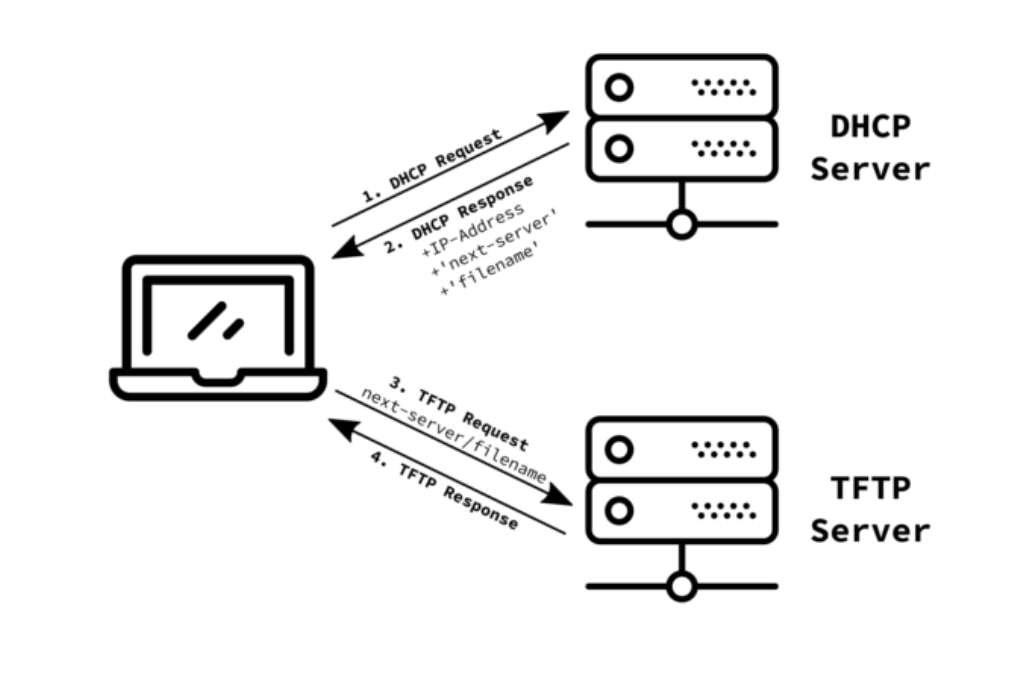
Q1) What is PXE?

Ans 1) :-

**Definition:-**

Pre-boot execution environment (PXE) is a set of standards that enables a computer to load an operating system (OS) over a network connection or on multiple servers. PXE can be used to quickly install an OS and is commonly used for both servers and clients.It is also called PXE boot, network boot, boot from network, local area network boot.

**Working Of PXE:-**

Dynamic Host Configuration Protocol server

1. The client basic input/output system(BIOS) initiates PXE boot. This may be selected by the client operator or may be a fallback option when other boot media fails.

Trivial File Transfer Protocol server

1. The client broadcasts a DHCP request and a PXE request.
2. The DHCP server responds with the DHCP response so the client can set an IP address, and it replies with the IP address of the TFTP server and the file name of the NBP.
3. The client downloads and boots the NBP which is a small OS that contains just a kernel, basic drivers and basic programs that can download the remaining OS components.

Q2) What are virtual machines, What is the value add that they provide, at least mention 3 use cases?

Ans 2 :-

**What are VM’s**

A virtual machine (VM) is a virtual environment that functions as a virtual computer system with its own CPU, memory, network interface, and storage, created on a physical hardware system (located off- or on-premises). Software called hypervisor separates the machine’s resources from the hardware and provisions them appropriately so they can be used by the VM.

VMs are isolated from the rest of the system, and multiple VMs can exist on a single piece of hardware, like a server. They can be moved between host servers depending on demand or to use resources more efficiently.

VMs allow multiple different operating systems to run simultaneously on a single computer—like a Linux Os on a MacOS laptop. Each operating system runs in the same way an operating system or application normally would on the host hardware, so the end user experience emulated within the VM is nearly identical to a real-time operating system experience running on a physical machine.

**VM’s have a lot of use cases such as:**

1. **Server Consolidation -** It is one of the top reasons to use Vm’s. Most operating system and application deployments only use a small amount of the physical resources available when deployed to bare metal. By virtualizing our servers, we can place many virtual servers onto each physical server to improve hardware utilization. This keeps us from needing to purchase additional physical resources, like hard drives or hard disks, as well as reducing the need for power, space, and cooling in the datacenter. VMs provide additional disaster recovery options by enabling failover and redundancy that could previously only be achieved through additional hardware.
2. **Trying New Operating Systems -** If you have always worked on windows and want to try out a different OS like Linux. You have several options of trying Linux, including a dual-boot setup, but virtualization is a great way to try it out with little risk. On your Windows system, you just need to install VirtualBox (or another hypervisor) and create a new virtual machine. Then take any Linux installation ISO (like Ubuntu or Linux Mint) and install it as a virtual machine. Then you can run Linux (the guest OS) in a window within your Windows system (the host OS) like any other program.
3. **Run Old or Incompatible Software -** Virtual machines provide a great environment for running software that's not compatible with your current machine. As long as you have access to the installer, you should have no trouble installing it on a virtual OS. Plus, running outdated software in this way is much safer because it's sandboxed inside the virtual machine. Hopefully you don't have to run Windows XP-only software these days, but the option is there in case.
4. **Develop Software for Other Platforms** - Another important use for virtual machines is simplifying the workflow for testing apps and websites across multiple platforms. For example, say you're developing a game that works on both desktop and mobile platforms. You can use emulation to test the various versions right on your computer. Instead of moving installer files back and forth to your phone and other test computers, you can just emulate them.Virtualization also lets you compile to other executable types. Even if you use a cross-platform framework, you may only be able to compile APP files on Mac and EXE files on Windows. Instead of dual-booting for every build, virtualization simplifies the process. Sometimes there's no substitute for running apps on real hardware, as emulation isn't perfect. But for many uses, emulation is a convenient way to access other OS’s without much hassle.