**Your goals**

* Know about AWS Fundamentals Course
* Be familiar with "The Six Pillars of AWS"
* Understand "Shared Responsibility Model"
* Have an idea about AWS CLI
* Learn about Tags
* Know about AWS Support
* Learn about AWS Quotas
* Know about AWS Cost Management
* Get brief familiarity with AWS provided services and domains
* Budget alerts

AWS has the most extensive global cloud infrastructure. No other cloud provider offers as many Regions with multiple Availability Zones connected by low latency, high throughput, and highly redundant networking. AWS has 80 Availability Zones within 25 geographic regions around the world and has announced plans for 15 more Availability Zones and 5 more AWS Regions.

The platform is developed with a combination of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offering.

* EC2- Elastic compute cloud
* VPC – virtual private network
* S3 – Simple Storage Service
* Relational Database Service
* Route 53 (DNS service) global, scalable
* ELB – Elastic Load Balancing
* Auto scaling

Each region has multiple availability zones. One availability zone is like a big data centre.

What is cloud computing? – Cloud computing is the on-demand delivery of compute power, database, storage, applications, and other IT resources through a cloud services platform through the internet with pay-as-you-go pricing.

Advantages of cloud computing

1. The cloud provides developers with greater flexibility, scalability, and faster time to innovation. With cloud computing, you can
2. Pay as you go - Pay only when you use computing resources, and only for how much you use.
3. Benefit from massive economies of scale - AWS aggregates usage from hundreds of thousands of customers in the cloud, which leads to higher economies of scale. This translates into lower pay-as-you-go prices.
4. Stop guessing capacity - When you make a capacity decision prior to deploying an application, you often end up either sitting on expensive idle resources or dealing with limited capacity. With cloud computing, you can access as much or as little capacity as you need and scale up and down as required with only a few minutes notice.
5. Increase speed and agility - IT resources are only a click away, which means that you reduce the time to make resources available to your developers from weeks to minutes. This dramatically increases agility for the organization, because the cost and time it takes to experiment and develop is significantly lower.
6. Realize cost savings - Companies can focus on projects that differentiate their business instead of maintaining data centres. With cloud computing, you can focus on your customers, rather than on the heavy lifting of racking, stacking, and powering physical infrastructure.
7. Go global in minutes - Applications can be deployed in multiple Regions around the world with a few clicks. This means that you can provide lower latency and a better experience for your customers at a minimal cost.

Enable access to billing data for the IAM admin user that you will create as follows:

1. On the navigation bar, choose your account name, and then choose **Account**.
2. Next to **IAM User and Role Access to Billing Information**, choose **Edit**. You must be signed in as the root user for this section to be displayed on the account page.
3. Select the check box to **Activate IAM Access** and choose **Update**.

Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) contains the basic building blocks for cloud IT, and typically provides access to networking features, computers (virtual or on dedicated hardware), and data storage space. IaaS provides you with the highest level of flexibility and management control over your IT resources and is most like existing IT resources that many developers are familiar with today.

Platform as a Service (PaaS)

Platform as a Service (PaaS) removes the need for you to manage the underlying infrastructure (usually hardware and operating systems) and allows you to focus on the deployment and management of your applications. This helps you be more efficient because you don’t need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.

Platform as a Service (PaaS)

Platform as a Service (PaaS) removes the need for you to manage the underlying infrastructure (usually hardware and operating systems) and allows you to focus on the deployment and management of your applications. This helps you be more efficient because you don’t need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.

AWS Cloud infrastructure is built around AWS Regions and Availability Zones. A Region is a physical location in the world where we have multiple Availability Zones. Availability Zones consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities. These Availability Zones offer you the ability to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible from a single data center.

How to interact with AWS

When infrastructure becomes virtual, as with cloud computing, the way developers work with infrastructure changes slightly. Instead of physically managing infrastructure, you logically manage it, through the AWS application programming interface (API). When you create, delete, or change any AWS resource, you will use API calls to AWS to do that.

You can make these API calls in several ways, but we will focus on these to introduce this topic:

* The AWS Management Console
* The AWS Command Line Interface (AWS CLI)
* IDE and IDE toolkits
* AWS Software Development Kits (SDKs)

The AWS Management Console

When first getting started with AWS, people often begin with the AWS Management Console, a web-based console that you log in to through a browser. The console comprises a broad collection of service consoles for managing AWS resources. By working in the console, you do not need to worry about scripting or syntax. You can also select the specific Region you want an AWS service to be in.

After working in the console, you may want to move away from manual deployment of AWS service, perhaps because you have become more familiar with AWS or are working in a production environment that requires a degree of risk management. This is where the AWS Command Line Interface (CLI) comes in.

AWS CLI

The [AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html?pg=cloudessentials) is an open source tool that enables you to create and configure AWS services using commands in your command-line shell. You can run commands in Linux or macOS using common shell programs such as bash, zsh, and tcsh, or on Windows, at the Windows command prompt or in PowerShell. One option for getting up and running quickly with the AWS CLI is [AWS CloudShell](https://aws.amazon.com/cloudshell/?pg=cloudessentials), a browser-based shell that provides command-line access to AWS resources. CloudShell is pre-authenticated with your console credentials. Common development and operations tools are pre-installed, so no local installation or configuration is required.

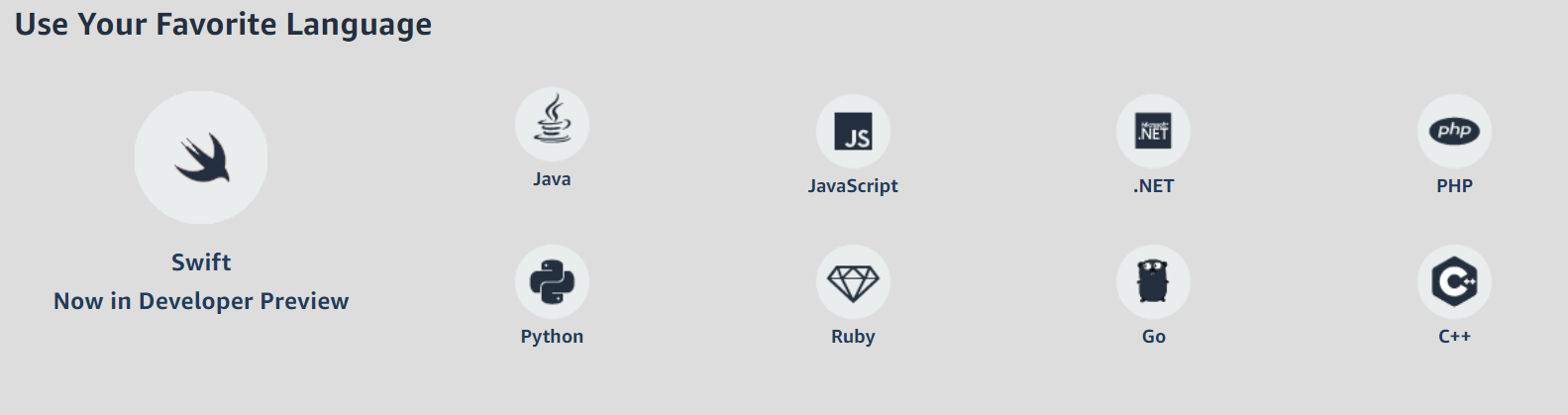
By moving to the AWS CLI, you can script or program the API calls. Instead of using a GUI, you create commands using a defined AWS syntax. One benefit of the CLI is that you can create single commands to create multiple AWS resources, which could help reduce the chance of human error when selecting and configuring resources. With the CLI, you need to learn the proper syntax for forming commands, but as you script these commands, you make them repeatable. This should save you time in the long run.

IDE and IDE Toolkits

AWS offers support for popular Integrated Development Environments (IDEs) and IDE toolkits so you can author, debug, and deploy your code on AWS from within your preferred environment. Supported IDEs and toolkits include [AWS Cloud9](https://aws.amazon.com/cloud9/?pg=cloudessentials), IntelliJ, PyCharm, Visual Studio, Visual Studio Code, Azure DevOps, Rider, and WebStorm.

SDKs

Software Development Kits (SDKs) are tools that allow you to interact with the AWS API programmatically. AWS creates and maintains SDKs for most popular programming languages, including those shown in the following diagram.

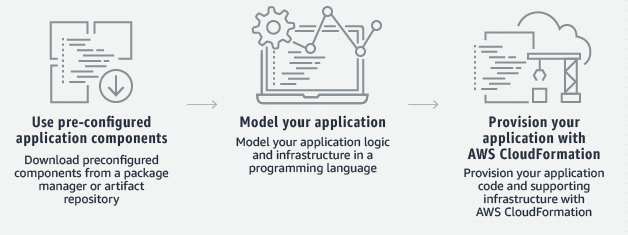


SDKs come in handy when you want to integrate your application source code with AWS services. For example, you might use the [Python SDK](https://boto3.amazonaws.com/v1/documentation/api/latest/guide/index.html?pg=cloudessentials) to write code to store files in Amazon Simple Storage Service (Amazon S3) instead of on your local hard drive. The ability to manage AWS services from a place where you can run source code, with conditions, loops, arrays, lists, and other programming elements, provides a lot of power and creativity.

AWS CDK – Infrastructure as a code

The AWS Cloud Development Kit (AWS CDK) is a software development framework for defining cloud infrastructure in code and provisioning it through AWS CloudFormation. The AWS CDK supports familiar programming languages, such as TypeScript, JavaScript, Python, Java, C#/.Net, and Go (in developer preview).

Provisioning cloud applications can be challenging, requiring you to write custom scripts, maintain templates, or learn domain-specific languages. AWS CDK uses the familiarity and expressive power of programming languages for modelling your applications. It provides high-level components called constructs that preconfigure cloud resources with proven defaults, so you can build cloud applications with ease. AWS CDK provisions your resources in a safe, repeatable manner through AWS CloudFormation. It also allows you to compose and share your own custom constructs incorporating your organization's requirements, helping you expedite new projects.



Well-structured architecture

Graphical user interface

Description automatically generated

Operational Excellence Pillar

The operational excellence pillar focuses on running and monitoring systems, and continually improving processes and procedures. Key topics include automating changes, responding to events, and defining standards to manage daily operations.

Security Pillar

The security pillar focuses on protecting information and systems. Key topics include confidentiality and integrity of data, managing user permissions, and establishing controls to detect security events.

Reliability Pillar

The reliability pillar focuses on workloads performing their intended functions and how to recover quickly from failure to meet demands. Key topics include distributed system design, recovery planning, and adapting to changing requirements.

Performance Efficiency Pillar

The performance efficiency pillar focuses on structured and streamlined allocation of IT and computing resources. Key topics include selecting resource types and sizes optimized for workload requirements, monitoring performance, and maintaining efficiency as business needs evolve

Cost Optimization Pillar

The cost optimization pillar focuses on avoiding unnecessary costs. Key topics include understanding spending over time and controlling fund allocation, selecting resources of the right type and quantity, and scaling to meet business needs without overspending.

Sustainability Pillar

The sustainability pillar focuses on minimizing the environmental impacts of running cloud workloads. Key topics include a shared responsibility model for sustainability, understanding impact, and maximizing utilization to minimize required resources and reduce downstream impacts.

Security

When you build applications on AWS, managing security and compliance is a shared responsibility between AWS and you. To depict this shared responsibility, AWS created the shared responsibility model (see the following diagram). The distinction of responsibility is commonly referred to as security “of” the cloud compared to security “in” the cloud.

AWS responsibility

Being responsible for security of the cloud means that AWS protects and secures the infrastructure that runs the services offered in the AWS Cloud. AWS is responsible for:

1. Protecting and securing AWS Regions, Availability Zones, and data centres, down to the physical security of the buildings.
2. Managing the hardware, software, and networking components that run AWS services, such as the physical servers, host operating systems, virtualization layers, and AWS networking components.

Customer responsibility

1. Customers, or anyone building on the cloud, are responsible for security in the cloud. When using any AWS service, you’re responsible for properly configuring the service and your applications, in addition to ensuring that your data is secure.
2. Your level of responsibility depends on the AWS service. Some services require you to perform all the necessary security configuration and management tasks, while other more abstracted services require you to only manage the data and control access to your resources.
3. Due to the varying levels of effort, customers must consider which AWS services they use and review the level of responsibility required to secure each service. They must also review how the shared security model aligns with the security standards in their IT environment, in addition to any applicable laws and regulations.
4. A key concept is that customers maintain complete control of their data and are responsible for managing the security related to their content.

**AWS responsibility “Security of the Cloud”** - AWS is responsible for protecting the infrastructure that runs all of the services offered in the AWS Cloud. This infrastructure is composed of the hardware, software, networking, and facilities that run AWS Cloud services.

**Customer responsibility “Security in the Cloud”** – Customer responsibility will be determined by the AWS Cloud services that a customer selects. This determines the amount of configuration work the customer must perform as part of their security responsibilities. For example, a service such as Amazon Elastic Compute Cloud (Amazon EC2) is categorized as Infrastructure as a Service (IaaS) and, as such, requires the customer to perform all the necessary security configuration and management tasks. Customers that deploy an Amazon EC2 instance are responsible for management of the guest operating system (including updates and security patches), any application software or utilities installed by the customer on the instances, and the configuration of the AWS-provided firewall (called a security group) on each instance. For abstracted services, such as Amazon S3 and Amazon DynamoDB, AWS operates the infrastructure layer, the operating system, and platforms, and customers access the endpoints to store and retrieve data. Customers are responsible for managing their data (including encryption options), classifying their assets, and using IAM tools to apply the appropriate permissions.

AWS for each account provides a basic plan. The basic plan is free for each AWS customer. You can change the basic plan to three different paid plans:

* Developer
* Business
* Enterprise

**Tagging AWS resources**

You can assign metadata to your AWS resources in the form of tags. Each tag is a label consisting of a user-defined key and value. Tags can help you manage, identify, organize, search for, and filter resources. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags that you've assigned to it.

Tags for resource organization Tags are a good way to organize AWS resources in the AWS Management Console, you can:

* Configure tags to be displayed with resources
* Search and filter resources by tag

**What are the services provided by AWS?**

Each type of service is categorized under a domain:

* AWS Networking and Content Delivery services
* AWS Compute services
* AWS Storage services
* Database services
* Analytics
* Developer Tools
* Management tools
* IoT
* Security
* Enterprise app

AWS Budgets and Budget actions

Imagine you can put some real teeth to your AWS Budgets.  Rather than warning you of a forecasted or actual cost overage, AWS can act on your behalf and stop the activities that caused the overage. We are pleased to announce that this came to a reality with the launch of AWS Budget Actions. Our Senior Product Manager, Matt Cowsert, will share how you can create your very own budget actions today.

AWS Budgets now allows you to configure actions — responses to cost and usage in your account or set of accounts— that will be applied automatically or via a workflow approval process once a budget target has been exceeded. There are three action types: Identity and Access Management (IAM) policies, Service Control policies (SCPs), or target running instances (EC2 or RDS). Actions can be configured for actual (after they’ve occurred) or for forecasted (before they occur) budgeted amounts.

Service control policies (SCPs) are a type of organization policy that you can use to manage permissions in your organization. SCPs offer central control over the maximum available permissions for all accounts in your organization. You can configure the service control policies (SCPs) in your organization to work as either of the following:

* A deny list – actions are allowed by default, and you specify what services and actions are prohibited
* An allow list – actions are prohibited by default, and you specify what services and actions are allowed