3. What problem does Virtualization solve and what is its drawback in context to modern application deployments?

Before virtualization when a company had a server like their own database server or a server where Jenkins were running, Company manage their own server

When OS is directly installed on the hardware then the OS is tightly coupled to the hardware that means

- If the hardware component of that computer failed (hard disk, motherboard failed,etc) and is not repairable any more,
- Then your whole computer will be useless, and the OS, applications installed and all the data will be gone.

With Virtualization we can have our OS as a portable file that we can move around and these files (Virtual Machine Images)

So our OS, application installed on it, all the configurations and all the data will be inside that portable file. And we can even make copies of Image files as a purpose of backup.

More benefits can illustrated as:

Cost savings:

We can run multiple virtual machines on one piece of infrastructure. We don't have to maintain nearly as many servers.

Learn and Experiment

- You don't need to buy a new computer
- You don’t endanger your main os

Test your app on different OS

Example: if we are developing a website and we want to see how that works and how it looks like in different OS in different browsers i.e how your application performs in linux machine in a firefox browser or on windows in internet explorer browser.

And when you are done we can easily delete them.

Efficient uses of Hardware resources

- Use all resources of a performant big server.
- Users can choose any resource combinations.

Each VMs runs in its own isolated environment i.e. completely independent of each other.

Issues and incompatibilities

If something breaks inside VM, it does not affect the host machine

Agility and speed

Spinning a virtual machine is relatively easy and quick. And helps developers for dev-test scenarios.

Lowers the down time

If the host goes down we can simply migrate VMs from one host to another.

Abstraction of OS from the hardware

VMs' drawback in context to modern application deployments

In context of Modern Application Deployments, VMs are losing popularity when IT services needs :-

- **Faster development and release**

Since for installing any application we need multiple commands, checks for multiple dependencies, resolve the conflict of libraries if any and we have to repeat the same process if we are installing the same application in a different environment(development, QA, Production).

- **High Scalability**

Workload operation rarely consumes all the resources made available to associated VM. As a result, the remaining unused resources may not be incorporated in capacity planning abd distribution across all VMs and workloads. i.e. significant resource wastage

- **Flexibility to evolve application development in response to changing business and market needs.**

Monolithic app development practices are losing popularity and organizations are pursuing infrastructure architecture solutions to further optimize hardware utilization. This is precisely why containerization was invented and gained popularity as a viable alternative.