AWS EC2 – Ubuntu Connection Methods (Step-by-Step)

Default Ubuntu EC2 User: ubuntu

Method 1: Connect Using PEM Key (Recommended)

Step 1: Set correct permission for the PEM file

chmod 400 mykey.pem

Step 2: Connect to EC2 using SSH and PEM key

ssh -i mykey.pem ubuntu@<public-ip-or-dns>

Method 2: Connect Using Username and Password (Not Recommended)

Step 1: Login using PEM key first

ssh -i mykey.pem ubuntu@<public-ip-or-dns>

Step 2: Set password for ubuntu user

sudo passwd ubuntu

Step 3: Enable password authentication (Main SSH config)

sudo nano /etc/ssh/sshd_config
Set: PasswordAuthentication yes

IMPORTANT (Ubuntu 22.04 / 24.04 – AWS)

AWS uses cloud-init to enforce SSH security defaults. Even if sshd_config is updated, password login will NOT work unless cloud-init configuration is also updated.

Step 4: Update cloud-init SSH override (MANDATORY)

sudo nano /etc/ssh/sshd_config.d/50-cloud-init.conf

Set: PasswordAuthentication yes

Step 5: Restart SSH service

sudo systemctl restart ssh

Step 6: Verify configuration

sudo sshd -T | grep passwordauthentication

Expected output: passwordauthentication yes

Step 7: Login using username and password

ssh ubuntu@<public-ip>

Method 3: Passwordless Login (Best Practice)

Step 1: Create a new user (example: appuser)

sudo adduser appuser

Step 2: Generate SSH key on local machine

ssh-keygen -t ed25519 -C "appuser@ec2"

Step 3: Add public key to authorized_keys on EC2

sudo mkdir -p /home/appuser/.ssh
sudo nano /home/appuser/.ssh/authorized_keys

Step 4: Fix SSH permissions

sudo chown -R appuser:appuser /home/appuser/.ssh
sudo chmod 700 /home/appuser/.ssh
sudo chmod 600 /home/appuser/.ssh/authorized_keys

Step 5: Login without PEM and password

ssh appuser@<public-ip>

Security Best Practices

After validation, disable password authentication and root login to secure the server.

PasswordAuthentication no
PermitRootLogin no
sudo systemctl restart ssh

Elastic Cloud Compute (EC2)

An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2).

EC2 Instance Types (Updated – 2025)

EC2 Instance Types describe the "hardware" components that an EC2 instance runs on:

- Compute power (vCPU / processor)
- Memory (RAM)
- Storage options / optimization
- Network performance (bandwidth, latency)

Instance types are grouped into families, each optimized for specific workloads.

1. General Purpose Instances

Balanced compute, memory, and networking.

Families (2025): T4g, T3, T3a, M7g, M7a, M7i, M6g, M6a, M6i

Use cases: Web servers, application servers, small to medium databases, enterprise applications

2. Compute Optimized Instances

High performance processors, ideal for CPU-intensive workloads.

Families (2025): C7g, C7a, C7i, C6g, C6a, C6i

Use cases: High-performance web servers, batch processing, media transcoding, scientific & engineering applications, game servers

3. Memory Optimized Instances

Designed for memory-intensive workloads.

Families (2025): R7g, R7a, R7i, R6g, R6a, R6i, X2gd, X2idn, X2iedge, High Memory (u-instances)

Use cases: High-performance databases, in-memory databases (SAP HANA, Redis), big data analytics, real-time processing

4. Accelerated Computing Instances

Use hardware accelerators (GPU, FPGA, Inferentia, Trainium).

GPU Instances: P5, P4d, P4de, G6, G5, G4dn

AWS AI Accelerators: Trn1, Trn1n, Inf2

FPGA: F1

Use cases: Machine learning (training & inference), AI workloads, 3D rendering, video encoding, high-performance computing (HPC)

5. Storage Optimized Instances

High disk throughput and IOPS.

Families (2025): I4i, I4g, Im4gn, D3, D3en, H1 (legacy)

Use cases: NoSQL databases (Cassandra, MongoDB), data warehouses, Elasticsearch / OpenSearch, log processing, distributed file systems

6. Network Optimized / High Throughput Instances

Families: M7i-flex, C7gn, R7iz

Use cases: Network appliances, high-throughput microservices, large-scale distributed systems

7. ARM (Graviton-based) Instances (Strongly Recommended)

Powered by AWS Graviton3 / Graviton4 processors.

Families: T4g, M7g, C7g, R7g, X2gd

Benefits: Lower cost, better price-performance, lower power consumption

EC2 Purchasing Options (Updated – 2025)

1. On-Demand Instances

• Choose any instance type
• Launch and terminate at any time
• Most flexible option
• Charged only while running (per-second billing, 60s minimum)
• Most expensive pricing model

Best for: Short-term workloads, dev/test, unpredictable traffic

2. Reserved Instances (RI)

• Commit for 1-year or 3-year term
• Significant discount compared to On-Demand
• Standard RI (highest discount, fixed attributes)
• Convertible RI (lower discount, flexible changes)
• Zonal RI provides capacity reservation

Discount: Up to 72% vs On-Demand

Best for: Steady-state production workloads

3. Savings Plans (Preferred in 2025)

• Commit to a $/hour spend instead of instance type
• Automatically applied across usage
• Compute Savings Plan (most flexible)
• EC2 Instance Savings Plan (higher discount, less flexible)

Discount: Up to 66%

Best for: Predictable compute usage across services

4. Spot Instances

• Uses unused AWS capacity
• Up to 90% discount
• Can be interrupted with 2-minute notice
• No bidding model (AWS sets price automatically)
• Charged per second

Best for: Batch jobs, CI/CD, ML training, non-prod workloads

5. Dedicated Hosts

• Dedicated physical server
• Full control over sockets and cores
• Supports BYOL licensing
• Highest cost model

Best for: License compliance and regulatory workloads

6. Dedicated Instances

• Runs on dedicated hardware
• Hardware shared only within your account
• Less control than Dedicated Hosts

Best for: Compliance without host-level control

7. Capacity Reservations

• Reserve EC2 capacity in a specific AZ
• Guarantees capacity availability
• No inherent discount
• Charged whether used or not

Best for: Mission-critical and DR workloads

Summary: Choose On-Demand for flexibility, Savings Plans or RIs for cost optimization, Spot for fault-tolerant workloads, and Dedicated options or Capacity Reservations for compliance and guaranteed capacity.

EBS (Elastic Block Store) – Updated 2025

Amazon Elastic Block Store (EBS) provides persistent, block-level storage for Amazon EC2 instances.

EC2 Elastic Block Store (EBS) – Basics

• EBS volumes are persistent and survive instance stop/terminate unless deleted
• Network-attached block storage
• Can be attached to one EC2 instance at a time (except Multi-Attach for io1/io2)
• Volumes can be attached/detached within the same AZ
• Supports point-in-time snapshots stored in Amazon S3
• Replicated within the Availability Zone by default
• Common device names: /dev/xvda, /dev/nvme0n1

EBS Volume Types (2025)

1. General Purpose SSD (gp3 / gp2)

gp3 (Recommended): Decoupled size, IOPS, and throughput. Cheaper and faster than gp2.

Performance: Up to 16,000 IOPS, 1,000 MB/s throughput
Volume size: 1 GiB – 16 TiB
Use cases: Boot volumes, dev/test, app servers, small/medium databases

2. Provisioned IOPS SSD (io2 / io2 Block Express)

Designed for mission-critical workloads with guaranteed performance.

Performance: Up to 256,000 IOPS, 4,000 MB/s throughput
Volume size: 4 GiB – 64 TiB
Use cases: Oracle, SQL Server, SAP, latency-sensitive applications

3. Throughput Optimized HDD (st1)

Optimized for large sequential workloads. Cannot be a boot volume.

Volume size: 125 GiB – 16 TiB
Use cases: Big data, log processing, ETL workloads

4. Cold HDD (sc1)

Lowest-cost EBS volume for infrequently accessed data. Cannot be a boot volume.

Volume size: 125 GiB – 16 TiB
Use cases: Archives, backup data, cold datasets

5. Magnetic (Standard) – Deprecated

Previous-generation EBS volumes. Not recommended.

EBS Snapshots

• Incremental backups stored in Amazon S3
• Can be copied across regions and shared across accounts
• Supports encryption
• Used for AMIs, disaster recovery, and volume restoration

Interview Summary: EBS provides persistent block storage for EC2, with gp3 for general workloads, io2 for high-performance databases, and HDD volumes for throughput or cold data.

EBS (Elastic Block Store) – Updated 2025

Amazon Elastic Block Store (EBS) provides persistent, block-level storage for Amazon EC2 instances.

EBS Volume Quick Comparison

Volume Type	Use Case	Boot Volume	Cost
gp3	General purpose	Yes	$
io2	Mission critical	Yes	$$$
st1	High throughput	No	$
sc1	Cold data	No	$
Magnetic	Deprecated	No	N/A

Note: gp3 is recommended for most workloads. io2 should be used only for latency-sensitive, mission-critical applications. HDD volumes cannot be used as boot volumes.