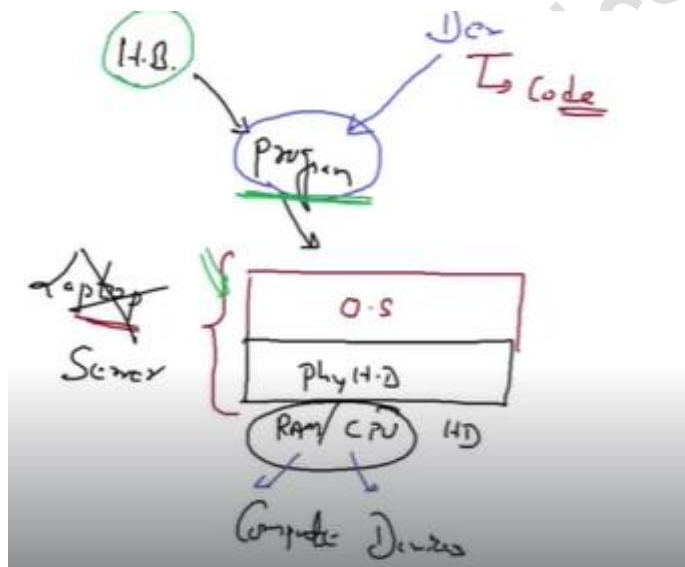




## AWS Training Session No.1

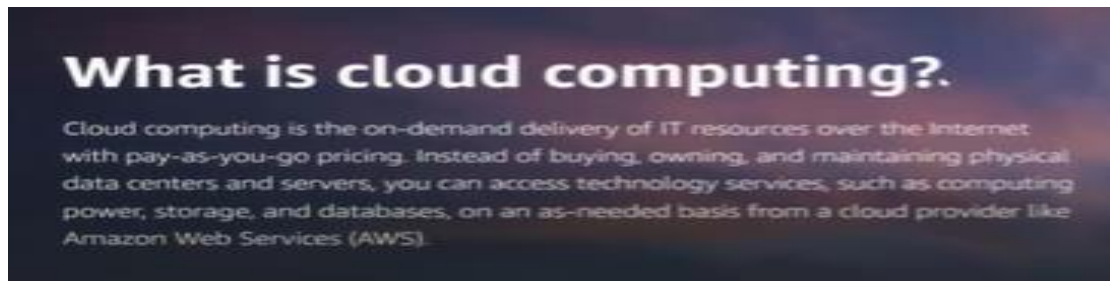
### Summary 21-02-2024

- If you want to run any program We need an **operating system** OS, only and only for running(executing)programs. Programs can be Notepad, Video Player, Web server, or your favorite game. There is no way to interact with the operating system without the program.
- If you have a program then it will generally require an **operating system** to run.
- In the real world we need hardware to maintain the OS or run any program

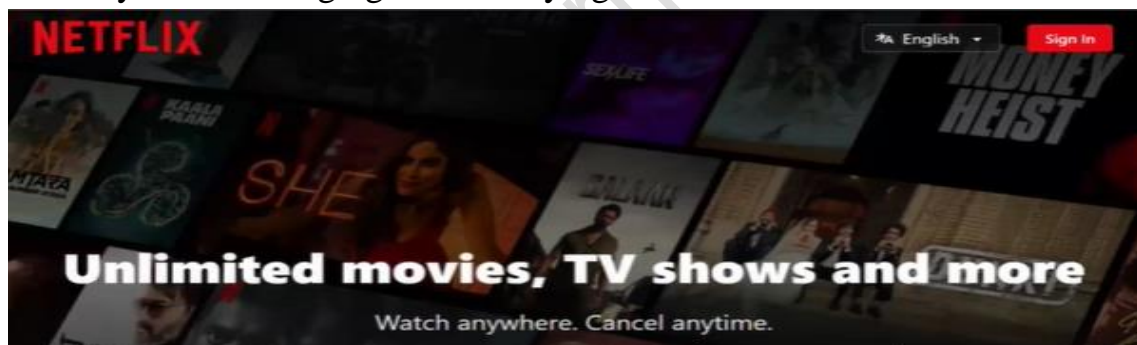


- **Serverless** is a cloud computing application development and execution model that enables developers to build and run application code without provisioning or managing servers or backend infrastructure.
- With the help of serverless we can manage all infrastructure they provide **scalability**
- So instead of investing in physical hardware or data centers, organizations can choose to **lease or outsource their computing resources** from some service providers, and that provider is known as **cloud service providers**.  
**Eg. AWS cloud, Azure, etc**
- **Outsource** means if you want hardware to run the program we take Hardware from outside the world

- Cloud computing allows businesses to access virtualized computing resources, such as virtual machines, storage, and networking, over the internet.
- Cloud computing gives operating systems and stores data from outside means we don't need real hardware



- For example, Netflix is a media company, and its core business is streaming movies and TV shows. It doesn't make sense for Netflix to own and manage its data centers and computing resources. That would be a distraction from its core business, and it would also be expensive. By outsourcing its computing resources to a third-party provider, Netflix can focus on what it does best such as content creation and customer experience, without having to worry about managing the underlying infrastructure.



- Many companies offer cloud computing services. Some of the most popular cloud service providers include:
  - **Amazon Web Services (AWS):** AWS is a cloud computing platform provided by Amazon.
  - **Microsoft Azure:** Azure is Microsoft's cloud computing platform.
  - **Google Cloud Platform (GCP):** GCP is Google's cloud computing. And many more.
- The "**pay as you go**" model in cloud computing refers to a pricing model where users pay for the cloud services they consume based on actual usage. Instead of making upfront investments in hardware and infrastructure, users are charged based on the resources they utilize, such as computing power, storage, network bandwidth, and other services.

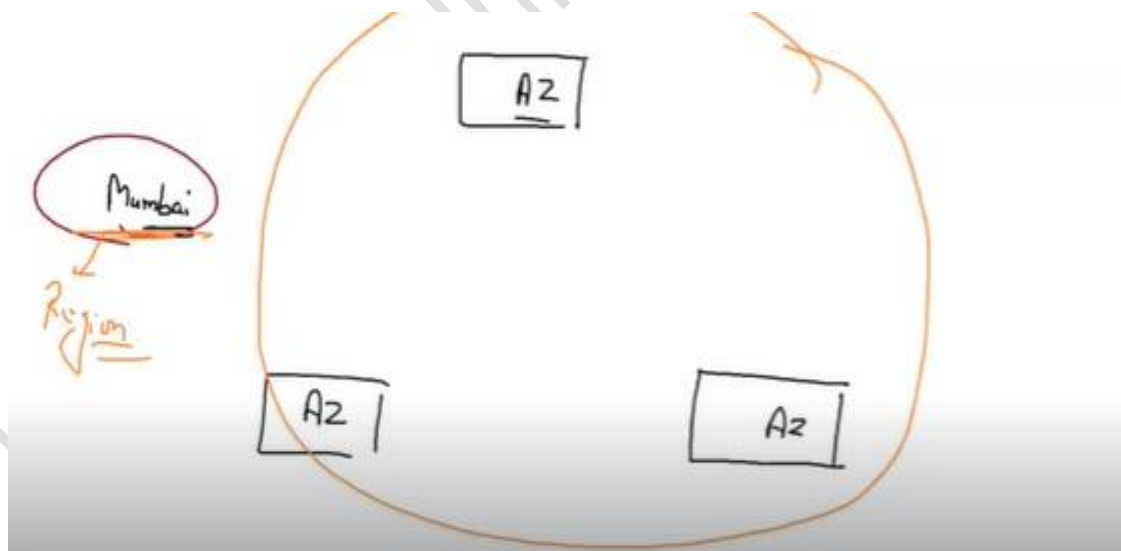
- In cloud computing, RAM and CPU are known as **computing devices or units**, while hard disks are known as **storage devices or units**.
- The term "**cloud**" in cloud computing is a metaphor for the Internet. It represents the fact that cloud computing services are delivered over the Internet, rather than being installed on a local computer. As a user, you don't need to know where the resources are physically located. You only need to know that they are available to you when you need them.
- One of the biggest reasons Why AWS is the top leader in the CC field is the "**Global network of AWS Regions**. AWS has the most extensive global cloud infrastructure. No other cloud provider offers as many Regions with multiple **Availability Zones** connected by **low latency**".
- **Low latency** means if you search for something on your website and your website gives the result after a long time then the customer does not come to your website AWS provides a Lot of **Availability Zones**
- when you go and search After that AWS goes and searches the result in the nearest **Availability Zones** then we solve the **Low latency**
- In **Availability Zones** they have a lot of hardware or compute units



- **Amazon Web Services (AWS)** is a cloud computing platform offered by Amazon. AWS provides a wide range of cloud services, offering over 200 different services and features across various categories.
- If you want to launch an operating system (OS) in AWS, the **Amazon Elastic Compute Cloud (EC2)** service is commonly used.
- An availability zone (AZ) is a physical location within a region. Each AZ consists of one or more discrete data centers. A region is a geographic area that contains multiple AZs.



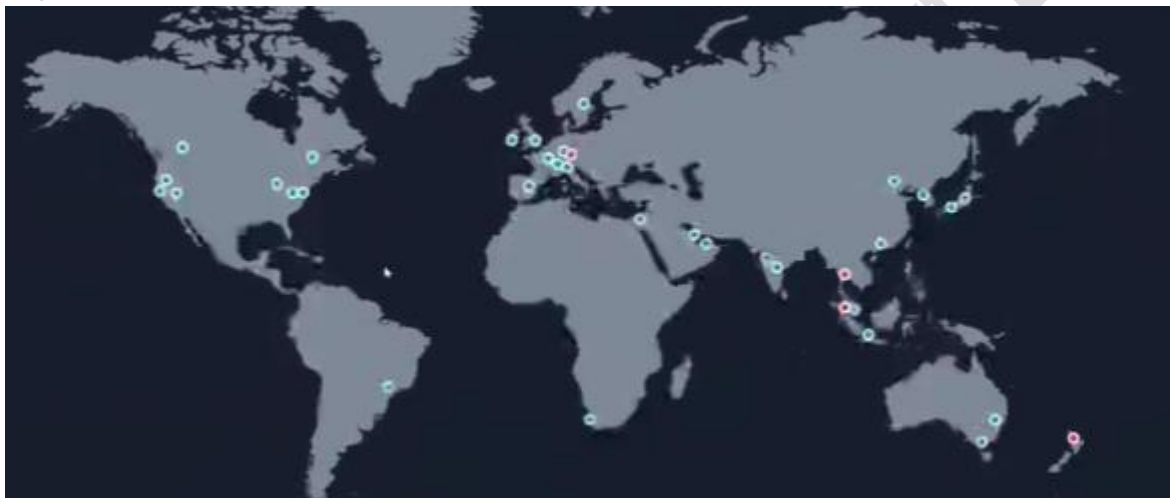
- AWS regions and availability zones are designed to provide high **availability and redundancy** for your applications and data. By launching your applications and data in multiple AZs, you can protect them from the failure of a single AZ.
- For example we launch the instance in the Mumbai region and you store the data but for some reason, your data gets deleted for that AWS launch same data center near the old data center
- If your website gets down then they go to the other AZ and give the result to the client
- Also AWS stores all data in different regions means your data is stored in different data centers



- In the Mumbai region we have three AZ

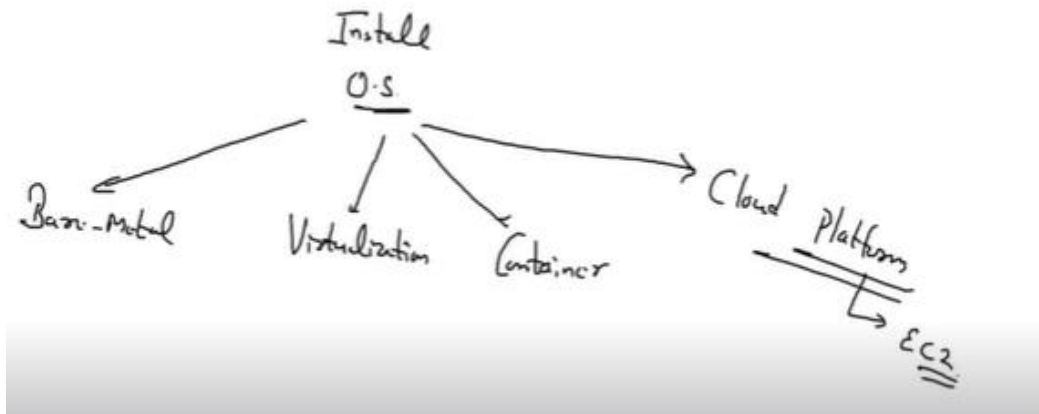
Zones	
Zone name	Zone ID
ap-south-1a	aps1-az1
ap-south-1b	aps1-az3
ap-south-1c	aps1-az2
<a href="#">Enable additional Zones</a>	

- Aws launched a total of 33 regions in the different countries and in every region we have different AZ



- There are three main ways to launch an operating system (OS):
  - **Bare metal** is the most traditional way to launch an OS. In this approach, the OS is installed directly on the physical hardware. But it also requires more hardware resources and can be more difficult to manage.
  - **Virtualization** allows you to run multiple OS on the same physical hardware. This is done by creating virtual machines (VMs), which are software-based representations of physical computers. VMs can be used to run different operating systems or to isolate different applications from each other.
  - **Containerization** is a newer technology that allows you to run applications in isolated containers. Containers are similar to VMs, but they are more lightweight and portable.
  - **Cloud Platform**

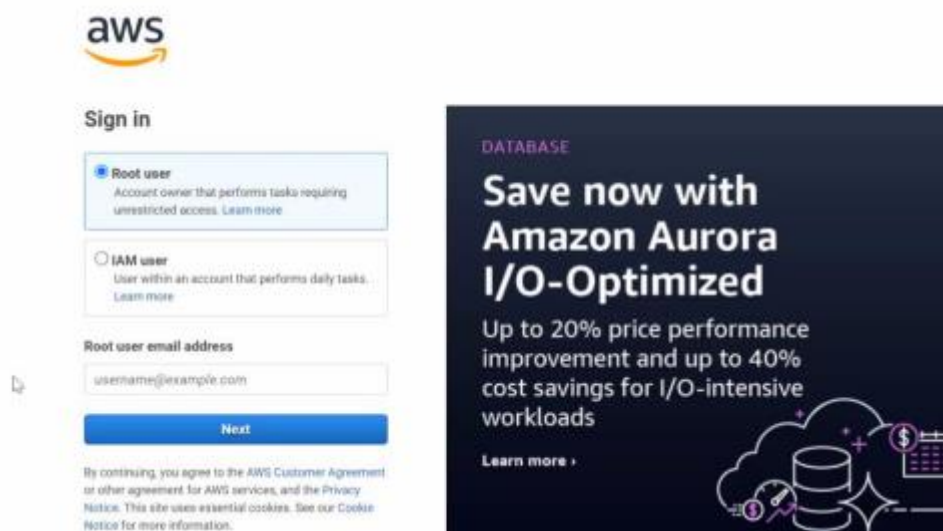




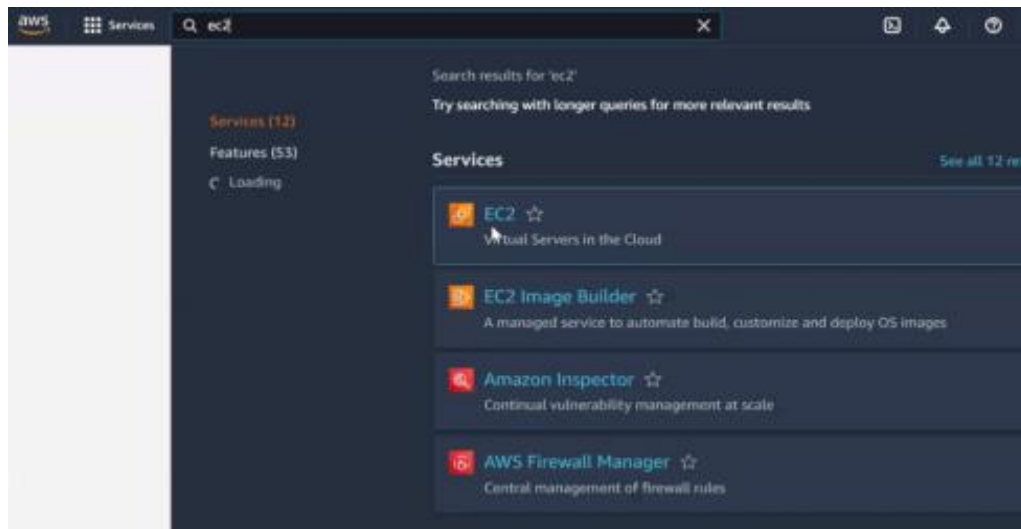
- AWS uses a variety of methods for launching OS, including bare metal, virtualization, and containerization. The specific method that is used depends on the specific service that you are using.
- For example, Amazon Elastic Compute Cloud (EC2) allows you to launch virtual machines (VMs) that can run a variety of operating systems.
- Amazon Elastic Container Service (ECS) allows you to launch containers, which are lightweight processes that share the same operating system kernel.
- In AWS, the boot or OS image is referred to as an Amazon Machine Image (AMI). An AMI is a pre-configured template that contains the necessary information to launch an EC2 instance with a specific operating system and additional software or configurations.

### To launch an EC2 instance in AWS:

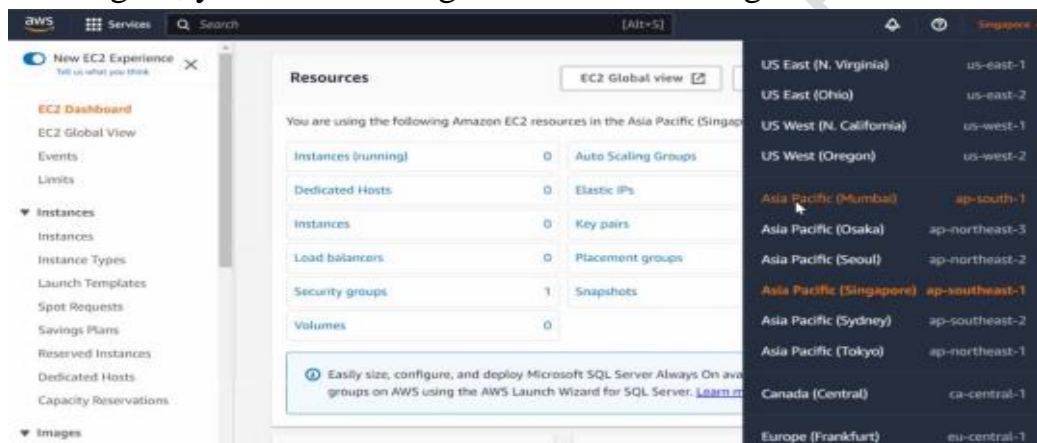
- Go to the AWS Management Console and sign in to your account



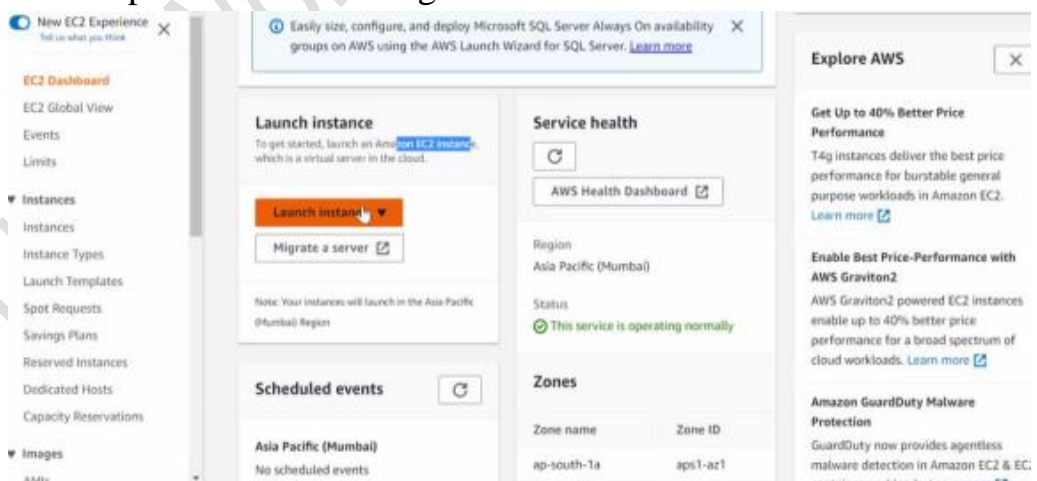
- Click the "EC2" service after searching for it.



- If you want to launch an instance in the Mumbai region but are currently in another region, you should first go to the Mumbai region to do so.



- On the EC2 Dashboard, click on the "Launch Instances" button. This will initiate the process of launching a new EC2 instance.



- Enter the name of the instance.

the simple steps below.

### Name and tags [Info](#)

Name

[Add additional tags](#)

- Choose the appropriate AMI based on your desired OS and requirements.

Recents

My AMIs

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI  
ami-049a62eb90480f276 (64-bit (x86)) / ami-0120894612ac9b862 (64-bit (ARM))  
Virtualization: hvm · ENA enabled: true · Root device type: ebs

Free tier eligible

Description

Amazon Linux 2023 AMI 2023.0.20230607.0 x86\_64 HVM kernel-6.1

Architecture

64-bit (x86)

AMI ID

ami-049a62eb90480f276

Verified provider

- Select the instance type that best suits your needs in terms of CPU, memory, and storage.

▼ Instance type [Info](#)

Instance type

t2.micro

Family: t2 · 1 vCPU · 1 GB Memory · Current generation: true

On-Demand Linux pricing: 0.0125 USD per Hour

On-Demand Windows pricing: 0.0125 USD per Hour

On-Demand RHEL pricing: 0.0724 USD per Hour

On-Demand SUSE pricing: 0.0124 USD per Hour

Free tier eligible

All generations

[Compare instance types](#)

- You need a key to access the instance. However, you can disable it right now and use another method to log in without the key that you will see in the upcoming process.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

[Create new key pair](#)

Proceed without a key pair (Not recommended) [Default value](#)

[Network Settings](#) [Info](#)

Edit

- In the Network Settings section, we will keep everything to default. We can see SSH traffic is allowed from Anywhere (any IP Address).



The screenshot shows the 'Network settings' step of the AWS 'Launch instances' wizard. It includes fields for VPC (vpc-05ae69498ffef9dc9), Subnet (No preference), and Auto-assign public IP (Enabled). Under the 'Firewall (security groups)' section, there are two radio buttons: 'Create security group' (selected) and 'Select existing security group'. Below this, it states that a new security group named 'launch-wizard-32' will be created with the following rules:

- ☒ Allow SSH traffic from: Anywhere (0.0.0.0/0)
- ☐ Allow HTTPS traffic from the internet
- ☐ Allow HTTP traffic from the internet

At the bottom, there are buttons for 'Cancel', 'Launch instance', and 'Review commands'.

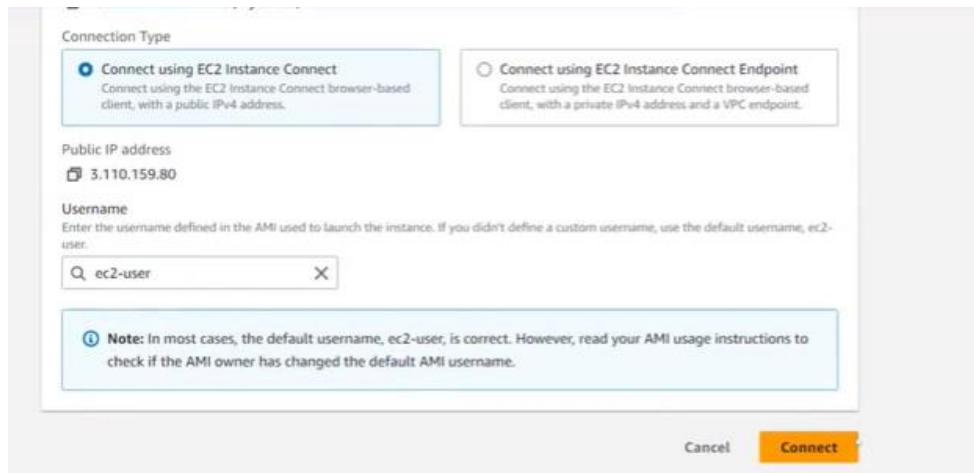
- Click on the "Launch Instances" button. AWS will start provisioning the EC2 instance based on your selected configuration.

- Connect to an AWS EC2 instance without a key: Select the instance that you want to connect to and click on the "Connect" button located at the top of the console.

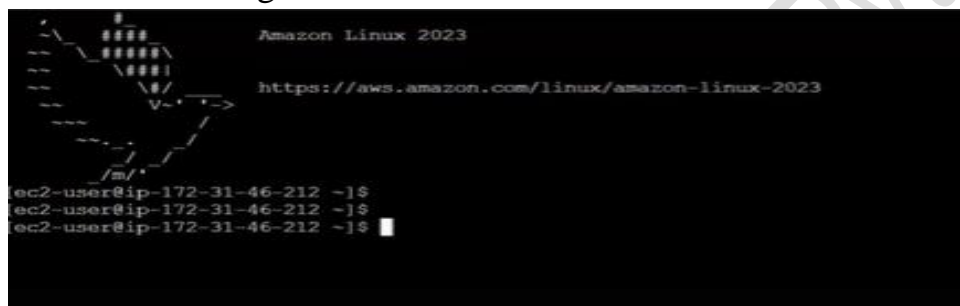
The screenshot shows the 'Instances (1/39)' page in the AWS Management Console. At the top, there are buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below these is a search bar and a filter dropdown set to 'Any state'. The main content is a table of EC2 instances.

	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>		i-00b02606e274074a3	Running	t2.micro	2/2 checks pas
<input type="checkbox"/>		i-0798c9c299566a1f1	Running	t2.micro	2/2 checks pas
<input type="checkbox"/>		i-08146b644b6eb5ea9	Running	t2.micro	2/2 checks pas
<input type="checkbox"/>		i-0456d5262e7262197	Running	t2.micro	2/2 checks pas
<input checked="" type="checkbox"/>	myos123	i-0c3e5c57edb56fc72	Running	t2.micro	2/2 checks pas
<input type="checkbox"/>	openstack cloud	i-09b5b92f5b00f642e	Running	t4g.xlarge	2/2 checks pas
<input type="checkbox"/>	Openstack Compute Node2	i-04a113b395ab67bd7	Running	t4g.xlarge	1/2 checks pas

- Click on "Connect".



- You will now be connected to the instance in a web browser. You can use the console to manage the instance.



- From the list of instances displayed on the EC2 Dashboard, select the instance you want to terminate or stop.
- With the instance selected, you have two options:
  1. To permanently remove the instance, click on the "Instance State" button, and then choose "Terminate". Confirm the termination when prompted.
  2. To temporarily stop the instance and preserve its data and configuration, click on the "Instance State" button, and then choose "Stop". Confirm the stop action when prompted.

