

Unit-II AWS

Cloud Computing and DevOps

Ms. Vidya S. Gaikwad
vidya.gaikwad@viit.ac.in

Department of Computer Engineering



BRACT's, Vishwakarma Institute of Information Technology, Pune-48

**(An Autonomous Institute affiliated to Savitribai Phule Pune University)
(NBA and NAAC accredited, ISO 9001:2015 certified)**

Contents

- Cloud Fundamentals:
 - What is Cloud?
 - Properties of cloud
 - Benefits of using cloud,
 - Service models
 - Deployment models
- AWS Accounts –
 - The Basics, Creating an AWS Account
 - Securing An AWS Account
 - Creating a Budget
 - IAM
 - Creating Access keys and setting up AWS CLI v2 tools
 - Simple Storage Service (S3)

Contents

- VIRTUAL PRIVATE CLOUD (VPC) BASICS
- ELASTIC COMPUTE CLOUD (EC2) BASICS
- Infrastructure as Code (CloudFormation)
- Load Balancing
- Systems Manager
- AWS Lambda

- Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing.
- Instead of buying, owning, and maintaining physical data centres and servers, we can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider.

- Who is using cloud computing?
- Organizations of every type, size, and industry are using the cloud for a wide variety of use cases, such as data backup, disaster recovery, email, virtual desktops, software development and testing, big data analytics, and customer-facing web applications.
- For example, healthcare companies are using the cloud to develop more personalized treatments for patients.
- Financial services companies are using the cloud to power real-time fraud detection and prevention.
- Video game makers are using the cloud to deliver online games to millions of players around the world.

Benefits of Cloud

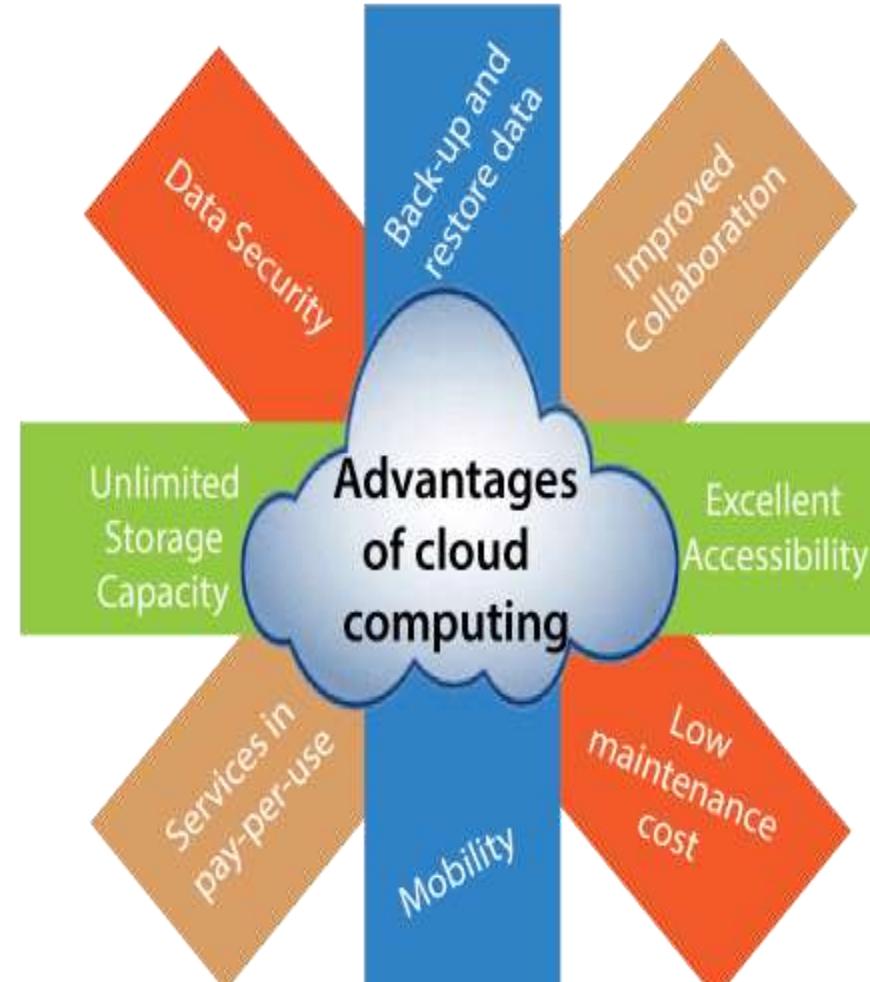


Figure 1: Benefits of Cloud [1]

1) Back-up and restore data

Once the data is stored in the cloud, it is easier to get back-up and restore that data using the cloud.

2) Improved collaboration

Cloud applications improve collaboration by allowing groups of people to quickly and easily share information in the cloud via shared storage.

3) Excellent accessibility

Cloud allows us to quickly and easily access store information anywhere, anytime in the whole world, using an internet connection. An internet cloud infrastructure increases organization productivity and efficiency by ensuring that our data is always accessible.

4) Low maintenance cost

Cloud computing reduces both hardware and software maintenance costs for organizations.

5) Mobility

Cloud computing allows us to easily access all cloud data via mobile.

6) Services in the pay-per-use model

Cloud computing offers Application Programming Interfaces (APIs) to the users for access services on the cloud and pays the charges as per the usage of service.

7) Unlimited storage capacity

Cloud offers us a huge amount of storing capacity for storing our important data such as documents, images, audio, video, etc. in one place.

8) Data security

Data security is one of the biggest advantages of cloud computing. Cloud offers many advanced features related to security and ensures that data is securely stored and handled.

What is AWS?

- AWS stands for **Amazon Web Services**.
- The AWS service is provided by the Amazon that uses distributed IT infrastructure to provide different IT resources available on demand.
- Provides different services such as infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS).

Pay-As-You-Go

Based on the concept of Pay-As-You-Go, AWS provides the services to the customers.

What is AWS?

- AWS provides services to customers when required without any prior commitment or upfront investment.
- Pay-As-You-Go enables the customers to procure services from AWS.
- Computing
- Programming models
- Database storage
- Networking



How to SignUp to the AWS platform

- Firstly visit the website <https://aws.amazon.com>.
- The following screen appears after opening the website, then click on the **Complete Sign Up** to create an account and fill the required details.



How to SignUp to the AWS platform

- The following screen appears after clicking on the "**Complete Sign Up**" button. If you are an already existing user of an AWS account, then enter the email address of your AWS account otherwise "**create an aws account**".
- On clicking on the "**create an aws account**" button, the following screen appears that requires some fields to be filled by the user.

The screenshot shows a web browser window with the URL <https://portal.aws.amazon.com/billing/signup#/start>. The title bar of the browser says "Create an AWS account". The page content includes:

- A banner at the top left states "AWS Accounts Include 12 Months of Free Tier Access" and provides a link to "aws.amazon.com/free".
- Input fields for "Email address" (gakshita123@gmail.com), "Password", and "Confirm password".
- An input field for "AWS account name" with the value "Akshita Gupta".
- A large yellow "Continue" button.
- A link "Sign in to an existing AWS account".
- Small text at the bottom right: "© 2010 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy | Terms of Use".

How to SignUp to the AWS platform

- Now, fill your contact information.

The screenshot shows the 'Contact Information' step of the AWS sign-up process. At the top right, there's a language selection dropdown set to 'English'. Below it, a note says 'All fields are required.' The form asks for account type (Professional or Personal), full name (Akhila Gupta), phone number (7842176966), country/region (India), address (sector -14 block c), city (Faridabad), state/province (Haryana), and postal code (121006). A note states: '* If you select India, your country/region selection cannot be changed after creating the account.' Below the address fields, there's a section about the Amazon Internet Services Pvt. Ltd. Customer Agreement, mentioning AISPL and its role as the local seller for AWS services in India. A checkbox is present for accepting the terms of the agreement. At the bottom is a yellow 'Create Account and Continue' button.

Contact Information

All fields are required.

Please select the account type and complete the fields below with your contact details.

Account type Professional Personal

Full name

Phone number

Country/Region

* If you select India, your country/region selection cannot be changed after creating the account.

Address
Apartment, suite, unit, building, floor, etc.

City

State / Province or region

Postal code

Amazon Internet Services Pvt. Ltd. Customer Agreement
Customers with an India contact address are now required to contract with Amazon Internet Services Private Ltd. (AISPL). AISPL is the local seller for AWS infrastructure services in India.

Check here to indicate that you have read and agree to the terms of the [AISPL Customer Agreement](#).

Create Account and Continue

© 2012 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
[Privacy Policy](#) | [Terms of Use](#) | [Sign Out](#)

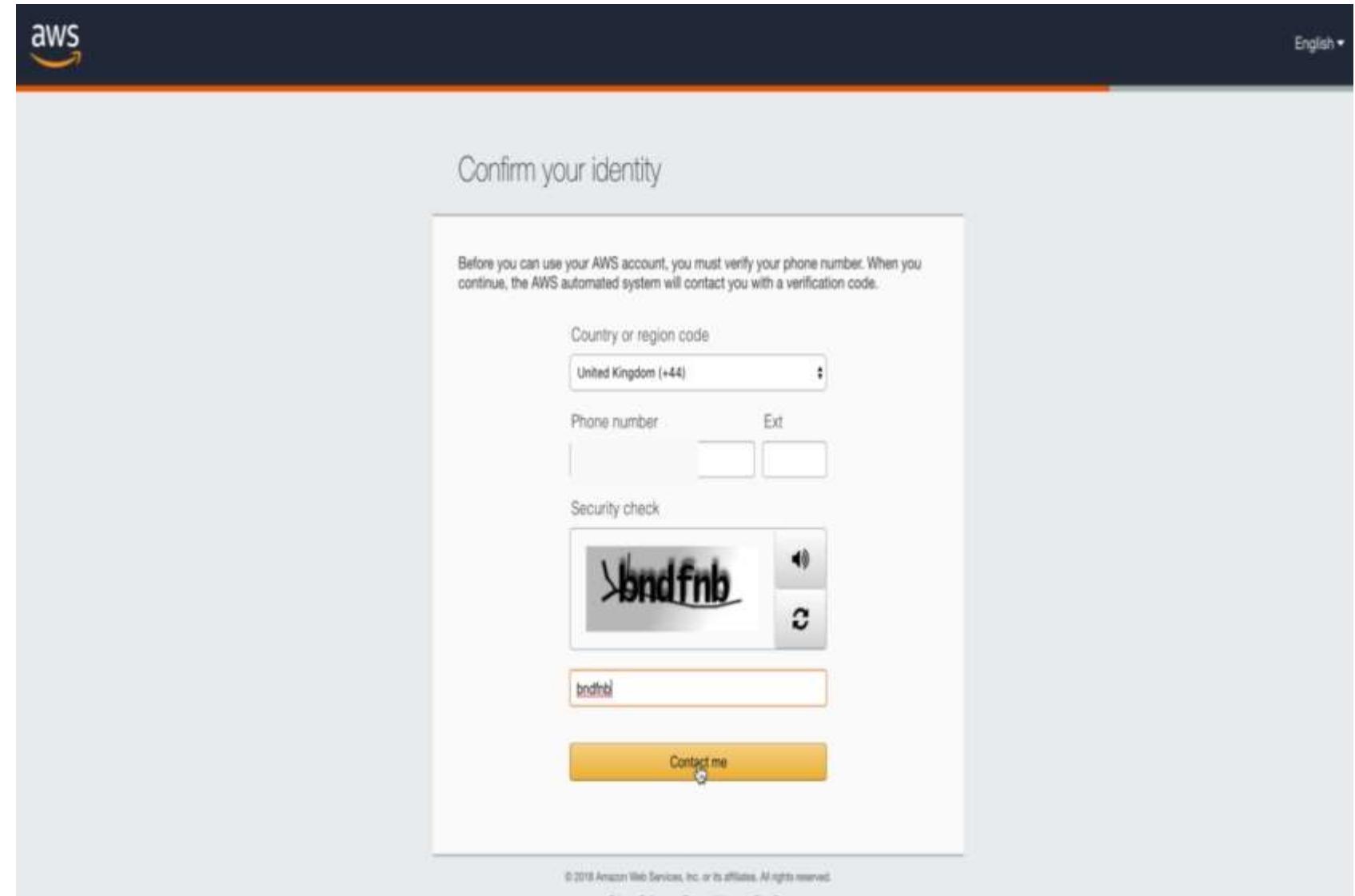
How to SignUp to the AWS platform

- After providing the contact information, fill your payment information.

The screenshot shows the 'Payment Information' step of the AWS sign-up process. At the top, the AWS logo is visible, and the text 'Payment Information' is centered. Below this, a note states: 'Please type your payment information so we can verify your identity. We will not charge you unless your usage exceeds the [AWS Free Tier Limits](#). Review [frequently asked questions](#) for more information.' A callout box provides additional information about a \$2 charge during verification: 'As part of our card verification process we will charge INR 2 on your card when you click the "Secure Submit" button below. This will be refunded once your card has been validated. Your bank may take 3-5 business days to show the refund. Mastercard/Visa customers may be redirected to your bank website to authorize the charge.' The form includes fields for 'Credit/Debit card number' (an input field with a placeholder '|'), 'Expiration date' (two dropdown menus showing '01' and '2019'), and 'Cardholder's name' (an input field).

How to SignUp to the AWS platform

- After providing your payment information, confirm your identity by entering your phone number and security check code, and then click on the "Contact me" button.
-



How to SignUp to the AWS platform

- AWS will contact you to verify whether the provided contact number is correct or not.

Call in progress...

Please answer the call from AWS and, when prompted, enter the 4-digit number on your phone keypad.

3 0 8 7



How to SignUp to the AWS platform

- When number is verified, then the following message appears on the screen.
- The final step is the confirmation step. Click on the link to log in again; it redirects you to the "**Management Console**".



What is IAM?

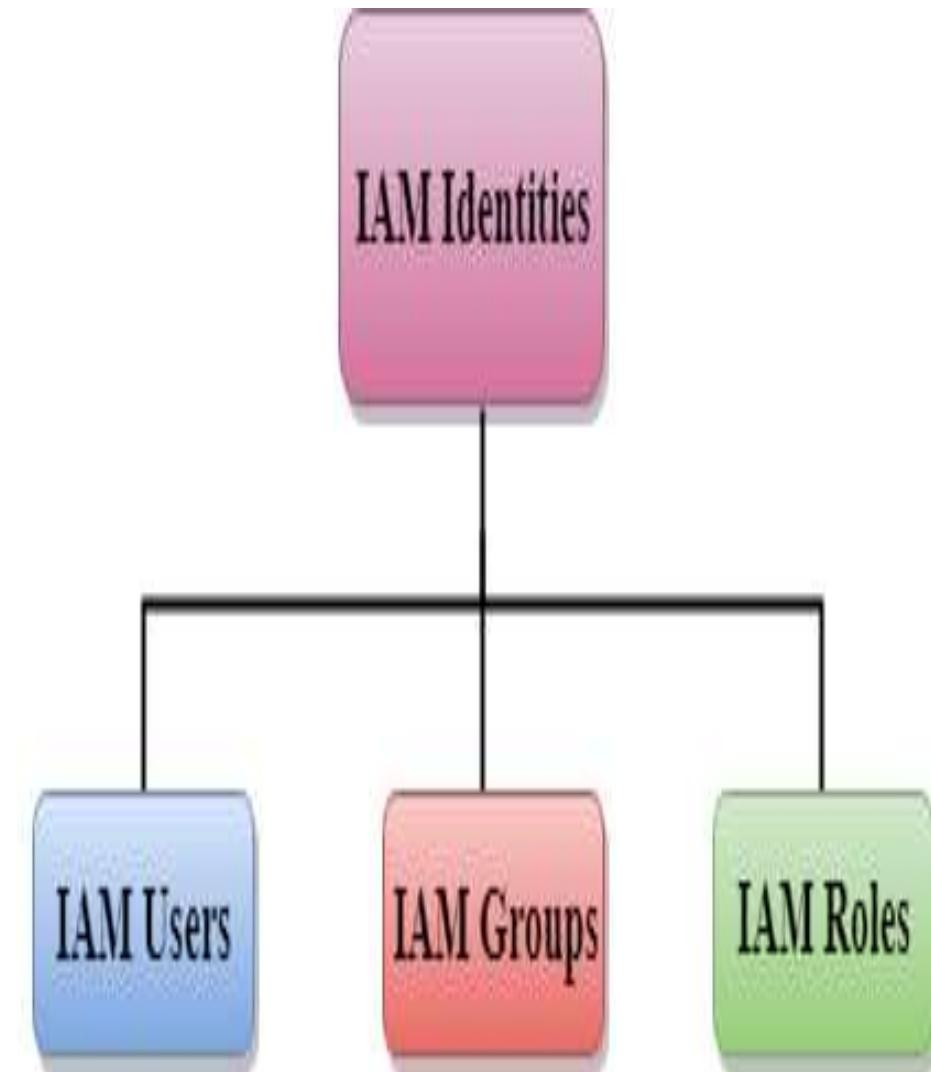
- IAM stands for **Identity Access Management**.
- IAM allows you **to manage users and their level of access to the aws console**.
- It is **used to set users, permissions and roles**.
- It allows you to **grant access to the different parts of the aws platform**.
- AWS Identity and Access Management is a web service that enables Amazon Web Services (AWS) customers **to manage users and user permissions in AWS**.
- With IAM, Organizations can centrally manage users, security credentials such as **access keys, and permissions that control which AWS resources users can access**.
- Without IAM, Organizations with multiple users must either create multiple user accounts, each with its own billing and subscriptions to AWS products or share an account with a single security credential.

What is IAM?

- Without IAM, you also **don't have control about the tasks that the users can do.**
- IAM enables the organization to **create multiple users, each with its own security credentials, controlled and billed to a single aws account.**
- IAM allows the user to do only what they need to do as a part of the user's job.

IAM Identities

- IAM identities are created to provide authentication for people and processes in your aws account.
- IAM Users
- IAM Groups
- IAM Roles



- **AWS Account Root User**
- When you first create an AWS account, you create an account as a root user identity which is used to sign in to AWS.
- You can sign to the AWS Management Console by entering your email address and password.
- The combination of email address and password is known as **root user credentials**.
- When you sign in to AWS account as a root user, you have unrestricted access to all the resources in AWS account.
- The Root user can also access the billing information as well as can change the password also.

- **IAM Users**
- An IAM User is an entity created in AWS that provides a way to interact with AWS resources.
- Purpose of IAM Users is that they can sign in to the AWS Management Console and can make requests to the AWS services.
- Newly created **IAM users** have no password and no access key.
- Newly created IAM Users do not have permissions, i.e., they are not authorized to access the AWS resources.
- Advantage of using individual IAM Users is that you can assign the permissions individually.
- We can even assign the administrative permissions, who can administer your AWS resources and also administer other IAM Users.
- Each IAM User is associated with one and only one AWS account

- **IAM Group**
- An IAM Group is a collection of users.
- Group specifies the permission for a collection of users, and it also makes it possible to manage the permissions easily for those users.
- **IAM Role**
- A role is a set of permissions that grant access to actions and resources in AWS. These permissions are attached to the role, not to an IAM User or a group.
- A role is not uniquely associated with a single person; it can be used by anyone who needs it.
- A role does not have long term security credential, i.e., password or security key.

- **What is the AWS Command Line Interface?**
- The AWS Command Line Interface (AWS CLI) is an **open source tool** that enables you to interact with AWS services using commands in your command-line shell.
- The AWS CLI enables you to start running commands that implement functionality equivalent to that provided by the browser-based AWS Management Console from the command prompt in your terminal.

- AWS offers a wide range of storage services that can be provisioned depending on your project requirements and use case.
- AWS storage services have different provisions for highly confidential data, frequently accessed data, and the not so frequently accessed data.
- You can choose from various storage types namely, ***object storage, file storage, block storage services, backups, and data migration options.***

What is S3? [3]

- S3 stands for **Simple Storage Service**.
- S3 provides developers and IT teams with **secure, durable, highly scalable object storage**.
- It is easy to use with a simple web services interface to store and retrieve any amount of data from anywhere on the web.
- It is **Object-based storage**, i.e., you can **store the images, word files, pdf files**, etc.
- The files which are stored in S3 can be from **0 Bytes to 5 TB**.
- It has unlimited storage means that you can store the data as much you want.
- Files are stored in **Bucket**.
- **A bucket is like a folder available in S3 that stores the files**.
- S3 is a universal namespace, i.e., the names must be unique globally.
- Bucket contains a **DNS address**.
- The bucket must contain a **unique name to generate a unique DNS address**.
- There is a limit of **100 buckets per AWS accounts**. But it can be increased if requested from AWS support.

What is S3? [3]

- If you create a bucket, URL look like:

<https://s3-eu-west-1.amazonaws.com/acloudguru>



Region name



Bucket name

AWS S3 Terminology [3]

- **Bucket Owner:** The person or organization that owns a particular bucket is its *bucket owner*.
- **Import/Export Station:** A machine that uploads or downloads data to/from S3.
- **Key:** Key, in S3, is a unique identifier for an object in a bucket. For example in a bucket ‘ABC’ your *GFG.java* file is stored at *javaPrograms/GFG.java* then ‘*javaPrograms/GFG.java*’ is your object key for *GFG.java*.
- It is important to note that ‘bucketName+key’ is unique for all objects.
- Only one object for a key in a bucket.
- If you upload 2 files with the same key.
- The file uploaded latest will overwrite the previously contained file.

AWS S3 Terminology [3]

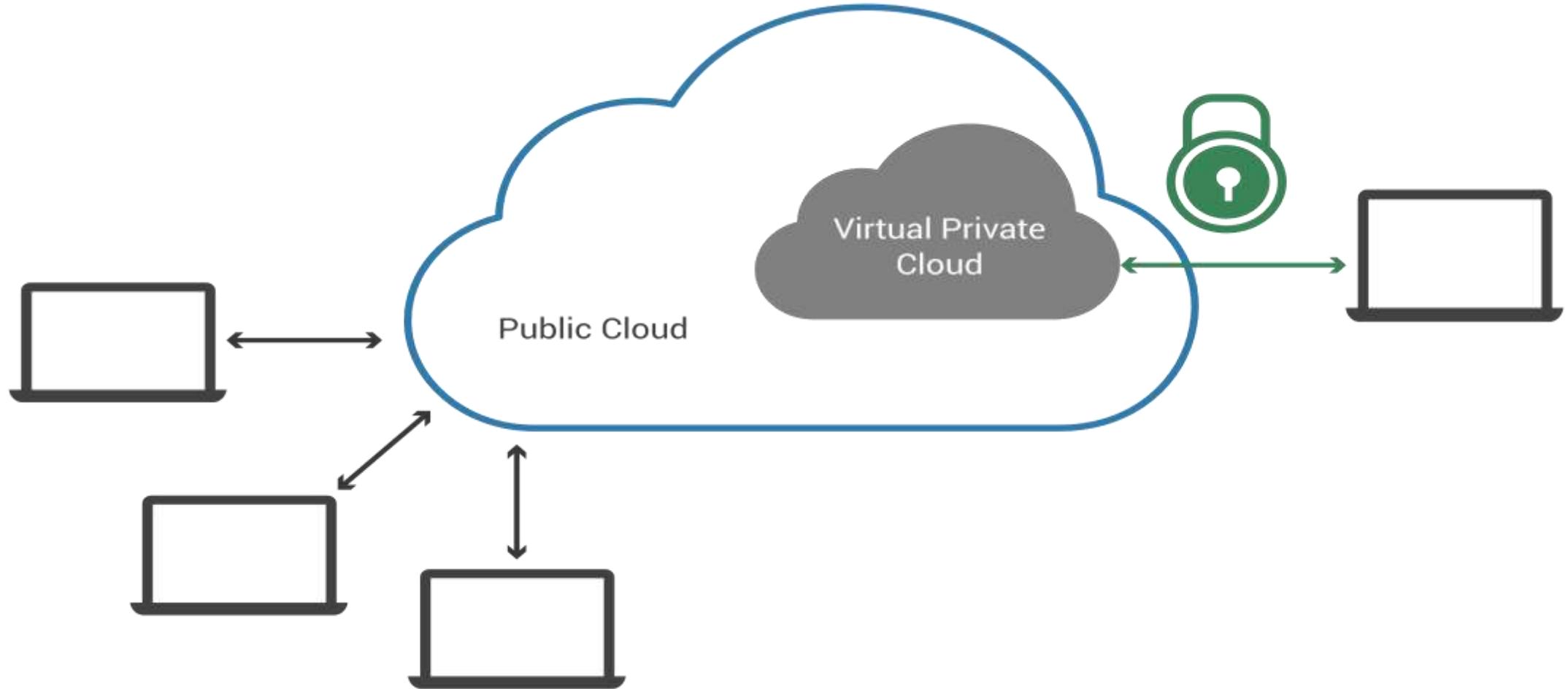
- **Versioning:**
- Versioning means to always **keep a record of previously uploaded files in S3**.
- Versioning is not enabled by default. Once enabled, it is enabled for all objects in a bucket.
- Versioning keeps all the copies of your file, so, it adds cost for storing multiple copies of your data.
- For example, 10 copies of a file of size 1GB will have you charged for using 10GBs for S3 space.
- Versioning is helpful to prevent unintended overwrites and deletions.
- Note that objects with the same key can be stored in a bucket if versioning is enabled (since they have a unique version ID).

AWS S3 Terminology

- **null Object:**
- Version ID for objects in a bucket where versioning is suspended is null.
- Such objects may be referred to as null objects.
- For buckets with versioning enabled, each version of a file has a specific version ID.
- **Object:** Fundamental entity type stored in AWS S3.
- **Access Control Lists (ACL):**
- A document for verifying the access to S3 buckets from outside your AWS account.
- Each bucket has its own ACL.
- **Bucket Policies:**
- A document for verifying the access to S3 buckets from within your AWS account, this controls which services and users have what kind of access to your S3 bucket.
- Each bucket has its own Bucket Policies.

- **Lifecycle Rules:**
- This is a cost-saving practice that can move your files to AWS Glacier (The AWS Data Archive Service) or to some other S3 storage class for cheaper storage of old data or completely delete the data after the specified time.
- **Features of AWS S3:**
- **Durability**
- **Availability**
- **Server-Side-Encryption (SSE):** AWS S3 supports three types of SSE models:
 - **SSE-S3:** AWS S3 manages encryption keys.
 - **SSE-C:** The customer manages encryption keys.
 - **SSE-KMS:** The AWS Key Management Service (KMS) manages the encryption keys.
 - **AWS-S3** is region-specific.

What is a virtual private cloud (VPC)?[6]



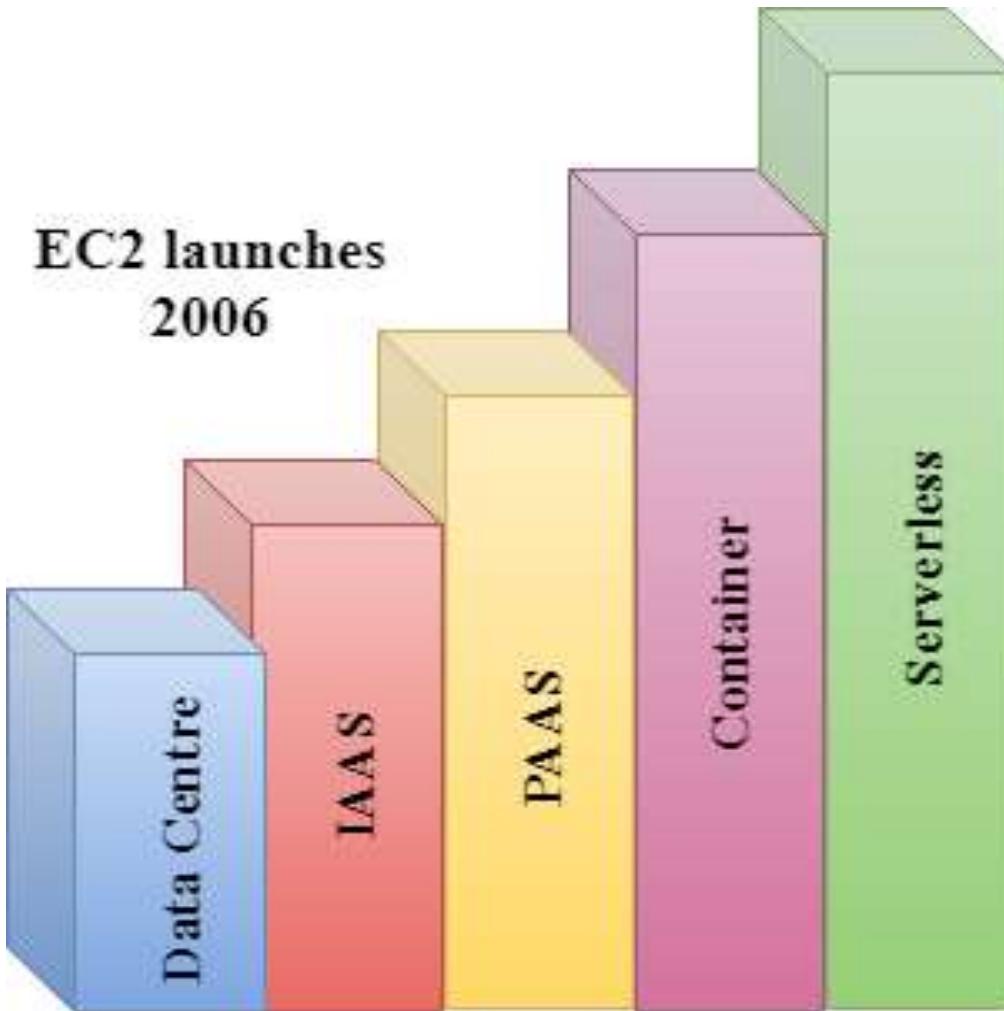
What is a virtual private cloud (VPC)? [6]

- A virtual private cloud (VPC) is a secure, isolated private cloud hosted within a public cloud.
- VPC customers can run code, store data, host websites, and do anything else they could do in an ordinary private cloud, but the private cloud is hosted remotely by a public cloud provider.
- VPCs combine the scalability and convenience of public cloud computing with the data isolation of private cloud computing.

What is a virtual private cloud (VPC)? [6]

- **What is a public cloud? What is a private cloud?**
- A public cloud is shared cloud infrastructure.
- Multiple customers of the cloud vendor access that same infrastructure, although their data is not shared – just like every person in a restaurant orders from the same kitchen, but they get different dishes.
- Public cloud service providers include AWS, Google Cloud Platform, and Microsoft Azure, among others.
- The technical term for multiple separate customers accessing the same cloud infrastructure is "multitenancy" (see [What Is Multitenancy?](#) to learn more).
- A private cloud, however, is single-tenant.
- A private cloud is a cloud service that is exclusively offered to one organization.
- A virtual private cloud (VPC) is a private cloud within a public cloud; no one else shares the VPC with the VPC customer.

What is Lambda?



- **Lambda:**
- Amazon released Lambda in 2015.
- You **do not have to take care of managing Data centre, managing infrastructure as a service, managing platform as a service or container.**
- You need to upload the code and Amazon will do everything for you.

What is Lambda?

- Lambda is used to encapsulate Data centres, Hardware, Assembly code/Protocols, high-level languages, operating systems, AWS APIs.
- Lambda is a compute service where you can upload your code and create the Lambda function.
- Lambda takes care of provisioning and managing the servers used to run the code.
- While using Lambda, you don't have to worry about scaling, patching, operating systems, etc.

How does AWS Lambda work? [4]



AWS Lambda Block Diagram

How does AWS Lambda work? [4]

- **Step 1:** First upload your AWS Lambda code in any language supported by AWS Lambda. [Java](#), Python, Go, and C# are some of the languages that are supported by AWS Lambda function.
- **Step 2:** These are some AWS services which allow you to trigger AWS Lambda.
- **Step 3:** AWS Lambda helps you to upload code and the event details on which it should be triggered.
- **Step 4:** Executes AWS Lambda Code when it is triggered by AWS services:
- **Step 5:** AWS charges only when the AWS lambda code executes, and not otherwise.

How does AWS Lambda work? [4]

- This will happen in the following scenarios:
 - Upload files in an S3 bucket
 - When HTTP get/post endpoint URL is hit
 - For adding/modifying and deleting Dynamo DB tables
 - In the process of data streams collection
 - Push notification
 - Hosting of website
 - Email sending

How does AWS Lambda work? [4]

- **Events that Trigger AWS Lambda**
- Here, are Events which will be triggered when you use AWS Lambda.
- Insert, updating and deleting data Dynamo DB table
- To include push notifications in SNS
- To search for log history in CloudTrail
- Entry into an S3 object
- DynamoDB can trigger AWS Lambda whenever there is data added, modified, and deleted in the table.
- Helps you to schedule the event to carry out the task at regular time pattern.
- Modifications to objects in S3 buckets
- Notifications sent from Amazon SNS.
- AWS Lambda can be used to process the CloudTrail logs
- API Gateway allows you to trigger AWS Lambda on GET/POST methods.

What is Elastic Compute Cloud (EC2)?

- EC2 stands for Elastic Compute Cloud.
- EC2 is on-demand computing service on the AWS cloud platform. U
- It also allows the user to configure their instances as per their requirements i.e. allocate the RAM, ROM, and storage according to the need of the current task
- EC2 has resizable capacity current task.
- EC2 offers security, reliability, high-performance and cost-effective infrastructure so as to meet the demanding business needs.

Features of Amazon EC2:

- **Functionality** : Amazon EC2 itself comes with a set of default AMI(Amazon Machine Image) options supporting various operating systems along with some pre-configured resources like RAM, ROM, storage, etc.
- **Operating Systems** – Amazon EC2 includes a wide range of operating systems to choose from while selecting your AMI.
- Currently, AWS has the following most preferred set of operating systems available on the EC2 console.
- **Amazon Linux, Windows Server, Ubuntu Server, SUSE Linux, Red Hat Linux**

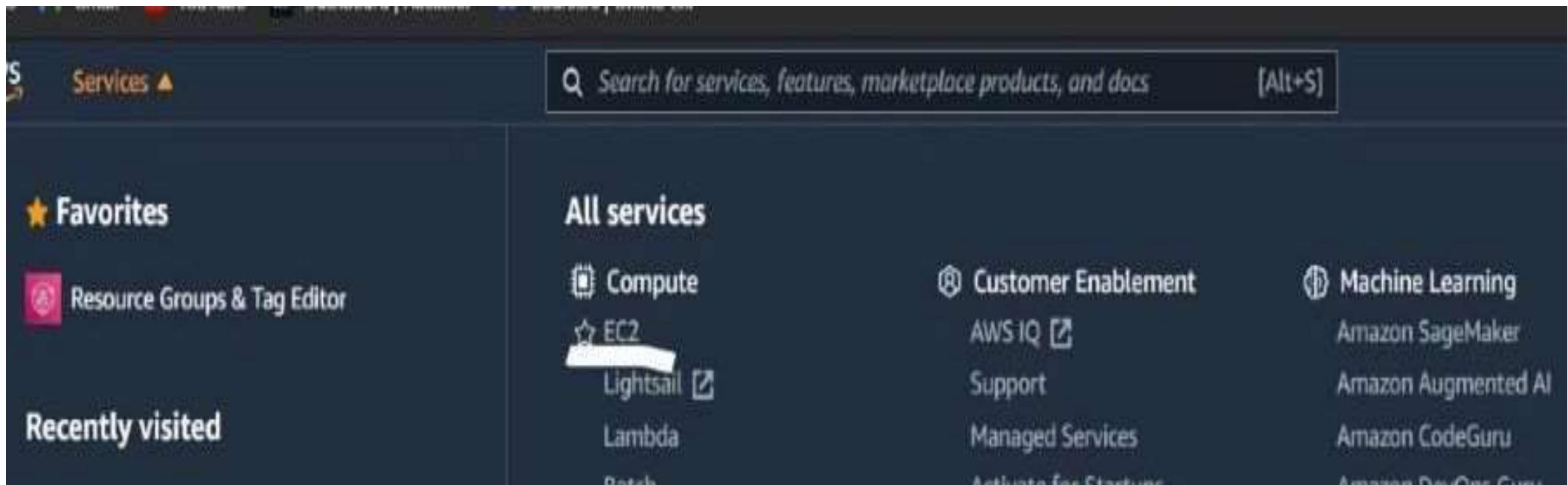


Features of Amazon EC2:

- **Software :**
- It allows its users to choose from various software present to run on their EC2 machines.
- Numerous software like SAP, LAMP and Drupal, etc are available on AWS to use.
- **Scalability and Reliability** – EC2 provides us the facility to scale up or scale down as per the needs.

Features of Amazon EC2:

- First login into your AWS account.
- Once you are directed to the management console. From the left click on “Services” and from the listed options click on EC2.



Features of Amazon EC2:

- Afterward, you will be redirected to the EC2 console. Here is the image attached to refer to various features in EC2.

The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar menu includes 'New EC2 Experience' (with a 'Tell us what you think' link), 'EC2 Dashboard' (marked as 'New'), 'Events', 'Tags', 'Limits', 'Instances' (with sub-options like 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances' (marked as 'New'), 'Dedicated Hosts', 'Capacity Reservations'), and 'Images' (with 'AMIs'). The main content area has a blue header 'Welcome to the new EC2 console!' with a message about the redesign. Below it, the 'Resources' section displays a grid of metrics: Instances (running) 0, Dedicated Hosts 0, Elastic IPs 0, Instances 1, Key pairs 1, Load balancers 0, Placement groups 0, Security groups 3, Snapshots 0, and Volumes 0. To the right, the 'Account attributes' section lists 'Supported platforms' (VPC), 'Default VPC' (vpc-7d4a8f16), 'Settings', 'EBS encryption', 'Zones', 'Default credit specification', and 'Console experiments'. At the bottom, there's a callout for 'Microsoft SQL Server Always On availability groups'.

How to Launch a WordPress Website using Amazon EC2 Server ?

- Setting up WordPress website using an EC2 instance:
 1. Launch an EC2 instance:
- Go to your [AWS console](#) and sign in with your credentials.
- After login to the AWS account, select the EC2 service from the list of all services.

The screenshot shows the AWS Management Console homepage. On the left, there's a sidebar titled "AWS services" with sections for "Find Services" (a search bar), "Recently visited services" (empty), and "All services". Under "Compute", "EC2" is highlighted with a green border. Other services listed under Compute include Lightsail, ECR, ECS, EKS, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, and EC2 Image Builder. Under "Storage", S3, EFS, FSx, S3 Glacier, and Storage Gateway are listed. The main content area shows various AWS services: Blockchain (Amazon Managed Blockchain), Satellite (Ground Station), Quantum Technologies (Amazon Braket), Management & Governance (AWS Organizations, CloudWatch, AWS Auto Scaling, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog), Security, Identity, & Compliance (IAM, Resource Access Manager, Cognito, Secrets Manager, GuardDuty, Inspector, Amazon Macie, AWS Single Sign-On, Certificate Manager, Key Management Service, CloudHSM, Directory Service, WAF & Shield, Artifact, Security Hub, Detective). To the right, there are sections for "Access resources on the go" (AWS Console Mobile App), "Explore AWS" (Amazon Redshift, Run Serverless Containers with AWS Fargate, Scalable, Durable, Secure Backup & Restore with Amazon S3), and "AWS Marketplace".

How to Launch a WordPress Website using Amazon EC2 Server ?

- Click the **Launch instance** button to create an instance.

The screenshot shows the AWS EC2 Dashboard in the US East (Ohio) Region. The left sidebar includes links for New EC2 Experience, EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area displays EC2 Resources: Running instances (0), Dedicated Hosts (0), Volumes (1), Key pairs (3), Placement groups (0), Elastic IPs (0), Snapshots (0), Load balancers (0), and Security groups (4). A callout box provides information about launching Microsoft SQL Server Always On availability groups. The 'Launch instance' button is highlighted with a green border at the bottom left of the dashboard. The 'Service health' section indicates the service is operating normally in the US East (Ohio) Region.

How to Launch a WordPress Website using Amazon EC2 Server ?

- **2. Configuring EC2 instance:**
- Search for WordPress in the marketplace tab (left side) in the dashboard and select the instance named as WordPress Certified by Bitnami and Automattic.

The screenshot shows the AWS Marketplace search results for "wordpress". The search bar at the top contains "wordpress". Below it, a navigation bar has tabs: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. The "1. Choose AMI" tab is selected. The main content area displays search results. A green box highlights the first result, "WordPress Certified by Bitnami and Automattic". This item is marked as "Free tier eligible". It has a 4.5-star rating from 114 reviews. The description states: "WordPress is the world's most popular content management platform. It includes the new Gutenberg editor and over 45,000 themes and plugins. This image is certified by Bitnami as secure, up-to-date, and packaged using industry best practices, and approved by Automattic, the experts behind WordPress." A "Select" button is visible. Below this, two more results are shown: "WordPress with NGINX and SSL Certified by Bitnami and Automattic" and "WordPress Multisite Certified by Bitnami and Automattic", each with its own "Select" button. On the left sidebar, there are filters for "Categories" (All Categories, Infrastructure-Software, DevOps, Business Applications, Industries), "Operating System" (All Linux/Unix), and "AWS Marketplace" (128 products). The "AWS Marketplace" filter is currently selected and highlighted with a green box.

How to Launch a WordPress Website using Amazon EC2 Server ?

- Now you'll see the pricing details in which you have to click continue.

Infrastructure as Code – What is it and Why is it important?

- Managing IT infrastructure was a manual process.
- People would physically put servers in place and configure them.
- Applications would be deployed only after the machines were configured to the correct settings required by the OS and applications.
- Unsurprisingly, this manual process would often result in several problems such as follows –
 - cost
 - scalability
 - availability
 - inconsistency

Infrastructure as Code – What is it and Why is it important?[7,8]

- Infrastructure as code (IaC) tools allow you to **manage infrastructure with configuration files rather than through a graphical user interface**.
- IaC allows you to **build, change, and manage your infrastructure** in a safe, consistent, and repeatable way by **defining resource configurations that you can version, reuse, and share**[8].
- Infrastructure as code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.
- The IT infrastructure managed by this comprises both physical equipment's such as bare-metal servers as well as virtual machines and associated configuration resources[7].

How IaC Works?

- The imperative approach “gives orders”.
- It defines a sequence of commands or instructions so the infrastructure can reach the final result.
- A declarative approach, on the other hand, “declares” the desired outcome.
- Instead of explicitly outlining the sequence of steps the infrastructure needs to reach the final result, the declarative approach shows what the final result looks like.

Types of IaC

Imperative

Declarative

Benefits of infrastructure as code

- **Speed**
- With Infrastructure as code, you can quickly set up your complete infrastructure by running a script.
- **Consistency**
- Manual processes sometimes result in mistakes. Manual infrastructure management will result in discrepancies, no matter how hard you try. IaC solves that problem.
- **Accountability**
- This one is quick and easy. Since you can version IaC configuration files like any source code file, you have full traceability of the changes each configuration suffered.



Benefits of infrastructure as code

- **Increased Efficiency**
- IaC can make the entire software development lifecycle more efficient. By employing infrastructure as code, you can deploy your infrastructure architectures in many stages.
- **Lower Cost**
- Lowering the costs of infrastructure management. By employing cloud computing along with IaC, you dramatically reduce your costs.
- Won't have to spend money on hardware, hire people to operate it, and build or rent physical space to store it.



Infrastructure-as-code tools AWS CloudFormation[7]



- CloudFormation permits users to model their infrastructure within a JSON or YAML template file.
- The service also adds automation features to help you with the deployment of resources in a repeatable and manageable way, and you only pay for the resources that you use.
- With the template configured to your application specifications, CloudFormation will take care of the rest of the tasks for you.
- The use of plaintext is very handy.
- Both YAML or JSON are supported, and it is easy to establish a secure infrastructure model at any complexity level from the many templates available from CloudFormation.

Azure Resource Manager[7]



- Using this tool, users are able to provision infrastructure and handle dependencies in one seamless cycle through Azure Resource Manager templates (ARM templates).

Google Cloud Deployment Manager [7]



- This tool bases its execution on config files such as YAML and templates (JINJA2 or PYTHON) all within the Google Cloud Platform.



- Terraform automation has various shapes and is orchestrated in varying degrees with the focus on the core plan/apply cycle.
- Some teams run Terraform locally but they use wrapper scripts to set up a consistent working directory for Terraform to run in.
- Other development teams may also run Terraform entirely within an alternate orchestration tool such as Jenkins.
- It is by far the most adaptable tool on this list but subsequently potentially intimidating, to begin with at least.

**CHEF**™

- Chef is a quite popular IaC tool among CI/CD practitioners.
- It uses Ruby-based DSL and this is certainly a huge plus.
- It has “cookbook” versioning from the beginning and allows you to maintain a consistent configuration.
- This is possible even when the infrastructure needs to keep up with the rapid growth of the app it hosts.
- Chef gives recipes and cookbooks at the heart of its configuration.
- These are self-styled appellations for templates and collections of templates that you can use out of the box.



- Ansible is a tool designed with the perspective of automation from the start.
- This tool focuses on providing “radically simple” configuration language as well as being able to manage cloud instances immediately with no modifications.
- It is also useful for performing arbitrary IT orchestration such as zero downtime rolling updates, hotfixes, and so on as opposed to being configuration management specific.
- Rather than managing systems as individual units, you just describe how components and the system in general interact with each other and Ansible will handle the rest.



- This tool runs the data centers for several significant companies like Reddit, Dell, and Google and runs on all OS systems.

Load Balancing in Cloud Computing

- Load balancing is the method that allows you to have a proper balance of the amount of work being done on different pieces of device or hardware equipment.
- The load of the devices is balanced between different servers or between the CPU and hard drives in a single cloud server.
- Benefit : To improve the speed and performance of each single device, and the other is to protect individual devices from hitting their limits by reducing their performance.
- Load balancing is beneficial with almost any type of service, such as HTTP, SMTP, DNS, FTP, and POP/IMAP.
- It also increases reliability through redundancy. A dedicated hardware device or program provides the balancing service.

Load Balancing in Cloud Computing

- Different Types of Load Balancing Algorithms in Cloud Computing:

1. Static Algorithm

- Static algorithms are built for systems with very little variation in load.
- The entire traffic is divided equally between the servers in the static algorithm.
- This algorithm requires in-depth knowledge of server resources for better performance of the processor, which is determined at the beginning of the implementation.
- The decision of load shifting does not depend on the current state of the system.

2. Dynamic Algorithm

- First finds the lightest server in the entire network and gives it priority for load balancing.
- This requires real-time communication with the network which can help increase the system's traffic. Here, the current state of the system is used to control the load.

3. Round Robin Algorithm

- Uses round-robin method to assign jobs.
- First, it randomly selects the first node and assigns tasks to other nodes in a round-robin manner.
- This is one of the easiest methods of load balancing.
- Processors assign each process circularly without defining any priority.
- It gives fast response in case of uniform workload distribution among the processes.
- All processes have different loading times.

4. Weighted Round Robin Load Balancing Algorithm

- Developed to enhance the most challenging issues of Round Robin Algorithms.

Load Balancing in Cloud Computing

- In this algorithm, there are a specified set of weights and functions, which are distributed according to the weight values.
- Processors that have a higher capacity are given a higher value.
- Therefore, the highest loaded servers will get more tasks.
- When the full load level is reached, the servers will receive stable traffic.

5. Opportunistic Load Balancing Algorithm

- Allows each node to be busy.
- It never considers the current workload of each system.
- OLB distributes all unfinished tasks to these nodes.

Types of Load Balancing

- Network Load Balancing
- HTTP(S) load balancing
- Internal Load Balancing
- Hardware Load Balancer
- Software Load Balancer
- Virtual Load Balancer

WHY CLOUD LOAD BALANCING IS IMPORTANT IN CLOUD COMPUTING?

- Offers better performance
- Helps Maintain Website Traffic
- Can Handle Sudden Bursts in Traffic
- Greater Flexibility

- Systems Manager is an AWS service that lets customers centrally manage their EC2 instances to ensure security and compliance are in place based on the organizational policies.
- Customers can also make use of the service to manage on-premises VM's as well as the VM's hosted on hybrid cloud environments.
- Systems Manager makes use of SSM agent, a piece of software that can be installed and configured on virtual machines running Linux, Windows, MacOS, and Raspbian operating systems.
- **How Systems Manager works**

References

- [1] <https://www.javatpoint.com/advantages-and-disadvantages-of-cloud-computing>
- [2] <https://aws.amazon.com/what-is-cloud-computing/>
- [3] <https://www.geeksforgeeks.org/introduction-to-aws-simple-storage-service-aws-s3/>
- [4] <https://www.guru99.com/aws-lambda-function.html>
- [5] <https://blog.vsoftconsulting.com/blog/aws-lambda-function-how-it-works-and-how-to-create-it>
- [6] <https://www.cloudflare.com/learning/cloud/what-is-a-virtual-private-cloud/>
- [7] <https://www.edureka.co/blog/infrastructure-as-code/>
- [8] <https://developer.hashicorp.com/terraform/tutorials/aws-get-started/infrastructure-as-code>
- [9] <https://www.javatpoint.com/load-balancing-in-cloud-computing>

Presentation Topic

Unit IV - Amazon Web Services

Vishal Ambadas Meshram
vishal.meshram@viit.ac.in

Department of Computer Engineering



BRACT'S, Vishwakarma Institute of Information Technology, Pune-48

(An Autonomous Institute affiliated to Savitribai Phule Pune University)
(NBA and NAAC accredited, ISO 9001:2015 certified)





Cloud Computing:

The practice of using network of remote servers hosted on the Internet to store, manage, and process data rather than a local server or a personal computer.

Cloud Concept



On-Premise

1. You own the Server.
2. You hire IT people.
3. You pay or rent the real-estate.
4. You take all the risk.

Cloud



1. Someone else owns the Server.
2. Someone else hire IT people.
3. Someone else pay or rent the real-estate.
4. You are responsible for configuring your cloud services & code.
Someone else takes care of the rest.

Cloud Computing Models

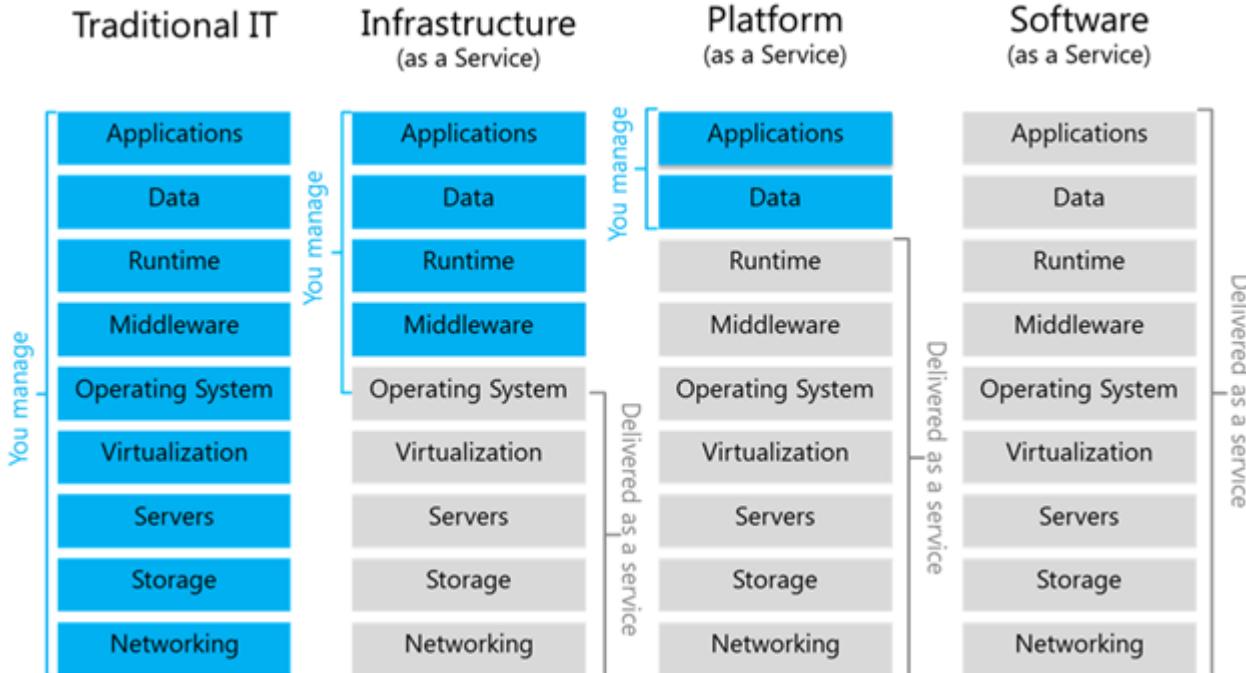


Fig Ref : <https://dachou.github.io/2018/09/28/cloud-service-models.html>

Cloud Computing Deployment Models

Cloud

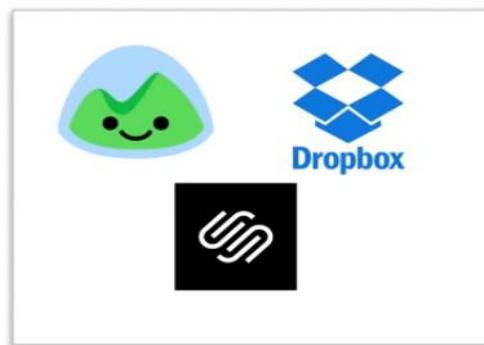
Fully utilizing cloud computing

Hybrid

Using both Cloud and On-Premise

On-Premise

Deploying resources on-premises, using virtualization and resource management tools, is sometimes called “private cloud”.



- Startups
- SaaS offerings
- New projects and companies



- Banks
- FinTech, Investment Management
- Large Professional Service providers
- Legacy on-premise



- Public Sector eg. Government ACTIVATE WINDOWS
Go to Settings to activate Windows.
- Super Sensitive Data eg. Hospitals
- Large Enterprise with heavy regulation eg. Insurance Companies

Contents:

- **What is AWS**
- Why are enterprises choosing AWS?
- AWS Account setup and billing alarms
- Compute services
- Storage services (S3 Buckets, EBS)
- Database services (RDS, Dynamo DB)
- Elastic Load Balancer (ELB)
- IAM

What is AWS?

aws

Link: https://www.youtube.com/watch?v=a9__D53WsUs

Contents:

- What is AWS
- Why are enterprises choosing AWS?
- AWS Account setup and billing alarms
- Compute services
- Storage services (S3 Buckets, EBS)
- Database services (RDS, Dynamo DB)
- Elastic Load Balancer (ELB)
- IAM

AWS Global Infrastructure Map

AWS serves over a million active customers in more than **190 countries**. AWS now spans **77 Availability Zones** within **24 geographic regions** around the world and has announced plans for 18 more Availability Zones and 6 more AWS Regions in Australia, India, Indonesia, Japan, Spain, and Switzerland.



AWS Global Infrastructure Map

Regions

- Physical Location in the world with multiple Availability Zones (AZ's).
- It's a geographical distinct location.
- Every region is physically isolated from and independent of every other region in terms of location, power, water supply.
- Each region has at least 2 AZ's
- AWS largest region is "**US-EAST**" (**Northern Virginia**)
- US-EAST1 is the region where you see all your billing information.

Availability Zones

- Availability Zones consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities.
- AZ's 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.
- AZ's are represented by region code, followed by a letter identifier. Ex. "us-east-1a".
- "MultiAZ", distributing your instance across multiple AZ's allows failover configuration for handling request when goes down.

Edge Locations

- **An Edge location is a data center owned by a trusted partner of AWS which has a direct connection to the AWS network.**
- These location serve requests for CloudFront & Route 53. Requests going to either of these services will be routed to the nearest edge location automatically.
- S3 transfer acceleration traffic & API Gateway endpoint traffic also use the AWS Edge network.

“Customers are increasingly choosing AWS to host their cloud-based infrastructure and realize increased performance, security, reliability, and scale wherever they go. For the tenth year in a row, AWS is evaluated as a Leader in the [2020 Gartner Magic Quadrant for Cloud Infrastructure and Platform Services](#), placed highest in both axes of measurement—Ability to Execute and Completeness of Vision—among the top 7 vendors named in the report.”



Figure 1. Magic Quadrant for Cloud Infrastructure and Platform Services



Gartner, Magic Quadrant for Cloud Infrastructure & Platform Services, Raj Bala, Bob Gill, Dennis Smith, David Wright, Kevin Ji, 1 September 2020. This graphic was published by Gartner, Inc. as part of a larger research document and should be evaluated in the context of the entire document. The Gartner document is available upon request from AWS. Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

Lower Costs with AWS Up-Front and Increase Savings as Your Usage Grows

1

Replace up-front capital expense with low variable cost

“Average of 400 servers replaced per customer”

2

Economies of scale have allowed us to consistently lower costs

44 Price Reductions

3

Pricing model choice to support variable & stable workloads

On-demand
Reserved
Spot

4

Save more money as you grow bigger

Tiered Pricing
Volume Discounts



Source: IDC Whitepaper, sponsored by Amazon, “The Business Value of Amazon Web Services Accelerates Over Time.” July 2012

4X More Reliable & 1/4 the Cost of On-Premises Infrastructure

END USERS BENEFITED FROM FEWER SERVICE DISRUPTIONS AND QUICKER RECOVERY ON AMAZON CLOUD INFRASTRUCTURE,
REDUCING DOWNTIME BY 72%

AMAZON CLOUD INFRASTRUCTURE REPRESENTS A
70% SAVINGS COMPARED WITH ON-PREMISE SOLUTIONS



WHITE PAPER

The Business Value of Amazon Web Services Accelerates Over Time

Sponsored by: Amazon

Randy Perry Stephen D. Hendrick
July 2012

EXECUTIVE SUMMARY

In early 2012, IDC interviewed 11 organizations that deployed applications on Amazon cloud infrastructure services. The purpose of the IDC analysis was to understand the economic impact of Amazon cloud infrastructure services over time, beyond the well-documented benefit of reduction in capex and opex. Specifically,

IDC set out to understand the long-term economic implications of moving workloads onto Amazon cloud infrastructure services, the impact of moving applications on developer productivity and business agility, and the new opportunities that businesses could address by moving resources onto Amazon cloud infrastructure services. The organizations interviewed ranged from small and medium-sized companies to companies with as many as 160,000 employees.

Organizations in our study had been Amazon Web Services (AWS) customers for as few as seven months to as many as 5.3 years. Our interviews were designed to elicit both quantifiable information and anecdotes so that IDC could interpret the full return-on-investment (ROI) impact of Amazon cloud infrastructure services on these organizations. The study represents a broad range of

Business Value Highlights: Applications Running on AWS

- Five-year ROI: 625%
- Payback period: 7.1 months
- Software development productivity increase: 507%
- Average savings per application: \$518,990
- Downtime reduction: 72%
- IT productivity increase: 52%
- Five-year TCO savings: 70%



Architected for Enterprise Security Requirements

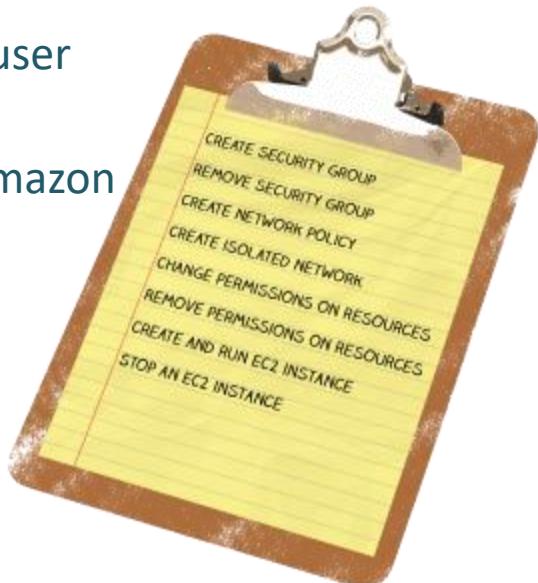
Certifications and accreditations for workloads that matter



AWS CloudTrail - AWS API call logging for governance & compliance

Log and review user activity

Stores data in Amazon S3, or archive to Amazon Glacier



Increased agility has become the
#1 reason businesses use the AWS
cloud



Enterprises Can't Afford to be Slow

Old World: Infrastructure in Weeks



AWS: Infrastructure in Minutes



- Add New Dev Environment
- Add New Prod Environment
- Add New Environment in Japan
- Add 1,000 Servers
- Remove 1,000 servers
- Deploy 2 PB Data warehouse
- Shut down 2 PB Data warehouse

Everything changes with this kind of agility

A Culture of Innovation: Experiment Often and Fail Without Risk



On-Premises

Experiment Infrequently

Failure is expensive

Less Innovation



Experiment Often

Fail quickly at a low cost

More Innovation

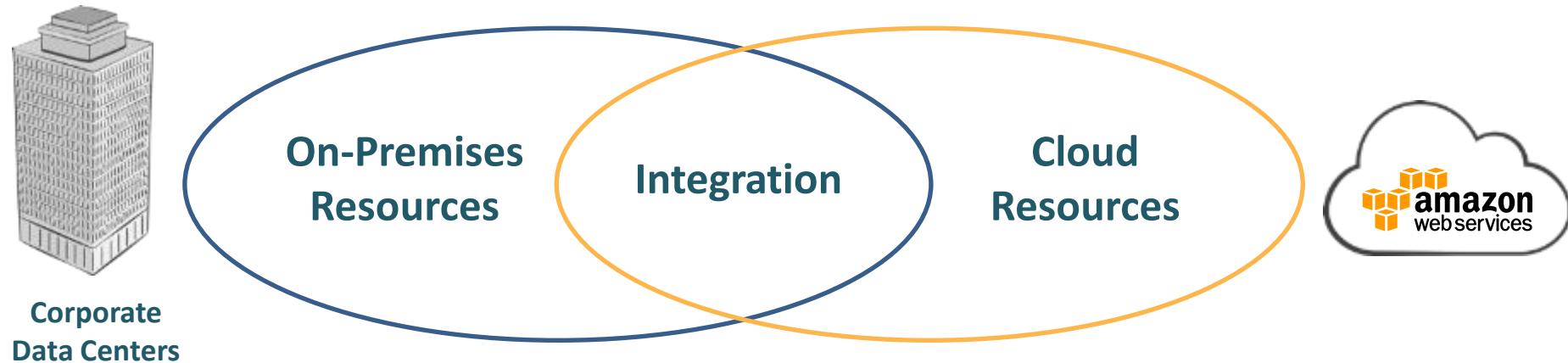
Many Enterprises Worry That These are the Only Two Choices

Build a
“private”
cloud

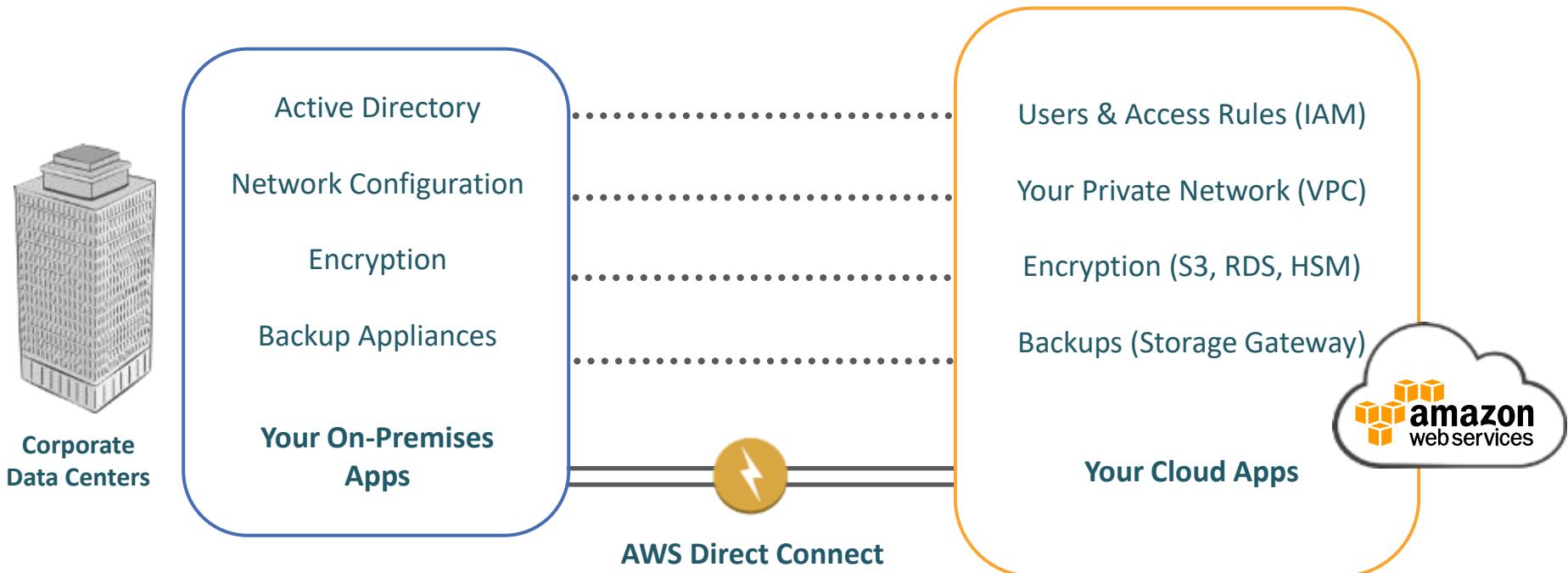


Rip everything out
and move to
AWS

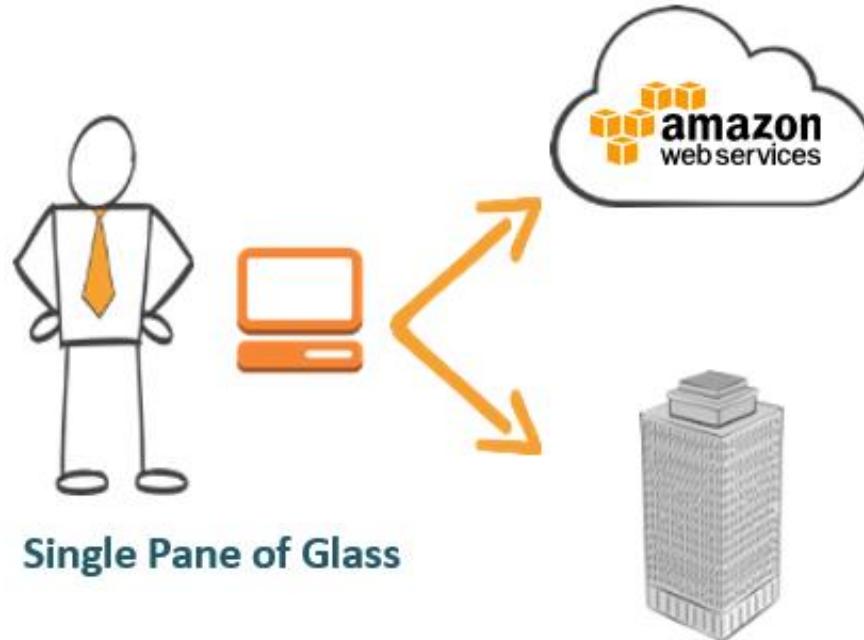
The Good News is that Cloud isn't an 'All or Nothing' Choice



Integrating AWS with Your Existing On-Premises Infrastructure



Tools to Help Customers Manage Resources Across Environments



Enterprises Use Cases on AWS

**Enterprise Apps
and Dev./Test**

**Big Data and
HPC**

**Storage, Backup
and Archival**

**Web, Mobile, and
Social Apps**

**Disaster
Recovery**

**Virtual
Desktops**

Contents:

- What is AWS
- Why are enterprises choosing AWS?
- **AWS Account setup and billing alarms**
- Compute services
- Storage services (S3 Buckets, EBS)
- Database services (RDS, Dynamo DB)
- Elastic Load Balancer (ELB)
- IAM

AWS billing alarms

1. Billing Preferences

2. Budget

3. Cloud Watch

Screenshot of the AWS Billing & Cost Management Dashboard.

The left sidebar shows navigation links: Home, Cost Management, Cost Explorer, **Budgets**, Budgets Reports, Savings Plans, Cost & Usage Reports, Cost Categories, Cost allocation tags, Billing, Bills, Orders and invoices, Credits, Purchase orders, Preferences, and **Billing preferences**.

The main content area displays the "Billing & Cost Management Dashboard". It includes a "Spend Summary" section showing a balance of **0.00 INR** at today's exchange rate of 73.48255. A bar chart shows a value of **\$1.59**. To the right is a donut chart with a value of **\$0**.

The top navigation bar includes the AWS logo, Services dropdown, search bar ("Search for services, features, marketplace products, and docs [Alt+S]"), and user information (vishal, Global dropdown, Support dropdown).

The "Month-to-Date Spend by Service" chart is partially visible on the right.

The "Global" dropdown menu is highlighted with a red box and contains options: My Account (933126026785), My Organization, My Service Quotas, **My Billing Dashboard** (highlighted with a red box), My Security Credentials, and Sign Out.

Contents:

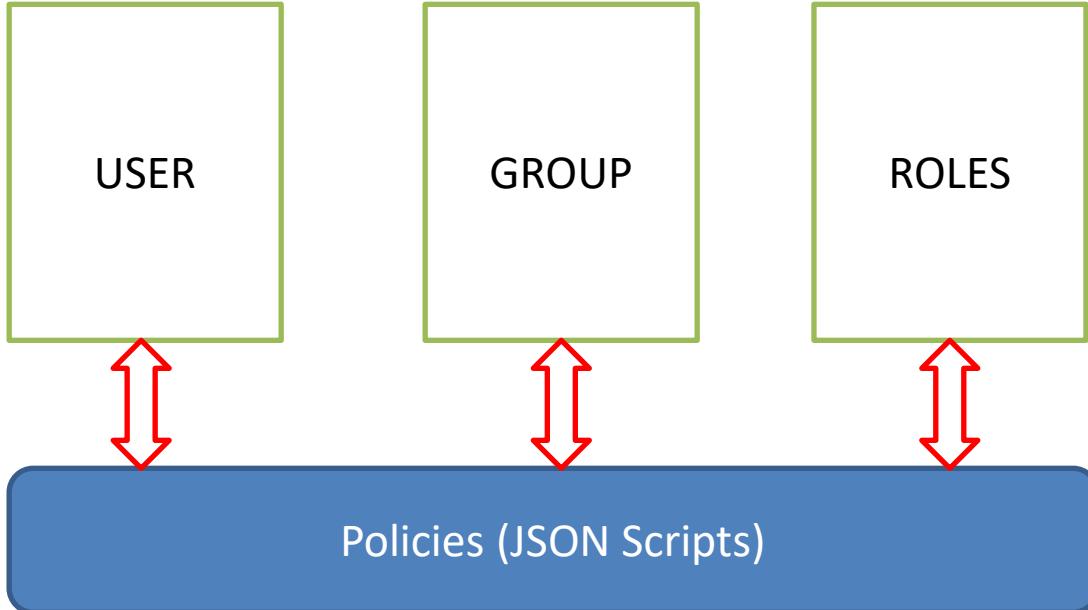
- What is AWS
- Why are enterprises choosing AWS?
- AWS Account setup and billing alarms
- **IAM Introduction**
- Compute services (EC2)
- Storage services (S3 Buckets, EBS)
- Database services (RDS, Dynamo DB)
- Elastic Load Balancer (ELB)

IAM Introduction

- IAM stands for Identity Management Service.
- IAM is global service.
- Instead of using root account create a IAM user.
- Never share your root or IAM credentials.
- Policies are written in JSON (JavaScript Object Notation)

The screenshot shows the AWS IAM dashboard. A red box highlights the left sidebar menu under 'Identity and Access Management (IAM)'. Another red box highlights the 'Sign-in URL for IAM users in this account' field, which contains the URL <https://viit.signin.aws.amazon.com/console>. A third red box highlights the 'Best practices' section at the bottom of the dashboard, which lists several recommendations:

- Grant least privilege access: Establishing a principle of least privilege ensures that identities are only permitted to perform the most minimal set of functions necessary to fulfill a specific task, while balancing usability and efficiency.
- Use AWS Organizations: Centrally manage and govern your environment as you scale your AWS resources. Easily create new AWS accounts, group accounts to organize your workflows, and apply policies to accounts or groups for governance.
- Enable Identity federation: Manage users and access across multiple services from your preferred identity source. Using AWS Single Sign-On centrally manage access to multiple AWS accounts and provide users with single sign-on access to all their assigned accounts from one place.
- Enable MFA: For extra security, we recommend that you require multi-factor authentication (MFA) for all users.



The screenshot shows the AWS IAM Security Credentials page. The left sidebar lists various IAM management options like Dashboard, Access management, and Policies. A search bar at the top has 'iam' typed into it. The main content area is titled 'Your Security Credentials' and provides instructions for managing AWS credentials. It highlights 'Password' and 'Multi-factor authentication (MFA)' with red boxes. A vertical navigation bar on the right includes links for My Account, My Organization, My Service Quotas, My Billing Dashboard, and My Security Credentials, with the latter also highlighted by a red box. The 'Global' button in the top right is also outlined in red.

aws Services ▾

Identity and Access Management (IAM)

Dashboard

Access management

Groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Search IAM

Q iam X

vishal Global Support

Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, see [Managing IAM User Credentials](#).

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in AWS General Reference.

▼ Password

You use an email address and password to sign in to secure pages on AWS, such as the AWS Management Console, AWS Forum, and AWS CLI. Create a password that contains many characters, including numbers and punctuation. Store your password securely, do not share it, and do not reuse it.

[Click here](#) to change the password, name, or email address for your root AWS account.

▲ Multi-factor authentication (MFA)

▲ Access keys (access key ID and secret access key)

▲ CloudFront key pairs

▲ X.509 certificate

▲ Account identifiers

My Account 933126026785

My Organization

My Service Quotas

My Billing Dashboard

My Security Credentials

Sign Out

For MFA: Install “Google Authenticator” App.

Contents:

- What is AWS
- Why are enterprises choosing AWS?
- AWS Account setup and billing alarms
- IAM Introduction
- **Compute services (EC2)**
- Storage services (S3 Buckets, EBS)
- Database services (RDS, Dynamo DB)
- Elastic Load Balancer (ELB)

EC2 (Elastic Compute Cloud)

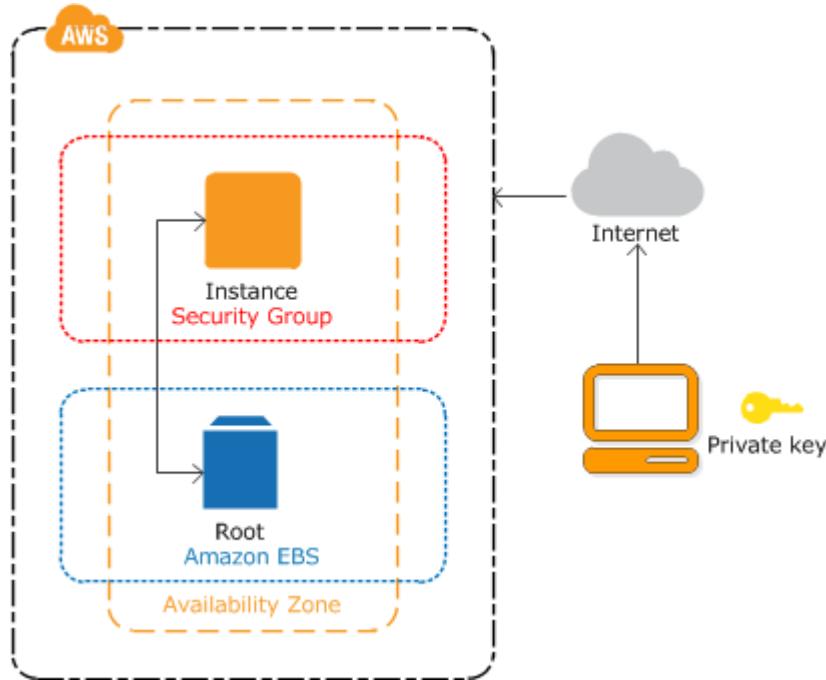
- ❖ EC2 stands for Elastic Compute Cloud.
- ❖ EC2 is a virtual machine on AWS.
- ❖ As per amazon “Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2’s simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment.”



Get started with Amazon EC2 Linux instances



- Overview



Steps:

- Step 1: Launch an instance
- Step 2: Connect to your instance
- Step 3: Clean up your instance

Ref Link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

Steps to create EC2 Instance.



Step 1: To launch an instance

- 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 instance type, which is selected by default. The t2.micro instance type is eligible for the free tier. In Regions where t2.micro is unavailable, you can use a t3.micro instance under the free tier. For more information, see [AWS 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 set up 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 set up.
- 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.

Ref Link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

aws Services ▾

Search for services, features, marketplace products, and docs [Alt+S]

vishal Mumbai

New EC2 Experience Tell us what you think

EC2 Dashboard New

Events

Tags

Limits

Instances

- Instances New
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

Images

- AMIs

Elastic Block Store

- Volumes

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances	2
Key pairs	1	Load balancers	1
Placement groups	0	Security groups	3
Snapshots	1	Volumes	0

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▾

Service health

C Service Health Dashboard

Region Status

Asia Pacific (Mumbai) This service is operating

Supported platforms VPC

Default VPC vpc-30308859

Settings

EBS encryption

Zones

Default credit specification

Console experiments

Explore AWS

Enable Best Price-Performance with AWS Graviton2

AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad spectrum of cloud workloads. Learn more

Get Up to 40% Better Price

The screenshot shows the AWS EC2 Dashboard. The left sidebar lists navigation items like 'New EC2 Experience', 'EC2 Dashboard', 'Instances', 'Images', and 'Elastic Block Store'. The main content area displays 'Resources' usage in the Asia Pacific (Mumbai) Region, with a callout for launching Microsoft SQL Server Always On availability groups. Below this is a 'Launch instance' section with a button to start a new instance. To the right, there's a 'Service health' summary and a sidebar for 'Account attributes' and 'Explore AWS'.

1. Search for “ec2” service.
2. Click on “Launch Instance”.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

[Cancel and Exit](#)

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs

Free tier only ⓘ

AMI Name	Description	Root device type	Virtualization type	ENI Enabled	Action
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-08e0ca9924195beba (64-bit x86) / ami-0437d5dbe8fdc3d52 (64-bit Arm)	Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.	ebs	hvm	Yes	Select <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0a9d27a9f4f5c0efc (64-bit x86) / ami-0816d75a127c17a49 (64-bit Arm)	Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type	ebs	hvm	Yes	Select <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0b3acf3edf2397475 (64-bit x86) / ami-0ab71076ab9b53b0d (64-bit Arm)	SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.	ebs	hvm	Yes	Select <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-073c8c0760395aab8 (64-bit x86) / ami-029dbbe5a11f53cf7 (64-bit Arm)	Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).	ebs	hvm	Yes	Select <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)

3. Select “Amazon Linux 2 AMI (HVM).
4. Click on Select.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, ~ 1 GiB memory, EBS only)

	Family	Type	vCPUs ⓘ	Memory (GiB) ⓘ	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ	IPv6 Support ⓘ
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel

Previous

Review and Launch

Next/Configure Instance Details

3. Select “t2.micro”
4. Click on “Configure Instance Details”.

Services ▾ Search for services, features, marketplace products, and docs [All 13]

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances Launch into Auto Scaling Group

Purchasing option Request Spot instances

Network vpc-30308859 (default) Create new VPC

Subnet No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP No preference (default subnet in any Availability Zone)

subnet-af65abd4 | Default in ap-south-1c
subnet-db9613b2 | Default in ap-south-1a
subnet-883cc7c5 | Default in ap-south-1b

Placement group

Capacity Reservation Open

Domain join directory No directory Create new directory

IAM role None Create new IAM role

CPU options Specify CPU options

Shutdown behavior Stop

Stop - Hibernate behavior Enable hibernation as an additional stop behavior

Cancel Previous Review and Launch Next: Add Storage

5. Click on “Add Storage”.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-07e0efc01c68d3978	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

5. Click on “Add Tags”.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances ⓘ	Volumes ⓘ	Network Interfaces ⓘ
Name	My-First-Instance	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel

Previous

Review and Launch

Next: Configure Security Group

5. Click on “Configure Security Group”.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group

Select an existing security group

Security group name:

My-SG

Description: launch-wizard-1 created 2021-02-14T17:27:05.680+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	allow traffic from all IP addresses

Add Rule

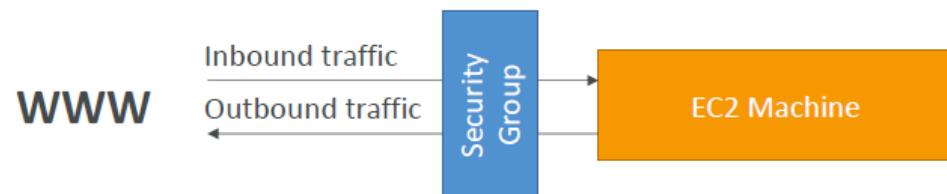


Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous **Review and Launch**
Go to Settings to activate Windows.

5. Click on “Review and launch”.



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, My-SG, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details

[Edit AMI](#)**Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-08e0ca9924195beba****Free tier eligible**

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is a...

Root Device Type: ebs Virtualization type: hvm

Instance Type

[Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

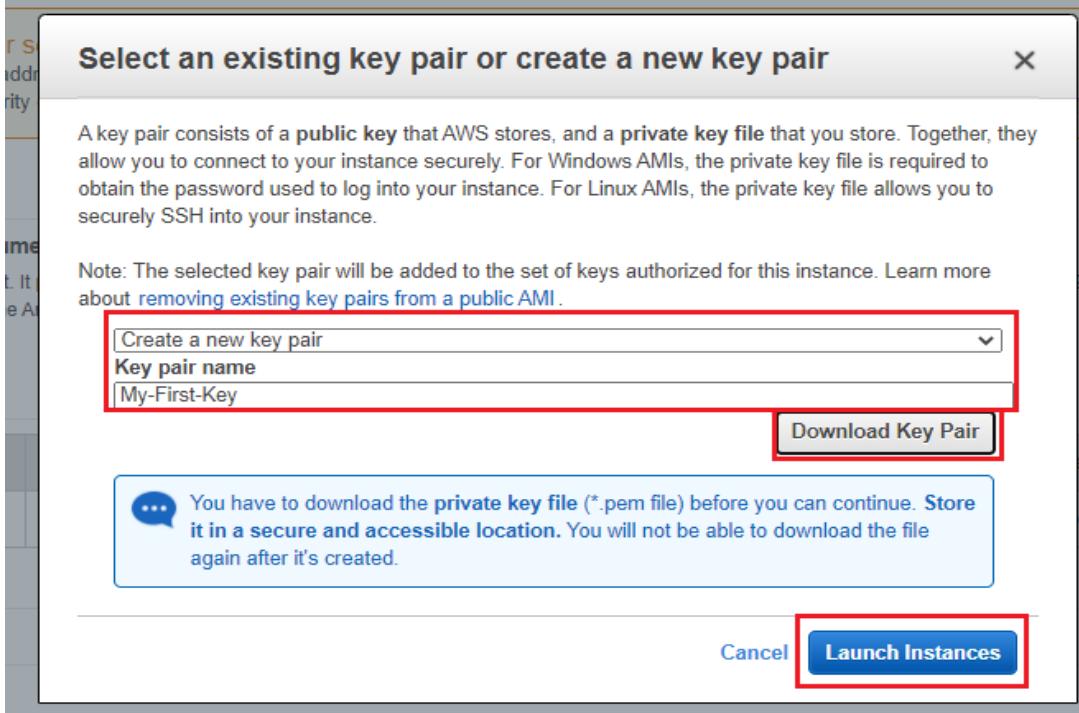
Security Groups

[Edit security groups](#)

Security group name: My-SG

Activate | Deactivate | [Cancel](#) | [Previous](#)**Launch**

5. Click on “Launch”.



5. Click on “Create a new key pair”.
6. Download Key Pair.
7. Click on “Launch Instance”.
8. Click on “View Instance”.

The screenshot shows the AWS EC2 Instances page. At the top, there's a header with 'Instances (1/3)' and 'Info' buttons, followed by 'Connect', 'Instance state', 'Actions', and 'Launch instances' buttons. Below the header is a search bar with 'Filter instances' placeholder text and navigation arrows. The main table lists three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
First	i-0610ff454f83aca5b	Terminating	t2.micro	-	No alarms	ap-south-1a	-
Second	i-0da7e305183d1c90a	Terminating	t2.micro	-	No alarms	ap-south-1b	-
My-First-Inst...	i-091152db64fd104a5	Running	t2.micro	Initializing	No alarms	ap-south-1b	ec2-15

Below the table, a modal window is open for the selected instance 'My-First-Instance'. The title is 'Instance: i-091152db64fd104a5 (My-First-Instance)'. The modal has tabs for 'Details', 'Security', 'Networking', 'Storage', 'Status checks', 'Monitoring', and 'Tags'. The 'Details' tab is active, showing sections for 'Instance summary' and 'Instance details'. The 'Platform' and 'AMI ID' fields are visible at the bottom.

9. Select your instance and check the Instance state, it must be "Running".

Congratulations!!!!!!

Steps to create EC2 Instance.



Step 2: Connect to your instance

There are several ways to connect to your Linux instance. For more information, see [Connect to your Linux instance](#).

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main area displays three instances: First, Second, and My-First-Inst... (selected). The 'My-First-Inst...' row has a context menu open, with 'Connect' highlighted. At the top of the page, there's another 'Connect' button highlighted with a red box.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
First	i-06f0ff454f83aca5b	Terminated	t2.micro	-	No alarms	ap-south-1a	-
Second	i-0da7e305183d1c90a	Terminated	t2.micro	-	No alarms	ap-south-1b	-
My-First-Inst...	i-091152db64fd104a5	Running	t2.micro	2/2 checks ...	No alarms	ap-south-1b	ec2-15-

1. Click on “Connect”.
2. Or right click on Instance and select “Connect”.

Ref Link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

Steps to create EC2 Instance.

Step 2: Connect to your instance



Connect to instance Info

Connect to your instance i-091152db64fd104a5 (My-First-Instance) using any of these options

EC2 Instance Connect **Session Manager** **SSH client**

Instance ID
 [i-091152db64fd104a5 \(My-First-Instance\)](#)

Public IP address
 [15.206.212.136](#)

User name

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

 **Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel **Connect**

1. There are 3 ways to connect your instance.
2. Click on “EC2 Instance Connect”.
3. Click on “Connect”.

Ref Link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

Steps to create EC2 Instance.



Step 2: Connect to your instance

Connect to instance Info

Connect to your instance i-091152db64fd104a5 (My-First-Instance) using any of these options

EC2 Instance Connect **Session Manager** **SSH client**

Instance ID
 [i-091152db64fd104a5 \(My-First-Instance\)](#)

Public IP address
 [15.206.212.136](#)

User name

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

 **Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel **Connect**

1. There are 3 ways to connect your instance.
2. Click on “EC2 Instance Connect”.
3. Click on “Connect”.

Ref Link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

Steps to create EC2 Instance.



How to install Apache Web server on EC2 instance.

The screenshot shows a terminal window with the URL <https://ap-south-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-091152db64fd104a5>. The terminal output shows:

```
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-14-195 ~]$ sudo su
[root@ip-172-31-14-195 ec2-user]# yum update -y
```

The screenshot shows a terminal window with the URL <https://ap-south-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-091152db64fd104a5>. The terminal output shows several blank lines followed by:

```
root@ip-172-31-14-195 ec2-user]#
root@ip-172-31-14-195 ec2-user]#
root@ip-172-31-14-195 ec2-user]#
root@ip-172-31-14-195 ec2-user]#
root@ip-172-31-14-195 ec2-user]#
root@ip-172-31-14-195 ec2-user]# sudo yum install -y httpd.x86_64
```

1. Give command -> sudo su and yum update -y
2. Sudo yum install -y httpd.x86_64 => to install Apache HTTP Server

Steps to create EC2 Instance.



How to install Apache Web server on EC2 instance.

```
← → C https://ap-south-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-091152db64fd104a5
[ec2-user@ip-172-31-14-195 ~]$ 
[ec2-user@ip-172-31-14-195 ~]$ 
[ec2-user@ip-172-31-14-195 ~]$ 
[ec2-user@ip-172-31-14-195 ~]$ 
[ec2-user@ip-172-31-14-195 ~]$ sudo su
root@ip-172-31-14-195 ec2-user]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
root@ip-172-31-14-195 ec2-user]# systemctl start httpd
root@ip-172-31-14-195 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2021-02-14 12:57:54 UTC; 6s ago
     Docs: man:httpd.service(8)
 Main PID: 4950 (httpd)
   Status: "Processing requests..."
  CGroup: /system.slice/httpd.service
          ├─4950 /usr/sbin/httpd -DFOREGROUND
          ├─4951 /usr/sbin/httpd -DFOREGROUND
          ├─4952 /usr/sbin/httpd -DFOREGROUND
          ├─4953 /usr/sbin/httpd -DFOREGROUND
          ├─4954 /usr/sbin/httpd -DFOREGROUND
          └─4955 /usr/sbin/httpd -DFOREGROUND
Feb 14 12:57:54 ip-172-31-14-195.ap-south-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Feb 14 12:57:54 ip-172-31-14-195.ap-south-1.compute.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-14-195 ec2-user]#
```

1. Give command
2. Systemctl enable httpd => to enable httpd service
3. Systemctl start httpd => to start httpd service
4. Systemstl status httpd => to check httpd service status
5. Curl localhost:80 => to check wheather apache server is responding on port 80

Steps to create EC2 Instance.



How to install Apache Web server on EC2 instance.

The screenshot shows a terminal window with the following session:

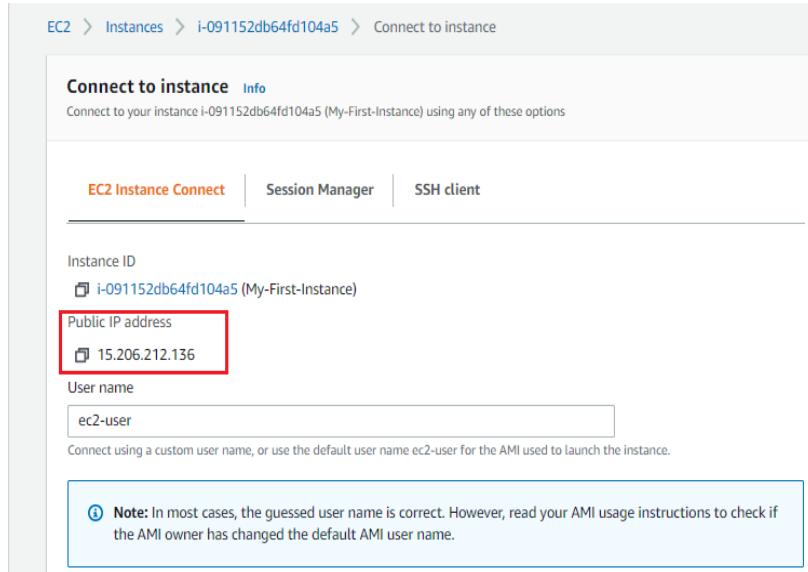
```
[root@ip-172-31-14-195 ~]#  
[root@ip-172-31-14-195 ~]#  
[root@ip-172-31-14-195 ~]#  
[root@ip-172-31-14-195 ~]# hostname -f  
ip-172-31-14-195.ap-south-1.compute.internal  
[root@ip-172-31-14-195 ~]#  
[root@ip-172-31-14-195 ~]# echo "My Name is Bond. James Bond 007. $(hostname -f)" > /var/www/html/index.html  
[root@ip-172-31-14-195 ~]#  
[root@ip-172-31-14-195 ~]#
```

The command `echo "My Name is Bond. James Bond 007. $(hostname -f)" > /var/www/html/index.html` and its output are highlighted with a red box.

1. Give command
2. Hostname -f => to know name of your ec2 instance
3. Echo “My name is bond. James Bond 007. \$(hostname -f)” > /var/www/html/index.html

Steps to create EC2 Instance.

How to install Apache Web server on EC2 instance.



EC2 > Instances > i-091152db64fd104a5 > Connect to instance

Connect to instance Info

Connect to your instance i-091152db64fd104a5 (My-First-Instance) using any of these options

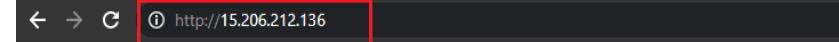
EC2 Instance Connect Session Manager SSH client

Instance ID: i-091152db64fd104a5 (My-First-Instance)

Public IP address: 15.206.212.136 (highlighted with a red box)

User name: ec2-user

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.



← → C ⓘ http://15.206.212.136



This site can't be reached

15.206.212.136 took too long to respond.

Try:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

1. Copy the public IP of ec2 instance and paste in browser.
2. You will get an error “ERR_CONNECTION_TIMED_OUT”

Lets Solve this issue now.....

Steps to create EC2 Instance.

How to solve connection error. Whenever there is timeout problem, check your “Security Groups”.



The screenshot shows the AWS Management Console interface for the EC2 service. On the left, a navigation menu lists various AWS services like Spot Requests, Savings Plans, Reserved Instances, etc., with 'Network & Security' expanded to show 'Security Groups'. The 'Security Groups' link is highlighted with a red box. The main content area displays a table titled 'Instances (1/1)'. It shows one instance named 'My-First-Inst...' with the ID 'i-091152db64fd104a5', which is currently 'Running'. Other columns include 'Instance type' (t2.micro), 'Status check' (2/2 checks ...), 'Alarm status' (No alarms), 'Availability Zone' (ap-south-1b), and 'Public IP' (ec2-15-). Below the table, there are sections for 'State transition message', 'Usage operation' (RunInstances), and 'RAM disk ID'.

Click on “Security Groups” from left side menu. Our EC2 instance is attached with “My-SG” security group. Select that and click on “Actions” then select “Edit Inbound Rules”.

The screenshot shows the AWS Management Console interface for the Security Groups service. The left sidebar shows the same navigation menu as the previous screenshot. The main content area displays a table titled 'Security Groups (1/4)'. It lists four security groups: 'ELB-SecurityGroup', 'My-First-Security', 'My-SG' (which is selected and highlighted with a red box), and 'default'. The 'Actions' button is highlighted with a red box, and the 'Edit inbound rules' option under its dropdown menu is also highlighted with a red box. Other actions listed in the dropdown include 'Edit outbound rules', 'Manage tags', 'Manage stale rules', and 'Copy to new security group'.

Steps to create EC2 Instance.

How to solve connection error. Whenever there is timeout problem, check your “Security Groups”.



The screenshot shows the AWS Security Groups Inbound rules configuration. It displays two rules:

- Rule 1:** Type: SSH, Protocol: TCP, Port range: 22, Source: Custom (0.0.0.0/0), Description: allow traffic from all IP addresses.
- Rule 2:** Type: HTTP, Protocol: TCP, Port range: 80, Source: Custom (0.0.0.0/0), Description: allow traffic from anywhere on port 80.

A red box highlights the "Add rule" button at the bottom left. Another red box highlights the second rule's source field. A note at the bottom states: "⚠ NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created." At the bottom right, there are "Cancel", "Preview changes", and "Save rules" buttons, with "Save rules" also highlighted by a red box.

Click on “Add Rule” . Select “HTTP”. And then click on “Save rules”.

Now, again go on browser and paste public IP and hit enter. You will get the output.

The screenshot shows a browser window with the address bar displaying "Not secure | http://15.206.212.136". The main content area shows the text "My Name is Bond. James Bond 007. ip-172-31-14-195.ap-south-1.compute.internal", which is highlighted by a red box.

Congratulations!!!!!!

Steps to create EC2 Instance.



Step 3: Clean up your instance

- 1) After you've finished with the instance that you created for this tutorial, you should clean up by terminating the instance.
- 2) 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.

The screenshot shows the AWS EC2 Instances page. A single instance, "My-First-Inst...", is listed. A context menu is open over this instance, showing options like "Launch instances", "Launch instance from template", "Connect", and four options highlighted with red boxes: "Stop instance", "Start instance", "Reboot instance", and "Terminate instance". Another context menu is open over the "Instance state" column header, also showing "Stop instance", "Start instance", "Reboot instance", and "Terminate instance" options highlighted with red boxes. The instance details panel on the left shows the instance ID "i-091152db64fd1015", the owner "933126026785", and various settings like networking and security.

Click on “Instance State” . Or right click on your “Instance Id”. Select “Stop instance” or “Terminate instance”.

Ref Link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

Steps to create EC2 Instance.



Step 3: Clean up your instance

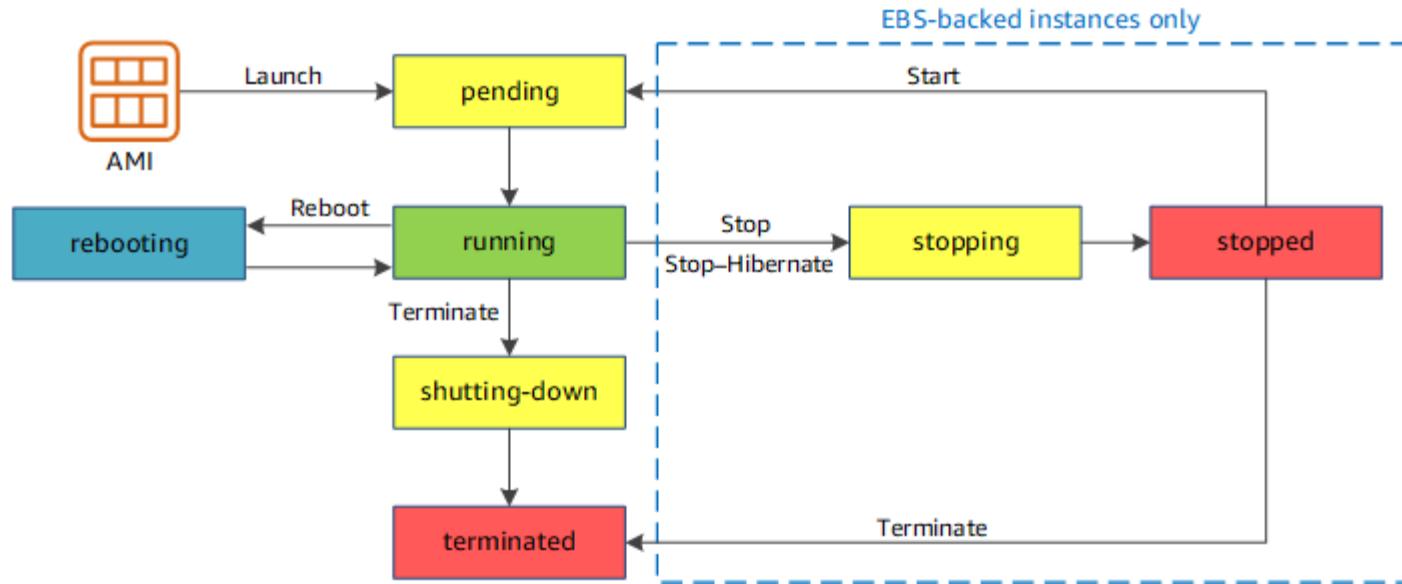
- 1) After you've finished with the instance that you created for this tutorial, you should clean up by terminating the instance.
- 2) If you launched an instance that is not within the [AWS Free Tier](#), you'll stop incurring charges for that instance as soon as the instance status changes to shutting down or terminated.

The screenshot shows the AWS EC2 Instances page. A single instance, "My-First-Inst...", is listed. A context menu is open over this instance, showing options like "Launch instances", "Launch instance from template", "Connect", and four options highlighted with red boxes: "Stop instance", "Start instance", "Reboot instance", and "Terminate instance". Another context menu is open over the "Instance state" column header, also showing "Stop instance", "Start instance", "Reboot instance", and "Terminate instance" options highlighted with red boxes. The instance details panel on the left shows the instance ID "i-091152db64fd1015", the owner "933126026785", and various settings like networking and security.

Click on “Instance State” . Or right click on your “Instance Id”. Select “Stop instance” or “Terminate instance”.

Ref Link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

Instance lifecycle



Ref Link: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html>

Contents:

- What is AWS
- Why are enterprises choosing AWS?
- AWS Account setup and billing alarms
- IAM Introduction
- Compute services (EC2)
- **Storage services (S3 Buckets, EBS)**
- Database services (RDS, Dynamo DB)
- Elastic Load Balancer (ELB)

S3 (Simple Storage Service)



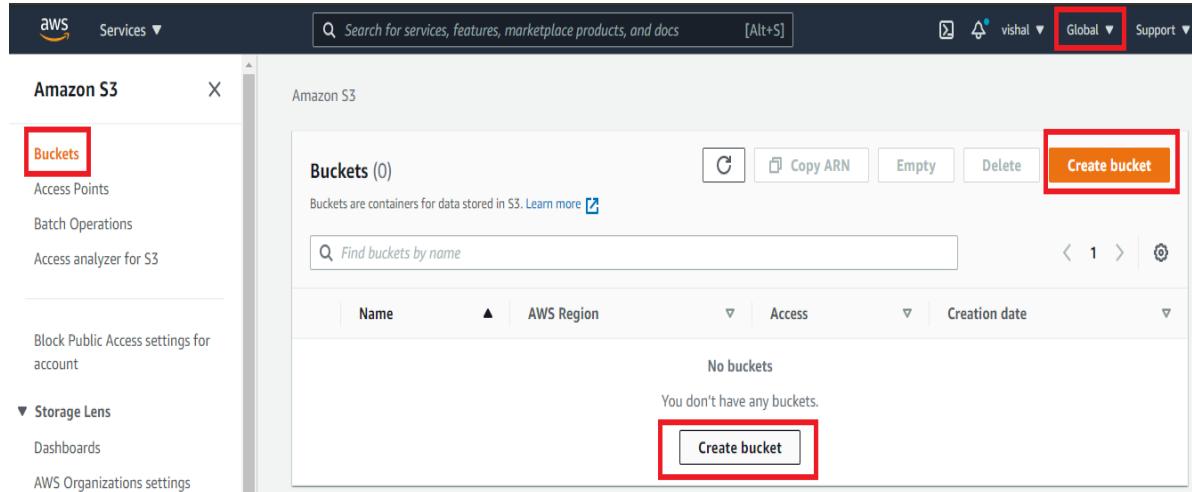
- ❖ Amazon Simple Storage Service (Amazon S3) is storage for the internet.
- ❖ You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web.
- ❖ Amazon S3 stores data as objects within buckets.
- ❖ An object is a file and any optional metadata that describes the file.
- ❖ To store a file in Amazon S3, you upload it to a bucket.
- ❖ When you upload a file as an object, you can set permissions on the object and any metadata.
- ❖ Buckets are containers for objects. You can have one or more buckets.
- ❖ You can control access for each bucket, deciding who can create, delete, and list objects in it.
- ❖ You can also choose the geographical Region where Amazon S3 will store the bucket and its contents and view access logs for the bucket and its objects.

Ref link: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>

Use Case: Deploy your own static website on S3

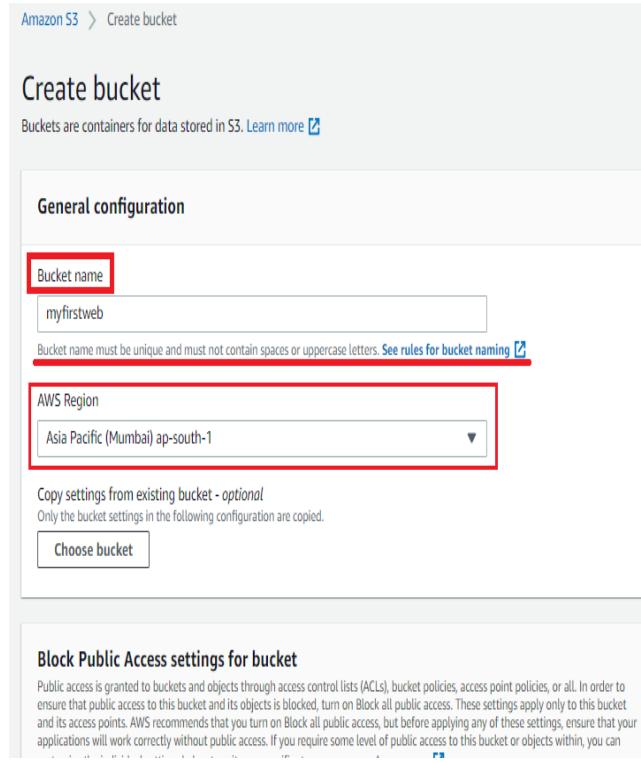
Steps:

- Step 1: Create a sample web application on your local machine.
- Step 2: Login to AWS console
- Step 3: Search for S3 service



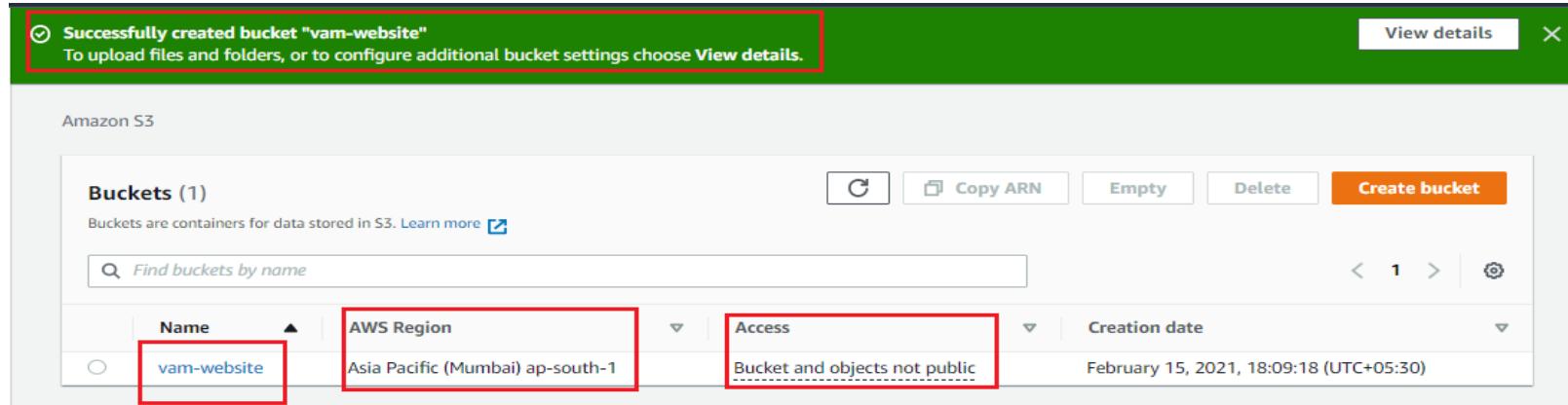
The screenshot shows the AWS S3 service dashboard. On the left, there's a sidebar with options like 'Buckets' (which is selected and highlighted with a red box), 'Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for account', 'Storage Lens', 'Dashboards', and 'AWS Organizations settings'. The main area is titled 'Amazon S3' and shows a table with one row: 'Buckets (0)'. Below the table, it says 'No buckets' and 'You don't have any buckets.' At the bottom right of the main area, there's a large orange 'Create bucket' button, which is also highlighted with a red box. The top navigation bar includes the AWS logo, 'Services ▾', a search bar ('Search for services, features, marketplace products, and docs [Alt+S]'), and user information ('vishal ▾ Global ▾ Support ▾').

- Step 4: Click on “Create Bucket”
- Step 5: give “Bucket Name” which must be globally unique.
- Step 6: select region
- Step 7: Click on “Create Bucket”



The screenshot shows the 'Create bucket' wizard. The first step is 'General configuration'. It has a 'Bucket name' field containing 'myfirstweb' (which is highlighted with a red box) and an 'AWS Region' dropdown menu set to 'Asia Pacific (Mumbai) ap-south-1' (also highlighted with a red box). Below these fields, there's a note about copy settings from existing buckets and a 'Choose bucket' button. At the bottom, there's a section for 'Block Public Access settings for bucket' with a note about public access settings and a link to learn more.

Use Case: Deploy your own static website on S3



Successfully created bucket "vam-website". To upload files and folders, or to configure additional bucket settings choose [View details](#).

View details X

Amazon S3

Buckets (1)

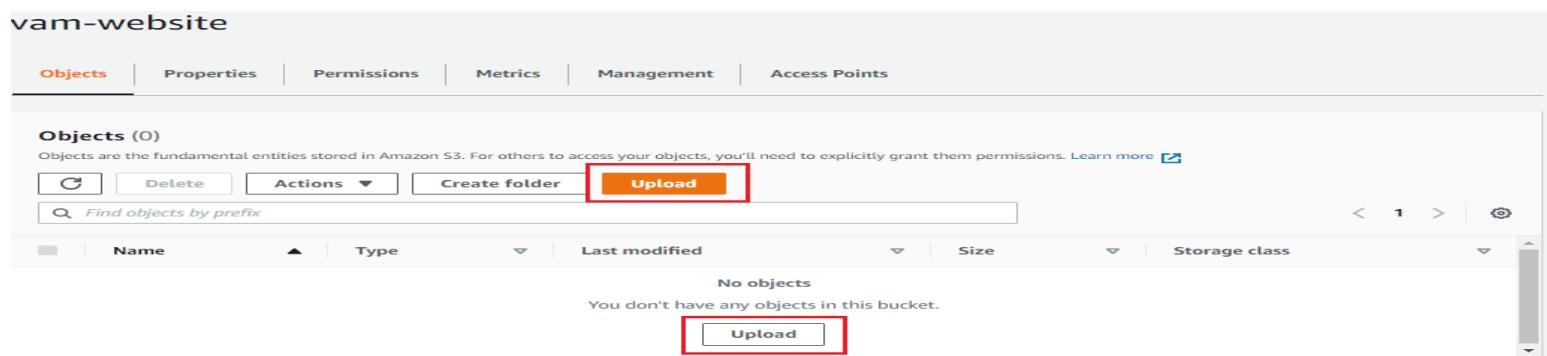
Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

Name	AWS Region	Access	Creation date
vam-website	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	February 15, 2021, 18:09:18 (UTC+05:30)

Congratulations!!!! - A bucket is successfully created.

Step 8: Right click on Bucket Name. and click on “Upload”



vam-website

Objects Properties Permissions Metrics Management Access Points

Objects (0)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Actions Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
No objects				

You don't have any objects in this bucket.

Upload

Use Case: Deploy your own static website on S3

Upload

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.

Files and folders (0)					
All files and folders in this table will be uploaded.					
<input type="text"/> Find by name					
Name	Folder	Type	Size		
No files or folders					
You have not chosen any files or folders to upload.					

[Remove](#) [Add files](#) [Add folder](#)

Step 9: Click on “Add Folder”. Select your website folder and upload it. And click on “Upload” Button.

Step 10: go on your Object Page.

Upload

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.

Files and folders (3 Total, 862.0 B)					
All files and folders in this table will be uploaded.					
<input type="text"/> Find by name					
Name	Folder	Type	Size		
error.html	mywebsite/	text/html	269.0 B		
index.html	mywebsite/	text/html	312.0 B		
page2.html	mywebsite/	text/html	281.0 B		

[Remove](#) [Add files](#) [Add folder](#)

Destination

Destination
<s3://vam-website>

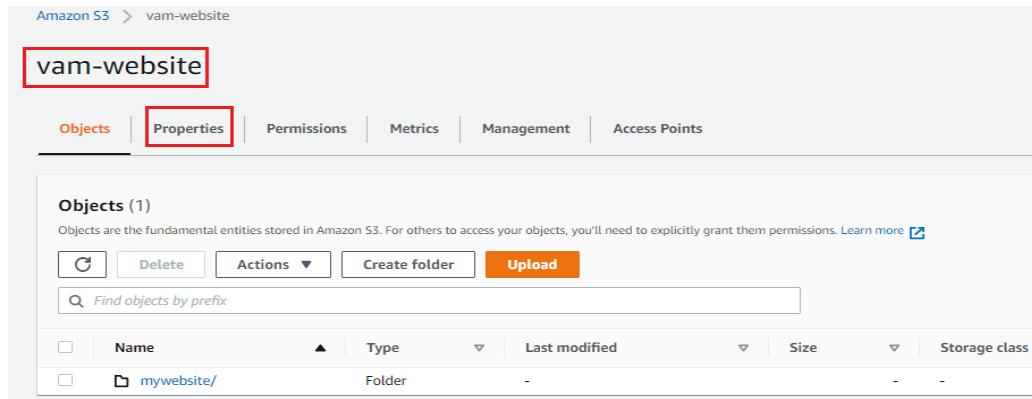
Use Case: Deploy your own static website on S3

Step 9: Click on “Add Folder”. Select your website folder and upload it. And click on “Upload” Button.

Step 10: go on your Object Page.

Step 11: Click on “Properties”

Step 12: At bottom “Static Website Hosting” click on “Edit”



Amazon S3 > vam-website

vam-website

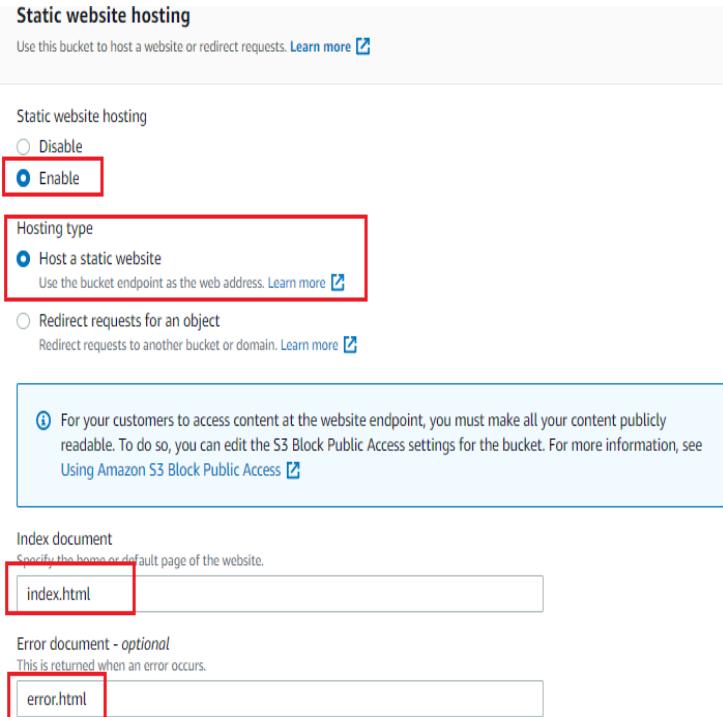
Objects (1)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Name	Type	Last modified	Size	Storage class
mywebsite/	Folder	-	-	-

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)



Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Disable

Enable

Hosting type

Host a static website

Use the bucket endpoint as the web address. [Learn more](#)

Redirect requests for an object

Redirect requests to another bucket or domain. [Learn more](#)

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document

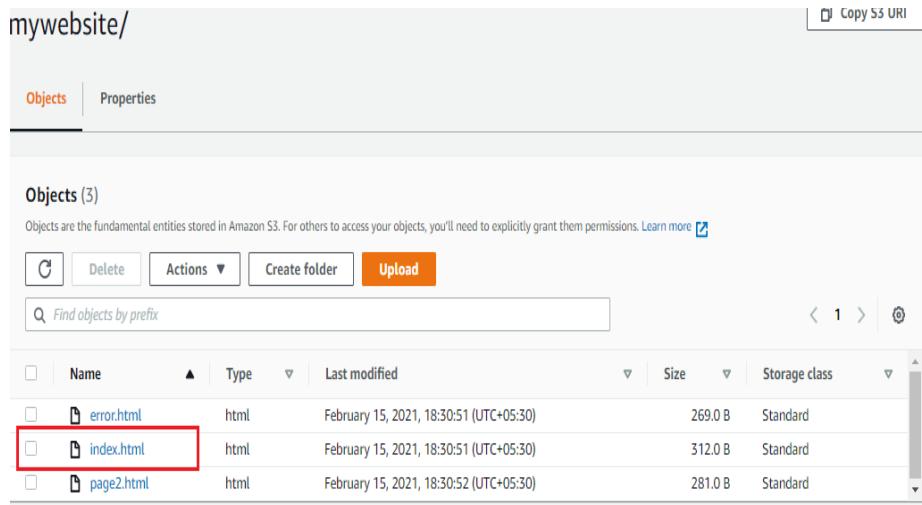
Specify the home or default page of the website.

Error document - optional

This is returned when an error occurs.

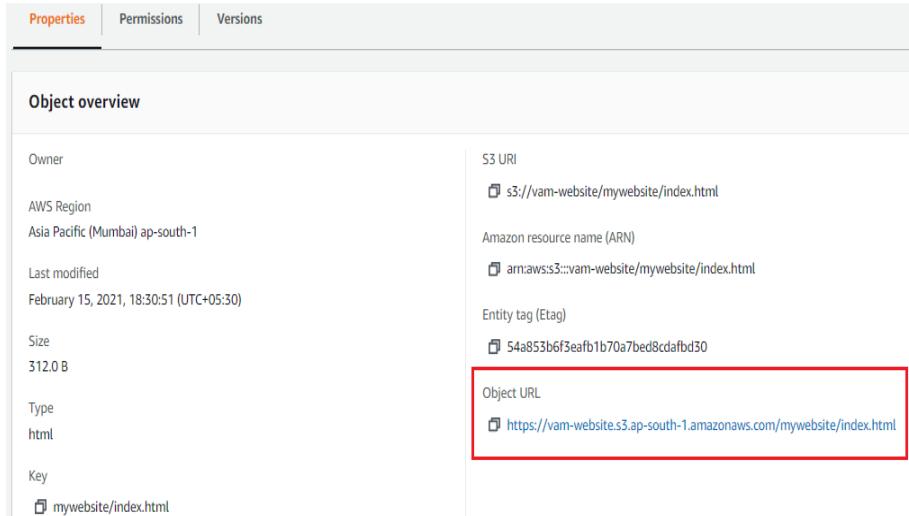
Use Case: Deploy your own static website on S3

Step 12: Click on “index.html” object. And get the object URL



The screenshot shows the AWS S3 console interface. At the top, there's a search bar with 'mywebsite/' and a 'Copy S3 URI' button. Below it, tabs for 'Objects' and 'Properties' are visible. Under 'Objects (3)', there's a note about objects being fundamental entities. A toolbar below includes 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar says 'Find objects by prefix'. The main table lists three objects:

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	error.html	html	February 15, 2021, 18:30:51 (UTC+05:30)	269.0 B	Standard
<input type="checkbox"/>	index.html	html	February 15, 2021, 18:30:51 (UTC+05:30)	312.0 B	Standard
<input type="checkbox"/>	page2.html	html	February 15, 2021, 18:30:52 (UTC+05:30)	281.0 B	Standard



The screenshot shows the 'Properties' tab of an S3 object. It displays various details about the object:

- Owner: AWS Region: Asia Pacific (Mumbai) ap-south-1
- Last modified: February 15, 2021, 18:30:51 (UTC+05:30)
- Size: 312.0 B
- Type: html
- Key: mywebsite/index.html
- S3 URI: s3://vam-website/mywebsite/index.html
- Amazon resource name (ARN): arn:aws:s3:::vam-website/mywebsite/index.html
- Entity tag (Etag): 54a853b6f3eafb1b70a7bed8cdafbd30
- Object URL: https://vam-website.s3.ap-south-1.amazonaws.com/mywebsite/index.html

Step 13: Copy the URL in browser and you will get an Error “Access Denied”.



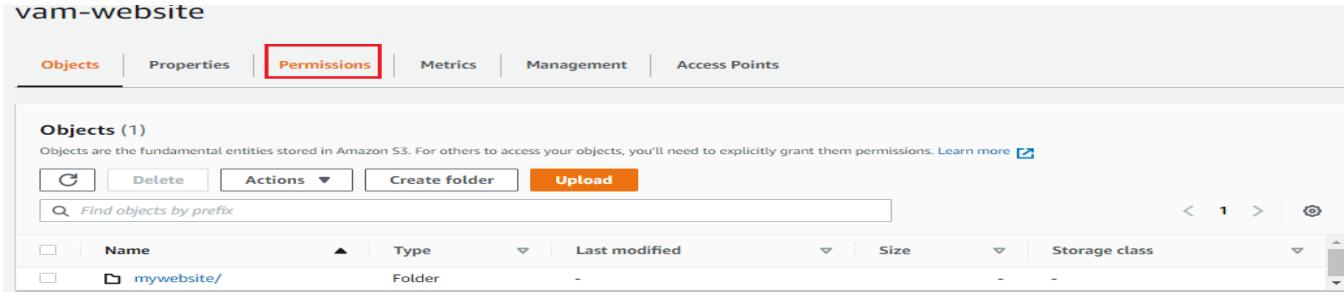
The screenshot shows a browser window with the URL https://vam-website.s3.ap-south-1.amazonaws.com/mywebsite/index.html. The page content is an XML error response:

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>D9FF1B7487C054F0</RequestId>
<HostId>M5z/VKfygnJPVXlkqnqS2nXKdMY/RxpjnkZaF3eTja9p9COOXeF3hT5cDGo6Tf1Uj7xbxFWh4Q=</HostId>
```

Lets Solve this issue now.

Use Case: Deploy your own static website on S3

Step 14: Visit your bucket page and click on “Permission”



The screenshot shows the AWS S3 console for a bucket named 'vam-website'. The 'Permissions' tab is highlighted with a red border. Under the 'Objects' section, there is one object named 'mywebsite/'. The object type is listed as 'Folder'.

Step 15: Click on “Bucket Public access - Edit” . Uncheck the “Block all public access” box. Click on “Save Changes”

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

On

- Block public access to buckets and objects granted through new access control lists (ACLs)
 On
- Block public access to buckets and objects granted through any access control lists (ACLs)
 On
- Block public access to buckets and objects granted through new public bucket or access point policies
 On
- Block public and cross-account access to buckets and objects through any public bucket or access point policies
 On

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

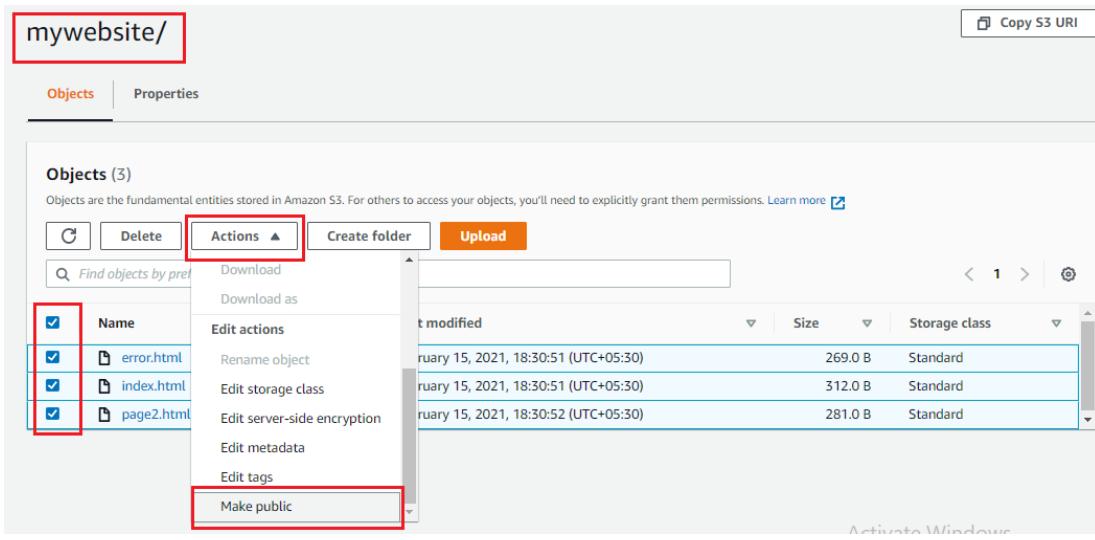
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- **Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- **Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- **Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- **Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Use Case: Deploy your own static website on S3

Step 16: again go to object page, copy “index.html” link and open in browser. If still problem remain then do the following steps:

Step 17: open Object page. Select all “Object” click on “Action” and select “Make Public”.



The screenshot shows the AWS S3 console with the path "mywebsite/" selected in the left sidebar. The main area displays three objects: "error.html", "index.html", and "page2.html". The "Actions" button is highlighted with a red box. A dropdown menu is open, showing options like "Edit actions", "Rename object", "Edit storage class", etc., followed by a "Make public" option at the bottom, which is also highlighted with a red box.

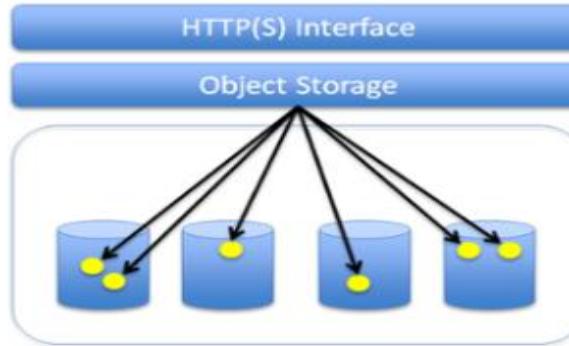


The screenshot shows a web browser window with the URL <https://vam-website.s3.ap-south-1.amazonaws.com/mywebsite/index.html>. The page content reads "How to publish static website using S3 tutorial" and "This is very easy." Below that is a "Visit Next Page" link.

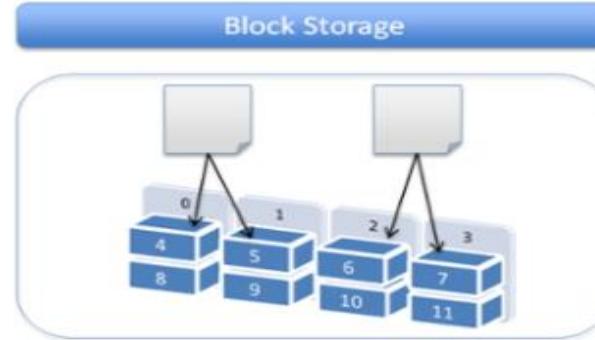
Step 18: Now copy the index.html url and open in browser.

Congratulations!!!!!!

Difference between S3, EBS and EFS



- Store virtually unlimited files.
- Maintain file revisions.
- HTTP(S) based interface.
- Files are distributed in different physical nodes.



- File is split and stored in fixed sized blocks.
- Capacity can be increased by adding more nodes.
- Suitable for applications which require high IOPS, database, transactional data.

Image Source: [NetApp Cloud](#) (used with permission)

Difference between S3, EBS and EFS

Category	S3	EBS	EFS
Storage Type	Object Storage	Block Storage	File Storage
Pricing	Pay as you Use	Pay for provisioned capacity	Pay as you Use
Storage Size	Unlimited Storage	Limited storage	Unlimited Storage
Scalability	Unlimited Scalability	Increase/decrease size manually	Unlimited Scalability
Durability	Stored redundantly across multiple Azs	Stored redundantly in a Single AZ	Stored redundantly across multiple Azs
Availability	Max is 99.99% with S3	99.99%	No SLAs
Security	Supports Data at Rest and Data in Transit encryption	Supports Data at Rest and Data in Transit encryption	Supports Data at Rest and Data in Transit encryption
Back up and Restore	Use Versioning or cross-region replication	Automated Backups and Snapshots	EFS to EFS replication
Performance	Slower than EBS and EFS	Faster than S3 and EFS	Faster than S3, Slower than EBS
Accessibility	Publicly and Privately accessible	Accessible only via the attached EC2 instance	Accessible simultaneously from multiple EC2 and on-premises instances
Interface	Web Interface	File System Interface	Web and File System Interface
Use cases	Media, Entertainment, Big data analytics, backups and archives, web serving and content management	Boot volumes, transactional and NoSQL databases, data warehousing ETL	Media, Entertainment, Big data analytics, backups and archives, web serving and content management, home directories

Ref: <https://jayendrapatil.com/aws-s3-vs-ebs-vs-efs/>

Contents:

- What is AWS
- Why are enterprises choosing AWS?
- AWS Account setup and billing alarms
- IAM Introduction
- Compute services (EC2)
- Storage services (S3 Buckets, EBS)
- Database services (RDS, Dynamo DB)
- Elastic Load Balancer (ELB)
- Lambda

Thank You

Presentation Topic

Cloud Computing

Vishal Ambadas Meshram
vishal.meshram@viit.ac.in

Department of Computer Engineering



BRACT'S, Vishwakarma Institute of Information Technology, Pune-48

(An Autonomous Institute affiliated to Savitribai Phule Pune University)
(NBA and NAAC accredited, ISO 9001:2015 certified)

UNIT 1:

Introduction To Cloud Computing

Reference:

- 1) Thomas Erl, Zaigham Mahmood and Ricardo Puttini, “*Cloud Computing: Concepts, Technology and Architecture*”,
- 2) Wikipedia
- 3) Edureka

Contents

- Overview, Applications, Intranets and the Cloud.
- Your Organization and Cloud Computing- Benefits, Limitations, Security Concerns.
- Software as a Service (SaaS)- Understanding the Multitenant Nature of SaaS Solutions,
- Understanding SOA.
- Platform as a Service (PaaS).
- Infrastructure as a Service (IaaS)
- Case Study: Google Cloud Platform

CO Achieved in First Lecture

- 1) To understand cloud computing concepts
- 2) To study supporting technologies of cloud

Course Outcomes:-

At the end of the unit you will be able to:

- 1) Summarize the basic concepts of cloud computing
(Understand)
- 2) Make use of supporting technologies for cloud computing
(Understand, Apply)

University Questions

Q1)

- a) Comment on “Cloud and virtualization”
- b) What do you mean by pods, aggregation and silos in terms of IAAS
- c) Enlist and explain the challenges and obstacles while adopting cloud computing?

OR

Q2)

- a) What is administrating and monitoring cloud services? Enlist and explain different tools used for administrating and monitoring cloud services.
- b) Enlist services provided by SAAS, PAAS and IAAS
- c) How do you relate cloud computing with utility computing?

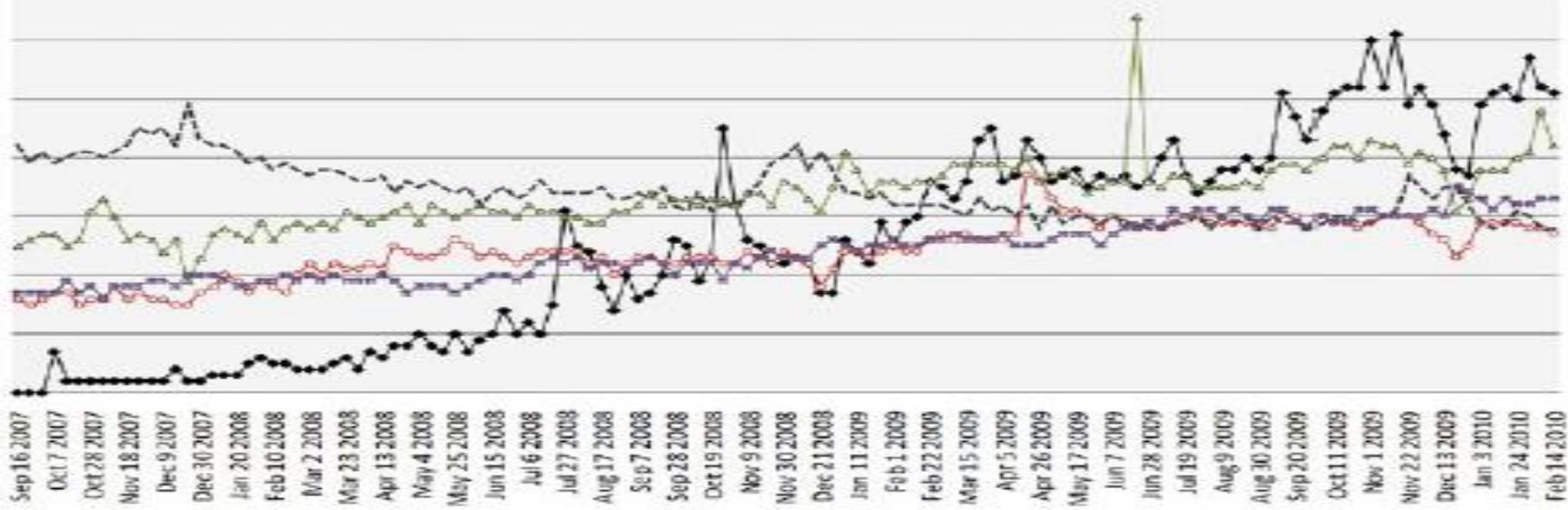
Popularity of cloud computing

Top 10 Strategic Technology Areas for 2010 (Powered By Gartner)

- 1. Cloud Computing
- 2. Advanced Analytics
- 3. Client Computing
- 4. IT for Green
- 5. Reshaping the Data Center
- 6. Social Computing
- 7. Security – Activity Monitoring
- 8. Flash Memory
- 9. Virtualization for Availability
- 10. Mobile Applications

Top Strategic Technologies (Powered By Google Trends)

—●— Cloud Computing —●— GreenIT —●— Social Network —●— Flash Memory —●— Mobile Application



A brief history

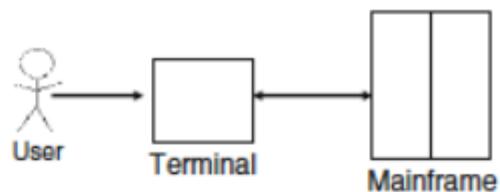
- The idea of computing in a “cloud” traces back to the origins of utility computing, a concept that computer scientists **John McCarthy** publically proposed in **1961**:

“ If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility..... the computer utility could become the basis of a new and important industry”

History

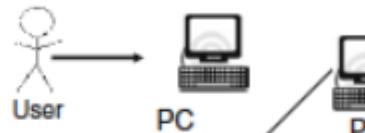
Many users shared powerful mainframes using dummy terminals

1. Mainframe Computing



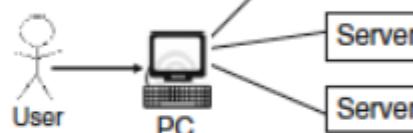
Stand-alone PCs became powerful enough to meet the majority of users' needs

2. PC Computing



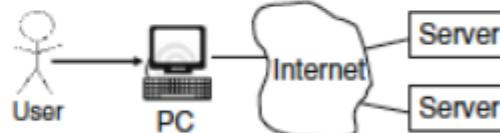
PCs, laptops, and servers were connected together through local networks to share resources and increase performance

3. Network Computing



Local networks were connected to other local networks forming a global network such as the Internet to utilise remote applications and resources

4. Internet Computing



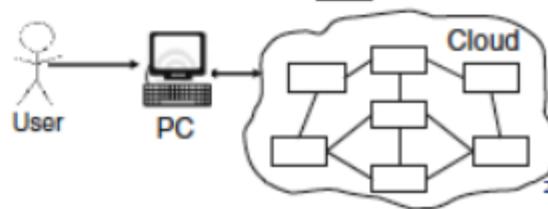
Grid computing provided shared computing power and storage through a distributed computing system

5. Grid Computing



Cloud computing further provides shared resources on the Internet in a scalable and simple way

6. Cloud Computing



Borko Furht, "Handbook of Cloud Computing"

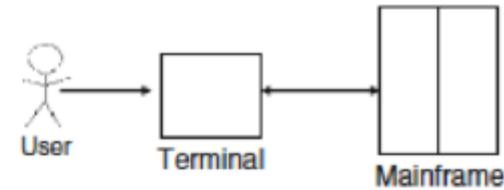
29

History

Offers finite computing power

Dummy terminals acted as user interface devices

1. Mainframe Computing

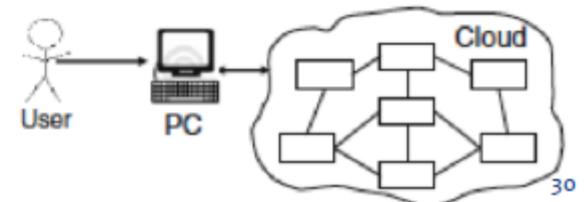


Quite similar?

Provides almost infinite power and capacity

Powerful PCs can provide local computing power and caching support

6. Cloud Computing



Borko Furht, "Handbook of Cloud Computing"

30

Introduction to Cloud Computing

- **Cloud Computing can be defined as**

“a new style of computing in which dynamically scalable and often virtualized resources are provided as a services over the Internet”.

Forrester Research provided its own definition of cloud computing:

“.... A standardized IT capability (services, software, or infrastructure) delivered via internet technologies in a pay-per-use, self-service way.”

Introduction to Cloud Computing

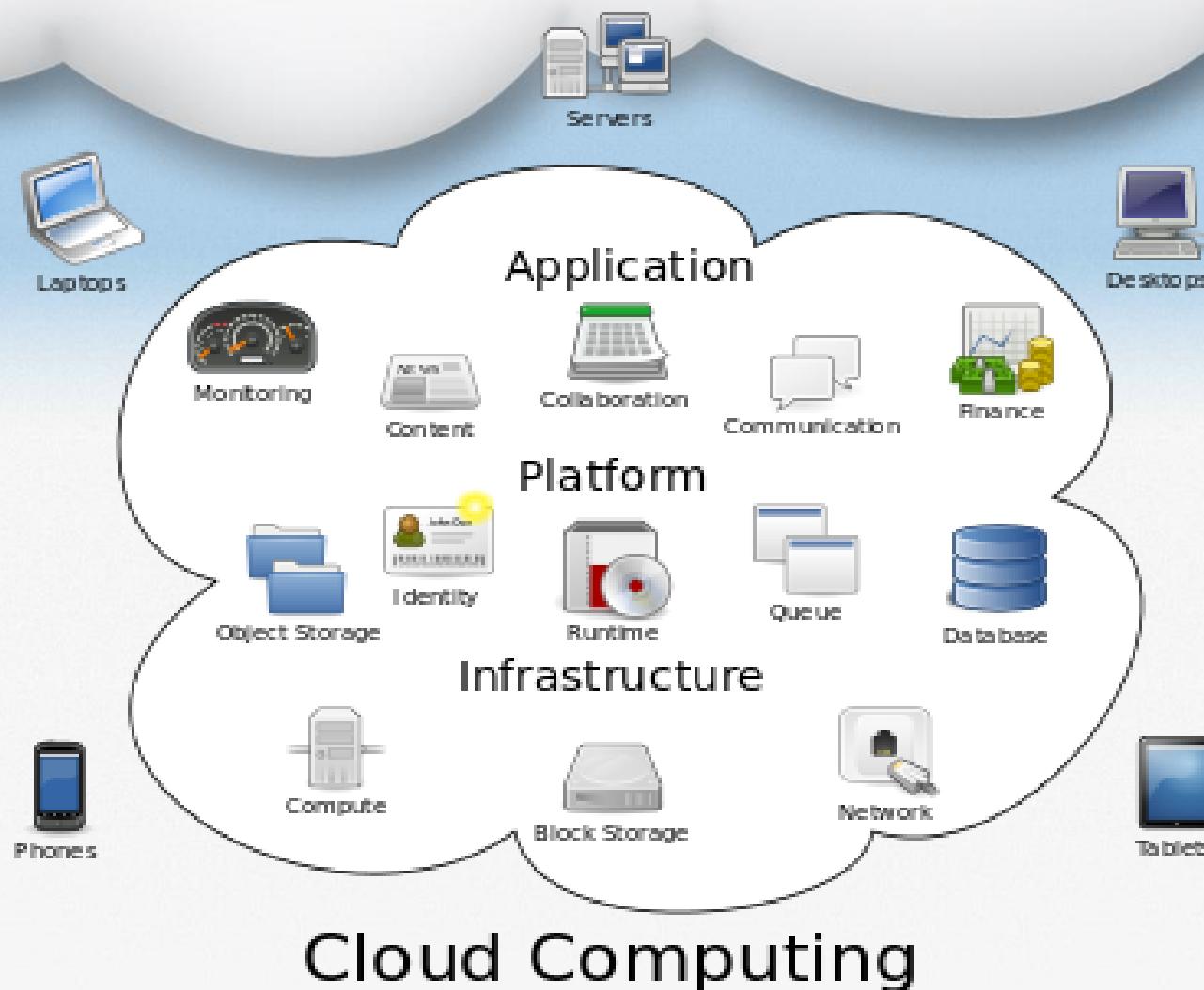
- **NIST (National Institute of Standards & Technology) Definition:**

*“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. *network, servers, storage, applications, and services*) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models. ”*

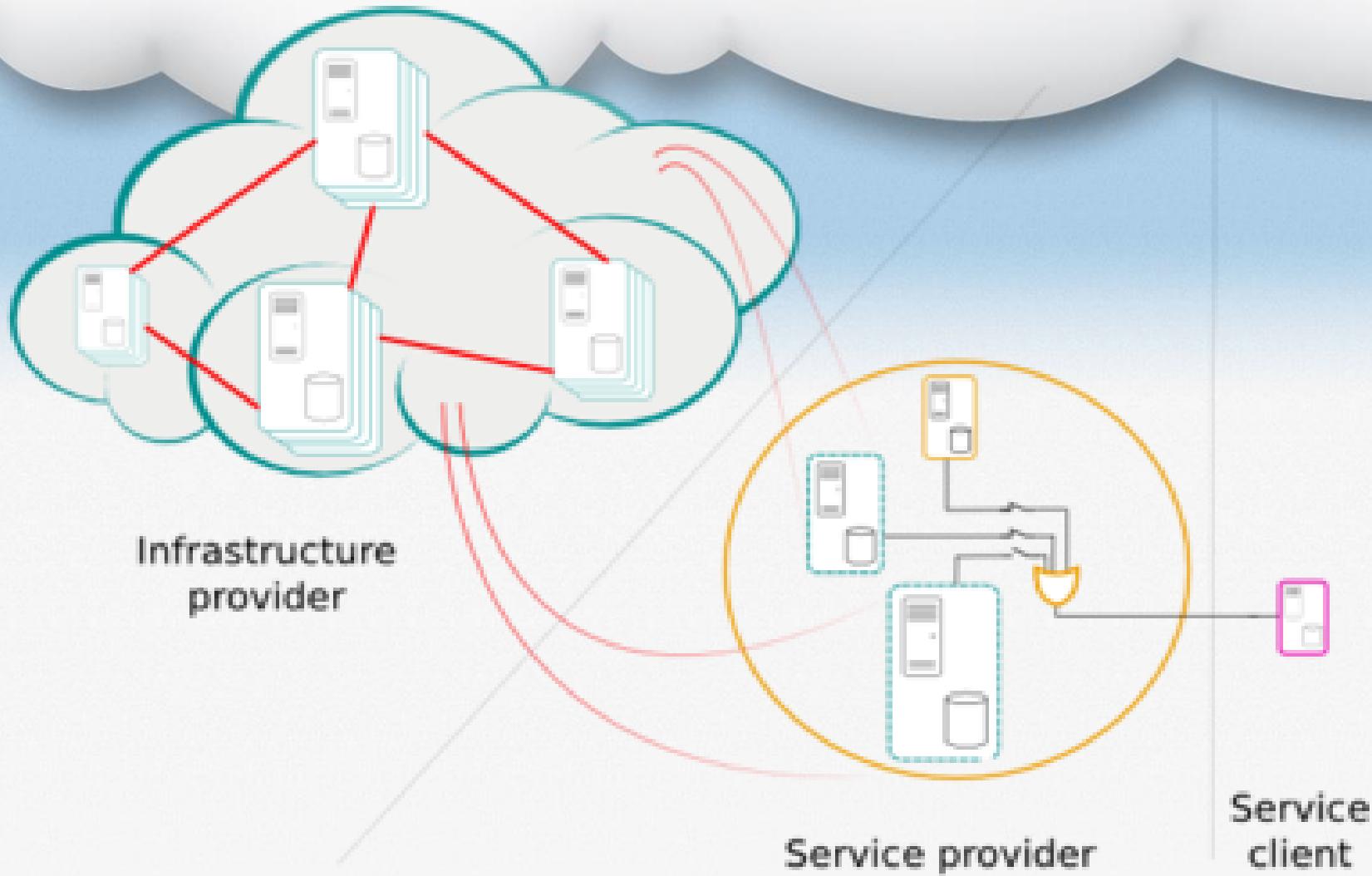
- A more concise definition:

“Cloud computing is a specialized form of distributed computing that introduces utilization models for remotely provisioning scalable and measured resources”

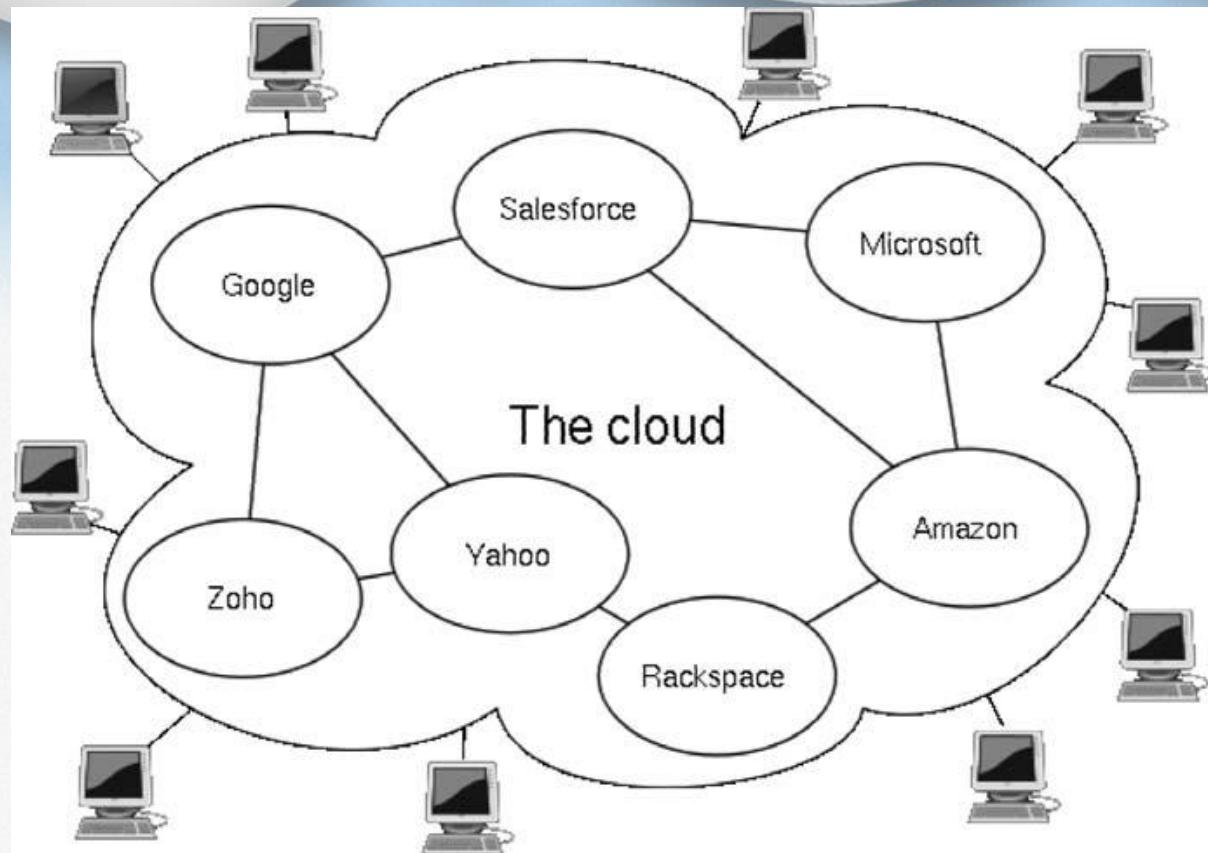
What is cloud computing?



What is cloud computing?



What is cloud computing?



Business Drivers

- Capacity Planning
- Cost Reduction
- Organizational Agility

Technology Innovations

- Clustering:

A cluster is a group of independent IT resources that are interconnected and work as a single system. System failure rates are reduced while availability and reliability are increased, since redundancy and failover are inherent to the cluster.

A general prerequisite of hardware clustering is that its component systems have reasonably identical hardware and operating systems to provide similar performance levels when one failed component is to be replaced by another. Component devices that form a cluster are kept in synchronization through dedication, high-speed communication links.

the basic concept of built-in redundancy and failover is core to cloud platforms.

Technology Innovations

- Grid Computing:

A computing grid (or “Computational grid”) provides a platform in which computing resources are organized into one or more logical pools. These pools are collectively coordinated to provide a high performance distributed grid, sometimes referred to as a “super virtual computer”.

Grid computing differs from clustering in that grid systems are much loosely coupled and distributed. As a result, grid computing systems can involve computing resources that are heterogeneous and geographically dispersed, which is generally not possible with cluster-based systems.

The technological advancements achieved by grid computing projects have influenced various aspects of cloud computing platforms and mechanism, specifically in relation to common feature- sets such as networked access, resource pooling, and scalability and resiliency.

These types of features can be established by both grid computing and cloud computing, in their own distinctive approaches.

Technology Innovations

- Virtualization:

Virtualization represents a technology platform used for the creation of virtual instances of IT resources.

A layer of virtualization software allows physical IT resources to provide multiple virtual images of themselves so that their underlying processing capabilities can be shared by multiple users.

Prior to the virtualization technology, software was limited to residing on and being coupled with static hardware environments. The virtualization process servers this software-hardware dependency, as hardware requirements can be simulated by emulation software running in virtualized environments.

Established virtualization technologies can be traced to several cloud characteristics and cloud computing mechanisms, having inspired many of their core features.

Modern virtualization technologies emerged to overcome the performance, reliability and scalability limitations of traditional virtualization platforms.

Technology Innovations Vs. Enabling Technologies

- It is essential to highlight several other areas of technology that continue to contribute to modern-day cloud-based platforms. There are distinguished as cloud-enabling technologies :
 - Broadband Networks and internet architecture
 - data center technology
 - (Modern) virtualization technology
 - Web technology
 - Multitenant technology
 - Service technology

Each of these cloud-enabling technologies existed in some form prior to the formal advent of cloud computing some were refined further, and on occasion even redefined, as a result of the subsequent evolution of cloud computing.

Summary of key points

- The primary business drivers that exposed the need of cloud computing and led to its formation include capacity planning, cost reduction, and organizational agility.
- The primary technology innovations that influenced and inspired key distinguishing features and aspects of cloud computing include clustering, grid computing, and traditional forms of virtualization.

Home Assignment Question

- Difference between Cluster, Grid and cloud computing??
- Tomorrow's Points:
 - Five essential characteristics,
 - Three service models, and
 - Four deployment models.

Muddiest Point

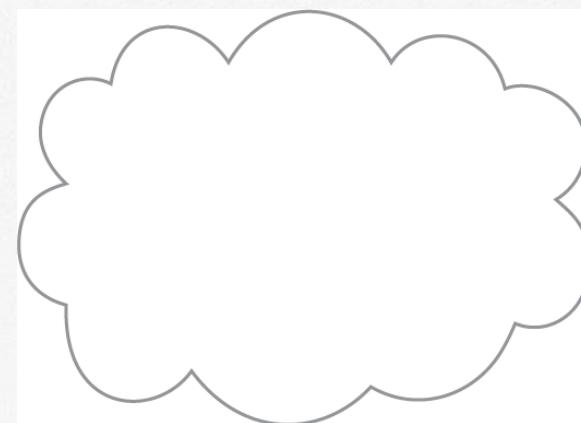
Thank You!!!!

Basic Concepts & Terminology

Cloud:

- A cloud refers to a distinct IT environment that is designed for the purpose of remotely provisioning scalable and measured IT resources.
- the symbol of cloud was commonly used to represent the Internet in a verity of specifications.
- This same symbol is now used to specifically represent the boundary of cloud environment, as shown in fig.

Fig: The symbol used to denote the boundary of a cloud environment.



Copyright © Arcitura Education

Basic Concepts & Terminology

Difference between “Cloud” and Internet :

1. As a specific environment used to remotely provision IT resources, a cloud has a finite boundary.
whereas the Internet provides open access to many Web-based IT resources, a cloud is typically privately owned and offers access to IT resources that is metered.

2. Much of the internet is dedicated to the access of content-based IT resources published via the World Wide Web,
on the other hand, are dedicated to supplying back-end processing capabilities and user-based access to these capabilities.

Basic Concepts & Terminology

□ IT Resources:

- An IT resource is a physical or virtual IT-related artifact that can be either software-based, such as virtual server or a custom software program, or hardware-based, such as a physical server or a network device.

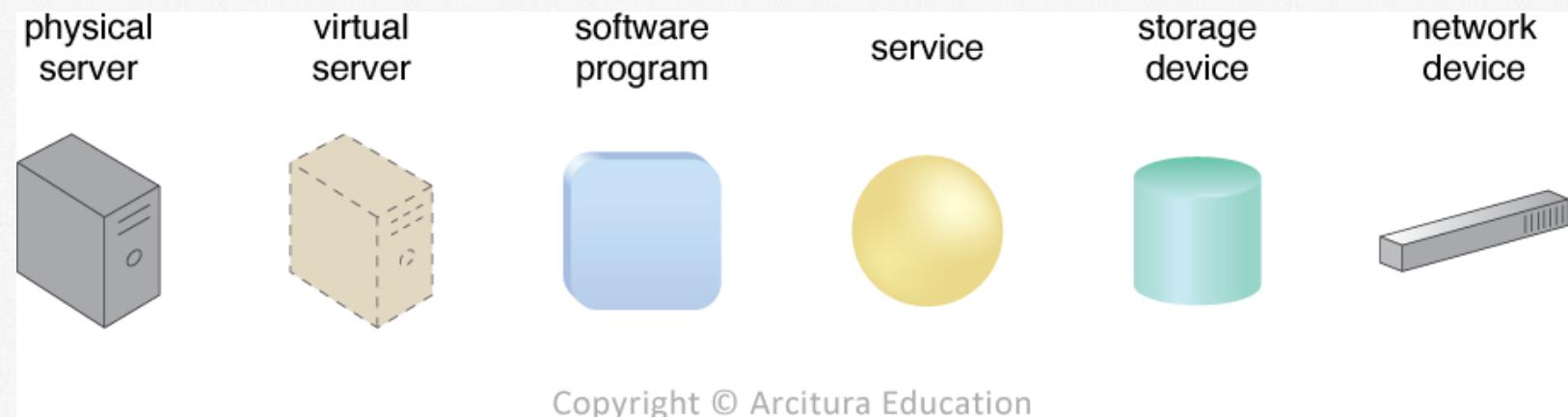


Figure: Examples of common IT resources and their corresponding symbols

Basic Concepts & Terminology

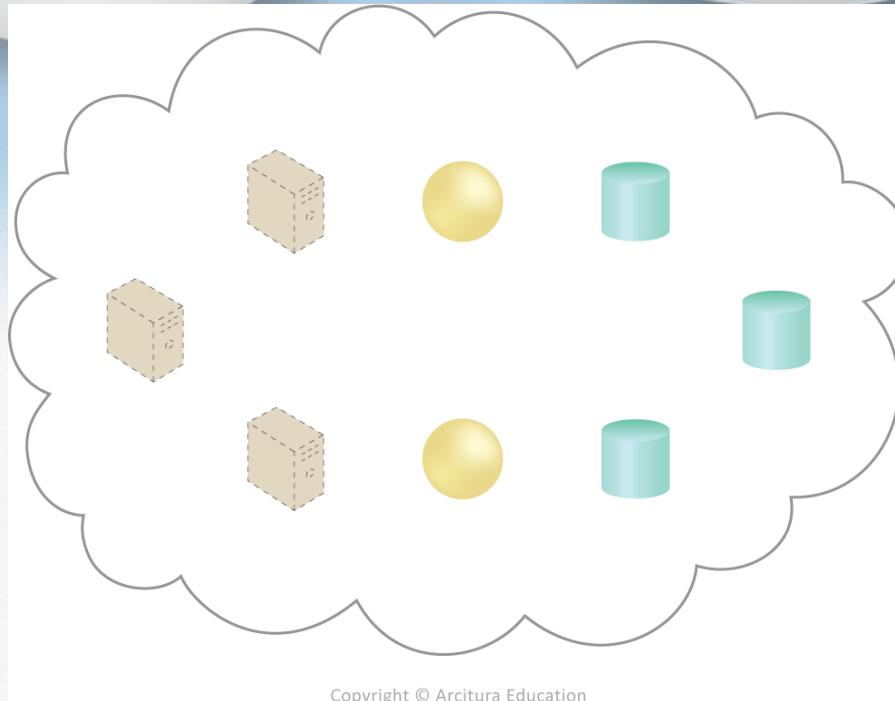


Figure: A cloud is hosting eight IT resources: three virtual servers, two cloud services, and three storage devices.

Basic Concepts & Terminology

On-Premise:

- As a distinct and remotely accessible environment, a cloud represents an option for the deployment of IT resources.
- An IT resources that hosted in a conventional IT enterprise within an organizational boundary (that does not specifically represent a cloud) is considered to be located on the premises of the IT enterprise, or on-premise for short.
- ***“on the premises of a controlled IT environment that is not cloud-based”***
- ***An IT resources that is on-premise cannot be a cloud-based, and vice-versa.***

Basic Concepts & Terminology

Scaling:

Scaling, from an IT recourse perspective, represents the ability of the IT resource to handle increased or decreased usage demands.

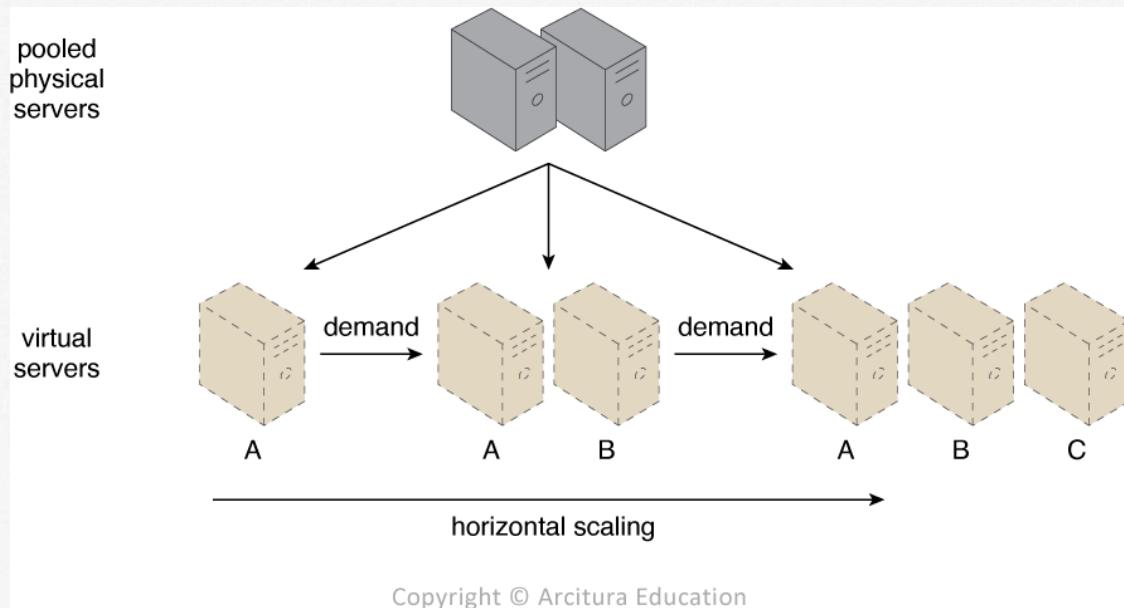
The following are types of scaling:

- Horizontal Scaling : scaling out and scaling in
- Vertical Scaling : scaling up and scaling down

Basic Concepts & Terminology

□ Horizontal Scaling :

- ✓ The allocating or releasing of IT resources that are of the same type is referred to as **horizontal scaling**, as shown in fig.
- ✓ The horizontal allocation of resources is referred to as **scaling out** and the horizontal releasing of resources is referred to as **scaling in**.
- ✓ Horizontal scaling is a common form of scaling within cloud environment.

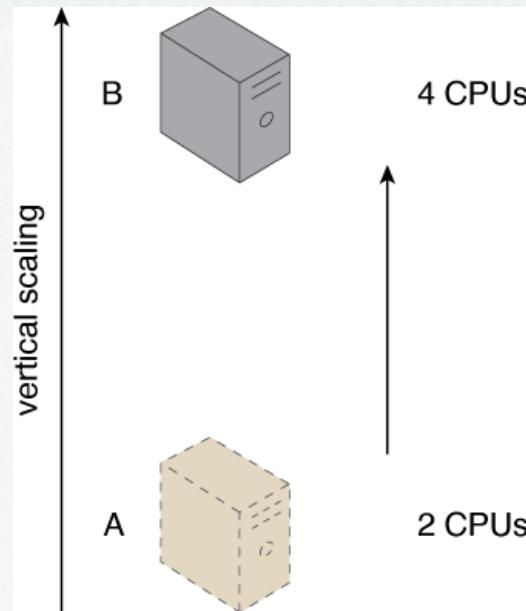


Basic Concepts & Terminology

□ Vertical Scaling :

- ✓ When an existing IT resources is replaced by another with higher or lower capacity, vertical scaling is considered to have occurred, as shown in the fig.
- ✓ The replacing of an IT resource with higher capacity is referred to as scaling up and the replacing an IT resource with lower capacity is referred to as scaling down.

Figure: An IT resource (a virtual server with two CPUs) is scaled up by replacing it with a more powerful IT resource with increased capacity for data storage (a physical server with four CPUs)

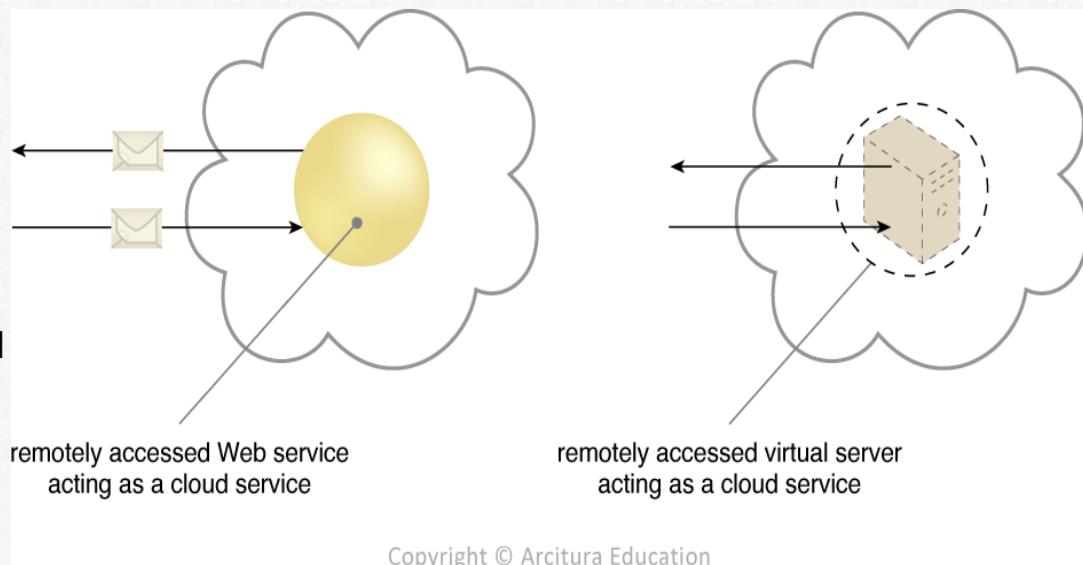


Basic Concepts & Terminology

□ Cloud Service:

- A **cloud service** is any IT resource that is made remotely accessible via a cloud.
- The term “service” within the context of cloud computing is especially broad.
- A cloud service can exist as a simple web-based software program with a technical interface invoked via the use of a messaging protocol, or as a remote access point for administrative tools or large environment and other IT resources.

Figure: A cloud service with a published technical interface is being accessed by a consumer outside of the cloud (left). A cloud service that exists as a virtual server is also being accessed from outside of the cloud's boundary (right)



Copyright © Arcitura Education

Basic Concepts & Terminology

□ Cloud Service consumers:

- A **cloud service consumer** is a temporary runtime role assumed by a software program when it accesses a cloud service.
- As shown in the fig. common type of cloud service consumers can include software programs and services capable of remotely accessing cloud services with published service contracts.



Figure: Examples of cloud service consumers. Depending on the nature of a given diagram, an artifact labeled as a cloud service consumer may be a software program or a hardware device (in which case it is running a software program capable of acting as a cloud service consumer).

Goals & Benefits

- Reduced Investments and proportional costs
- Increased scalability
- Increased availability and reliability

Cloud Reference Model

- The design of the NIST cloud computing reference architecture serves the following objectives:
 - to illustrate and understand the various cloud services in the context of an overall cloud computing conceptual model;
 - to provide a technical reference to USG agencies and other consumers to understand, discuss, categorize and compare cloud services;
 - and to facilitate the analysis of candidate standards for security, interoperability, and portability and reference implementations.

Cloud Reference Model

3. Cloud Computing Reference Architecture

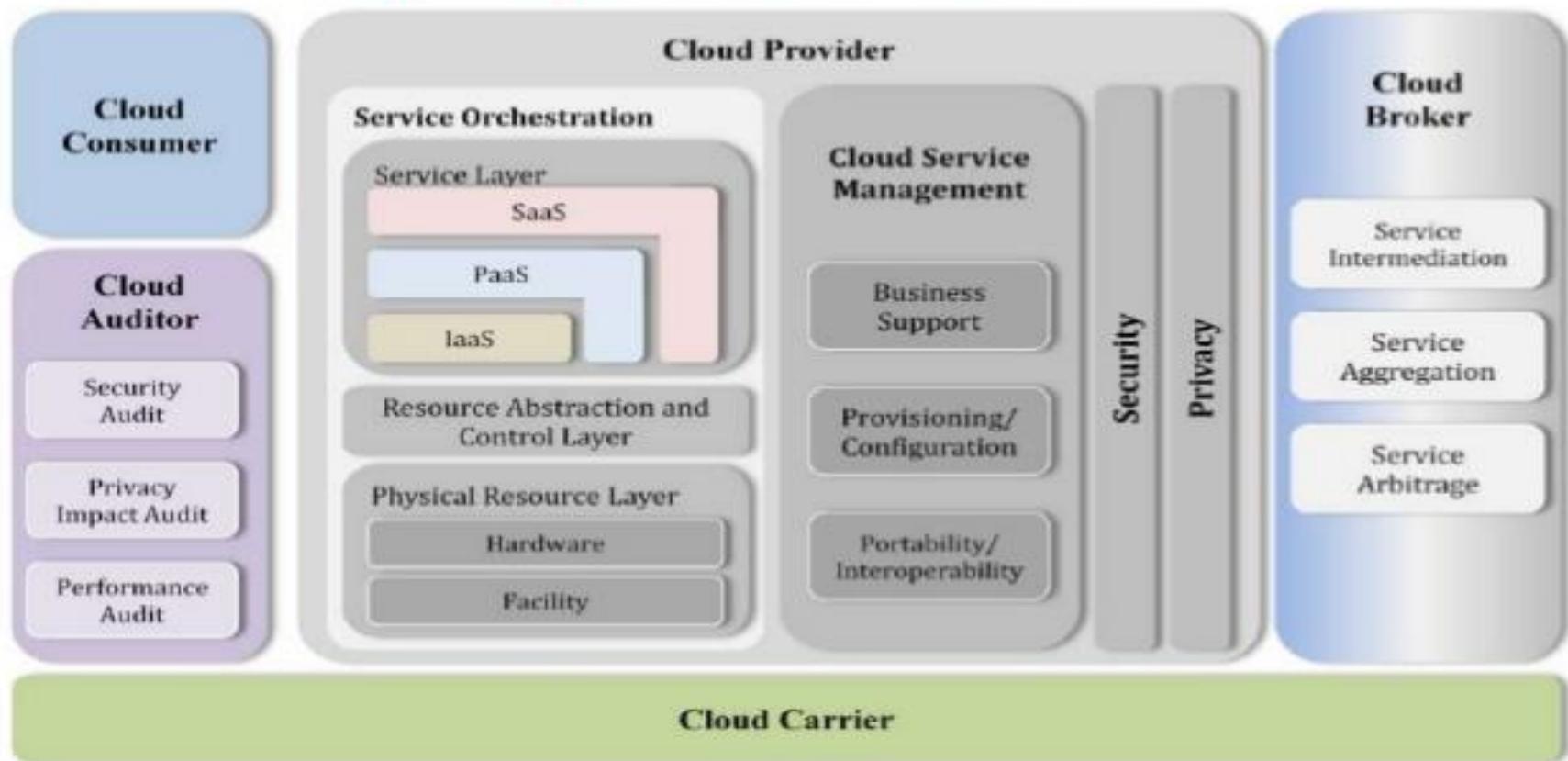


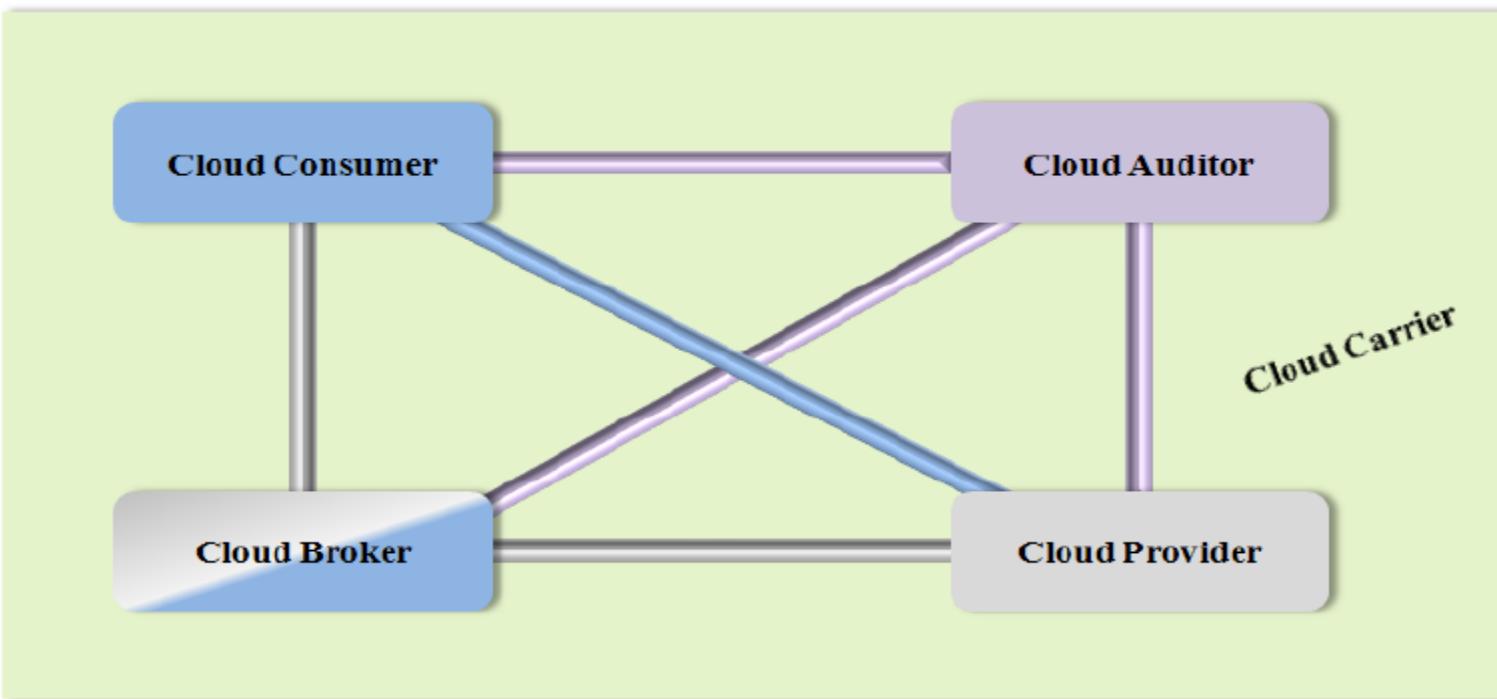
Figure 1: The Conceptual Reference Model

Roles in Cloud Computing

Actor	Definition
Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, <i>Cloud Providers</i> .
Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between <i>Cloud Providers</i> and <i>Cloud Consumers</i> .
Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from <i>Cloud Providers</i> to <i>Cloud Consumers</i> .

Table 1: Actors in Cloud Computing

Communication between actors



- The communication path between a cloud provider and a cloud consumer
- The communication paths for a cloud auditor to collect auditing information
- The communication paths for a cloud broker to provide service to a cloud consumer

Figure 2: Interactions between the Actors in Cloud Computing

Cloud Usage Scenario Examples

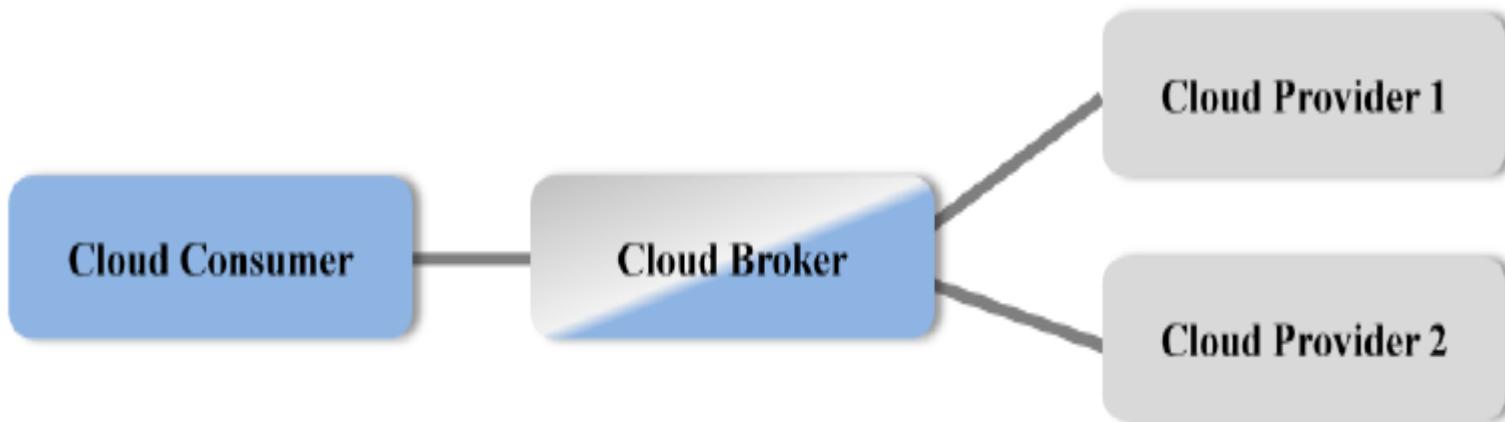
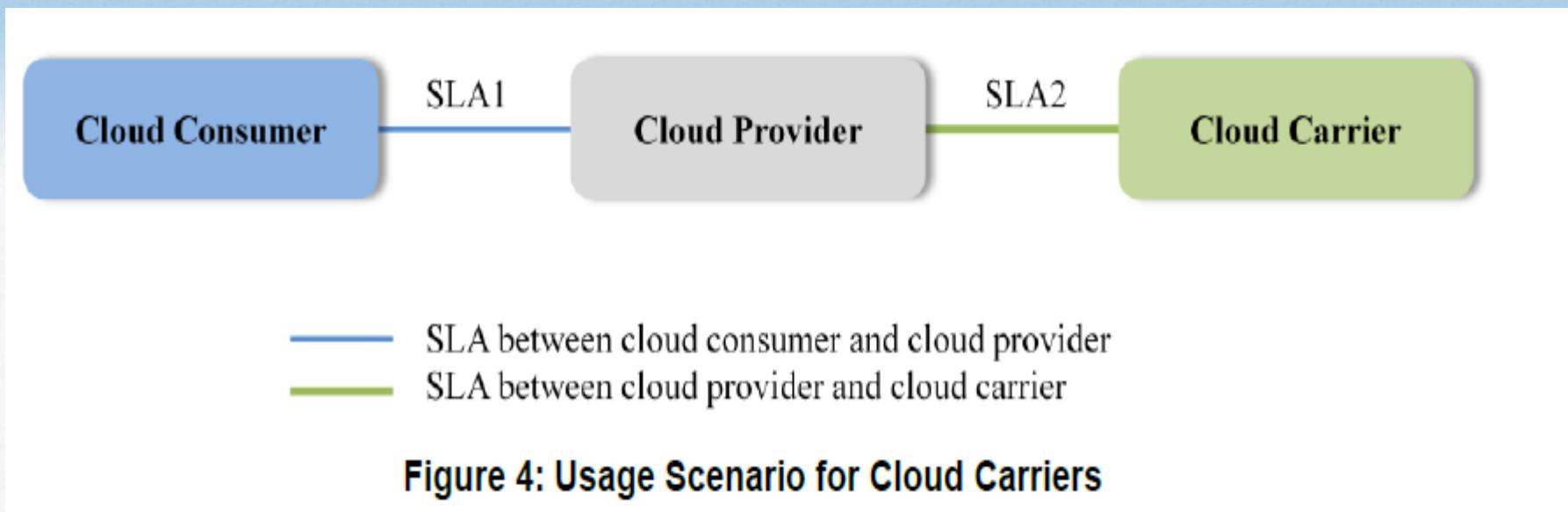


Figure 3: Usage Scenario for Cloud Brokers

Cloud Usage Scenario Examples



Cloud Usage Scenario Examples

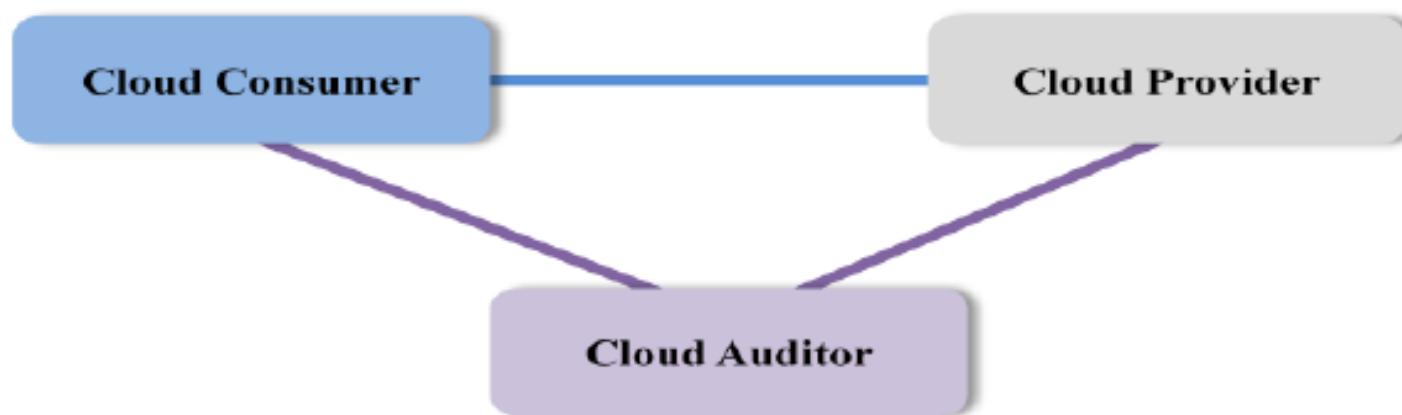
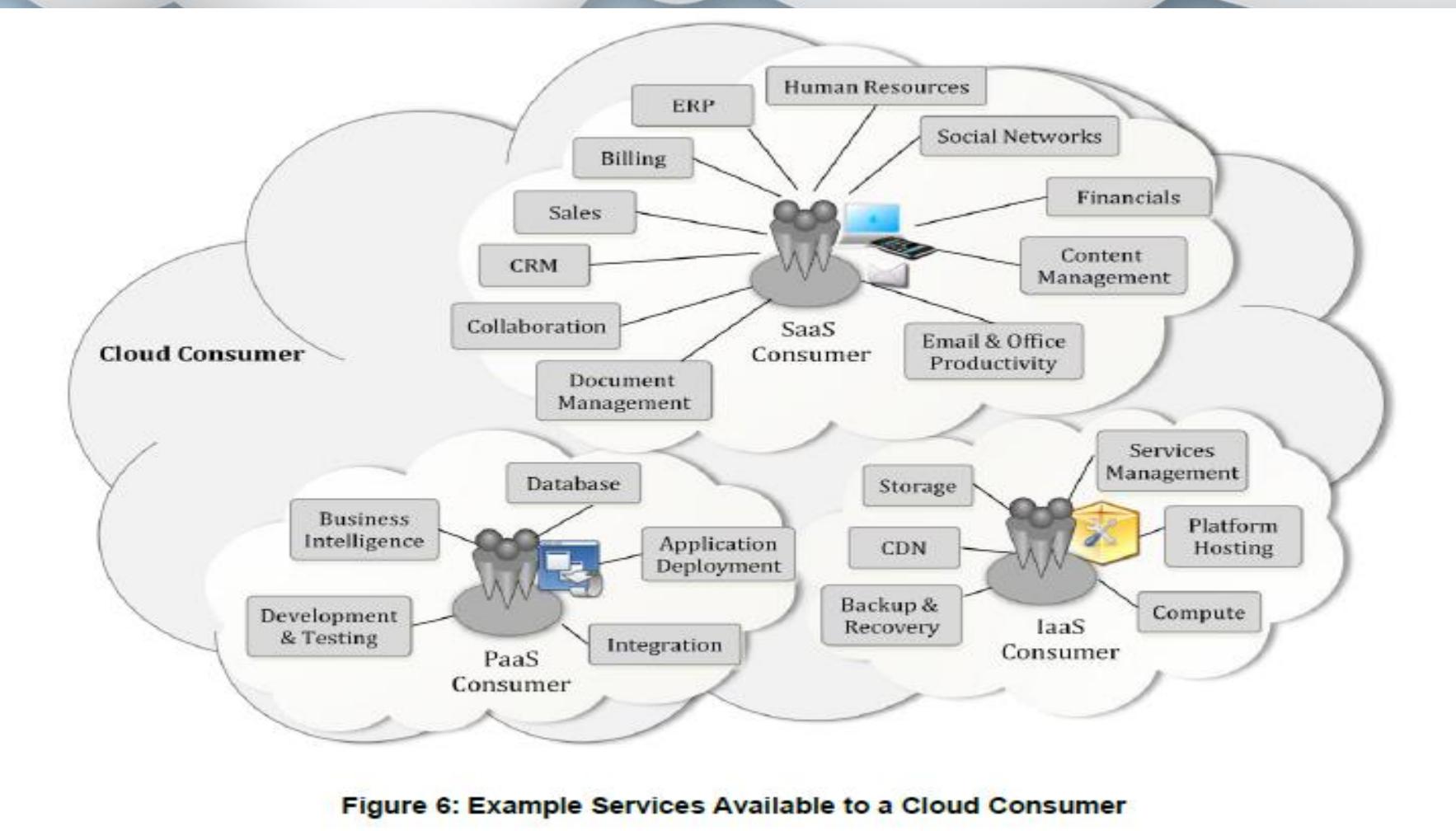


Figure 5: Usage Scenario for Cloud Auditors

Cloud Consumer



Cloud Service Provider

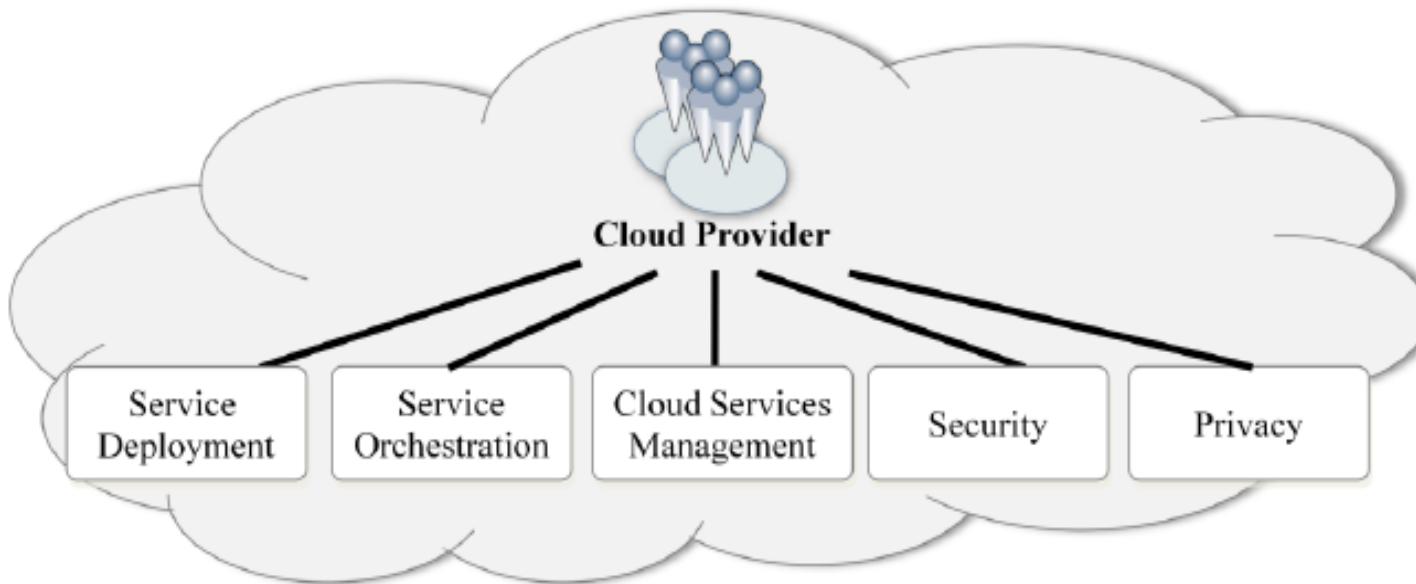


Figure 7: Cloud Provider - Major Activities

Cloud Auditor

- A cloud auditor is a party that can perform an independent examination of cloud service controls with the intent to express an opinion thereon.
- Audits are performed to verify conformance to standards through review of objective evidence.
- A cloud auditor can evaluate the services provided by a cloud provider in terms of security controls, privacy impact, performance, etc.
- **COMPANY EXAMPLES:**
 - ISACA (Infor. System audit & Control Association)
 - Product of ISACA is COBIT 5. (Control Objectives for Information and Related Technologies)
 - CloudChecker.com
 - Etc....

Cloud Broker

- As cloud computing evolves, the integration of cloud services can be too complex for cloud consumers to manage.
- A cloud consumer may request cloud services from a cloud broker, instead of contacting a cloud provider directly.
- A cloud broker is an entity that manages the use, performance and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers.
- Cloud Broker Companies: (*Total market is of \$2.03 billions in 2018*)
 - Appirio
 - AWS Marketplace
 - Bluewolf
 - CloudCompare
 - RED HAT Cloudforms is multicloud management platform.

Cloud Broker

In general, a cloud broker can provide services in three categories:

- **Service Intermediation:** A cloud broker enhances a given service by improving some specific capability and providing value-added services to cloud consumers. The improvement can be managing access to cloud services, identity management, performance reporting, enhanced security, etc.
- **Service Aggregation:** A cloud broker combines and integrates multiple services into one or more new services. The broker provides data integration and ensures the secure data movement between the cloud consumer and multiple cloud providers.
- **Service Arbitrage:** Service arbitrage is similar to service aggregation except that the services being aggregated are not fixed. Service arbitrage means a broker has the flexibility to choose services from multiple agencies. The cloud broker, for example, can use a credit-scoring service to measure and select an agency with the best score.

Cloud Carrier

- A cloud carrier acts as an intermediary that provides connectivity and transport of cloud services between cloud consumers and cloud providers.
- Cloud carriers provide access to consumers through network, telecommunication and other access devices. For example, cloud consumers can obtain cloud services through network access devices, such as computers, laptops, mobile phones, mobile Internet devices (MIDs), etc.
- The distribution of cloud services is normally provided by network and telecommunication carriers or a *transport agent*, where a transport agent refers to a business organization that provides physical transport of storage media such as high-capacity hard drives.

Scope of Control between Provider & Consumer

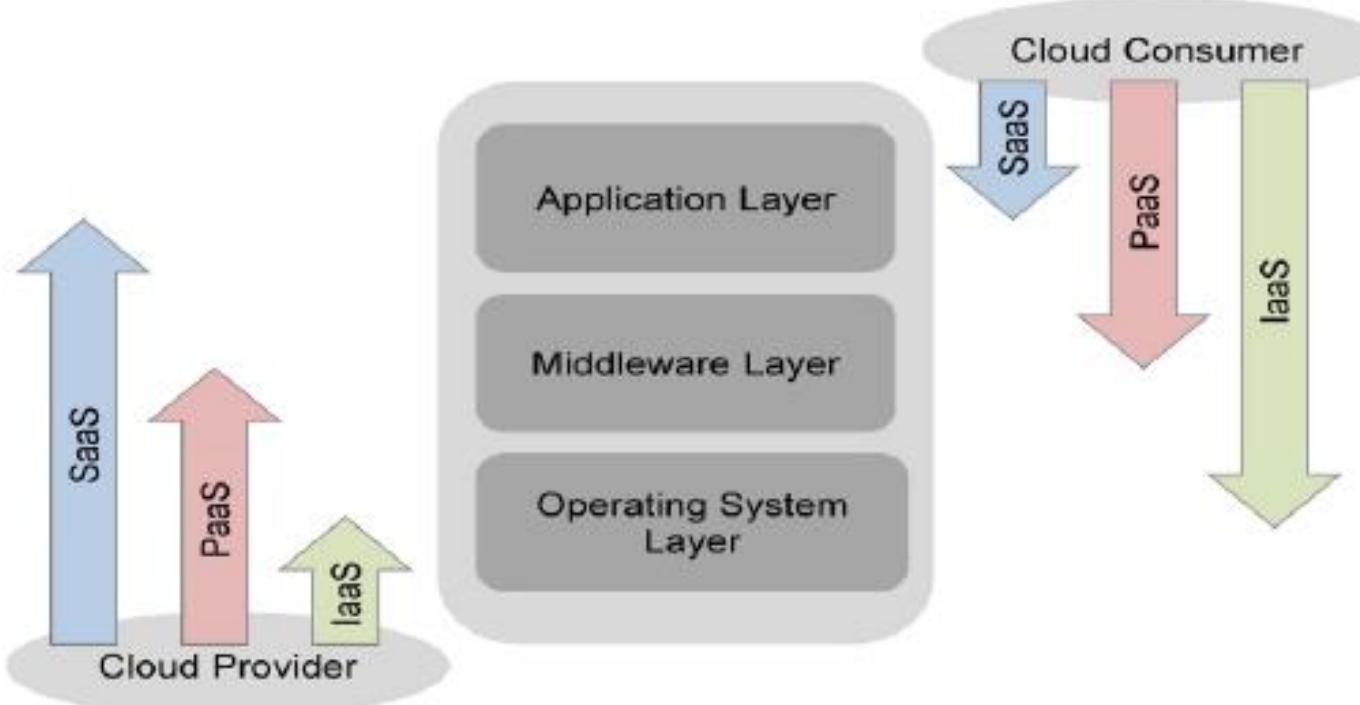
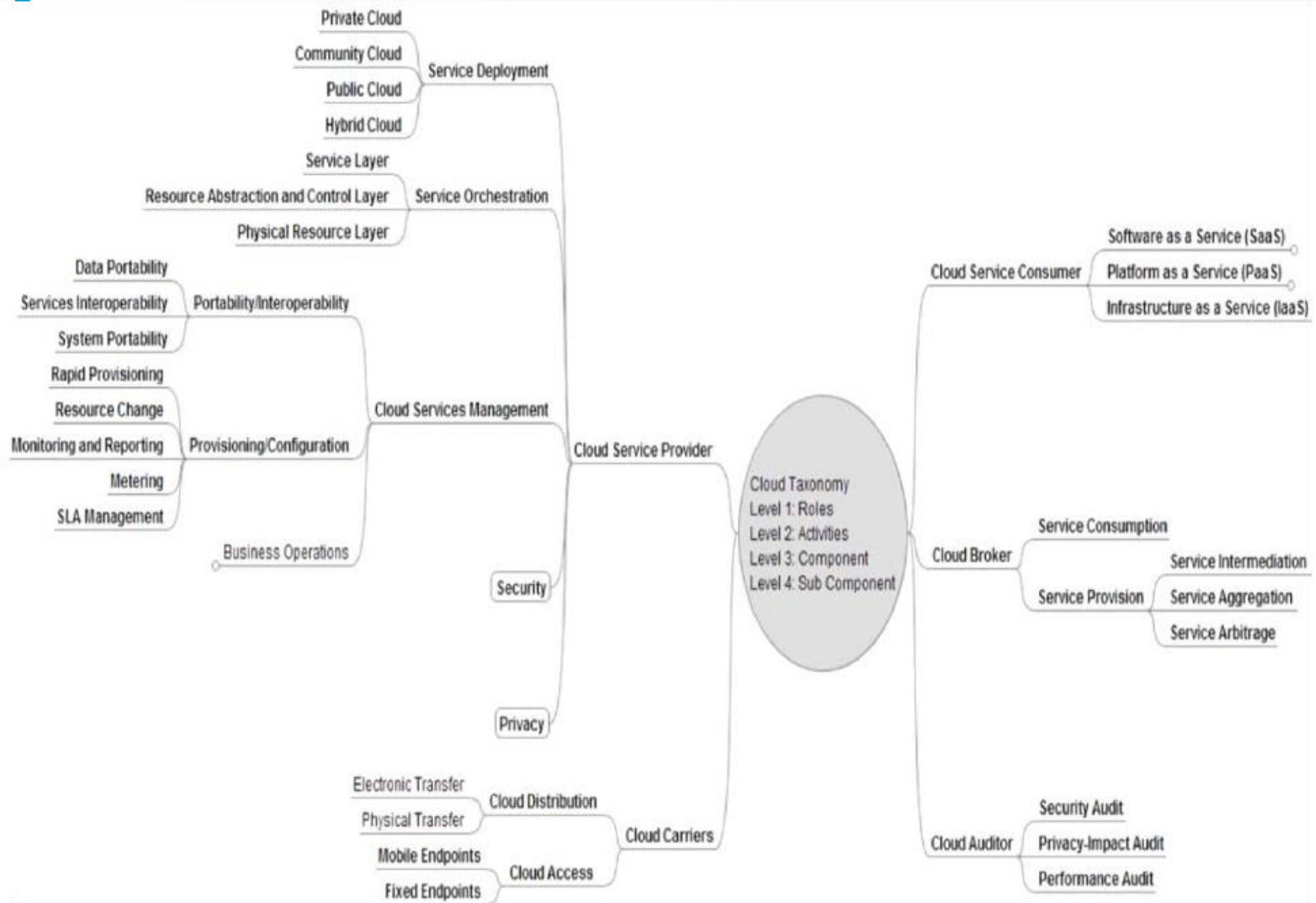


Figure 8: Scope of Controls between Provider and Consumer

Cloud Taxonomy

- Taxonomy is the science of categorization, or classification, of things based on a predefined system.
- Typically, taxonomy contains a controlled vocabulary with a hierarchical tree-like structure.



Cloud Taxonomy

- Figure presents the taxonomy associated with the cloud computing reference architecture discussed in this document. In the figure, a four-level taxonomy is presented to describe the key concepts about cloud computing.
- **Level 1:** *Role*, which indicates a set of obligations and behaviors as conceptualized by the associated actors in the context of cloud computing.
- **Level 2:** *Activity*, which entails the general behaviors or tasks associated to a specific role.
- **Level 3:** *Component*, which refer to the specific processes, actions, or tasks that must be performed to meet the objective of a specific activity.
- **Level 4:** *Sub-component*, which present a modular part of a component.

Roles

- **Cloud Consumer** - Person or organization that maintains a business relationship with, and uses service from, Cloud Service Providers.
-
- **Cloud Provider** – Person, organization or entity responsible for making a service available to service consumers.
-
- **Cloud Carrier** – The intermediary that provides connectivity and transport of cloud services between Cloud Providers and Cloud Consumers.
-
- **Cloud Broker** – An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.
-
- **Cloud Auditor** – A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.

Cloud Service provider activities

1. **Service Deployment** – All of the activities and organization needed to make a cloud service available
2. **Service Orchestration** - Refers to the arrangement, coordination and management of cloud infrastructure to provide different cloud services to meet IT and business requirements.
3. **Cloud Service Management** – Cloud Service Management includes all the service-related functions that are necessary for the management and operations of those services required by or proposed to customers.

Cloud Service provider activities

4. **Security** – Refers to information security. “information security” means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide:
 - (A) **integrity**, which means guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity;
 - (B) **confidentiality**, which means preserving authorized restrictions on access and disclosure, including means for protecting personal privacy and proprietary information;
 - (C) **availability**, which means ensuring timely and reliable access to and use of information.
5. **Privacy** - Information privacy is the assured, proper, and consistent collection, processing, communication, use and disposition of disposition of personal information (PI) and personally-identifiable information (PII) throughout its life cycle.

Cloud Carrier activities:

1. **Cloud Distribution** – The process of transporting cloud data between Cloud Providers and Cloud Consumers.
2. **Cloud Access** – To make contact with or gain access to Cloud Services.

Risks & Challenges

- Increased security vulnerabilities.

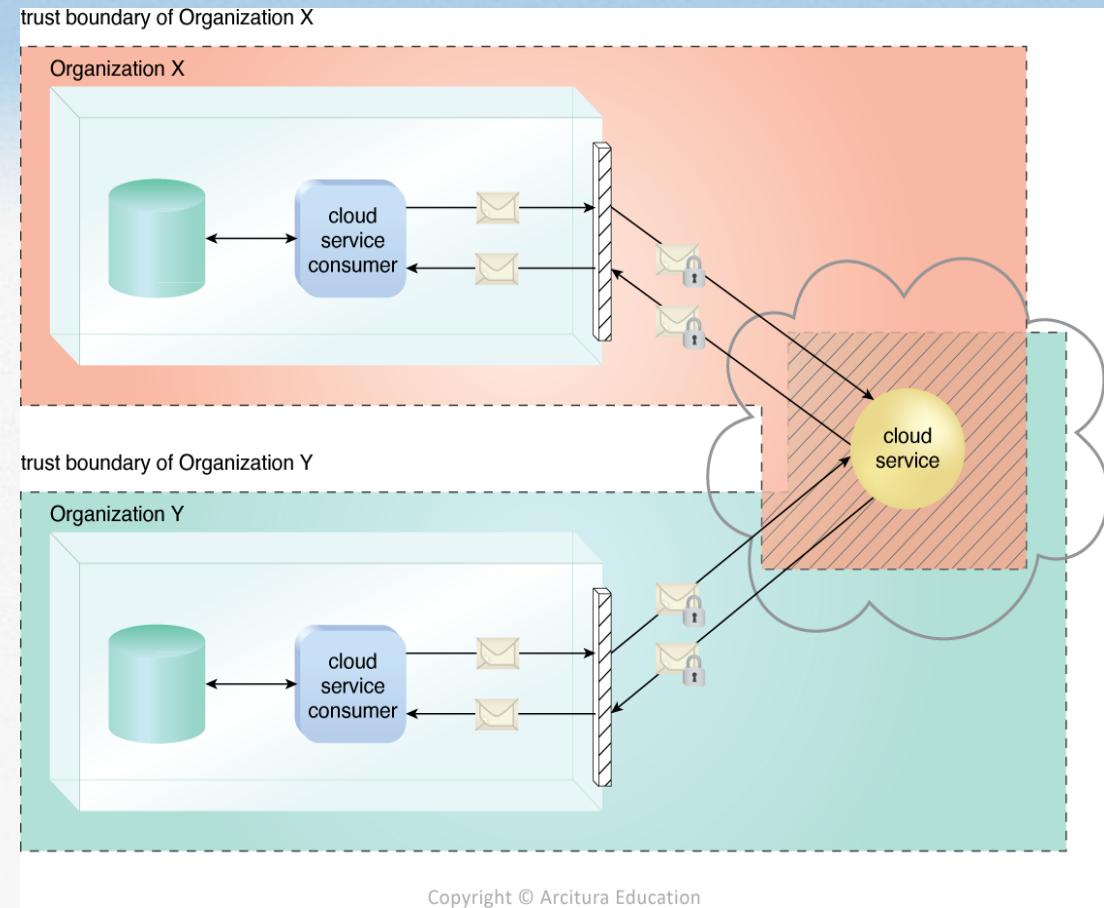
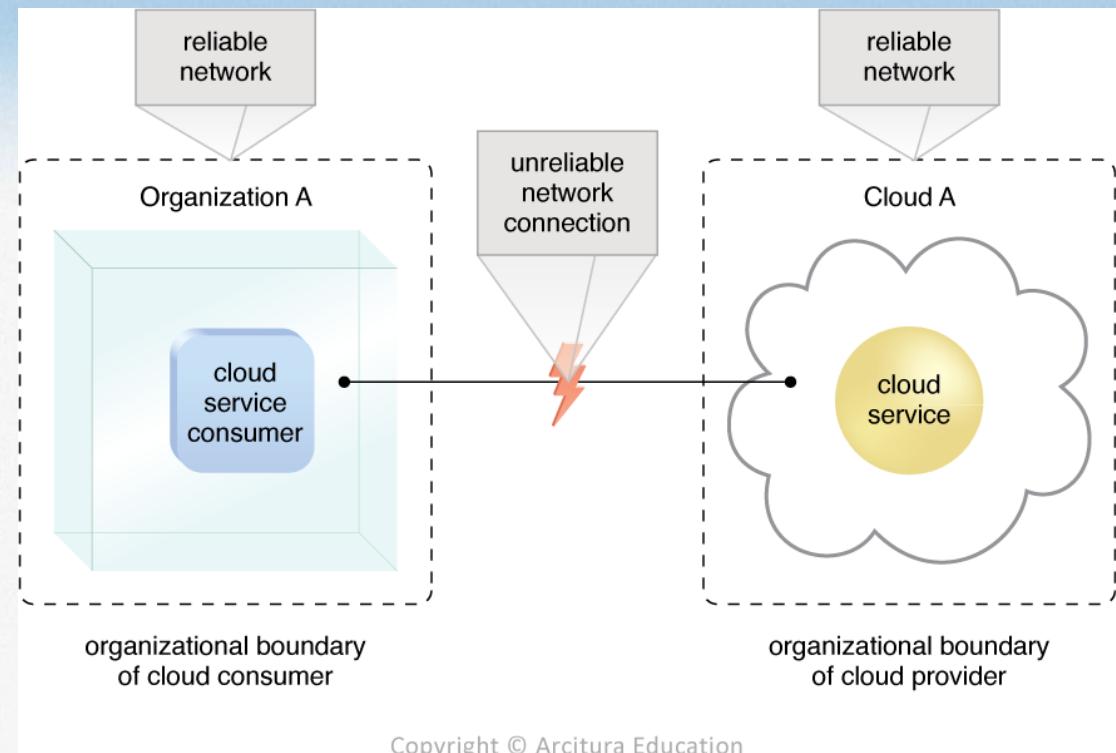


Figure: The shaded area with diagonal lines indicates the overlap of two organization's trust boundaries.

Risks & Challenges

- Reduced Operational Governance Control

Figure: An unreliable network connection compromises the quality of communication between cloud consumer and cloud provider environments.



Copyright © Arcitura Education

Risks & Challenges

- Limited Portability Between Cloud Providers

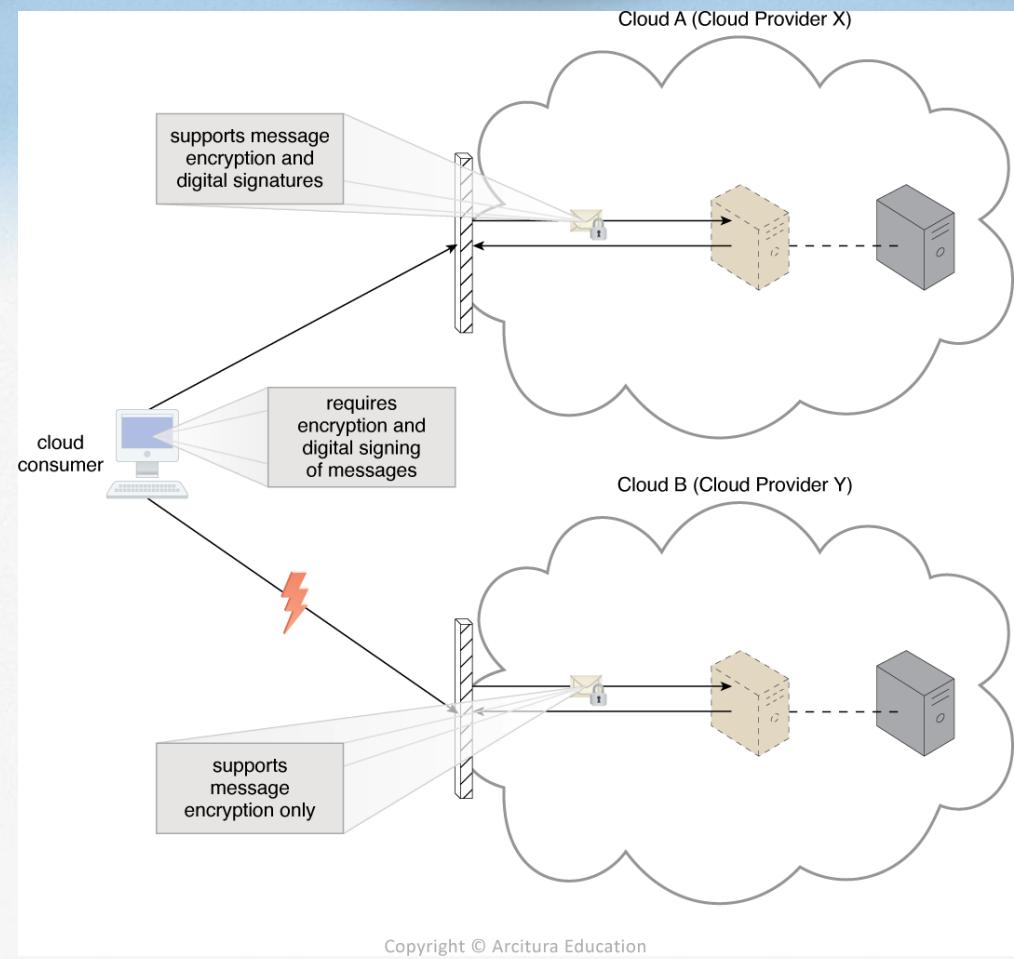


Figure: A cloud consumers application has a decreased level of portability when accessing a potential migration from Cloud A to Cloud B, because the cloud provider of Cloud B does not support the same security technologies as Cloud A.

Risks & Challenges

- Multiregional Compliance and Legal Issues

Summary of Key Points

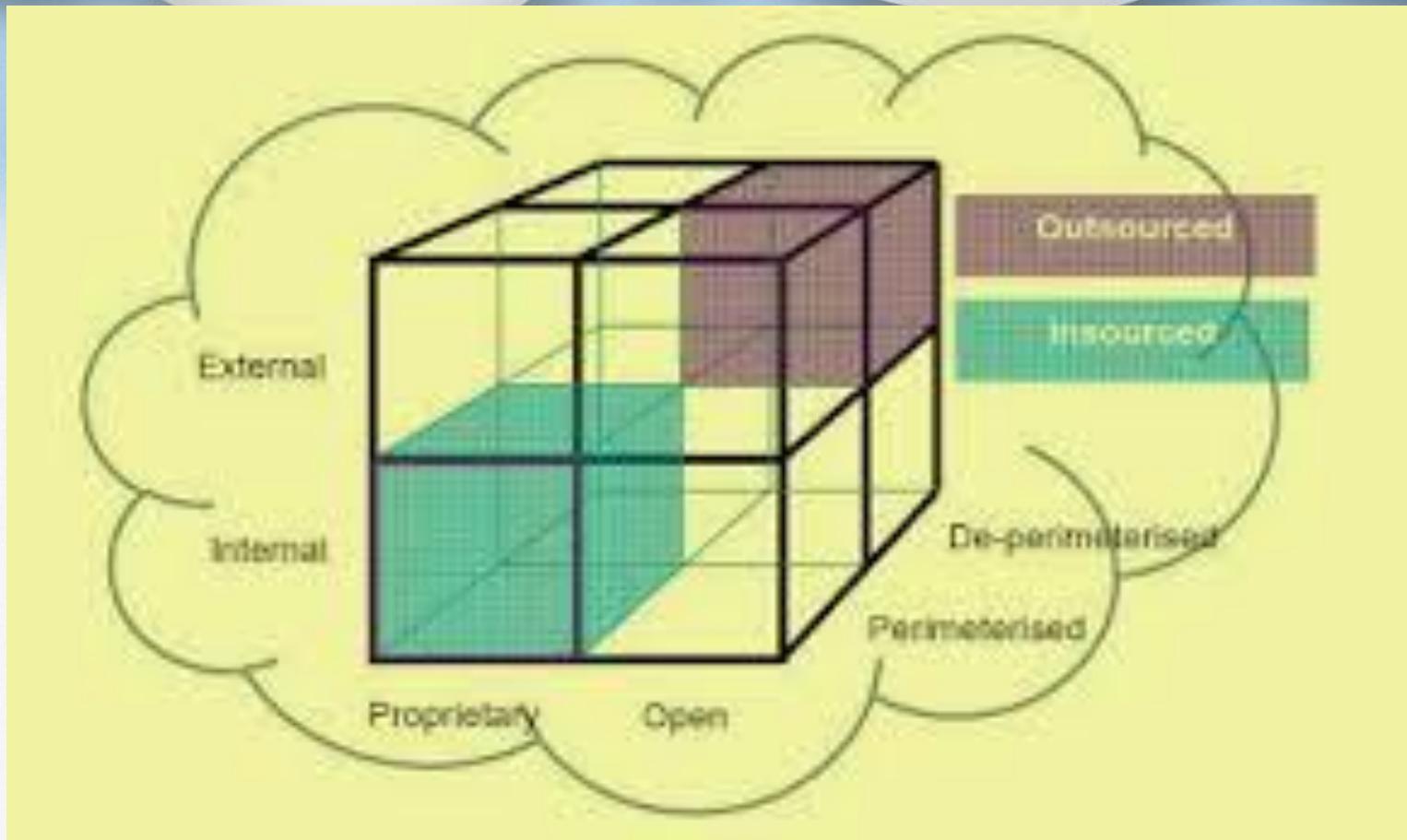
- Cloud environments can introduce distinct security challenges, some of which pertain to overlapping trust boundaries imposed by a cloud provider sharing IT resources with multiple cloud consumers.
- A cloud consumer's operational governance can be limited within cloud environments due to the control exercised by a cloud provider over its platforms.
- The portability of cloud-based IT resources can be inhibited by dependencies upon proprietary characteristics imposed by a cloud.
- The geographical location of data and IT resources can be out of cloud consumer's control when hosted by a third-party cloud provider. This can introduce various legal and regulatory compliance concern.

Muddiest Point

Cloud Characteristics

- The following six specific characteristic are common to the majority of cloud environments:
 - 1) On-demand usage
 - 2) Ubiquitous access
 - 3) Multitenancy (and resource pooling)
 - 4) Elasticity
 - 5) Measured usage
 - 6) Resiliency

Cloud Cube Model



Cloud Cube Model

- The Jericho Forum has designed the Cloud Cube Model to help select cloud formations for security cooperation.
- Their fascinating new cloud model helps IT managers and business tycoons assess the benefits of cloud computing.
- The Cloud Cube Model looks at the several different "cloud formations".
- They amount to the cloud service and deployment models.
- The sourcing dimension addresses the delivery of service.
- The Cloud Cube Model may be designed to let users show that the traditional notion of network ranges & its boundaries with network firewall no longer applies in Cloud computing.

Cloud Cube Model

- **Cloud Cube Model**, designed and developed by **Jericho forum**.
- Which helps to categorize the cloud network based on the four-dimensional factor: Internal/External, Proprietary/Open, De-Perimeterized/Perimeterized, and Insourced/Outsourced.

• **Dimension 1: Internal/External**

- This dimension defines the physical location of the data; where does the cloud form exist – inside or outside organization boundaries? If the cloud form is within the organization's physical boundaries, then it is internal.
- If it is outside the organization's physical boundaries, then it is external. It's important to note that the assumption that internal is necessarily more secure than external is false. The most secure usage model is the effective use of both internal and external cloud forms.

ii. Proprietary/Open

- The second type of cloud formation is **proprietary and open**. The proprietary or open dimension states about the state of ownership of the **cloud technology** and interfaces. It also tells the degree of interoperability, while enabling data transportability between the system and other cloud forms..
- The **proprietary dimension** means, that the organization providing the **service is securing** and protecting the data under their ownership.
- The **open dimension** is using a technology in which there are more suppliers. Moreover, the user is not constrained in being able to share the data and collaborate with selected partners using the open technology.ther cloud forms.

iii. De-Perimeterized / Perimeterized

- The third type of cloud formation is **De-perimeterized and Perimeterized**.
- To reach this form, the user needs collaboration oriented architecture and Jericho forum commandments.
The Perimeterised and De-perimeterized dimension tells us whether you are operating inside your traditional it mindset or outside it.
- **Perimeterized dimension** means, continuing to operate within the traditional it boundary, orphan signaled by network firewalls. With the help of VPN and operation of the virtual server in your own IP domain, the user can extend the organizations perimeter into external Cloud Computing domain. This means that the user is making use of the own services to control access.

iii. De-Perimeterized / Perimeterized

- **De-perimeterized dimension** means the system perimeter is architected on the principles outlined in the Jericho forums commandments. In De-perimeterized dimension, the data will be encapsulated with metadata and mechanisms, which will further help to protect the data and limit the inappropriate usage.

iv. Insourced/Outsourced

- The **Insourced and outsourced dimensions** have two states in each of the eight cloud forms. In the *outsourced dimension* the services provided by the third party, whereas in the *insourced dimension* the services provided by the own staff under the control.
- In this few organizations that are traditional bandwidth software or hardware, providers will run fluently on becoming cloud service providers.
- The organizations which are seeking to procedure cloud services must have the ability to set legally binding collaboration agreement. In this, an organization should ensure that data is deleted from the service provider's Infrastructure.

Questions For Cloud Cube Model

- The Jericho forum states that there are three key questions, which a customer should ask their **Cloud Computing** supplier. So, that they must be aware that the data is secure and protected. The three questions are-
- Q 1. Wherein the cloud cube model is the cloud supplier operating while providing the services?
- Q 2. How will the clouds suppliers get a surety when the customer is using services in a cloud from that has maintained the features as per the expectations?
- Q 3. How can a customer ensure that the data which is stored in the cloud services will be available at the time of mishappenings such as bankruptcy or change in business direction?

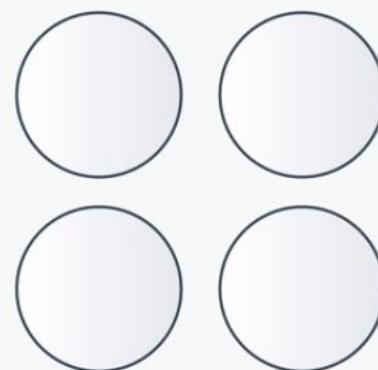
MONOLITHIC, SOA & MICROSERVICES

Monolithic vs. SOA vs. Microservices



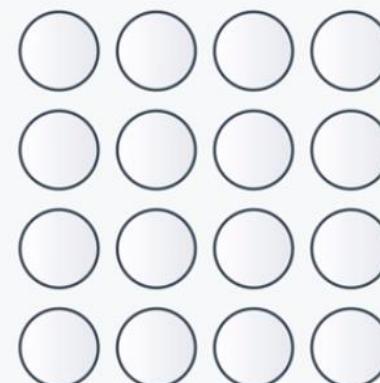
Monolithic

Single Unit



SOA

Coarse-grained



Microservices

Fine-grained

Activate Windows
Go to Settings to activate Windows.



SOA Vs Microservice



SOA is like an orchestra where each artist is performing with his/her instrument while the music director guides them all.

With Microservices each dancer is independent and know what they need to do. If they miss some steps they know how to get back on the sequence.

Activate Windows
Go to Settings to activate Windows.

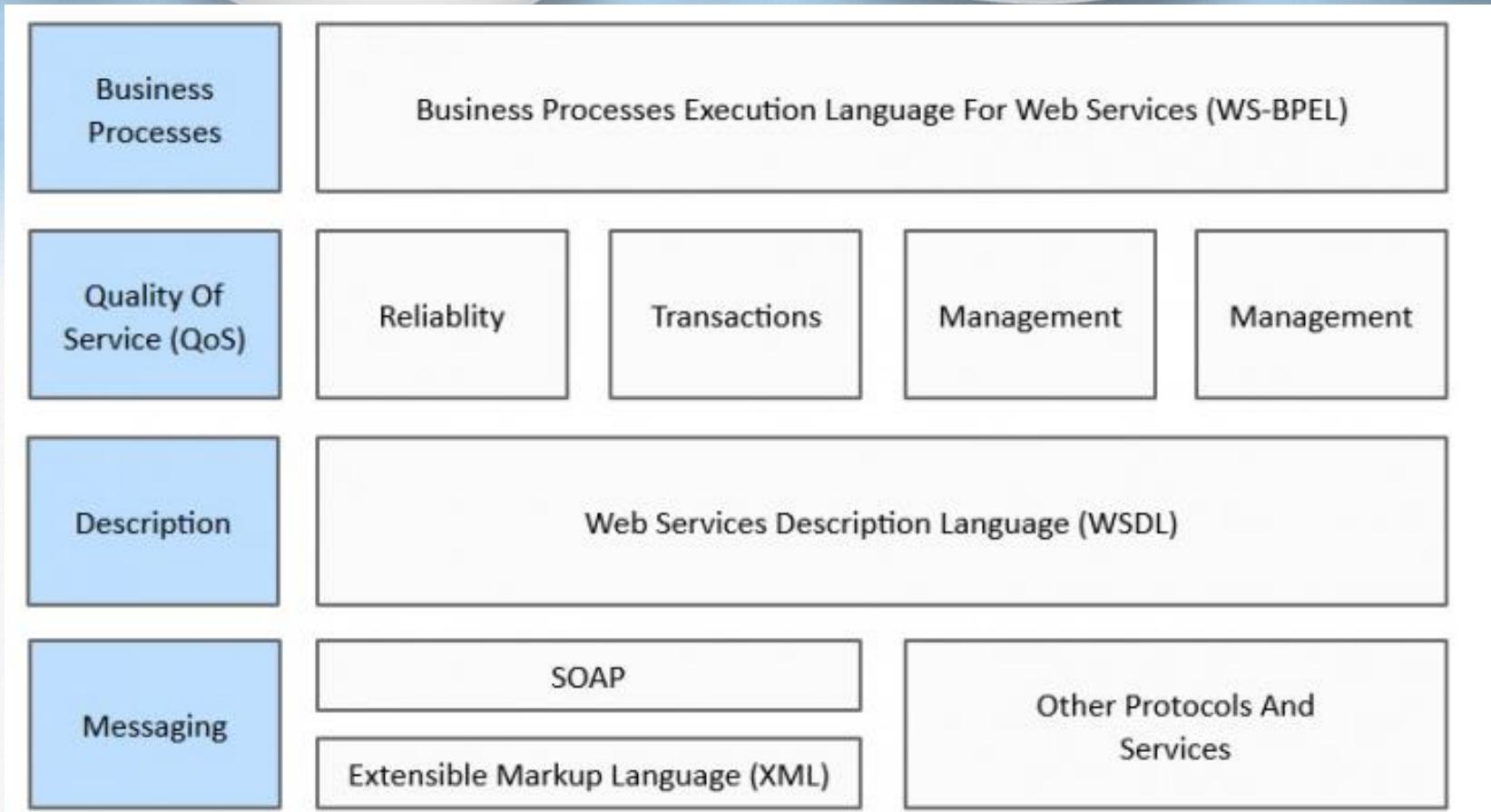
Service Oriented Architecture

- **Service-oriented architecture (SOA)** is a style of software design where services are provided to the other components by application components, through a communication protocol over a network.
- A SOA service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently, such as retrieving a credit card statement online. SOA is also intended to be independent of vendors, products and technologies.

Service Oriented Architecture

- A service has four properties according to one of many definitions of SOA:
 - It logically represents a business activity with a specified outcome.
 - It is self-contained.
 - It is a black box for its consumers, meaning the consumer does not have to be aware of the service's inner workings.
 - It may consist of other underlying services.
- Different services can be used in conjunction to provide the functionality of a large software application, a principle SOA shares with modular programming.

SOA Architecture and Protocols



Service Oriented Architecture

- SOA architecture is viewed as five horizontal layers. These are described below:
 - 1) **Consumer Interface Layer:** These are GUI based apps for end users accessing the applications.
 - 2) **Business Process Layer:** These are business-use cases in terms of application.
 - 3) **Services Layer:** These are whole-enterprise, in service inventory.
 - 4) **Service Component Layer:** are used to build the services, such as functional and technical libraries.
 - 5) **Operational Systems Layer:** It contains the data model.

Service Oriented Architecture

- A manifesto was published for service-oriented architecture in October, 2009. This came up with six core values which are listed as follows:
 - 1) **Business value** is given more importance than technical strategy.
 - 2) **Strategic goals** are given more importance than project-specific benefits.
 - 3) **Intrinsic interoperability** is given more importance than custom integration.
 - 4) **Shared services** are given more importance than specific-purpose implementations.
 - 5) **Flexibility** is given more importance than optimization.
 - 6) **Evolutionary refinement** is given more importance than pursuit of initial perfection.

Guiding Principles of SOA:

- **Standardized service contract:** Specified through one or more service description documents.
- **Loose coupling:** Services are designed as self-contained components, maintain relationships that minimize dependencies on other services.
- **Abstraction:** A service is completely defined by service contracts and description documents. They hide their logic, which is encapsulated within their implementation.
- **Reusability:** Designed as components, services can be reused more effectively, thus reducing development time and the associated costs.

Guiding Principles of SOA:

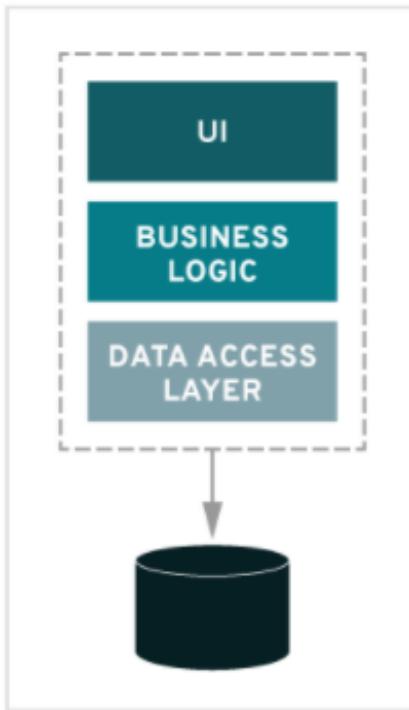
- **Autonomy:** Services have control over the logic they encapsulate and, from a service consumer point of view, there is no need to know about their implementation.
- **Discoverability:** Services are defined by description documents that constitute supplemental metadata through which they can be effectively discovered. Service discovery provides an effective means for utilizing third-party resources.
- **Composability:** Using services as building blocks, sophisticated and complex operations can be implemented. Service orchestration and choreography provide a solid support for composing services and achieving business goals.

Microservices

- **Microservice** architecture – a variant of the service-oriented architecture (SOA) structural style – arranges an application as a collection of loosely coupled services.
- In a microservices architecture, services are fine-grained and the protocols are lightweight.

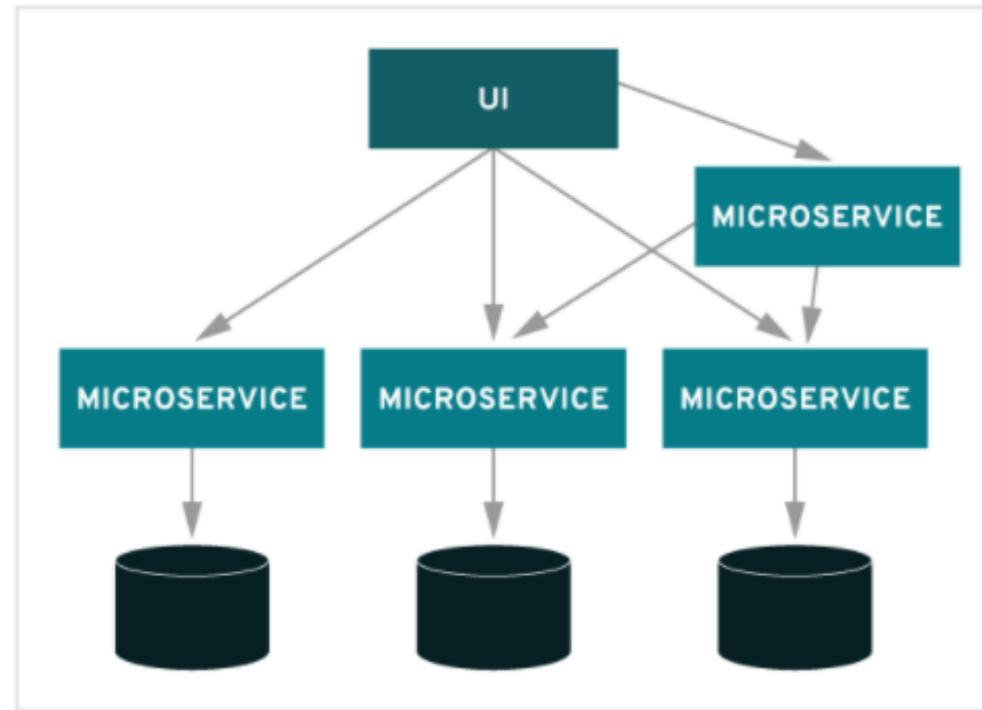
Microservices

MONOLITHIC



MICROSERVICES

VS.



Microservices Features



WEB 1.0 To 5.0

- **Web 0.0** – Developing the internet
- **Web 1.0** – The shopping carts & static web
- **Web 2.0** – The writing and participating web
- **Web 3.0** – The semantic executing web
- **Web 4.0** – “Mobile Web”
- **Web 5.0**- Open, Linked and Intelligent
Web = Emotional Web. “The next web”

THANK YOU!!!!