



**Course:** SBL: Cloud Computing

**Course code:** CSL605

**Year:** TE **SEM:** VI

### Experiment No. 01

AIM:- To study and implement Infrastructure as a Service using AWS

Name:

Roll Number:

Date of Performance:

Date of Submission:

### Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission.	10	
<b>Total</b>	<b>20</b>	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	5	3	2
Understanding	5	3	2
Journal work and timely submission.	10	8	4

### Checked by

**Name of Faculty** : Ichhanshu Jaiswal

**Signature** :

**Date** :



## Experiment No. 1

**Aim:** To study and implement Infrastructure as a Service using AWS

**Objective:** To demonstrate the steps to create and run virtual machines inside Amazon EC2

### Theory:

**IaaS:** Infrastructure as a service (IaaS) is a type of cloud computing service that offers essential compute, storage, and networking resources on demand, on a pay-as-you-go basis. IaaS is one of the four types of cloud services, along with software as a service (SaaS), platform as a service (PaaS), and serverless.

In an IaaS service model, a cloud provider hosts the infrastructure components that are traditionally present in an on-premises data center. This includes servers, storage and networking hardware, as well as the virtualization or hypervisor layer.

IaaS providers also supply a range of services to accompany those infrastructure components.

IaaS customers access resources and services through a wide area network (WAN), such as the internet, and can use the cloud provider's services to install the remaining elements of an application stack. For example, the user can log in to the IaaS platform to create virtual machines (VMs); install operating systems in each VM; deploy middleware, such as databases; create storage buckets for workloads and backups; and install the enterprise workload into that VM. Customers can then use the provider's services to track costs, monitor performance, balance network traffic, troubleshoot application issues and manage disaster recovery.

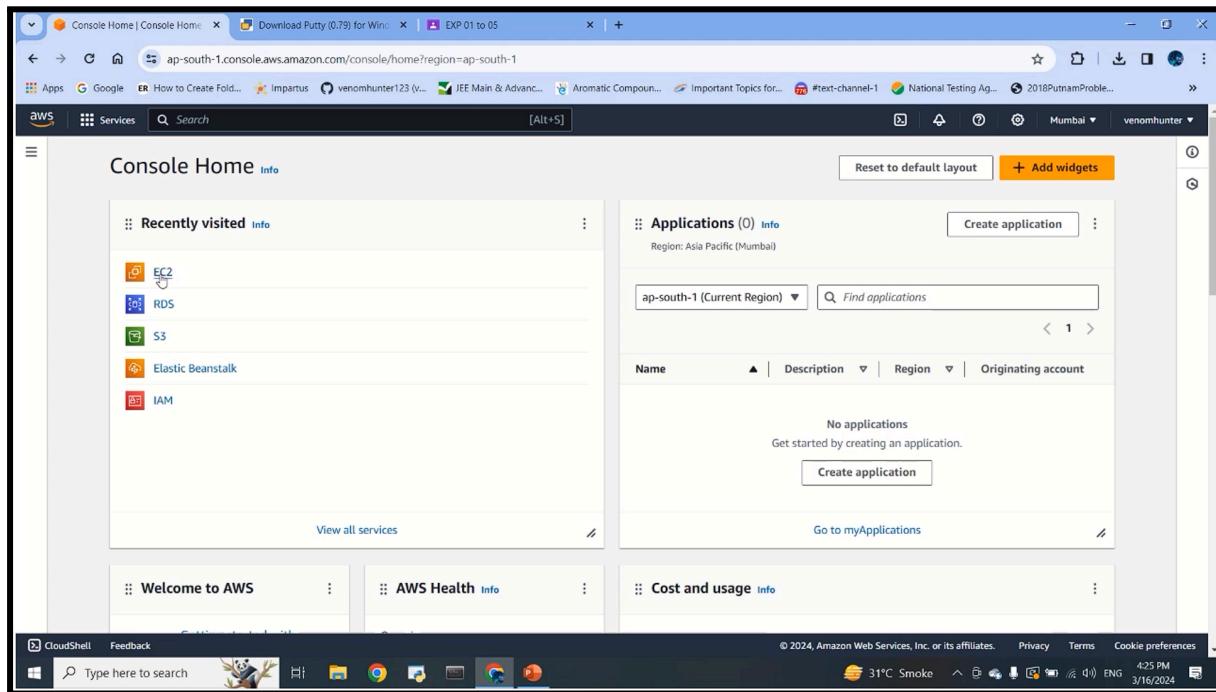
**AWS:** AWS enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application requires. This eases the migration process for existing applications while preserving options for building new solutions. You pay only for the compute power, storage, and other resources you use, with no long-term contracts or up-front commitments.



AWS is designed to allow application providers, ISVs, and vendors to quickly and securely host your applications – whether an existing application or a new SaaS-based application.

**EC2:** Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.

### Snapshots of implementation:





# Vidyavardhini's College of Engineering & Technology

## Department of Computer Science and Engineering (Data Science)

The screenshot shows the AWS EC2 'Launch an instance' page. The instance name is set to 'sairaj'. Under 'Software Image (AMI)', 'Amazon Linux 2023 AMI 2023.3.2...' is selected. The 'Virtual server type (instance type)' is set to 't2.micro'. A tooltip for the 'Free tier' indicates it includes 750 hours of t2.micro or t3.micro usage in the first year. The 'Launch instance' button is visible at the bottom right.

The screenshot shows the AWS EC2 'Launch an instance' page after the instance has been successfully launched. A green success message box displays the message: 'Successfully initiated launch of instance (i-097b6a79a8da51fd9)'. Below this, there is a 'Next Steps' section with several options: 'Create billing and free tier usage alerts', 'Connect to your instance', 'Connect an RDS database', and 'Create EBS snapshot policy'. Each option has a corresponding button and a 'Learn more' link.



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Science and Engineering (Data Science)

The screenshot shows the AWS Management Console interface for the EC2 service. On the left, the navigation pane includes options like EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. The main content area displays a list of instances with one selected: 'sairaj' (Instance ID: i-097b6a79a8da51fd9). The 'Details' tab is active, showing the instance ID, IPv6 address, Hostname type (IP name: ip-172-31-45-251.ap-south-1.compute.internal), and Answer private resource DNS name (IPv4 (A)). To the right, a 'PuTTY Configuration' window is open, showing the basic options for connecting to the instance. The host name is set to 'Ubuntu@35.154.83.16' and the port is '22'. The connection type is set to 'SSH'. The 'Status' tab in the EC2 interface shows that the instance is running. The 'Actions' dropdown menu is open, showing options like 'Stop', 'Terminate', and 'Launch instances'.

This screenshot shows the AWS Management Console with the EC2 Instances page open. The left sidebar has the same navigation as the previous screenshot. The main area lists instances including 't5.micro', 't3.micro', 't2.micro', and 't2.micro'. The 't2.micro' instance (Instance ID: i-097b6a79a8da51fd9) is selected, and its details show it is running with a Private IP DNS name of 'ip-172-31-45-251.ap-south-1.compute.internal'. A terminal session window titled 'ubuntu@ip-172-31-45-251:' is open, showing the user has run the command 'sudo apt install apache2'. The terminal output indicates the package is already installed. The status bar at the bottom of the terminal window shows the command was run at 4:29 PM on 3/16/2024.



VideoLink:  [Dhanashree\\_Thakur\\_56\\_Exp1.mp4](#)

**Conclusion:**

Comment on the advantages of EC2 for IaaS

**Ans:** EC2 (Elastic Compute Cloud) is a cornerstone service offered by Amazon Web Services (AWS) for Infrastructure as a Service (IaaS) solutions, and it brings several.

**Advantages:**

Scalability: EC2 allows users to scale computing resources up or down according to demand. This elasticity is crucial for handling varying workloads efficiently without over-provisioning or under-provisioning resources.

Flexibility: EC2 offers a wide range of instance types optimized for various use cases, such as compute-optimized, memory-optimized, storage-optimized, and GPU instances. This flexibility enables users to choose the right instance type for their specific workload requirements.

Cost-effectiveness: EC2 follows a pay-as-you-go pricing model, allowing users to pay only for the resources they consume. Additionally, AWS offers various pricing options, such as spot instances and reserved instances, to further optimize costs based on usage patterns and workload characteristics.

Global Availability: AWS maintains data centers in multiple regions worldwide. This global footprint enables users to deploy EC2 instances closer to their end-users, reducing latency and improving performance.

Integration with AWS Ecosystem: EC2 seamlessly integrates with other AWS services, such as Amazon S3 for storage, Amazon RDS for databases, and Amazon VPC for networking. This integration simplifies the deployment and management of complex applications and infrastructure architectures.



## Vidyavardhini's College of Engineering & Technology

### Department of Computer Science and Engineering (Data Science)

---

**Reliability and Availability:** AWS ensures high availability and reliability of EC2 instances through features like Auto Scaling, which automatically adjusts the number of instances based on workload demand, and Amazon EC2 Auto Recovery, which automatically recovers instances in case of failures.

**Security:** EC2 provides various security features, including security groups, network access control lists (ACLs), and encryption options. Users can implement security best practices to secure their EC2 instances and data.