**DEVOPS**:

DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams.

Software: collection of computer prgms that helps us to perform a task. Also different types

1. System software – To run the system

(Device drivers (keyboard, mouse), operating sys, servers, utilities)

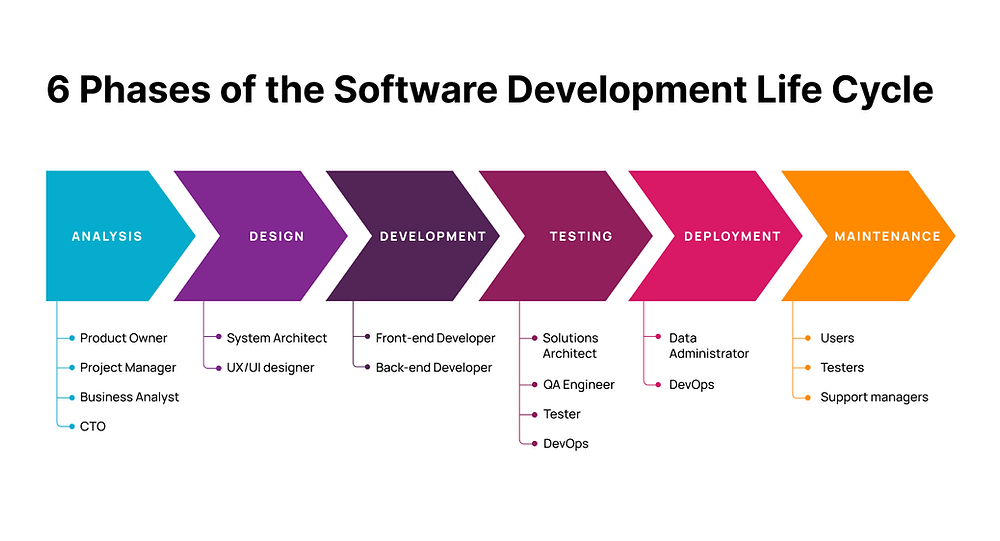
2)programing software—Developers write the code & compile the output.

(compilers, debuggers, interpreters)

3)Application software—users will use the Application to perform specific tasks.

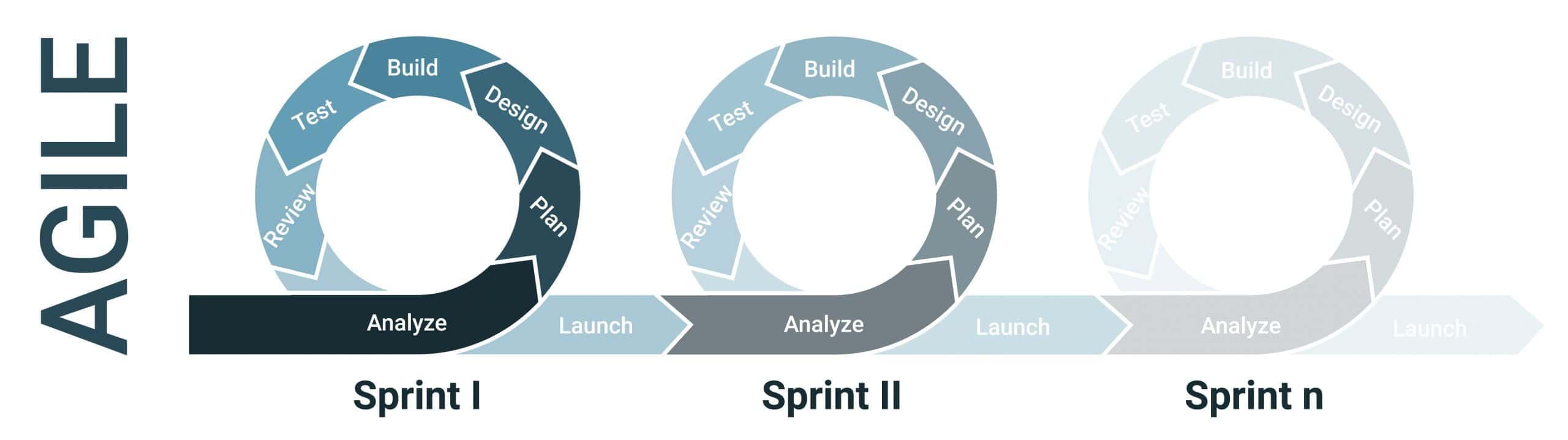
(web App, mobileapp, desktop app)

**SDLC**: It is a process used by software industry to design, develop, & test the software’s.

Initially, there are Waterfall model, Spiral model, V-model.

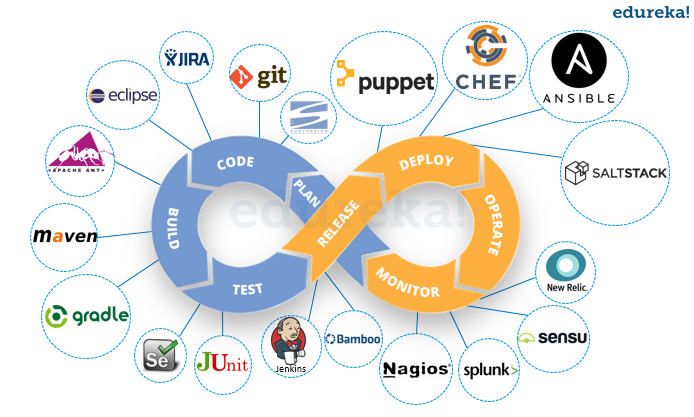
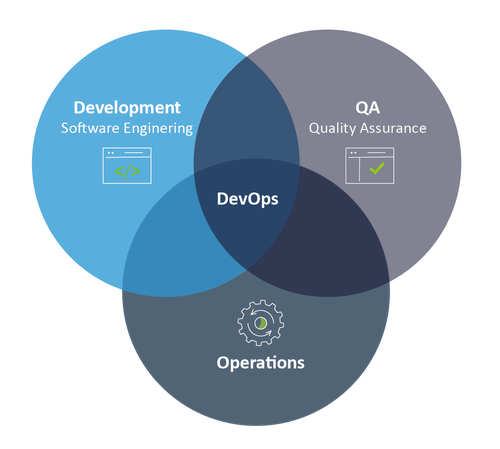
Present, Agile Model/ Agile Methodology: It is an iterative & Incremental approach/process.

Scrum, Sprint Retrospective meetings ..

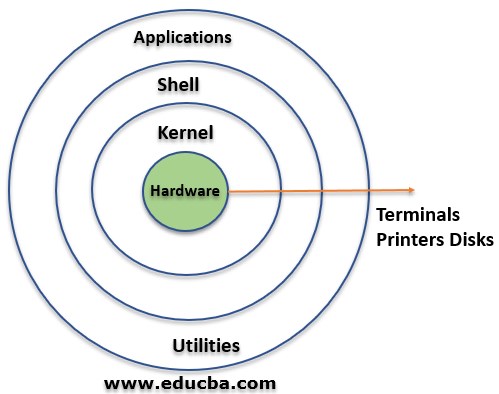


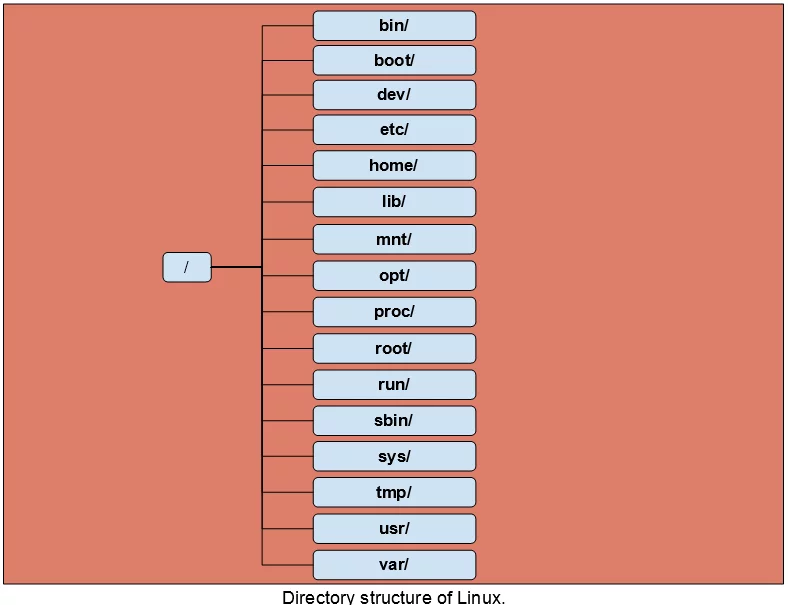
**DEVOPS**

**Main benfits of devops**: (Increase Release velocity,shorter deploment cycl,deployment rollback, bugs detection,recovery from failure, collaboration,performance oriented.)

**Principles**: (Incremented, Automated, collaborative,Iterative, self-service)  

LINUX





**/root** 🡪 home directory for the root user.

* /**dev** 🡪 contains device files such as /dev/sda.

• **/boot** directory 🡪 Static boot files

/**etc** 🡪 system configuration files & log files

•Applications and user utilities are found in the **/usr** directory.

•**/var/logs**  🡪contains log files of various system applications.

•The **/home** directory is where user folders are located. These

include Desktop, Documents, Downloads, Music, Public, and

Videos.

* **/opt** directory 🡪 application packages

•The **/media** directory stores files for removable devices such as USB drives.

•**/mnt** directory contains subdirectories that act as temporary mount points for mounting devices such as CD-ROMs.

•The /**proc** directory is a virtual file system that holds information on currently running processes. It’s a strange file system that is created upon a system boot and destroyed

upon shutdown.

• **/bin** 🡪 user command binary files.

•**/lib** 🡪 stores shared library images and kernel modules.

**Shell:** it is user interface to access by users. **Types**: Korn shell, BASH, Bin, C shell etc

**Absolute path** is the complete path of a file or directory from the root directory (/)

**Relative path** is the path relative the current location in directory.

A **soft link** is a pointer to a file, directory or a program located in a different location.

A **hard link** can point to a program or a file but not to a directory.

cmd ln –s for creating a soft link., hard link can be created by ln command without –s option.

**Linux commands:**

Ls-> listing directory

Ls –l 🡪 long list files shows time & date

Ls –a-> list with hidden files

Cd -> change directory

Mkdir -> creating a directory

Removing directories (rm –r, rm –f, rm –rf )

Cp file1 file2 (copies file1 to file 2)

Cp –r dir1 dir2 (copy dir1 to dir2; if dir2 not present it will create)

Touch <filename> 🡪 creates the file

Cat > filename (to create a file)

Cat filename 🡪 to view/read the content or file

Vim or vi 🡪 create and open the file

Wget, curl 🡪 downloading the files or urls from web

Cut 🡪 used to cut string based on delimiter

Awk 🡪 processing/analyzing txt files (lines, rows, columns)

Filter commands (cat, awk, grep, head, tail cut etc)

**System info:**

Date, df(disk usage), du (dir space usage), free (show memory & swap usage)

% pingwww.google.com(remote host alive?), % history | tail -2(to get last executed cmnd)

**File permissions**: (chmod) A file or directory have 3 options (read(r), write(w), execute(x) )

Numerical notation:

1 🡪Execute, 2🡪 write, 3🡪 w+x 4🡪Read 5🡪 r+x 6 🡪r+w 7🡪 All

Eg : chmod 777 filename , chmod 667 filename2 ,

Alphabetical notation: owner (u) group (g) other (o) all users(a)

+ 🡪giving permission - 🡪 removing permission = 🡪 replace Eg: chmod u+w file1, chmod u-w file2, chmod u=x file3, chmod u+w, g-x file4, chmod o+wx file 6

**Chown commands: (master to root)**

chown owner\_name <file\_name>(to chnge owner of the file),

chown :group1 file1.txt(to chng group of the file), chown master:group1 file1.txt

(to chng both owner & grp of the file), chown --from=master root greek1 (chng the owner from particular ownership only), chown --from=:group1 root greek1(chng the grp from particular group), chown master:group greek2 greek3 (chngng ownership of multi files)

chown --reference=greek1 greek2(copy ownership from one file to another)

**Searching:** grep, find, eg: grep google, grep –i google password,

**Pipe commands**: o/p of pipe1 becoming i/p to pipe2 & o/p of pipe2 will i/p to the pipe3

Eg cat password | grep google, head -5 file1 | tail -2

**Process Management**: ps -ef , top, kill, kill pid, pkill pattern, fg n (bing job n to foregroud) , sleep,

**Package management**: Ubuntu – apt –get

Centos – yum, linux – yum install

**Service Management**: Every resource have port numbrs

IP,STTP,HTTP,HTDP,HTTPS, TOMCAT 8080

Systemctl start nginx, systemctl status/enable nginx

Start Jenkins.service,

**Network management**: netstat , netstat –lntp, ps –ef | grep nginx, ping host, ssh, telenet, tracert, ftp,ifconfig, dig,nslookup,wireshark.

**User management:**

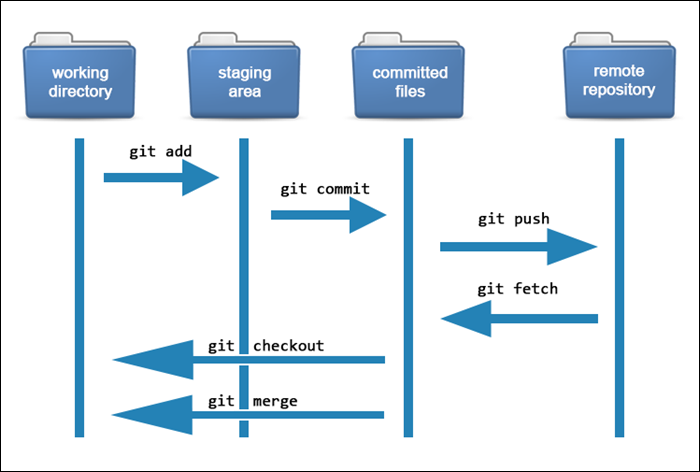
Sudo su, sudo –I, sudo useradd username, passwd username. usermod -u new\_id username (**change the user ID for a user**)

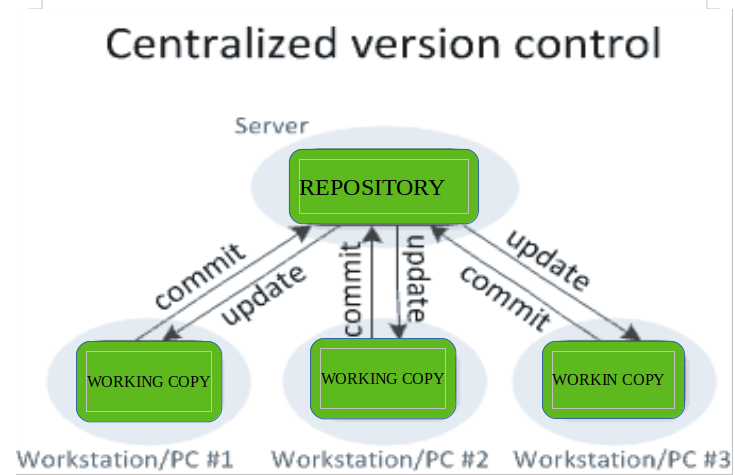
usermod -g new\_group\_id username(**Modify the group ID of a user.)**

userdel -r username(deleting user), usermod -d new\_home\_directory\_path username

(c**ommand to change the home directory**), sudo addgroup groupname, sudo addgroup groupname --gid 12345 .

**GIT**: Git is a distributed version control system that tracks changes in any set of computer files, used for coordinating work among programmers who are collaboratively developing source code during software development.





**Git config –global username =”nmae”, Git config –global user.email=”myemailid”, Git config –global --list**

git pull(pulls latest changes from remote repo to local) git restore –staged <filenm1>

git push(sends latest chngs from local to remote repo), git branch(lists of all branchs)

git branch branchname(creates a new branch)

git checkout branchname(switch to a new branch), git merge brancname(merge branch into active branch), git stash(stores safely in hidden place), git stash apply, git log, git log --online git reset(reset ur head pointer to a previous commit), git rm filename, git rebase(reapply commits on top of another base tip)

Difference b/w: **git merge** is a way of combining changes from one branch (source branch) into another branch (target branch) whereas **git rebase** is a way of moving the changes from one branch onto another branch.

Git revert (create a new commit that undoes all of the changes made in a particular commit, then apply it to the current branch)

Git cherry-pick commitid (applied the chngs introduced by some existing commits)

Git amend (new minor changes added by dev ina code)

Eg: git commit –-amend – m “my message”

**Networking concepts**

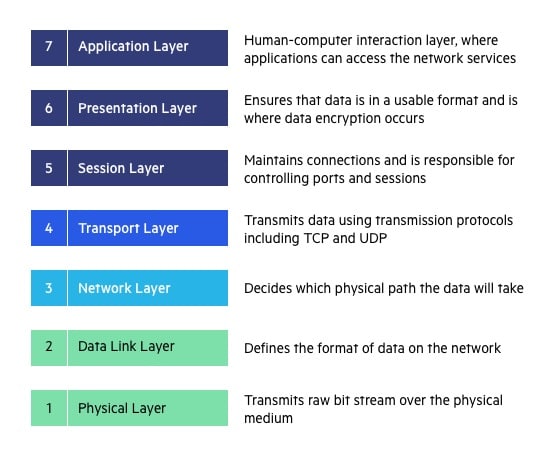
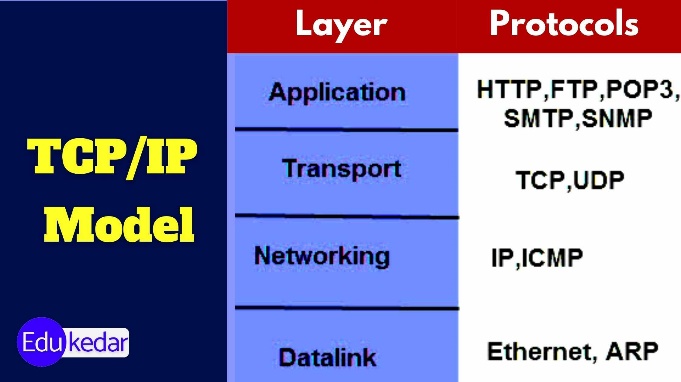
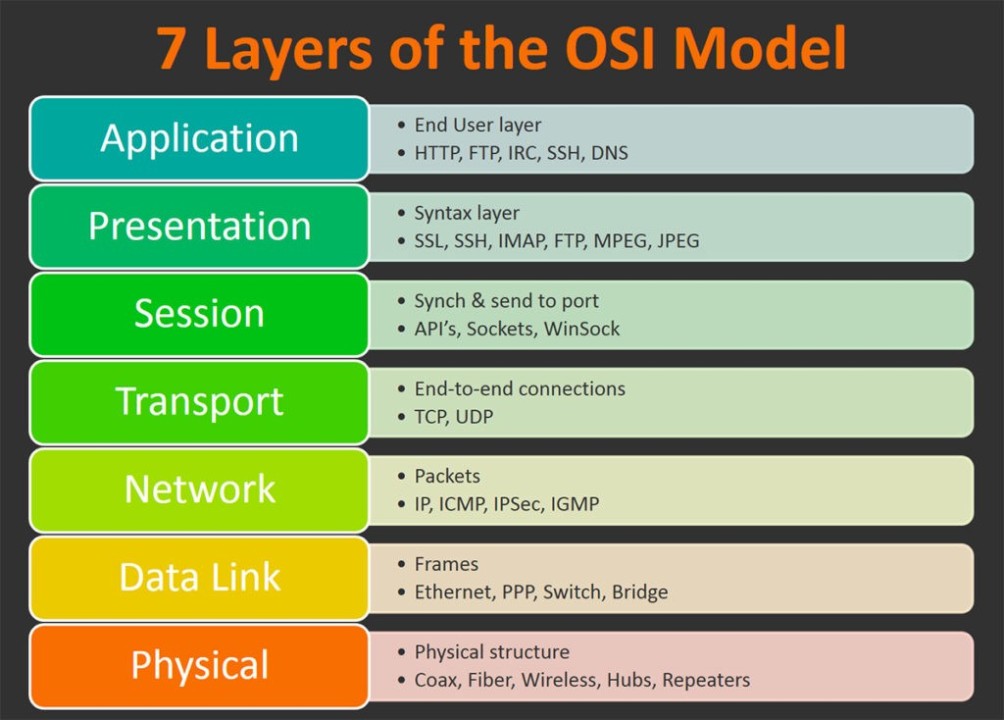
**Network**: A group of computers/systems.

**Internet**: A group of Networks.

Systems communicate through cables, fiber optics, wireless etc., which are the physical entities needed. The communication follows a protocols like TCP/IP, ssh, sftp, ftp, sockets, telnet etc.

1. OSI model (open systems connections)
2. Protocols: TCP/IP
3. Ports
4. Sub netting
5. Routing
6. DNS
7. Firewalls & Security Groups
8. HTTP/HTTPS
9. CDN (Content Delivery Network)
10. VPN (Virtual Private Network)
11. Networking Tools

**Types of computer network models: (OSI model & TCP/IP model)**



**Protocols**: A set of rules that defines how data is transmitted & received b/w devices in a network. (TCP/IP,HTTP,HTTPS & SMTP)

**TCP (connection oriented)**: it breaks data into packets,sequence num & users ack msgs to gurantee delivery.

**UDP(connectionless)**: it sends data without establishing connection, providing low latency communication.

**IP**: It assigns unique IP address to devices and uses routing tables to direct data across n/ws.

**PORTS**:

Ports are comm end points that allow diff services on a device to send & receive data.

Eg: ssh 22, tomcat 8080, sonar 9000, DNS 53, SMTP 465

**Subnetting**:

Subnetting is a technq used in computer n/w to divide a larger IP n/w into smaller, more manageable sub-n/w or subnets.

