# Cloud Computing:

## Cloud:

Everything that is available publicly on internet.

Or we can say, it is on demand use of computing resources where direct management from user is not required. It’s like electricity service, you pay for what you consume (like units in case of electricity).

## Reasons of switching to Cloud:

Think about on-premise infrastructure, and say you want to deploy web application may be a social web application.

1. You may require servers, which are very costly.
2. Monitoring servers are your job
3. Keeping things updated as per trend of technology is again your job
4. You don’t know about traffic, right? Sometimes traffic may increase much than you expected. What will you do in such case? Buy more servers? Is this worthy? What if in future business go down, you wasted much on servers right? Means not easily scalable.

Cloud caters these drawbacks and thus these mentioned draw backs give motivation to switch to cloud.

## Service Models of Cloud:

### SaaS:

Software as a Service. These services are available publicly on internet. You pay rent for them and you start using them. Sometimes they are free. Your main concern is on **data.**

**Example:** Google Drive. Dropbox etc.

### PaaS:

Platform as a Service. Your main concern is on **data and application.** You will pay for a platform and you will start using platform for may be hosting your applications and managing data.

**Example**: Go Daddy.

### IaaS:

Infrastructure as a Service. You will purchase underlying infrastructure from some vendor may be Amazon web services, or from Microsoft Azure, it could be Google Cloud Services. Maintenance will be your concern. Like Installing OS, middleware’s, hosting applications and managing data will be your task.

**Example:** AWS EC2 instance.

## Deployment Models in Cloud:

### Public Cloud:

This model of cloud is available on internet and can be easily accessed from anywhere and by anyone.

#### Advantages:

1. Minimal charges
2. No setup cost
3. No maintenance required
4. Infrastructure management is not required
5. Scalable

#### Disadvantages:

1. Limited Control
2. Limited Features
3. Less secure

### Private Cloud:

Type of cloud which can only be accesses under certain premises is private cloud. It can be like accessed by VPN.

#### Advantages:

1. Better Control
2. Security and privacy
3. Customization
4. Supports legacy system

#### Disadvantages:

1. Not scalable
2. Higher cost than public cloud

**Example:** Comsats Library Management.

### Hybrid Cloud:

This type of cloud is actually mixture of both public and private model. For example, HEC website is available publicly to everyone but what about data of students? Is it available publicly?

#### Advantages:

1. Flexibility & customization
2. Lower cost
3. Scalable

#### Disadvantages:

1. Difficult to integrate
2. Requires maintenance which results in more cost

# Myths About Cloud Computing:

1. **Cloud is “one size fits all”:** One service will not serve you well, there are different services which you can choose among available services based on business.
2. **Virtualization is Cloud:** No virtualization is not cloud. They are totally separate concepts.
3. **Cloud Locks you in:** Vendors aim to have good report with customers but it doesn’t mean they lock you in. As per client its your responsibility to choose services.
4. **Cloud costs Job:** No this is not the cause, when existing structure is moving to cloud from on Premise, people have to do same jobs in cloud. Management is now on cloud.
5. **Need more than one cloud:** Not true, because AWS has hundreds of services, if it’s for their business than they make sure you will utilize all. But it’s not the case you use what you want.
6. **Cloud is For Tech Company:** It basically started as storage services. But now they cater need of different audience.
7. **In the cloud or not:** There are businesses which are on premise and some on cloud. Well, it’s totally on business needs.
8. **Everything works better here:** It provides you better services but if you are not smart enough, you may not get potential advantage from it.
9. **Cloud is not secure:** Yes, there were chances of security breach in beginning but now it’s very secure
10. **Cloud is always affordable**: Some services are so expensive, such that if you don’t choose wisely you will suffer.

**Some others are:**

1. Public cloud is only one true cloud
2. Migrating to cloud is not worthy
3. Cloud Cases Environmental hazards.

# Virtualization:

It occurs at hardware level. On the top of hardware is hypervisor, it is responsible for virtualizing NICs, and other resources may be.



In virtualization we are achieving machine level isolation, each machine is independent of each other. They really don’t know about each other. It appears as completely other machine running on somewhere else.

It can be type-I or type-II virtualization.

## Type-I:

In type-I we have hyper visor running as OS on a machine and then we instantiate or virtualize resources using it.

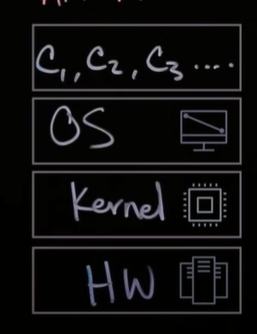
## Type-II:

In this type software’s like VMware, Oracle VM box are installed on OS for creating virtual machines.

Virtualization provides huge amount of **flexibility.** You can very flexibly virtualize machines as per desired resources.

# Containerization:

It occurs at software level that’s why it is also known as OS level virtualization. Here on the bottom, we have hardware and then on top we’ve kernel and then Host OS. And on the top of OS, we have runtime engine which is then responsible for creating containers. Since OS is common it is known as OS level virtualization.



We are talking about process level Isolation. As containers are running on base OS which is common, we normally isolate them so that they appear to be different and may be for security purposes.

Containerization provides high level of portability. They are just like a piece of code. (Normally known as docker files) You wrote necessary files or libraries, you can share with others and run them very easily.

Resources are purely dynamic, if some container is not using resources, its resources may be shared with other containers.

# GitHub:

It’s basically distributed version control system. It stores repositories which are stored initially in local system. It follows SaaS service model of Cloud Computing.

## Git:

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

### Installation of Git:

<https://git-scm.com/downloads>

### Usage:

#### Configure user name email and other settings:

*Git config –global user.name “m-ali-hasnain”*

*Git config –global user.email* [*ali@gmail.com*](mailto:ali@gmail.com)

*Git config –global color.ui auto*

You can check configurations by: *git config -l*

#### Initialize git repository:

First of all, initialize repository by command*: git init.* This repository will later serve you to track files, share or push to remote server.

#### Viewing Status:

You can view status of staging area by command: *git status*

#### Staging Area in Git:

These files are also referred to as "untracked files." Staging area is files that are going to be a part of the next commit, which lets git know what changes in the file are going to occur for the next commit. The repository contains all of a project's commits. They can also be called untracked files.

#### Adding in staging area:

Git add <file>

*Git add .* -> add all files inside current directory. But it has some draw backs when say directory structure is something like this:

Main

.git

Index.js

Test

Index.html

Index.css

Now if you are in test directory and if you say *git add .* , it will only add index.html and index.css and not index.js, so to add it in staging area you can use command: *git add -A*

#### Remove file from staging area:

*Git rm –cached <filename>*

To remove all files from staging area use command: *git rm –cached -r .*

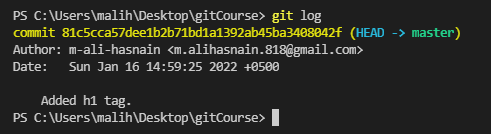
#### Commits:

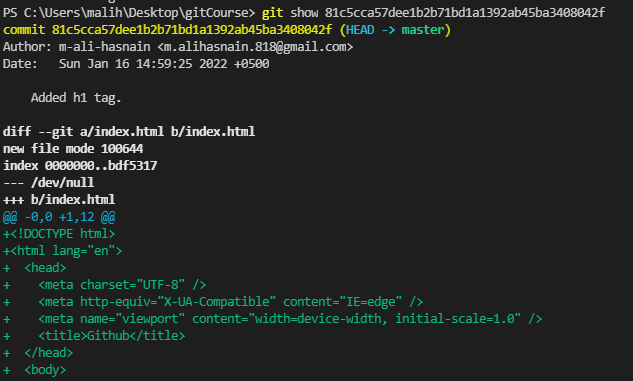
They are basically safe points. We can commit files that are in staging area. History of these commits are mentioned by git.

To commit use command: *git commit -m “msg”*

#### Viewing commit logs:

*Git log*

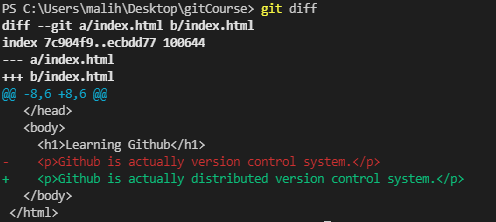




Git show <commitHash> showed you what has changed in file.

#### Find difference by modified and unmodified file:

Command: *git diff*



#### Push to Remote Server: adding origin:

*Git remote add origin <repo\_link>*

##### Setting branch:

*Git branch -M main*

##### Pushing code:

*Git push -u origin main*