_-CrescentSoftServices.com

Introduction to Cloud Computing

- Introduction to Cloud Computing
- Features, Benefits and Advantages
- Types of Cloud Computing
- Public, Private, Community and Hybrid Cloud
- Public vs. Private Cloud Computing
- Service Models
- IAAS
- PAAS
- SAAS
- Pros and Cons of Cloud Computing

Cloud computing is the delivery of computing services over the Internet.

Cloud Computing is a Utility Service, giving you access to technology resources managed by experts and available on-demand

Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations

Examples of cloud services include online file storage, social networking sites, webmail, and online business applications.

The cloud computing model allows access to information and computer resources from anywhere that a network connection is available.

Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications

Cloud Computing Definition :

The following definition of cloud computing has been developed by the U.S. National Institute of Standards and Technology (NIST):

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

This cloud model promotes availability and is composed of

Five essential characteristics

Three service models, and

Four deployment models

Characteristics :

On-demand self service - customers (usually organizations) can request and manage their own computing resources

Broad network access - allows services to be offered over the Internet or private networks

Resource pooling - customers draw from a pool of computing resources, usually in remote data centres **Rapid elasticity** - Services can be scaled larger or smaller

Measured service - Customers are billed according to the use of the Service made

Service models :

The cloud computing service models are **Software as a Service (SaaS)**
Platform as a Service (PaaS) and Infrastructure as a Service (IaaS)

In a **Software as a Service model**, a pre-made application, along with any required software, operating system, hardware, and network are provided.

Ex: Google Apps, Salesforce, Citrix GoToMeeting, Cisco WebEx

In **PaaS**, an operating system, hardware, and network are provided, and the customer installs or develops its own software and applications.

Ex: AWS Elastic Beanstalk, Windows Azure

The **IaaS** model provides just the hardware and network; the customer installs or develops its own operating systems, software and applications

Ex: Amazon Web Services (AWS), Microsoft Azure

Deployment of cloud services (Types):

Private cloud

Public cloud

Community cloud

Hybrid cloud

In a private cloud, the cloud infrastructure is operated solely for a specific organization, and is managed by the organization or a third party.

Services provided by a public cloud are offered over the Internet and are owned and operated by a cloud provider.

In a community cloud, the service is shared by several organizations and made available only to those groups. The infrastructure may be owned and operated by the organizations or by a cloud service provider.

A hybrid cloud is a combination of different methods of resource pooling (for example, combining public and community clouds)

Self-service provisioning: End users can spin up computing resources for almost any type of workload on-demand.

Scalability and Elasticity: Companies can scale up as computing needs increase and then scale down again as demands decrease.

Pay per use: Computing resources are measured at a granular level, allowing users to pay only for the resources and workloads they use.

No Capital Expenditure

Deploy Projects Faster

Lower Maintenance Costs

Redundancy and High Availability

Work from anywhere

Automatic Software updates and Disaster recovery etc…

Public Cloud:

Public clouds are made available to the general public by a service provider who hosts the cloud infrastructure

Public Cloud customers benefit from economies of scale, because infrastructure costs are spread across all users, allowing each individual client to operate on a low-cost, " pay-as-you-go" model

A public cloud is the obvious choice when:

- Your standardized workload for applications is used by lots of people, such as e-mail
- You need to test and develop application code.
- You need incremental capacity (the ability to add compute resources for peak times).
- You" re doing collaboration projects

Examples :

Amazon Elastic Compute Cloud (EC2),

IBM's Blue Cloud,

Sun Cloud,

Google AppEngine and Windows Azure Services Platform

Private Cloud:

A private cloud is a particular model of cloud computing that involves a distinct and secure cloud based environment in which only the specified client can operate.

As with other cloud models, private clouds will provide computing power as a service within a virtualized environment using an underlying pool of physical computing resource.

However, under the private cloud model, the cloud (the pool of resource) is only accessible by a single organization providing that organization with greater control and privacy.

Two variations in Private :

On-Premise Private Cloud - hosted within an organization" s own facility

Externally Hosted Private Cloud - hosted by a third party specializing in cloud infrastructure

When is a Private Cloud for you?

- You need data sovereignty but want cloud efficiencies
- You want consistency across services
- You have more server capacity than your organization can use
- Your data center must become more efficient
- You want to provide private cloud services

Hybrid Cloud:

Hybrid Clouds are a composition of two or more clouds (private, community or public) that remain unique entities but are bound together offering the advantages of multiple deployment models

Hybrid cloud architecture requires both on-premise resources and off-site server based cloud infrastructure

Here are a couple of situations where a hybrid environment is best:

Your company wants to use a SaaS application but is concerned about security.

Your company offers services that are tailored for different vertical markets.

You can use a public cloud to interact with the clients but keep their data secured within a private cloud. You can provide public cloud to your customers while using a private cloud for internal IT

Community Cloud:

A community cloud is a is a multi-tenant cloud service model that is shared among several or organizations and that is governed, managed and secured commonly by all the participating organizations or a third party managed service provider. The goal of community clouds is to have

participating organizations realize the benefits of a public cloud with the added level of privacy, security, and policy compliance usually associated with a private cloud.

Community clouds can be either on-premise or off-premise.

Here are a couple of situations where a community cloud environment is best:

- Government organizations within a state that need to share resoures
- A private HIPAA (Health Insurance) compliant cloud for a group of hospitals or clinics
- Telco community cloud for telco DR

Both public and private clouds have management implications. However, by choosing a public cloud solution, an organization can offload much of the management responsibility to its cloud vendor.

In a private cloud scenario, there is significant demand on resources to specify, purchase, house, update, maintain, and safeguard the physical infrastructure. Financially, deploying a private cloud can also create a large initial **capital expense**, with subsequent investment required as new equipment and capacity is added.

In a public cloud scenario, capital expense is virtually **eliminated**; the financial burden is shifted to a fee-for-service, often based on utilization and data volume.

Maintaining and **securing** public cloud infrastructure is the responsibility of the vendor, enabling the customer organization to streamline IT operations and minimize time and money spent on system upkeep

In addition to reducing or eliminating capital expense, many organizations prefer a public cloud solution for its available, **on-demand capacity**. Accessing the shared resources of a public cloud on an as-needed basis can remove ‘headroom' worries from the IT planning process

SaaS:

SaaS, or Software as a Service, describes any cloud service where consumers are able to access software applications over the internet.

The applications are hosted in “the cloud” and can be used for a wide range of tasks for both individuals and organisations.

Google, Twitter, Facebook and Flickr are all examples of SaaS, with users able to access the services via any internet enabled device.

Enterprise users are able to use applications for a range of needs, including accounting and invoicing, tracking sales, planning, performance monitoring and communications.

Benefits:

No additional hardware costs

No initial setup cost

Usage is scalable

Updates are automated

Cross device compatibility

Accessible from any location

Gmail is one famous example of an SaaS mail provider

PaaS:

PaaS (Platform as a Service), as the name suggests, provides you computing platforms which typically includes operating system, programming language execution environment, database, web server etc.

PaaS makes the development, testing, and deployment of applications quick, simple, and cost-effective, eliminating the need to buy the underlying layers of hardware and software

Platform as a Service" (PaaS) deliver computational resources through a platform, developers gain with PaaS is a framework they can build upon to develop or customize applications

Examples: AWS Elastic Beanstalk, Windows Azure, Google App Engine

IaaS:

Infrastructure as a service delivers basic storage and compute capabilities as

standardized services over the network. Servers, storage systems, switches, routers,

and other systems are pooled and made available to handle workloads that range

from application components to high-performance computing applications

Instead of having to purchase hardware outright, users can purchase IaaS based on consumption, similar to electricity or other utility billing.

Compared to SaaS and PaaS, IaaS users are responsible for managing applications, data, runtime, middleware, and OSes.

Providers still manage virtualization, servers, hard drives, storage, and networking. Many IaaS providers now offer databases, messaging queues, and other services above the virtualization layer as well

Examples: Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine (GCE)

Pros

- Lower upfront costs and reduced infrastructure costs.
- Easy to grow your applications.
- Scale up or down at short notice.
- Only pay for what you use.
- Everything managed under SLAs.
- Overall environmental benefit (lower carbon emissions) of many users efficiently sharing large systems.

Cons

- Greater dependency on service providers. Can you get problems resolved quickly, even with SLAs?
- Risk of being locked into proprietary or vendor-recommended systems? How easily can you migrate to another system or service provider if you need to?
- What happens if your supplier suddenly decides to stop supporting a product or system you've come to depend on?
- Potential privacy and security risks of putting valuable data on someone else's system in an unknown location?
- Dependency on a reliable Internet connection.

AWS Introduction and Architecture

- Introduction to Amazon Web Services
- AWS and other Cloud Service Providers
- AWS Security
- AWS Architecture
- AWS Services and Products
- AWS Icons Overview
- AWS Management Console

- Setting up of the AWS Account

Amazon Web Services (AWS) provides on-demand computing resources and services in the cloud, with pay-as-you-go pricing.

For example, you can run a server on AWS that you can log on to, configure, secure, and run just as you would a server that's sitting in front of you

Using AWS resources instead of your own is like purchasing electricity from a power company instead of running your own generator

It provides many of the same benefits:

Capacity exactly matches your need,

You pay only for what you use,

Economies of scale result in lower costs, and

The service is provided by a vendor experienced in running large-scale networks **Verizon Cloud** - Almost perfect, except support sucks, just like PWSRN.

Joyent - networking just not flexible enough

Microsoft Azure - Almost perfect. Lack of 'certified' Debian image unfortunate. Cost for advanced router is too high.

Rackspace - no static IP's a blocker, just like Joyent and Google Cloud

Century Link - no private network choice unfortunate but not blocker. Very helpful and responsive support.

IBM/SoftLayer - Strong backend network infrastructure/DC complex. Still evaluating.

How Do I Access AWS?

AWS provides several ways to create and manage resources.

AWS Management Console

A web interface.

AWS Command Line Interface (AWS CLI)

Commands for a broad set of AWS products.

Command Line Tools

Commands for individual AWS products.

AWS Software Development Kits (SDK) APIs that are specific to your programming language or platform.

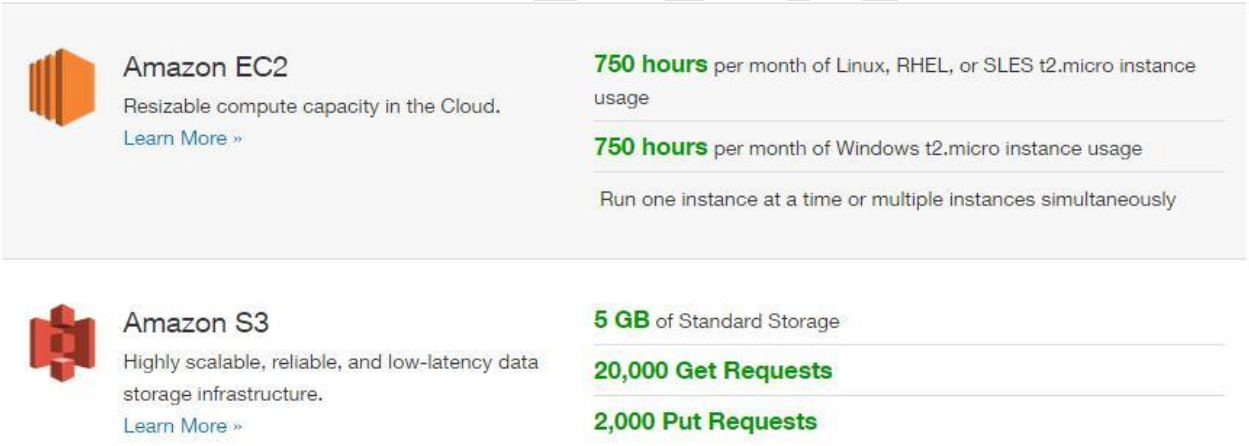
Query APIs Low-level APIs that you access using HTTP requests.

Pricing

AWS can offer significant cost savings compared to the equivalent on-premises infrastructure.

You can use the **AWS Simple Monthly Calculator** to estimate what it would cost to use AWS.

Note that if you created your AWS account within the last 12 months, you are eligible for the **AWS Free Tier**.



Amazon has **data centers** in different areas of the world (for example, North America, Europe, and Asia). Correspondingly, AWS products are available to use in different *regions*. By placing resources in separate regions, you can design your website or app to be closer to specific customers or to meet legal or other requirements. Note that prices for AWS usage vary by region.

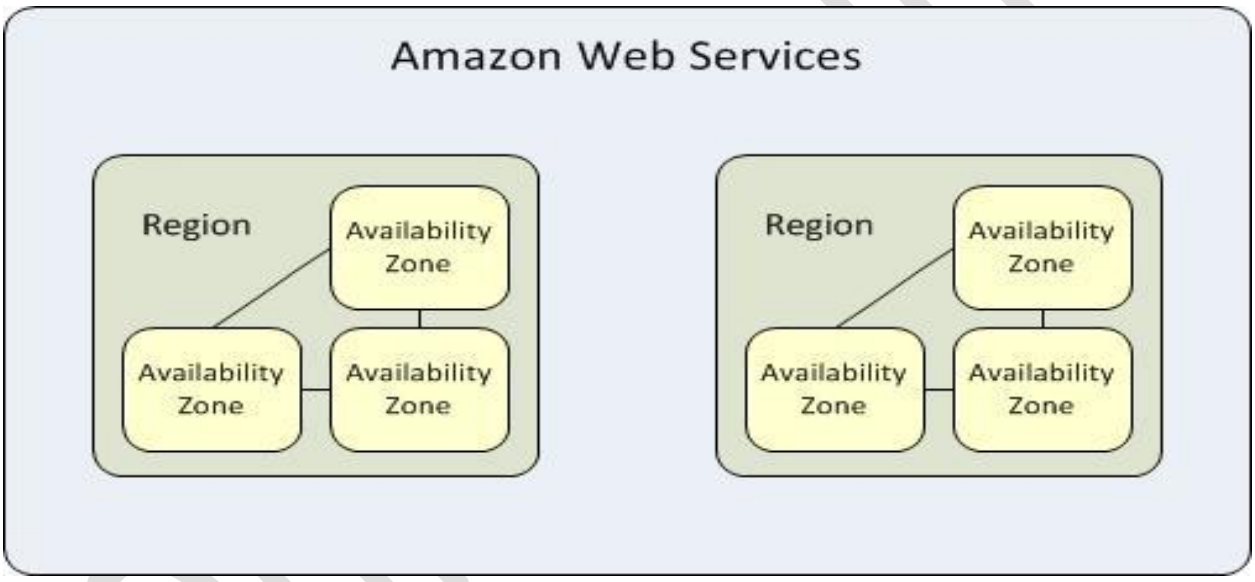
Each region contains multiple distinct locations called *Availability Zones*. Each Availability Zone is engineered to be isolated from failures in other Availability Zones, and to provide inexpensive, low-latency network connectivity to other zones in the same region. By placing resources in separate Availability Zones, you can protect your website or app from the failure of a single location.

AWS resources can be tied to a **region** or tied to an **Availability Zone**. Not every region or Availability Zone supports every AWS resource. When you view your resources, you'll only see the resources tied to the region you've specified. This is because regions are isolated from each other, and we don't replicate resources across regions automatically

Each region is completely independent.

Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links.

The following diagram illustrates the relationship between regions and Availability Zones



Name	Region Code
Asia Pacific (Tokyo)	ap-northeast-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
EU (Frankfurt)	eu-central-1
EU (Ireland)	eu-west-1
South America (Sao Paulo)	sa-east-1
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2

Regions

Each Amazon EC2 region is designed to be completely isolated from the other Amazon EC2 regions. This achieves the greatest possible fault tolerance and stability.

Amazon EC2 provides multiple regions so that you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements

When you view your resources, you'll only see the resources tied to the region you've specified. This is because regions are isolated from each other, and we don't replicate resources across regions automatically.

When you work with an instance using the command line interface or API actions, you must specify its regional endpoint.

When you launch an instance, you must select an AMI that's in the same region.

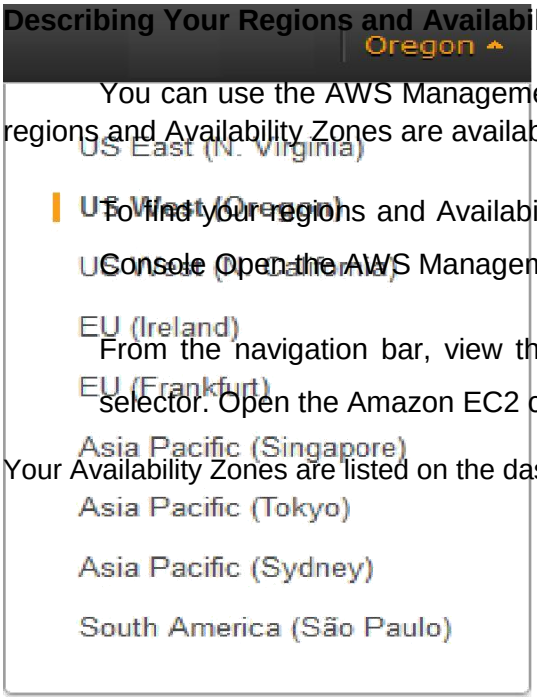
If the AMI is in another region, you can copy the AMI to the region you're using.

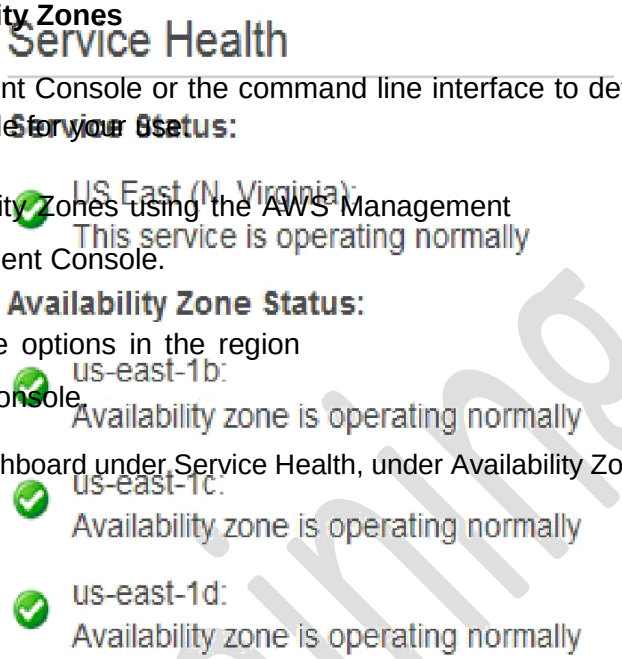
All communications between regions is across the public Internet.

Therefore, you should use the appropriate encryption methods to protect your data.

Data transfer between regions is charged at the Internet data transfer rate for both the sending and the receiving instance.

Describing Your Regions and Availability Zones





You can use the AWS Management Console or the command line interface to determine which regions and Availability Zones are available for your use.

To find your regions and Availability Zones using the AWS Management Console Open the AWS Management Console.

From the navigation bar, view the options in the region selector. Open the Amazon EC2 console.

Your Availability Zones are listed on the dashboard under Service Health, under Availability Zone Status.

AWS provides a secure global infrastructure, plus a range of features that you can use to secure your data in the cloud. The following are highlights:

Physical access to AWS data centers is strictly controlled, monitored, and audited. Access to the AWS network is strictly controlled, monitored, and audited.

You can manage the security credentials that enable users to access your AWS account using AWS Identity and Access Management (IAM).

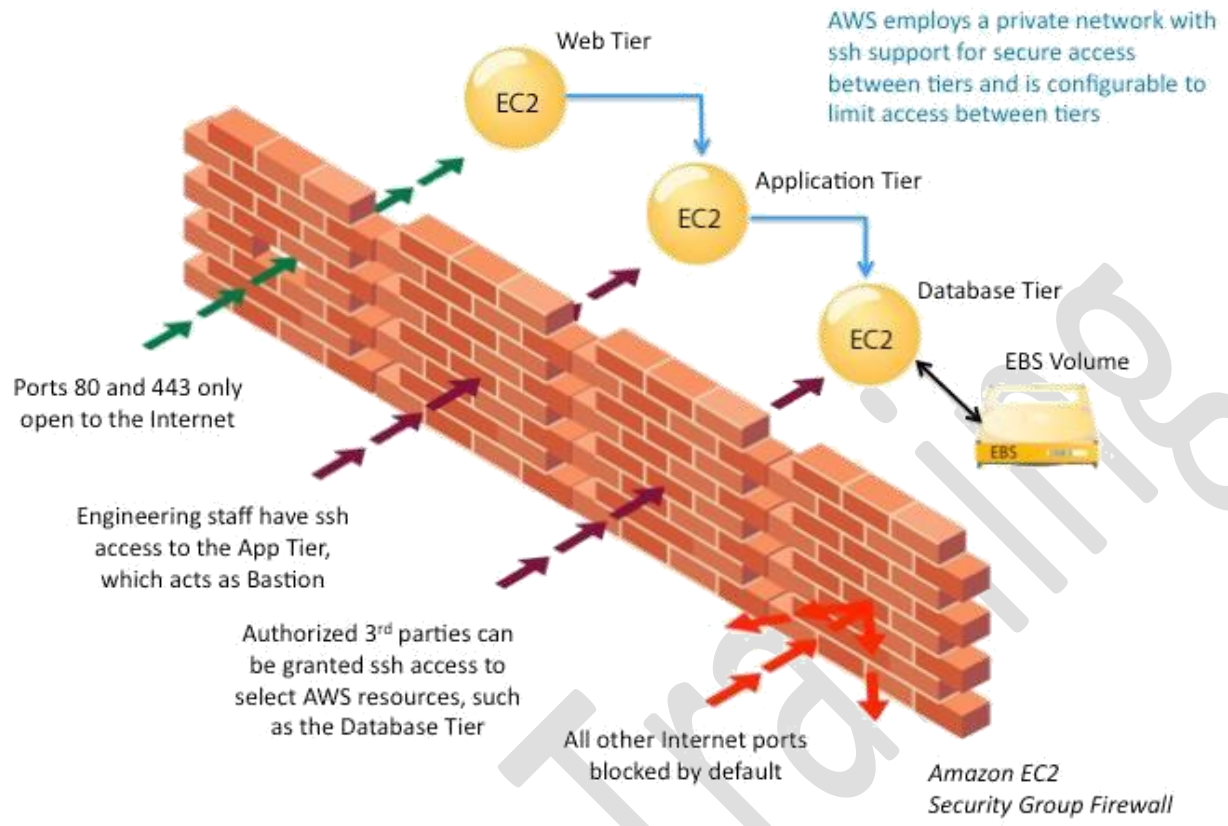
You can apply ACL-type permissions on your data and can also use encryption of data at rest.

You can set up a virtual private cloud (VPC), which is a virtual network that is logically isolated from other virtual networks in the AWS cloud. You can control whether the network is directly routable to the Internet.

You control and configure the operating system on your virtual server.

You can set up a security group, which acts as a virtual firewall to control the inbound and outbound traffic for your virtual servers.

You can specify a key pair when you launch your virtual server, which is used to encrypt your login information. When you log in to your virtual server, you must present the private key of the key pair to decrypt the login information



AWS Product Categories

AWS offers a broad set of services.

Compute and Networking Services

Storage and Content Delivery Services Database Services

Analytics Services App Services

Deployment Services Management Services

Amazon EC2
Auto Scaling

Elastic Load Balancing Amazon

VPC
Amazon Route 53

AWS provides a variety of computing and networking services to meet the needs of your applications. You can provision virtual servers, set up a firewall, configure Internet access, allocate and route IP addresses, and scale your infrastructure to meet increasing demand.

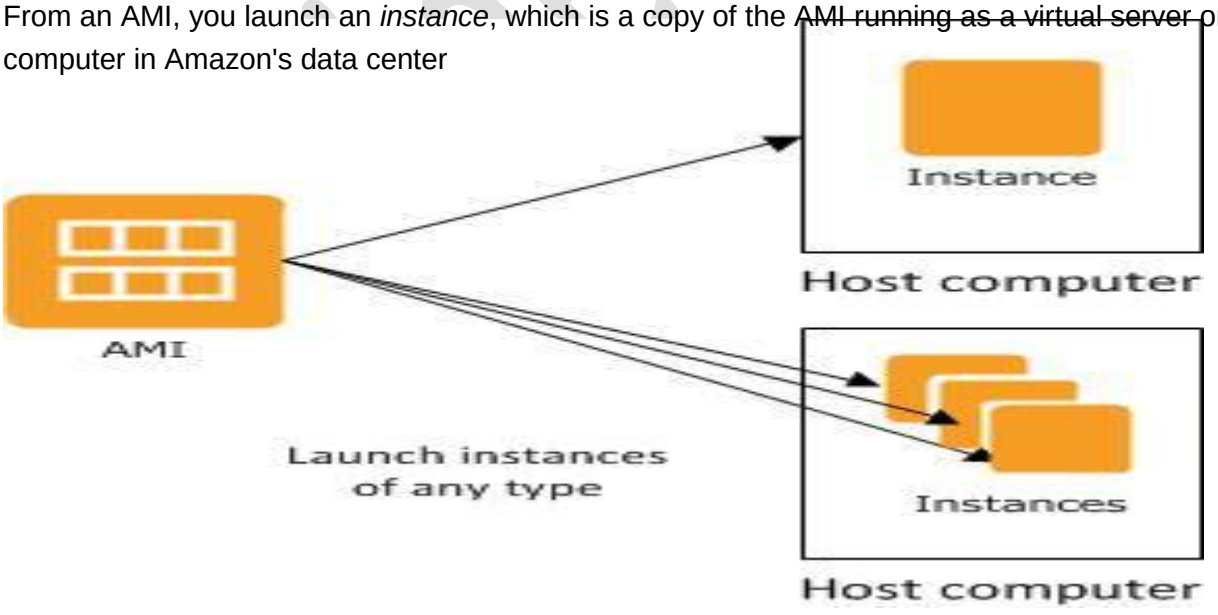
You can use the compute and networking services with the storage, database, and application services to provide a complete solution for computing, query processing, and storage across a wide range of applications

Instances and AMIs

Amazon Elastic Compute Cloud (Amazon EC2) provides resizeable computing capacity—literally, servers in Amazon's data centers—that you use to build and host your software systems.

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications).

From an AMI, you launch an *instance*, which is a copy of the AMI running as a virtual server on a host computer in Amazon's data center



When you launch an instance, you select an *instance type*, which determines the hardware capabilities (such as memory, CPU, and storage) of the host computer for the instance. You can access your instance using its assigned public DNS name or public IP address. The public DNS names for instances are as follows:

The US East (N. Virginia) region

ec2-*public_ip*.compute-1.amazonaws.com

Other regions

ec2-*public_ip.region_code*.compute.amazonaws.com

Your instances keep running until you stop or terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI.

You start from an existing AMI that most closely meets your needs, log on to the instance, and then customize the instance with additional software and settings. You can save this customized configuration as a new AMI, which you can then use to launch new instances whenever you need them

VPCs and Subnets

A *virtual private cloud* (VPC) is a virtual network dedicated to your AWS account.

It is logically isolated from other virtual networks in the AWS cloud, providing security and robust networking functionality for your compute resources.

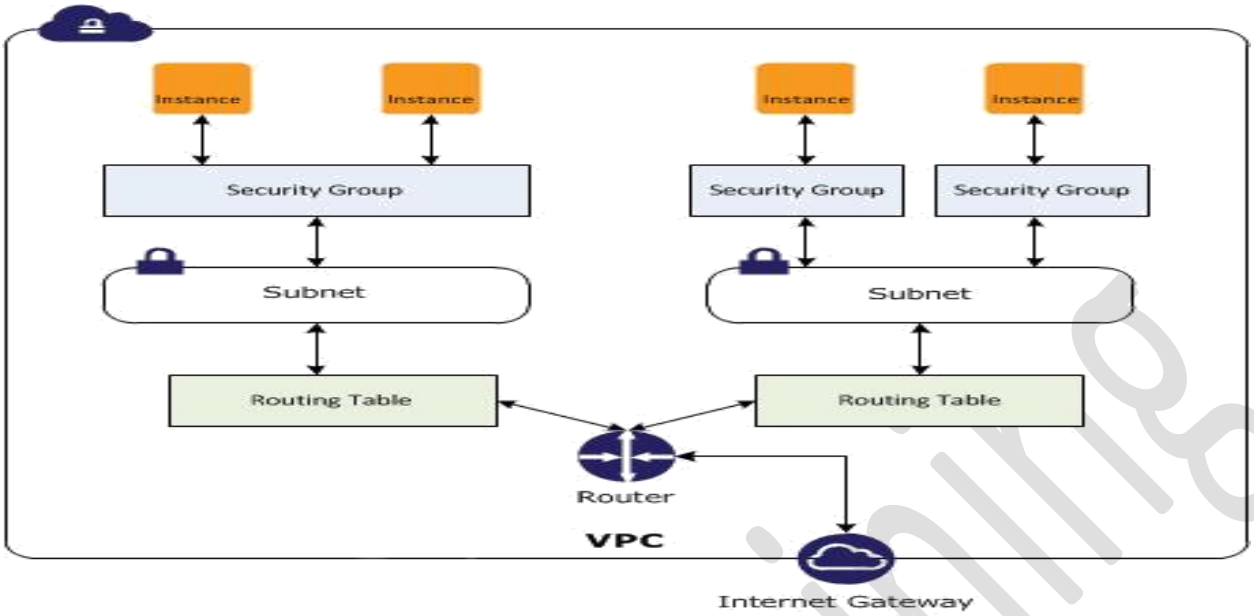
A VPC closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

A *subnet* is a segment of a VPC's IP address range that you can launch instances into. Subnets enable you to group instances based on your security and operational needs. To enable instances in a subnet to reach the Internet and AWS services, you must add an Internet gateway to the VPC and a route table with a route to the Internet to the subnet.

We recommend that you launch your EC2 instances into a VPC. Note that if you created your AWS account after 2013-12-04, you have a default VPC and you must launch EC2 instances into a default or a nondefault VPC

Security Groups

A *security group* acts as a virtual firewall for your instance to control inbound and outbound traffic. You can specify one or more security groups when you launch your instance. When you create a security group, you add *rules* that control the inbound traffic that's allowed, and a separate set of rules that control the outbound traffic. All other traffic is discarded. You can modify the rules for a security group at any time and the new rules are automatically enforced.



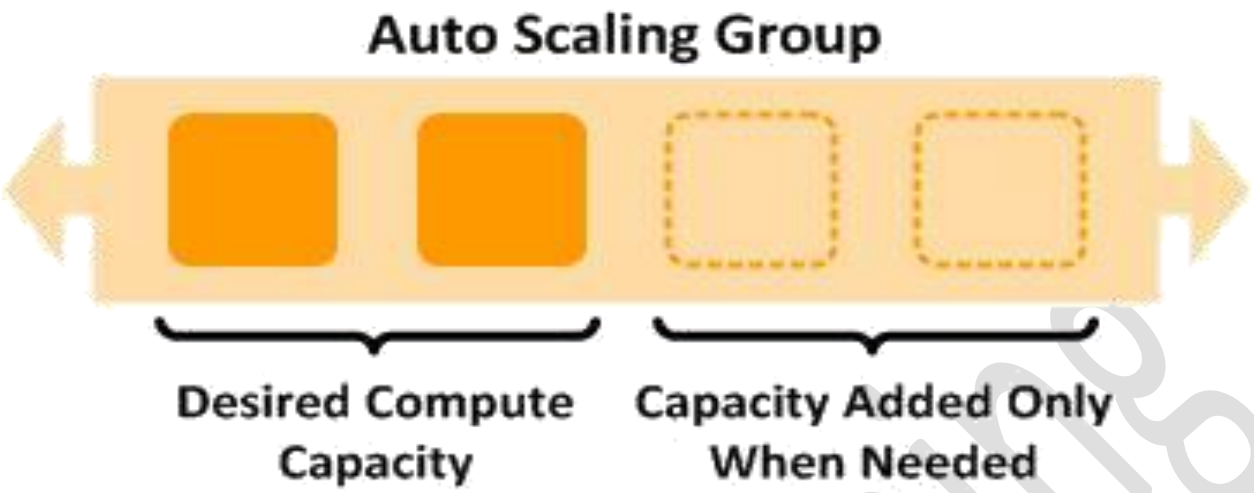
Amazon Route 53 is a highly available and scalable cloud **Domain Name System (DNS) web service**. It is designed as an extremely reliable and cost-effective way to route visitors to websites by translating domain names (such as www.example.com) into the numeric IP addresses (such as 192.0.2.1) that computers use to connect to each other.

AWS assigns URLs to your AWS resources, such as your EC2 instances. However, you might want a URL that is easy for your users to remember.

For example, you can map your domain name to your AWS resource. If you don't have a domain name, you can search for available domains and register them using Amazon Route 53. If you have an existing domain name, you can transfer it to Amazon Route 53.

Auto Scaling Groups

Auto Scaling supports groups of virtual servers, an *Auto Scaling group* that can grow or shrink on demand



Load Balancer

A *load balancer* distributes traffic to multiple instances.

You can achieve even higher levels of fault tolerance by using your load balancer with instances in multiple Availability Zones.

As instances are launched and terminated, the load balancer automatically directs traffic to the running instances.

Elastic Load Balancing also performs health checks on each instance. If an instance is not responding, the load balancer can automatically redirect traffic to the healthy instances

Amazon S3

Amazon CloudFront

Amazon EBS

Amazon Glacier

AWS Import/Export

AWS Storage Gateway

Storage Option	Usage
Amazon S3	Use Amazon S3 for a wide range of scenarios, from backing up your data, to storing your images and videos (to be accessed directly or through a CDN), to hosting static websites.
Amazon EBS	Use Amazon EBS for data that changes frequently and must persist. For example, use Amazon EBS volumes as the primary storage for a database or file system, or for applications that require access to raw block-level storage.
Instance store volumes	Use instance store volumes for temporary storage of data that changes frequently, such as buffers, caches, or scratch data, or data that is replicated across a fleet of instances. If your data must persist beyond the lifetime of the EC2 instance, use Amazon EBS volumes instead.
Amazon CloudFront	Use CloudFront edge locations to improve the speed of your website. This is especially important if your website displays large media files, such as high-resolution images, audio, or video.
AWS Import/Export	Use AWS Import/Export to transfer data to or from AWS (Amazon S3 buckets, Amazon EBS snapshots, or Amazon Glacier vaults), using portable storage devices. This is a good option if it would be too costly or slow (more than a week) to transfer your data to AWS over the Internet.
AWS Storage Gateway	Use AWS Storage Gateway to provide a seamless and secure connection between an on-premises software appliance and Amazon S3. This is useful for corporate file sharing, enabling existing on-premises backup applications to store primary backups in Amazon S3, and data mirroring.

Amazon Glacier	Use Amazon Glacier when cost is paramount, you need the data infrequently, and you can wait several hours for the data to be retrieved. If you need fast or frequent access to your data, use Amazon S3 instead

CloudFront Distributions

Use Amazon CloudFront to create a content delivery network (CDN) that makes your website content available from data centers around the world, called *edge locations*. You store your content on an *origin server*, such as an Amazon S3 bucket or an HTTP server running on an EC2 instance.

You create a CloudFront *distribution*, associate the distribution with the origin server, and then use a CloudFront URL to access your content:

http://*distribution_id*.cloudfront.net/*file*.*ext*Alternatively, you can associate your own domain name with your CloudFront distribution.When a user accesses an object that's part of a CloudFront distribution, CloudFront checks whether the object is already in a cache that's near the user. If it is, CloudFront serves the content from the cache; otherwise, CloudFront copies the requested content from the origin server to the cache

Amazon RDS

Amazon

DynamoDB

Amazon

ElastiCache

Amazon Redshift

AWS provides fully-managed relational and NoSQL database services, as well as in-memory caching as a service and a petabyte-scale data-warehouse solution

Relational Database

A relational database is organized into tables that are related to each other by key values. It is the traditional type of database.

NoSQL Database

A NoSQL database offers schema flexibility (for example, to provide JSON document model support), fast read and write performance, virtually limitless scaling, and high availability. NoSQL database tables are schemaless and can be used to store JSON-style documents or key-value pairs. This functionality makes NoSQL databases ideal for managing structured or unstructured data.

If your application requires joins or complex transactions, consider a relational database instead. If you have large binary files (audio, video, and image), consider storing the files in Amazon S3 and storing the metadata for the files in your database.

DB Instance

A DB instance is the basic building block of Amazon Relational Database Service (Amazon RDS); it is an isolated database environment in the cloud. A DB instance can contain multiple databases.

When you launch a DB instance, you select a database engine (**MySQL, PostgreSQL, Oracle, or Microsoft SQL Server**) and a DB instance class, which determines the compute and memory capabilities for the DB instance. You also specify a security group for the DB instance.

The firewall for the DB instance prevents any access to its databases other than what you've granted through the rules for the security group. You can launch your DB instance in a virtual private cloud (VPC) for additional

network access control. You can also use the security features of your DB engine the same way that you'd use them on your local network.

In a **Multi-AZ deployment**, Amazon RDS automatically provisions and maintains a synchronous *standby replica* in a different Availability Zone. The primary DB instance is synchronously replicated **across Availability Zones** to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. In the event of a planned or unplanned outage of your DB instance, Amazon RDS automatically switches to a standby replica.

Amazon RDS can use replication functionality built into the PostgreSQL or MySQL DB engine to create a special type of DB instance called a *read replica* from a source DB instance.

Amazon RDS provides two different methods for backing up and restoring your DB instances: **automated backups and user-initiated backups known as *DB snapshots***.

Database Option	Usage
Amazon RDS	Provides a fully-managed relational database that scales to large datasets. It is easy to scale the database storage and compute resources and provide read replicas. You have a choice of database

	engines: MySQL, PostgreSQL, Oracle, or Microsoft SQL Server. Most software designed for use with these databases should work unmodified with Amazon RDS.
	If you need a specific relational database that isn't supported by Amazon RDS, host the database on Amazon EC2 instead.
Amazon Redshift	Provides a fast, fully-managed, petabyte-scale data warehouse that makes it easy and cost-effective to analyze a vast amount of data.
	If you need to perform online transaction processing, use Amazon RDS instead.
Amazon DynamoDB	Provides a fully-managed NoSQL database with fast performance at a low cost. Common use cases include mobile apps, gaming, digital ad serving, live events, metadata storage for Amazon S3 objects, e-commerce shopping carts, and web session management.

Amazon

EMR

Amazon

Kinesis

AWS Data Pipeline

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner

Hadoop is designed for any application which can take advantage of massively parallel distributed-processing, particularly with clusters composed of unreliable hardware.

For example, suppose you have 10 terabytes of data, and you want to process it somehow, (suppose you need to sort it). Using a single computer, this could take a *very* long time. Traditionally, a high-end super-computer with exotic hardware would be required to do this in a reasonable amount of time.

Hadoop provides a framework to process data of this size using a computing cluster made from normal, commodity hardware.

There are two major components to Hadoop: the file system, which is a distributed file system that splits up large files onto multiple computers,

and the MapReduce framework, which is an application framework used to process large data stored on the file system.

Today, the amount of data that companies need to analyze is vast. AWS provides analytics tools that can scale to very large data stores efficiently and cost-effectively.

Amazon Elastic MapReduce (Amazon EMR) uses Hadoop, an open source framework, to manage and process data. Hadoop uses the MapReduce engine to distribute processing using a *cluster*.

Amazon EMR makes it easier to install, configure, and manage Hadoop. You identify the data source, specify the number and type of EC2 instances for the cluster and what software should be on them, and provide a MapReduce program or run interactive queries. Amazon EMR manages the computing resources and runs your MapReduce program or provides tools like Hive or Pig for queries.

AWS Data Pipeline makes it easy for you to regularly move and process data. You create a*pipeline*, which defines the input data source, the compute resources (EMR clusters or EC2 instances) to perform the processing, any conditions that must be met before performing any processing, and the output data location (such as Amazon S3, Amazon Redshift, Amazon RDS, and Amazon DynamoDB).

Amazon Kinesis enables real-time processing of **streaming** data at a massive scale. You can send data from Amazon Kinesis to a data warehouse, such as Amazon Simple Storage Service (Amazon S3) or Amazon **Redshift**, or to an Amazon EMR cluster

Amazon

CloudSearch

Amazon SES

Amazon SNS

Amazon

SQS

Amazon

SWF

Amazon CloudSearch is a managed service in the AWS Cloud that makes it simple and cost-

effective to set up, manage, and scale a search solution for your website or application.

Amazon CloudSearch supports 34 languages and popular search features such as highlighting and autocomplete

With Amazon CloudSearch, you can quickly add rich search capabilities to your website or application. You don't need to become a **search expert** or worry about hardware provisioning, setup, and maintenance. With a few clicks in the AWS Management Console, you can create a search domain and upload the data that you want to make searchable, and Amazon CloudSearch will automatically provision the required resources and deploy a **highly tuned** search index

You can easily change your search parameters, fine tune search relevance, and apply new settings at any time. As your volume of data and **traffic fluctuates**, Amazon CloudSearch seamlessly **scales** to meet your needs

Configuration changes do not require you to re-upload your data Scalable

Reliable

High

Performance

Secure

Amazon Simple Email Service (Amazon SES) is a cost-effective **outbound-only** email-sending service built on the reliable and scalable infrastructure that Amazon.com has developed to serve its own customer base.

With Amazon SES, you can send transactional email, marketing messages, or any other type of high-quality content and **you only pay for what you use**.

Along with high deliverability, Amazon SES provides easy, real-time access to your sending statistics and built-in notifications for bounces, complaints, and deliveries to help you fine-tune your cloud-based email-sending strategy.

Features and Benefits

Simple - Sending email through Amazon SES is as simple as using the SMTP interface Inexpensive

Reliab

le

Scala

ble

Designed for use with other Amazon Web Services

Uses

Marketing

Transactional -Order confirmations, shipping notices, order status updates,password resets

Social Networking - Invitations, posts, status changes

Amazon Simple Notification Service (Amazon SNS) is a fast, flexible, fully managed push notification service that lets you send individual messages or to fan-out messages to large numbers of recipients.

Amazon SNS makes it simple and cost effective to send push notifications to mobile device users, email recipients or even send messages to other distributed services.

With Amazon SNS, you can send notifications to Apple, Google and Windows devices, as well as to Android devices in China.

You can use SNS to **send SMS messages** to mobile device users in the US or to email recipients worldwide.

Beyond these endpoints, Amazon SNS can also deliver messages toAmazon Simple Queue Service (SQS),

You can seamlessly scale from a handful of messages per day to millions of messages or higher

Amazon SNS has no upfront costs and you can pay as you go. It costs $1.00 to send one million notifications

Amazon Simple Queue Service (Amazon SQS) offers a reliable, highly scalable hosted queue for storing messages as they travel between computers.

By using Amazon SQS, developers can simply move data between distributed application components performing different tasks, without losing messages or requiring each component to be always available.

Amazon SQS makes it easy to build an automated workflow, working in close conjunction with the Amazon Elastic Compute Cloud (Amazon EC2) and the other AWS infrastructure web services

Applications using Amazon SQS include key business processes for the Amazon.com retail web site and Amazon Web Services

Amazon SWF helps developers build, run, and scale background jobs that have parallel or sequential steps. You can think of Amazon SWF as a fully-managed state tracker and task coordinator in the Cloud.

If your app's steps take more than 500 milliseconds to complete, you need to track the state of processing, and you need to recover or retry if a task fails, Amazon SWF can help you

Features:

Relia

ble

Simpl

e

Scala

ble

Flexi

ble

Use Case : Video Encoding

Video encoding using Amazon S3 and Amazon EC2. In this use case, large videos are uploaded to Amazon S3 in chunks. The upload of chunks has to be monitored. After a chunk is uploaded, it is encoded by downloading it to an Amazon EC2 instance. The encoded chunk is stored to another Amazon S3 location. After all of the chunks have been encoded in this manner, they are combined into a complete encoded file which is stored back in its entirety to Amazon S3. Failures could occur during this process due to one or more chunks encountering encoding errors. Such failures need to be detected and handled through Amazon SWF's cloud workflow management

AWS CloudFormation

AWS ElasticBeanstalk

Elastic Beanstalk Environments

Elastic Beanstalk enables you to easily move your app from its development environment to AWS infrastructure in minutes. You provide basic configuration information, and Elastic Beanstalk launches an *environment*, which runs a single version of your app, provisions the AWS resources required to support your app, such as EC2 instances, and then deploys your app. After your app is deployed, you can update your environment and the changes are applied automatically.

If you know which AWS resources you want to use and how to configure them, you might prefer to use AWS CloudFormation instead.

AWS CloudFormation Templates and Stacks

AWS CloudFormation enables you to provision the AWS resources that you need predictably and repeatedly

AWS CloudFormation Templates and Stacks

AWS CloudFormation enables you to provision the AWS resources that you need predictably and repeatedly.

First, you create a *template*, a JSON file that defines the required AWS resources, such as EC2 instances and Amazon EBS volumes.

You can start from an existing template, or create one from scratch. You can save your template locally or in Amazon S3. Then you create a *stack*, which manages your resources as a single unit, and AWS CloudFormation provisions the resources described in your template and configures them to work together.

If stack creation fails, AWS CloudFormation **deletes** any resources that were created.

If stack creation succeeds, you can begin to use your resources. If needed, you can edit your template and update the stack based on the modified template. AWS CloudFormation updates the resources as specified

AWS Identity and Access Management AWS CloudTrail

Amazon CloudWatch AWS Directory Service

User Permissions

AWS Identity and Access Management (IAM) enables you to manage users and user permissions in AWS. You can control which users can access which AWS resources.

Directory Services

AWS Directory Service enables you to grant access to AWS to directory users and groups. You can create a new directory in the cloud, or connect to an existing on-premises directory, such as Microsoft Active

Directory.

CloudWatch Metrics

CloudWatch monitors your AWS resources and provides you with data known as *metrics*. You can also configure CloudWatch to send alerts. For example, you can receive an email when your AWS bill reaches a certain amount, or configure Auto Scaling to add or remove EC2 instances from your Auto Scaling group as demand changes.

CloudTrail Logging

CloudTrail captures AWS API calls made for your AWS account by the AWS Management Console, AWS SDKs, command line tools, and AWS deployment and management services, and delivers log files to Amazon S3 so that you can get a history of the calls. The logs contain the identify the users and accounts that made the calls, the source IP addresses the calls were made from, and when the calls occurred.