**Batch: A-3 Roll No.: 16010122104**

**Experiment No. 8**

**Grade: AA / AB / BB / BC / CC / CD /DD**



**Objective:**

To investigate system virtualization by deploying and managing a Windows virtual machine on AWS EC2, thereby demonstrating its role in enabling scalable, secure cloud infrastructure.

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **CO** | **Outcome** |
| **CO2** | Investigate the system virtualization and outline its role in enabling the cloud computing System model |

# Books/ Journals/ Websites referred

* Mateos, A., & Müller, J. (2018). *Cloud Computing: Concepts, Technology & Architecture*. Wiley.
* Barham, P., et al. (2003). “Xen and the Art of Virtualization.” *ACM Symposium on Operating Systems Principles*.
* AWS Documentation: Amazon EC2 User Guide (<https://docs.aws.amazon.com/ec2/>)

# Abstract

In this experiment, we explore system virtualization within the context of cloud computing by deploying a virtual machine (VM) on Amazon Web Services (AWS). We demonstrate the end-to-end process—from AWS account creation to VM launch and connection—highlighting how virtualization underpins the cloud service model. Our investigation emphasizes the benefits of virtualization, such as resource abstraction, elasticity, and multi-tenant isolation, and illustrates AWS EC2’s practical implementation of these concepts.

# Introduction

Cloud computing transforms how organizations consume IT resources, delivering compute, storage, and networking on demand. At the heart of cloud platforms lies **system virtualization**, which decouples hardware resources from operating systems and applications. Virtualization enables multiple isolated VMs to run concurrently on a single physical host, providing the flexibility, scalability, and cost-efficiency that modern enterprises require.

In this write-up, we:

* Introduce virtualization concepts and cloud service models (IaaS, PaaS, SaaS).
* Describe AWS EC2 as an exemplar of Infrastructure as a Service (IaaS). ● Walk through the practical steps for launching a Windows VM on AWS.

# Related Theory

## 1. System Virtualization

* **Definition:** System virtualization uses a software layer—the **hypervisor**—to abstract physical hardware and present virtualized compute, memory, storage, and network resources to multiple isolated guest operating systems.
* **Hypervisor Types:**
  + **Type 1 (Bare-Metal):** Runs directly on the host’s hardware (e.g., VMware ESXi, Microsoft Hyper-V, Xen). Provides high performance and strong isolation.
  + **Type 2 (Hosted):** Runs on top of a host operating system (e.g., Oracle VirtualBox, VMware Workstation). Easier to set up but incurs extra overhead from the host OS.

## 2. AWS EC2 and Virtual Machines

* **EC2 Overview:** Amazon Elastic Compute Cloud (EC2) is AWS’s IaaS offering that delivers resizable compute capacity in the cloud. EC2 lets us launch, stop, and manage **Instances**, which are virtual machines running on AWS’s global data centers.
* **Instance Components:**
  + **AMI (Amazon Machine Image):** A templated bundle containing an operating system, application server, and applications—used to launch new EC2 instances.
  + **Instance Type:** Defines virtual hardware—vCPUs, memory, storage options, and network performance. Examples include **t2.micro** (1 vCPU, 1 GiB RAM) for light workloads, **m5.large** (2 vCPUs, 8 GiB RAM) for general-purpose applications, and **c5.4xlarge** for compute-optimized tasks.
  + **EBS (Elastic Block Store):** Network-attached block storage volumes that persist independently from the life of an instance. Root volumes and additional data volumes attach to instances as virtual disks.
* **EC2 Virtualization Modes:**
  + **HVM (Hardware Virtual Machine):** Leverages processor virtualization extensions for near-native performance, required for enhanced networking and GPU instances.
  + **PV (Paravirtual):** Uses modified guest OS to interact cooperatively with the hypervisor; legacy mode with lower performance on modern hardware.

## 3. Cloud Service Models & EC2’s Role

* **IaaS (Infrastructure as a Service):**
  + EC2 exemplifies IaaS by providing virtualized compute resources with full control over the guest OS, middleware, and applications.
  + We manage instance lifecycle, OS patches, and application deployment, while AWS handles the underlying physical infrastructure, network fabric, and hypervisor.
* **PaaS and SaaS Contrast:**
  + In PaaS (e.g., AWS Elastic Beanstalk), the platform automates instance provisioning and OS management—developers focus on code.
  + In SaaS (e.g., Salesforce), end users only interact with the application via a web interface; no server or OS management is exposed.

## 4. Virtualization’s Benefits in AWS EC2

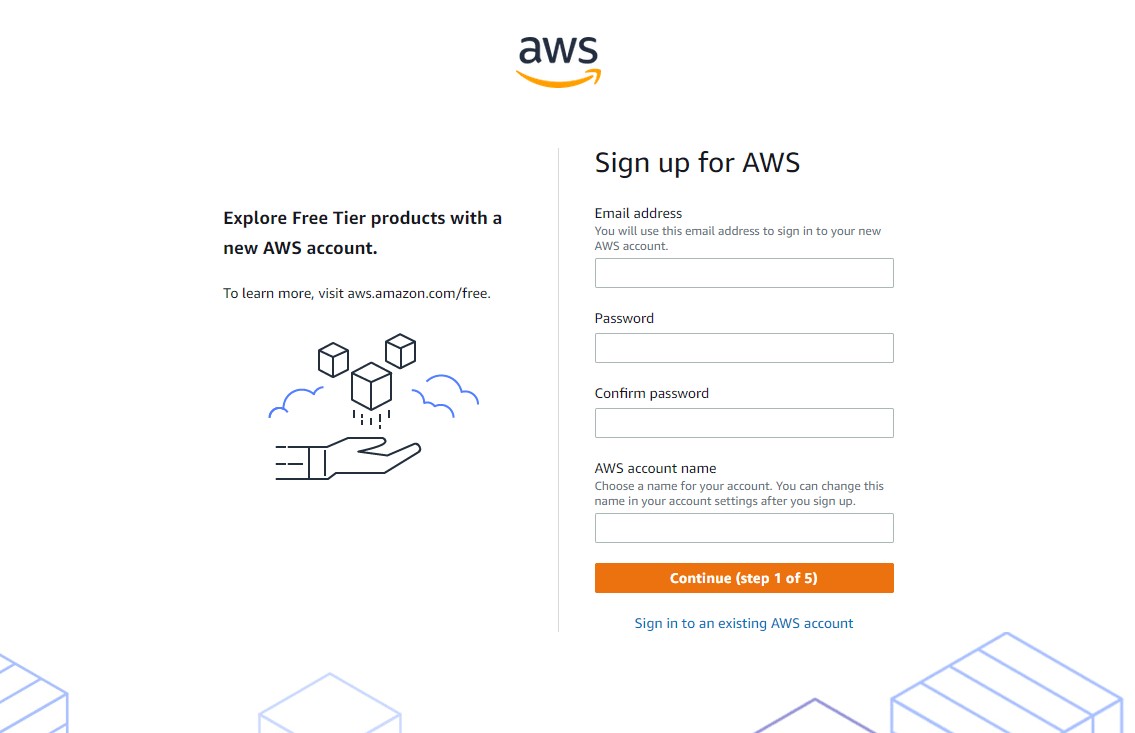
* **Elasticity:** Instantly scale capacity—spin up new instances or terminate idle ones to match demand.
* **Isolation & Security:** Each EC2 instance runs in its own virtual private cloud (VPC) with security groups acting as virtual firewalls. The hypervisor ensures strong tenant separation.
* **Portability & Reproducibility:** AMIs encapsulate complete OS and application stack configurations, enabling consistent environment replication across regions and accounts.
* **Cost Efficiency:** Pay-as-you-go billing on per-second (Linux) or per-hour

(Windows) basis, with additional options like Reserved Instances and Spot Instances to optimize costs.

# Implementation

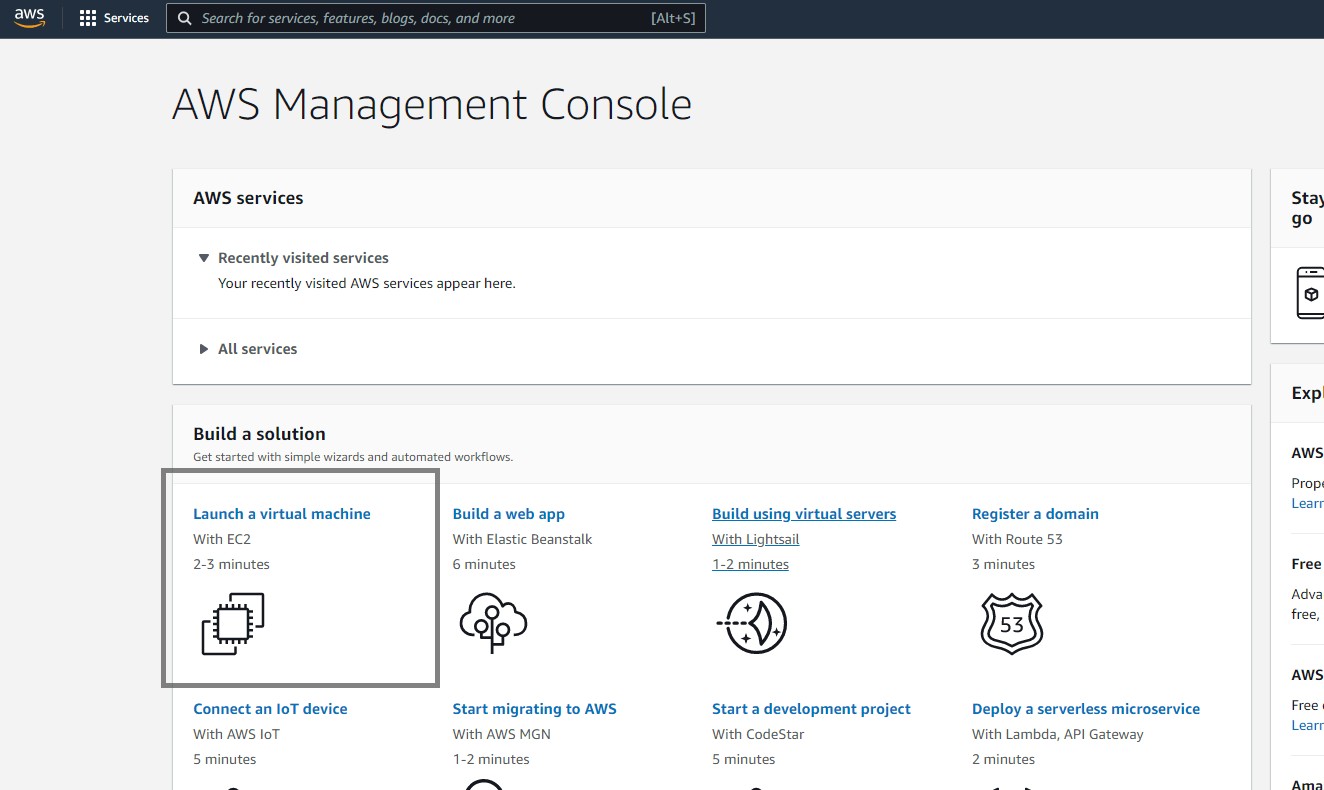
## 1. Create an AWS Account

* We navigate to the AWS Console signup page (<https://aws.amazon.com/>)  We click **Create an AWS Account** and follow the on-screen instructions.
* **Tip:** As new users, we receive 12 months of Free Tier usage.



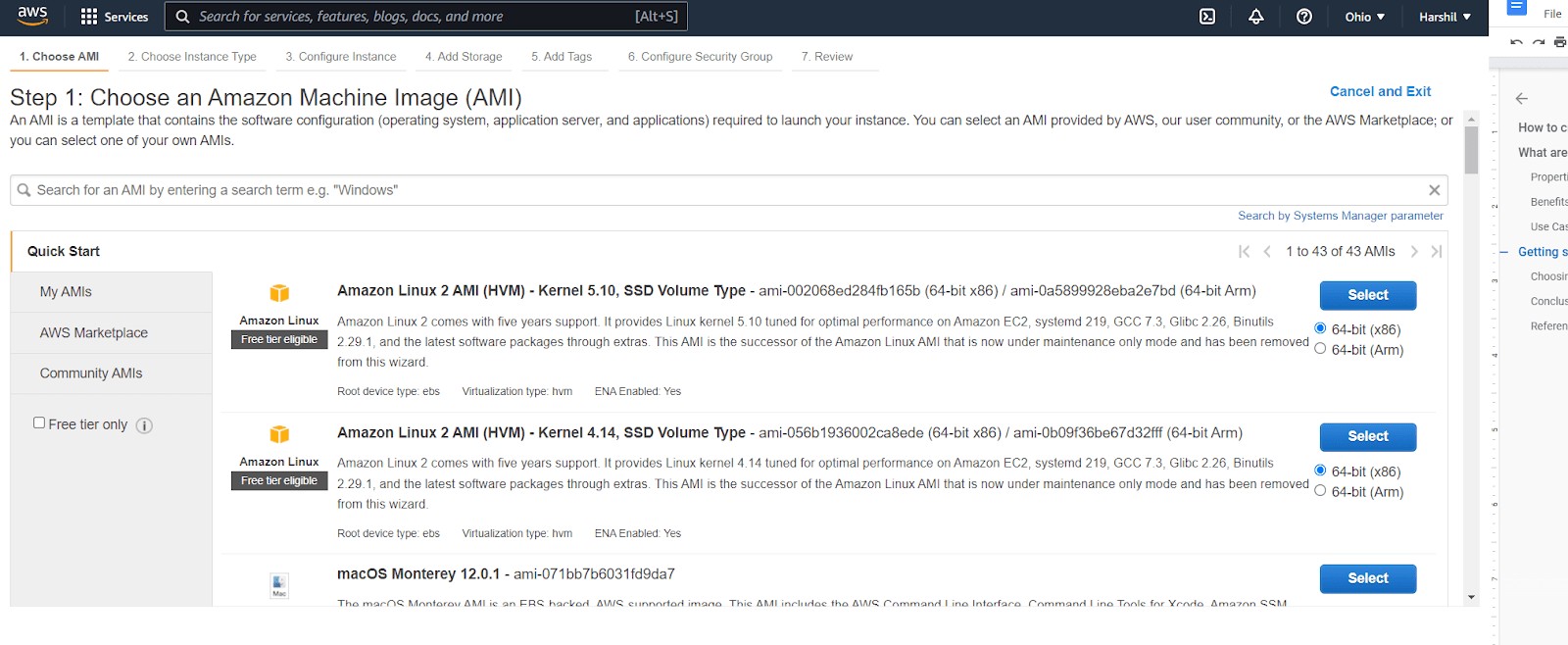
## 2. Launch AWS virtual machine

* We sign in to the AWS Management Console.
* We click the **AWS logo** (top-left) and select **EC2** under “Compute.”  On first use, we allow up to **24 hours** for full account activation.



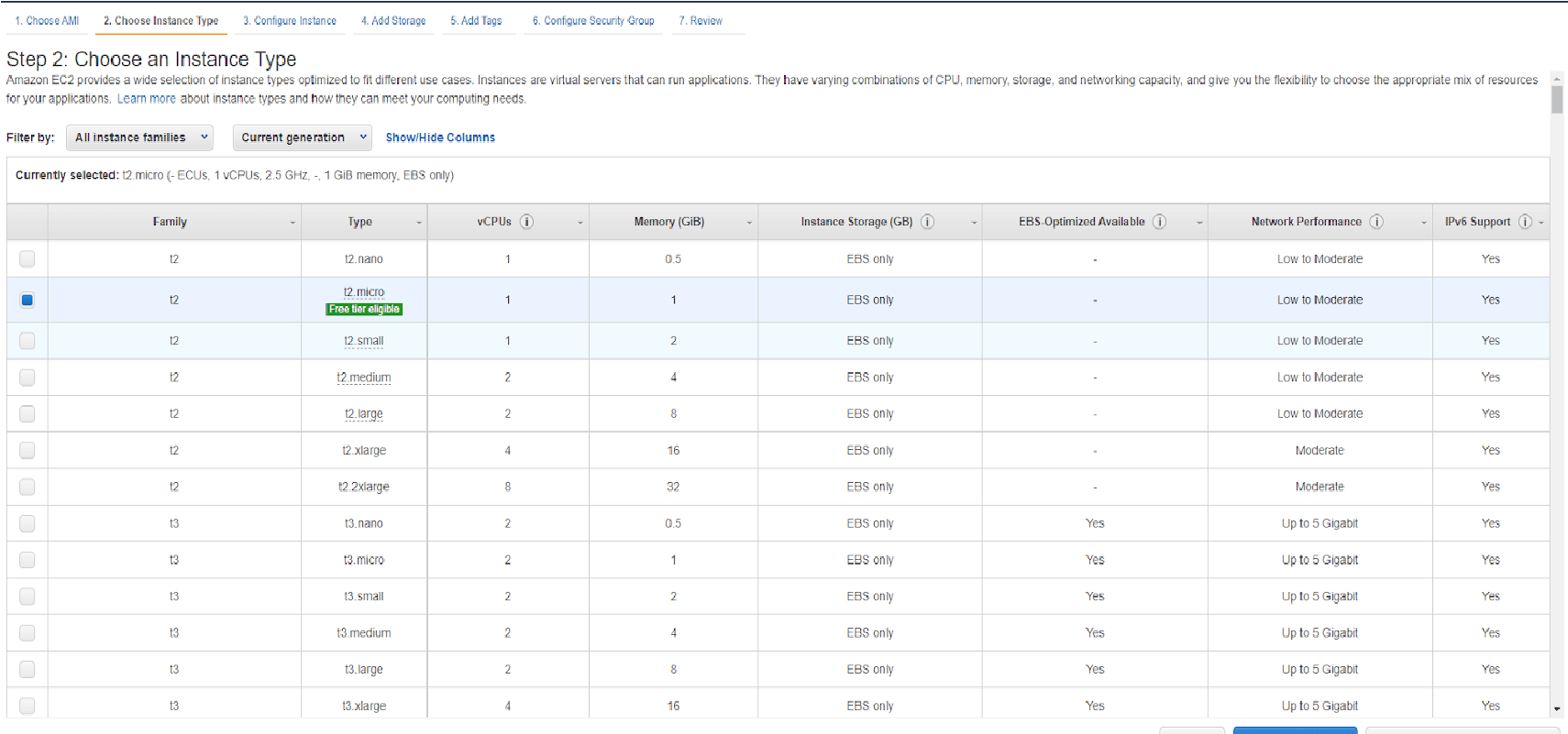
## 3. Choose AMI

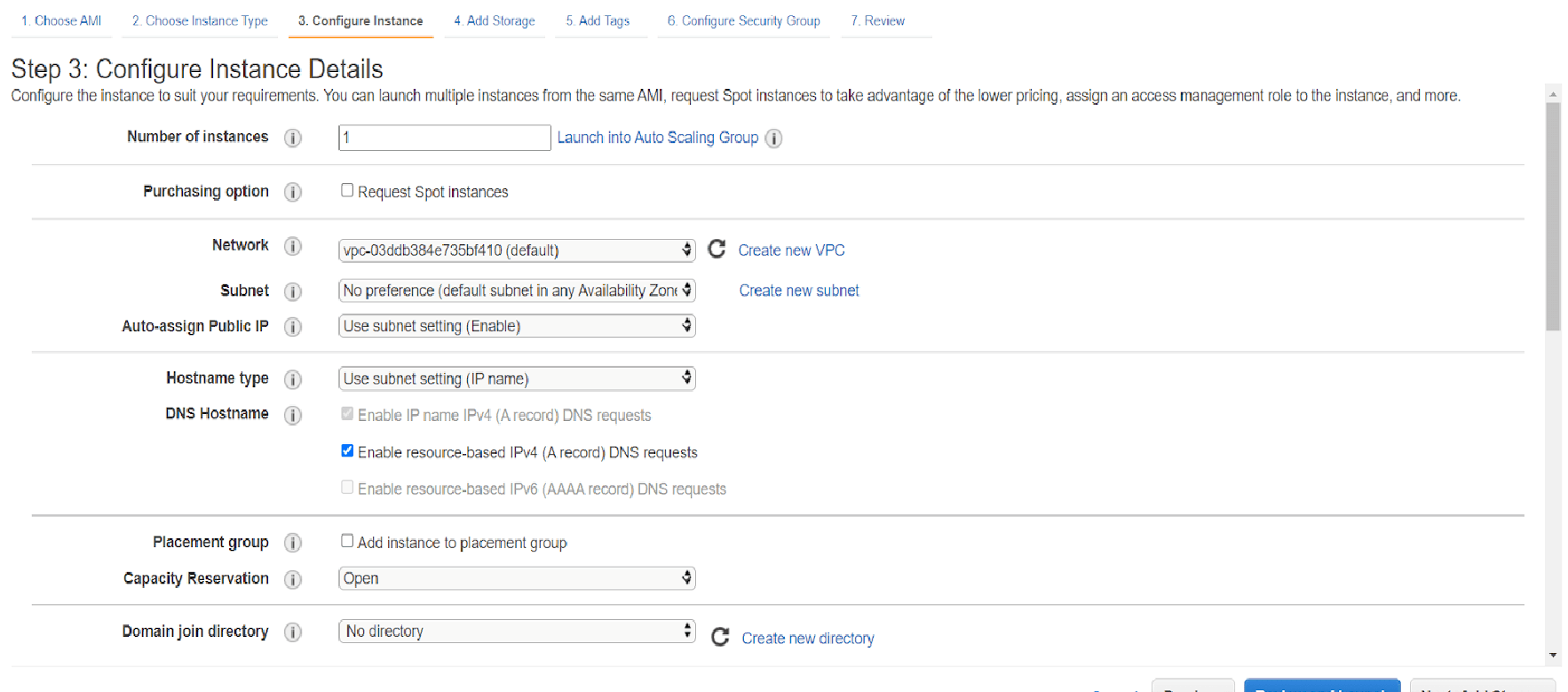
* In the EC2 dashboard, we click **Launch Instance**.
* We browse the list of AMIs—preconfigured templates bundling an OS and software.
* We select **Microsoft Windows Server** (or our preferred Windows version).



## 4. Choose and configure instance type

* We pick an instance type (e.g., **t2.micro** to stay within Free Tier).
* We click **Next: Configure Instance Details**.
* We leave defaults or adjust as needed:
  + Number of instances
  + VPC/subnet settings
  + IAM role (optional)





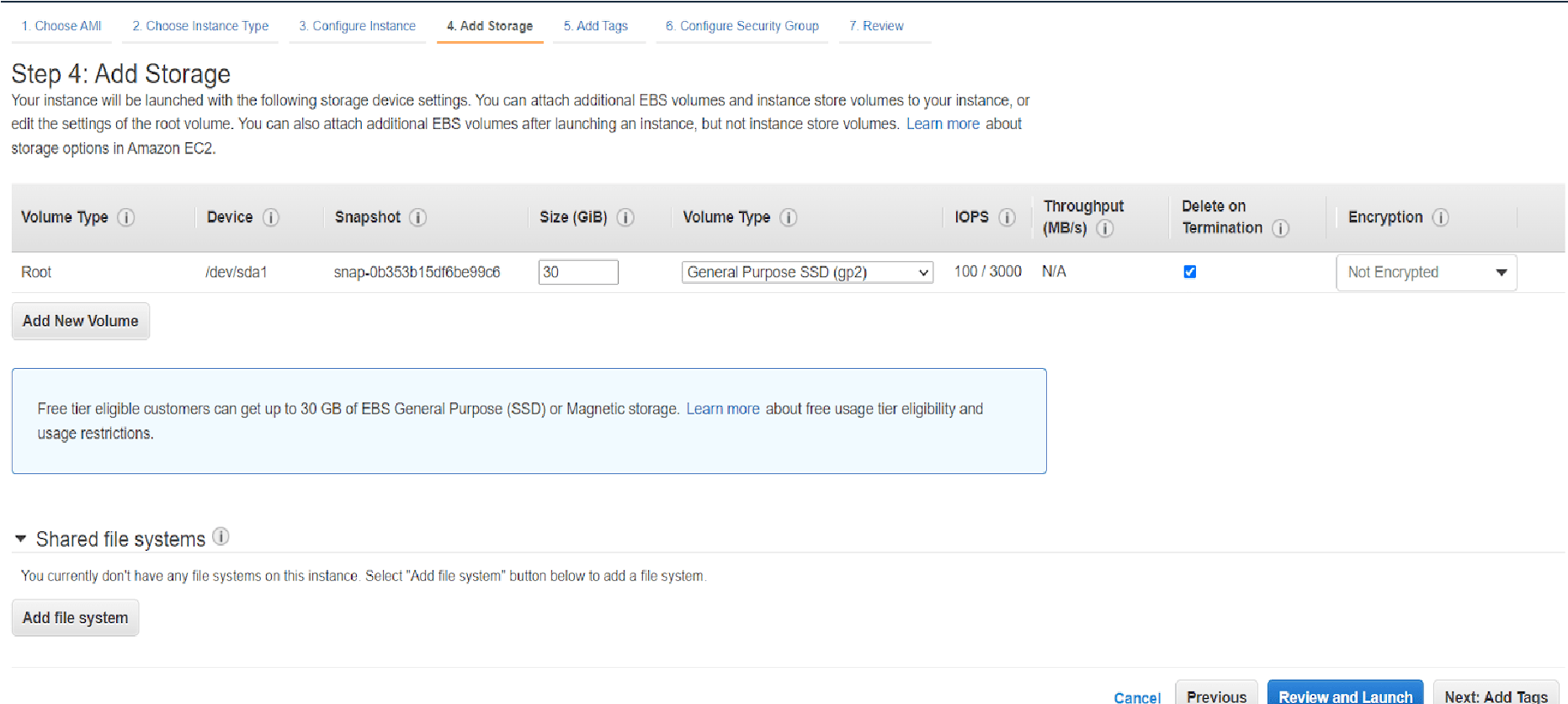
**5. Add storage and tags**

##  Add Storage

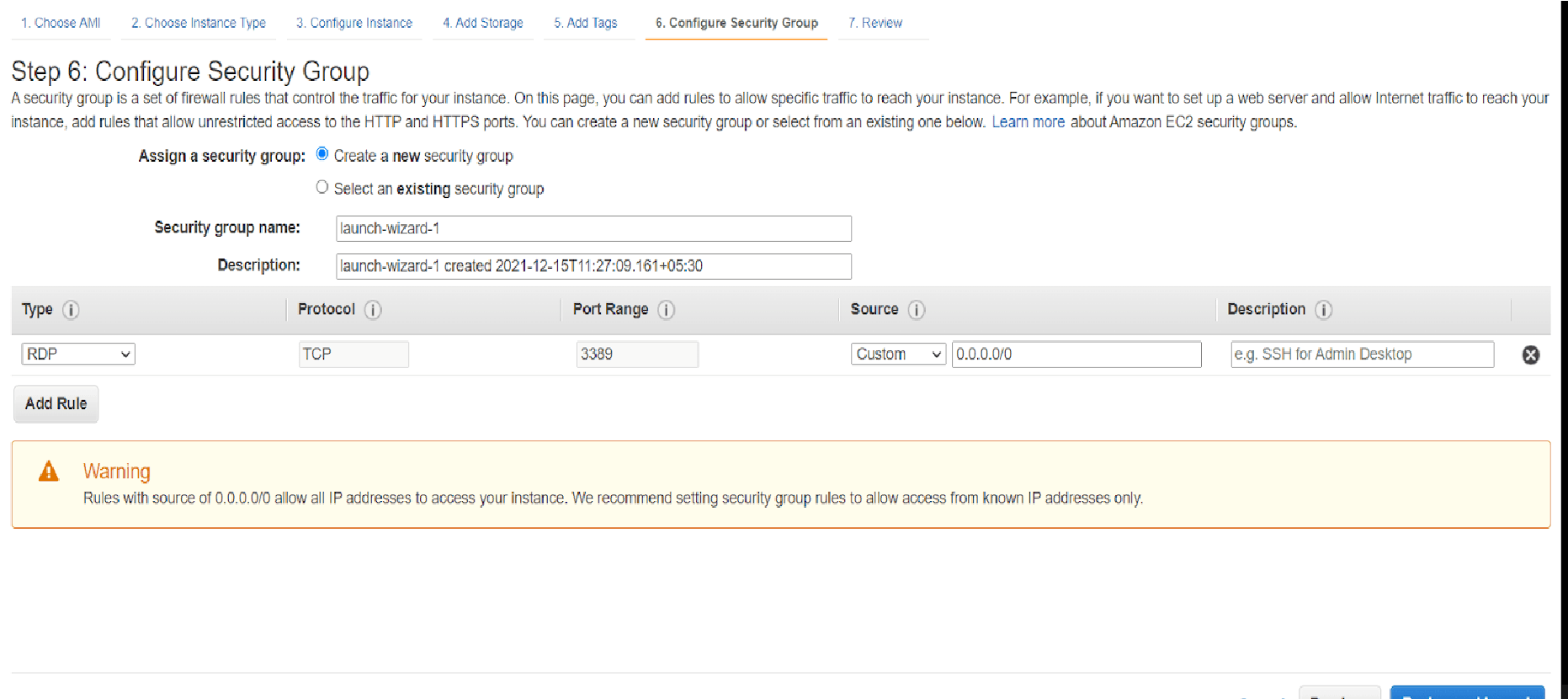
* We modify the root EBS volume’s size or type if required.
* We attach any additional EBS volumes for extra data capacity.

##  Add Tags

● We click **Add Tag**, then define a Key (e.g., Name) and Value (e.g., OurWindowsVM) to label the instance.



## 6. Configure security



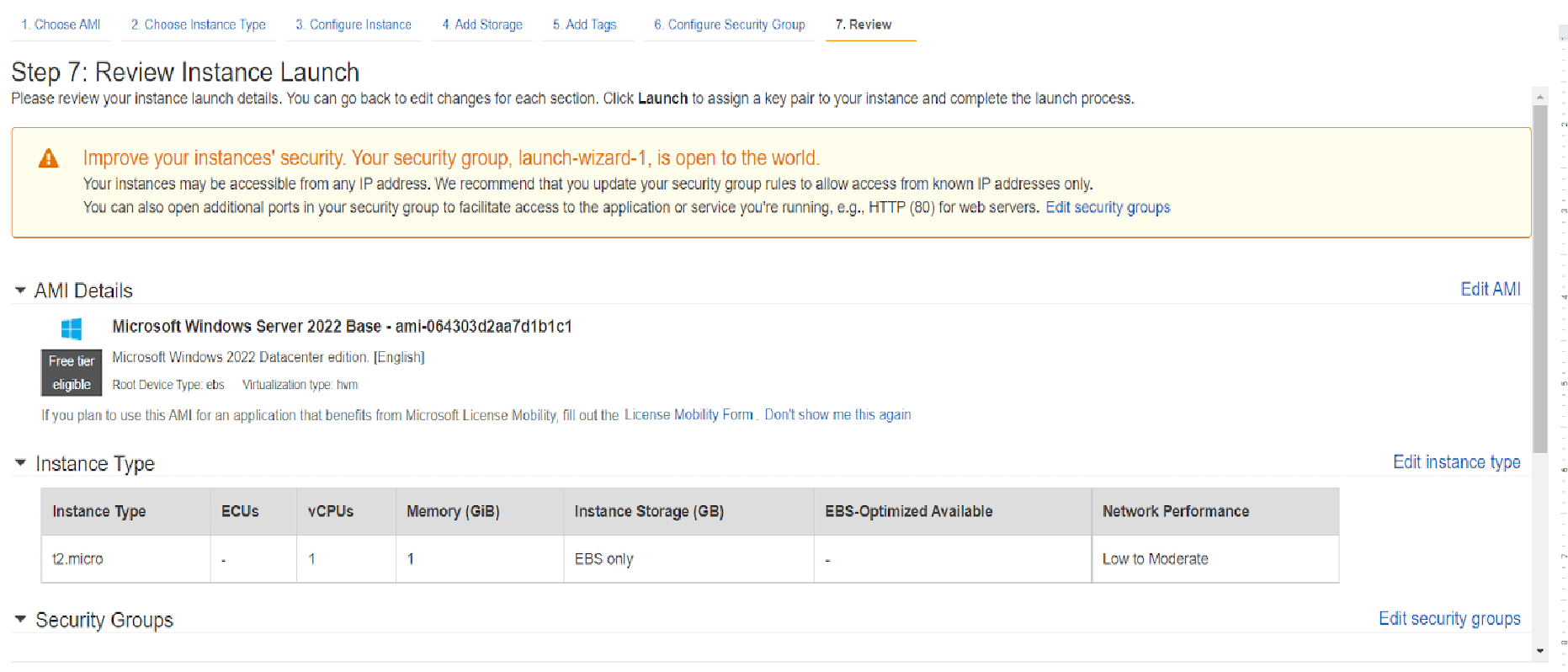
## 7. Review and launch your AWS virtual

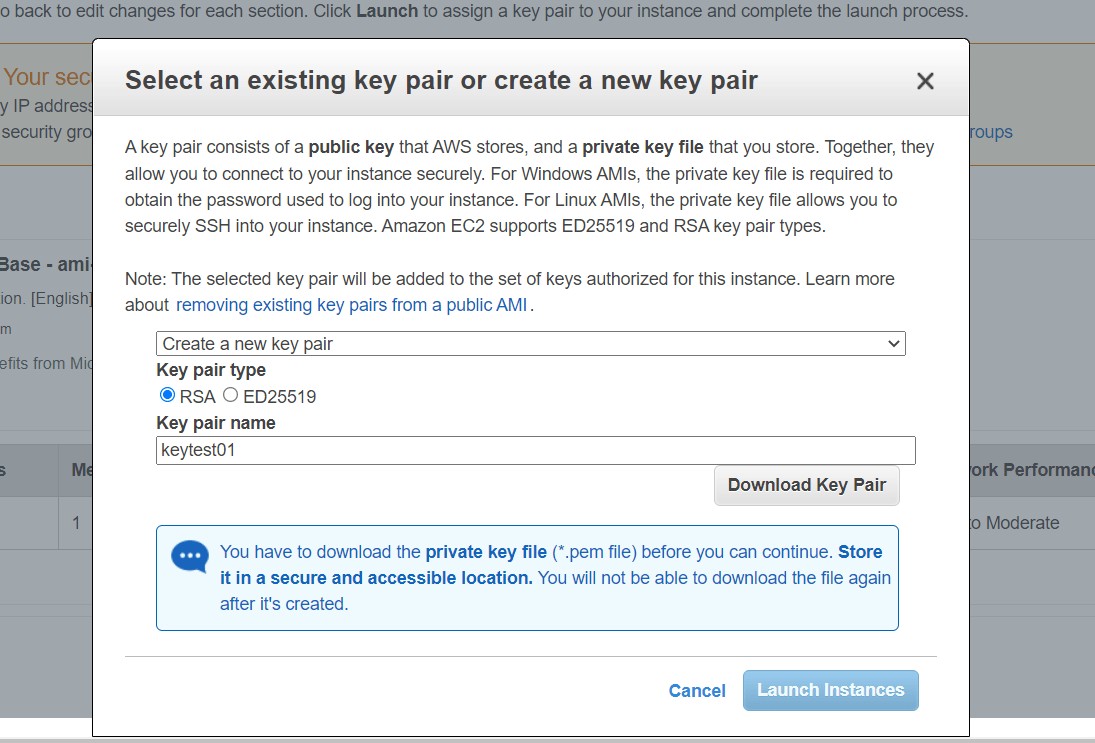
* We review every setting on the **Review Instance Launch** page.
* We click **Launch**.

##  Key Pair

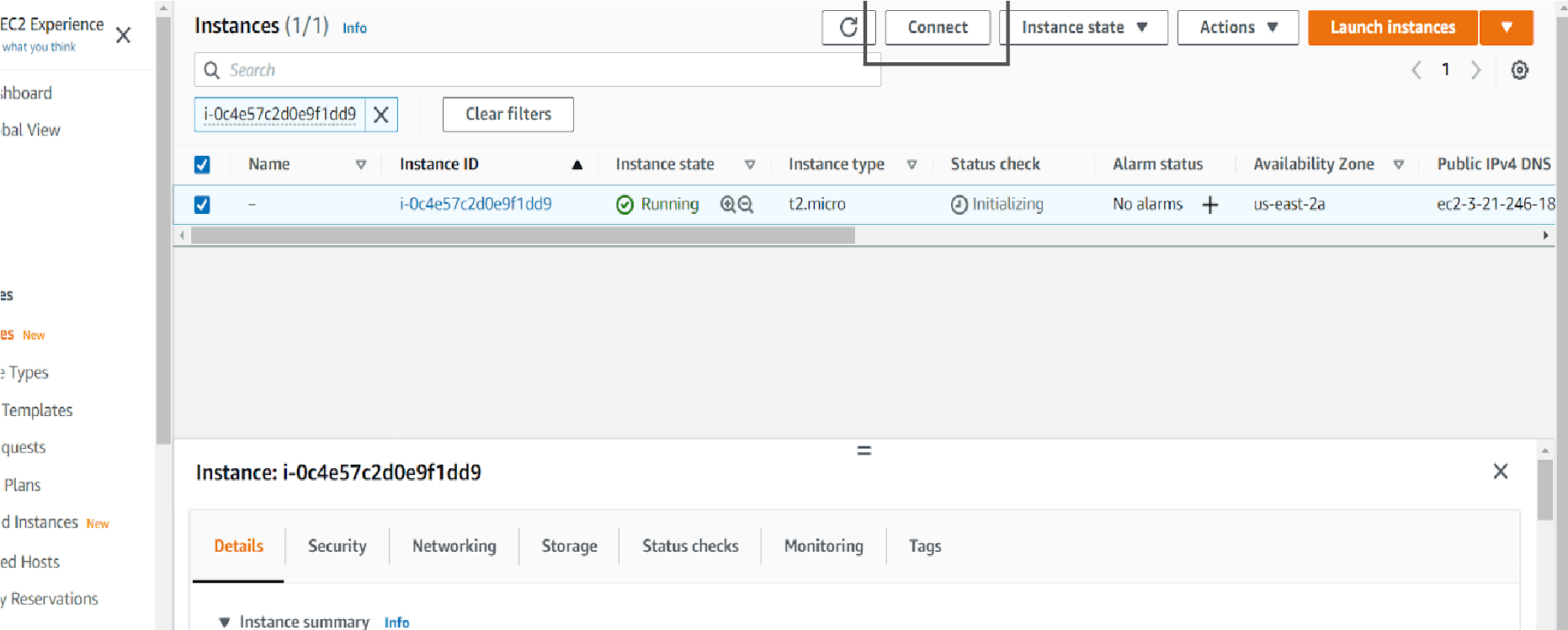
● We select **Create a new key pair**, name it (e.g., keytask), and download the .pem file—**this is our only copy!**

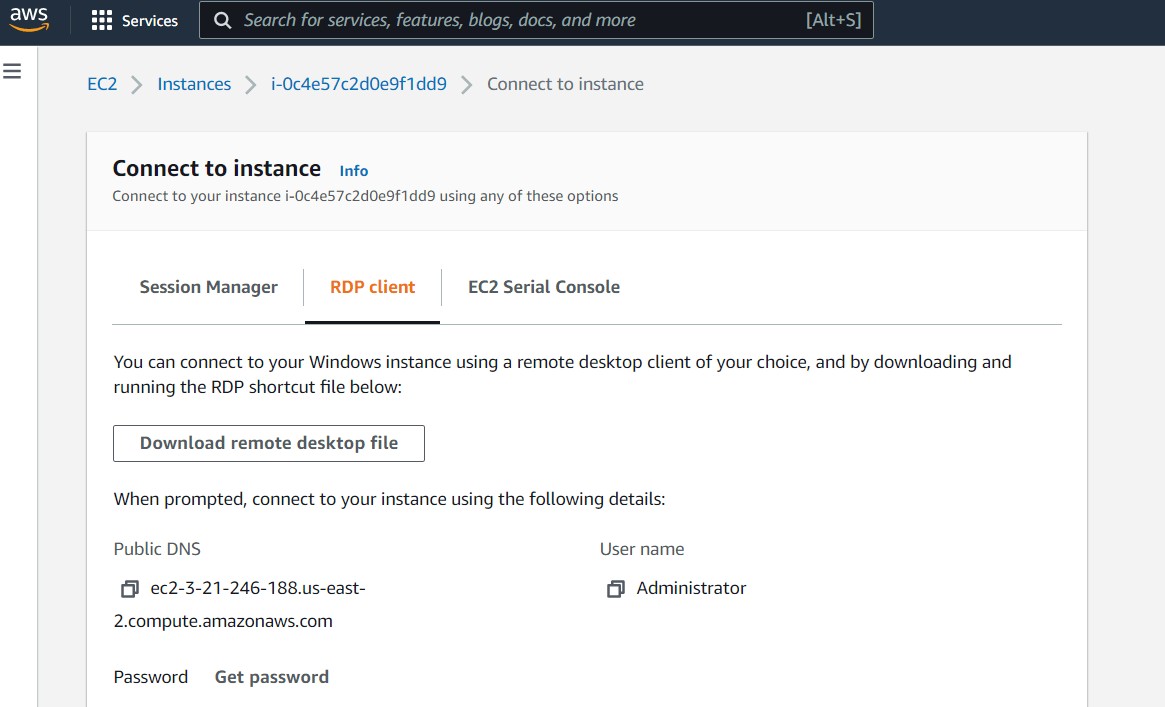
 We click **Launch Instances**.





## 8. Connect to an instance







# Conclusion

Through this experiment, we have:

* Demonstrated the process of provisioning a Windows VM on AWS EC2—an IaaS example.
* Highlighted how system virtualization enables the core cloud benefits of elasticity, resource abstraction, and secure multi-tenancy.
* Shown that leveraging prebuilt AMIs and managed services accelerates deployment and reduces infrastructure overhead.

By mastering these steps, we gain practical insight into virtualization’s pivotal role in cloud computing and prepare to build scalable, secure environments for diverse workloads.