

Types of Virtualization in Cloud Computing



- Virtualization abstracts physical hardware into software-defined, isolated environments



- The foundation of cloud computing: resource pooling, multi-tenancy, on-demand scaling



- Types: hypervisor/server, OS/container, network, storage, desktop, and application



- Audience: developers and technical teams building and operating cloud workloads



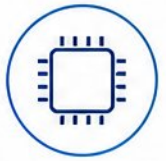
Why Virtualization Matters for Cloud Workloads



- Decouples software from hardware, enabling isolated environments on one physical machine



- Benefits: higher hardware utilization, lower costs, faster provisioning, and improved disaster recovery



- The hypervisor allocates CPU, memory, storage, and network to virtual machines



- Virtualization is the technology; cloud computing is the service model built on it



The Core Abstraction Layer: Hypervisors



- **Hypervisor:** software layer creating and managing virtual machines



- **Type 1:** runs directly on hardware, lower latency and higher density



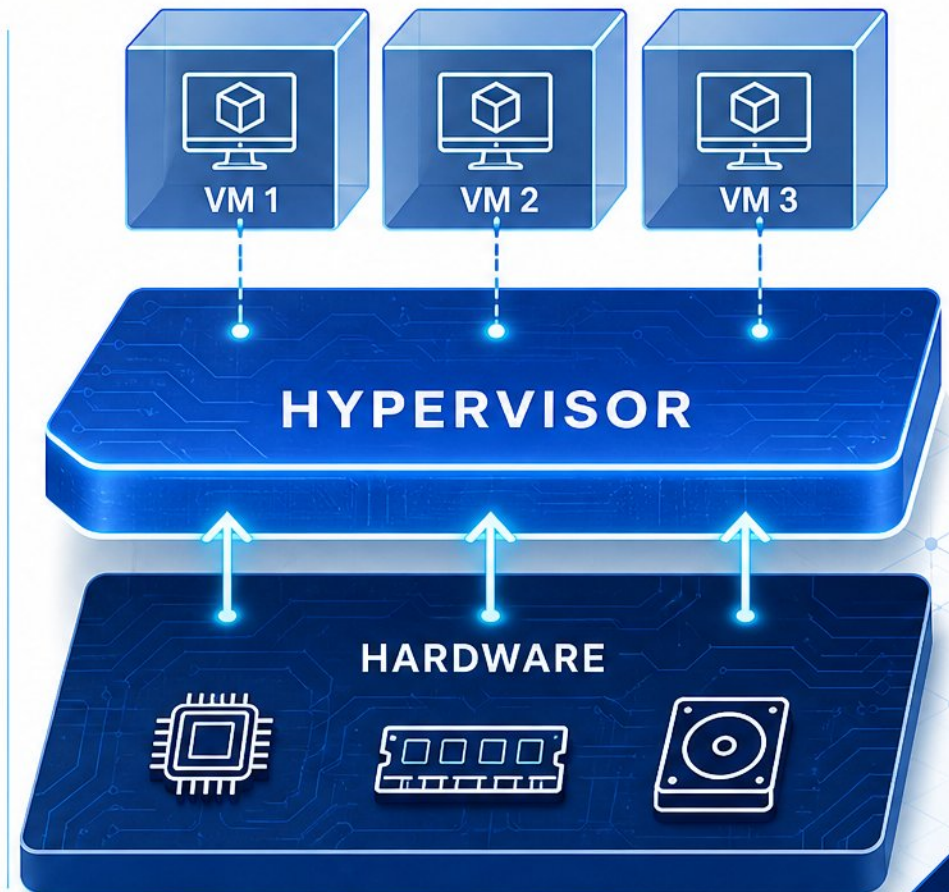
- **Type 2:** runs on host OS, convenient for development and testing



- **Common hypervisors:** KVM, VMware ESXi, Hyper-V, Xen



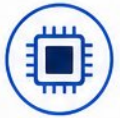
- **Modern CPU overhead:** only 3–15%;
VirtIO boosts I/O performance



Server Virtualization and Hardware Abstraction



- Partitions one physical server into multiple isolated VMs



- Full virtualization emulates complete hardware



- Paravirtualization uses a hypervisor-aware API for faster I/O



- Hardware-assisted virtualization uses Intel VT-x/AMD-V extensions



- Enables resource pooling, live migration, snapshots, high availability



Use cases: legacy consolidation, diverse OS support, strong tenant isolation



OS-Level Virtualization and Containers



- Virtualize the OS by sharing the host kernel



- Isolate user spaces with namespaces and cgroups



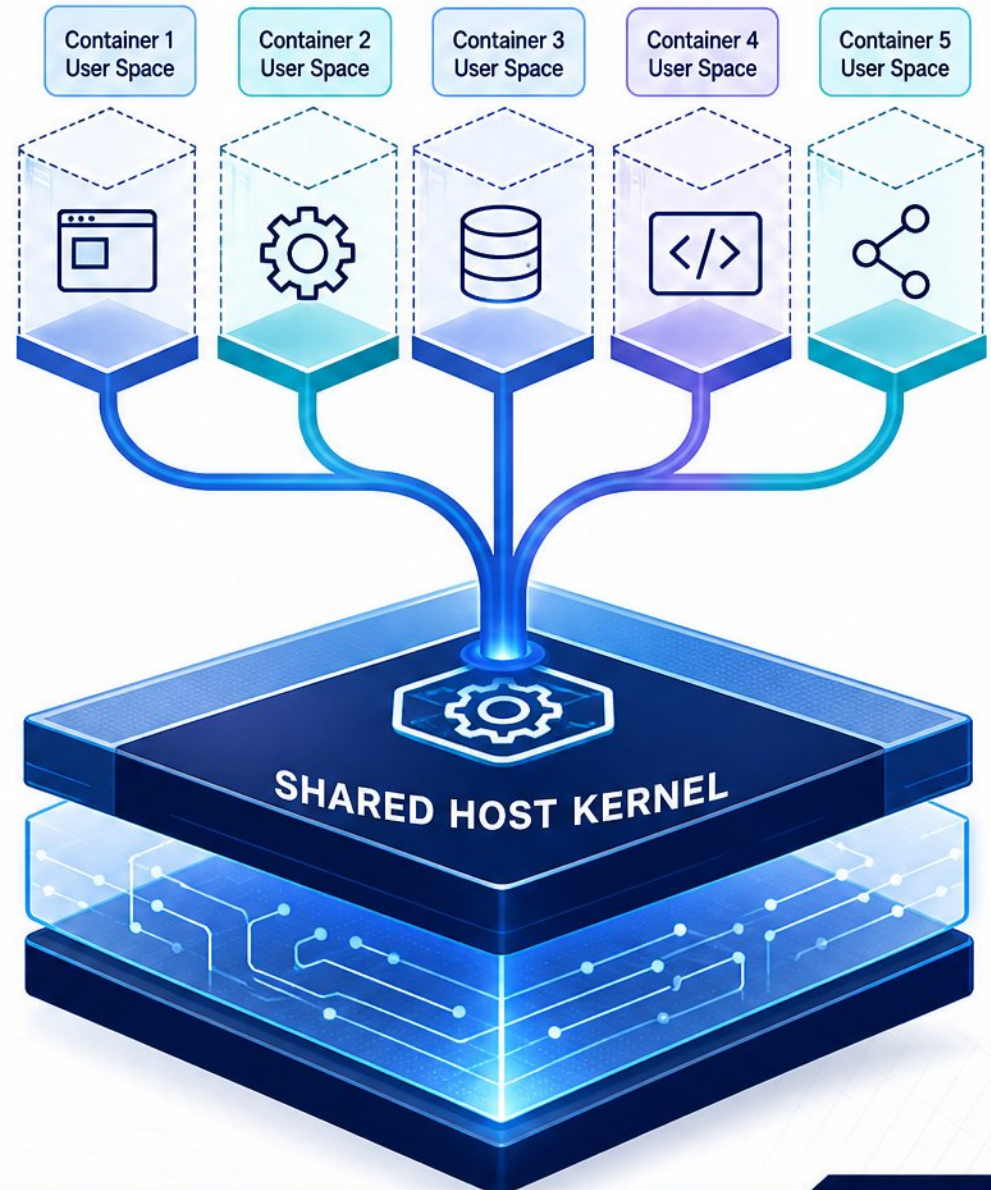
- Lightweight (MB-scale) and start in seconds



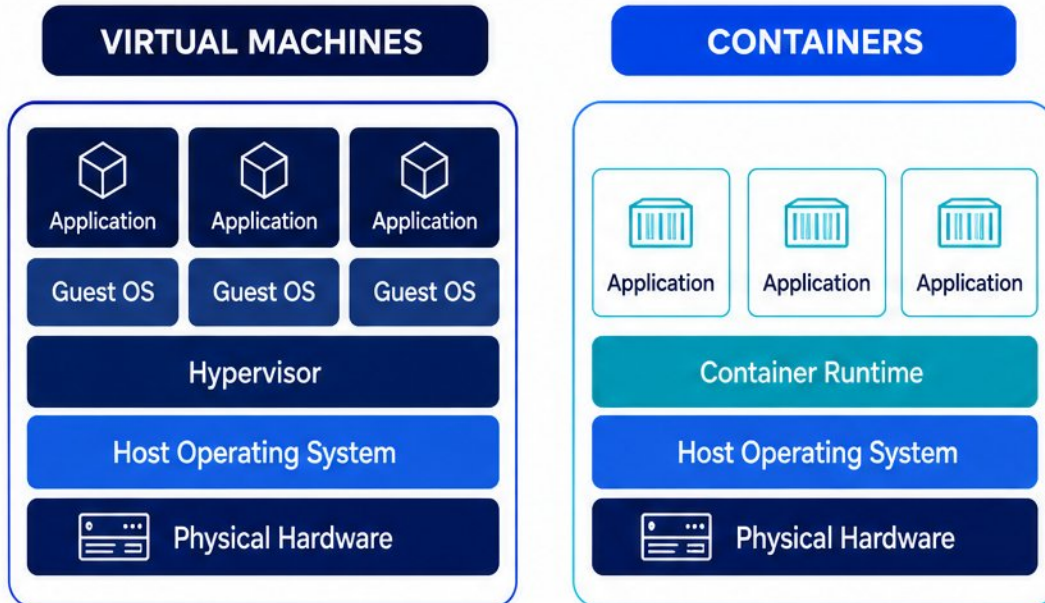
- Enable high application density for cloud-native apps



- Run via Docker, containerd, Podman, Kubernetes



Containers vs. Virtual Machines: When They Overlap



Larger Resource Footprint



Smaller Resource Footprint

- VMs: hardware isolation, full OS flexibility
- Containers: speed, portability, efficient resource use
- Production reality: Kubernetes workers typically run on VMs
- Serverless platforms use microVMs like AWS Firecracker
- MicroVMs offer VM-grade isolation with container-grade speed
- Best results: combine VMs for infrastructure, containers for apps

Network Virtualization in Cloud Infrastructure



- Abstracts physical networks into software-defined overlays



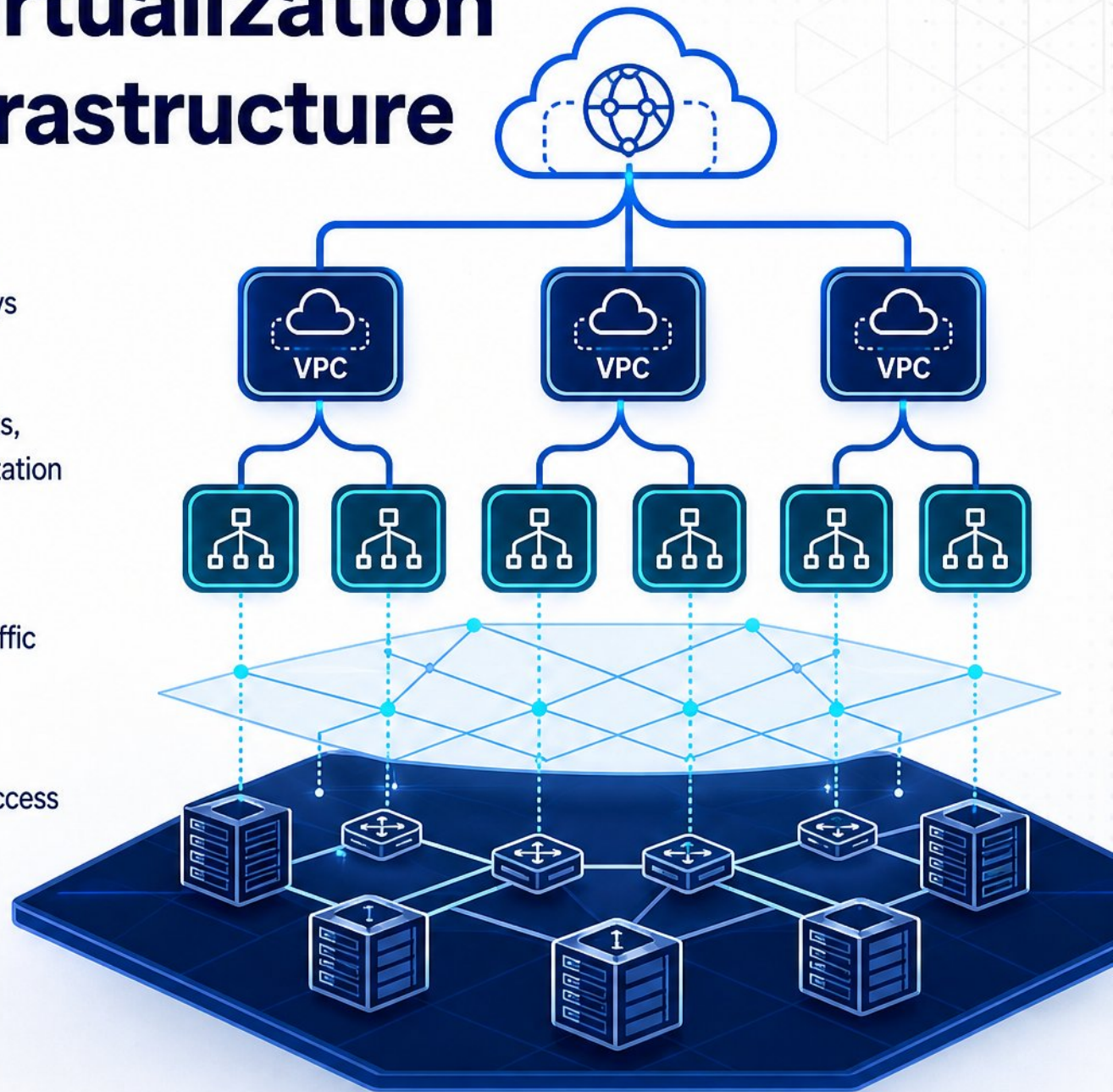
- Core concepts: VPCs, subnets, VLANs, SDN, micro-segmentation



- Managed virtual networks isolate tenants and control traffic

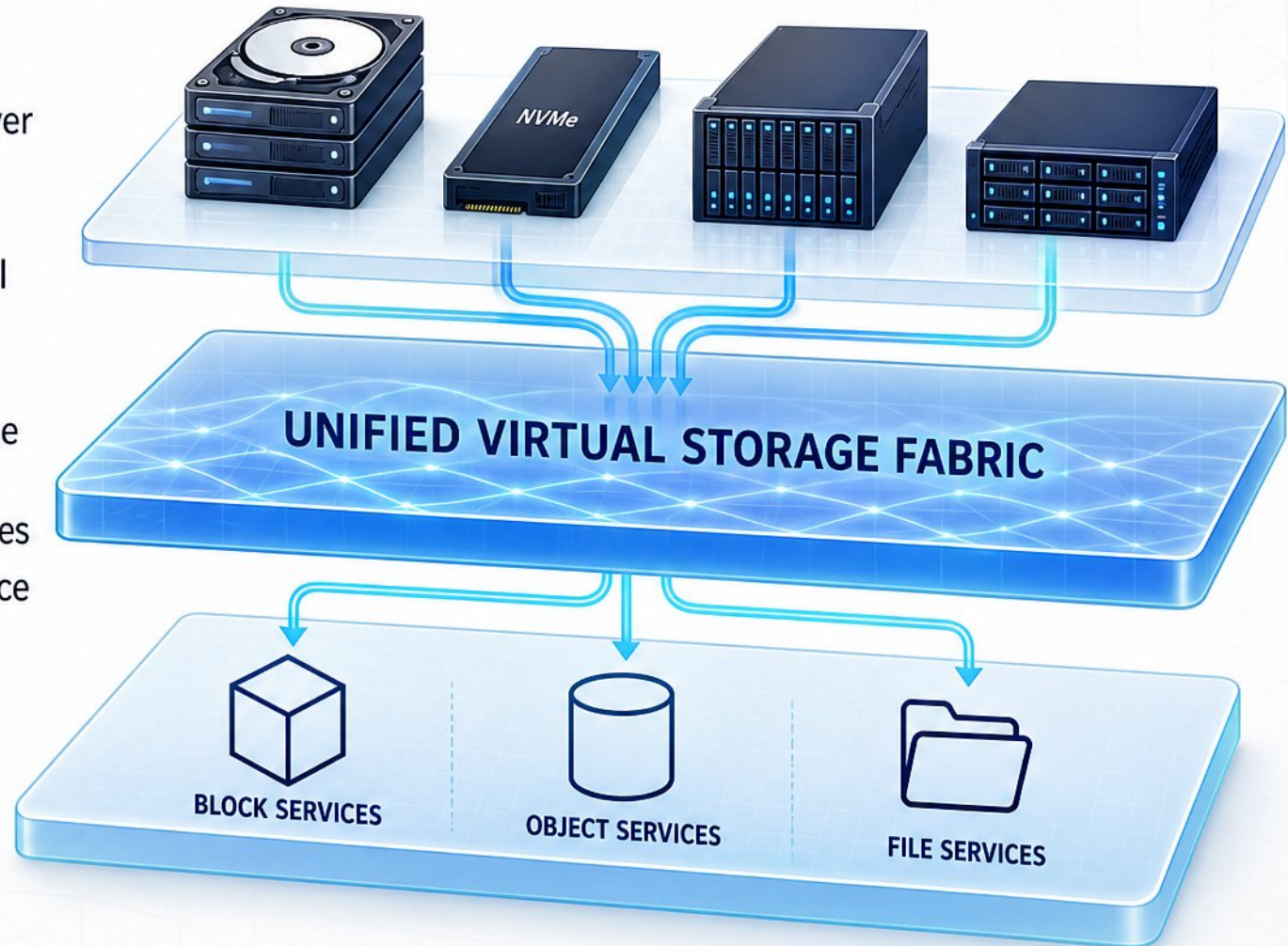


- Essential for multi-tenant security and least-privilege access



Storage Virtualization and Software-Defined Storage

- Pools physical storage into a unified, centrally managed layer
- Approaches: SAN, NAS, software-defined storage, HCI
- Cloud block, object, and file services built on virtual storage
- Native NVMe/TCP disaggregates storage for scalable performance



Desktop and Application Virtualization



- Centralizes desktops and apps on servers, accessed remotely



- Key models: VDI, hosted shared desktops, app streaming



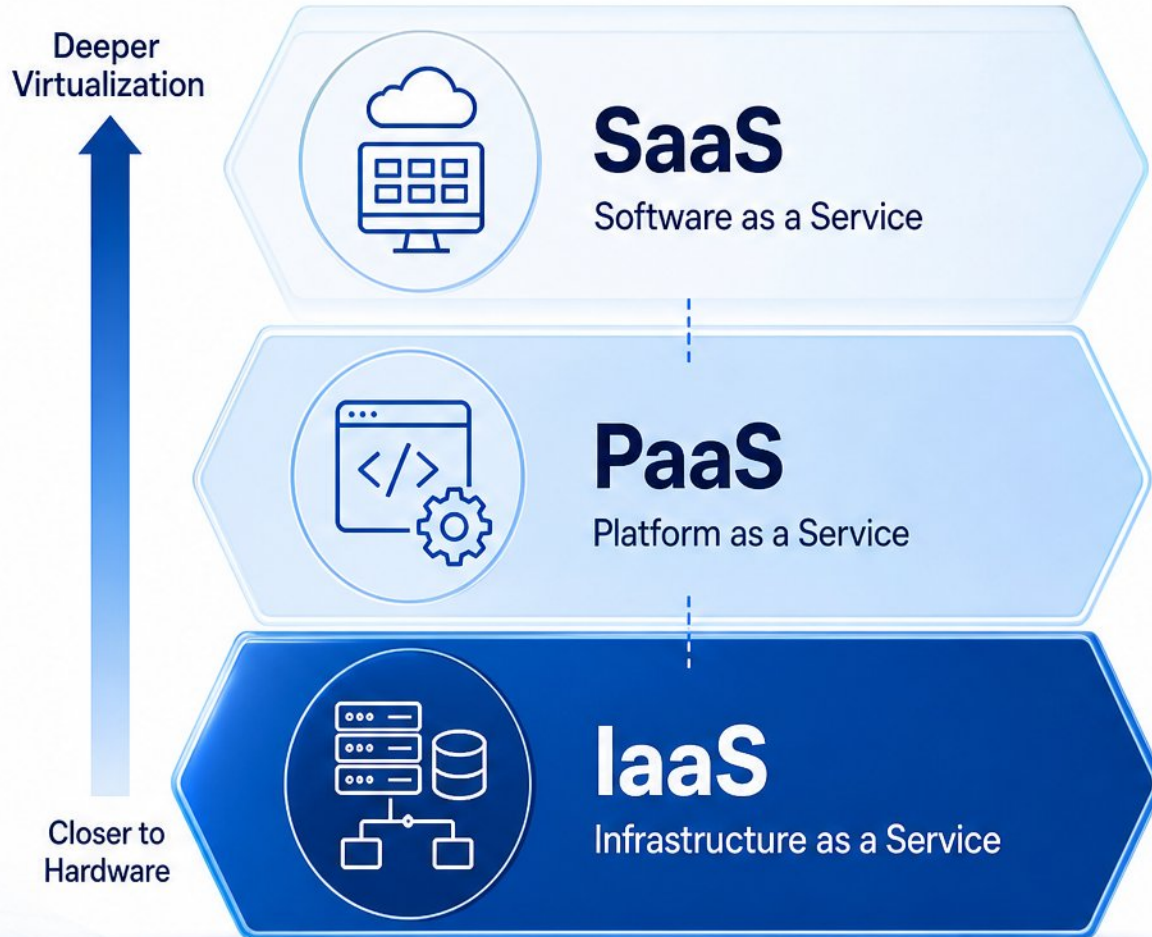
- Use cases: remote work, secure access, legacy apps



- Key design factors: security, licensing, user experience



Mapping Virtualization to IaaS, PaaS, and SaaS



- IaaS virtualizes servers, storage, and networks
- PaaS abstracts OS and runtime for developers
- SaaS shifts most operational burden to providers
- AWS Nitro, Azure Boost, and Google Titanium secure tenants

Tenant Isolation and the Shared Responsibility Model



- **Multi-layer isolation:**
hypervisor, network segmentation, storage separation, and resource governance



- **Noisy neighbor problem:**
resource spikes impact other tenants; CPU reservations and QoS controls mitigate it



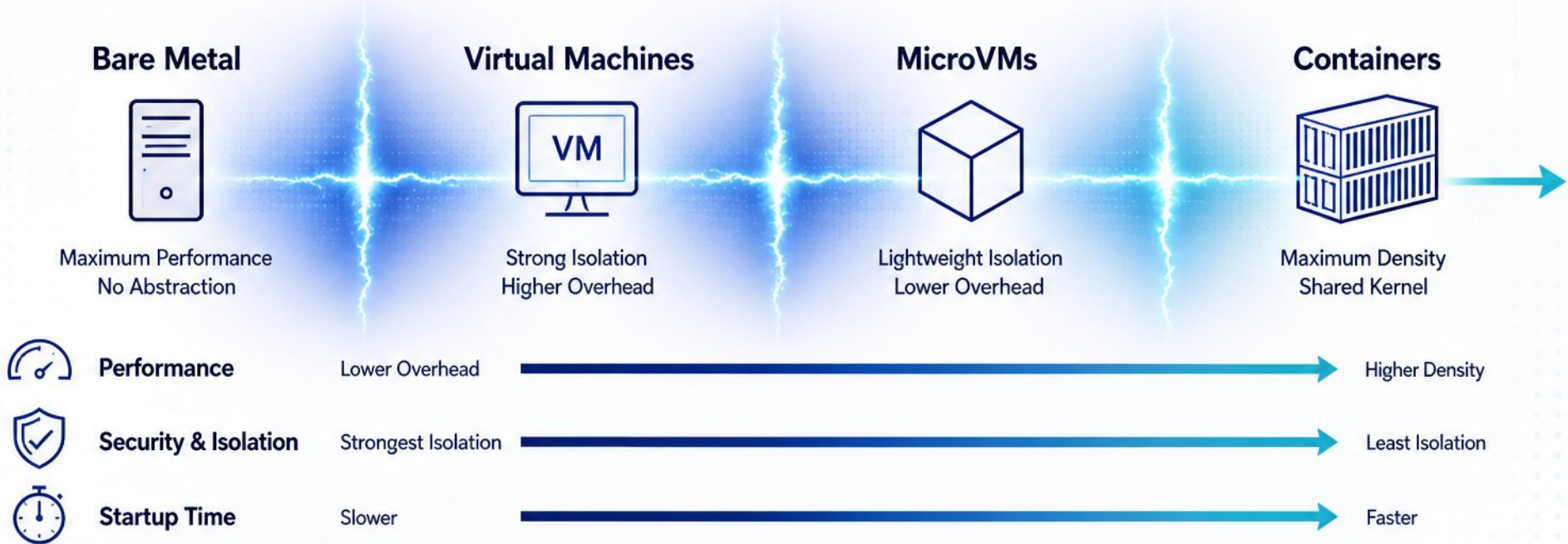
- **Shared responsibility:**
providers secure the cloud; customers secure workloads in the cloud



- **Compliance-heavy workloads**
may require full-stack or account-per-tenant isolation



Performance, Security, and Isolation Trade-offs



- Hypervisor overhead is modest for most workloads; I/O-heavy tasks may still favor bare metal.



- VMs provide hardware-level isolation; containers share the host kernel.

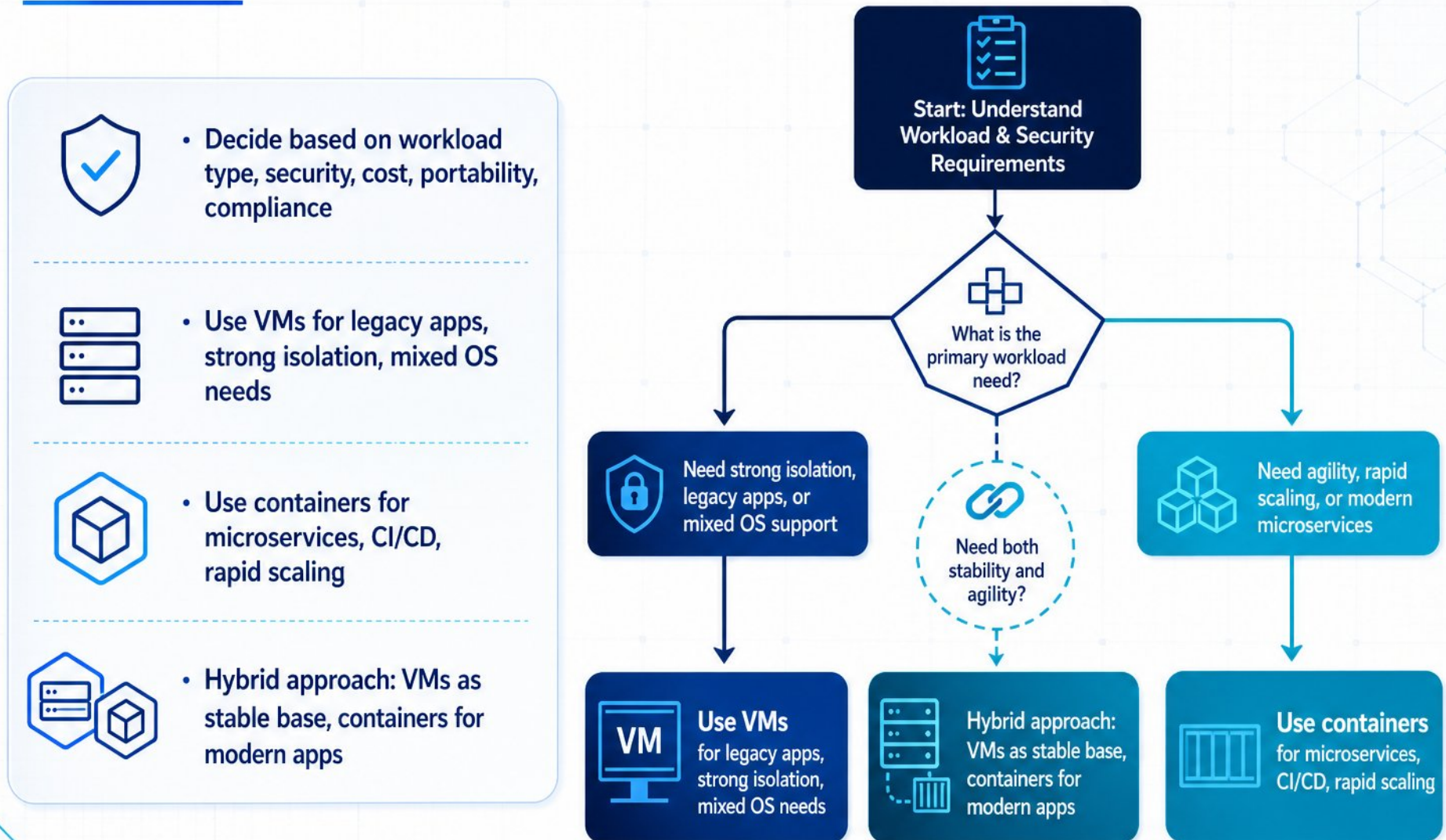


- Best practices: define isolation levels, use microsegmentation, scan images, enforce quotas.



- Containers start in under a second, VMs in minutes, microVMs in 100-300 ms.

Choosing the Right Virtualization Approach



Hands-On Practice and Next Steps



- **Lab:** create a free-tier Linux VM, install a web server, secure it, then tear it down



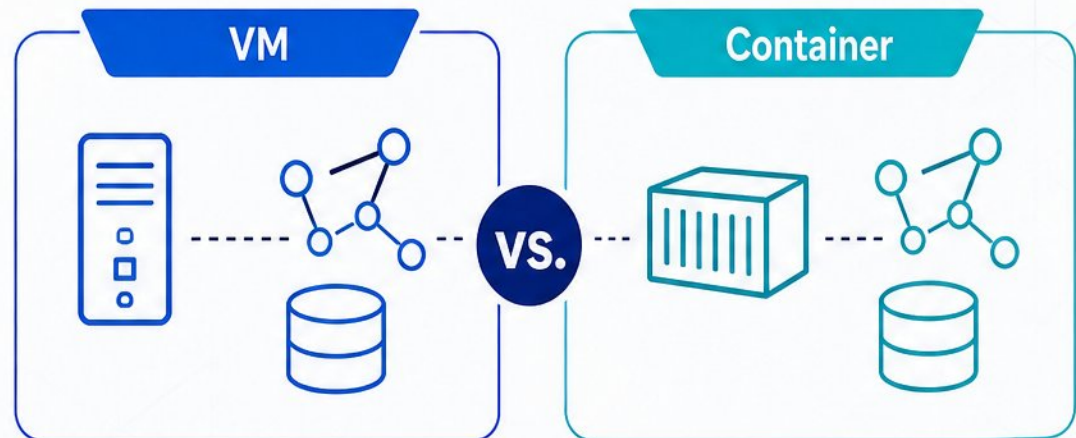
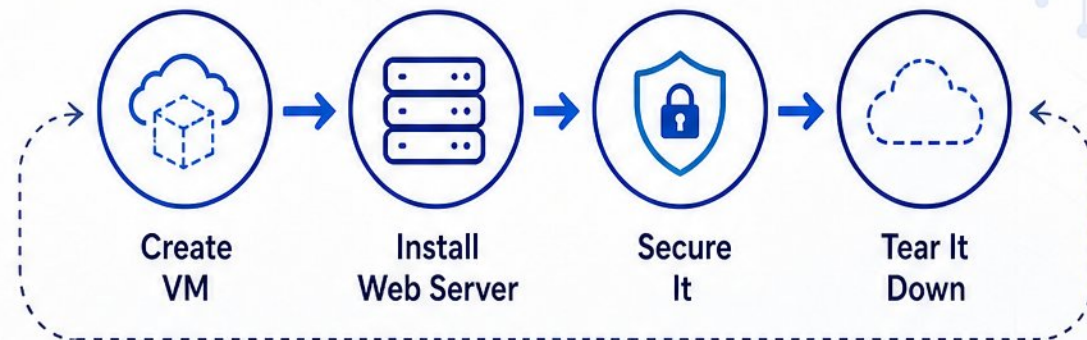
- **Extend** with Docker; compare VM vs. container networking and storage



- **Free options:** Oracle Cloud Always Free, Google Cloud e2-micro, AWS credits, Microsoft Learn sandboxes



- **Next:** cloud certifications, Kubernetes fundamentals, deeper network and storage study



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/711674bc/public