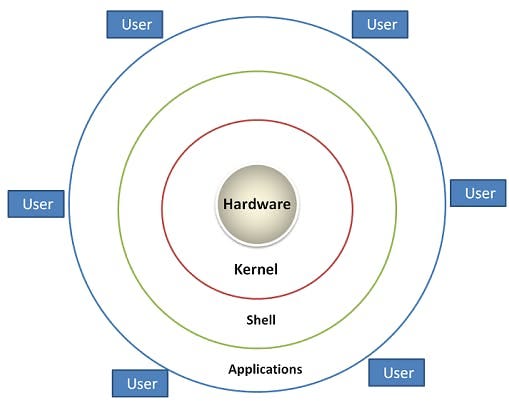
**Linux**

Linux is an open-source operating system. It is like Windows, Mac, Android,



1. Hardware

The lowest level of the Linux architecture is the hardware layer. This layer comprises the physical components of a computer, such as the hard drive, RAM, motherboard, CPU, network interfaces, and peripherals. These components are the tangible pieces of your system on which the rest of the architecture is built.

2. Kernel

Directly interfacing with the hardware layer is the kernel, the heart of the Linux operating system. As the core part of the OS, the kernel is responsible for low-level tasks such as disk management, task scheduling, memory management, and controlling peripherals.

3. Shell

One layer up from the kernel is the shell. In simplest terms, the shell is a user interface that allows users to interact with the kernel. In Linux, most interactions with the shell occur in a command-line interface (CLI), where users type commands interpreted by the shell. There are several different shells available in Linux, each with its unique features and syntax, such as the Bourne Again Shell (bash), the C Shell (csh), and the Z Shell (zsh).

4. Applications:

The topmost layer of the Linux architecture consists of applications. These are the software programs that you, as the user, interact with directly. They range from system applications like file managers, text editors, and network managers, to user applications like browsers.

**Distributions**

Linux comes in various distributions, which are different versions of the operating system packaged with specific collections of software and configurations. Some popular distributions include:

1. **Ubuntu**:
   * A user-friendly distro popular for desktops and servers.
2. **Fedora**:
   * Sponsored by Red Hat, known for its cutting-edge features and technologies.
3. **Debian**:
   * Known for its stability and vast repository of software.
4. **CentOS/AlmaLinux/Rocky Linux**:
   * Enterprise-focused distributions that are popular for servers.
5. **Arch Linux**:
   * A rolling-release distribution known for its simplicity and user control.
6. **Mint**:
   * Aimed at desktop users, based on Ubuntu, with a focus on ease o



**url**: https://medium.com/@ritusherke86/top-50-linux-commands-560b56343d00

**Storages**

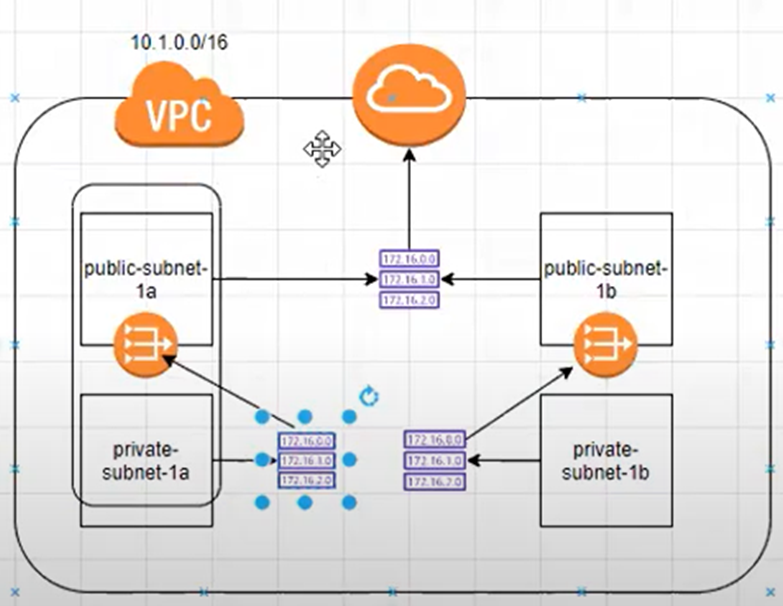
Linux storage refers to the various methods and technologies used to store, manage, and access data on a Linux operating system. Here's an overview of the key concepts and components related to storage in Linux:

**Types of Storage**

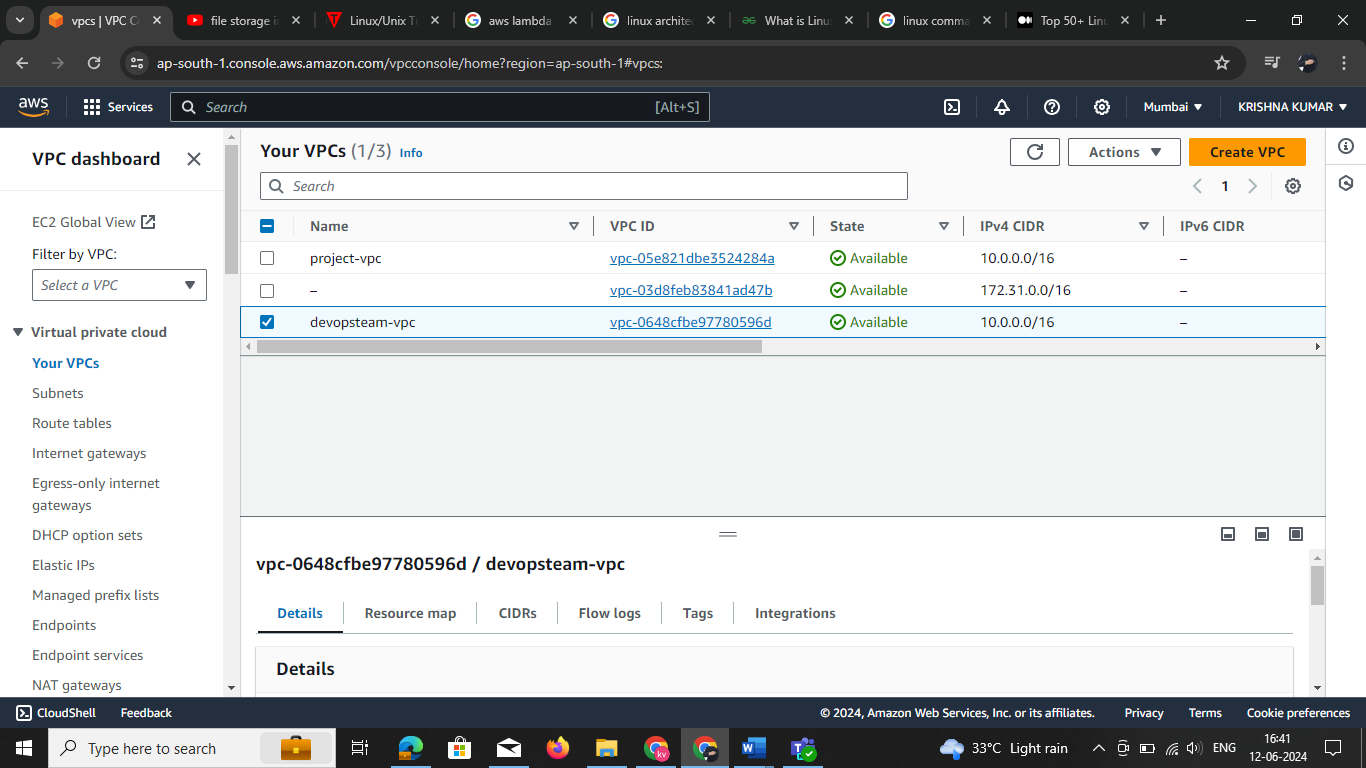
1. **Local Storage**:
   * **Hard Disk Drives (HDDs)**: Traditional spinning disk storage.
   * **Solid-State Drives (SSDs)**: Faster, more reliable storage using flash memory.
   * **NVMe Drives**: High-speed SSDs that use the Non-Volatile Memory Express (NVMe) protocol for faster data access.
2. **Network Storage**:
   * **NFS (Network File System)**: A distributed file system protocol that allows a user to access files over a network as if they were on the local storage.
   * **SMB/CIFS (Server Message Block/Common Internet File System)**: Protocols for network file sharing, commonly used in Windows environments but also supported in Linux.
   * **iSCSI (Internet Small Computer Systems Interface)**: A protocol that allows clients to send SCSI commands to storage devices on remote servers.
3. **Cloud Storage**:
   * Services like AWS S3, Google Cloud Storage, and Azure Blob Storage provide scalable and resilient storage accessible over the internet.
4. **Virtual Storage**:
   * **LVM (Logical Volume Manager)**: A device mapper framework that provides logical volume management for the Linux kernel, allowing for flexible disk management.
   * **RAID (Redundant Array of Independent Disks)**: A data storage virtualization technology that combines multiple physical disk drive components into one or more logical units for redundancy or performance improvement.

**Networking :**

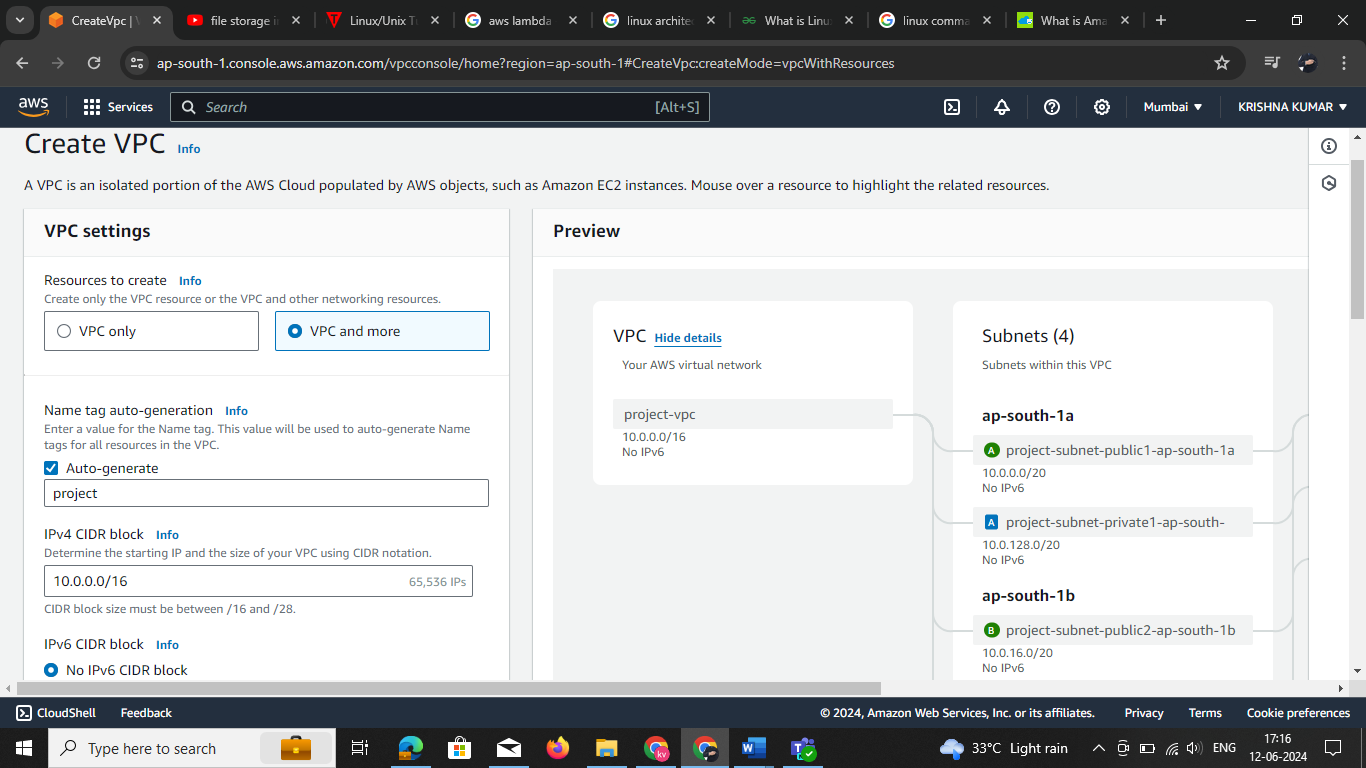
## Amazon VPC

Amazon VPC or Virtual Private Cloud is a service that allows us to create an isolated virtual network for our Amazon resources. A virtual network is a private network that is always hidden from the outside world, and you can perform certain operations that you don’t want to make public. Any user with their AWS account can host Amazon VPC. You can create, access, and manage Amazon VPC with the help of certain tools and services like the Amazon Web Service Management Console, Amazon CLI (Command Line Interface), Amazon SDK, and Query API.

Step1 : AWS console and click on create VPC



Setp2 : select any one VPC are VPC more option



https://www.davidc.net/sites/default/subnets/subnets.html

* **Route Table:**

In AWS Virtual Private Cloud, route Tables are the set of rules, that are used to determine where the network traffic has to be directed. The route table specifies the destination (IP address) and target (where do want to send the traffic to that destination). The target can be an Internet gateway, NAT gateway, Virtual private gateway, VPC peering connection, etc

* **Subnet:**

 It is a portion of the network that shares a common address component. All devices whose addresses have the same prefix are in the same subnet. For example, all those devices whose IP address would start with 172.31.1 would be part of the same subnet. There are two types of subnets. **Private Subnet**where resources are not exposed to the outside world and **Public Subnet**where resources are exposed to the internet through Internet Gateway.

* **Security Groups:**

Security groups are a set of firewall rules that controls the traffic for your instance. In Amazon Firewall the only action that can be carried out is allowed. You cannot create a rule to deny. The destination is always the instance on which the service security group is running. You can have a single security group associated with multiple instances.

* **NAT Gateway:**

Network Address Translation (NAT) Gateway is used when higher bandwidth and availability with lesser administrative effort is required. NAT gateway always resides inside the public subnet of an Availability Zone. It updates the routing table of the private subnet such that it sends the traffic to the NAT gateway. Elastic IP must be attached to the NAT gateway while creating. It supports only TCP, UDP, and ICMP protocols.

* **VPC Peering:**

A VPC peering connection allows you to route traffic between two Virtual Private Cloud’s using IPv4 or IPv6 private addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. A VPC peering connection helps you to facilitate the transfer of data

* **Network Access Control Lists (NACL):**

an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC. The default network ACL is configured to allow all traffic to flow in and out of the subnets to which it is associated.

* **Virtual Private Gateway:**

A virtual private gateway is the VPN concentrator on the Amazon side of the VPN connection. You create a virtual private gateway and attach it to the VPC from which you want to create the VPN connection.

* **Customer Gateway:**  An Amazon VPC VPN connection links your data center (or network) to your Amazon VPC (virtual private cloud). A customer gateway is an anchor on your side of that connection. It can be a physical or software appliance.
* **Elastic IP:**It is a static IP address that never changes and is a reserved public IP address that can be assigned to any Instance in a particular region. An elastic IP is reserved for your AWS account and is yours until you release it.
* **Network Interface:**Network Interface is a point of connection between a public and a private network. Every instance has a default network interface, called the primary network interface. Network traffic is automatically shifted to the new instance if you move it from one instance to the other.
* **VPC Endpoints:**VPC endpoints allow private connection between your AWS VPC and other AWS services without using the internet. VPC endpoint devices are scaled, redundant, and highly available VPC components. There are two types of AWS Virtual Private Cloud endpoints **Interface endpoints**and**Gateway Endpoints.**

## **What is Cloud Computing**

The term cloud refers to a network or the internet. It is a technology that uses remote servers on the internet to store, manage, and access data online rather than local drives. The data can be anything such as files, images, documents, audio, video, and more.

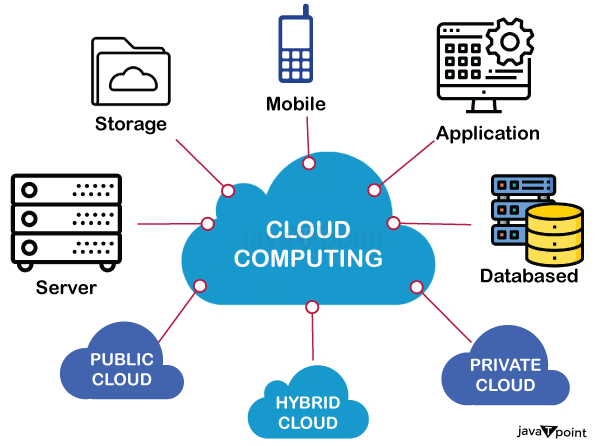
* Developing new applications and services
* Storage, back up, and recovery of data
* Hosting blogs and websites
* Delivery of software on demand
* Analysis of data
* Streaming videos and audios

## **Why Cloud Computing**

Small as well as large IT companies, follow the traditional methods to provide the IT infrastructure. That means **for any IT company, we need a Server Room that is the basic need of IT companies**.

In that server room, there should be a database server, mail server, networking, firewalls, routers, modem, switches, QPS (Query Per Second means how much queries or load will be handled by the server), configurable system, high net speed, and the maintenance engineers.

To establish such IT infrastructure, we need to spend lots of money. To overcome all these problems and to reduce the IT infrastructure cost, Cloud Computing comes into existence.



**1) Agility**

The cloud **works in a distributed computing environment**. It shares resources among users and works very fast.

**2) High availability and reliability**

The availability of servers is high and more reliable because the **chances of infrastructure failure are minimum**.

**3) High Scalability**

Cloud offers **"on-demand" provisioning of resources on a large scale**, without having engineers for peak loads.

**4) Multi-Sharing**

With the help of cloud computing, **multiple users and applications can work more efficiently** with cost reductions by sharing common infrastructure.

**5) Device and Location Independence**

Cloud computing enables the users to access systems using a web browser regardless of their location or what device they use e.g. PC, mobile phone, etc. **As infrastructure is off-site** (typically provided by a third-party) **and accessed via the Internet, users can connect from anywhere**.

**6) Maintenance**

Maintenance of cloud computing applications is easier, since they **do not need to be installed on each user's computer and can be accessed from different places**. So, it reduces the cost also.

**7) Low Cost**

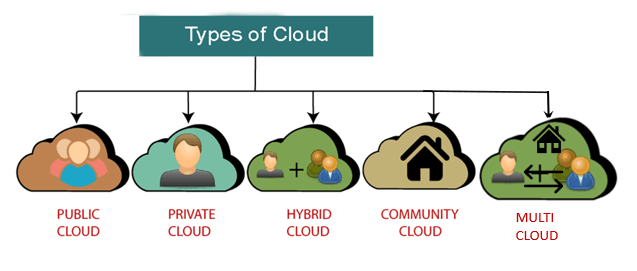
By using cloud computing, the cost will be reduced because to take the services of cloud computing, **IT company need not to set its own infrastructure** and pay-as-per usage of resources.

**8) Services in the pay-per-use mode**

Application Programming Interfaces**(APIs) are provided to the users so that they can access services on the cloud** by using these APIs **and pay the charges as per the usage of services**.

**Types of Cloud Computing**

There are the following 5 types of cloud that you can deploy according to the organization's needs-



## **Public Cloud**

* Public clouds are managed by third parties which provide cloud services over the internet to the public, these services are available as pay-as-you-go billing models.

**2. Private cloud**

* Private clouds are distributed systems that work on private infrastructure and provide the users with dynamic provisioning of computing resources. Instead of a pay-as-you-go model in private clouds, there could be other schemes that manage the usage

## **Hybrid cloud**

A hybrid cloud is a heterogeneous distributed system formed by combining facilities of the public cloud and private cloud. For this reason, they are also called **heterogeneous clouds.**

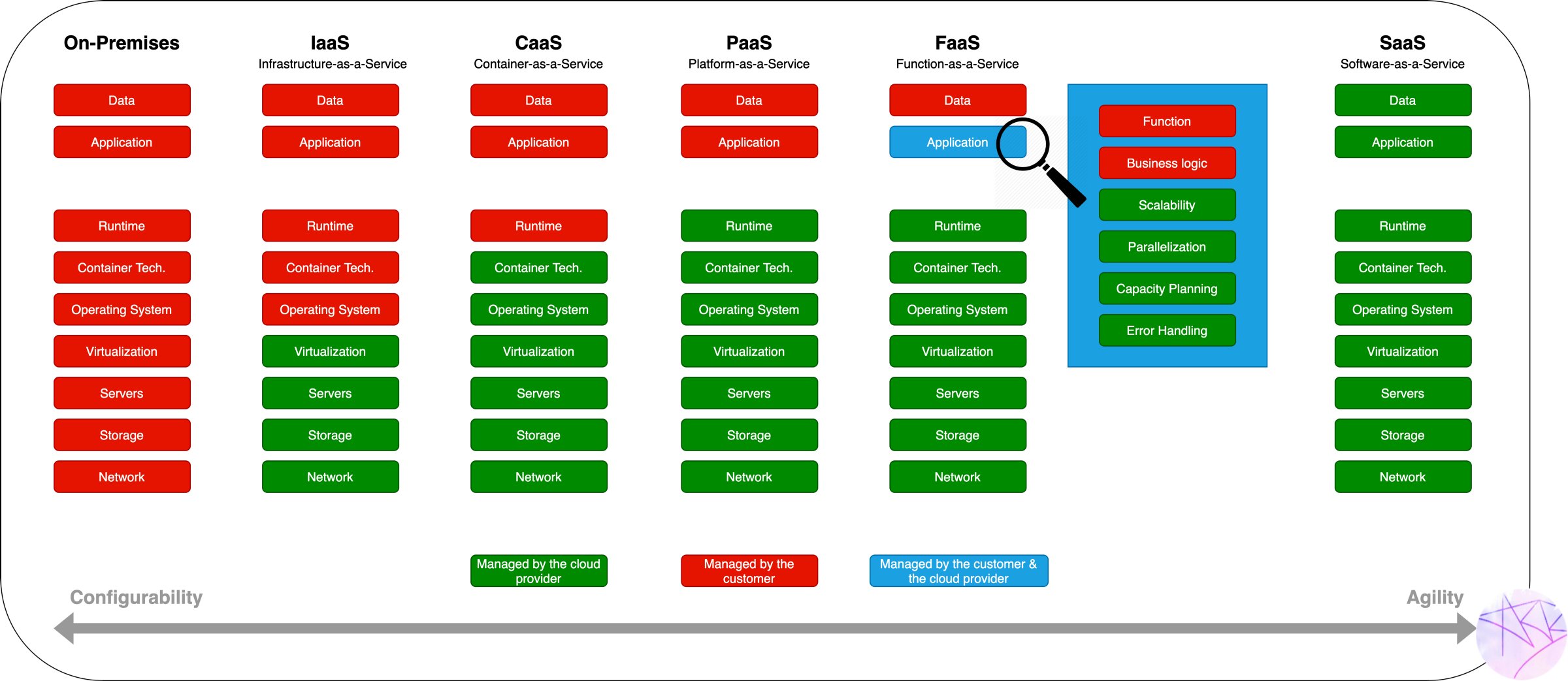
## **Community Cloud**

## Community clouds are distributed systems created by integrating the services of different clouds to address the specific needs of an industry, a community, or a business sector. But sharing responsibilities among the organizations is difficult.

**Multicloud**

Multicloud is the use of multiple cloud computing services from different providers, which allows organizations to use the best-suited services for their specific needs and avoid vendor lock-in.

**Cloud Computing models**

****

**Green Manage by the cloud**

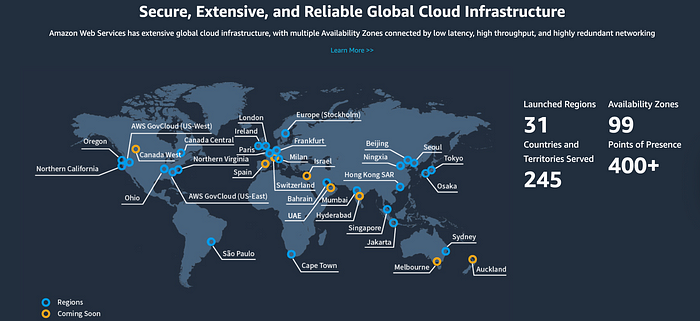
**Red managed by customer and**

**Blue Managed by both**

**what is aws**

Amazon Web Services (AWS) is the world’s most comprehensive and broadly adopted cloud, offering over 200 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster

**AWS Global Infrastructure**



**AWS Region**

1. **Definition**:
   * An AWS region is a physical location in the world where AWS has multiple, isolated, and physically separated data centers. Each region is a separate geographic area.
2. **Purpose**:
   * Regions are designed to allow users to deploy their applications and services closer to their end users to reduce latency and improve performance.
   * They also provide a way to ensure data sovereignty by allowing users to keep their data within a specific geographical boundary.

**Availability Zone (AZ)**

1. **Definition**:
   * An availability zone is one or more discrete data centers with redundant power, networking, and connectivity in an AWS region. Each region comprises multiple availability zones.
2. **Purpose**:
   * Availability zones are designed to be isolated from failures in other availability zones within the same region, providing fault tolerance and high availability.
   * They allow users to design and operate applications and databases that automatically failover between availability zones without interruption.

## **Edge Locations**?

AWS Edge Locations are physical sites dispersed across the globe, designed to deliver AWS services closer to the end-users. They are part of Amazon’s vast content delivery network (CDN) infrastructure, which plays a crucial role in reducing the distance that data has to travel to reach the user. By caching content in these Edge Locations, AWS ensures that data is delivered with the lowest possible latency and the highest possible speed.

## **How Do Edge Locations Work**?

When a user requests content (like a web page, image, or video), the request is routed to the nearest Edge Location. If the requested content is cached in this location, it is immediately served to the user. If the content is not available at the Edge Location, it is fetched from the origin server, which could be an Amazon S3 bucket, an EC2 instance, or any other custom origin. Once fetched, the content is cached at the Edge Location for future requests, improving the speed for subsequent accesses.

### Identity and Access Management (IAM)

Identity and Access Management (IAM) is a framework of policies and technologies for ensuring that the right individuals have access to the right resources at the right times for the right reasons. It involves a variety of components and practices that help organizations manage digital identities and control access to their resources.

#### **a) Users**

Users in IAM are the individual entities that need access to resources. These can be employees, customers, partners, or any other individuals who need to interact with the system. Each user has a unique identity and can be assigned various permissions based on their role within the organization.

Key points:

* **Unique Identity:** Each user has a unique identifier.
* **Credentials:** Users authenticate using credentials like passwords, multi-factor authentication (MFA), or biometric data.
* **Permissions:** Users are granted permissions that define what actions they can perform on resources.

Got to Aws IAM select user

#### **b) Groups**

Groups are collections of users. By organizing users into groups, it is easier to manage permissions and access control. Instead of assigning permissions to each user individually, permissions can be assigned to groups, and users inherit the permissions of the groups they belong to.

**Key points**:

* **Ease of Management:** Simplifies the management of permissions.
* **Inheritance:** Users inherit permissions from the groups they are part of.
* **Organization:** Helps in logically organizing users based on roles, departments, or other criteria.

#### **c) Roles**

Roles are sets of permissions that can be assigned to users or groups. Roles help in implementing the principle of least privilege by providing users only the permissions they need to perform their job functions.

Key points:

* **Permission Sets:** Define what actions are allowed on specific resources.
* **Assign to Users/Groups:** Roles can be assigned directly to users or groups.
* **Least Privilege:** Ensures users have the minimal level of access necessary.

#### **d) Policies**

Policies are documents that define permissions and are attached to users, groups, or roles to manage their access to resources. Policies can be of different types, such as identity-based policies, resource-based policies, and permissions boundaries.

Key points:

* **Definition of Permissions:** Clearly specify what actions are allowed or denied.
* **Attachment:** Can be attached to users, groups, or roles.
* **Types:** Include identity-based, resource-based, and others.

#### e) Password Policy

A password policy is a set of rules designed to enhance security by encouraging users to create strong, secure passwords and change them regularly. It defines the requirements for password complexity, length, expiration, and reuse.

Key points:

* **Complexity Requirements:** Specifies requirements like length, character types, and restrictions on commonly used passwords.
* **Expiration:** Determines how often passwords must be changed.
* **Reuse Restrictions:** Prevents users from reusing recent passwords.

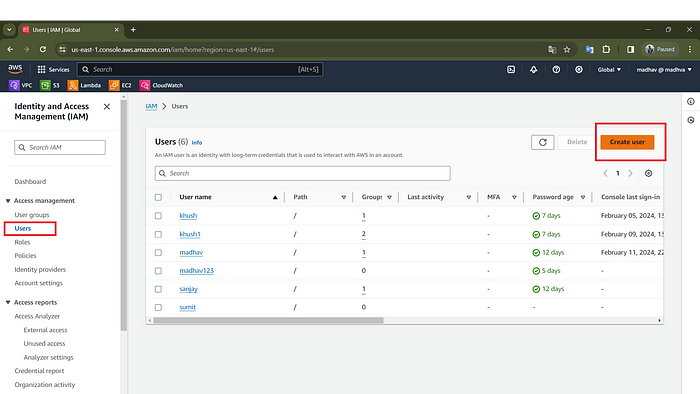
#### **f) Federation Overview**

Federation in IAM refers to the practice of linking a user’s identity across multiple identity management systems. This allows users to use a single set of credentials to access multiple systems and applications, enhancing security and user convenience.

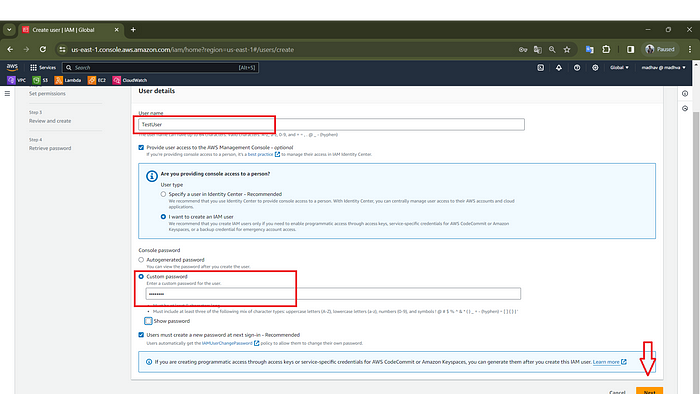
Key points:

* **Single Sign-On (SSO):** Allows users to log in once and gain access to multiple systems without re-entering credentials.
* **Trust Relationships:** Establishes trust between different identity providers and service providers.
* **Security and Convenience:** Improves security by reducing the need to manage multiple sets of credentials and enhances user experience by simplifying access.

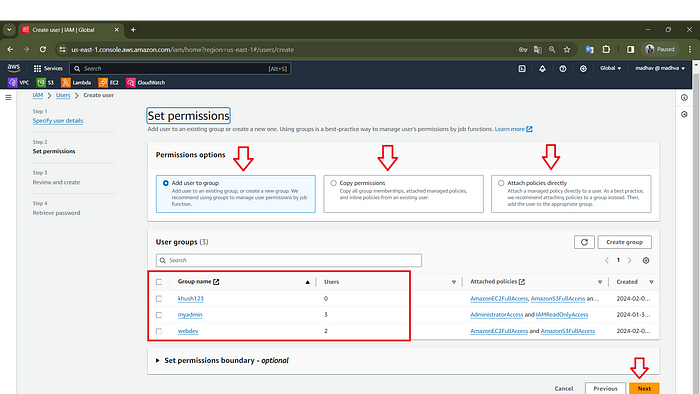
1. On the **Console Home** page, select the IAM service.
2. In the navigation pane, select **Users** and then select **Add Users**.



1. For the **User name**, enter **TestUser**. Names cannot contain spaces.
2. Select the check box next to **Provide user access to the AWS Management Console– optional** and then choose **I want to create an IAM user**.
3. Under **Console password**, select **Custom** **password**.
4. Clear the check box next to the **User must create a new password at the next sign-in (recommended)**. Because this IAM user is for emergency access, a trusted administrator retains the password and only provides it when needed.



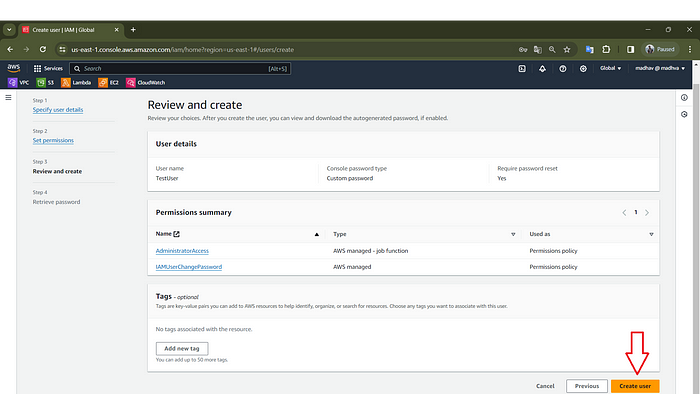
1. On the **Set permissions** page, under **Permissions options**, select **Add user to group**. so user Group you can add that particular user to the particular group and in one group have multiple users with the same permission



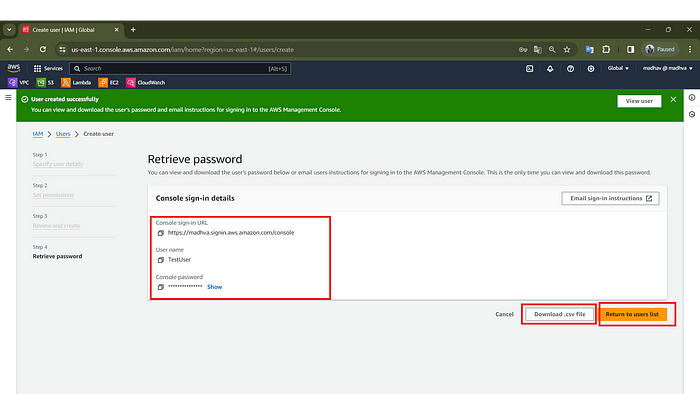
* But in my case, I am using **Attach Policies Directly.**
* under **Permissions Policies,**select the permission that you want to give to the user, Here I am using AdministratorAccess
* Select **Next** to proceed to the **Review and Create** page.



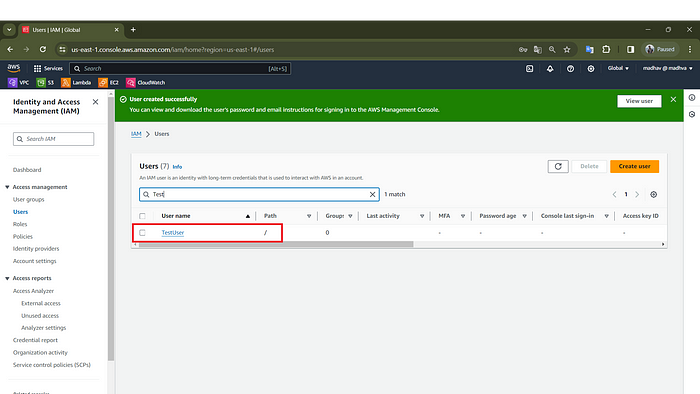
* On the **Review and Create** page, review the list of user group memberships to be added to the new user. When you are ready to proceed, select **Create User**.



* On the **Retrieve password** page, select **Download .csv file** to save a .csv file with the user credential information (Connection URL, user name, and password).
* Save this file to use if you need to sign-in to IAM and do not have access to your federated identity provider



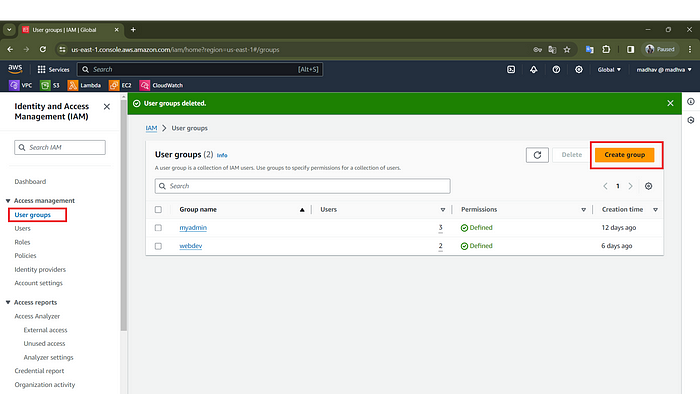
Here you can see the user that we created, With the name of **TestUser**



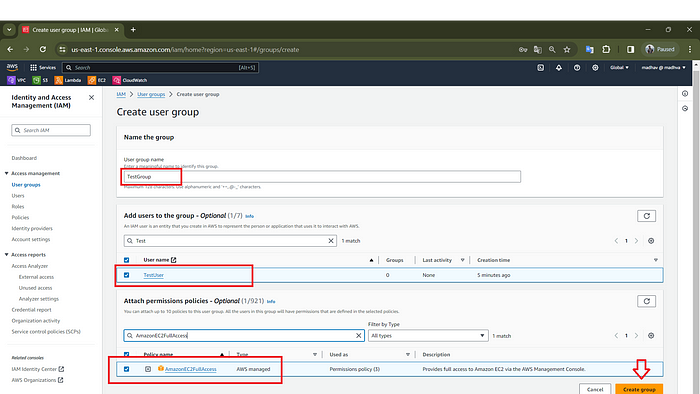
# Create your first IAM user Group:

Groups make it easy to manage permissions for lots of users at once. You can put users into groups and give the whole group certain permissions. It’s like giving everyone in a club the same access to club stuff.

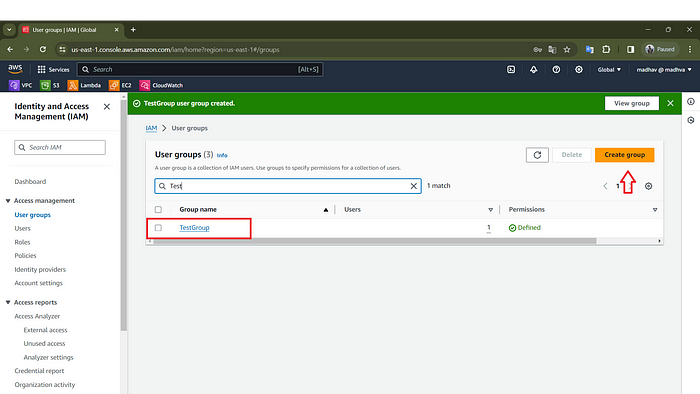
* In the navigation pane, choose **User Groups** and then choose **Create group**.



* For the **User group name**, type the name of the group.
* In the list of users, select the check box for each user that you want to add to the group, Like**TestUser**.
* In the list of policies, select the check box for each policy that you want to apply to all members of the group like **AdminstratorAccess**.
* Choose **Create group**.



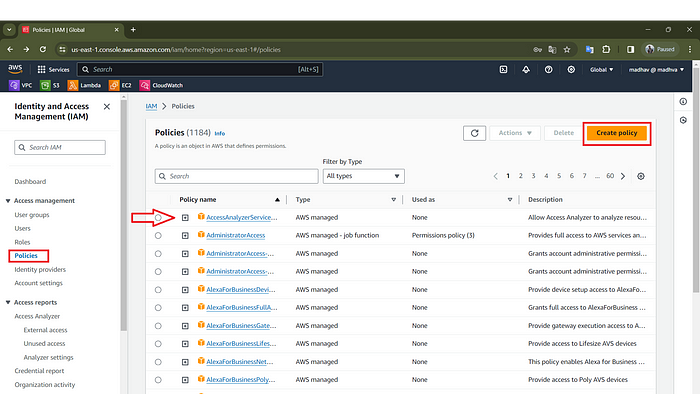
Here you can see the Group that we created, With the name of **TestGroup**



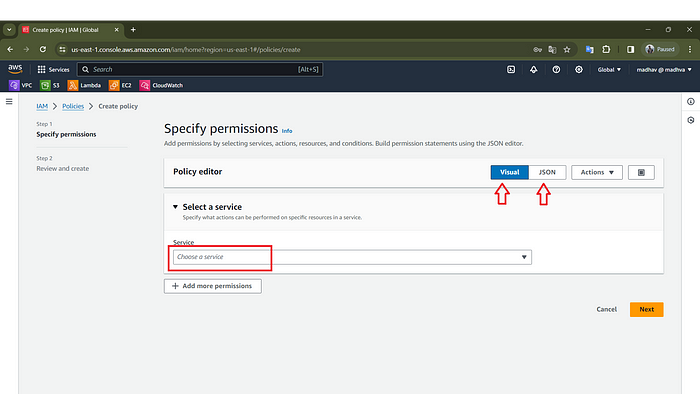
# create your first IAM policy

Policies are rules that say what users and groups can or can’t do in AWS. They’re written in a special way called JSON. Policies make sure everyone only does what they’re supposed to, which keeps everything safer.

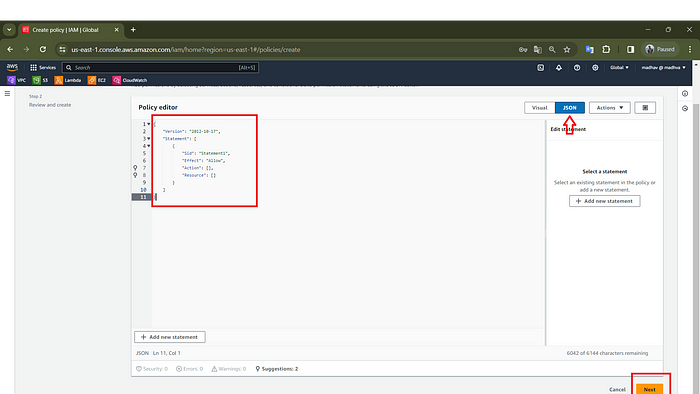
* In the navigation pane, choose **Policies**.
* If this is your first time choosing **Policies**, the **Welcome to Managed Policies** page appears. Choose **Get Started**.
* Choose **Create policy**.



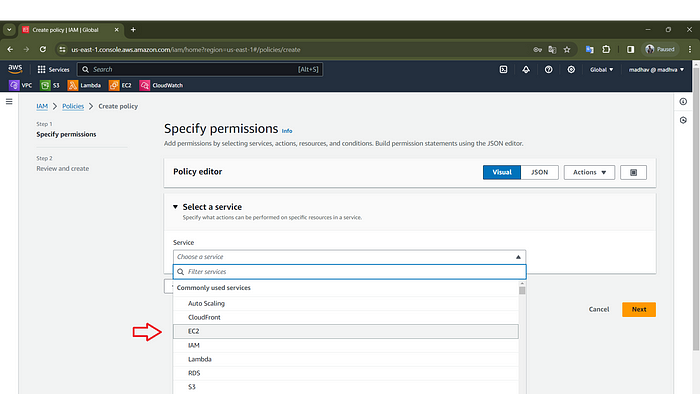
* On the **Create policy** page,
* There are two options **Visual** And **Json**
* In Visual, we can create policies manually, by using GUI



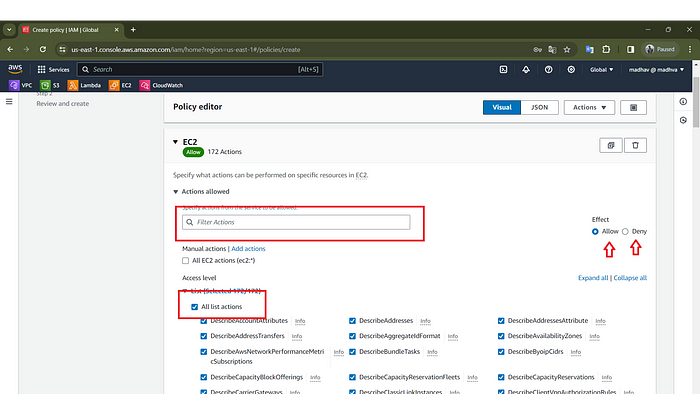
* In Json we use Json Language to create Policies, In my case I am using Visual for create policies,
* Choose **Next**.



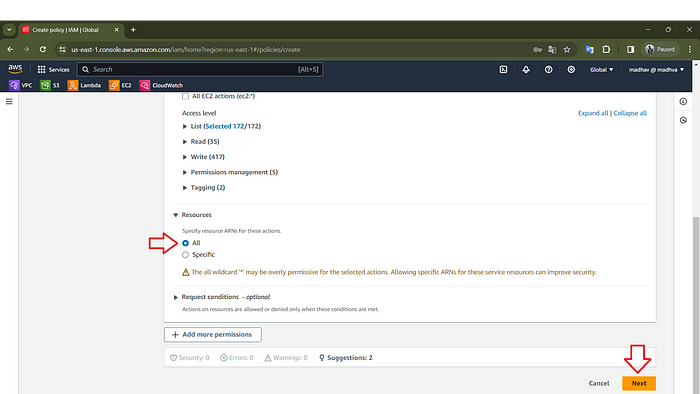
* In this **visual**, Under **Select a service**, We want to choose the service, for creating policies, Here I am using **EC2**
* Then Click, **Next**



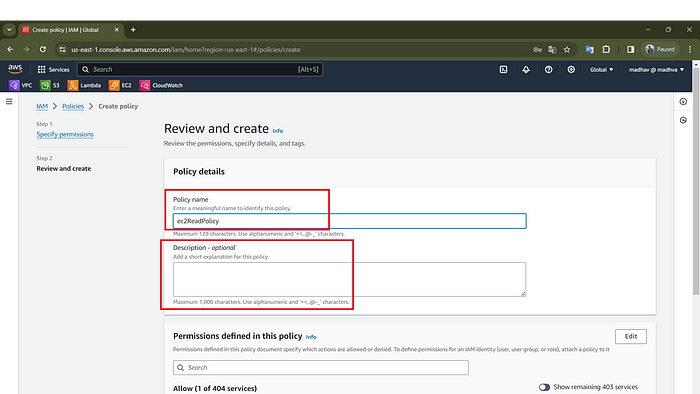
Here you can choose **effect**, **Allow** and **Deny** for service, now you can select the action that you want to allow or deny



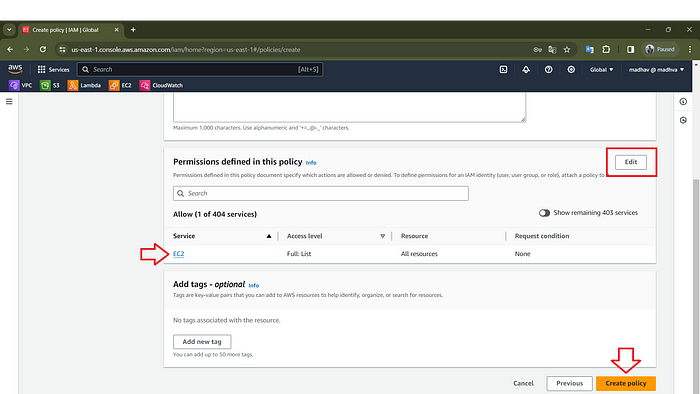
* In the **resource**, Select **All,**
* Then click **Next,**



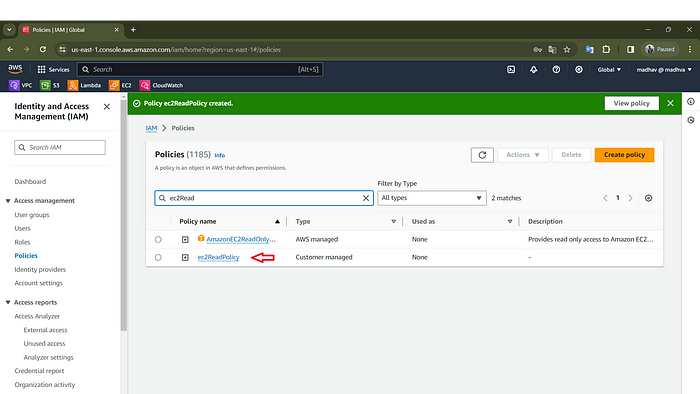
* On the **Review and Create** page, for the **Policy name**, type **ec2ReadPloicy**. For **Description**, You can describe the Policy type.



* Then choose **Create policy** to save the policy.



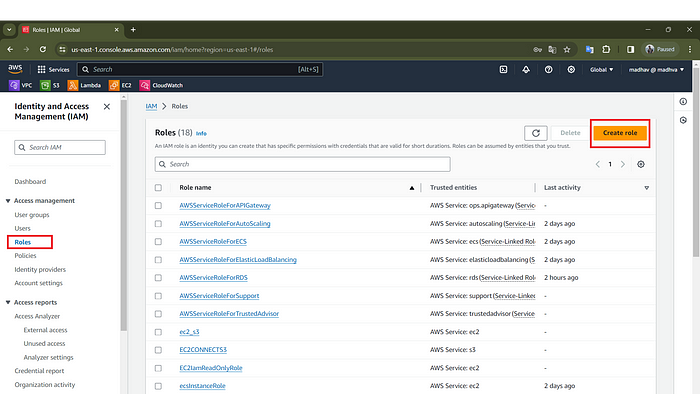
Here you can see the Policy that we created, With the name that you gave, Here You need to search for the Policy Because in AWS there Are many policy exist.



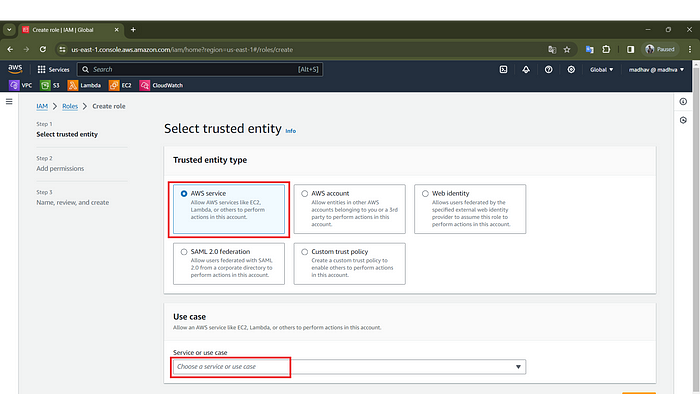
# Create your first role:

In AWS IAM, a role is a set of permissions that define what actions can be performed on AWS resources. Roles are assigned to entities like users, applications, or services, granting temporary access as needed. Roles help ensure security by allowing users or services to access resources only when necessary, without needing permanent credentials.

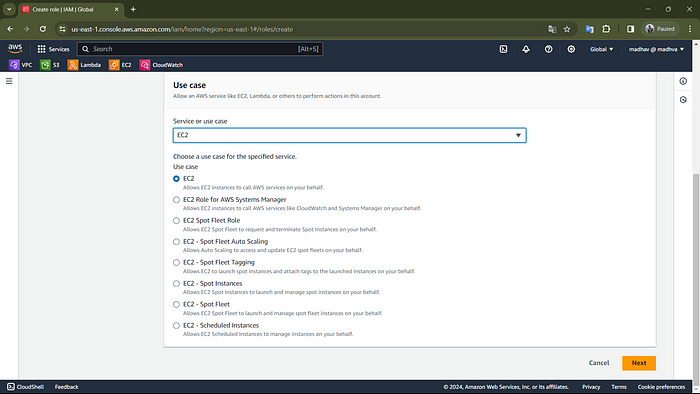
* In the navigation pane of the IAM console, choose **Roles** and then choose **Create role**.



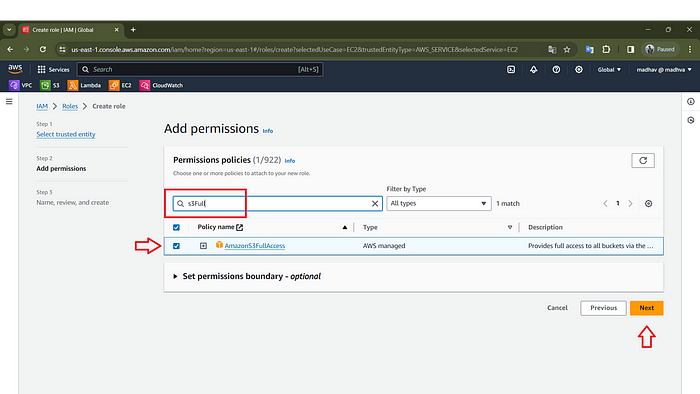
* Choose **AWS account** role type and select **AWS Services**.



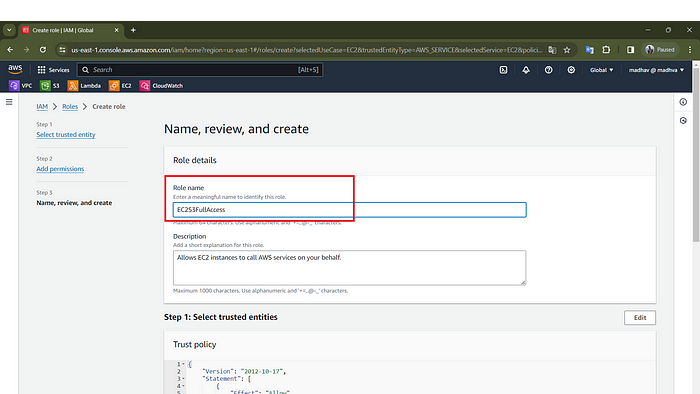
* Under **Use case**, Select the service or Use case, Here I select**EC2**as the service.
* Then Click **Next**



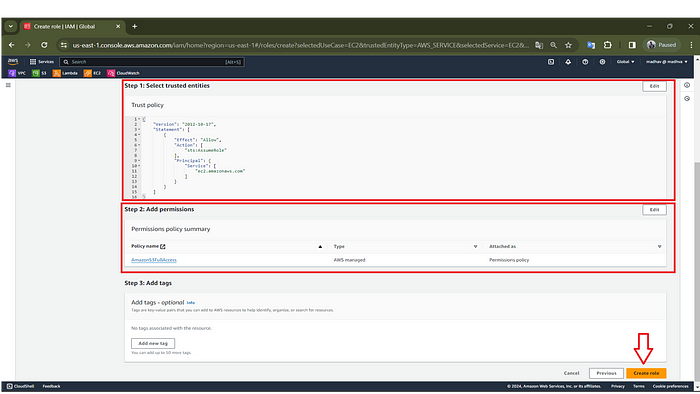
* In **Add Permissions**, select the check box next to the permissions policy to apply. For this Article, we are going to select the **AmazonS3FullAccess** policy.
* Then Click, **Next**



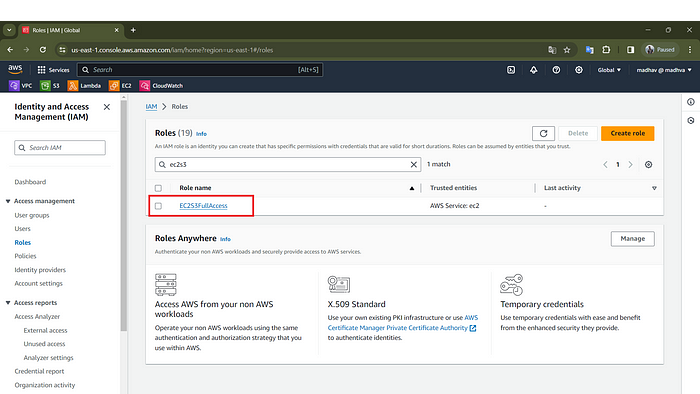
* On the **Name** **Review and Create** page, for the **Role** **name**, type **EC2S3FullAccess, Role name**, enter a name that identifies this role For **Description**, You can describe the Role type.



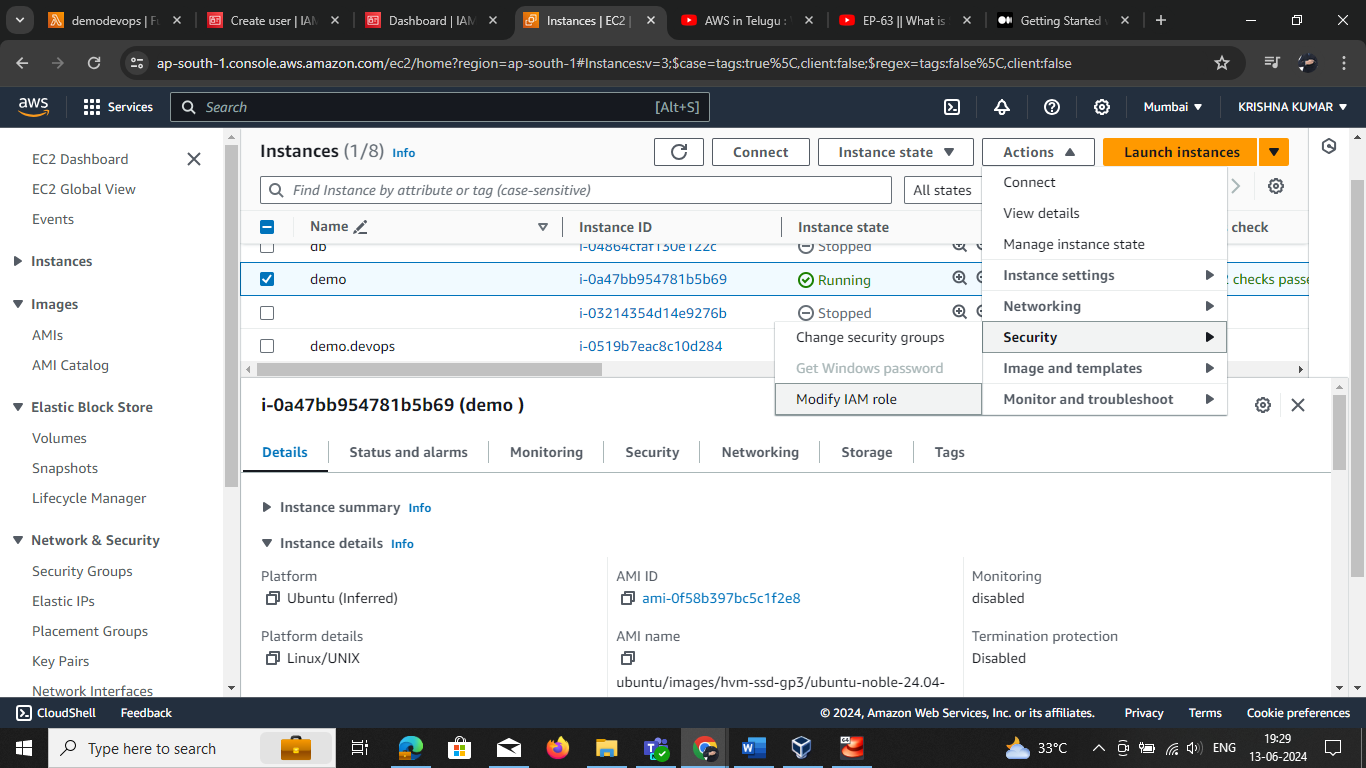
Select **Create role**.



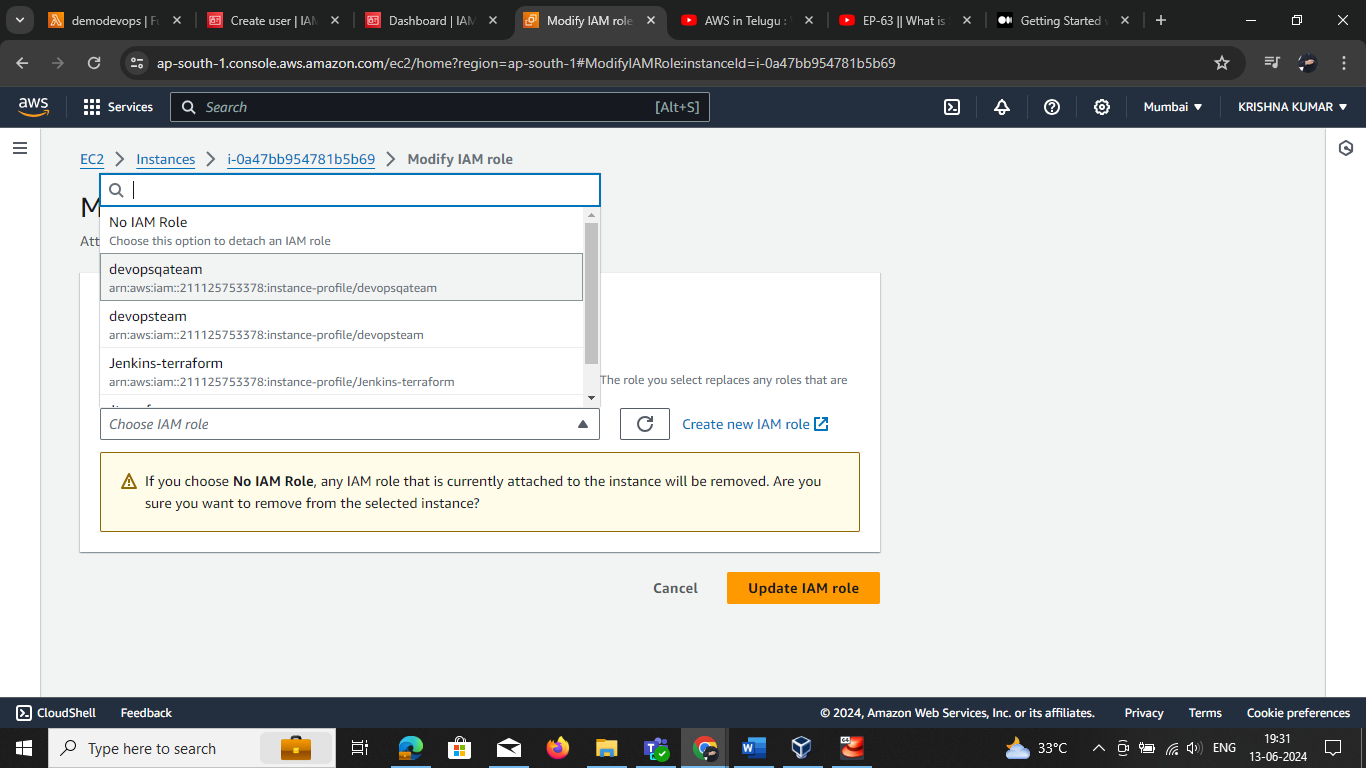
Here you can see the Role that we created, With the name that you gave, Here You need to search for the Role.



Attach the roles in ec2 instance



Select the IAM role and click update it



We can use IAM roles in different services

**Federation overview**