**What is the difference between VM and containers?**

In summary, virtual machines provide full OS virtualization with dedicated resources, while containers offer lightweight application-level virtualization with shared resources and faster startup times. Both have their use cases depending on factors such as performance requirements, resource utilization, portability, and isolation needs.

1. **Operating System (OS) Isolation:** Virtual machines emulate an entire computer system, including a guest OS, on top of a physical host machine. Each VM has its own kernel, libraries, and system resources, which provide complete OS isolation. In contrast, containers leverage the host OS and share its kernel, enabling multiple containers to run on a single host without the need for separate guest OS instances.
2. **Resource Allocation:** VMs require a hypervisor to allocate dedicated system resources such as CPU, memory, and storage for each virtual machine. Each VM runs its own OS, which consumes a significant amount of resources. Containers, on the other hand, share the host's resources. They use containerization technologies like Dockers or Kubernetes to isolate and manage processes within their own runtime environment.
3. **Performance:** Since VMs run on top of a hypervisor, there is a slight overhead in terms of performance due to the additional layer of abstraction between the guest OS and the host hardware. Containers, being lightweight and using the host's kernel, have less overhead and provide better performance. Containers can start quickly and consume fewer system resources compared to VMs.
4. **Portability and Scalability:** VMs are known for their portability. They encapsulate the entire guest OS and application stack, allowing VMs to be moved and run on different host machines or cloud platforms. However, VMs require more time to start and occupy more storage space. Containers excel in terms of portability as well but with a focus on application-level portability. Containers package applications and their dependencies, making them highly portable across different environments. Containers are also highly scalable, allowing multiple instances of a container to be easily replicated and managed.
5. **Isolation and Security:** VMs provide stronger isolation between different VMs since they run separate OS instances. This isolation enhances security and helps prevent one VM from affecting others. Containers provide a level of isolation, but they share the same host OS. While container security has improved with technologies like Linux namespaces and control groups, a vulnerability in the host kernel could potentially affect all containers running on it.

**Create 4 virtual machines in the server to Ubuntu with 1cpu, 2GB, 8GB hard disk**

MariaDB 192.168.200.23/24

MongoDB 192.168.200.24/24

Neo4j 192.168.200.25/24

Minio 192.168.200.26/24