EC2 (Elastic Compute Cloud)

EC2 (Elastic Compute Cloud) is a core compute service provided by Amazon Web Services (AWS) that allows you to run virtual servers (called instance) in the cloud. The service provides access to virtual machines on the hardware within Amazon data centers. To put it simply, EC2 is a service that allows users to get access to computers to run the applications they desire. Through various payment methods, the user effectively "rents" a certain amount of computing power from AWS through EC2.

So, EC2 removes the need to manage physical hardware and gives you full control over:

- Operating systems
- Instance size and performance
- Networking and security
- Storage and scaling

You can launch, stop, resize, and terminate instances within minutes, paying only for what you use.

Key Takeaways

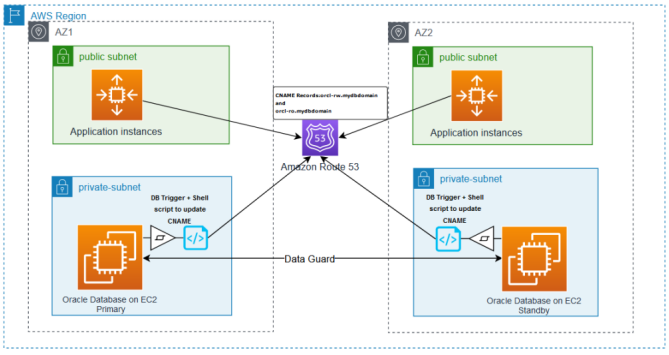
- Global availability – deploy in multiple Regions and AZs
- EC2 Instances are virtual machines that are provisioned via AWS - through the console, or programmatically.
- EC2 Instances fall under 6 categories - General Purpose, Compute Optimized, Memory Optimized, Accelerated Computing, Storage Optimized, and HPC Optimized.
- EC2 Instances are specified via AWS-defined nomenclature that dictates the number of vCPUs, Memory, storage, bandwidth, etc.

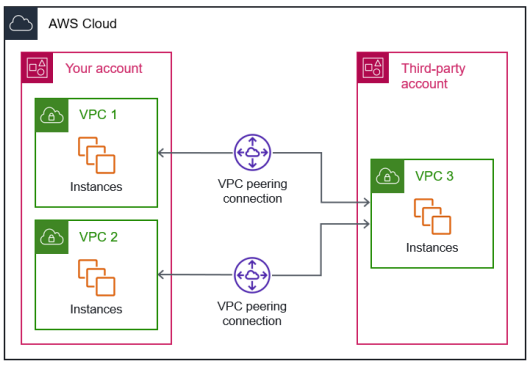
- EC2 Instances are closely tied to volumes specified under other AWS services (Ex: EBS, EFS)
- EC2 Instances can be paid for via various purchasing options. These options range from by the second to reserving and instance for a period of 3 years. By choosing the option that matches the requirements of the application, the user can benefit from substantial savings.
- EC2 Instances are closely tied to Security Groups which dictate how traffic flows into and out of a given EC2 Instance.

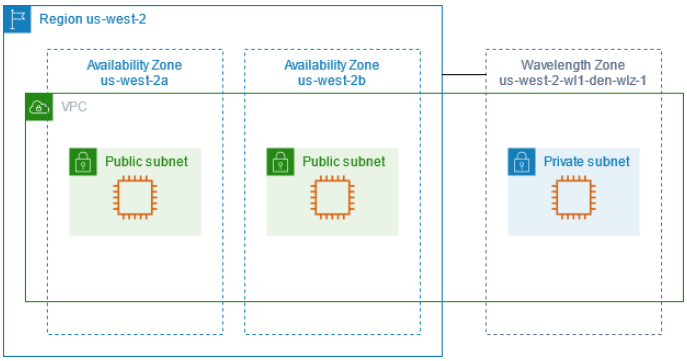
Benefits

- Elastic scalability – scale up or down based on demand
- Wide instance selection – optimized for compute, memory, storage, or networking
- Full OS control – Linux, Windows, or custom images
- Deep AWS integration – VPC, IAM, ELB, Auto Scaling
- Pay-as-you-go pricing – no upfront hardware costs

High Level EC2 Architecture







An EC2 instance typically runs:

- Inside a VPC
- Within a specific Subnet
- In one Availability Zone
- Protected by Security Groups
- Using EBS volumes for storage

EC2 Use Cases

- Hosting websites and APIs
- Running backend applications and microservices
- CI/CD runners and build servers
- Batch processing and analytics
- Machine learning workloads
- High-performance computing (HPC)
- Development and testing environments

EC2 Purchasing Options

Amazon EC2 offers a range of purchasing options to suit different usage patterns and budget requirements.

1. On-Demand Instances:

With On-Demand Instances, you pay for the compute capacity you use by the second. This provides you with flexibility and is suitable for situations where your workload fluctuates, and you need instances immediately without any long-term commitments.

2. Savings Plans:

Savings Plans allow you to commit to a certain amount of usage in USD per hour over a term of 1 or 3 years. This option offers flexibility by applying savings to a wide range of instance types, sizes, and families.

3. Reserved Instances:

Reserved Instances involve making a commitment to a specific instance configuration, including instance type and Region, for a term of 1 or 3 years. This commitment results in cost savings compared to On-Demand pricing, making it ideal for stable workloads.

4. Spot Instances:

Spot Instances let you bid on unused EC2 capacity, allowing you to access instances at a significantly lower cost. However, these instances can be terminated if the capacity is needed by On-Demand or Reserved Instances, making them suitable for fault-tolerant and flexible workloads.

5. Dedicated Hosts:

Dedicated Hosts involve paying for a physical server dedicated exclusively to running your instances. This option is beneficial when you need to meet compliance requirements or bring existing software licenses.

6. Dedicated Instances:

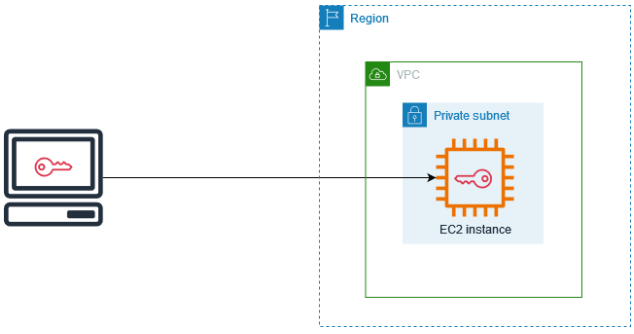
With Dedicated Instances, you pay by the hour for instances that run on single-tenant hardware. This provides isolation for compliance and regulatory needs while paying only for the instances you use.

7. Capacity Reservations:

Capacity Reservations allow you to reserve capacity in a specific Availability Zone for any duration. This ensures that the instances you need will always be available, which is useful for scenarios where you need to guarantee resources for specific workloads.

SSH Key Pairs

AWS EC2 key pairs are a fundamental security feature that helps you securely connect to your EC2 instances, which are virtual servers in the cloud. Key pairs are like a digital lock and key system that ensures only authorized users can access your instances.



How Key Pairs Work

- AWS stores the **public key** on the instance
- You keep the **private key** locally
- Authentication happens via cryptographic verification

Usage

- Linux: SSH access
- Windows: RDP (initial password decryption)

Best Practices

- Never share private keys
- Use correct permissions (`chmod 400 key.pem`)
- Rotate keys periodically
- Prefer **AWS Systems Manager Session Manager** for secure access

Amazon Machine Image (AMI)

An **AMI** is a template used to launch EC2 instances.

What an AMI Contains

- Operating system
- Installed software
- Application configuration
- Launch permissions

Types of AMIs

- AWS-provided (Amazon Linux, Ubuntu, Windows)
- Marketplace AMIs
- Custom AMIs (golden images)

Why AMIs Matter

- Faster deployments
- Consistent environments

- Easier scaling and recovery

Instance User Data

Instance User Data allows you to **run scripts automatically at launch**.

Common Uses

- Install packages
- Configure services
- Deploy applications
- Register instances with load balancers

Example (Ubuntu 22.04)

```
TEXT
#!/bin/bash

sudo apt-get update -y
sudo apt install apache2 -y
```

Important Notes

- Runs only on first launch (by default)
- Executed as root
- Logs stored in `/var/log/cloud-init-output.log`

Security Groups

Security Groups act as **virtual firewalls** for EC2.

Key Characteristics

- Stateful (return traffic automatically allowed)
- Apply at the instance level
- Only allow rules (no explicit deny)

Rule Types

- **Inbound rules** – control incoming traffic

- **Outbound rules** – control outgoing traffic

Best Practices

- Follow least privilege
- Allow only required ports
- Restrict source IP ranges
- Separate SGs by role (web, app, DB)

Elastic IP (EIP)

An Elastic IP is a **static public IPv4 address**.

Why Use Elastic IPs

- Instance IPs change on stop/start
- EIPs remain constant
- Can be reassigned instantly

Common Use Cases

- Static website endpoints
- Email servers
- Failover architectures
- Network appliances

Note: AWS charges for unused Elastic IPs.

Creating an EC2 Instance in AWS

In this short section, we will create an EC2 Instance on AWS.

Step 1 - Navigate to the EC2 Console

1.1 - From the AWS Console, click on “Services.”

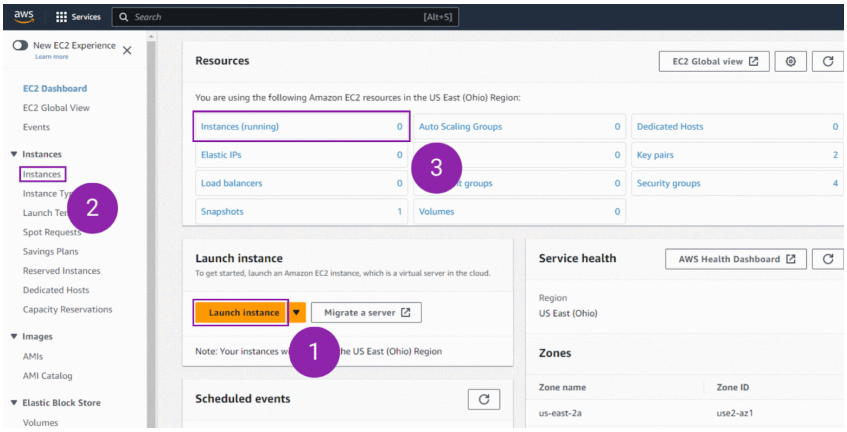
1.2 - From the drop-down menu, click on “Compute.”

1.3 - From the side menu, click on “EC2.”

You should now be on the EC2 Console.

Step 2 - Launch a new EC2 Instance

You can launch a new EC2 Instance directly from the Dashboard by pressing the “Launch instance” button. However, it’s typically done from the “Instances” page, that can be accessed via one of the two buttons on the Dashboard:
[Option 1] 2.1 - From the EC2 Dashboard, click on “Launch instance.”
[Option 2] 2.2 - From the EC2 Dashboard sidebar, click on “Instances.”
[Option 3] 2.3 - From the Resources panel, click on “Instances (running).”
If you’ve clicked “Launch instance,” go to Step 4. If you’ve chosen to navigate to the “Instances” page, follow the steps in Step 3 below.



Step 3 - [Optional] Launch a new EC2 Instance via “Instances”

The “Instances” page will display all the running EC2 instances and give you visibility into their parameters, metrics, and options.

3.1 - From the Instances page, click on “Launch instances.”

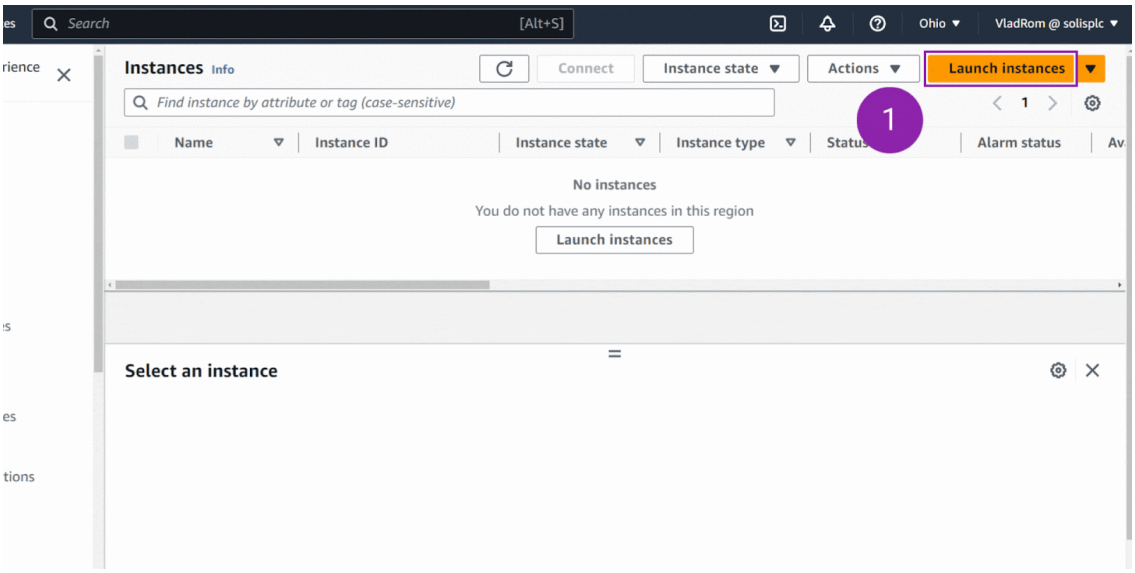
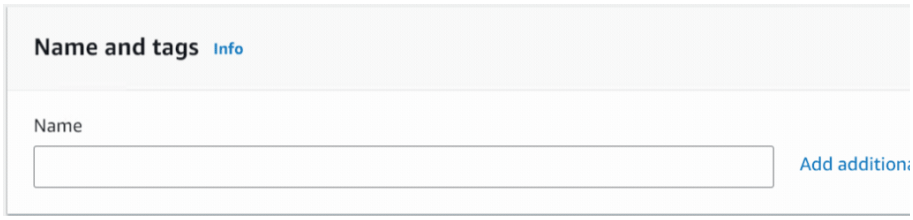


Figure 2.3 - AWS EC2 | Launching a new EC2 Instance from the “Instances” page

Step 4 - Configure the EC2 Instance

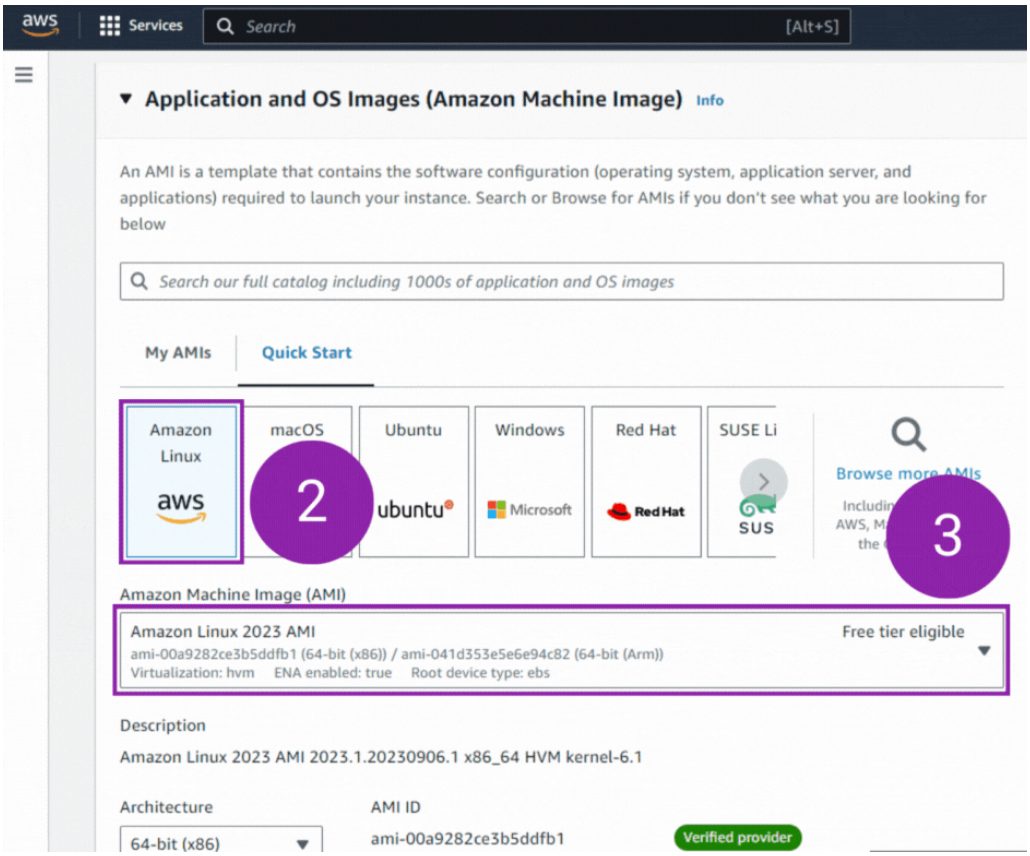
4.1 [Optional] - Type a name for your instance.



4.2 - Select the OS for your Instance

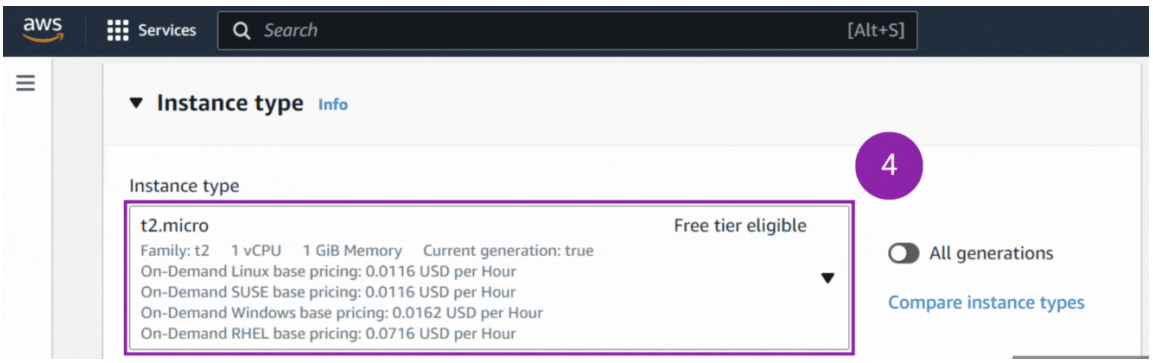
4.3 - Select the AMI for your Instance

Note: We will cover AMIs (Amazon Machine Images) in a separate tutorial. If you’re new to AWS, we recommend that you choose “Amazon Linux” as the OS, and one of the “Free tier eligible” AMIs to get started. It’s important to note that you will be billed based on the OS & AMI you select at this stage.



4.4 - Choose an Instance type.

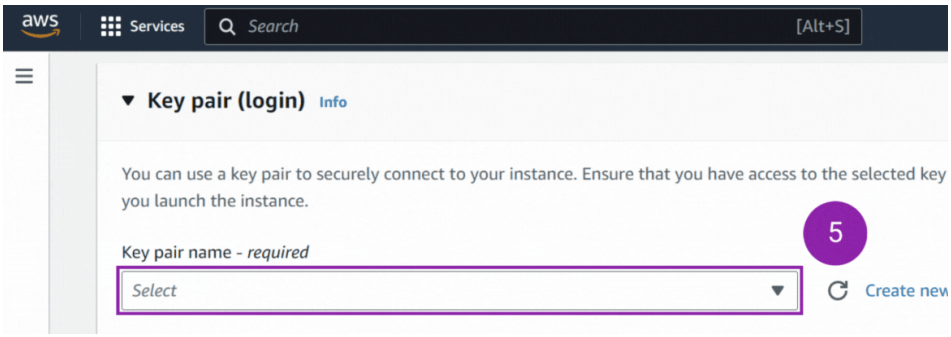
Note: We cover Instance types for various EC2 instances further in this tutoriall. If you're new to AWS, we recommend that you choose "t2.micro" as the instance type as the "Free tier eligible" instance type to get started. It's important to note that you will be billed based on the instance type you select at this stage.



4.5 - Choose a Key pair for your instance.

The key pair allows the instance to stay protected from unauthorized access. Every key pair is unique and by creating one for your instance and yourself, you can

ensure that the only ones who can access the instance are the ones with the key provided by you (admin).

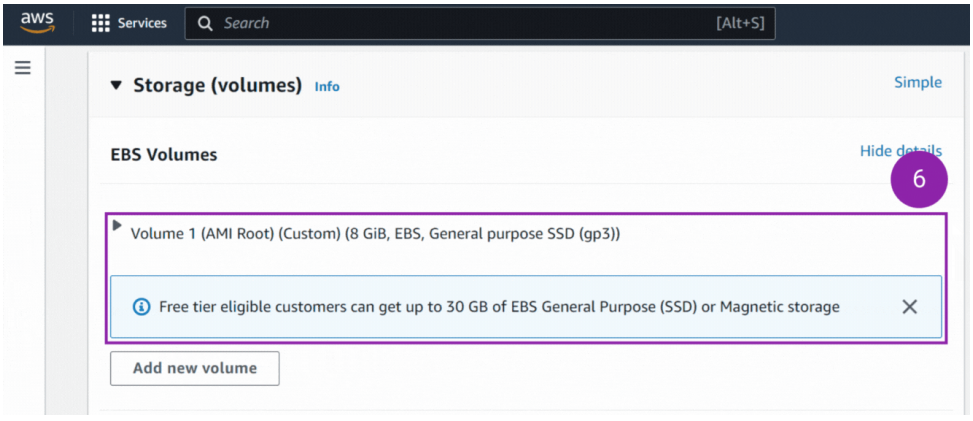


4.6 - Choose a volume that will be attached to your instance.

There are many options when it comes to volumes; we'll cover them in a separate tutorial. However, at this stage, you'll be offered two main options - EBS and EFS. If you're new to AWS, we recommend that you choose "EBS - gp3 with up to 8 GiB" as this volume falls under "Free tier eligible". It's important to note that you will be billed based on the volume type you select at this stage.

By clicking on "Show details," you'll be presented with other options for your volume: Size, Volume type, IOPS, Delete on termination, Encrypted, and Throughput. These concepts will be covered in-depth under the EBS section.

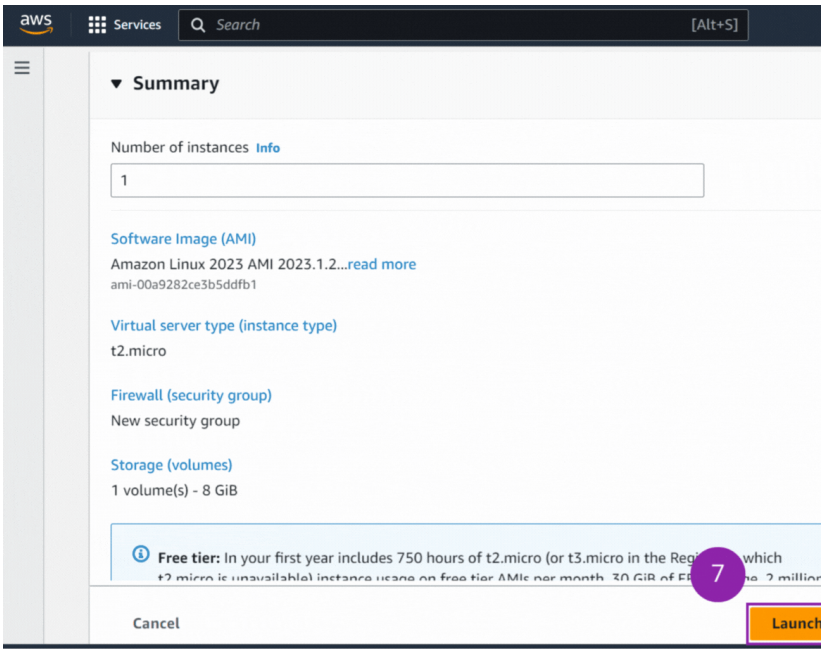
Note: you may attach multiple volumes to a single instance by pressing the "Add new volume" button.



4.7 - Review the Settings & Launch Instance

At this point, you should be able to see all the options you've selected in the steps above. You can revisit any of the steps prior to committing to the instance. It's important to note that many of these settings can't be changed on an instance that

has already been launched. If you launch an instance that has the wrong configuration, you'll most likely (not always) have to terminate the instance and re-create a new one with the correct settings.



Step 5 - Wait for the Instance to Launch

The AWS interface will bring you back to the EC2 Instances page once you launch your instance. The instance you've launched will take a moment to be accessible (typically, the process takes 1-2 minutes). The instance will be listed in the list presented to the user on this page.

Terminating an EC2 Instance in AWS

Once an instance has been launched, billing will commence. Although there are other ways to provision an instance and different billing structures than the "default" one, it's important to remove any unutilized resources from your infrastructure. In this section, we'll briefly cover how to terminate an EC2 Instance that is currently "Running."

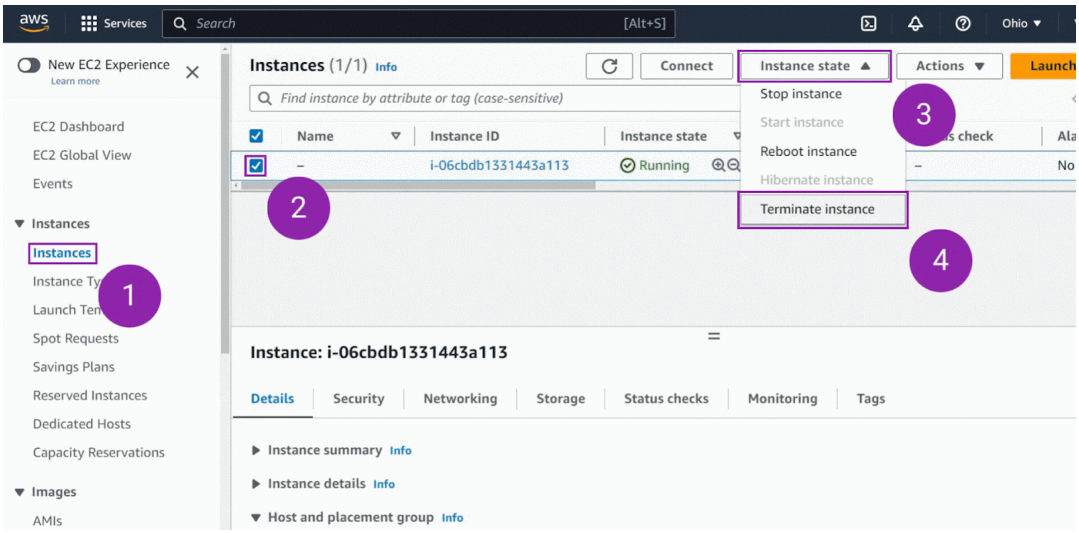
Step 1 - Terminate an EC2 Instance

1.1 - From the EC2 Dashboard, click on "Instances."

1.2 - From the list of Instances, choose the Instance(s) you want to terminate.

1.3 - From the top bar, click on "Instance state."

1.4 - From the drop-down menu, click on "Terminate instance."



Step 2 - Confirm Volume Deletion

In general, a volume is tied to an instance. There are scenarios in which they aren’t, but users typically want to be able to access their volumes from a certain instance. That being said, during the deletion process of an instance, it’s possible to terminate the associated volumes as well. By default, an EBS volume that was created at the same time as an EC2 instance will be terminated at the termination of that instance.

2.1 - Click on “Terminate”

Note that after termination, the EC2 Instance will remain in the list of instances on the “Instances” page. Verify that the “Instance state” column properly displays the state of the terminated instance as “Terminated.” after a short period of time.