

UNIT III

3

Common Standards and Cloud Platforms

Syllabus

Common Standards : The Open Cloud Consortium, Open Virtualization Format, Standards for Application Developers : Browsers (Ajax), Data (XML, JSON), Solution Stacks (LAMP and LAPP), Syndication (Atom, Atom Publishing Protocol, and RSS), Standards for Security.

Amazon web services : Compute services Storage Services Communication Services Additional services.

Google AppEngine : Architecture and core concepts, Application life cycle, Cost model.

Microsoft Azure : Azure core concepts, SQL Azure, Windows Azure platform appliance.

Contents

3.1	Common Standards	
3.2	Standards for Application Developers April-18, May-18,
	 Dec.-18,
3.3	Standards for Security Marks 8
3.4	Amazon Web Services May-18,
3.5	Amazon Database Service Marks 4
3.6	Google AppEngine April-18,
3.7	Microsoft Azure Marks 4
3.8	Multiple Choice Questions April-18,
	 Marks 4

3.1 Common Standards

3.1.1 Open Cloud Consortium (OCC)

- ✓ The Open cloud Consortium is a 501(c)(3) non-profit venture which provides computing and data commons resources to support "scientific, environmental, medical and health care research."
- ✓ The Open Cloud Consortium has four working groups, one of which is the Open Science Data Cloud (OSDC).
- ✓ The infrastructure of the OSDC has been designed to address the challenges inherent in transporting large datasets, to balance the needs of data management and data analysis, and to archive data.
- ✓ The OSDC is based on a shared community infrastructure where hardware and software are shared among researchers and projects at the scale where it is more efficient to centrally locate and process data.
- ✓ It supports the development of standards for cloud computing and frameworks for interoperating between clouds; develops benchmarks for cloud computing; and supports reference implementations for cloud computing, preferably open source reference implementations.
- ✓ The OCC has a particular focus in large data clouds. It has developed the "MalStone Benchmark" for large data clouds and is working on a reference model for large data clouds

3.1.2 Open Virtualization Format (OVF)

- ✓ OVF is an open standard, specified by the Distributed Management Task Force (DMTF), for packaging and distributing a virtual appliance consisting of one or more virtual machines (VMs).
- An OVF Package is composed of metadata and file elements that describe virtual machines, plus additional information that is important to the deployment and operation of the applications in the OVF package. Its file extension is .ovf.
- An OVF Package always includes a descriptor file (*.ovf) and may also include a number of other files

more info

File type	Description
Descriptor	The descriptor specifies the virtual hardware requirements of the service and can also include other information such as descriptions of virtual disks, the service itself, and guest operating systems, a license agreement (EULA), instructions to start and stop VMs in the appliance, and instructions to install the service. The descriptor file extension is .ovf.

Manifest

The manifest is an SHA-1 digest of every file in the package, allowing the package contents to be verified by detecting any corruption. The manifest file extension is .mf.

Signature

The signature is the digest of the manifest signed with the public key from the X.509 certificate included in the package, and allows the package author to be verified. The signature file extension is .cert.

Virtual disks

OVF does not specify a disk image format. An OVF package includes files comprising virtual disks in the format defined by the virtualization product that exported the virtual disks. XenServer produces OVF packages with disk images in Dynamic VHD format; VMware products and Virtual Box produce OVF packages with virtual disks in Stream-Optimized VMDK format

3.2 Standards for Application Developers

SPPU : April-18, May-18, Dec.-18

3.2.1 AJAX

- AJAX is an acronym for Asynchronous JavaScript And XML. It is not a programming language but simply a development technique for creating interactive web applications.
- The technology uses JavaScript to send and receive data between a web browser and a web server.
- The AJAX technique makes web pages more responsive by exchanging data with a server behind the scenes, instead of reloading an entire web page each time a user makes a change.
- With AJAX, web applications can be faster, more interactive, and more user friendly.
- AJAX uses an XMLHttpRequest object to send data to a web server, and XML is commonly used as the format for receiving server data, although any format including and plain text can be used.
- AJAX is based on Open Standards.
- Ajax is a group of interrelated web development methods used on the client-side to create interactive web applications. The name implies that XML is used.
- However, many prefer to use URL-encoded data when sending data from the client to the server and JSON as the response.
- URL-encoded data sent by JavaScript code matches HTML form data sent by a browser when a user clicks the submit button. Managing form data received from a client is very easy when using LSP. The server also includes functions that automatically create JSON from Lua tables.

- Benefits of Ajax :
 - a. Callbacks : Ajax is used to perform a callback, making a quick round trip to the server to retrieve and/or save data without posting the entire page back to the server.
 - b. Making asynchronous calls : Ajax allows you to make asynchronous calls to the web server.
 - c. Increased speed : The main purpose of Ajax is to improve the performance and usability of a web application.
 - d. User-friendly : Because a page postback is being eliminated, Ajax enabled applications will always be more responsive, faster and more user-friendly.

3.2.2 JSON

- JavaScript Object Notation (JSON) is used to format data. It is commonly used on the Web as a vehicle to describe data being sent between systems.
- JSON is much easier to use with JavaScript than XML. When it comes to Ajax and JavaScript, JSON Web Services are replacing XML Web Services.
- The JSON format is often used for serializing and transmitting structured data over a network connection. It is often used to transmit data between a server and a web application, serving as an alternative to XML.
- JSON is based on a subset of Javascript, containing object and array. Objects contain pairs of property and value. Arrays contain values. A value could be a string, number, object array, true, false or null.
- On average, JSON requires less characters, and so less bytes, than the same data in XML. Because it uses JavaScript syntax, it requires less parsing than XML when used in Ajax Applications

3.2.3 XML

- XML stands for eXtensible Markup Language. It is emerging as a standard for exchanging data on the Web. It enables separation of content (XML) and presentation (XSL).
- The XML standard was created by W3C to provide an easy to use and standardized way to store self describing data.
- XML is a markup language in a standard plain text format. It contains structured or semi-structured data in verbose user-defined tags presented in a hierarchical way (tree-like structure)

- XML is not a replacement for HTM and traditional databases. XML documents are used either as a container to store semi-structured data or a media to exchange data between heterogeneous application.
- XML can be used to provide more information about the structure and meaning of the data in the Web pages rather than just specifying how the Web pages are formatted for display on the screen.
- XML provides the ability to structure, optionally validate, and transform data, allowing it to be used across various applications in a platform independent manner.
- The term "Extensible" refers to the capability of being extended while the phrase "Markup Language" refers to the set of conventions used for encoding textual information.
- XML documents have both logical and physical structure. A document is built up from storage units called entities. They can contain parsed or unparsed data.
- Parsed entities contain characters that formed either character data or markups. Markups are used to encode the logical and physical structure of the document. Both structures are subject of limitations.
- Logical structure : The document logical structure consists of declarations, elements, comments, processing instructions and character references.
- Every well-formed document contains one or more elements that form a tree hierarchy. Consequently, there is exactly one element at the highest level of the hierarchy that serves as a root for the tree. Every element has content and zero or more attributes.
- Physical structure : Every XML document is composed of storage units called entities. An entity has a name and content. The name is used to form a reference to the entity. There are two exceptions of entities without names - the document entity and the part of the DTD that is not contained in the document.
- An entity can contain references to other entities. There is a special entity called document entity or root that serves as a main storage unit. XML processors always start document processing from that unit, which can contain the whole document.

Similarities and comparison of JSON and XML

Similarities :

1. Both are human readable
2. Both have very simple syntax
3. Both are hierarchical
4. Both are language independent

5. Both can be used by Ajax
6. Both supported in APIs of many programming languages

Differences :

1. Syntax is different
2. JSON is less verbose
3. JSON can be parsed by JavaScript's eval method
4. JSON includes arrays
5. Names in JSON must not be JavaScript reserved words
6. XML can be validated

3.2.4 Atom and RSS

Atom :

- Atom is the name of an XML-based Web content and metadata syndication format, and an application-level protocol for publishing and editing Web resources belonging to periodically updated websites.
- Before the development of ATOM, RSS was the used format for web content syndication. People using RSS 2.0 felt there were considerable deficiencies in the format. RSS 2.0 was frozen and not changed.
- All Atom feeds must be well-formed XML documents, and are identified with the application/atom+xml media type.

RSS :

- RSS stands for Really Simple Syndication. It's an easy way for you to keep up with news and information that's important to you, and helps you avoid the conventional methods of browsing or searching for information on websites.
- Now the content you want can be delivered directly to you without cluttering your inbox with e-mail messages.. This content is called a "feed."
- RSS is written in the Internet coding language known as XML (eXtensible Markup Language).
- RSS is a format based on XML and is used for rapid distribution of various facilities.
- RSS content can be monitored using an RSS browser or aggregator. RSS aggregator comes in the form of client application. Application can be run on personal computer or as web application accessed via web browser.
- RSS has become the standard technology for automatically publishing information to large audiences.

Working of RSS reader :

- Websites summarize content in an RSS feed (XML document). Visitors download an RSS reader.
- There are generally two different types of RSS readers. The first kind of feed reader is a self contained program and the second kind of feed reader uses a web browser.
- Many of the programs are free, but generally those that have a small fee are more robust. Visitors select the content and summaries what they wish to view in a news aggregator or RSS reader.
- Content is added to the news reader by entering the URL or web address of the XML file. Clicking on the RSS link will provide the URL of the feed.
- Some RSS readers will auto-detect an XML file on a site indicating that a feed is available. Each time the feed is updated the content being viewed in the RSS reader indicates that there is new content.
- This insures that the customer has current information related to the topics they choose. Professionals estimate that RSS will soon rival e-mail's popularity as a content delivery method.

3.2.5 Difference between RSS and Atom

RSS	Atom
Real Simple Syndication (RSS) is a family of web feeds formats that are used to publish frequently updated works.	Atom refers to a pair of standards for web feeds.
RSS shows the date timestamps of data when the feed was created and last updated.	Atom shows the date timestamp when the website was last updated.
RSS has a more complicated aggregating and extracting process	Atom has an easier aggregating and extracting process.
RSS does not distinguish a partial from an excerpt.	Atom distinguishes a partial from an excerpt
RSS has two main publishing protocols; Blogger protocol and MetaWeblog	Atom has one standardized protocol

3.2.6 Solution Stacks (LAMP and LAPP)

- Solution stack is a set of different programs or application software that are bundled together in order to produce a desired result or solution.
- This may refer to any collection of unrelated applications taken from various subcomponents working in sequence to present a reliable and fully functioning

software solution. Many computer companies like Microsoft and Linux provide different solution stacks to clients.

a) LAMP

- LAMP is an archetypal model of web service stacks named as an acronym of the names of its original four open-source components: the Linux operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language.
- The common software components that make up a traditional LAMP stack are :
 1. Linux : The operating system (OS) makes up our first layer. Linux sets the foundation for the stack model. All other layers run on top of this layer.
 2. Apache : The second layer consists of web server software, typically Apache Web Server. This layer resides on top of the Linux layer. Web servers are responsible for translating from web browsers to their correct website.
 3. MySQL : Our third layer is where databases live. MySQL stores details that can be queried by scripting to construct a website. MySQL usually sits on top of the Linux layer alongside Apache/layer 2. In high end configurations, MySQL can be off loaded to a separate host server.
 4. PHP : Sitting on top of them all is our fourth and final layer. The scripting layer consists of PHP and/or other similar web programming languages. Websites and Web Applications run within this layer.

b) LAPP

- The LAPP stack is an open source web platform that can be used to run dynamic web sites and servers. It is considered by many to be a powerful alternative to the more popular LAMP stack and includes Linux, Apache, PostgreSQL (instead of MySQL) and PHP, Python and Perl.
- LAPP uses Linux as the operating system, Apache as the Web server, PostgreSQL as the RDBMS, and PHP as the object-oriented scripting language. Perl or Python may be substituted for PHP.

Review Questions

1. Write a short note on LAPP solution stack.
2. Write a short note on open virtualization format.
3. Write short note on following standards of application developer :
 - i) Solution stack (LAMP and LAPP)
 - ii) Syndication (Atom and RSS).

SPPU : April-18 In Sem, Marks 4

SPPU : May-18 End Sem, Marks 4

SPPU : Dec.-18 End Sem, Marks 8

3.3.1 SAML

- Security Assertion Markup Language (SAML) is an XML standard that allows a user to log on once to the log-on site for all the trusted websites.
- SAML relies on XML-based data to transfer authentication and authorization details. Using SAML, users, applications and services can exchange identity information. This is done through SAML Assertions, which are compressed, encoded and possibly encrypted XML nodes.
- SAML defines three roles :
 1. Identity Provider (IDP) : This role will validate the identity of a user who is asking for a service.
 2. Service Provider (SP) : This role will provide services to user.
 3. Principal : This is typically the user asking for a service from SP and getting validated by IDP.
- SAML is designed for business to business (B2B) and business to consumer (B2C) transactions.
- SAML has the following components :
 1. Assertions : Authentication, attribute, authorization
 2. Protocols : HTTP, SMTP, FTP, SOAP
 3. Bindings : SAML over SOAP, SAML over HTTP

3.3.2 OAuth

- The latest iteration of OAuth, formalized in 2012, is the version 2.0. OAuth provides authorization to APIs
- As one can imagine, it is much more accommodating to current trends and needs in the industry.
- OAuth includes the notion of Access Token as the mechanism of choice for allowing access to restricted resources. In other words, an Access Token is the authorization issued to a client.
- OAuth has following roles :
 1. A resource server called as OAuth Provider - the entity which is hosting the resource.
 2. End user that owns the resource that is being requested.
 3. Client - OAuth Consumer - the entity requesting for resource.

- Fig. 3.3.1 shows working of OAuth.

1. User is logged into a social network platform like Facebook (OAuth Consumer in this case) and wants to import contacts from email provider such as Gmail (OAuth Provider in this case).

2. OAuth Consumer redirects the request to OAuth Provider to generate a request token and secret key.

3. OAuth Provider generates request token and secret key and sends it to OAuth Consumer.

4. OAuth Consumer asks the user to authorize and sends the user the request token received from OAuth Provider.

5. On clicking the link, secret key is passed over to OAuth Provider using API.

6. OAuth Provider asks the user to authorize the OAuth Consumer to consume its data.

7. User confirms the OAuth Consumer to consume its data.

8. OAuth Provider sends user a "request token" and a "request token secret" and redirects the user to OAuth Consumer.

9. User gets redirected to OAuth Consumer with a "request token" and a "request token secret".

10. OAuth Consumer presents the "request token" and the "request token secret" and asks for user contacts.

Limitations :

1. OAuth 1.0 was vulnerable to session fixation attack.
2. OAuth 2.0 does not have native encryption capabilities.

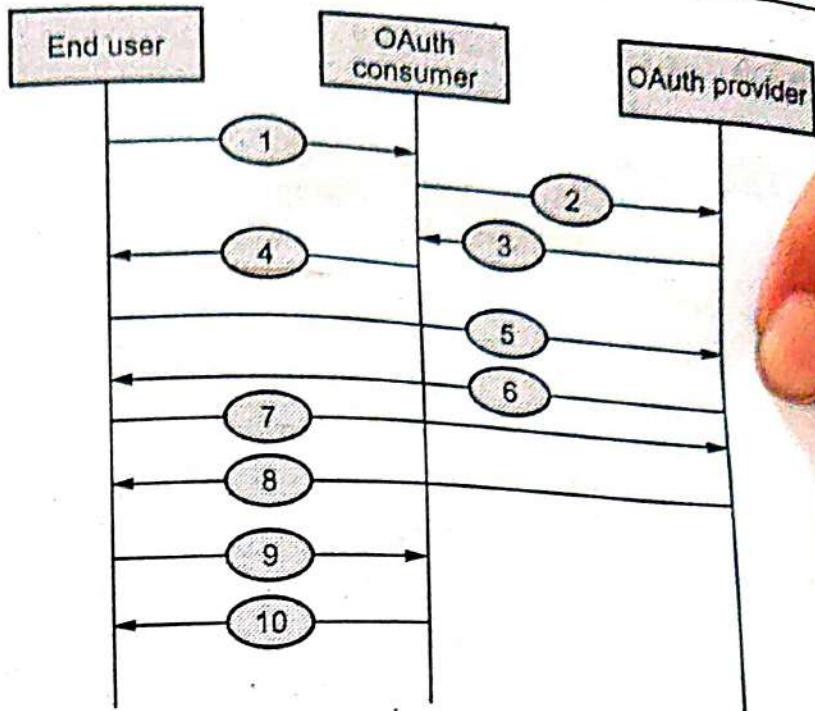


Fig. 3.3.1 OAuth working

3.3.3 Virtualization Attack

- Each component of virtualization layer can act as an attack vector to launch multiple attacks on the system.
 - Attacks that target different components of virtualization environment may result in security issues such as compromise of complete cloud infrastructure, stealing of customer data and system hacking.
1. **Service provider attacks** : If the attacker has physical access to the Cloud hardware, he may run malicious application or code in the system to damage the VMs by modifying their source code and changing their functionality. With the help of physical access to system, attackers can also launch cross VM side channel attacks. These attacks include CPU cache leakage to measure the load of other virtual web server on the network.
 2. **Hypervisor attacks** : A cloud customer can lease a guest VM to install a malicious guest OS, which attacks and compromises the hypervisor by changing its source code in order to gain access to the memory contents of VMs present in the system. Fig. 3.3.2 shows VM escape attack.

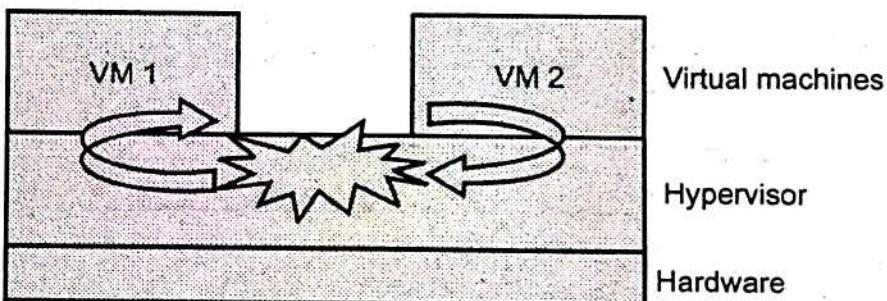


Fig. 3.3.2 VM escape attack

3. **Virtual machine attacks** : Malicious programs in different virtual machines can achieve required access permissions to log keystrokes and screen updates across virtual terminals that can be exploited by attackers to gain sensitive information.
4. **Guest image attacks** : Unnecessary guest OS images in cloud can result in different security issues if the security of each image is not maintained. If a malicious guest OS image is migrated to another host, it can compromise the other system as well.

Review Question

1. Write a short note on virtualization attack.

SPPU : May-18 End Sem, Marks 4

3.4 Amazon Web Services

- Amazon Web Services (AWS) is a cloud computing platform from Amazon that provides customers with a wide array of cloud services.

- Amazon first debuted its Amazon Web Services in 2006 as a way to enable the use of online services by client-side applications or other web sites via HTTP, REST or SOAP protocols.
- Amazon bills customers for Amazon AWS based on their usage of the various Amazon Web Services.
- In 2012, Amazon launched the AWS Marketplace to accommodate and grow the emerging ecosystem of AWS offerings from third-party providers that have built their own solutions on top of the Amazon Web Services platform.
- The AWS Marketplace is an online store for Amazon Web Services customers to find, compare and begin using AWS software and technical services.
- Amazon Web Services is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.
- In 2017, AWS comprised more than 90 services spanning a wide range including computing, storage, networking, database, analytics, application services, deployment, management, mobile, developer tools, and tools for the Internet of Things.
- Today, Amazon Web Services provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers hundreds of thousands of businesses in 190 countries around the world.
- In 2016 AWS partnered with Digital Currency Group to create a laboratory environment allowing companies to experiment with block chain technologies.
- In January 2018, Amazon launched an autoscaling service on AWS.

What is Amazon Web Services ?

- Amazon Web Services (AWS) is a collection of remote computing services (web services) that together make up a cloud computing platform, offered over the Internet by Amazon.com.
- The AWS Cloud infrastructure is built around Regions and Availability Zones (AZs). A Region is a physical location in the world where we have multiple AZs. AZs consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities.
- These AZs offer you the ability to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible from a single data center.
- The AWS cloud operates 42 AZs within 16 geographic regions around the world, with five more availability zones and two more regions coming online in 2017.

- Each availability zone is designed as an independent failure zone. This means that availability zones are physically separated within a typical metropolitan region and are located in lower risk flood plains.

3.4.1 Components

- AWS consists of many cloud services that you can use in combinations tailored to your business or organizational needs.
- With Amazon Web Services you will find a complete cloud platform ready to use for virtually any workload.
- The user requests to the server by the method such as e-mail either to register or to transfer the domain.
- Your request which includes all information will be sent to Amazon API Gateway restful service.
- API Gateway will transfer the collected user information to an AWS Lambda function.
- AWS Lambda function will generate an e-mail and forward it to the 3rd party mail server using Amazon SES. *Simple email service*
- Components of Amazon Web Service architecture are Amazon API Gateway, AWS Lambda, Amazon Simple Email Service.
- API Gateway is a front-door to access data, business logic and functionality. API Gateway will provide a restful API endpoint for our AWS Lambda function.
- API works at small as well as large-scale and helps developers to manage, spectate, create and provide security to the API's.

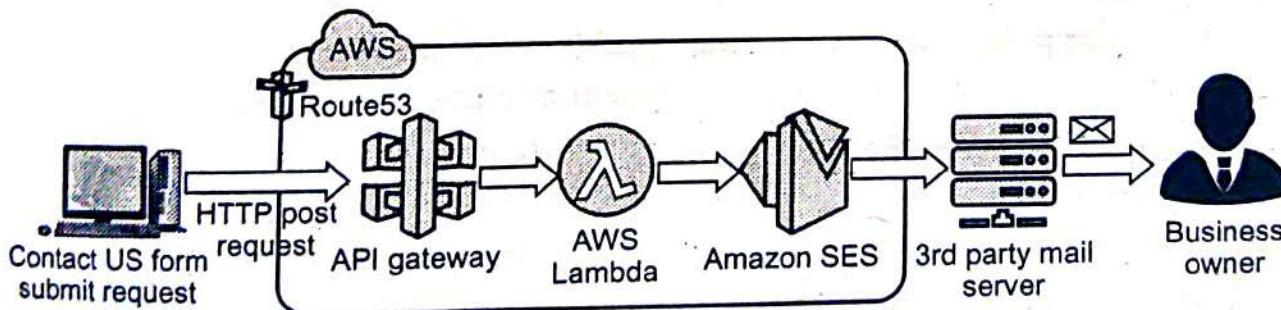


Fig. 3.4.1 AWS

- AWS Lambda is a compute service that runs your back-end code and responds to events such as object uploads to Amazon S3 bucket, Dynamo DB or in-app activity. The Lambda function will get all the information from a user through API Gateway.

- Amazon Simple email service helps us to send e-mail with minimal setup and maximum deliverability. It is integrated with AWS management console so that you can monitor your sending activity. Amazon Simple Email Service helps us by monitoring insecurity.

3.4.2 Advantages and Disadvantages of AWS

Advantages :

1. Easy to use.
2. No capacity limits : Organizations launch different projects and the guess what capacity they will need.
3. Provides speed and agility.
4. Secure and reliable : AWS provides security and also helps to protect the privacy as it is stored in AWS data centers.

Disadvantages :

1. Limitations of Amazon EC2 : AWS sets default limits on resources which vary from region to region. These resources consist of images, volumes, and snapshots.
2. Technical support fee : AWS charges you for immediate support.
3. Security limitations.

3.4.3 Compute Service

- Compute services contains the fundamental element of cloud computing systems. Example of compute service is Amazon EC2.
- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers and system administrators.
- Amazon EC2 reduces the time required to obtain and boot new server instances (called Amazon EC2 instances) to minutes, allowing user to quickly scale capacity, both up and down, as your computing requirements change.
- EC2 allows creating Virtual Machines (VM) on-demand. Pre-configured template Amazon Machine Image (AMI) can be used get running immediately. Creating and sharing your own AMI is also possible via the AWS Marketplace.

3.4.3.1 Amazon Machine Image

- Amazon Machine Image (AMI) is a template for software configuration (Operating System, Application Server, and Applications).

- Machine imaging is a process that is used to provide system portability, and provision and deploy systems in the cloud through capturing the state of systems using a system image.
- A system image makes a copy or a clone of the entire computer system inside a single file. The image is made by using a program called system imaging program and can be used later to restore a system image
- An AMI typically contains three things : template, permission to launch, block device mapping.
 1. **Template** : For the root volume for the instances (An application server, an OS, and applications)
 2. **Permissions to launch** : Which account can use this AMI to launch instances.
 3. **Block device mapping** : That specifies the volumes to attach to the instance during its launch time.
- AMIs provide a template for the root volume required to launch a particular instance. This will typically include the operating systems, an application server and applications.
- It also includes in the AMI are launch permissions that restrict the ability to launch instances from that AMI to defined AWS accounts. Finally, a block device mapping specifies the volumes to attach to the instance once it is launched.
- Once an AMI has been created and registered, it can be used to launch new instances. An AMI can be copied to different regions, and it can also be deregistered. Fig. 3.4.2 shows an AMI lifecycle.

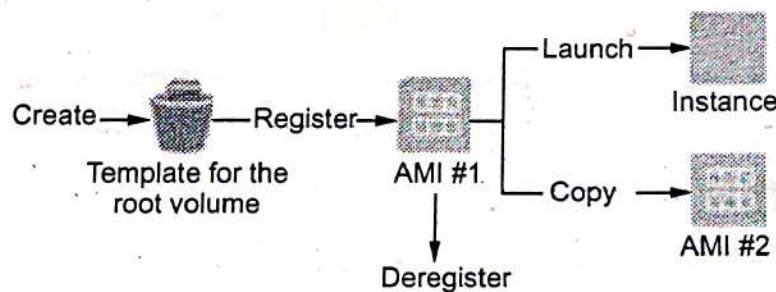


Fig. 3.4.2 AMI lifecycle

- Once an AMI is created, it is stored in an S3 bucket and the user can decide whether to make it available to other users or keep it for personal use.
- Instance is AMI running on virtual servers in the cloud. Each instance type offers different compute and memory facilities. Create an Amazon Machine Image containing your applications, libraries, data and associated configuration settings. Or use pre-configured, templated images to get up and running immediately.
- Auto scaling allows automatically scale of the capacity up seamlessly during demand spikes to maintain performance and scales down during demand lulls to minimize costs.

- Elastic load balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It provides tools to build failure resilient applications by launching application instances in separate availability zones.
- AMIs can be attained directly from AWS, can be created and shared through communities, or can be purchased from vendors via the AWS Marketplace.
- Pay only for resources actually consume, instance-hours. VM Import/Export enables user to easily import virtual machine images from existing environment to Amazon EC2 instances and export them back at any time.
- Boto is a Python package that provides programmatic connectivity to Amazon Web Services.
- The AMI files are encrypted and compressed for security purpose and stored in Amazon S3 (Simple Storage System) buckets as a set of 10 MB chunks.
- Machine imaging is mostly run on virtualization platform due to this it is also called as virtual appliances and running virtual machines are called instances.
- The AMI file system is not a standard bit-for-bit image of a system that is common to many disk imaging programs. AMI omits the kernel image and stores a pointer to a particular kernel that is part of the AWS kernel library.
- Among the choices are Red Hat Linux, Ubuntu, Microsoft Windows, Solaris and others. Files in AMI are compressed and encrypted and an XML file is written that describes the AMI archive.
- Machine images are sometimes referred to as "virtual appliances", systems that are meant to run on virtualization platforms.

3.4.3.2 EC2 Instances

- Amazon Elastic Compute Cloud (Amazon EC2) instances represent virtual machines. EC2 instances are launched by created by an Amazon Machine Image (AMI). An AWS template that describes and defines the OS and operating environment for one or more EC2 instances of one or more EC2 instance types.
- Each instance type delivers a mix of CPU, memory, storage, and networking capacity, across one or more size options, and should be carefully matched to your workload's unique demands.
- EC2 functions :
 1. Load variety of operating system.
 2. Install custom applications.
 3. Manage network access permission.
 4. Run image using as many/few systems as customer desire.

- Currently available configurations for EC2 instances are as follows :
 1. **Standard instances** : Among the most popular and widely used EC2 instance types. Standard instances have memory to CPU ratios suitable for most general-purpose applications. General-purpose instances include A1, M5, M5a, M4, T3, T3a, and T2.
 2. **Micro instances** : Micro instances can be used for small Web applications with limited traffic.
 3. **High-memory instances** : EC2 high memory instances offer 6, 9, 12, 18, and 24 TB of memory in an instance. These instances are purpose-built to run large in-memory databases.
 4. **High-CPU instances** : This types of instances are used in compute-intensive applications.
 5. **Cluster compute instances** : Cluster compute instances provide a high-performance network interconnect along with a high-performance CPU.
 6. **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.
- 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.
- EC2 advantages :
 1. Amazon EC2 enables you to increase or decrease capacity within minutes.
 2. User have complete control of your Amazon EC2 instances.
 3. Support flexible cloud hosting services.
 4. Secure : Amazon EC2 works in conjunction with Amazon VPC to provide security and robust networking functionality.
 5. Reliable : Amazon EC2 offers a highly reliable environment where replacement instances can be rapidly and predictably commissioned.

3.4.3.3 Configuring Amazon EC2 Linux Instances

- Let's get started with Amazon Elastic Compute Cloud (Amazon EC2) by launching, connecting to and using a Linux instance. An instance is a virtual server in the AWS cloud. With Amazon EC2, you can setup and configure the operating system and applications that run on your instance.
- When you sign up for AWS, you can get started with Amazon EC2 using the AWS Free Tier.

- The instance is an Amazon EBS-backed instance (meaning that the root volume is an EBS volume). You can either specify the availability zone in which your instance runs or let Amazon EC2 select an availability zone for you. When you launch your instance, you secure it by specifying a key pair and security group. When you connect to your instance, you must specify the private key of the key pair that you specified when launching your instance.
- Various steps to configure Amazon EC2 Linux instance is shown in Fig. 3.4.3.

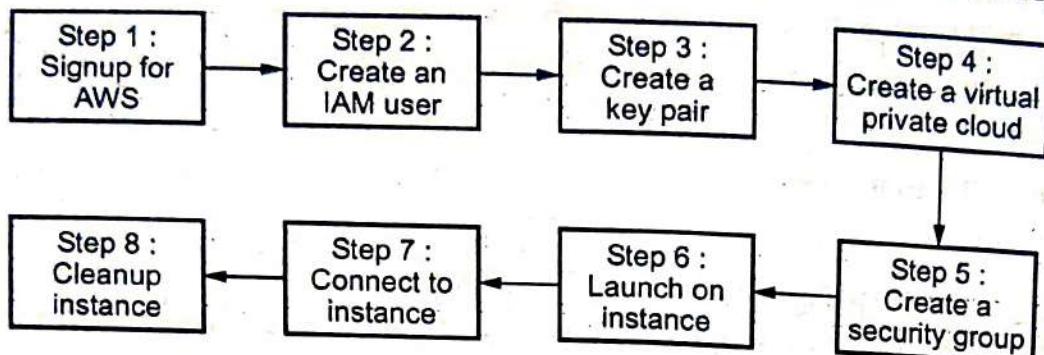


Fig. 3.4.3 Steps to sign up for EC2

Step 1 : SignUp for AWS

- When you signup for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon EC2. You are charged only for the services that you use.
- With Amazon EC2, you pay only for what you use. If you are a new AWS customer, you can get started with Amazon EC2 for free.

Step 2 : Create an IAM user

- Services in AWS, such as Amazon EC2, require that you provide credentials when you access them, so that the service can determine whether you have permission to access its resources. The console requires your password.
- You can create access keys for your AWS account to access the command line interface or API. However, we don't recommend that you access AWS using the credentials for your AWS account; we recommend that you use AWS Identity and Access Management (IAM) instead.
- Create an IAM user and then add the user to an IAM group with administrative permissions or grant this user administrative permissions. You can then access AWS using a special URL and the credentials for the IAM user. If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console.

Step 3 : Create a key pair

- AWS uses public-key cryptography to secure the login information for your instance. A Linux instance has no password; you use a key pair to log in to your

instance securely. You specify the name of the key pair when you launch your instance, then provide the private key when you log in using SSH. *Secure shell*

- If you haven't created a key pair already, you can create one using the Amazon EC2 console. Note that if you plan to launch instances in multiple regions, you'll need to create a key pair in each region.

Step 4 : Create a Virtual Private Cloud (VPC)

- Amazon VPC enables you to launch AWS resources into a virtual network that you've defined, known as a Virtual Private Cloud (VPC). The newer EC2 instance types require that you launch your instances in a VPC. If you have a default VPC, you can skip this section and move to the next task, create a security group. To determine whether you have a default VPC, open the Amazon EC2 console and look for default VPC under account attributes on the dashboard.

Step 5 : Create a security group

- Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your instance from your IP address using SSH. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere. Note that if you plan to launch instances in multiple regions, you'll need to create a security group in each region.

Step 6 : Launch an instance

- You can launch a Linux instance using the AWS management console as described in the following procedure.
 1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 2. From the console dashboard, choose Launch Instance.
 3. The Choose an Amazon Machine Image (AMI) page displays a list of basic configurations, called Amazon Machine Images (AMIs), that serve as templates for your instance. Select an HVM version of Amazon Linux 2. Notice that these AMIs are marked "Free tier eligible."
 4. On the Choose an Instance Type page, you can select the hardware configuration of your instance. Select the t2.micro type, which is selected by default. Notice that this instance type is eligible for the free tier.
 5. Choose Review and Launch to let the wizard complete the other configuration settings for you.
 6. On the Review Instance Launch page, under security groups, you'll see that the wizard created and selected a security group for you. You can use this security group or alternatively you can select the security group that you created when getting setup using the following steps.

- a) Choose Edit security groups.
- b) On the Configure Security Group page, ensure that Select an existing security group is selected.
- c) Select your security group from the list of existing security groups and then choose Review and Launch.
7. On the Review Instance Launch page, choose Launch.
8. When prompted for a key pair, select Choose an existing key pair, then select the key pair that you created when getting setup. When you are ready, select the acknowledgement check box and then choose launch instances.
9. A confirmation page lets you know that your instance is launching. Choose View Instances to close the confirmation page and return to the console.
10. On the Instances screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running and it receives a public DNS name.
11. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks; you can view this information in the status checks column.

Step 7 : Connect to your Instance

Several ways to connect to your Linux instance is shown in Table 3.4.1.

Your computer OS	Topic
Linux	Connecting to your Linux instance using SSH.
Windows	Connecting to your Linux instance from Windows using PuTTY.
	Connecting to your Linux instance from Windows using Windows Subsystem for Linux.
Other	Connecting to your Linux instance using MindTerm

Table 3.4.1 Ways to connect to Linux Instance

Step 8 : Cleanup your instance

- After you've finished with the instance, you should cleanup by terminating the instance.
- Terminating an instance effectively deletes it; you can't reconnect to an instance after you've terminated it.
- If you launched an instance that is not within the AWS free tier, you'll stop incurring charges for that instance as soon as the instance status changes to

shutting down or terminated. If you'd like to keep your instance for later, but not incur charges, you can stop the instance now and then start it again later.

- To terminate your instance following steps can be used :
 - 1) In the navigation pane, choose instances. In the list of instances, select the instance.
 - 2) Choose actions, instance state, terminate.
 - 3) Choose yes, terminate when prompted for confirmation.
- Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while and then the entry is deleted.

3.4.4 Storage Service

- AWS provides a collection of services for data storage and information management. It is represented by Amazon Simple Storage Service (S3).
- Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web. S3 can serve as a raw data store for IoT systems for storing raw data, such as sensor data, log data, audio and video data.

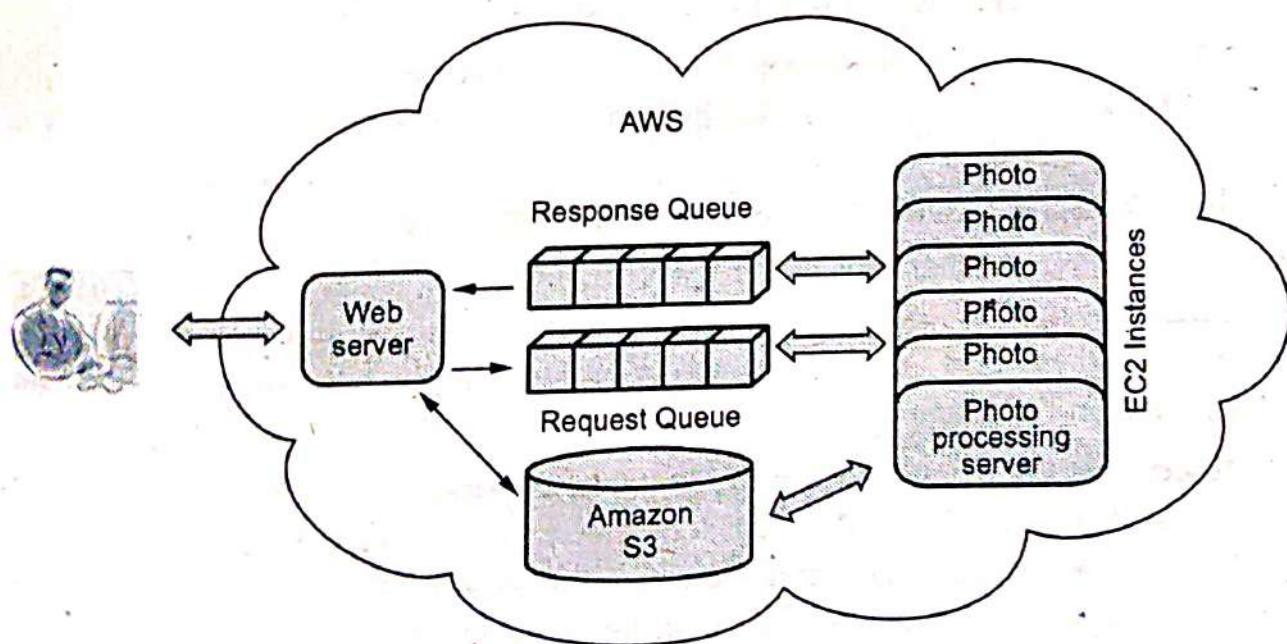


Fig. 3.4.4 Amazon S3 working

Features :

1. Unlimited storage
2. Highly scalable : In terms of storage, request rate and concurrent users.
3. Reliable : Store redundant data in multiple facilities and on multiple devices.

- 4. Secure : Flexibility to control who / how / when / where to access the data.
- 5. Performance : Choose region to optimize for latency / minimize costs.
- Example : Online photo processing service.

Procedure :

1. Web server receive request.
 2. Put request message in the queue.
 3. Pictures stored in S3.
 4. Multiple EC2 instances run photo processing.
 5. Put back in the queue.
 6. Return
- Store data on Amazon's distributed system containing multiple servers within Amazon's data center locations. Amazon doesn't offer you a GUI based tool to access your data. You can use one of the several tools online or build one through APIs.
 - Amazon EC2 provides three type of storage option : Amazon EBS, Amazon S3 and Instance Storage. Amazon EBS (Elastic Block Store) provides with persistent, block-level storage. Basically additional hard disk that you can attach to instance. It suitable for apps which require database, filesystem, block level storage.
 - A **bucket** is a container for objects stored in Amazon S3. Every object is contained in a bucket. For example, if the object named "photos/puppy.jpg" is stored in the rakshita bucket, then it is addressable using the URL <http://rakshita.s3.amazonaws.com/photos/puppy.jpg>
 - Buckets serve several purposes : They organize the Amazon S3 namespace at the highest level, they identify the account responsible for storage and data transfer charges, they play a role in access control and they serve as the unit of aggregation for usage reporting.
 - Objects are the fundamental entities stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3. The metadata is a set of name-value pairs that describe the object. These include some default metadata, such as the date last modified and standard HTTP metadata, such as content-type. You can also specify custom metadata at the time the object is stored.
 - A **key** is the unique identifier for an object within a bucket. Every object in a bucket has exactly one key. Because the combination of a bucket, key and version ID uniquely identify each object, Amazon S3 can be thought of as a basic data map between "bucket + key + version" and the object itself. Every object in

Amazon S3 can be uniquely addressed through the combination of the web service endpoint, bucket name, key and optionally, a version.

- **Regions :** You can choose the geographical region where Amazon S3 will store the buckets you create.. Objects stored in a region never leave the region unless you explicitly transfer them to another region.

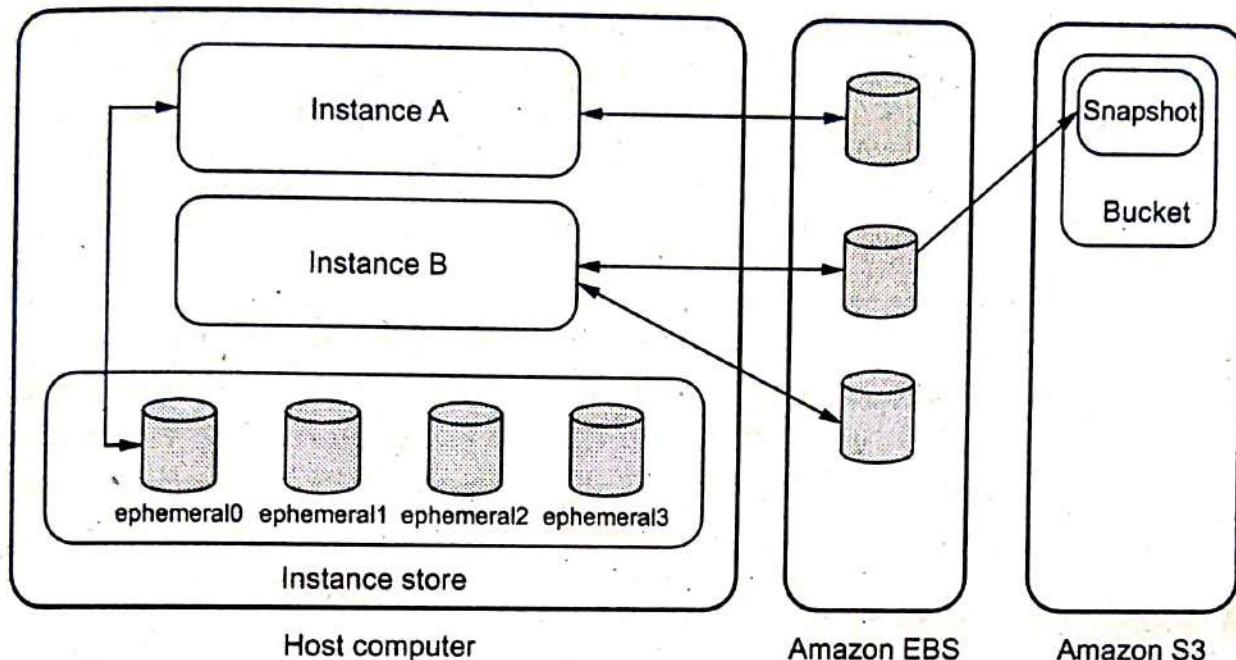


Fig. 3.4.5 Amazon EBS and S3

3.4.4.1 Bucket

- Amazon S3 defines a bucket name as a series of one or more labels, separated by periods, that adhere to the following rules : The bucket name can be between 3 and 63 characters long, and can contain only lower-case characters, numbers, periods, and dashes
- Amazon S3 defines a bucket name as a series of one or more labels, separated by periods, that adhere to the following rules :
 1. The bucket name can be between 3 and 63 characters long, and can contain only lower-case characters, numbers, periods, and dashes.
 2. Each label in the bucket name must start with a lowercase letter or number.
 3. The bucket name cannot contain underscores, end with a dash, have consecutive periods, or use dashes adjacent to periods.
 4. The bucket name cannot be formatted as an IP address (198.51.100.24).
- A bucket is owned by the AWS account that created it. By default, you can create up to 100 buckets in each of your AWS accounts. If you need additional buckets, you can increase your bucket limit by submitting a service limit increase
- The following are the rules for naming S3 buckets in all AWS Regions :
 1. Bucket names must be unique across all existing bucket names in Amazon S3.

2. Bucket names must comply with DNS naming conventions.
3. Bucket names must be at least 3 and no more than 63 characters long.
4. Bucket names must not contain uppercase characters or underscores.
5. Bucket names must start with a lowercase letter or number.
6. Bucket names must be a series of one or more labels. Adjacent labels are separated by a single period (.). Bucket names can contain lowercase letters, numbers, and hyphens. Each label must start and end with a lowercase letter or a number.
7. Bucket names must not be formatted as an IP address (for example, 192.168.5.4).
8. When you use virtual hosted-style buckets with Secure Sockets Layer (SSL), the SSL wildcard certificate only matches buckets that don't contain periods. To work around this, use HTTP or write your own certificate verification logic. We recommend that you do not use periods (".") in bucket names when using virtual hosted-style buckets.

3.4.2 Amazon Elastic Block Store

- Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud.
- Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability.
- EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone.
- EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage.
- The size of an EBS volume can be configured by the user and can range from 1 GB to 1 TB.
- The network-based EBS storage service is delivered in volumes, which can be attached to an EC2 instance and used just like a disk drive. Because a volume can become unformatted, it must have a file system installed (formatted) on it before it can be used.
- Amazon EBS is well suited to both database-style applications that rely on random reads and writes, and to throughput-intensive applications that perform long continuous reads and writes.

- Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build, manage, and secure your own key management infrastructure.
- When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted.
- Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes.
- EBS can make your applications more reliable, because the storage is separate from any specific instance.
- A large repository of public data set snapshots can be restored to EBS volumes and seamlessly integrated into AWS cloud-based applications.
- Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console.
- These metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need.
- Amazon EBS storage costs depend on how much EBS storage, in terms of gigabyte-per-month, is provisioned in a particular account.
- While EC2 instances only accrue charges while they're running, the EBS volumes attached to instances continue to retain information and hence accrue charges, even when the instance is stopped.
- An EBS snapshot is a point-in-time backup of an EBS volume. It is a "copy" of the data on in EBS volume. EBS snapshots are billed at a lower rate than active EBS volumes.
- If an EBS block has low access volume, the active volume of this EBS block can be deleted after the information stored in EBS is copied to an EBS Snapshot.

EBS Snapshots

- Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones.
- The volume does not need to be attached to a running instance in order to take a snapshot.
- As you continue to write data to a volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes.

- These snapshots can be used to create multiple new EBS volumes or move volumes across Availability Zones. Snapshots of encrypted EBS volumes are automatically encrypted.
- When you create a new volume from a snapshot, it's an exact copy of the original volume at the time the snapshot was taken.
- EBS volumes that are restored from encrypted snapshots are automatically encrypted. The snapshots can be shared with specific AWS accounts or made public.
- When you create snapshots, you incur charges in Amazon S3 based on the volume's total size. For a successive snapshot of the volume, you are only charged for any additional data beyond the volume's original size.
- Snapshots are incremental backups, meaning that only the blocks on the volume that have changed after your most recent snapshot are saved.

3.4.4.3 Amazon ElastiCache

- It is a fully managed caching service.
- ElastiCache is protocol-compliant with Memcached, an open source, high-performance, distributed memory object caching system for speeding up dynamic web applications by alleviating database load.
- According to the Amazon website, ElastiCache makes it easy to deploy, operate, and scale an in-memory cache in the cloud.
- The service improves the performance of web applications by enabling information retrieval from a fast, managed, in-memory caching system, instead of relying entirely on slower disk-based databases.
- ElastiCache is a managed, in-memory data store service. It has two engines AWS Redis and Memcached which is used to power real-time applications.
- Memcached is a general-purpose distributed memory caching system. It is often used to speed up dynamic database-driven websites by caching data and objects in RAM to reduce the number of times an external data source must be read.
- Memcached is free and open-source software, licensed under the Revised BSD license.
- ElastiCache offloads the administrative overhead of running a caching service by :
 1. Creating the server pool based on commands issued via the AWS Management Console or API.
 2. Managing the pool to ensure caching server availability.

- 3. Automatically patching servers with necessary software changes and migrating your data from an un-patched version to a new, patched version.
- 4. Allowing you to grow or shrink the pool with a simple command
- ElastiCache runs in the Amazon Virtual Private Cloud environment, giving you complete control over network access to your cache cluster.
- Amazon ElastiCache automatically detects and replaces failed nodes, reducing the overhead associated with self-managed infrastructures and provides a resilient system that mitigates the risk of overloaded databases, which slow website and application load times.
- Node is the smallest building block of an ElastiCache deployment. It is a fixed-size chunk of secure, network-attached RAM. Each cache node runs an instance of either Memcached or Redis.
- Memcached cluster can have up to 20 nodes.

3.4.4.4 Amazon SimpleDB

- SimpleDB provides a simplified data model based on the relational database data model. SimpleDB provides support for semi structured data, the model for which is based on the concept of domains, items, and attributes.
- This service works in close conjunction with Amazon Simple Storage Service (Amazon S3) and Amazon Elastic Compute Cloud, collectively providing the ability to store, process and query data sets in the cloud. These services are designed to make web-scale computing easier and more cost-effective for developers.
- SimpleDB differs from relational databases where user must define a schema for each database table before user can use it and where user must explicitly change that schema before user can store data differently.
- In SimpleDB, there is no schema requirement. Although user still have to consider the format of data, this approach has the benefit of freeing from the time it takes to manage schema modifications.
- The lack of schema means that there are no data types; all data values are treated as variable length character data. As a result, there is literally nothing extra to do if user want to add a new field to an existing database. Just add the new field to whichever data items require it. There is no rule that forces every data item to have the same fields.
- The drawbacks of a schema-less database include the lack of automatic integrity checking in the database and an increased burden on the application to handle formatting and type conversions.

3.4.4.5 Amazon CloudFront

- Amazon CloudFront is a content delivery web service (CDN). It integrates with other AWS Cloud services to give developers and businesses an easy way to distribute content to users across the world with low latency, high data transfer speeds, and no minimum usage commitments.
- Amazon CloudFront uses RTMP protocol for video streaming and HTTP or HTTPS for web content. Content delivery networks are suited for delivery of bulky data, like video streaming, downloading larger files and software, and to make website access faster.
- Amazon CloudFront is a pay-as-you-go model that can easily be integrated with all Amazon Web Services.
- Amazon CloudFront operates by caching the instance of each object on its different CDN locations, therefore reducing the time it takes to deliver content.
- Amazon CloudFront accesses the data from Amazon S3 through supported application programming interfaces and places it in regional data buckets.

Advantages :

1. No server hardware infrastructure to set up or maintain
2. No up-front investment in software licenses
3. No long-term commitment
4. Global delivery using CloudFront
5. Pay for what you use
6. Easy to get started with self service management console

3.5 Amazon Database Service

- Amazon Relational Database Service (RDS) is a web service that makes it easy to set up, operate and scale a relational database in the cloud. RDS gives access to the capabilities of a familiar MySQL, Oracle or Microsoft SQL Server database engine. Code, applications and tools already used with existing databases can be used with RDS.
- Amazon RDS is just a replacement to running your own database server. With a simple sequence of commands at the console, you can choose from two commercial DBMS (Oracle or Microsoft SQL Server) or two open source DBMS (MySQL or PostgreSQL).
- Amazon RDS automatically patches the database software and backs up the database, storing the backups for a user-defined retention period and enabling point-in-time recovery.

- Amazon RDS provides scaling the compute resources or storage capacity associated with the Database Instance. Pay only for the resources actually consumed, based on the DB Instance hours consumed, database storage, backup storage and data transfer.
- On-Demand DB Instances let you pay for compute capacity by the hour with no long-term commitments. Reserved DB Instances give the option to make a low, one-time payment for each DB Instance and in turn receive a significant discount on the hourly usage charge for that DB Instance.
- Amazon RDS provides you six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle and Microsoft SQL Server.
- Create the actual RDS instance :

```
instance = conn.create_dbinstance ( id=identifier, instance_class="db." + instance_class,
allocated_storage=allocated_storage,
engine=engine, db_name=db_name, master_username=master_username,
master_password=master_password,
db_subnet_group_name=identifier, vpc_security_groups=security_groups, port=port)
```

Advantages :

- Fast and easy to administer : Amazon RDS makes it easy to go from project conception to deployment.
- Available and durable : Amazon RDS runs on the same highly reliable infrastructure used by other Amazon Web Services.
- Highly scalable : You can scale your database's compute and storage resources.
- Secure : Amazon RDS makes it easy to control network access to your database.

3.5.1 Dynamo DB

- DynamoDB is a fast, fully managed NoSQL database service that makes it simple and cost-effective to store and retrieve any amount of data and serve any level of request traffic.
- All data items are stored on Solid State Drives (SSDs) and are replicated across 3 Availability Zones for high availability and durability.
- DynamoDB tables do not have fixed schemas and each item may have a different number of attributes.
- DynamoDB has no upfront costs and implements a pay as you go plan as a flat hourly rate based on the capacity reserved.
- With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance

degradation and use the AWS management console to monitor resource utilization and performance metrics.

- DynamoDB allows you to delete expired items from tables automatically to help you reduce storage usage and the cost of storing data that is no longer relevant.
- The first step in accessing DynamoDB is to create a connection to the service :

```
import boto.dynamodb  
conn = boto.dynamodb.connect_to_region('us-west-2',  
aws_access_key_id='<YOUR_AWS_KEY_ID>',  
aws_secret_access_key='<YOUR_AWS_SECRET_KEY>')
```

- Python program for creating a DynamoDB table with other operation :

```
import boto.dynamodb  
from boto.dynamodb.condition import *  
connection = boto.dynamodb.connect_to_region('eu-west-1')  
table = connection.get_table('table')
```

```
id = '1'  
timestamp = 1234  
atrrs = {  
    'key1': 'value1',  
    'key2': set(['value2', 'value3'])  
}
```

```
# create  
item = table.new_item(hash_key=id, range_key=timestamp, attrs=atrrs)  
item.put()
```

```
# read  
item = table.get_item(hash_key=id)  
key2 = list(item['key2'])
```

```
# update  
item['key1'] = 'foo'  
item['key3'] = 'bar'  
item.put()
```

```
# query  
table.query(hash_key=id, range_key_condition=LT(1500))
```

```
# scan  
table.scan(scan_filter={'key1': EQ('foo'))}
```

```
# delete  
item = table.get_item(hash_key=id)  
item.delete()
```

Advantages :

1. **Flexible** : Amazon DynamoDB supports both document and key-value data structures.
2. **Fully managed** : Amazon DynamoDB is a fully managed cloud NoSQL database service.
3. **Highly scalable** : When you create a table, simply specify how much request capacity you require.
4. **Event-driven programming** : Amazon DynamoDB integrates with AWS lambda to provide triggers

3.5.2 Difference between DynamoDB and Amazon S3

DynamoDB	Amazon S3
DynamoDB is database	S3 is file storage
Used in semi structured data	It is used for unstructured data
Size limit is 400 kB	Size limit is 5 TB
DynamoDb supports two kinds of primary keys, Partition Key and Sort key.	S3 uses unique Ids called Keys to retrieve files from the bucket
DynamoDb is used to store key-value. It uses items and attributes for its tables	S3 stores files in a flat organisation of containers called Buckets

3.6 Google AppEngine**SPPU : April-18**

- Google App Engine (GAE) is a Platform as a Service cloud computing platform for developing and hosting web applications in Google-managed data centers.
- Google App Engine is a way to write your own web applications and have them hosted on Google servers. It enables developers to build their web applications on the same scalable system that power Google applications.
- An app is a piece of software which can run on the computer, internet, phone or any other electronic device. Google refers to their online services as Apps. They also sell a specific suite of services known as Google Apps.
- Google's providing both SaaS and PaaS solutions in cloud computing. Some of the examples for SaaS solutions including Google Apps which including Gmail, Doc, etc. and PaaS includes Google App engine.
- Services provided by App engine includes :
 - a) Platform as a Service (PaaS) to build and deploy scalable applications.
 - b) Hosting facility in fully-managed data centers.

- c) A fully-managed, flexible environment platform for managing application server and infrastructure.
- d) Support in the form of popular development languages and developer tools.
- Major feature of Google App Engine :
 1. Automatic scaling and load balancing.
 2. Authentication using Google Accounts API.
 3. Provides dynamic web services based on common standards.
 4. Integration with other Google Cloud Services and API.
 5. Support persistent storage, with query access sorting and transaction management features.
- Google App engine offers users the ability to build and host web applications on Google's infrastructure.

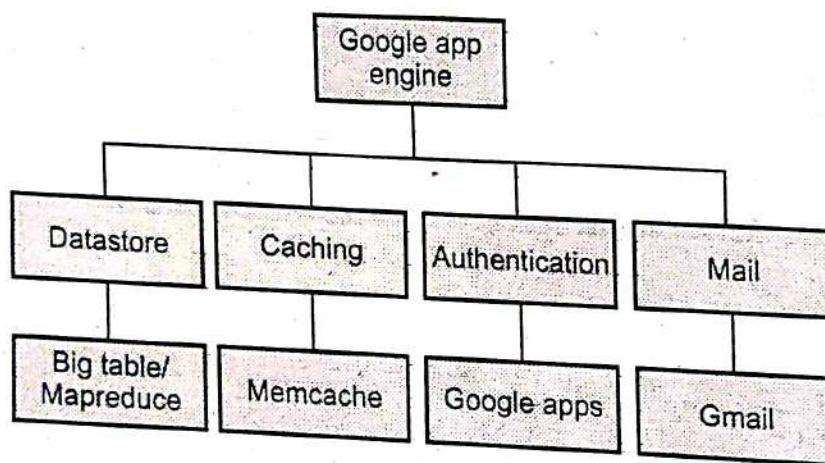


Fig. 3.6.1

- The App Engine offers a number of services that enable you to perform several common operations when managing your application. The following APIs are available to access these services :
 1. **Mail** : Using the mail API, the developers can send email messages.
 2. **Memcache** : The Memcache service gives the users the benefit of working efficiently by providing high retrieval speed, even when multiple users access the same application at the same instance of time.
 3. **Image manipulation** : The Image service allows you to manipulate images of your application. With the use of this API, you can resize, crop, rotate and flip images in JPEG and PNG formats.
- In the PaaS space Google is a key player. App Engine is a platform to create, store and run applications on Google's servers using development languages as java and python.

- App Engine includes tools for managing the data store, monitoring the site and its resource consumption and debugging and logging. A user can serve the app from his own domain name using Google Apps.

• Key features of GAE programming mode using java and python :

- The Google App engine Software Development Kit (SDK) provides Java and Python programming languages.
- The languages have their own web server application that contains all Google App Engine services on a local computer. The web server also simulates a secure sandbox environment.
- The Google App engine SDK has APIs and libraries including the tools to upload applications. The architecture defines the structure of applications that run on the Google App engine.

1. Python :

- The Google App engine allows implementation of applications using python programming language and running them on its interpreter.
- The Google App engine provides rich APIs and tools for designing web applications, data modeling, managing, accessing apps data, support for mature libraries and frameworks like Django.
- The main characteristics of Google App engine are its DataStore, configuration file app.yaml and how it serves an application.

2. Java :

- The Google App engine provides tools and APIs required for the development of web applications that run on the Google App engine Java run time.
- The application interacts with the environment using servlets and web technologies like Java Server Pages (JSPs) which can be developed using Java6.
- The GAE environment uses Java SE Runtime JRE platform 6 and libraries which the applications can access using APIs.
- Java SDK has implementations for Java Data Objects (JDO) and Java Persistence (JPA) interface.
- To exchange email messages with Google App engine, it provides the Google App Engine mail service through the Java Mail API.
- Support for other languages like JavaScript, Ruby or Scala is also provided by Google App engine with the use of JVM compatible compilers and interpreters.
- When Google App engine gets a web request that corresponds to the URL mentioned in the applications deployment descriptor it invokes a servlet

corresponding to the request and uses Java Servlets API to provide requested data and accepts response data.

- Google App engine makes it easy to build an applications that runs reliably, even under heavy load and with large amounts of data.
- App engine includes the below features :
 - a) Dynamic web serving, with full support for common web technologies.
 - b) Persistent storage with queries, sorting and transactions.
 - c) Automatic scaling and load balancing.
 - d) APIs for authenticating users and sending email using Google accounts.
 - e) Scheduled tasks for triggering events at specified times and regular intervals.

Review Question

1. Draw and explain programming environment of Google App Engine.

SPPU : April-18 In Sem, Marks 4

3.7 Microsoft Azure

SPPU : April-18

- Windows Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft - managed data centers.
- Azure queue storage is a service for storing large numbers of messages that can be accessed from anywhere in the world via authenticated calls using HTTP or HTTPS. A single queue message can be up to 64 KB in size, and a queue can contain millions of messages, up to the total capacity limit of a storage account.
- Azure is a virtualized infrastructure to which a set of additional enterprise services has been layered on top, including, a virtualization service called Azure AppFabric that creates an application hosting environment. AppFabric is a cloud-enabled version of the .NET framework.
- Windows Azure is Microsoft's application platform for the public Cloud. Applications can be deployed on to Azure in various models
- Windows Azure is used to :
 1. Build a web application that runs and stores its data in Microsoft data centers.
 2. Store data while the applications that consume this data run on premise (outside the public Cloud).
 3. Create virtual machines to develop and test, or run SharePoint and other out-of-the-box applications.

- 4. Develop massively scalable applications with many users.
- 5. Offer a wide range of services
- Azure has three components : compute, storage and fabric
 1. **Compute** : Windows Azure provides a hosting environment for managed code. It provides a computation service through roles. Windows Azure supports three types of roles :
 - a) Web roles used for web application programming and supported by IIS7.
 - b) Worker roles are also used for background processing of web roles.
 - c) Virtual Machine (VM) roles are generally used for migrating windows server applications to Windows Azure in an easy way.
 2. **Storage** : Windows Azure provides storage in the cloud. It provides four different types of storage services :
 - a) Queues for messaging between web roles and worker roles.
 - b) Tables for storing structural data.
 - c) BLOBs (Binary Large Objects) to store text, files or large data.
 - d) Windows Azure Drives (VHD) to mount a page blob. They can easily be downloaded and uploaded via blobs.
 3. AppFabric provides infrastructure services for developing, deploying and managing Windows Azure application. It provides five services: Service bus, Access, Caching, Integration and Composite.
- Fig. 3.7.1 shows Windows Azure platform architecture.

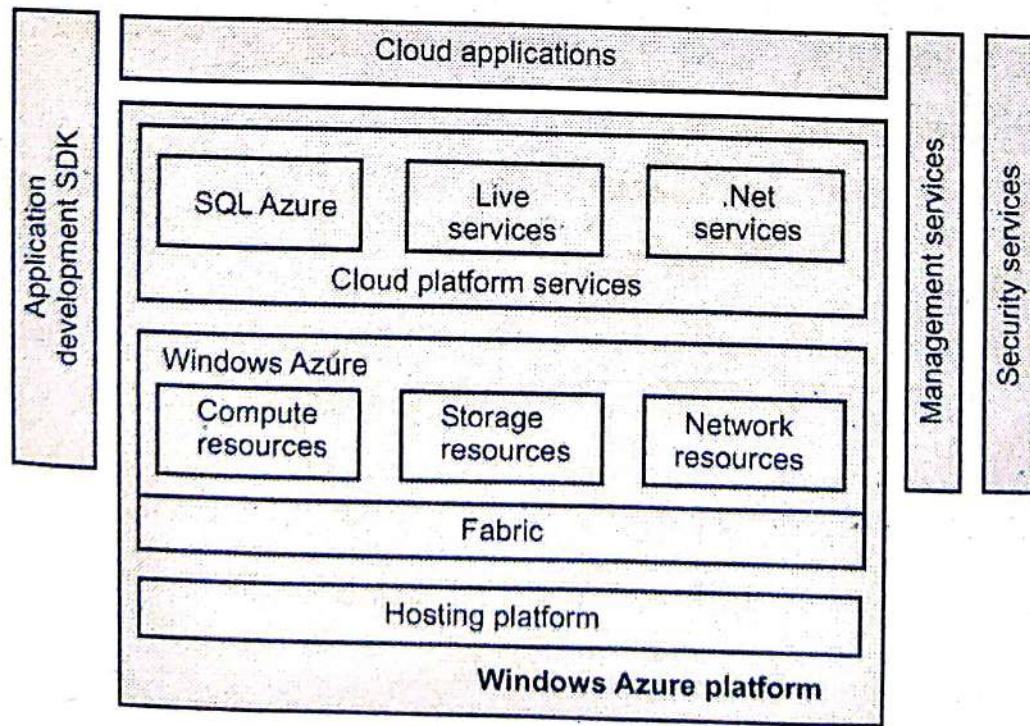


Fig. 3.7.1 Windows Azure platform architecture

- Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying and managing applications and services through a global network of Microsoft-managed data centers.
- It provides software as a service (SaaS), platform as a service and infrastructure as a service and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems.
- Windows Azure provides resources and services for consumers. For example, hardware is abstracted and exposed as compute resources.
- Physical storage is abstracted as storage resources and exposed through very well-defined interfaces.
- A common windows fabric abstracts the hardware and the software and exposes virtual compute and storage resources.
- Each instance of an application is automatically managed and monitored for availability and scalability.
- If an application goes down, the Fabric is notified and a new instance of the application is created. Because virtualization is a key element in cloud computing, no assumption must be made on the state of the underlying hardware hosting the application.
- Advantages of Microsoft Azure
 1. Microsoft Azure offers high availability
 2. It offers you a strong security profile
 3. It is a cost-effective solution for an IT budget.
 4. Azure allows you to use any framework, language, or tool.
 5. Azure allows businesses to build a hybrid infrastructure.

3.7.1 SQL Azure

- SQL Azure is Microsoft's cloud database service. Based on SQL Server database technology and built on Microsoft's Windows Azure cloud computing platform.
- SQL Azure Database is the relational database service on the Azure Services Platform. Data is hosted, managed and provisioned in Microsoft data centers.
- The SQL Azure database service offers a scalable and distributed database hosted in the cloud, and therefore highly available. Using SQL Azure, you can easily provision and deploy relational database solutions.
- Azure SQL Database scales both horizontally and vertically. Users can scale up instances and increase performance with a premium availability model, or scale