

Module 3.1

Cloud Infrastructure and Platforms in Industry
Amazon Web Services – Compute Services,
Storage Services, Communication Services,
Additional Services

Public Clouds and Service Offerings

| Table 4.5 Five Major Cloud Platforms and Their Service Offerings [30] | | | | | |
|--|---|---------------------------------------|---------------------------------------|--|---|
| Model | IBM | Amazon | Google | Microsoft | Salesforce |
| PaaS | BlueCloud, WCA, RC2 | | App Engine (GAE) | Windows Azure | Force.com |
| IaaS | Ensembles | AWS | | Windows Azure | |
| SaaS | Lotus Live | | Gmail, Docs | .NET service, Dynamic CRM | Online CRM, Gifttag |
| Virtualization | | OS and Xen | Application Container | OS level/ Hypel-V | |
| Service Offerings | SOA, B2, TSAM, RAD, Web 2.0 | EC2, S3, SQS, SimpleDB | GFS, Chubby, BigTable, MapReduce | Live, SQL Hotmail | Apex, visual force, record security |
| Security Features | WebSphere2 and PowerVM tuned for protection | PKI, VPN, EBS to recover from failure | Chubby locks for security enforcement | Replicated data, rule-based access control | Admin./record security, uses metadata API |
| User Interfaces | | EC2 command-line tools | Web-based admin. console | Windows Azure portal | |
| Web API | Yes | Yes | Yes | Yes | Yes |
| Programming Support | AMI | | Python | .NET Framework | |
| Note: WCA: WebSphere CloudBurst Appliance; RC2: Research Compute Cloud; RAD: Rational Application Developer; SOA: Service-Oriented Architecture; TSAM: Tivoli Service Automation Manager; EC2: Elastic Compute Cloud; S3: Simple Storage Service; SQS: Simple Queue Service; GAE: Google App Engine; AWS: Amazon Web Services; SQL: Structured Query Language; EBS: Elastic Block Store; CRM: Consumer Relationship Management. | | | | | |

Amazon Web Services (AWS)

- VMs can be used to share computing resources both flexibly and safely.
- Amazon has been a leader in providing public cloud services (<http://aws.amazon.com/>).
- Amazon applies the laaS model in providing its services.

AWS Architecture

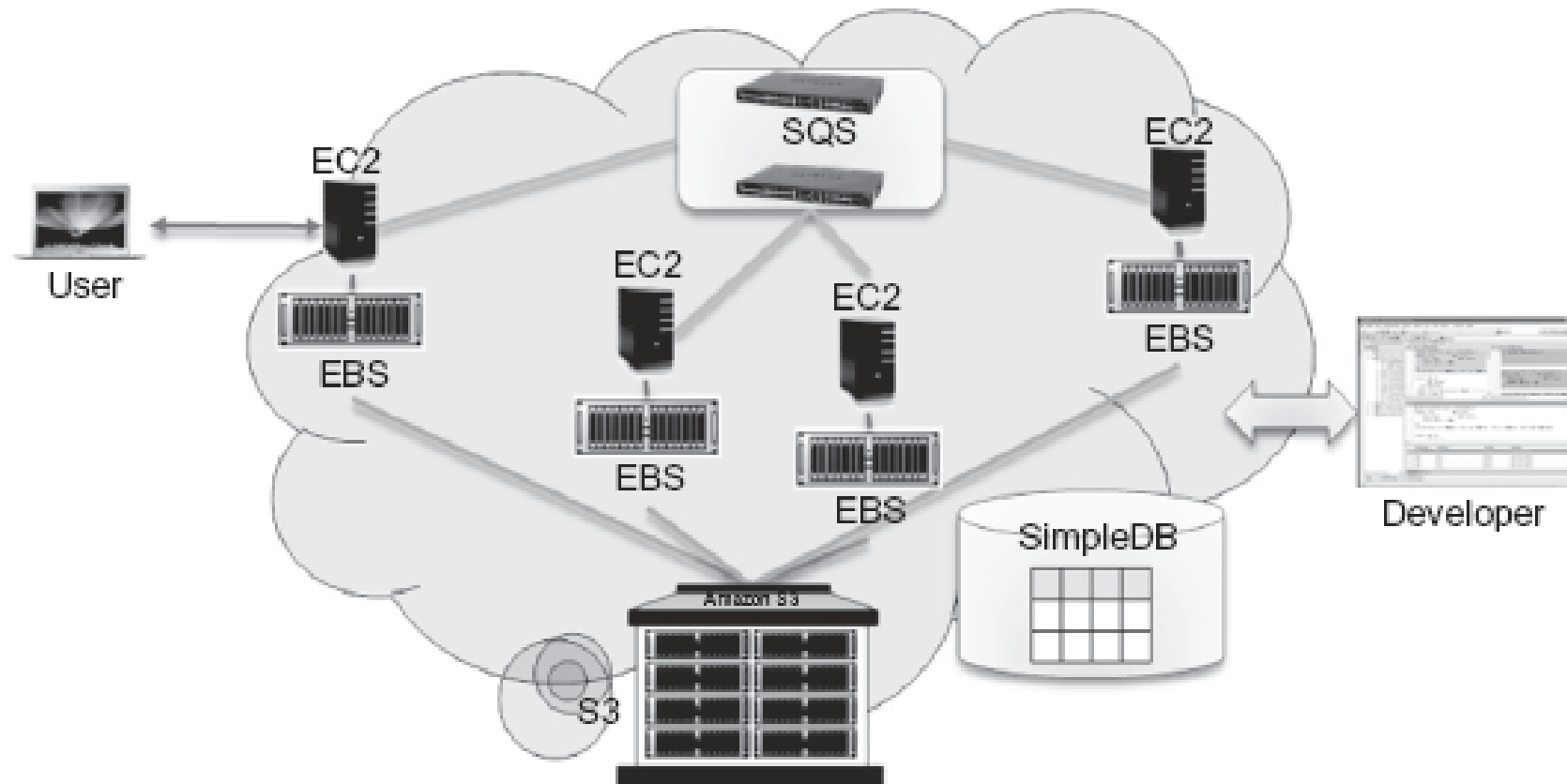


FIGURE 4.21

Amazon cloud computing infrastructure (Key service is identified here; many more are listed in Table 4.5).

(Courtesy of Kang Chen, Tsinghua University, China)

AWS Architecture

The Figure shows the AWS architecture.

- EC2 (Elastic Compute Cloud) provides the virtualized platforms to the host VMs where the cloud application can run.
- S3 (Simple Storage Service) provides the object-oriented storage service for users.
- EBS (Elastic Block Service) provides the block storage interface which can be used to support traditional applications.

AWS Architecture

- SQS stands for Simple Queue Service, and its job is to ensure a reliable message service between two processes. The message can be kept reliably even when the receiver processes are not running.
- Users can access their objects through SOAP with either browsers or other client programs which support the SOAP standard.

AWS Offerings

Compute

Amazon Elastic Compute Cloud (EC2)
Amazon Elastic MapReduce
Auto Scaling

Content Delivery

Amazon CloudFront

Database

Amazon SimpleDB
Amazon Relational Database Service (RDS)

E-Commerce

Amazon Fulfillment Web Service (FWS)

Messaging

Amazon Simple Queue Service (SQS)
Amazon Simple Notification Service (SNS)

Monitoring

Amazon CloudWatch

Networking

Amazon Virtual Private Cloud (VPC)
Elastic Load Balancing

Payments & Billing

Amazon Flexible Payments Service (FPS)
Amazon DevPay

Storage

Amazon Simple Storage Service (S3)
Amazon Elastic Block Storage (EBS)
AWS Import/Export

Support

AWS Premium Support

Web Traffic

Alexa Web Information Service
Alexa Top Sites

Workforce

Amazon Mechanical Turk

Amazon Web Services



FIGURE 9.1

Amazon Web Services ecosystem.

Compute services

- Compute services **constitute the fundamental element of cloud computing systems.**
- The fundamental service in this space is Amazon EC2, which delivers **an IaaS solution that has served as a reference model for several offerings from other vendors in the same market segment.**

Compute services

- Amazon EC2 allows **deploying servers in the form of virtual machines created as instances of a specific image.**
- **Images come with a preinstalled operating system and a software stack, and instances can be configured for memory, number of processors, and storage.**
- **Users are provided with credentials to remotely access the instance and further configure or install software if needed**

Amazon machine images

- Amazon Machine Images (AMIs) are **templates from which it is possible to create a virtual machine.**
- They are stored in Amazon Simple Storage Service S3 and
- Identified by a unique identifier in the form of ami-xxxxxx and a manifest XML file.

Amazon machine images

- An AMI contains a physical file system layout with a **predefined operating system installed.**
- These are specified by the
 - 1) **Amazon Ramdisk Image (ARI, id: ari-yyyyyy) and**
 - 2) **Amazon Kernel Image (AKI, id: aki-zzzzzz),**
- **which are part of the configuration of the template.**
- AMIs are either created from scratch or “bundled” from existing EC2 instances.

Amazon machine images

Prepare a new AMIs?

- 1) To create an instance from a preexisting AMI,
- 2) Log into it once it is booted and running, and install all the software needed, customize the instance.
- 3) Then save this updated configuration as a new custom AMI.
- 4) Instances launched from this new custom AMI include the customizations that you made when you created the AMI.
- 5) Once an AMI is created, it is stored in an S3 bucket

Amazon machine images

- **The user can decide whether to make it available to other users or keep it for personal use.**

Amazon machine images-Revenue Generation

- Finally, it is also possible to **associate a product code with a given AMI, thus allowing the owner of the AMI to get revenue every time this AMI is used to create EC2 instances**

Buy, share, and sell AMIs

- After you create an AMI, you can keep it private so that only you can use it, or
- You can share it with a specified list of AWS accounts.
- You can also make your custom AMI public so that the community can use it.
- You can also create an AMI and sell it to other Amazon EC2 users.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>

EC2 instances

- **EC2 instances represent virtual machines.**
- **They are created using AMI as templates**, which are specialized by selecting the number of cores, their computing power, and the installed memory.
- **The processing power is expressed in terms of virtual cores and EC2 Compute Units (ECUs).**

EC2

- **The ECU is a measure of the computing power of a virtual core;** it is used to express a predictable quantity of real CPU power that is allocated to an instance.
- By using compute units instead of real frequency values, **Amazon can change over time the mapping of such units to the underlying real amount of computing power allocated,** thus keeping the performance of EC2 instances consistent with standards set by the times.

EC2

- We can identify six major categories:
 - 1) **Standard instances**
 - 2) **Micro instances**
 - 3) **High-memory instances**
 - 4) **High-CPU instances**
 - 5) **Cluster Compute instances**
 - 6) **Cluster GPU instances**

EC2

Standard instances-

- **This class offers a set of configurations that are suitable for most applications.**
- EC2 provides **three different categories** of increasing computing power, storage, and memory.

EC2

Micro instances-

- This class is suitable for those applications that consume a **limited amount of computing power and memory** and **occasionally need** bursts in CPU cycles to **process surges in the workload**.
- Micro instances can be used for **small Web applications with limited traffic**.

EC2

High-memory instances

- This class targets applications that need to **process huge workloads and require large amounts of memory.**
- Three-tier **Web applications characterized by high traffic** are the target profile.
- **Three categories of increasing memory and CPU** are available, **with memory proportionally larger than computing power.**

EC2

High-CPU instances-

- This class targets **compute-intensive applications**.
- **Two configurations are available** where **computing power proportionally increases more than memory**.

EC2

Cluster Compute instances-

- This class is **used to provide virtual cluster services.**
- Instances in this category are characterized **by high CPU compute power and large memory and an extremely high I/O and network performance, which makes it suitable for HPC applications.**

EC2

Cluster GPU instances-

- This class provides **instances featuring graphic processing units (GPUs)** and high compute power, large memory, and extremely high I/O and network performance.
- This class is particularly **suited for cluster applications that perform heavy graphic computations**, such as rendering clusters.
- **Since GPU can be used for general-purpose computing, users of such instances can benefit from additional computing power, which makes this class suitable for HPC applications.**

EC2

- EC2 instances are **priced hourly according to the category they belong to.**
- At the beginning of every hour of usage, **the user will be charged the cost of the entire hour.**
- The hourly expense charged for one instance is constant.

EC2

- EC2 instances can be **run either by**
- **using the command-line tools provided by Amazon**, which connects the Amazon Web Service that provides remote access to the EC2 infrastructure, or
- **via the AWS console, which allows the management of other services, such as S3.**

Table 9.2 Amazon EC2 (On-Demand) Instances Characteristics

| Instance Type | ECU | Platform | Memory | Disk Storage | Price (U.S. East) (USD/hour) |
|-----------------------|------------------------|-----------|---------|--------------|------------------------------|
| Standard instances | | | | | |
| Small | 1(1×1) | 32 bit | 1.7 GB | 160 GB | \$0.085 Linux \$0.12 Windows |
| Large | 4(2×2) | 64 bit | 7.5 GB | 850 GB | \$0.340 Linux \$0.48 Windows |
| Extra Large | 8(4×2) | 64 bit | 15 GB | 1,690 GB | \$0.680 Linux \$0.96 Windows |
| Micro instances | | | | | |
| Micro | ≤ 2 | 32/64 bit | 613 MB | EBS Only | \$0.020 Linux \$0.03 Windows |
| High-Memory instances | | | | | |
| Extra Large | 6.5(2×3.25) | 64 bit | 17.1 GB | 420 GB | \$0.500 Linux \$0.62 Windows |
| Double Extra Large | 13(4×3.25) | 64 bit | 34.2 GB | 850 GB | \$1.000 Linux \$1.24 Windows |
| Quadruple Extra Large | 26(8×3.25) | 64 bit | 68.4 GB | 1,690 GB | \$2.000 Linux \$2.48 Windows |
| High-CPU instances | | | | | |
| Medium | 5(2×2.5) | 32 bit | 1.7 GB | 350 GB | \$0.170 Linux \$0.29 Windows |
| Extra Large | 20(8×2.5) | 64 bit | 7 GB | 1,690 GB | \$0.680 Linux \$1.16 Windows |
| Cluster instances | | | | | |
| Quadruple Extra Large | 33.5 | 64 bit | 23 GB | 1,690 GB | \$1.600 Linux \$1.98 Windows |
| Cluster GPU instances | | | | | |
| Quadruple Extra Large | 33.5 | 64 bit | 22 GB | 1,690 GB | \$2.100 Linux \$2.60 Windows |

EC2 environment

- EC2 instances are executed within a virtual environment, which **provides them with the services** they require to host applications.
- **The EC2 environment is in charge of**
- **allocating addresses,**
- **attaching storage volumes, and**
- **configuring security** in terms of **access control and network connectivity**

EC2 environment

- By default, instances are created with **an internal IP address, which makes them capable of communicating within the EC2 network and accessing the Internet as clients.**

EC2 environment

- It is possible to **associate an Elastic IP to each instance, which can then be remapped to a different instance over time.**
- Elastic IPs allow instances running in EC2 to act as servers reachable from the Internet and,
- since they are **not strictly bound to specific instances, to implement failover capabilities.**

- The auto-assigned public IP address associated with my Amazon Elastic Compute Cloud (Amazon EC2) instance changes every time I stop and start the instance.

How can I assign a static public IP address to my Windows or Linux EC2 instance that doesn't change when I stop/start the instance?

Solution-

Short description

- An Elastic IP address is a static public IPv4 address associated with your AWS account in a specific Region.
- Unlike an auto-assigned public IP address, **an Elastic IP address is preserved after you stop and start your instance in a virtual private cloud (VPC).**

EC2 environment

- Together with an external IP,
- **EC2 instances are also given a domain name** that generally is in the form
- **ec2-xxxxxx-xxx.compute-x.amazonaws.com,**
- where
- xxx-xxx-xxx =four parts of the external IP address separated by a dash,
- compute-x = information about the availability zone where instances are deployed.

EC2 environment

- Currently, there are five availability zones that are priced differently:
- **two in the United States (Virginia and Northern California),**
- **one in Europe (Ireland), and**
- **two in Asia Pacific (Singapore and Tokyo).**

EC2 environment

- **Instance owners can partially control where to deploy instances.**
- Instead, they have a finer control over the security of the instances as well as their network accessibility.
- **Instance owners can associate a key pair to one or more instances when these instances are created.**
- **A key pair allows the owner to remotely connect to the instance once this is running and gain root access to it.**

EC2 environment

- **Amazon EC2 controls the accessibility of a virtual instance with basic firewall configuration**, allowing the specification of source address, port, and protocols (TCP, UDP, ICMP).
- **Rules can also be attached to security groups, and instances can be made part of one or more groups before their deployment.**
- **Security groups and firewall rules constitute a flexible way of providing basic security for EC2 instances**, which has to be complemented by appropriate security configuration within the instance itself.

Amazon elastic MapReduce

- Amazon Elastic MapReduce provides AWS users with a **cloud computing platform for MapReduce applications**.
- It utilizes **Hadoop as the MapReduce engine, deployed on a virtual infrastructure composed of EC2 instances**, and uses Amazon S3 for storage needs.

END of Lecture

Storage services AWS

- Storage services AWS provides **a collection of services for data storage**
- **The core service in this area is represented by Amazon Simple Storage Service (S3).**

Amazon S3

- S3 is a distributed object store that allows users to store information in different formats.

Compute

Amazon Elastic Compute Cloud (EC2)
Amazon Elastic MapReduce
Auto Scaling

Content Delivery

Amazon CloudFront

Database

Amazon SimpleDB
Amazon Relational Database Service (RDS)

E-Commerce

Amazon Fulfillment Web Service (FWS)

Messaging

Amazon Simple Queue Service (SQS)
Amazon Simple Notification Service (SNS)

Monitoring

Amazon CloudWatch

Networking

Amazon Virtual Private Cloud (VPC)
Elastic Load Balancing

Payments & Billing

Amazon Flexible Payments Service (FPS)
Amazon DevPay

Storage

Amazon Simple Storage Service (S3)
Amazon Elastic Block Storage (EBS)
AWS Import/Export

Support

AWS Premium Support

Web Traffic

Alexa Web Information Service
Alexa Top Sites

Workforce

Amazon Mechanical Turk

S3 key concepts

- As the name suggests, S3 has been designed to provide a simple storage service that's **accessible through a Representational State Transfer (REST) interface**, which is quite similar to a distributed file system but which presents some important differences that **allow the infrastructure to be highly efficient**:
 - 1) **The storage is organized in a two-level hierarchy**
 - 2) **Stored objects cannot be manipulated like standard files.**
 - 3) **Content is not immediately available to users**
 - 4) **Requests will occasionally fail.**

Storage Service (S3).

The storage is organized in a two-level hierarchy-

- **Buckets & Objects**
- **S3 organizes its storage space into buckets that cannot be further partitioned.**
- **This means that it is not possible to create directories or other kinds of physical groupings for objects stored in a bucket.**

Storage Service (S3).

Stored objects cannot be manipulated like standard files-

- S3 has been designed to essentially provide storage for **objects that will not change over time.**
- Therefore, **it does not allow**
- **renaming,**
- **modifying, or**
- **relocating an object.**

Storage Service (S3).

Stored objects cannot be manipulated like standard files-

- Once an object has been added to a bucket,
- **its content and position is immutable**, and
- the only way to change it is **to remove the object from the store and add it again.**

Storage Service (S3).

Content is not immediately available to users.

- The main design goal of S3 is to provide an eventually **consistent data store**. As a result, because it is a large distributed storage facility, changes are not immediately reflected.
- For instance, S3 uses **replication to provide redundancy** and efficiently serve objects across the globe; this practice introduces **latencies when adding objects to the store—especially large ones—which are not available instantly across the entire globe**

Requests will occasionally fail

- Due to the large distributed infrastructure being managed,
- **requests for object may occasionally fail.**
- Under certain conditions, **S3 can decide to drop a request by returning an internal server error.**
- Therefore, it is expected to have **a small failure rate during day-to-day operations**, which is generally **not identified as a persistent failure.**

Simple Storage Service (S3).

The core components of S3 are two:

- **Buckets**
- **Objects.**

Simple Storage Service (S3).

- Buckets represent virtual containers in which to store objects;
- Objects represent the content that is actually stored.
- Objects can also be enriched with metadata that can be used to tag the stored content with additional information.

Buckets

- A bucket is a container of objects.
- It can be thought of as a virtual drive hosted on the S3 distributed storage, **which provides users with a flat store to which they can add objects.**

Buckets

- Buckets are top level elements of the S3 storage architecture and **do not support nesting**.
- That is, or other kinds of physical divisions.

Buckets

- A bucket is **located in a specific geographic location and eventually replicated for fault tolerance** and better content distribution.
- Users **can select the location at which to create buckets,**
- **By default buckets are created in Amazon's U.S. datacenters.**
- Once a bucket is created, **all the objects that belong to the bucket will be stored in the same availability zone of the bucket.**

Buckets

- Buckets, objects, and attached metadata are made accessible through a REST interface.
- Therefore, they are **represented by uniform resource identifiers (URIs) under the s3.amazonaws.com domain.**
- All the operations are then performed **by expressing the entity they are directed to in the form of a request for a URI.**
- One way of addressing a bucket:
- Canonical form: **http://s3.amazonaws.com/bucket_name/.**
The bucket name is expressed as a path component of the domain name s3.amazonaws.com.

Buckets

- **Users create a bucket by sending a PUT request to `http://s3.amazonaws.com/` with**
- **the name of the bucket and,**
- **if they want to specify the availability zone,**
- **additional information about the preferred location.**

Buckets

- **The content of a bucket can be listed by sending a GET request specifying the name of the bucket.**

Buckets

- Once created, the bucket cannot be renamed or relocated.
- If it is necessary to do so, **the bucket needs to be deleted and recreated.**
- The deletion of a bucket is performed by a DELETE request, **which can be successful if and only if the bucket is empty**

Objects

- Objects constitute **the content elements stored in S3.**
- An object is **identified by a name that needs to be unique within the bucket** in which the content is stored.
- **Users either store files or push to the S3 text stream** representing the object's content.

Objects

- **The name cannot be longer than 1,024 bytes when encoded in UTF-8, and it allows almost any character.**
- **Since buckets do not support nesting, even characters normally used as path separators are allowed.**
- **This actually compensates for the lack of a structured file system, since directories can be emulated by properly naming objects.**

Objects

- **Users create an object via a PUT request that specifies**
- **the name of the object together**
- **with the bucket name,**
- **its contents, and**
- **additional properties.**

Objects

- The maximum size of an object is 5 GB.
- Once an object is created, **it cannot be**
- **modified,**
- **renamed, or**
- **moved into another bucket.**

Objects

- It is possible to **retrieve an object via a GET request**;
- Deleting an object is performed via a **DELETE request**.

Metadata

- **Objects can be tagged with metadata, which are passed as properties of the PUT request.**
- Meta Data can be retrieved either
- with a GET request or
- with a HEAD request, **which only returns the object's metadata without the content.**

Access control and security of Buckets and Objects

- **Amazon S3 allows controlling the access to buckets and objects by means of Access Control Policies (ACPs).**
- **An ACP is a set of grant permissions that are attached to a resource expressed by means of an XML configuration file.**

Access control and security of Buckets and Objects

- Currently, five different permissions can be used:
 - 1) READ allows the grantee to **retrieve an object and its metadata** and to list the content of a bucket as well as getting its metadata.
 - 2) WRITE allows the grantee **to add an object to a bucket as well remove it.**
 - 3) READ_ACP allows the grantee **to read the ACP of a resource.**
 - 4) WRITE_ACP allows the grantee to **modify the ACP of a resource.**
 - 5) FULL_CONTROL grants **all of the preceding permissions.**

Amazon elastic block store

- The Amazon Elastic Block Store (EBS) allows AWS users to **provide EC2 instances with persistent storage**
- **in the form of volumes that can be mounted at instance startup.**
- They **accommodate up to 1 TB of space and are accessed through a block device interface**, thus allowing users to format them according to the needs of the instance they are connected to (raw storage, file system, or other).

Amazon elastic block store

- The content of an **EBS volume survives the instance life cycle and is persisted into S3.**
- EBS volumes can be cloned, used as boot partitions, and constitute durable storage since they rely on S3 and it is possible to take incremental snapshots of their content.

S3 vs EBS

Accessibility

- EBS is specifically meant for EC2 (Elastic Computing Cloud) instances and is not accessible unless mounted to one.
- The data stored in EBS is **only accessible by the instance to which it is connected to.**

VS

- The files within an S3 bucket are stored in an unstructured manner and **can be retrieved using HTTP protocols and even with BitTorrent.**
- Many sites use S3 to hold most of their files **because of its accessibility to HTTP clients;** web browsers for example.

Storage type

- Amazon Simple Storage Service is **object storage designed for storing large numbers of user files and backups** whereas

VS

- Elastic block storage is **block storage for Amazon EC2** compute instances
- It is just **similar to hard disks attached to your computers or laptops, but the only difference is that it is used for virtualized instances.**

Storage type

- With EBS, a volume can be mounted on an EC2 instance and it would appear just like a hard disk partition.
- It can be formatted with any file system and files can be written or read by the EC2 instance just like it would to a hard drive.

Size of data

- Simple storage service (S3) can store large amounts as compared to EBS.
- With S3, **the standard limit is of 100 buckets and each bucket has got an unlimited data capacity** whereas

Vs

- EBS has a **standard limit of 20 volumes and each volume can hold data up to 1TB.**
- In EBS there occurs an upper limit on the data storage.

Limitation of EBS

EBS

- **EBS Volume images cannot be shared among instances,**
- **But it is possible to attach multiple volumes to a single instance**
- **A limitation of EBS is its inability to be used by multiple instances at once. Once it is mounted by an instance, no other instance can use it.**

Pros + Cons of S3

- With EBS, data read or write occurs almost instantly.
- **S3 can have multiple images of its contents so it can be used by many at the same time.**
- An interesting side-effect of this capability is something called **‘eventual consistency’**.
- With S3, the changes are not written immediately so if you write something, it may not be the data that a read operation returns.

End of Lecture

Application

Amazon S3:

- Amazon S3 is a simple storage service offered by Amazon and it is useful for hosting website images and videos, data analytics, etc

EBS

- Use cases include business continuity, transactional and NO SQL database, software testing, etc.

Application

Amazon S3:

- **Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data**
- S3 can be used for a range of use cases, such as
 - websites,
 - mobile applications,
 - backup and restore,
 - archive,
 - enterprise applications,
 - IoT devices, and
 - big data analytics.

Application

Amazon S3:

- Amazon S3 provides management features so that you can **optimize, organize, and configure access to your data** to meet **your specific business, organizational, and compliance requirements**.

Security

- Amazon S3 can **prevent unauthorized accessing of data using**
- **its access management tools and**
- **encryption policies**

VS

- **No such feature is present in EBS.**
- **In EBS, if any user gets unauthorized access to the instance then he/she can easily access the attached EBS.**

Amazon ElastiCache

- ElastiCache is based on a **cluster of EC2 instances running the caching software**, which is made available through Web services.
- ElastiCache is an **implementation of an elastic in-memory cache based on a cluster of EC2 instances**.

Amazon ElastiCache

- Automatic **patch management and failure detection and recovery of cache nodes** allow the cache cluster to keep running
 - **without administrative intervention from AWS users,**
 - **AWS users only have to elastically size the cluster when needed**
-
- An ElastiCache cluster **can be dynamically resized according to the demand of the client applications.**

Amazon ElastiCache Pricing

- ElastiCache nodes are priced according to the **EC2 costing model**,
- **With a small price difference due to the use of the caching service installed on such instances.**

Amazon EC2 Cache Instances Pricing

- It is possible to choose between different types of instances;

Table 9.3 Amazon EC2 (On-Demand) Cache Instances Characteristics, 2011–2012

| Instance Type | ECU | Platform | Memory | I/O Capacity | Price (U.S. East) (USD/hour) |
|-----------------------|------------------------|----------|---------|--------------|---------------------------------|
| Standard instances | | | | | |
| Small | 1(1×1) | 64 bit | 1.3 GB | Moderate | \$0.095 |
| Large | 4(2×2) | 64 bit | 7.1 GB | High | \$0.380 |
| Extra Large | 8(4×2) | 64 bit | 14.6 GB | High | \$0.760 |
| High-Memory instances | | | | | |
| Extra Large | 6.5(2×3.25) | 64 bit | 16.7 GB | High | \$0.560 |
| Double Extra Large | 13(4×3.25) | 64 bit | 33.8 GB | High | \$1.120 |
| Quadruple Extra Large | 26(8×3.25) | 64 bit | 68 GB | High | \$2.240 |
| High-CPU instances | | | | | |
| Extra Large | 26(8×3.25) | 64 bit | 6.6 GB | High | \$0.760 |

Amazon EC2 Instances Pricing

Table 9.2 Amazon EC2 (On-Demand) Instances Characteristics

| Instance Type | ECU | Platform | Memory | Disk Storage | Price (U.S. East) (USD/hour) |
|-----------------------|------------------------|-----------|---------|--------------|------------------------------|
| Standard instances | | | | | |
| Small | 1(1×1) | 32 bit | 1.7 GB | 160 GB | \$0.085 Linux \$0.12 Windows |
| Large | 4(2×2) | 64 bit | 7.5 GB | 850 GB | \$0.340 Linux \$0.48 Windows |
| Extra Large | 8(4×2) | 64 bit | 15 GB | 1,690 GB | \$0.680 Linux \$0.96 Windows |
| Micro instances | | | | | |
| Micro | ≤ 2 | 32/64 bit | 613 MB | EBS Only | \$0.020 Linux \$0.03 Windows |
| High-Memory instances | | | | | |
| Extra Large | 6.5(2×3.25) | 64 bit | 17.1 GB | 420 GB | \$0.500 Linux \$0.62 Windows |
| Double Extra Large | 13(4×3.25) | 64 bit | 34.2 GB | 850 GB | \$1.000 Linux \$1.24 Windows |
| Quadruple Extra Large | 26(8×3.25) | 64 bit | 68.4 GB | 1,690 GB | \$2.000 Linux \$2.48 Windows |
| High-CPU instances | | | | | |
| Medium | 5(2×2.5) | 32 bit | 1.7 GB | 350 GB | \$0.170 Linux \$0.29 Windows |
| Extra Large | 20(8×2.5) | 64 bit | 7 GB | 1,690 GB | \$0.680 Linux \$1.16 Windows |
| Cluster instances | | | | | |
| Quadruple Extra Large | 33.5 | 64 bit | 23 GB | 1,690 GB | \$1.600 Linux \$1.98 Windows |
| Cluster GPU instances | | | | | |
| Quadruple Extra Large | 33.5 | 64 bit | 22 GB | 1,690 GB | \$2.100 Linux \$2.60 Windows |

Structured storage solutions

- Enterprise applications quite often rely **on databases to store data in a structured form**, index, and perform analytics against it.
- Traditionally, **RDBMS have been the common data back-end for a wide range of applications.**

Structured storage solutions

- Amazon provides applications with structured storage services in three different forms:
 - **Preconfigured EC2 AMIs,**
 - **Amazon Relational Data Storage (RDS), and**
 - **Amazon SimpleDB**

Preconfigured EC2 AMIs-

- Preconfigured EC2 AMIs are **predefined templates featuring an installation of a given database management system.**
- EC2 instances created from **these AMIs can be completed with an EBS volume for storage persistence.**

Preconfigured EC2 AMIs-

- Available AMIs include installations of
- IBM DB2,
- Microsoft SQL Server,
- MySQL,
- Oracle,
- PostgreSQL,
- Sybase, and
- Vertica.
- Instances are **priced hourly according to the EC2 cost model.**

Preconfigured EC2 AMIs-

- This solution poses most of the administrative burden on the **EC2 user, who has to configure, maintain, and manage the relational database,**
- **but offers the greatest variety of products to choose from.**

Amazon RDS

- RDS is **relational database service that relies on the EC2 infrastructure and is managed by Amazon.**
- Developers **do not have to worry about configuring the storage for high availability, designing failover strategies, or keeping the servers up-to-date with patches.**

Amazon RDS

- With respect to the previous solution, **users are not responsible for managing, configuring, and patching the database management software**, but these operations are performed by the AWS.
- Moreover, the service **provides users with automatic backups, snapshots, point-in-time recoveries, and facilities for implementing replications.**
- These and the common database management services are available through the AWS console or a specific Web service.

- **Amazon RDS is a managed database service. It's responsible for most management tasks.**
- By eliminating tedious manual tasks, Amazon RDS frees you to focus on your application and your users.
- **Amazon recommends Amazon RDS over Amazon EC2 as your default choice for most database deployments.**

-Amazon AWS

Preconfigures EC2 AMIs vs RDS

| Feature | Amazon EC2 management | Amazon RDS management |
|-----------------------------|-----------------------|-----------------------|
| Application optimization | Customer | Customer |
| Scaling | Customer | AWS |
| High availability | Customer | AWS |
| Database backups | Customer | AWS |
| Database software patching | Customer | AWS |
| Database software install | Customer | AWS |
| OS patching | Customer | AWS |
| OS installation | Customer | AWS |
| Server maintenance | AWS | AWS |
| Hardware lifecycle | AWS | AWS |
| Power, network, and cooling | AWS | AWS |

Advantages of RDS

- Amazon RDS provides the following specific advantages over database deployments that aren't fully managed:
- You can use the database products you are already familiar with:
 - **MariaDB,**
 - **Microsoft SQL Server,**
 - **MySQL,**
 - **Oracle, and**
 - **PostgreSQL.**

-Amazon AWS

Advantages of RDS

- Amazon RDS manages backups, software patching, automatic failure detection, and recovery.
- You can **turn on automated backups, or manually create your own backup snapshots.**
- You can use these backups to restore a database.
- The Amazon **RDS restore process works reliably and efficiently.**

-Amazon AWS

Advantages of RDS

- Users can get **high availability with a primary instance and a synchronous secondary instance** that you can fail over to when problems occur.
- Users can also **use read replicas to increase read scaling.**

-Amazon AWS

Advantages of RDS

- In addition to the security in your database package, **you can help control who can access your RDS databases.**
- To do so, you can use **AWS Identity and Access Management (IAM)** to define users and permissions.
- You can also **help protect your databases by putting them in a virtual private cloud (VPC).**

-Amazon AWS

DB instances

- The **basic building block of Amazon RDS**
- An **isolated database environment in the AWS Cloud.**
- User's DB instance can **contain one or more user-created databases.**

-Amazon AWS

DB instances

- You can access your DB instance by using the same tools and applications that you use with a standalone database instance.
- You can **create and modify a DB instance by**
- **using the AWS Command Line Interface (AWS CLI),**
- **the Amazon RDS API, or**
- **the AWS Management Console.**

-Amazon AWS

DB engines

- A *DB engine* is the **specific relational database software that runs on your DB instance.**

-Amazon AWS

DB engines

- **Amazon RDS currently supports the following engines:**
 - **MariaDB**
 - **Microsoft SQL Server**
 - **MySQL**
 - **Oracle**
 - **PostgreSQL**

-Amazon AWS

DB engines

- Each DB engine has its own supported features, and each **version of a DB engine can include specific features.**
- Additionally, **each DB engine has a set of parameters in a DB parameter group**
- **that control the behavior of the databases that it manages.**

-Amazon AWS

DB instance types

- Amazon RDS supports DB instance types as follows:
- **General-purpose**
- **Memory-optimized**
- **Burstable-performance**
- **Optimized Reads**

DB instance classes

- The DB instance class **determines the computation and memory capacity of an Amazon RDS DB instance type.**
- The DB instance class that you need **depends on your processing power and memory requirements.**

-Amazon AWS

DB instance classes

- A DB instance class consists of both the DB instance type and the size.

Eg1-

- **db.m6g** is a general-purpose DB instance type powered by AWS Graviton2 processors.
- Within the **db.m6g** instance type, **db.m6g.2xlarge** is a DB instance class.

Eg2-

- **db.r6g** is a memory-optimized DB instance class type powered by AWS Graviton2 processors.
- Within the **db.r6g** instance class type, **db.r6g.2xlarge** is a DB instance class. The size of this class is 2xlarge.

-Amazon AWS

General-purposeDB instance types

- The following are the general-purpose DB instance classes are available:
- **db.m7g**
- **db.m6g**
- **db.m6i**
- **db.m5**
- **db.m4**
- **db.m3**

Memory Optimized DB instance types

- The following are the memory-optimized DB instance types available:
 - **db.x2g**
 - **db.z1d**
 - **db.x2i**
 - **db.x1e**
 - **db.x1**
 - **db.r6g**
 - **db.r6gd**
 - **db.r6i**
 - **db.r5b**
 - **db.r5d**
 - **db.r5**
 - **db.r4**
 - **db.r3**

Burstable Performance DB instance types

- The following are the burstable-performance DB instance types available:
- **db.t4g**
- **db.t3**
- **db.t2**

-Amazon AWS

Supported DB engines for DB instance classes

| Instance class | MariaDB | Microsoft SQL Server | MySQL | Oracle | PostgreSQL |
|---|---|----------------------|-------------------------|--------|--|
| db.m6g – general-purpose instance classes powered by AWS Graviton2 processors | | | | | |
| db.m6g.16xlarge | All MariaDB 10.6 versions, all MariaDB 10.5 versions, and all MariaDB 10.4 versions | No | MySQL 8.0.23 and higher | No | All PostgreSQL 15 versions, all PostgreSQL 14 versions, all PostgreSQL 13 versions, PostgreSQL 12.7 and higher |
| db.m6g.12xlarge | All MariaDB 10.6 versions, all MariaDB 10.5 versions, and all MariaDB 10.4 versions | No | MySQL 8.0.23 and higher | No | All PostgreSQL 15 versions, all PostgreSQL 14 versions, all PostgreSQL 13 versions, PostgreSQL 12.7 and higher |
| db.m6g.8xlarge | All MariaDB 10.6 versions, all MariaDB 10.5 versions, and all MariaDB 10.4 versions | No | MySQL 8.0.23 and higher | No | All PostgreSQL 15 versions, all PostgreSQL 14 versions, all PostgreSQL 13 versions, PostgreSQL 12.7 and higher |

Supported DB engines for DB instance classes

| Instance class | MariaDB | Microsoft SQL Server | MySQL | Oracle | PostgreSQL |
|---|---|----------------------|-------------------------|--------|--|
| db.m6g – general-purpose instance classes powered by AWS Graviton2 processors | | | | | |
| db.m6g.4xlarge | All MariaDB 10.6 versions, all MariaDB 10.5 versions, and all MariaDB 10.4 versions | No | MySQL 8.0.23 and higher | No | All PostgreSQL 15 versions, all PostgreSQL 14 versions, all PostgreSQL 13 versions, PostgreSQL 12.7 and higher |
| db.m6g.2xlarge | All MariaDB 10.6 versions, all MariaDB 10.5 versions, and all MariaDB 10.4 versions | No | MySQL 8.0.23 and higher | No | All PostgreSQL 15 versions, all PostgreSQL 14 versions, all PostgreSQL 13 versions, PostgreSQL 12.7 and higher |
| db.m6g.xlarge | All MariaDB 10.6 versions, all MariaDB 10.5 versions, and all MariaDB 10.4 versions | No | MySQL 8.0.23 and higher | No | All PostgreSQL 15 versions, all PostgreSQL 14 versions, all PostgreSQL 13 versions, PostgreSQL 12.7 and higher |
| db.m6g.large | All MariaDB 10.6 versions, all MariaDB 10.5 versions, and all MariaDB 10.4 versions | No | MySQL 8.0.23 and higher | No | All PostgreSQL 15 versions, all PostgreSQL 14 versions, all PostgreSQL 13 versions, PostgreSQL 12.7 and higher |

Amazon DynamoDB

- Amazon DynamoDB is a **fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.**

Amazon DynamoDB

- DynamoDB lets you **offload the administrative burdens** of operating and scaling a distributed database so that you **don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.**
- DynamoDB also offers **encryption** , which eliminates the **operational burden and complexity** involved in protecting sensitive data

Amazon SimpleDB

- Amazon SimpleDB is a lightweight, highly scalable, and flexible data storage solution for applications **that do not require a fully relational model for their data.**
- SimpleDB provides support for semistructured data, the model for which is based on the concept of domains, items, and attributes.
- **With respect to the relational model, this model provides fewer constraints on the structure of data entries,** thus obtaining improved performance in querying large quantities of data.

Amazon SimpleDB

- Amazon SimpleDB is a **highly available NoSQL data store that offloads the work of database administration.**
- Developers simply store and query data items via web services requests and Amazon SimpleDB does the rest.

<https://aws.amazon.com/simplifiedb/>

Amazon SimpleDB

- The page is still up, and AWS is obviously still running the service (and will as long as customers keep using it), but it's not linked from anywhere. Here's a list of all the AWS services (<https://aws.amazon.com/products/>) and it's not listed.
- If you look at the developer guide, it hasn't been updated in three years.
- It may not be officially deprecated, but a service that hasn't been touched in three years, which AWS will only tell you about if asked very specifically.

- End of Lecture

Communication services

- Amazon provides **facilities to structure and facilitate the communication among existing applications and services** residing within the AWS infrastructure.
- These facilities can be organized into two major categories:
 - **Virtual networking and**
 - **Messaging.**

Communication services

Virtual networking-

- Virtual networking comprises **a collection of services that allow AWS users to control the connectivity to and between compute and storage services.**

Communication services

Virtual networking-

- **Amazon Virtual Private Cloud (VPC) and Amazon Direct Connect** provide connectivity solutions in terms of infrastructure;
- **Route 53** facilitates connectivity in terms of naming

Amazon VPC

- Amazon VPC provides a great degree of flexibility in **creating virtual private networks within the Amazon infrastructure and beyond.**

Amazon VPC

- The service **providers prepare either templates** covering most of the usual scenarios **or a fully customizable network service** for advanced configurations.
- **Prepared templates include**
 - **public subnets,**
 - **isolated networks,**
 - **private networks accessing Internet through network address translation (NAT), and**
 - **hybrid networks including AWS resources and private resources.**

Amazon VPC

- Also, it is possible to **control connectivity between different services (EC2 instances and S3 buckets) by using the Identity Access Management (IAM) service.**

Amazon Direct Connect

- Amazon Direct Connect **allows AWS users to create dedicated networks between the user private network and Amazon Direct Connect locations, called ports.**
- The advantage of using Direct Connect versus other solutions is the **consistent performance of the connection between the users' premises and the Direct Connect locations.**

Amazon Direct Connect

- This service is **compatible with other services such as EC2, S3, and Amazon VPC**
- **It can be used in scenarios requiring high bandwidth between the Amazon network and the outside world.**

Amazon Route 53

- Amazon Route 53 implements **dynamic DNS services that allow AWS resources to be reached through domain names different from the amazon.com domain.**

Communication services- Messaging

- Messaging services constitute the next step in connecting applications by leveraging AWS capabilities.
- The three different types of messaging services offered are
 - 1) **Amazon Simple Queue Service (SQS),**
 - 2) **Amazon Simple Notification Service (SNS), and**
 - 3) **Amazon Simple Email Service (SES).**

Communication services- Messaging

1) Amazon Simple Queue Service (SQS),

- Amazon SQS constitutes disconnected model for **exchanging messages between applications by means of message queues, hosted within the AWS infrastructure.**
- Using the AWS console or directly the underlying Web service AWS, users can **create an unlimited number of message queues and configure them to control their access.**

Communication services- Messaging

1) Amazon Simple Queue Service (SQS),

- **Applications can send messages to any queue they have access to.**
- **These messages are securely and redundantly stored within the AWS infrastructure for a limited period of time, and they can be accessed by other (authorized) applications.**
- While a message is being read, it is kept locked to avoid spurious processing from other applications. Such a lock will expire after a given period

Communication services- Messaging

Amazon Simple Notification Service (SNS), and

- **Amazon SNS provides a publish-subscribe method for connecting heterogeneous applications.**

Communication services- Messaging

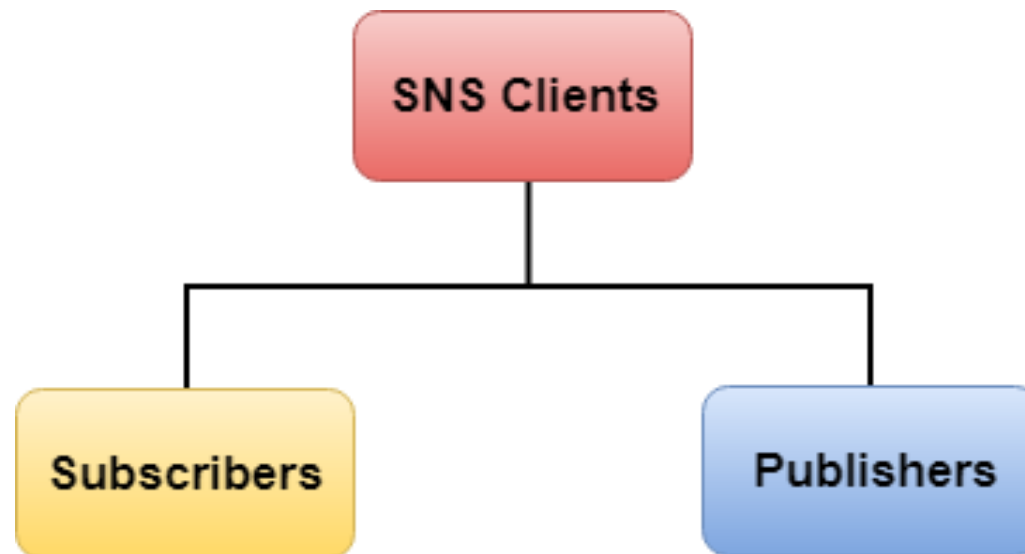
Amazon Simple Notification Service (SNS), and

- **Amazon SNS allows applications to be notified when new content of interest is available.**
- **This feature is accessible through a Web service whereby AWS users can create a topic, which other applications can subscribe to.**
- **At any time, applications can publish content on a given topic and subscribers can be automatically notified.**

Communication services- Messaging

Amazon Simple Notification Service (SNS)

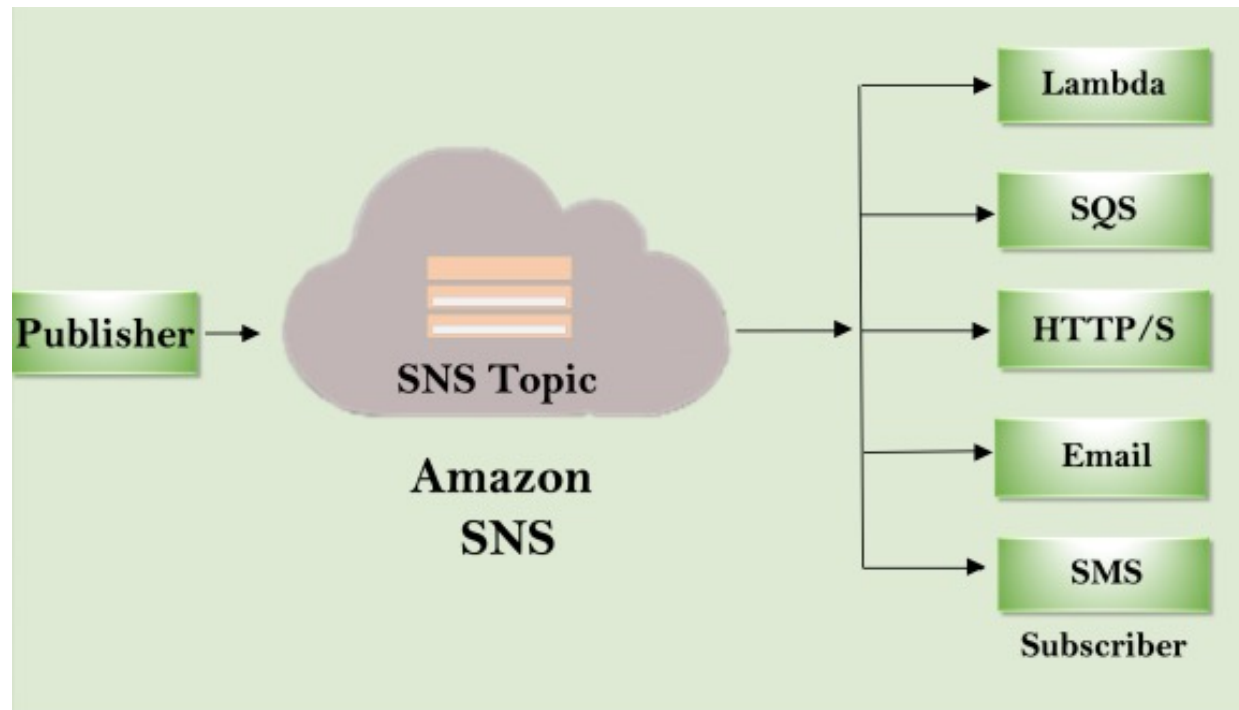
- Amazon SNS is a web service that manages sending messages to the subscribing endpoint.
- There are two clients of SNS:
- Subscribers
- Publishers



Amazon Simple Notification Service (SNS)

Publishers

- Publishers are also known as producers that produce and send the message to the SNS which is a logical access point.
- Subscribers such as web servers, email addresses, Amazon SQS queues, receive the message or notification from the SNS over one of the supported protocols (Amazon SQS, email, HTTP, SMS).



Amazon Simple Notification Service (SNS)

Publishers

- A publisher sends the message to the SNS topic that they have created.
- There is no need to specify the destination address while publishing the message as the topic itself matches the subscribers associated with the topic that the publisher has created and delivers the message to the subscribers.

Communication services- Messaging

Amazon Simple Email Service (SES).

- Amazon SES provides **AWS users with a scalable email service** that leverages the AWS infrastructure.

Communication services- Messaging

Amazon Simple Email Service (SES).

- 1) Once users are signed up for the service, they have to provide an email id that SES will use to send emails on their behalf.**
- 2) To activate the service, SES will send an email to verify the given address and provide the users with the necessary information for the activation.**
- 3) Upon verification, the user is given an SES sandbox to test the service**

Communication services- Messaging

Amazon Simple Email Service (SES).

- **SES also provides a wide range of statistics that help users to improve their email campaigns** for effective communication with customers

Communication services- Messaging

Amazon Simple Email Service (SES).

- Amazon SES is a **cloud email service provider that can integrate into any application for bulk email sending.** Whether you send transactional or marketing emails, you pay only for what you use.

-AMAZON

Additional services

- Besides compute, storage, and communication services,
- AWS provides a collection of services that allow **users to utilize services in aggregation.**
- The two relevant services are
 - 1) **Amazon CloudWatch and**
 - 2) **Amazon Flexible Payment Service (FPS)**

Additional services

Amazon CloudWatch

- is a service that provides a comprehensive set of statistics
- that **help developers understand and optimize the behavior of their application hosted on AWS.**

Amazon CloudWatch

- Observe and monitor resources and applications on AWS, on premises, and on other clouds

-AMAZON AWS

Additional services

Amazon CloudWatch

- CloudWatch collects **information from several other AWS services:**
- **EC2,**
- **S3,**
- **SimpleDB,**
- **CloudFront, and others.**
- Using CloudWatch, developers can see **a detailed breakdown of their usage of the service they are renting on AWS** and can devise more efficient and cost-saving applications.

Additional services

Amazon CloudWatch

- Earlier services of CloudWatch were offered only through subscription, but now it is made available for free to all the AWS users.

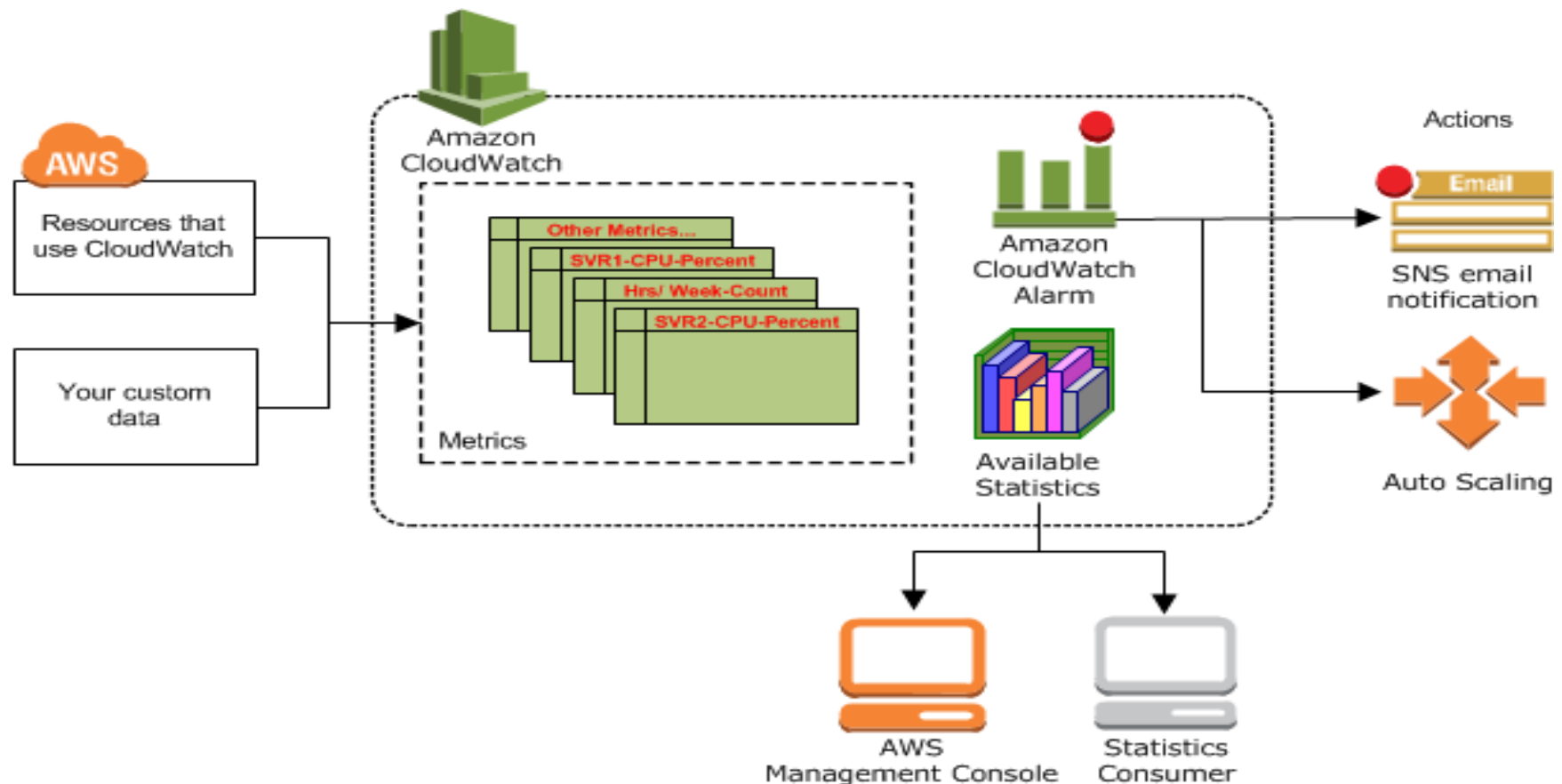
Amazon CloudWatch

- The CloudWatch home page automatically displays metrics about every AWS service you use.
- **You can additionally create custom dashboards to display metrics about your custom applications,** and display custom collections of metrics that you choose.

-AMAZON AWS

Amazon CloudWatch

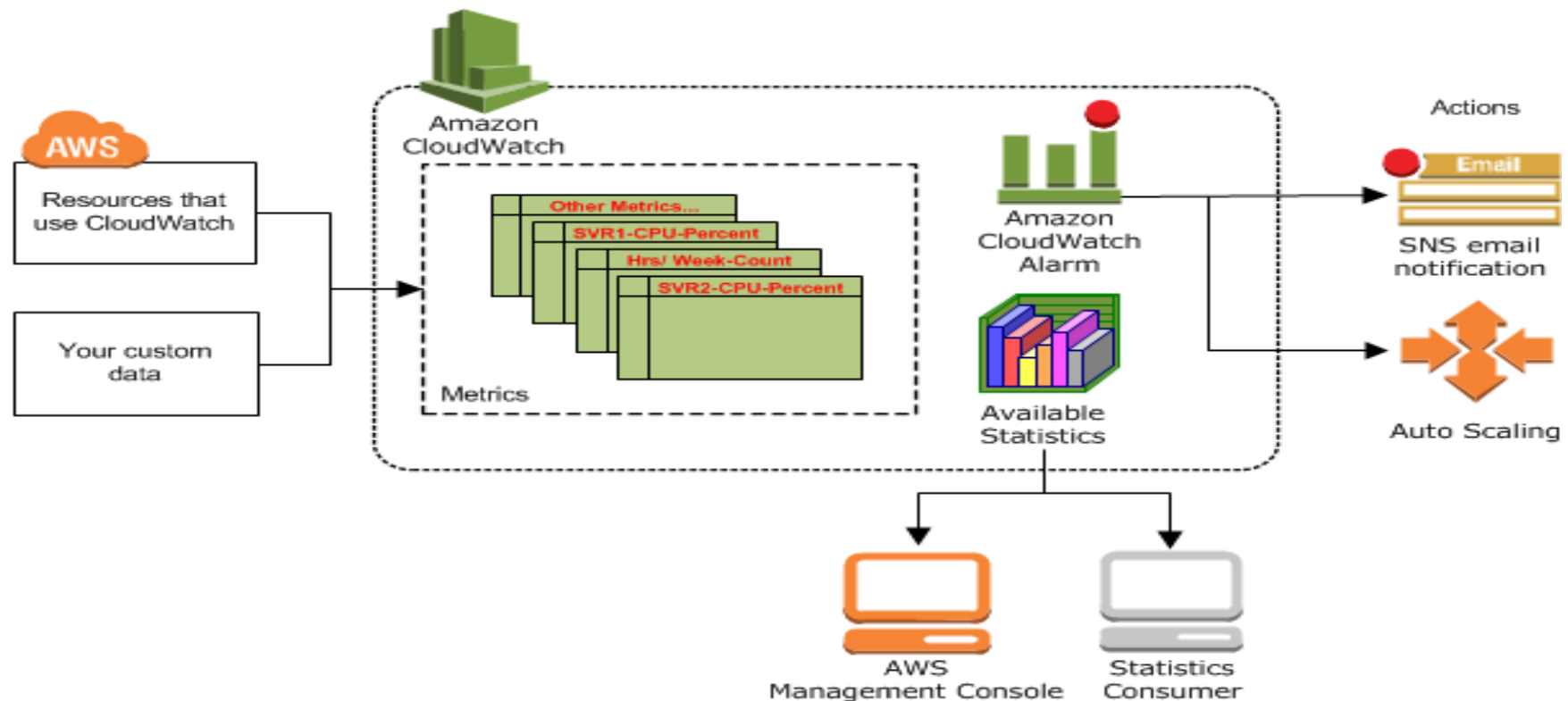
- You can create alarms that watch metrics and send notifications or
- **automatically make changes to the resources you are monitoring when a threshold is breached.**



Amazon CloudWatch

Eg-

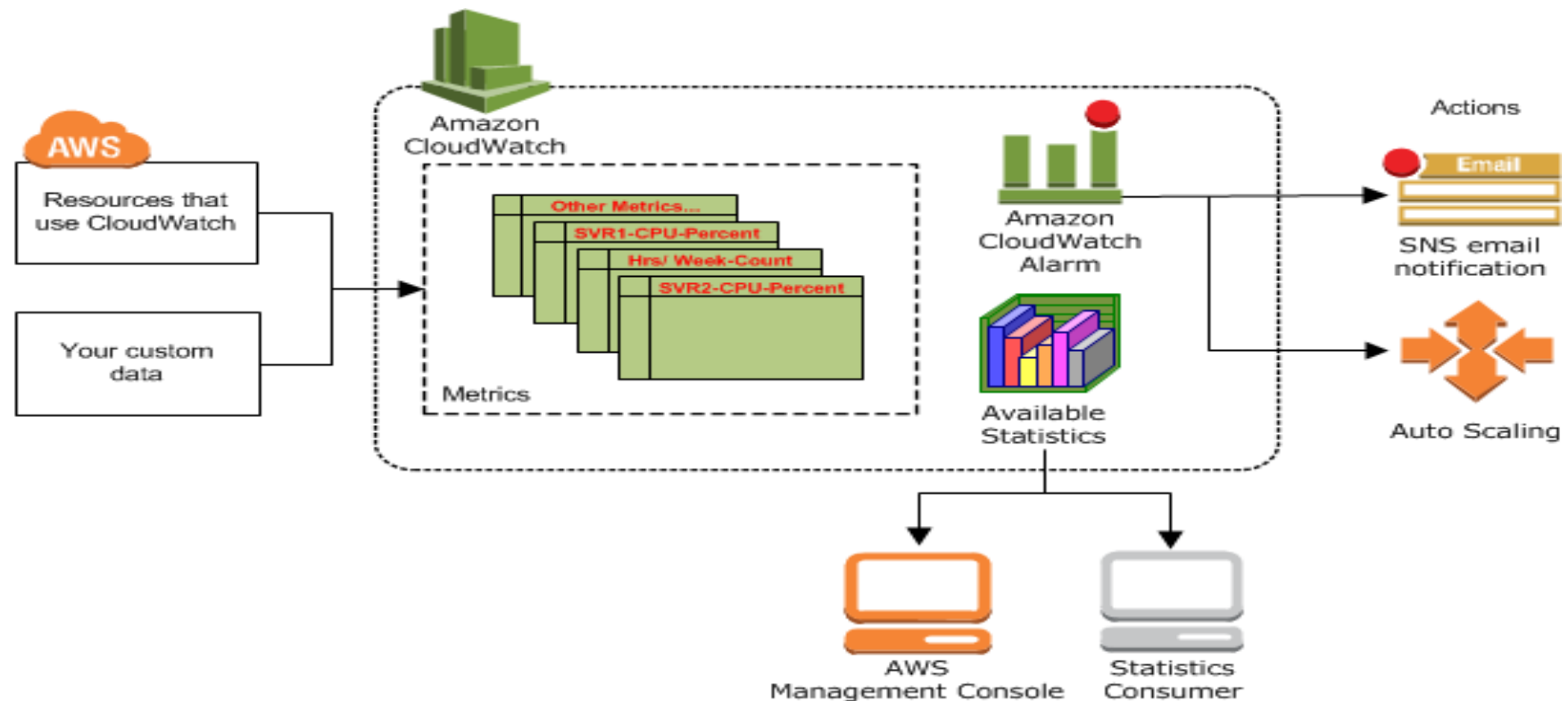
- One can monitor the CPU usage and disk reads and writes of your Amazon EC2 instances and
- **then use that data to determine whether you should launch additional instances to handle increased load.**



Amazon CloudWatch

Eg-

- You can also use this data to stop under-used instances to save money.



Additional services

Amazon Flexible Payment Service (FPS)-

- Amazon FPS infrastructure allows **AWS users to leverage Amazon's billing infrastructure to sell goods and services to other AWS users.**

Additional services

Amazon Flexible Payment Service (FPS)-

- Amazon FPS is the **first payments service designed from the ground up specifically for developers.**
- The set of web services APIs allows the movement of money between any two entities, humans or computers.

-AMAZON

Additional services

Amazon Flexible Payment Service (FPS)-

- Amazon FPS is a payment service that is **100% focused on the needs of developers.**
- Traditionally, developers have been limited in how they can manage payments.

-AMAZON

Additional services

Amazon Flexible Payment Service (FPS)-

- **If they need to charge with a certain frequency, execute a transaction at a specific time, combine many smaller payments into one single transaction or want to charge a commission on a transaction between two of their customers,**
- **they need to create their own payment infrastructure and processor relationships. This can be difficult, expensive, time-consuming, and with significant operational risks.**

-AMAZON

Additional services

Amazon Flexible Payment Service (FPS)-

- Amazon FPS changes the way developers can charge their customers.
- Payment Instructions give developers the flexibility to build multiple charging models that exactly meet their needs.

-AMAZON

Additional services

Amazon Flexible Payment Service (FPS)-

- **Using a capability called “Payment Instructions” developers can easily create the charging model that works best for them.**
- **For example, they can charge customers in small increments until their accumulated balance reaches a limit, pay a percentage of a digital transaction as a royalty, earn a commission on a marketplace transaction, or allow one customer to pay for another customer and limit their usage to a specific amount.**

-AMAZON