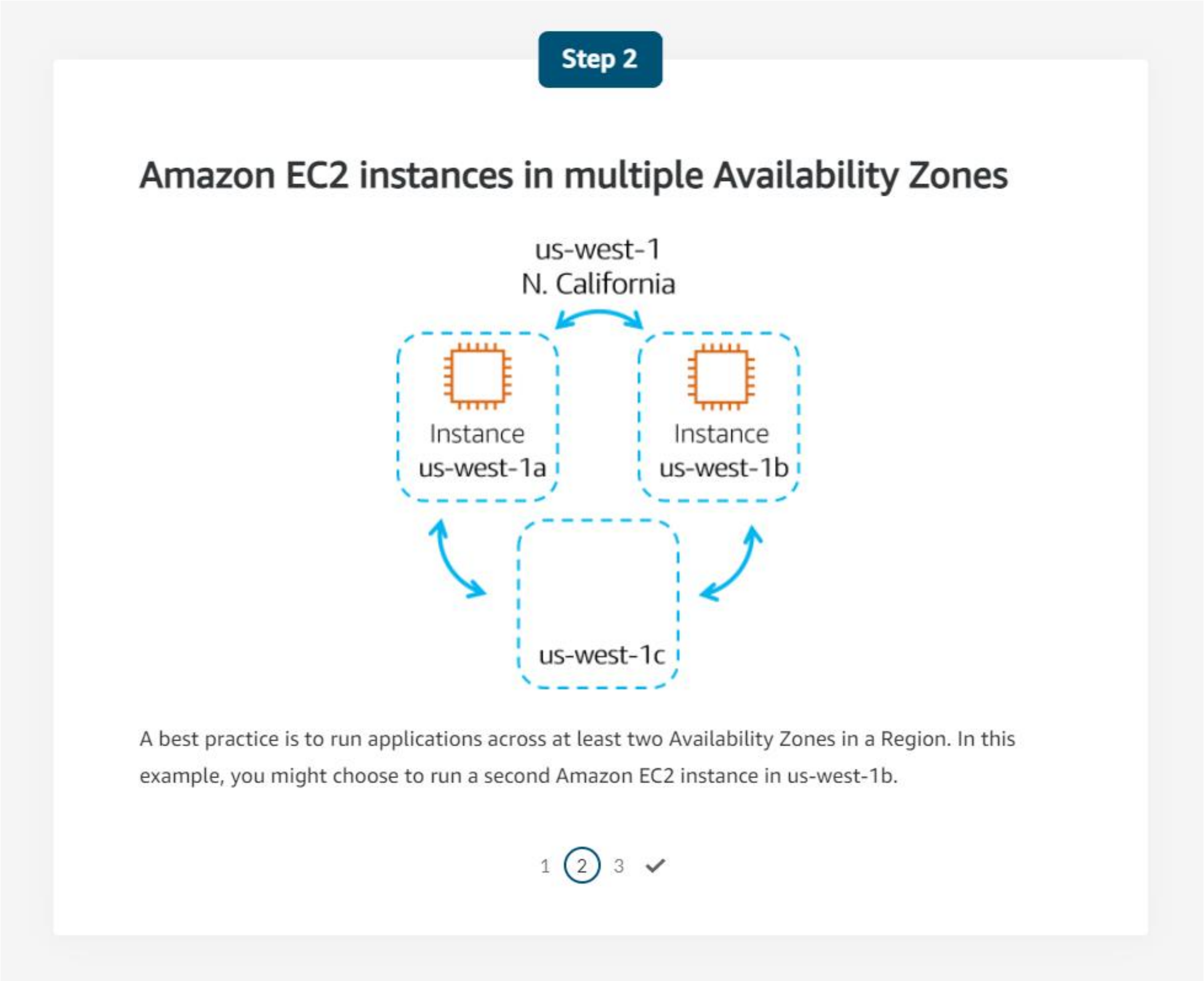
we always recommend you run across at least two Availability Zones in a Region. This means redundantly deploying your infrastructure in two different AZs



Step 1

Amazon EC2 instance in a single Availability Zone

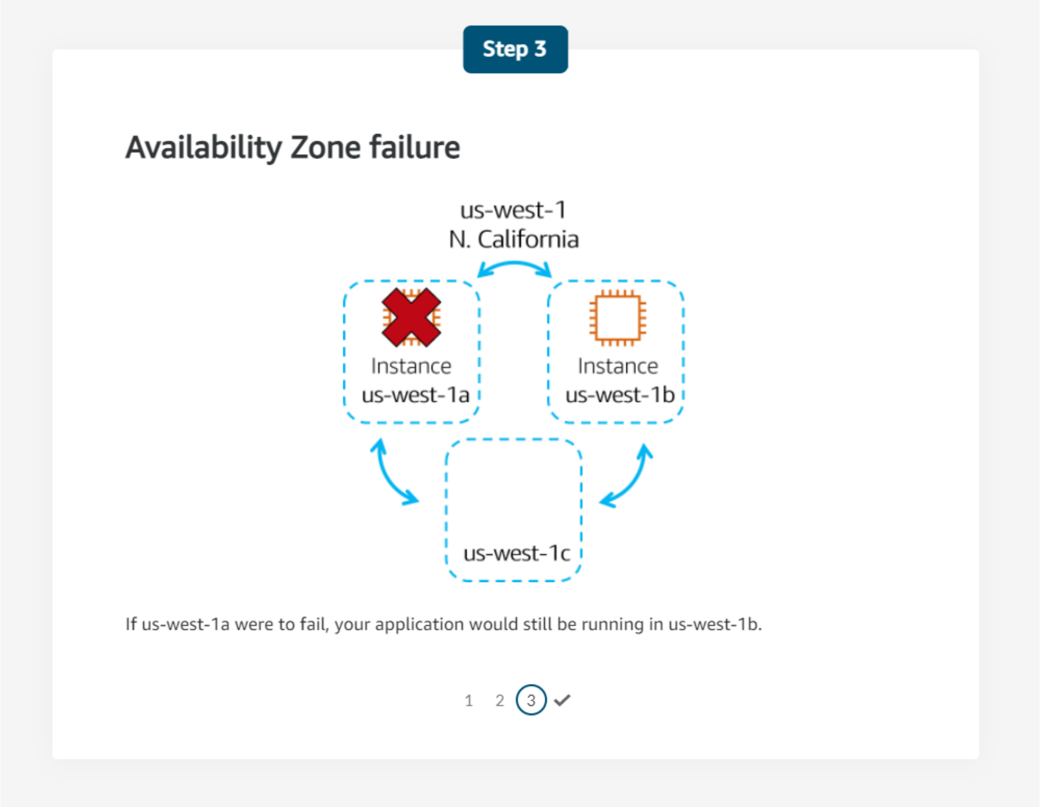
Suppose that you're running an application on a single Amazon EC2 instance in the Northern California Region. The instance is running in the us-west-1a Availability Zone. If us-west-1a were to fail, you would lose your instance.



Step 2

Amazon EC2 instances in multiple Availability Zones

A best practice is to run applications across at least two Availability Zones in a Region. In this example, you might choose to run a second Amazon EC2 instance in us-west-1b.



Step 3

Availability Zone failure

If us-west-1a were to fail, your application would still be running in us-west-1b.

Edge Locations

An **edge location** is a site that Amazon CloudFront uses to store cached copies of your content closer to your customers for faster delivery.

Caching copies of data closer to the customers all around the world uses the concept of content delivery networks, or CDNs.

CDNs are commonly used, and on AWS, we call our CDN Amazon CloudFront. Amazon CloudFront is a service that helps deliver data, video, applications, and APIs to customers around the world with low latency and high transfer speeds. Amazon CloudFront uses what are called Edge locations, all around the world, to help accelerate communication with users, no matter where they are.

Edge locations are separate from Regions, so you can push content from inside a Region to a collection of Edge locations around the world, in order to accelerate communication and content delivery. AWS Edge locations, also run more than just CloudFront. They run a domain name service, or DNS, known as Amazon Route 53, helping direct customers to the correct web locations with reliably low latency.

But what if your business wants to use, AWS services inside their own building? Well sure. AWS can do that for you. Introducing AWS Outposts, where AWS will basically install a fully operational mini Region, right inside your own data center. That's owned and operated by AWS, using 100% of AWS functionality, but isolated within your own building. It's not a solution most customers need, but if you have specific problems that can only be solved by staying in your own building, we understand, AWS Outposts can help.

key points

Number one, Regions are geographically isolated areas, where you can access services needed to run your enterprise.

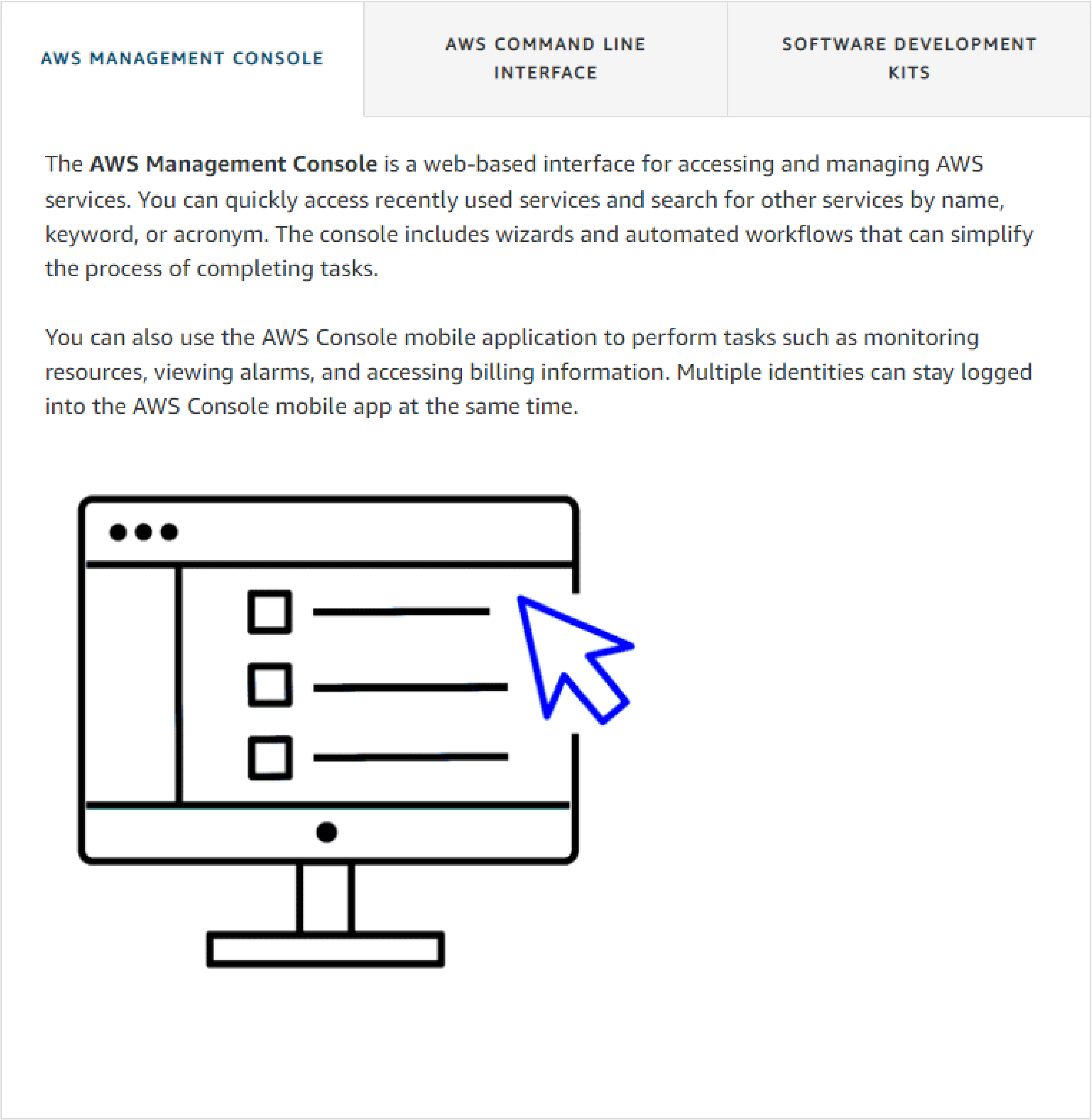
Number two, Regions contain Availability Zones, that allow you to run across physically separated buildings, tens of miles of separation, while keeping your application logically unified. Availability Zones help you solve high availability and disaster recovery scenarios, without any additional effort on your part.

Number three, AWS Edge locations run Amazon CloudFront to help get content closer to your customers, no matter where they are in the world.

How to provisions AWS resources

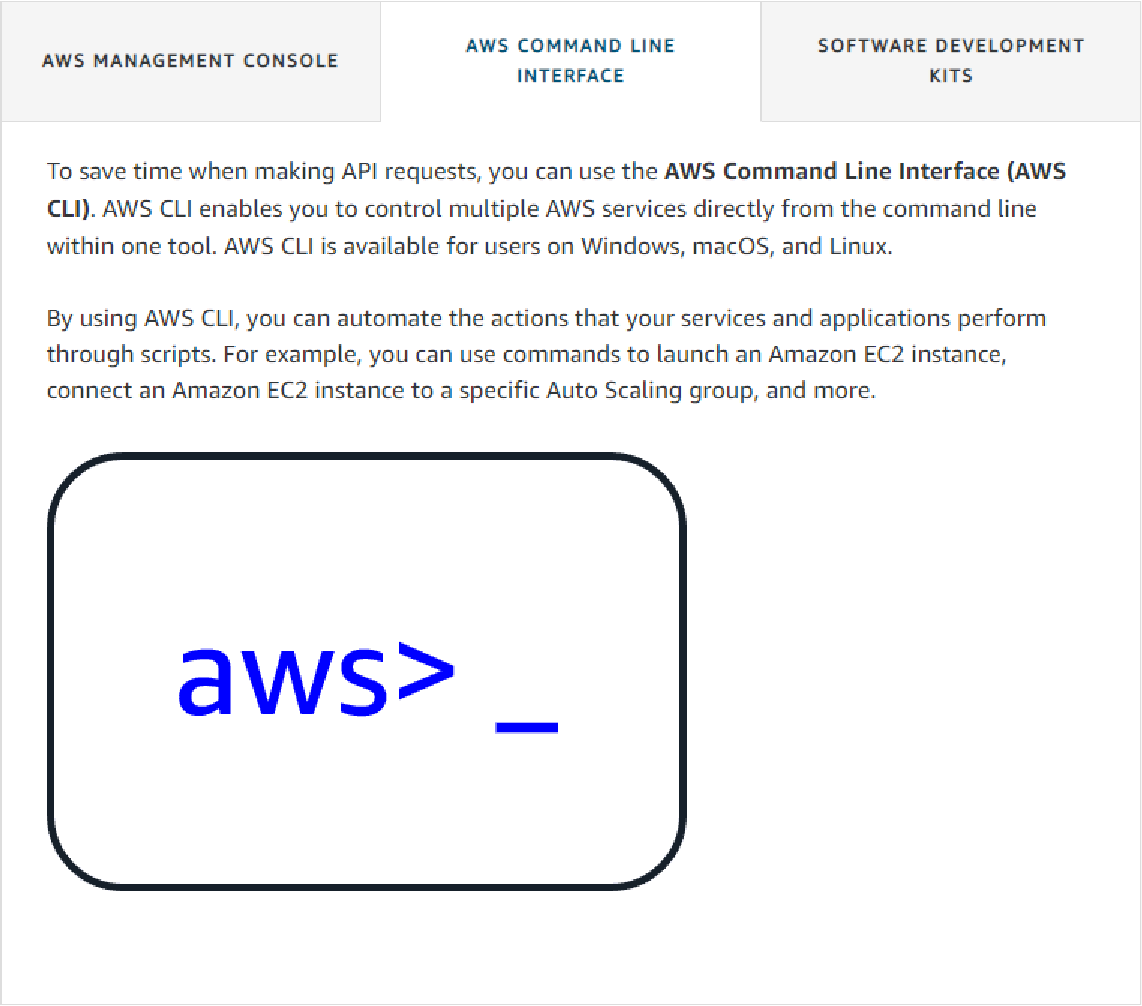
In AWS, everything is an API call. An API is an application programming interface.

You can use the AWS Management Console, the AWS Command Line Interface, the AWS Software Development Kits, or various other tools like AWS CloudFormation, to create requests to send to AWS APIs to create and manage AWS resources.



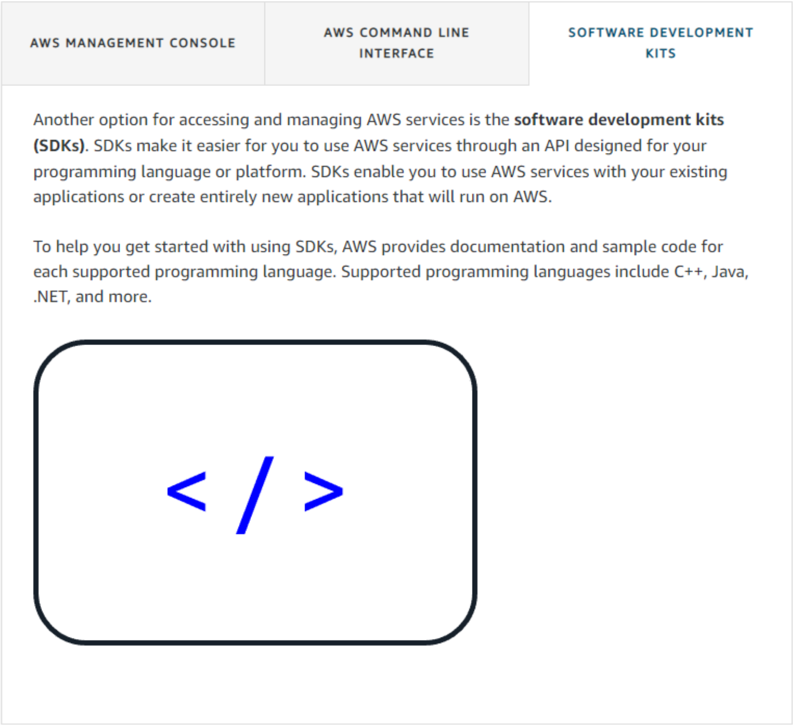
The **AWS Management Console** is a web-based interface for accessing and managing AWS services. You can quickly access recently used services and search for other services by name, keyword, or acronym. The console includes wizards and automated workflows that can simplify the process of completing tasks.

You can also use the AWS Console mobile application to perform tasks such as monitoring resources, viewing alarms, and accessing billing information. Multiple identities can stay logged into the AWS Console mobile app at the same time.



To save time when making API requests, you can use the **AWS Command Line Interface (AWS CLI)**. AWS CLI enables you to control multiple AWS services directly from the command line within one tool. AWS CLI is available for users on Windows, macOS, and Linux.

By using AWS CLI, you can automate the actions that your services and applications perform through scripts. For example, you can use commands to launch an Amazon EC2 instance, connect an Amazon EC2 instance to a specific Auto Scaling group, and more.



Another option for accessing and managing AWS services is the **software development kits (SDKs)**. SDKs make it easier for you to use AWS services through an API designed for your programming language or platform. SDKs enable you to use AWS services with your existing applications or create entirely new applications that will run on AWS.

To help you get started with using SDKs, AWS provides documentation and sample code for each supported programming language. Supported programming languages include C++, Java, .NET, and more.

AWS Elastic Beanstalk

With **AWS Elastic Beanstalk**, you provide code and configuration settings, and Elastic Beanstalk deploys the resources necessary to perform the following tasks:

- Adjust capacity
- Load balancing
- Automatic scaling
- Application health monitoring

AWS CloudFormation

With **AWS CloudFormation**, you can treat your infrastructure as code. This means that you can build an environment by writing lines of code instead of using the AWS Management Console to individually provision resources.

AWS CloudFormation provisions your resources in a safe, repeatable manner, enabling you to frequently build your infrastructure and applications without having to perform manual actions or write custom scripts. It determines the right operations to perform when managing your stack and rolls back changes automatically if it detects errors.