


AMI

AMI = Amazon Machine Image

- It's like a **template** for launching a virtual computer (EC2 instance) on AWS.
- **Snapshot** or **template** of an EC2 instance

 A **snapshot of a full machine** — including the operating system, installed software, settings, and sometimes even preloaded data or scripts.



Every time you launch an EC2 instance, it **must be based on an AMI**.

It includes:

- The **operating system** (e.g., Ubuntu, Windows).
- **Pre-installed software** (e.g., Python, Docker, LLM models).
- **Configuration settings** (users, permissions, security).

You can use AMIs to **launch identical EC2 instances** quickly.

Real-World Analogy

Imagine you're setting up 100 computers in a lab.

Would you install Windows and all the tools manually on each one? No. You'd:

- Configure one "master" PC with OS + software
- Take an image (copy) of it
- Use that image to clone the rest

That “image” is like an **AMI in AWS**.

✓ What Does an AMI Contain?

An AMI typically includes:

Component	Meaning
✓ OS	Ubuntu, Amazon Linux, Windows, etc.
✓ Software	Python, Docker, Apache, Jupyter, etc.
✓ Config files	Environment variables, shell scripts
✓ EBS snapshot	A snapshot of the root volume



So when you launch a new EC2 instance from an AMI, you instantly get everything preinstalled.



Example Use Case: Custom AMI

Let's say:

- You install Ubuntu EC2
- Set up Python, Conda, Apache, model files
- You want to reuse this setup later

👉 You can create a **Custom AMI** of that EC2 instance.

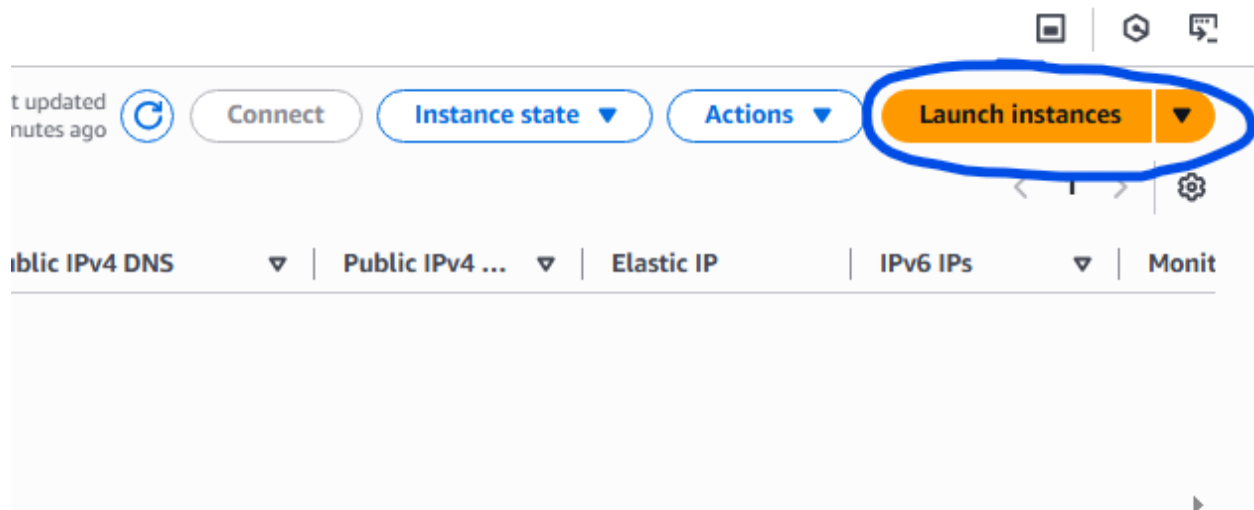
Then anytime you launch a new instance from that AMI:

- **You save hours of setup time**
- You get consistent results



How to Launch an EC2 Using an AMI (AWS Console)?

1. Go to EC2 → **Launch Instance**



2. **Choose an AMI**

- Pick from:
 - ☒ Amazon Linux 2023
 - ☒ Ubuntu Server 22.04
 - ☒ Deep Learning AMI (comes with PyTorch, TensorFlow)
 - ☒ Your Custom AMIs (if you've created one)

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

AMI

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents | **Quick Start**

Amazon Linux
aws

macOS
Mac

Ubuntu
ubuntu®

Windows
Microsoft

Red Hat
Red Hat

SUSE Linux
SUSE

Debian
debian

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-0d03cb826412c6b0f (64-bit (x86), uefi-preferred) / ami-086fa8714a317e8c6 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250623.1 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username ⓘ	Verified provider
64-bit (x86) ▼	uefi-preferred	ami-0d03cb826412c6b0f	2025-06-20	ec2-user	

How to Create a Custom AMI from EC2?

Step-by-Step:

1. Go to EC2 → Instances
2. Select your running instance
3. Click **Actions** → **Image** → **Create Image**
4. Give it:
 - A name (e.g. `llm-inference-server-v1`)
 - Description
 - Choose "No reboot" if you want a hot snapshot
5. Click **Create Image**

🎉 Done! You've created your own AMI.

It will appear under:

EC2 → AMIs → Owned by me

Advanced Use Cases

Use Case	Description
Scale identical servers	Launch 5–100 EC2s with same AMI
Backup a machine	AMI acts like a full backup
Share setup with others	AMIs can be shared with specific AWS accounts
CI/CD pipelines	Create versioned AMIs for software releases

Common Beginner Mistakes

Mistake	Fix
❌ Assuming AMI = EC2 instance	EC2 is the running computer , AMI is the template
❌ Creating AMIs without cleaning up	Always stop services and delete temp files before AMI
❌ Forgetting to update AMI after installing new things	If you install updates, make a new AMI
❌ Not checking AMI region	AMIs are region-specific — copy them if needed elsewhere

Key Concepts

Term	What It Means	Example
AMI ID	Unique identifier for an AMI (e.g., <code>ami-0abcdef1234567890</code>).	<code>ami-0c55b159cbfafa1f0</code> (Amazon Linux).

Term	What It Means	Example
Root Volume	The primary disk (usually 8GB–100GB) containing the OS.	An Ubuntu AMI with 20GB root volume.
Instance Type	The hardware specs (CPU, RAM) the AMI supports.	t2.micro , g5.xlarge .

How to Use AMIs?

1. Find a Public AMI

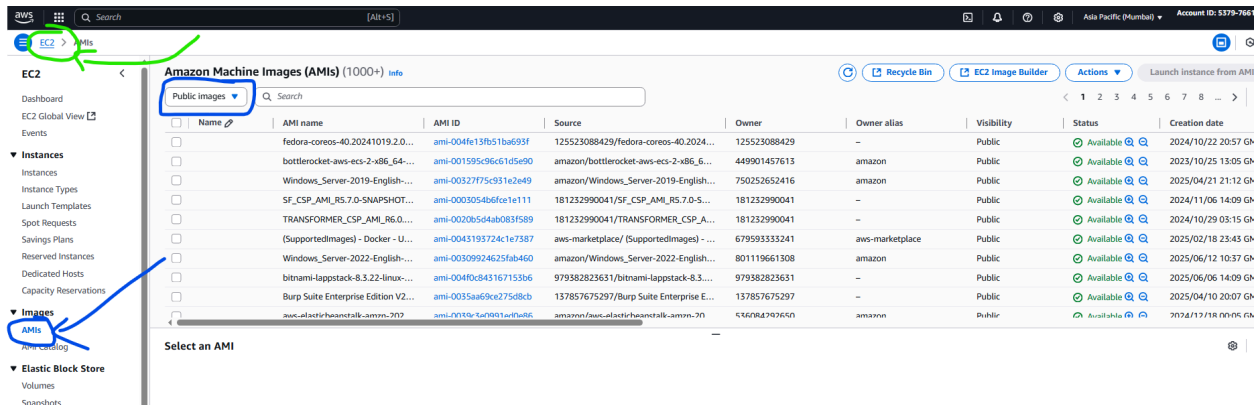
- **AWS Marketplace:** Pre-configured AMIs (e.g., Deep Learning AMI).
- **Community AMIs:** Free images (Ubuntu, CentOS, etc.).

Steps:

1. Go to **EC2 Dashboard** → **AMIs** → **Public images**.

2. Search for keywords like:

- **ubuntu** → Official Ubuntu AMI.
- **deep learning** → Pre-installed TensorFlow/PyTorch.



The screenshot shows the AWS Management Console interface for Amazon Machine Images (AMIs). The left sidebar has a green circle around the 'EC2' menu item and a blue arrow pointing to the 'AMIs' link. The main content area shows the 'Amazon Machine Images (AMIs) (1000+)' page with the 'Public images' tab selected. A search filter is applied. The table lists various AMIs with columns for Name, AMI name, AMI ID, Source, Owner, Owner alias, Visibility, Status, and Creation date.

Name	AMI name	AMI ID	Source	Owner	Owner alias	Visibility	Status	Creation date
fedora-coreos-40.20241019.2.0...	ami-004fe13fb51ba698f	125523088429/fedora-coreos-40.2024...	125523088429	amazon	Public	Available	2024/10/22 20:57 GK	
bottlerocket-aws-ecs-2-x86_64...	ami-001595c96c61d5e90	amazon/bottlerocket-aws-ecs-2-x86_6...	449901457613	amazon	Public	Available	2023/10/25 13:05 GK	
Windows_Server-2019-English-...	ami-00327f75c931e2e49	amazon/Windows_Server-2019-English...	750252652416	amazon	Public	Available	2025/04/21 21:12 GK	
SF_CSP_AMI_RS.7.0-SNAPSHOT...	ami-003054b6fce1e111	181232990041/SF_CSP_AMI_RS.7.0-S...	181232990041	Public	Available	2024/11/06 14:09 GK		
TRANSFORMER_CSP_AMI_R6.0...	ami-0020b544b083f589	181232990041/TRANSFORMER_CSP_A...	181232990041	Public	Available	2024/10/29 03:15 GK		
(SupportedImages) - Docker - U...	ami-0045193724c1e7387	aws-marketplace/(SupportedImages) - ...	679593333241	aws-marketplace	Public	Available	2025/02/18 23:43 GK	
Windows_Server-2022-English-...	ami-00309924625fab460	amazon/Windows_Server-2022-English...	801119661308	amazon	Public	Available	2025/06/12 10:37 GK	
bitnami-lappstack-8.3.22-linux-...	ami-004f0c843167153b6	979382823631/bitnami-lappstack-8.3...	979382823631	Public	Available	2025/06/06 14:09 GK		
Burp Suite Enterprise Edition V2...	ami-0035aa69ce275d8cb	137857675297/Burp Suite Enterprise E...	137857675297	Public	Available	2025/04/10 20:07 GK		
aws-ml-training-ami-2024.10.1...	ami-70103c-fa7001e0f0b6	amazon/aws-ml-training-ami-2024.10.1...	67676847976767	amazon	Public	Available	2024/10/18 00:06 GK	

2. Launch an EC2 Instance from an AMI

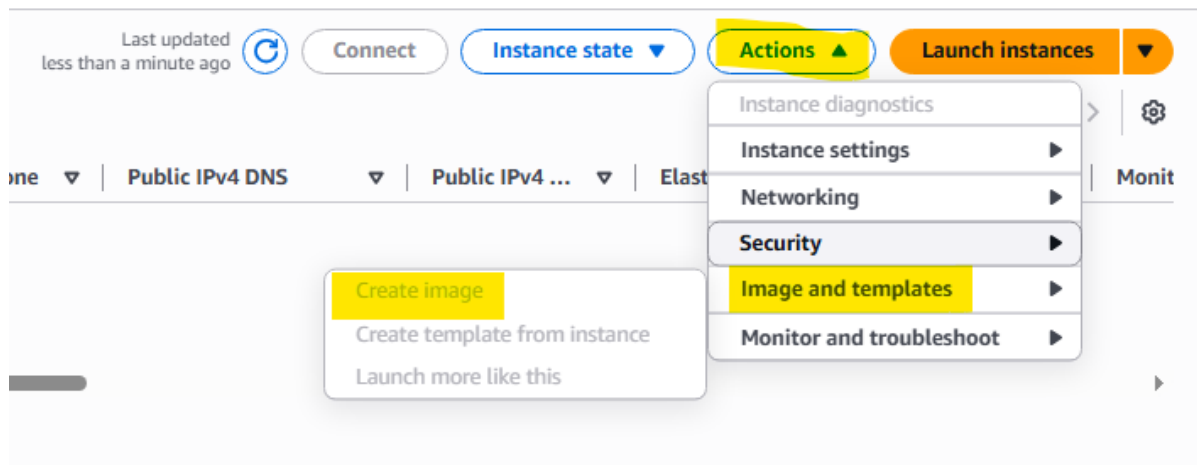
1. Select an AMI → Click **Launch Instance**.
2. Choose an **instance type** (e.g., **t2.micro** for testing).
3. Configure storage, security groups, and keys.

4. Click **Launch**.

3. Create Your Own AMI

If you customize an EC2 instance (e.g., install an LLM model), save it as an AMI:

1. Go to **EC2** → **Instances** → Select your instance.
2. Click **Actions** → **Image and templates** → **Create Image**.



3. Name it (e.g., `my-llm-server-ami`) → **Create**.

Image name

webserver-v1

Maximum 127 characters. Can't be modified after creation.

Image description - *optional*

Image description

Maximum 255 characters

☒ **Reboot instance**

When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
E...	/...	Create new sn...	8	EBS General P...	3000		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

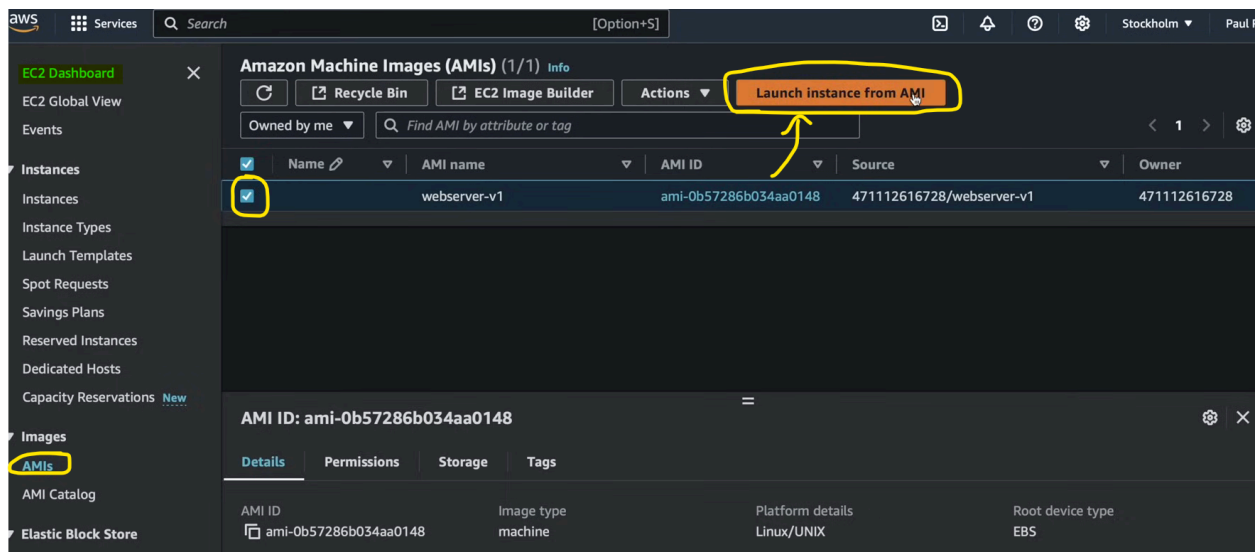
[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

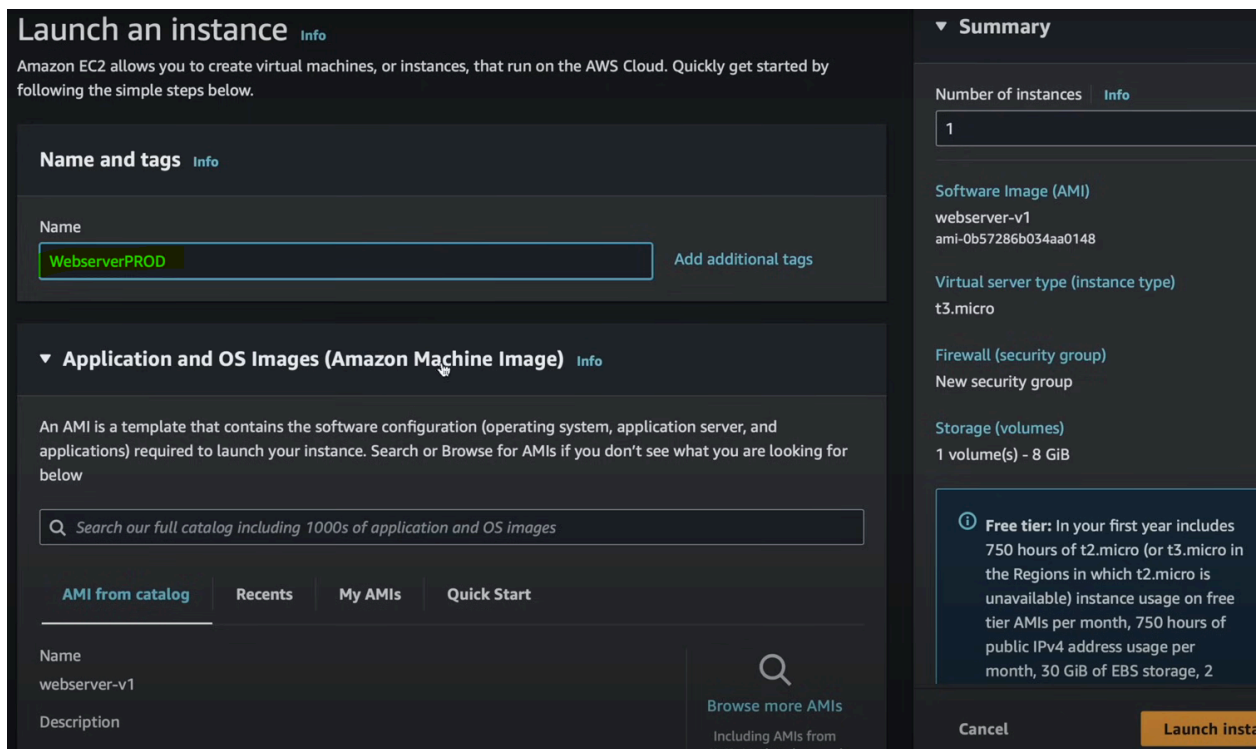
✓ Now, you can launch **identical copies** of this instance anytime!

Create an EC2 Instance from AMI

- EC2 → Images → AMI
- Select the AMI
- Click **Launch instance from AMI**



- Select the configurations



▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select ▲

Q

Proceed without a key pair (Not recommended) Default value

mywebserver-key
Type: rsa Edit

Network [Info](#)

vpc-0151338dee8ae5f78

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Number of instances [Info](#)

1

Software Image (AMI)

webserver-v1
ami-0b57286b034aa0148

Virtual server type (instance type)

t3.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2

- **New instance created**

Instances (1/2) [Info](#) Last updated less than a minute ago Connect Instance state ▼ Actions ▼ Launch instances ▼

Find Instance by attribute or tag (case-sensitive) All states ▼

Instance state = running Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	TESTING	i-01b2462302b9181e8	Running	t3.micro	3/3 checks passed	View alarm
<input checked="" type="checkbox"/>	WebserverPROD	i-0ed2e13d9aa9ea1e1	Running	t3.micro	Initializing	View alarm

i-0ed2e13d9aa9ea1e1 (WebserverPROD)

Details **Status and alarms** **Monitoring** **Security** **Networking** **Storage** **Tags**

Instance summary [Info](#)

Instance ID i-0ed2e13d9aa9ea1e1 (WebserverPROD)	Public IPv4 address 16.170.228.24 open address	Private IPv4 addresses 172.31.39.187
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-16-170-228-24.eu-north-1.compute.amazonaws.com

- If you Copy & open the public IP, the website deployed in the old EC2 instance will be available from the public IP of the new instance.

AMI Costs

- **Storing AMIs:**
 - You pay for **EBS snapshot storage** (~₹2.30/GB/month).
 - Example: A 20GB AMI costs ~₹46/month.
- **Public AMIs:**
 - Some are free (e.g., Ubuntu), others charge hourly (e.g., Windows).

Use Cases

1. Pre-installed AI Tools

- Use the **AWS Deep Learning AMI** (has PyTorch, TensorFlow preloaded).

2. Disaster Recovery

- Backup your server as an AMI → Restore if crashes.

3. Scaling

- Launch 10 identical servers from one AMI for load balancing.

Pitfalls to Avoid

- **Don't leave unused AMIs** → They still cost money!
- **Region-locked:** AMIs work only in the region they're created.
- **Licensing costs:** Windows/paid AMIs bill hourly.

AWS CLI Commands

Task	Command
List AMIs	<code>aws ec2 describe-images --owners self</code> (for your custom AMIs).
Create AMI	<code>aws ec2 create-image --instance-id i-123abc --name "My-LLM-AMI"</code>
Launch from AMI	<code>aws ec2 run-instances --image-id ami-0abc123 --instance-type t2.micro</code>

? FAQ

Q: Can I share my AMI with others?

A: Yes! Go to AMIs → Select AMI → Actions → Modify Image Permissions.

Q: How do I delete an AMI?

A: Deregister it (EC2 → AMIs → Select → Actions → Deregister). Then delete its **snapshot** (EC2 → Snapshots).

Q: Can I move an AMI to another region?

A: Yes! Use EC2 → AMIs → Actions → Copy AMI.

AMI Catalog

The screenshot displays the AWS AMI Catalog interface. On the left is a navigation sidebar with categories like EC2, Instances, Images, Elastic Block Store, and Network & Security. The 'Images' section is expanded, showing 'AMI Catalog'. The main area features a search bar and four tabs: 'Quick Start AMIs (47)', 'My AMIs (0)', 'AWS Marketplace AMIs (5895)', and 'Community AMIs (500)'. Below the tabs, a list of AMIs is shown, each with a logo (AWS or Ubuntu), a title, an ID, platform, release notes, root device type, virtualization, and ENA status. Each entry has a 'Select' button and a radio button to choose between '64-bit (x86)' and '64-bit (Arm)'.

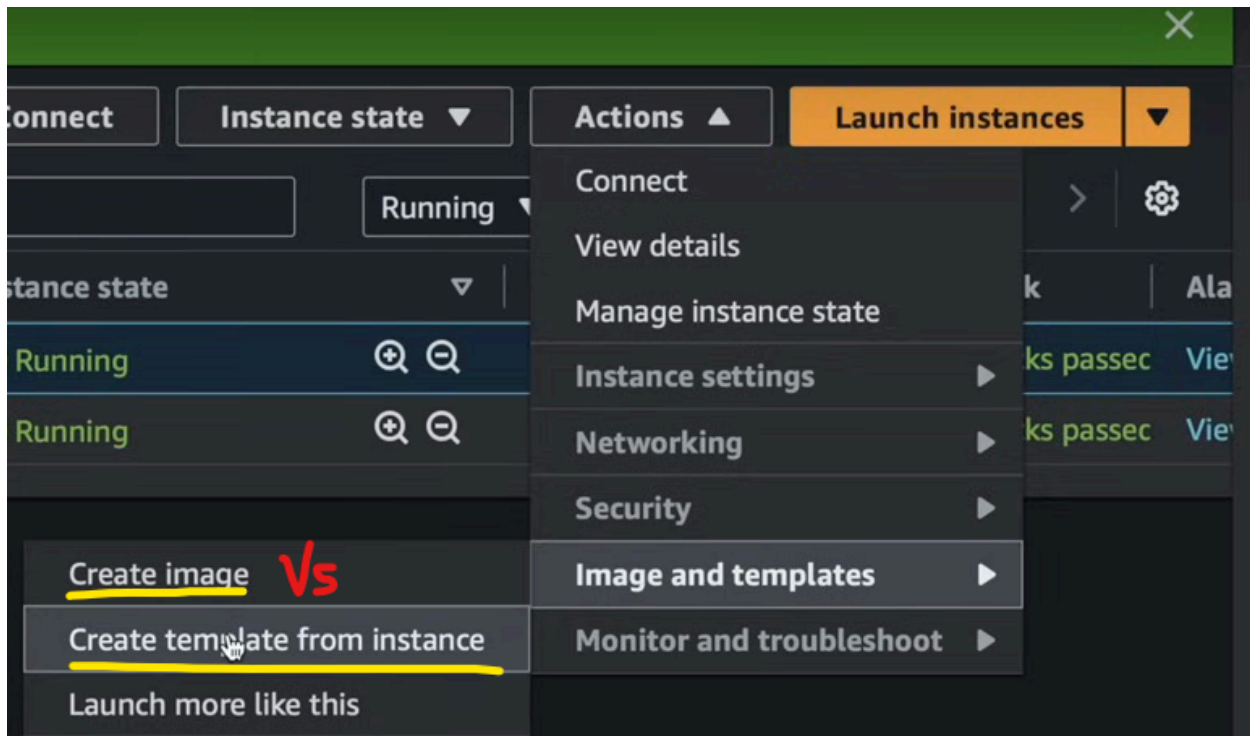
Provider	AMI ID	Platform	Root device type	Virtualization	ENA enabled	Architecture
AWS	ami-0342f316da6d5c301	amazon	ebs	hvm	Yes	64-bit (x86)
AWS	ami-0854fdee9e9a50d1b	amazon	ebs	hvm	Yes	64-bit (x86)
AWS	ami-0a245bc773de576bd	amazon	ebs	hvm	Yes	64-bit (x86)
Ubuntu	ami-0c7c2a4c211a14b72	ubuntu	ebs	hvm	Yes	64-bit (x86)
Ubuntu	ami-0140f55e736d9486	ubuntu	ebs	hvm	Yes	64-bit (x86)



Create Image from Instance

VS



Launch Template from Instance



Feature	Create Image (AMI)	Launch Template
Saves what?	A snapshot of the EC2 instance's disk (OS, software, data)	A blueprint of how to launch new EC2 instances (AMI, instance type, key, etc.) without storing the data
What is it?	A copy of the machine's disk	A recipe to launch machines with certain settings
Includes?	OS, app code, files, packages ✅	AMI ID, instance type, VPC, key pair, security group, user data, etc. ✅
Used for?	Backup, cloning, portability	Consistent, repeatable EC2 launches
Replaces EC2?	Yes — restores the entire machine state	No — launches a new EC2 based on saved settings
Cost?	Costs to store the image (in S3)	No cost — just saved metadata

 **Think of It Like:**

Real World Analogy	Description
Create Image	You take a snapshot of your laptop : full OS, installed apps, files, settings
Launch Template	You fill a form that says: "Next time I build a laptop, use Ubuntu, 16GB RAM, preinstall Python, use this password, etc."

Concept	What It's Like
Create Image (AMI)	 Ctrl+C & Ctrl+V of your full EC2 instance → OS, apps, config files, everything copied exactly
Launch Template	 Blueprint or form that says: " <i>Next time I want a server like this, use this AMI, this size, this startup script</i> "

- **AMI = actual snapshot (disk copy)**
- **Launch Template = how to launch a new instance using that snapshot and other configs**

What Is **Create Image** (AMI)?

✅ It creates a **complete image (AMI)** of your EC2 instance:

- OS (e.g., Ubuntu, Amazon Linux)
- Software installed (Python, Nginx, etc.)
- Files and folders
- Even user config (if included)

Use it when:

- You've finished setting up an EC2 and want to **reuse the same environment later**
- You want a **backup** (like a restore point)
- You want to launch **clones** in the future

How to use:

- Go to EC2 → Instances → Select → [Actions → Image → Create Image](#)

It becomes available in **EC2 → AMIs**

You can now launch new EC2 instances from this AMI.



What Is **Launch Template** ?

✓ It is a **saved configuration** for launching EC2 instances.

Includes settings like:

- AMI ID
- Instance type (e.g., [t2.micro](#) , [g4dn.xlarge](#))
- Key pair
- Security groups
- IAM role
- EBS volume size
- User data script (startup commands)

Use it when:

- You want to **automate EC2 creation**
- You want to always launch instances with the same settings
- You use **Auto Scaling Groups** or **Spot Fleets**
- You want to launch LLM or app servers consistently with custom startup scripts

How to create:

- EC2 → Instances → Select → [Actions → Create launch template from instance](#)

It becomes available under:

EC2 → Launch Templates

You can now launch new EC2s using that template **instantly with 1 click**.

Feature	Comes from AMI?	Comes from Launch Template?
Ubuntu + Python + Apache	✔ Yes	✗ No
Your model/code files	✔ Yes	✗ No
EC2 instance type (e.g. t2.micro)	✗ No	✔ Yes
IAM role, key pair, VPC	✗ No	✔ Yes
Auto-run script on boot	✗ No	✔ Yes
Launch policy (Spot vs On-Demand)	✗ No	✔ Yes