# Department of Computer Science

**SE-315: Cloud Computing**

**Class: BSCS Fall 2025**

# 

**Lab 04: Virtualization Technology**

|  |  |
| --- | --- |
| **Lab Engineer:** | Engr. Muhammad Abid Hussain |
| **Student Name:** |  |
| **CMS ID:** |  |
| **Department:** | BS Computer Science |
| **Semester** | 5th semester |
| **Date** | 01 October, 2025 |

|  |  |
| --- | --- |
| **Lab Module 04** | **Virtualization Technology** |
| **Purpose** |  To introduce students to **virtualization technology** as a core concept of modern cloud computing.   To demonstrate how a single **physical machine can host multiple independent environments** known as Virtual Machines (VMs).   To explain the **role of hypervisors** in enabling virtualization.   To differentiate between **Type-1 and Type-2 hypervisors** and their practical applications.   To explore the use of virtualization in both **personal computing** (e.g., VirtualBox, VMware) and **enterprise-level cloud infrastructure** (e.g., KVM, VMware ESXi). |
| **Objectives** | By the end of this lab, students should be able to:   * Define virtualization and explain its importance in cloud computing. * Differentiate between **Type-1** and **Type-2 hypervisors**. * Install and configure a **virtual machine** using VirtualBox or VMware. * Understand the role of **KVM (Kernel-based Virtual Machine)** as a Type-1 hypervisor. * Compare the performance, security, and use cases of different hypervisors. |
| **Procedure** | 1. The instructor will begin with an introduction to **virtualization** and explain the importance of hypervisors in cloud computing. 2. Students will install and configure a **Type-2 hypervisor** (VirtualBox or VMware Workstation) and create their first virtual machine. 3. The instructor will demonstrate the installation of a **Type-1 hypervisor (KVM)** on a Linux system, and students will observe and document the process. 4. Students will record the differences between Type-1 and Type-2 hypervisors in a comparison table. |

**Background Theory**

**1. What is Virtualization?**

Virtualization is the process of creating a **virtual version of hardware, operating systems, storage, or networks**. Instead of running directly on physical hardware, resources are abstracted and shared across multiple environments.

* For example, a computer with 16 GB RAM and 8 CPU cores can be partitioned to run four different operating systems, each with 4 GB RAM and 2 CPU cores.

Virtualization makes computing environments more **efficient, cost-effective, scalable, and flexible**, which is why it is the backbone of cloud computing.

**2. Hypervisors**

A **hypervisor** is the software (or firmware) responsible for managing virtual machines. It allocates CPU, memory, and storage resources from the physical host to different VMs, ensuring isolation between them.

There are two main types:

**Type-1 Hypervisor (Bare-Metal)**

* Runs **directly on physical hardware** without requiring a host operating system.
* Provides **high performance, scalability, and security**.
* Common in enterprise and data center environments.
* **Examples:** VMware ESXi, Microsoft Hyper-V, KVM, Xen.

**Type-2 Hypervisor (Hosted)**

* Runs **on top of an existing operating system** (Windows, Linux, or macOS).
* Easier to install and commonly used for labs, testing, and small-scale environments.
* Slightly less efficient than Type-1 due to dependency on the host OS.
* **Examples:** Oracle VirtualBox, VMware Workstation, Parallels Desktop.

**3. Importance of Virtualization in Cloud Computing**

Virtualization enables:

* **Resource Sharing:** Multiple virtual servers share the same physical hardware.
* **Scalability and Flexibility:** Resources (CPU, RAM, storage) can be allocated or adjusted dynamically.
* **Cost Reduction:** Organizations avoid purchasing separate hardware for every workload.
* **Isolation and Security:** Each VM runs independently, preventing crashes or security breaches in one VM from affecting others.
* **Testing and Development:** Developers can test applications in multiple environments on a single machine.

**Procedure**

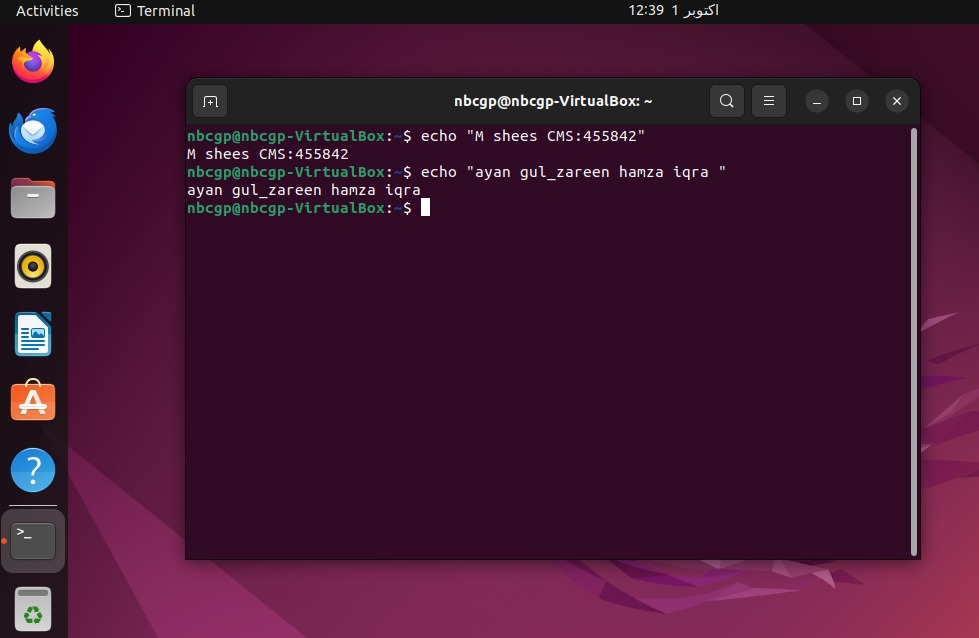
**Pre-Lab Preparation**

* Students should study the concepts of **virtualization** and **hypervisors** before coming to the lab.
* Pre-lab assignment: Write a short note comparing **Type-1 and Type-2 hypervisors** with at least two examples of each.

**In-Lab Tasks**

**Task 4.1 – Virtual Machine Setup (Type-2 Hypervisor)**

1. Install **Oracle VirtualBox** or **VMware Workstation Player** on your system.
2. Create a new **Virtual Machine (VM)** and install an operating system such as **Ubuntu Linux** or **Windows 10**.
3. Allocate resources (e.g., 2 CPU cores, 4 GB RAM, and 20 GB disk).
4. Boot the VM and verify that the OS is working correctly.
5. Take a **screenshot** of the running VM to include in your lab report.



**Task 4.2 – Install a Type-1 Hypervisor (KVM)**

1. On a Linux machine (Ubuntu recommended), update packages:

sudo apt update

sudo apt upgrade -y

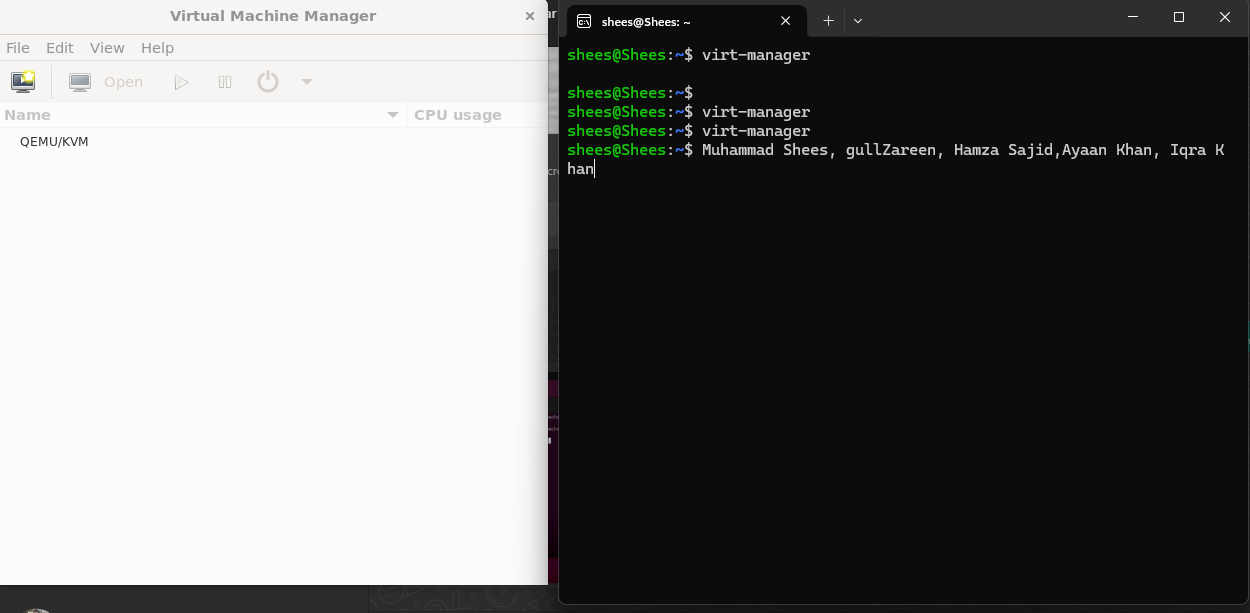
1. Install **KVM and supporting packages**:

sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager -y

1. Verify installation:

kvm-ok

1. Launch **Virtual Machine Manager (virt-manager)** and create a new VM.
2. Document the setup process with screenshots.



**Task 4.3 – Compare Type-1 and Type-2 Hypervisors**

* Prepare a table comparing:
  + Installation process
  + Performance
  + Security
  + Common use cases
* Discuss which type is more suitable for **cloud service providers** versus **students and individual developers**.

| **Feature** | **Type-1 Hypervisor (Bare-metal)** | **Type-2 Hypervisor (Hosted)** |
| --- | --- | --- |
| **Installation process** | Installed **directly on the hardware**. Requires dedicated machine (e.g., KVM, VMware ESXi, Microsoft Hyper-V). | Installed **on top of an existing OS** (e.g., VirtualBox, VMware Workstation). Easier to set up. |
| **Performance** | High performance (close to native hardware speed) since no underlying OS layer. | Lower performance due to overhead from the host operating system. |
| **Security** | More secure: minimal attack surface since it doesn’t rely on a host OS. | Less secure: relies on the host OS, so vulnerabilities in OS affect VMs. |
| **Common use cases** | Cloud computing platforms, enterprise data centers, production environments. | Personal use, testing, learning, software development, running multiple OS on a desktop/laptop. |

**For Cloud Service Providers (CSPs):**  
Type-1 hypervisors are more suitable because they provide **high performance, scalability, and stronger security**. Cloud platforms like **AWS, Azure, and Google Cloud** use Type-1 hypervisors to manage thousands of virtual machines efficiently.

**For Students and Individual Developers:**  
Type-2 hypervisors are better because they are **easy to install, run inside an existing OS, and don’t require a dedicated machine**. Tools like **VirtualBox** or **VMware Workstation** allow students to practice virtualization, networking, and OS concepts conveniently.

**Post-Lab Work**

* Students should compile observations into their lab reports, including screenshots and the comparison table.
* Students must complete the **home assignment** before the next lab.

**Observations Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Tool Used** | **Observation** | **Screenshot Attached** |
| 4.1 | VirtualBox / VMware | VM created and booted | Yes/No |
| 4.2 | KVM (virt-manager) | VM launched successfully | Yes/No |
| 4.3 | Comparison Table | Differences documented clearly | Yes/No |

**Results**

* Students successfully created and managed a virtual machine using a **Type-2 hypervisor** (VirtualBox/VMware).
* Students observed the installation and functionality of a **Type-1 hypervisor (KVM)** on Linux.
* A detailed comparison between Type-1 and Type-2 hypervisors was made, highlighting their strengths, weaknesses, and use cases.

**Conclusion**

This lab provided students with a practical understanding of **virtualization** and **hypervisors**, which are essential to cloud computing. By setting up VMs in VirtualBox (Type-2) and exploring KVM (Type-1), students gained insight into how virtualization works at both personal and enterprise scales. The experiment reinforced virtualization’s role in **resource sharing, cost efficiency, scalability, and system isolation** in modern IT environments.

**Home Task / Assignment**

1. Prepare a **detailed comparison table** between **VMware ESXi (Type-1)** and **VirtualBox (Type-2)** covering performance, cost, scalability, and usability.

| **Feature** | **VMware ESXi (Type-1)** | **Oracle VirtualBox (Type-2)** |
| --- | --- | --- |
| **Performance** | Runs directly on hardware with near-native performance. Optimized for high workloads, low latency, and resource-intensive applications. | Runs on top of a host OS. Performance is lower due to extra overhead from the host operating system. Best for light workloads. |
| **Cost** | Paid enterprise solution (VMware vSphere/ESXi licenses). Some free versions exist (with limited features). | Completely free and open-source (community edition). No licensing cost. |
| **Scalability** | Highly scalable – supports large-scale enterprise environments, data centers, and cloud infrastructure. Can manage **hundreds to thousands of VMs**. | Limited scalability – suitable for running a few VMs on desktops/laptops. Not designed for large production environments. |
| **Usability** | Requires technical expertise to install and configure. Management is done via **vSphere Client / vCenter**. More complex for beginners. | Very user-friendly GUI. Easy to install on Windows, Linux, and macOS. Ideal for students, testers, and developers. |
| **Security** | More secure since it runs directly on bare metal, with fewer layers. Enterprise-grade security features like VM isolation, role-based access, and patching. | Relies on the host OS, so security depends on host stability. Less secure for sensitive workloads. |
| **Use Cases** | Data centers, cloud service providers, enterprise IT environments, production servers. | Learning virtualization, testing operating systems, software development, experimenting in labs. |

1. Draw a **diagram** showing the architecture of Type-1 and Type-2 hypervisors, labeling hardware, hypervisor, operating systems, and applications.
2. Write a one-page note on **how virtualization enables scalability in cloud computing**.

**How Virtualization Enables Scalability in Cloud Computing**

Virtualization is one of the core technologies that makes cloud computing possible. It is the process of creating virtual versions of computing resources such as servers, storage, and networks. By abstracting physical hardware into multiple virtual machines (VMs), virtualization allows efficient resource utilization, flexibility, and on-demand scalability.

Scalability in cloud computing refers to the ability of a system to handle increasing workloads by dynamically allocating resources. Virtualization enables this scalability in several ways:

1. **Resource Pooling and Elasticity**: Virtualization aggregates hardware resources into pools that can be shared across multiple users. This means when demand increases, more virtual machines or containers can be provisioned instantly, and when demand decreases, resources can be scaled down. This elasticity ensures cost efficiency and optimal performance.
2. **Dynamic Resource Allocation**: Hypervisors (software that manages virtualization) allow cloud providers to allocate CPU, memory, and storage to VMs based on real-time needs. Applications that require more resources can be scaled up without physically changing hardware, supporting both vertical (adding more resources to a VM) and horizontal (adding more VMs) scaling.
3. **Isolation and Flexibility**: Each VM runs independently, ensuring that scaling one workload does not interfere with others. This isolation allows cloud providers to run multiple applications on the same physical server, making it easier to scale services for different customers simultaneously.
4. **Rapid Deployment and Automation**: Virtualization supports automated provisioning of VMs and containers through orchestration tools. This makes it possible to scale services quickly to meet spikes in demand, such as during online sales, streaming events, or data analysis workloads.

In summary, virtualization provides the foundation for scalable cloud services by enabling efficient use of hardware, dynamic resource allocation, and rapid elasticity. Without virtualization, cloud computing would lack the flexibility and responsiveness that organizations rely on to adapt to changing demands.

1. Research and give a real-world example where hybrid **use of Type-1 and Type-2 hypervisors** might be beneficial.

**Real-World Example of Hybrid Use of Type-1 and Type-2 Hypervisors**

A hybrid use of Type-1 and Type-2 hypervisors can be beneficial in environments where both **production-grade performance** and **development/testing flexibility** are required.

* **Type-1 Hypervisors** (e.g., VMware ESXi, Microsoft Hyper-V, KVM) run directly on server hardware. They provide high performance, stability, and scalability, making them suitable for enterprise-level production workloads.
* **Type-2 Hypervisors** (e.g., Oracle VirtualBox, VMware Workstation) run on top of a host operating system, offering ease of use, portability, and convenience for developers or testers.

**Example: Software Development and Deployment in a Financial Institution**

In a financial organization, mission-critical applications (such as online banking systems, transaction processing, and fraud detection engines) need **high performance, availability, and security**. These workloads run on a **Type-1 hypervisor** like VMware ESXi or KVM in the production data center.

At the same time, the institution’s software development team needs to build, test, and debug new versions of these applications. Instead of working directly on production servers, developers use **Type-2 hypervisors** like VirtualBox or VMware Workstation on their laptops/desktops. This allows them to:

* Run multiple test environments locally (different OS versions, database setups).
* Experiment with configurations without affecting production.
* Quickly clone, pause, or roll back VMs during testing.

Once the software is tested and stable, it is deployed onto the **production environment managed by the Type-1 hypervisor** for performance and reliability.

**Why This Hybrid Approach is Beneficial**

* **Cost efficiency**: Developers don’t need dedicated servers; they use their own systems with Type-2 hypervisors.
* **Separation of concerns**: Development/testing is isolated from mission-critical production.
* **Scalability and stability**: Production workloads benefit from Type-1 performance, while Type-2 supports rapid prototyping.
* **Flexibility**: Teams can replicate production-like environments on desktops before final deployment.

**Practical Tasks**

1. **Multi-VM Networking Experiment**
   * Create **two virtual machines** in VirtualBox (e.g., Ubuntu + Ubuntu, or Ubuntu + Windows).
   * Configure them on the same **internal network** in VirtualBox.
   * Test communication by using:

ping <IP\_of\_second\_VM>

* + Take screenshots showing successful communication.
  + Write a note on how this simulates **networking in cloud environments**.

1. **Shared Folder Setup (Host ↔ VM Communication)**
   * Configure a **shared folder** between your host OS and a virtual machine.
   * Copy a file (e.g., a text file or image) from host → VM and VM → host.
   * Demonstrate that virtualization supports **resource sharing** between environments.

1. **Snapshot and Rollback**
   * Create a VM and take a **snapshot** (saved state).
   * Make changes inside the VM (e.g., install software, create files).
   * Roll back to the snapshot and verify changes are gone.
   * Explain how snapshots are useful in **testing and cloud disaster recovery**.
2. **Resource Allocation Experiment**
   * Create a VM with **1 CPU and 1 GB RAM**. Run a benchmark or open multiple apps and record performance.
   * Modify the VM to **2 CPUs and 2 GB RAM**. Re-run the same workload.
   * Compare results and explain how resource allocation impacts **scalability in cloud systems**.
3. **Nested Virtualization (Optional – for advanced students)**
   * Enable nested virtualization (if supported by CPU).
   * Run **VirtualBox inside a VirtualBox VM** or install Docker inside your VM.
   * Write about how this simulates a “cloud inside a cloud” scenario.

### Deliverables

Compile a single Word document by filling in the solution section and submit this file on Google Classroom/MS Teams. Insert all answers/solutions in the document. You must also include the implementation of the tasks in the design tool within the same document to get your work graded. In case of any issues with submission on Google Classroom/MS Teams, you may submit your Lab assignments by emailing them to: [lab.engrcs@nbc.nust.edu.pk](mailto:lab.engrcs@nbc.nust.edu.pk)