



### 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.

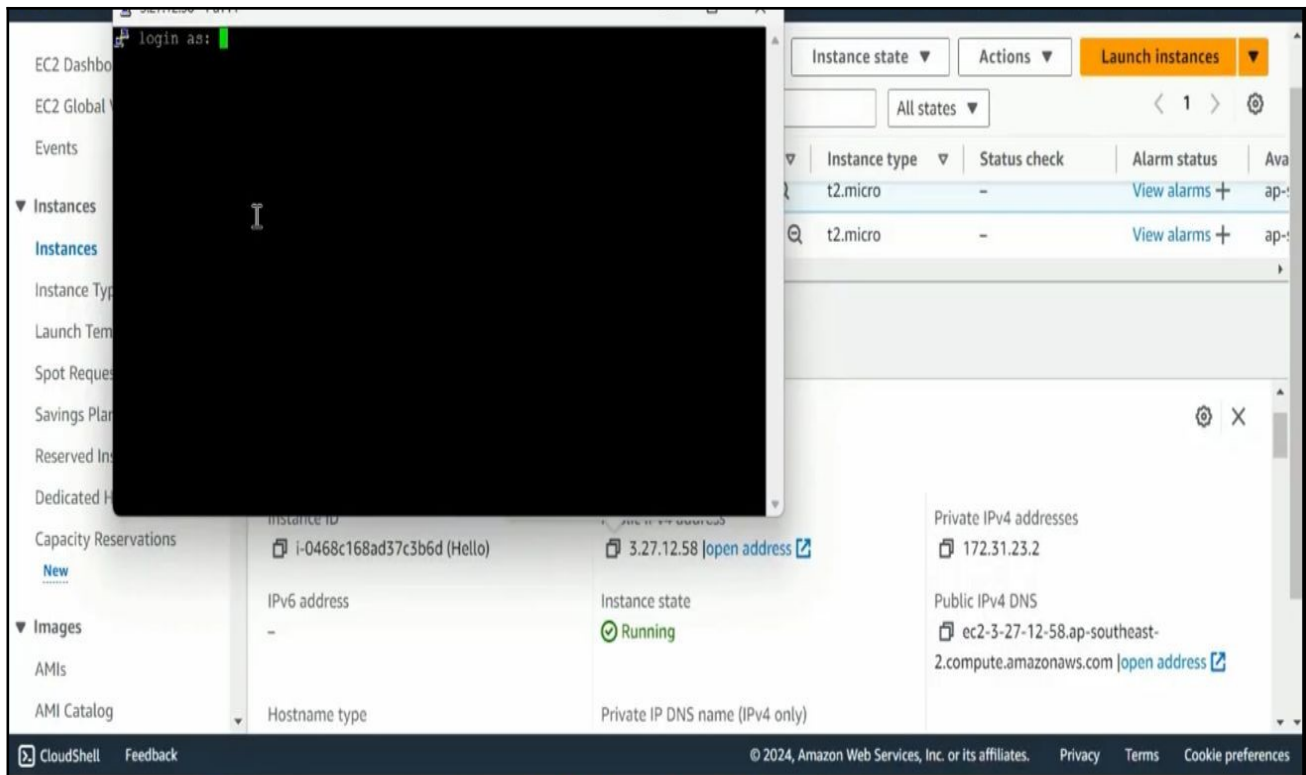
**EC2:** Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you 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.



### Output:

The screenshot displays the AWS Management Console's EC2 Dashboard. On the left, a navigation menu includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', and 'Images'. The 'Instances' section is active, showing a table with two instances: 'Hello' (ID: i-0468c168ad37c3b6d, state: Running) and 'Hello' (ID: i-028c9b6b0de97c4ad, state: Terminated). Below the table, the details for instance 'i-0468c168ad37c3b6d (Hello)' are shown, including its Instance ID, Public IPv4 address (3.27.12.58), Instance state (Running), and Hostname type. A 'PuTTY Configuration' dialog box is open in the foreground, showing the 'Basic options for your PuTTY session' tab. The 'Host Name (or IP address)' field is set to '3.27.12.58' and the 'Port' is '22'. The 'Connection type' is set to 'SSH'. The 'Close window on exit' option is set to 'Only on clean exit'.

This screenshot shows the AWS Management Console with the 'PuTTY Configuration' dialog box open. The 'Host Name (or IP address)' field is set to '3.27.12.58' and the 'Port' is '22'. The 'Connection type' is set to 'SSH'. The 'Close window on exit' option is set to 'Only on clean exit'. The 'About' and 'Help' buttons are visible at the bottom of the dialog. In the background, the EC2 instances table is visible, showing two instances: 'Hello' (ID: i-0468c168ad37c3b6d, state: Running) and 'Hello' (ID: i-028c9b6b0de97c4ad, state: Terminated). The details for instance 'i-0468c168ad37c3b6d (Hello)' are also visible, including its Instance ID, Public IPv4 address (3.27.12.58), Instance state (Running), and Hostname type.



### Conclusion:

#### Comment on the advantages of EC2 for IaaS:

- Scalability and Elasticity: Easily scale compute resources up or down to meet changing demands. You only pay for what you use.
- Flexibility: Wide range of instance types, operating systems, and software packages to choose from.
- Cost-effective: Pay-as-you-go model avoids upfront costs for hardware.
- Reliability: High availability with 99.99% uptime for regions.
- Security: Robust security features and integration with VPC for network isolation.