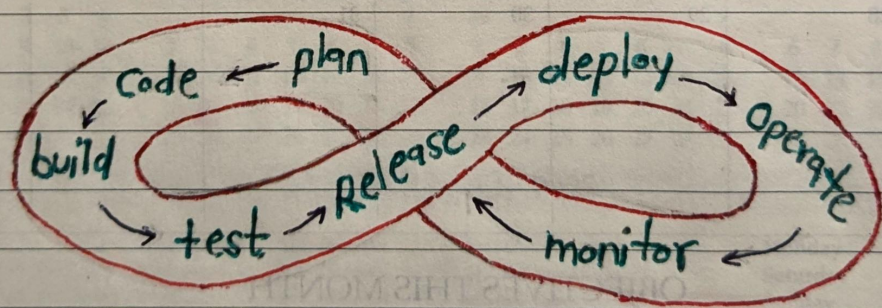
DevOps- DevOps is a combination of development and Operation. It means working together to make software faster and better.

It joins developers (who write code) and operations (who manage system) to help build, test and deliver apps/site quickly with fewer problems.

It's like teamwork and automation to make software development smooth.

Benifits of devOps- Faster delivery of software
Improved collaboration between teams.
Higher reliability and fewer software failures.
Quicker recovery from errors.



Server - A server is a computer that gives information or services to other computers.

For ex-, it can store websites, files or emails and share them when needed.

Web server – A web server is a software and hardware system that stores, processes and serves web content (HTML page, images, videos etc) to clients via the Internet.

Web servers are essential components of the WWW that handle requests from clients and send back the requested resources.

Features of web server –

1.) Storing and serving web content
2.) Handling client requests
3.) Running applications
4.) Managing security
5.) File upload and download

Some web servers are – Apache, NGINX, IIS, Litespeed, Tomcat etc.

How to web server works –

Client request → DNS resolution → server response → sending the response → Rendering in the browser.

Advantages of using web server -

Access to websites - It allows user to access websites from anywhere over the internet.

Efficiency - Web servers handle multiple requests at once, making in easy for many users to view the websites at the same time.

Storage - It stores all the files (like images, videos, and web pages) needed to display web-sites.

Security - Web servers can provide security features like encryption (HTTPS) to keep the data safe.

3-tier Architecture – 3-tier Architecture is a software design pattern that divides an application / sites into three logical layers, making it more scalable, secure and easier to maintain.

Presentation layer → Application layer → Database layer

Presentation layer – (Frontend)
This is the topmost part of the application. It directly interacts with the user.
It jobs is to take user input and show output.
Example Mobile app, web browser, desktop GUI

Application layer – (Backend)
This layer processes data and implements logic.
It understands the data coming from the presentation layer and connects it to the database layer.
Example API, code written on the server such as python...

Database layer – This layer stores and manages all the data of the application. It interacts with the application layer.

Example - Database such as MySQL, MongoDB, PostgreSQL

Working method -

User makes a request through the presentation layer.

The application layer processes the request and contacts the data layer for the required data.

The data layer sends the data back to the application layer.

The application layer processes the data and gives it to the presentation layer, and displays the result to the user.

Advantages -

Easier to update - Changes in one layer (like the user interface) don't affect the others.

Better performance - Each layer can work faster and independently.

Secure - Sensitive data can be protected in the database layer, away from the user.

Flexible- You can use different technologies for each layer.

Teamwork- Different teams can work on different layers at the same time.

Imp

Difference between devOps and Agile

DevOps and agile are two different but complementary methodologies used in software development, focusing on different aspects of the development lifecycle.

Agile- Agile is a way of working on software proyects where teams work in small steps, getting feedback quickly and making changes as needed. It's all about flexibility and delivering working software often.

In short:- (Approach)

Agile is for Improving how teams develop software.

DevOps is for improving how teams deliver and maintain software.

IIS Server - IIS is a web server created by Microsoft. It's main purpose is to host and run websites or web application.

With IIS, we can turn our computer into a 'server' to deliver website, web API's or other content over the internet or a local network.

Advantages -

1.) Host websites
2.) Security
3.) Request filtering
4.) Application pools.
5.) Support ASP.NET
6.) Logging and Diagnostics
7.) FTP Server

WSL - (Windows Subsystem for Linux) is a compatibility layer that allows user to run a 'Linux' directly on windows, without the need for a virtual machine or dual-booting.

It enables running Linux command-line tools, utilities and applications alongside Windows application.

WSL allowing seamless interaction between the two operating systems.

Advantages -

1.) Run Linux on windows
2.) Seamless integration
3.) GUI Support
4.) Multiple Distributions

Operating System - An operating system is the collection of software that helps to our computer / phone run and work smoothly.

It manages all the hardware and software on the device.

Features -

Managing Program - It helps run programs and apps on our device.

Memory Management - It decides how to use our device's memory (RAM) for different tasks.

Managing files - It organizes and keeps track of our files and folders.

Device Control - It helps to use devices - keyboard, mouse or printer.

Security - It keeps our device safe from harmful software and protects our data.

Examples - Windows, macOS, Linux, Android, iOS

Linux - Linux is 'kernel' not O.S.
Linux is not a UNIX derivative. It was written from scratch.
A Linux distribution is the Linux kernel and the collection of software that together, creat a 'Operating system'.

It was developed by "Linus Torvalds" in Sept-1991. That time, he was a student at the University of Helsinki, Finland, USA.

Kernel - Kernel is the central part (heart) of the system that controls everything, from hardware to software.
It manages the CPU, memory and I/O devices and allows programs to interact with the computer's hardware.

Sunday 12

Functions of Linux -

1) Process Management
2) Memory management
3) Device Management
4) Security
5) I/O management
6) Multitasking

Size and Complexity between kernal v/s OS

Kernel - The size of kernel is small because it handles only those things that connects the hardware and software of the computer. (A few MB's to 100+ MB)

Kernel is very complex because it works directly with the hardware.

Operating System - The size of OS is very large because it includes many other part along with the kernel. (100 MBs to GBs)

OS is more complex because it deals with not only the hardware but also the user

Ubuntu - Ubuntu is a Linux-based operating system, built around Linux kernel + system utilities + software + desktop environment. (Debian Linux)

Some basic Linux commands

whoami – The whoami command is used to display the currently logged-in user on the system. It simply returns the username that we are logged into the shell as.

Example

```
$ whoami
shaurya devops
```

This command is useful when we're working in a multi-user environment or if we're unsure which user account is currently active.

pwd – The pwd command displays the absolute path of the current directory we're in within the file system.

```
$ pwd
/home/shaurya
```

The command shows the directory path we're currently located in. This helps we track our position in the file system.

The output is usually an absolute path, meaning it starts from the root (/).

ls – The ls command lists the contents (file and directories) of a directory. By default, it lists the contents of the current directory.

Example-

```
$ ls
documents  downloads
```

This command shows the file and directories within the current directory. We can also use flags to modify its behavior.

- **-l** for a detailed listing (permissions, ownership, size, etc.)
- **-a** to show hidden files (those that start with a dot.)
- **-h** to display file sizes in human-readable format (eg KB, MB)
- **-R** for recursive listing of all directory and subdirectories.

cd - The cd command is used to navigate between directories in the file system.

Example-

```
$ cd /home/shaurya
$ pwd
/home/shaurya
```

→ (directory-path)

To move into a directory, specify the full or relative path of the directory we want to go to.

To go up one directory level (parent directory), use- cd ..

To move to the root directory (/), use- cd /

mkdir - The mkdir command is used to create a new directory in the filesystem.

Example- create a single directory:

```
mkdir my_directory
```

create a multiple directories at once:

```
mkdir dir1 dir2 dir3
```

touch – The touch command is used to create a new file or update timestamps.

usage- `touch file_name`

nano – nano is a simple, command-line text editor used to edit files in the terminal.

usage- `nano file_name`

CTRL + O – Save the file

CTRL + X – Exit

CTRL + K – Cut selected text.

CTRL + U – Paste the cut text

Example –

Open a file in read-only mode

`nano -v myFile.text`

cat - The cat command is used to display the contents of a file or concatenate multiple files.

- -n : Number the lines of output
- -b : Number non-blank lines only
- -E : Display $ at the end of each line

cp - The cp command is used to copy files and directories.

- -r : Copy directories recursively
- -i : Prompt before overwriting.
- -v : Verbose mode, show what is being copied.

Example -

Copy a file:


```
cp file1.text file2.text
```

Copy a directory and its contents:

```
cp -r dir1/ dir2/
```

Copy and show details of the operation

```
cp -v file1.text backup/
```

mv - The mv command is used to move or rename files and directories.

Example- Rename a file:

```
mv oldfile.text newfile.text
```

Move a file to a different directory.

```
mv file.text /home/user/documents/
```

Move multiple files:

```
mv file1.text file2.text /home/user/doc/
```

rm - The rm command is used to remove (delete) files and directories.

Usage- rm file_name

Example - Remove a file:

```
rm myfile.text
```

Forcefully remove a file without confirmation:

```
rm -f myfile.text
```

Delete multiple files:

```
rm file1.text file2.text
```

rm -r: The rm-r command is used to delete directories and their contents recursively.

Usage - `rm -r directory_name`

Example - Remove a directory and its contents:

```
rm -r mydirectory/
```

Forcefully remove a directory:

```
rm -rf mydirectory/
```

grep - The grep command is used to search for a string in files.

Example- `grep "pattern" file.text`

grep -i : The grep -i command is used for searching pattern in files while ignoring case differences.

Search case-insensitively for a term:

Example- `grep -i "hello" myfile.text`

This will match "hello", "Hello", "HELLO".

cd.. : The cd.. command is used to navigate up one level in the directory tree.

Move up one level from the current directory.

`cd..`

Linux File System

The Linux file system is a hierarchical structure used to organize files and directories on a linux-based operating system.
It is one of the core components of Linux.

/(Root Directory) - The top-level directory that contains all over files and directories in the system.

/bin - Essential binary executables (commands) needed for the system's basic functionality, available for all users.
(e.g, ls, cp, cat)

/boot - Contains files needed for booting the system, such as the Linux kernel and initial RAM disk.

/dev - Device files representing hardware (eg- hard drives, USB devices, terminal).
Linux treats devices as files.

/etc - System-wide configuration files and settings for the operating system and installed software.
(e.g - /etc/passwd, /etc/fstab)

/home - User home directories where personal files and user-specific configuration data are stored.
(e.g. /home/username)

/lib - Essential shared libraries and kernel modules required to run binaries in /bin and /sbin

/media - Mount point for removable media devices like USB drives, CDs and DVDs.

/mnt - Temporary mount point for file systems, often used for mounting external storage or network drives.

/opt - Optional software packages and third-party applications not part of the default installation.

/proc - Virtual filesystem providing system and process information.
(eg- /proc/cpuinfo , /proc/meminfo)

/root - The home directory for the root user (Superuser), not to be confused with /.

/sbin - System binaries essential for system administration (eg- shutdown , fsck), typically used by the root user.

/srv - Data for services provided by the system, like web servers or FTP services
(eg- /srv/www)

/sys - Virtual filesystem providing information about the kernel and devices. (eg- /sys/class /sys/devices).

Sunday 26

/tmp - Temporary files used by the system or applications, often cleared upon reboot.

/usr - User-related programs and data.
Subdirectories include:

/usr/bin : Non-essential user command binaries.

/var - Variable data such as log files, mail spools and application caches.
(e.g, /var/log, /var/sppol)

Basic System Commands

lscpu – Displays detailed information about the CPU architecture.

output –
CPU architecture (x86-64, i386)
Number of CPUs
Model name, Speed
Cache sizes
Vendor ID

free -h : Displays the amount of free and used memory in the system (RAM) in a human-readable format. (Swap memory)

Output –
Total, used and free memory
shared memory
Buffers/cache memory
Available memory

df -h : Reports the amount of disk space used and available on mounted filesystems in a human-readable format.

Filesystem name, size

used and available space
Percentage of used space
Mount point.

cat /etc/os-release - Displays information about the operating system

Output - The name of the OS
Version, ID
PRETTY_NAME etc.

hostname -I : Shows the IP address assigned to the machine.

uptime - Shows how long the system has been running and the system load.

top - Provides a dynamic real-time view of the system's running processes.

ps - Displays information about the currently running processes.

ps aux : Shows detailed information about all running processes

ps -ef : Displays all the processes in a full-format listing.

kill -9 : Forces the termination of a process using its PID (kill -9 [PID])

Note:- Use ps or top to find the PID of the process you want to kill

du -h : Estimates file and directory space usage in a human-readable format (e.g: KB, MB, GB)

- -s : Provides the total size of the specified directory or file
- -a : Includes file in the size calculations.

uname -a : Displays all information about system.

id : Displays information of user's 'Id' and groups.

M T W T F S S Wk Jan
1 2 3 4 5 01
6 7 8 9 10 11 12 02
13 14 15 16 17 18 19 03
20 21 22 23 24 25 26 04
27 28 29 30 31 05

Advanced Package tool (apt)

apt is a package managment system used by Debian-based Linux distributions such as Ubunt
It simplifies the process of installing, updating and removing software packages.

Package manager :- Manages the installation, removal and updating of softwa packages.

Why we use 'Linux' in DevOps ?

By using Linux, DevOps teams can leverage a flexible, secure and efficient platform for automating workflows, managing infrastructure and delivering software more reliably.

DevOps lifecycle - The DevOps lifecycle is all about creating, testing and delivering software faster and more reliably. Here's a simple explanation of its steps -

Plan - Think about what you want to build. Write down the tasks and goals. Use tools like 'Jira' or 'Trello' to organize the work.

Code - Write the code for the application or features. Store and manage the code using tools like Git or 'GitHub'.

Build - Convert the code into a working application. Use tools like 'Jenkins' or 'Maven' to automate the process.

Test - Check if the application works as expected. Run tests to find and fix any issues. Use tools like 'Selenium' or 'JUnit' help automate testing.

Release - Prepare the application for users. Packages the final version and get it ready for deployment.

Deploy – Put the application on servers or in the cloud.
Use tools like 'kubernetes' or 'Docker' for smooth deployments.

Operate – Keep the application running smoothly.
fix any problems that coms up.
Monitor performance using tools like Prometheus.

Monitor – Watch how the application is performing
Collect feedback and make improvements.

On-premises - 'On-prem' is a type of IT infrastructure in which servers, data and applications are hosted and managed at a company or organization's own location (office or data center).

This means that all hardware and software resources are under your control and you manage and operate them.

Key features of on-premises -

1.) We have complete control over hardware, data and software.
2.) On-prem infrastructure has higher data security because it does not depend on external servers
3.) We can customize it according to our needs.
4.) I've to buy a one-time license for on-prem software and hardware.

5.) On-prem is more expensive because you pay for everything upfront (equipment, software, staff) and ongoing costs (electricity, repair).

How On-prem Datacenter Works -

Components -

Servers - Physical machines that run applications and store data.

Storage - Hard drives, SSDs (Solid state drives) or SANs (Storage Area Network) to store data.

Networking - Routers, switches, firewalls and cabling to connect servers and ensure secure data flow.

Cooling - HVAC systems to maintain optimal temperature and humidity levels.

Power Supply - UPS (Uninterruptible Power Supply) and generators to ensure continuous power.

Operation -

Setup - Hardware is purchased, installed and configured on-site.

Maintenance - Regular updates, patches and upgrades are performed by IT staff.

Monitoring - Continuous monitoring of hardware and software to ensure performance and detect issues.

M T W T F S S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28

Wk Feb 2
05
06
07
08
09

Backup - Regular backups are taken to prevent data loss.

Disadvantages-

1) Expensive to start.
2) Scaling is slow
3) Need skilled people
4) High electricity bills.
5) Backup problems

Cloud - The cloud is like a big online storage space where you can save your files, photos and use apps over the internet instead of keeping everything on your computer or phone.

You can access this data from anywhere as long as you have an internet connection.

Advantages of cloud -

1) Saves money
2) Easy to use
3) Scalable (Storage In/Dr)
4) Accessible anywhere
5) Low maintenance
6) Secure
7) Collaboration
8) Always updated

Disadvantages -

1) Needs Internet
2) Less control
3) Privacy issues
4) Hack risks
5) Regular payment

How to Cloud Datacenter work -

Data Request - When a user accesses a cloud service (such as Google drive), a request is sent to the data center.

Processing - The servers in the data center process the request and extract the required data.

Data Delivery - The required data is sent back to the user via the internet.

Backup and Recovery - Data centers keep multiple copies of data at the different locations so that the data is safe and can be removed quickly.

Types of Cloud – There are three main types of cloud – (Based on Deployment)

Public Cloud – The cloud is owned and managed by a third-party provider (like AWS, Microsoft Azure, Google Cloud) and available to the public.

Features – Easy to use.
No hardware to maintain
Pay-as-you-go pricing

Who uses it? Startups, small businesses, or anyone who needs flexible and scalable resources.

Private Cloud – The cloud is used exclusively by one organization and can be hosted on-premises or by a third party (VMware vsphere, openStack)

Features – More security and privacy.
Customizable to specific business needs.
Higher cost than public cloud.

Who uses it – Bank, healthcare or finance

Hybrid Cloud - Combines public and private clouds, allowing data and application to move between them.

Features - Best of both worlds
Flexible and scalable
Complex to manage

Examples - An e-commerce website using the public cloud for the website and private cloud for sensitive customer data.

Cloud services - Cloud services is a technology that allows users to access computing resources such as storage, processing power and software via the internet.

Types of cloud services

There are three main types —

Infrastructure as a Service :- (IaaS) - It's like renting the hardware for your needs. With IaaS, you get virtualized resources like servers, storage and networking over the internet. You don't need to own physical servers. (or any resources)
(Rent hardware - like a computer)

Example - AWS, Google Cloud, Microsoft Azure.

Platform as a Service - (PaaS) - This is a platform that provides tools and services to develop, test and deploy software without worrying about managing the underlying infrastructure.
(Rent a platform to build and run apps.)

Example - Google App Engine, Microsoft Azure App Services

Software as a Service - (SaaS) - It's like using software on the cloud without having to install or maintain it on your devices.

You access the software via the internet, and everything is managed by the provider. (Use ready to go software)

Example:- Google Drive, Microsoft Office 365, Dropbox.

Virtualization -

Virtualization means creating multiple VMs on a single physical server. Each VMs acts like a separate computer with its own operating system and applications. This helps in efficient resource utilization and scalability.

Features -

Faster Server Setup - New VMs can be created in minutes without buying new hardware.

Environment Isolation - Separate VMs for development, Testing and Production.

Automation & Scalability - Easily scale VMs up or down as needed.

Cost-Effective -

Maximize hardware usage and reduces costs.

Azure - Azure is a cloud computing platform created by Microsoft. It provides a wide range of services like servers, storages, database, networking and even artificial intelligence tools, all available over the internet.

Purpose of Azure -

1) Host applications and websites.
2) Store data
3) Run VMs
4) Develop and test software
5) AI and analytics

Why is Azure useful -

Cost-Effective - You only pay for what you use.

Highly Scalable - Increase or decrease resources as needed (e.g. during high traffic on a website).

Global Access - Services are available worldwide, so you can access you data anywhere.

Secure - Azure provides advanced security to protect your data.

In simple terms, Azure is like renting a virtual IT Infrastructure over the Internet to run your business, store data or create software.

There are two task are performed in the cloud. It describes how you manage and operate Azure services. (Any cloud) —

1) Manual
2) Automation

Manual – Manual work involves performing tasks directly through the Azure portal or GUI

Advantages –

1.) User-friendly interface, suitable for beginners.

2.) Ideal for small and less complex projects.

Sunday 16

3.) Allows step-by step verification of actions.

Disadvantages –

1.) Time-consuming.

2.) Higher chances of human error.

3.) Difficult and inconsistent for large-scale projects.

Very Important

Automation – Automation involves using tools or scripts to complete tasks without manual intervention.

Methods of automation –
1.) Azure CLI
2.) Azure Powershell
3.) Azure RM Templates
4.) Azure devops Pipelines
5.) Terraform

Advantages –
1.) Saves time and effort for repetitive tasks.
2.) Reduces human error.
3.) Ideal for large-scale and complex deployments.
4.) Enables consistency in deployments.

Disadvantages –
1.) Requires learning scripting and tools.
2.) Initial setup may be time-intensive.

In Azure Automation, Imperative and Declarative approaches are two ways to manage and deploy resources.

Imperative (How to do it) - You describe how to do every step.

You have complete control over the process.

Example - Using Azure CLI, you say:
"Create a VMs, name it 'MyVM', and install Ubuntu."

```
az vm create --name MyVM --resources-group
MyResourcesGroup --image UbuntuLTS
```

This is imperative, where you explicity define each step.

When to use - When you're doing a small task or need full control.

Declarative (What you want) -

You only describe what you want.

Azure figures out the step to achieve it for you.

Example - You provide a template saying.
"I need a VM's named 'MyVM' with Ubunty installed."

```
{
  "resources": [
    {
      "type": "Microsoft.Compute/vm",
      "name": "MyVM",
      "properties": {
      "vmSize": "Standard_DS1_V2"
      }
    }
  ]
}
```

Azure reads the templete and creates the vm automate.

When to use - When automating or working on large projects.

Key Differences

Imperative	Declarative
1.) Focuses on how to do it.	1.) Focuses on what you want.
2.) You control every step manually.	2.) Azure decides the step for you.
3.) Best for small tasks.	3.) Best for automation and scaling.

Token-Based Authentication and Authorization -

Token based authentication and authorization are widely used mechanisms in modern web application and API's to ensure secure access to resources.

Here's a detailed explanation of how they work—

Authentication (Who are you) -

Authentication is the process of verifying the user's identity. It typically involves these steps —

Step 1 - Login - You type your username and password and send it do the server.

Step 2 - Token Creation - If you login details are correct, the server gives you a token.

Step 3 - Keep the Token - You store this token safely (In your app or browser).

(Think of it like a key that says, "I'am allowed to use this system.")

Using the Token - Every time you want to access something (like your profile or data), you send the token along with your request.

Example - "Hey server, here's my token. Can I get my data?"

Authorization (What are you allowed to do?)

Step-4. Token Check - The server checks if your token is valid.

* Is it real?
* Has it expired?
* Does it allow you to do this action?

Step-5 Grant Access -

If everything is fine, the server says, "Okay, here's what you asked for."

If not, it says, "No you can't do this."

Token expiry - Tokens don't last forever (Security). Once your token expires, you'll need a new one by:

Why use tokens?

You don't need to send your password everytime.
It works across multiple devices.
It's fast and secure.

Azure Entra ID - Azure Entra Id is Microsoft's cloud-based identity and access management service.

It is used to manage ~~servi~~ users, groups and devices, ensuring secure access to resources. (Azure AD). (Active directory)

Key-features-

Manage Users - You can easily manage user accounts and give them access to apps.

(Multi-factor Authentication)

Extra Security (MFA) - It makes your apps more secure by asking for extra verification, like a phone number or fingerprint, in addition to a password.

Single Sign-On (SSO) - Enables users to log in once and access multiple applications seamlessly.

✉ @ ☎

Role-Based Access - You can give people different levels of access based on their job role.

Collaboration with Partners - You can easily share access with other companies or external users securely.

Tenant ID - Tenant ID is a unique identifier that uniquely identifies each Entra ID.

When a company sets up Azure Entra ID, it gets a tenant and that tenant has a tenant ID

There is only one tenant ID for a company, but it can have multiple users within it.

"**Azure Hierarchy**" - Azure has a hierarchical structure designed to help organize and manage cloud resources in a scalable and secure way.

The structure is as follows-

Managgement group - These help organize multiple subscriptions into a hierarchy. Management groups enable centralized managment and policy enforcement across subscriptions.

Subscriptions - It is a container that holds resources like VM's, storage and database. It is a key component in Azure's hierarchy and is used to manage and organize your Azure services and billing.

Resource Group - Inside a subscription, used to organize related resources.

Resources - Resources are the individual services or objects (like VM's storage accounts, database) That we use within Azure.

Management Group -

Purpose - Management groups help to organize subscriptions in a hierarchical structure. This helps apply governance at a higher level (across multiple subscription) such as policies, access control and compliance.

Features -

* Management groups support up to six levels of hierarchy.

* You can apply policies and role-based access control (RBAC) to management groups, which will propagate to all subcriptions beneath them.

* Management groups allow you to apply consistent governance and security policies across large, complex organizations.

How to create a management group :-

* In the Azure portal, navigate to "Management Groups".
* Click "Add Management Group".
* Specify a name for the management group and select the parent management group (if applicable).
* Assign policies or RBAC roles to the management group as needed.

Subscriptions -

Purpose - Subscriptions are containers for Azure resources. Each subscription has a unique ID and a billing account.

Features -

* Each subscriptions has a separate billing model.

* Subscriptions can be used to isolate resources for different environments (e.g. development, testing, production).

* Subscriptions are tied to a specific Azure AD tenant for identity and access management.

"You can associate multiple subscriptions with a management group."

Resource Groups -

Purpose - A resource group is a logical container in which Azure resource are deployed and managed.

Features -

* Resources In a resource group can share lifecycle, permisslons, and policies.
* You can deploy, update and delete resources in a resource group as a unit.
* Each resource must belong to a resource group.

How to create a resource group -

* In the Azure portal, navigate to "Resource Groups".
* Click "Add".
* Provide a name and select the region for the resource group.
* Click "Review + Create" to deploy the resource Group.

Resources -

Purpose - Resources are the actual services and objects deployed in Azure. (e.g. virtual machines, storage accounts, network).

Features -

* Resources are deployed in resource groups and can be managed using Azure Resource Manager (ARM).

* Each resource has specific settings, configurations and capabilities.

* You can manage resources using the Azure portal, CLI or ARM templates.

Roles in Azure - Azure provides role-based access control (RBAC) to ~~mange~~ manage access to Azure resources.

Three key built-in roles are -

1) **Owner** -
Full control of all resources, including the ability to assign access to others.

Use Case -
Typically assigned to administrators or Individuals who need full control over a subscription or resource group.

2) **Contributor** - Can create and manage all types of Azure resources but can't grant access to others.

Use case - Suitable for developers or DevOps engineers who need to manage resources but don't need to assign permissions.

3.) **Reader** - Can only view resources and their configurations. No permissions to modify or delete resources.

Use case -
Best for auditors or team members who need to monitor resources without making changes.

RBAC - (Role-Based Access Control) :-

Scope : Manages access at the Azure resource level (subscriptions, resource group, VMs, storage, etc).

Purpose - Grants permissions to manage Azure resources.

How to create a New HR User (Shaurya) in Entra ID -

Sign in to Microsoft Entra ID - Go to Microsoft Entra ID (Azure AD).

Navigate to users → New User.

Create the User -

Select create user.

Fill in the details -

(User Principle Name) - Shaurya@yourdomain.com

Name - Shaurya

First name - Shaurya

Last name - HR

Set an initial password (ensure the user changes it upon first login).

Assign HR Role (Optional, If needed for HR functionalities) -

Go the Role and administrators.

Assign appropriate HR-related roles if required.

Grant Contributor Access to the Management Groups -

Navigate to Management Groups - Go to Azure Portal → Management Groups

Select the Target Management Group - Choose the Management Group where access is needed.

Assign IAM Role -
Click on Access Control (IAM) → Add role assignment.

Select Contributor role.

Under Members, add Shaurya@yourdomain.com

Click Review + Assign.

Verification -
Ask Shaurya to log in to Azure Portal and verify that he has contributor permissions on the Management Group.

Ensure he has can manage resources but not assign roles (Contributor role does not allow RBAC assignments).

Managing Groups in Entra ID -

Managing Groups help manage access to resources efficiently by assigning permissions to multiple users at once.

Types of Groups in Entra ID -

Security Groups - Used to manage user and device access to resources (e.g - applications, files, subscriptions).

Can be assigned to RBAC roles.

Supports dynamic membership based on user/device attributes.

Microsoft 365 Groups - Used to collaboration (e.g - in outlook, Teams, SharePoint)

Provides shared mailbox, calender and Onedrive.

Not used for RBAC assignments.

How to create a Group in Entra ID -

Sign into Entra ID - Go to Entra ID Portal.

Navigate to Groups → New group.

Choose Group Type - Select Security (for role-based access) or Microsoft 365 (for collaboration).

Fill in Details -

Group name - HR Team

Group Description - Group for HR team members with management access.

Membership Type -

Assigned - Manually add users.

Dynamic User - Auto-add users based on attributes (e.g - department = HR)

Dynamic Device - Auto-add devices based on attributes.

Add Members -

Click Members → Add

Shaurya@your domain.com and other HR users.

(Done)

Benefits of Using 'Groups' for Access Control -

1.) Easier Management
2.) Improved security
3.) Automation
4.) Scalability.