Meta-date:

In AWS (Amazon Web Services), metadata refers to the information associated with AWS resources. It provides descriptive data about the resources and can include details such as configuration settings, attributes, and other relevant information.

In the context of AWS EC2 (Elastic Compute Cloud) instances, each instance has metadata that can be accessed from within the instance. This metadata includes information like the instance ID, instance type, IP addresses, security groups, and more. It can be useful for applications running on the instance to retrieve this metadata dynamically and adapt their behavior based on the environment.

The metadata can be accessed by making HTTP requests to a special URL provided by the EC2 instance itself. The URL typically follows the pattern: http://169.254.169.254/latest/meta-data/. By appending specific paths or keys to this URL, you can retrieve different types of metadata. For example, http://169.254.169.254/latest/meta-data/instance-id retrieves the instance ID.

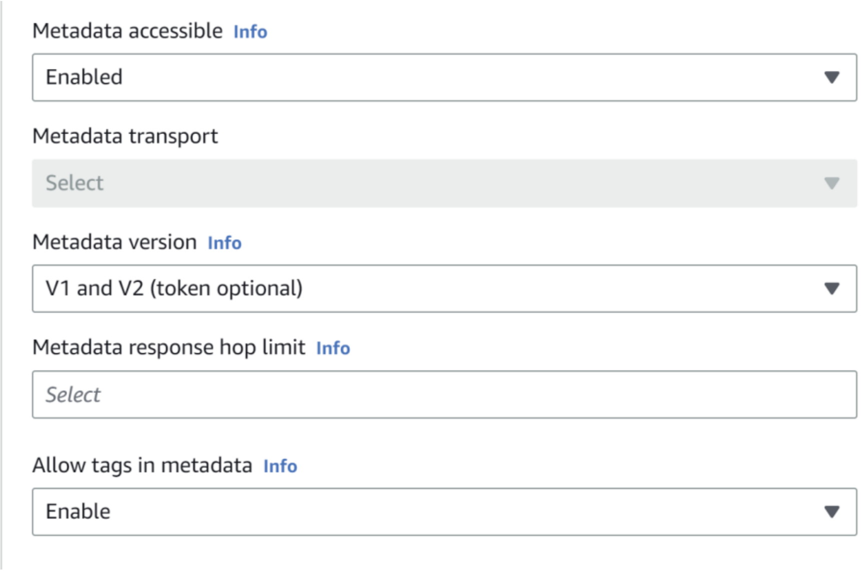
Metadata in AWS goes beyond EC2 instances. Various AWS services have their own metadata that can be accessed in different ways. For example, AWS Identity and Access Management (IAM) roles have associated metadata that provides details about the role and its permissions.

Overall, metadata in AWS plays an important role in providing information about resources, enabling dynamic adaptation of applications, and facilitating the interaction and management of AWS services.

Metadata url:

```
curl http://169.254.169.254/latest/meta-data
```

For getting metadata need to enable the following option when create a ec2 instance and we can get this under advanced details section



Metadata accessible Info

Enabled

Metadata transport

Select

Metadata version Info

V1 and V2 (token optional)

Metadata response hop limit Info

Select

Allow tags in metadata Info

Enable

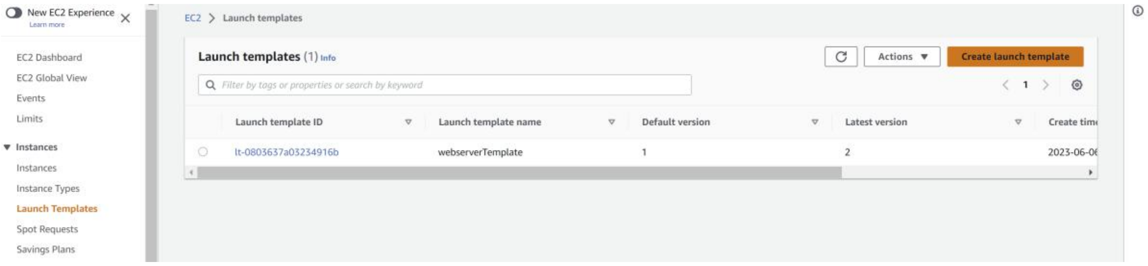
Launch Template:

An EC2 instance launch template is a configuration template that defines the settings for launching EC2 instances within an AWS account. It simplifies the process of launching instances by allowing you to create reusable templates with predefined configurations.

When you create a launch template, you can specify various parameters, including:

1. AMI (Amazon Machine Image): The template can specify the Amazon Machine Image (AMI) to use as the base for the instance.
2. Instance type: You can define the instance type, such as t2.micro, m5.large, etc., which determines the hardware specifications (CPU, memory, storage, etc.) of the EC2 instance.
3. Network settings: You can configure the networking aspects, including the VPC (Virtual Private Cloud), subnet, security groups, and IP addressing.
4. Storage options: Launch templates allow you to define the storage options for your instances, such as the root volume size, EBS (Elastic Block Store) settings, and instance store volumes.
5. Security settings: You can specify the security groups and IAM (Identity and Access Management) roles to associate with the instances launched from the template.
6. User data and startup scripts: Launch templates support adding user data, which can be used to run scripts or execute commands when the instance starts.
7. Instance tags: You can include tags to categorize and manage your instances more effectively.

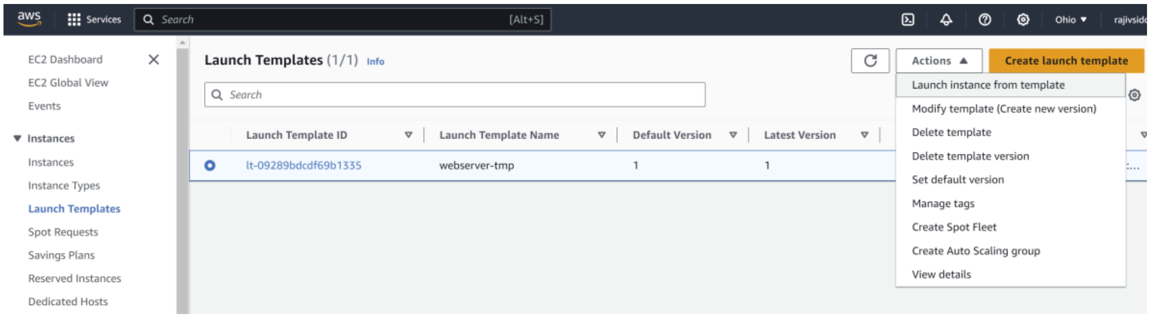
By using launch templates, you can create a standardized configuration for your instances and launch them easily with consistent settings. This helps automate and streamline the process of deploying EC2 instances within your AWS infrastructure. Additionally, launch templates are compatible with other AWS services such as Auto Scaling, allowing you to scale your instances efficiently based on demand.



Launch templates (1) Info

	Launch template ID	Launch template name	Default version	Latest version	Create tim
○	lt-0803637a03234916b	webserverTemplate	1	2	2023-06-0

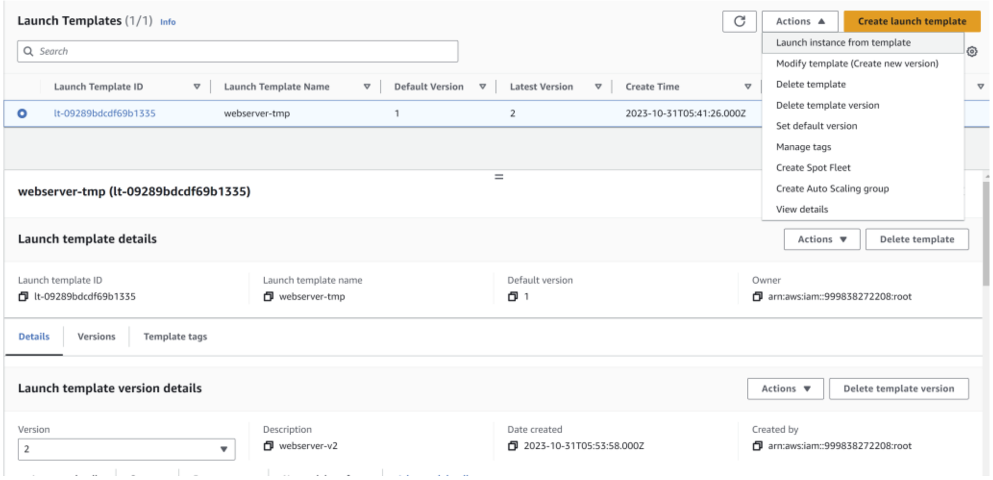
now launch an instance from that template



Launch Templates (1/1) Info

	Launch Template ID	Launch Template Name	Default Version	Latest Version
●	lt-09289bdcdf69b1335	webserver-tmp	1	1

Modify the launch template then launch an instance from that template's new version.



Launch Templates (1/1) Info

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time
lt-09289bdcdf69b1335	webserver-tmp	1	2	2023-10-31T05:41:26.000Z

webserver-tmp (lt-09289bdcdf69b1335)

Launch template details

Launch template ID: lt-09289bdcdf69b1335

Launch template name: webserver-tmp

Default version: 1

Owner: arn:aws:iam::999838272208:root

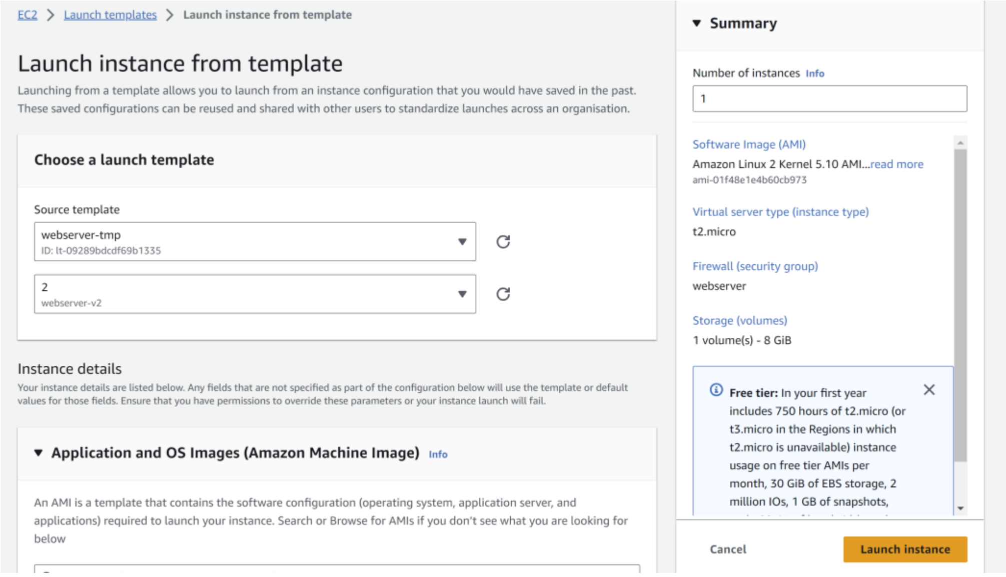
Launch template version details

Version: 2

Description: webserver-v2

Date created: 2023-10-31T05:53:58.000Z

Created by: arn:aws:iam::999838272208:root



Launch instance from template

Launching from a template allows you to launch from an instance configuration that you would have saved in the past. These saved configurations can be reused and shared with other users to standardize launches across an organisation.

Choose a launch template

Source template: webserver-tmp
ID: lt-09289bdcdf69b1335

2
webserver-v2

Instance details

Your instance details are listed below. Any fields that are not specified as part of the configuration below will use the template or default values for those fields. Ensure that you have permissions to override these parameters or your instance launch will fail.

▼ 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

▼ Summary

Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-01f48e1e4b60cb973

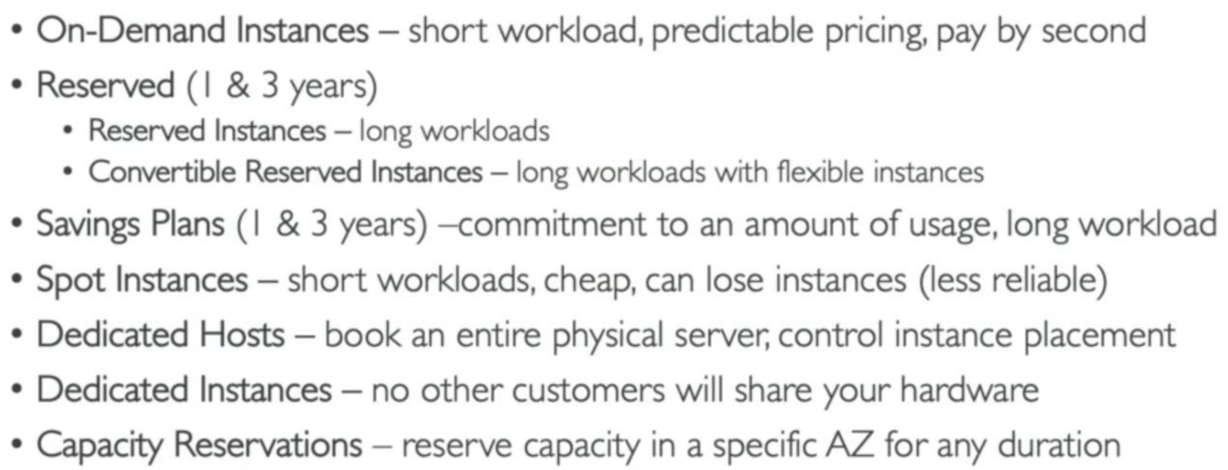
Virtual server type (instance type)
t2.micro

Firewall (security group)
webserver

Storage (volumes)
1 volume(s) - 8 GiB

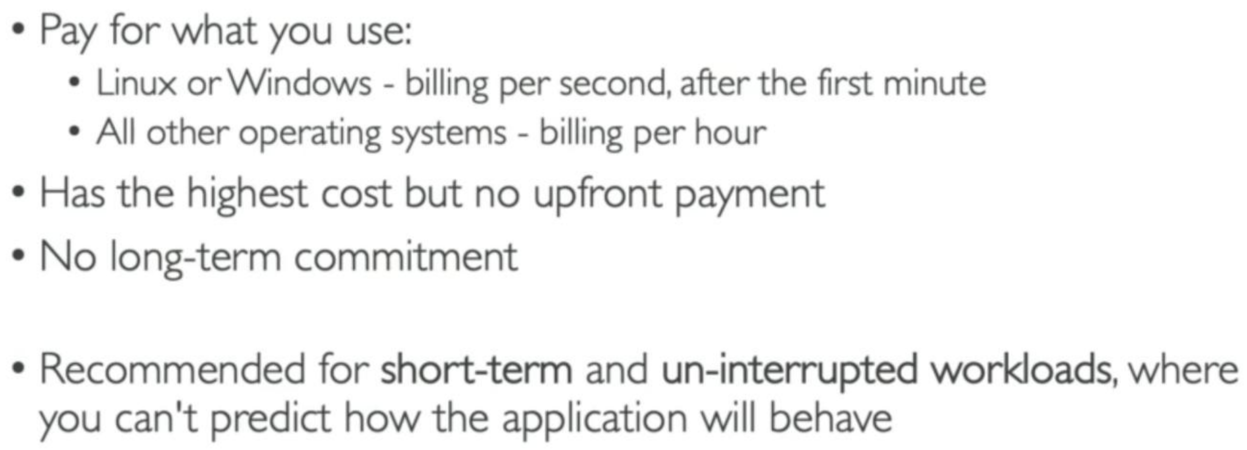
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots,

Ec2 instance purchasing options:



- **On-Demand Instances** – short workload, predictable pricing, pay by second
- **Reserved** (1 & 3 years)
  - **Reserved Instances** – long workloads
  - **Convertible Reserved Instances** – long workloads with flexible instances
- **Savings Plans** (1 & 3 years) –commitment to an amount of usage, long workload
- **Spot Instances** – short workloads, cheap, can lose instances (less reliable)
- **Dedicated Hosts** – book an entire physical server, control instance placement
- **Dedicated Instances** – no other customers will share your hardware
- **Capacity Reservations** – reserve capacity in a specific AZ for any duration

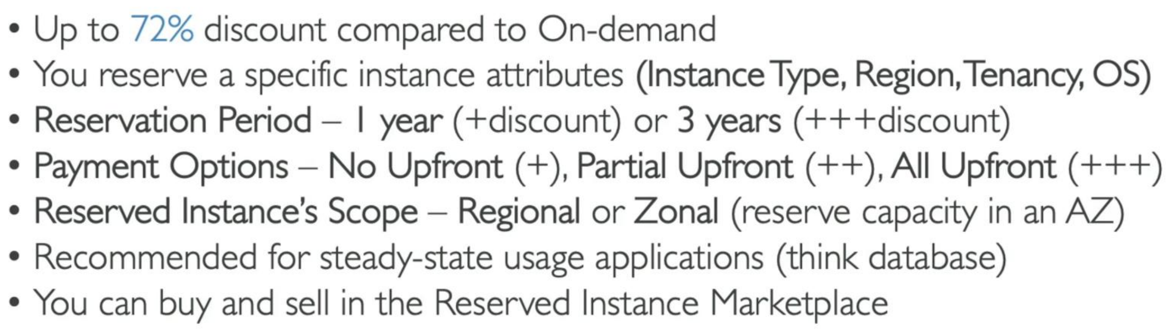
Ec2 On Demand



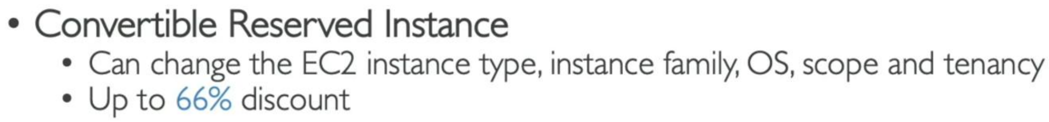
- Pay for what you use:
  - Linux or Windows - billing per second, after the first minute
  - All other operating systems - billing per hour
- Has the highest cost but no upfront payment
- No long-term commitment

- Recommended for **short-term** and **un-interrupted workloads**, where you can't predict how the application will behave

Ec2 Reserved instance

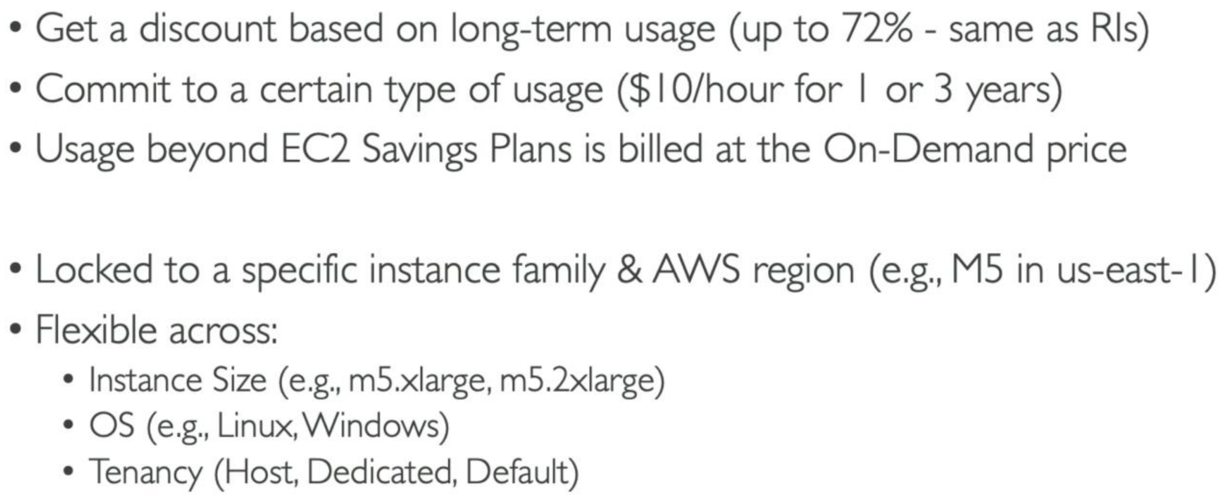


- Up to 72% discount compared to On-demand
- You reserve a specific instance attributes **(Instance Type, Region, Tenancy, OS)**
- **Reservation Period** – **1 year** (+discount) or **3 years** (+++discount)
- **Payment Options** – **No Upfront** (+), **Partial Upfront** (++), **All Upfront** (+++)
- **Reserved Instance's Scope** – **Regional** or **Zonal** (reserve capacity in an AZ)
- Recommended for steady-state usage applications (think database)
- You can buy and sell in the Reserved Instance Marketplace



- Convertible Reserved Instance
  - Can change the EC2 instance type, instance family, OS, scope and tenancy
  - Up to 66% discount

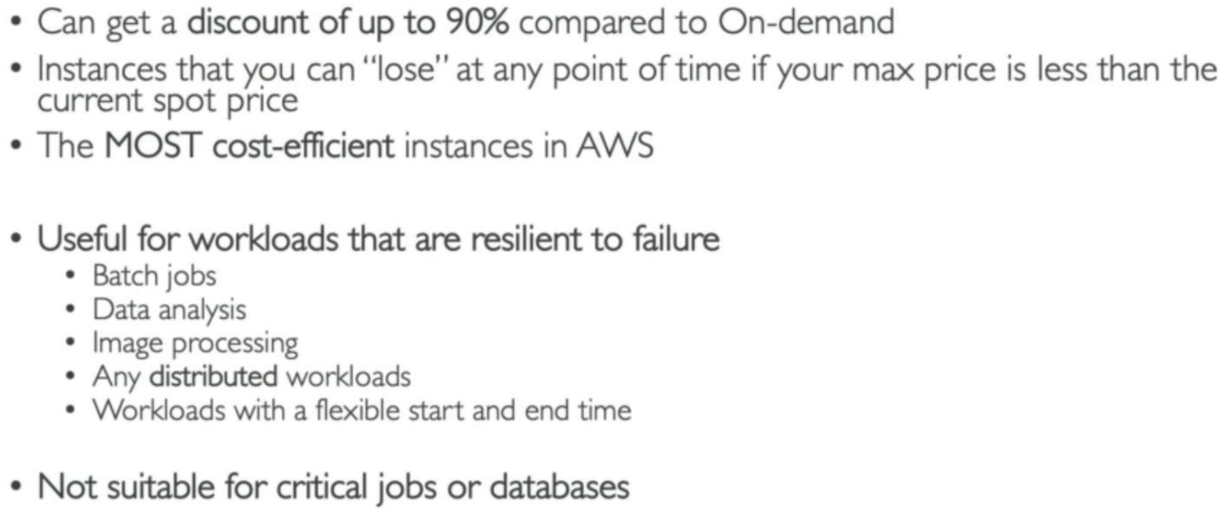
Ec2 saving plane



- Get a discount based on long-term usage (up to 72% - same as RIs)
- Commit to a certain type of usage ($10/hour for 1 or 3 years)
- Usage beyond EC2 Savings Plans is billed at the On-Demand price

- Locked to a specific instance family & AWS region (e.g., M5 in us-east-1)
- Flexible across:
  - Instance Size (e.g., m5.xlarge, m5.2xlarge)
  - OS (e.g., Linux, Windows)
  - Tenancy (Host, Dedicated, Default)

Ec2 spot instance

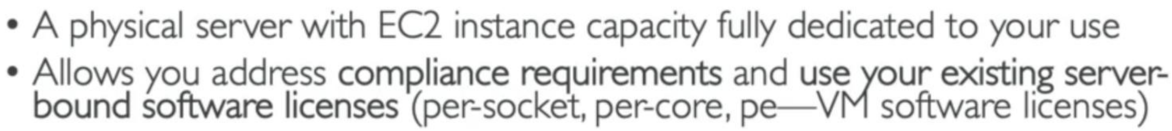


- Can get a **discount of up to 90%** compared to On-demand
- Instances that you can "lose" at any point of time if your max price is less than the current spot price
- The **MOST cost-efficient** instances in AWS

- **Useful for workloads that are resilient to failure**
  - Batch jobs
  - Data analysis
  - Image processing
  - Any **distributed** workloads
  - Workloads with a flexible start and end time

- **Not suitable for critical jobs or databases**

Ec2 dedicated host:



- A physical server with EC2 instance capacity fully dedicated to your use
- Allows you address **compliance requirements** and **use your existing server-bound software licenses** (per-socket, per-core, pe—VM software licenses)

- Purchasing Options:
  - **On-demand** – pay per second for active Dedicated Host
  - **Reserved** - 1 or 3 years (No Upfront, Partial Upfront, All Upfront)
- The most expensive option

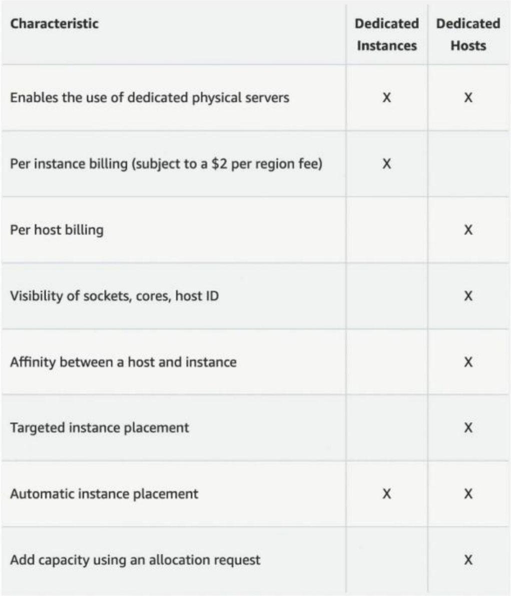
- Useful for software that have complicated licensing model (BYOL – Bring Your Own License)
- Or for companies that have strong regulatory or compliance needs

Ec2 Dedicated instances:

- Instances run on hardware that's dedicated to you

- May share hardware with other instances in same account

- No control over instance placement (can move hardware after Stop / Start)



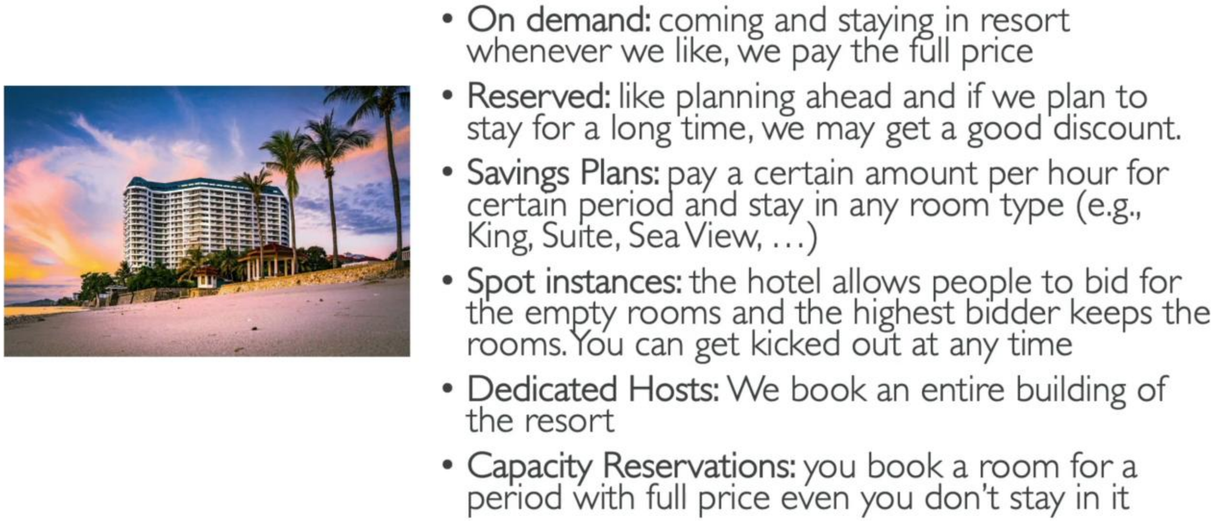
Characteristic	Dedicated Instances	Dedicated Hosts
Enables the use of dedicated physical servers	X	X
Per instance billing (subject to a $2 per region fee)	X	
Per host billing		X
Visibility of sockets, cores, host ID		X
Affinity between a host and instance		X
Targeted instance placement		X
Automatic instance placement	X	X
Add capacity using an allocation request		X

Ec2 capacity reservation

- Reserve **On-Demand** instances capacity in a specific AZ for any duration
- You always have access to EC2 capacity when you need it
- **No time commitment** (create/cancel anytime), **no billing discounts**
- Combine with Regional Reserved Instances and Savings Plans to benefit from billing discounts
- You're charged at On-Demand rate whether you run instances or not

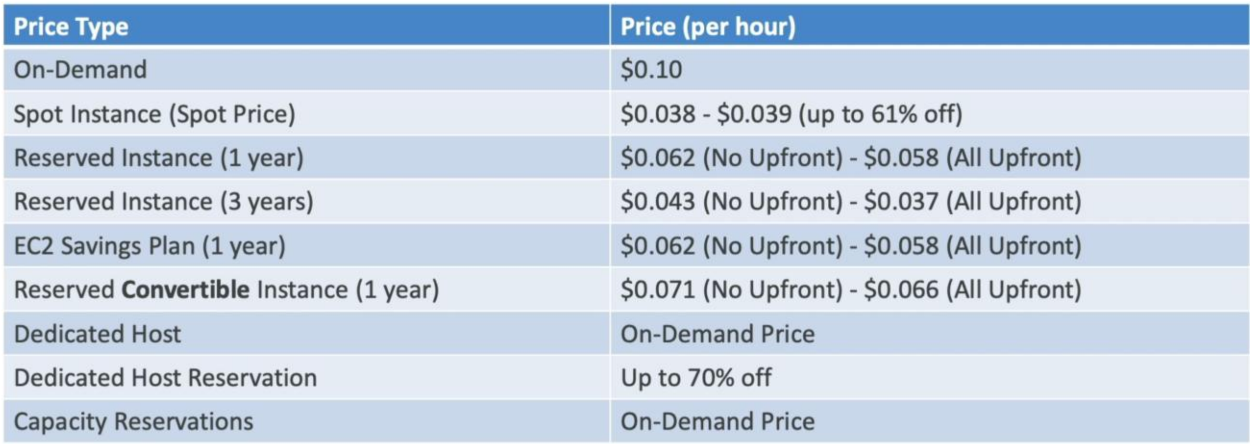
- Suitable for short-term, uninterrupted workloads that needs to be in a specific AZ

Explain with a practical example:



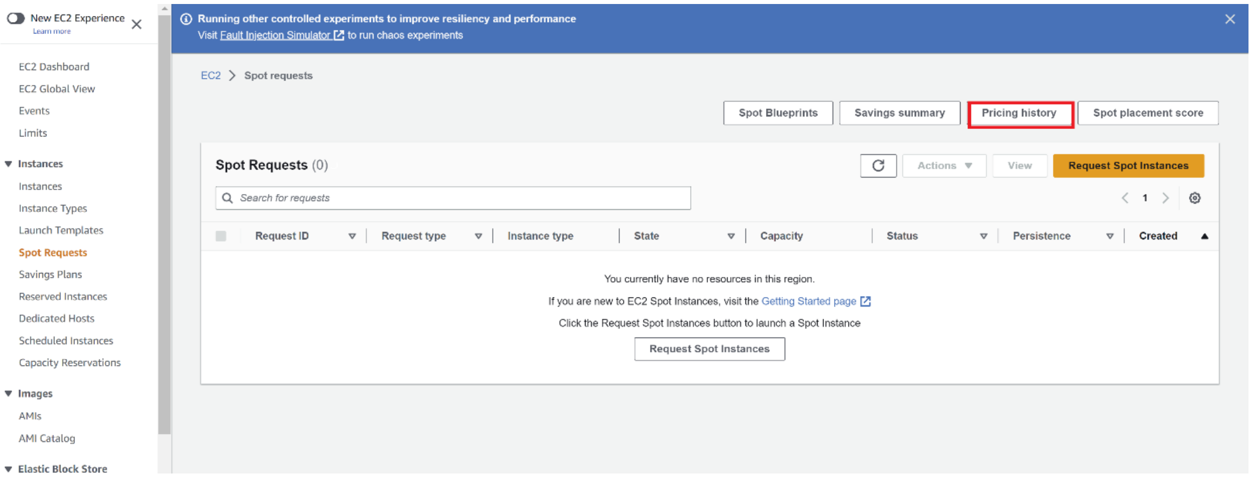
- **On demand:** coming and staying in resort whenever we like, we pay the full price
- **Reserved:** like planning ahead and if we plan to stay for a long time, we may get a good discount.
- **Savings Plans:** pay a certain amount per hour for certain period and stay in any room type (e.g., King, Suite, Sea View, …)
- **Spot instances:** the hotel allows people to bid for the empty rooms and the highest bidder keeps the rooms. You can get kicked out at any time
- **Dedicated Hosts:** We book an entire building of the resort
- **Capacity Reservations:** you book a room for a period with full price even you don't stay in it

Price comparison example – m4.large -us east 1



Price Type	Price (per hour)
On-Demand	$0.10
Spot Instance (Spot Price)	$0.038 - $0.039 (up to 61% off)
Reserved Instance (1 year)	$0.062 (No Upfront) - $0.058 (All Upfront)
Reserved Instance (3 years)	$0.043 (No Upfront) - $0.037 (All Upfront)
EC2 Savings Plan (1 year)	$0.062 (No Upfront) - $0.058 (All Upfront)
Reserved **Convertible** Instance (1 year)	$0.071 (No Upfront) - $0.066 (All Upfront)
Dedicated Host	On-Demand Price
Dedicated Host Reservation	Up to 70% off
Capacity Reservations	On-Demand Price

Spot instance



Running other controlled experiments to improve resiliency and performance
Visit Fault Injection Simulator to run chaos experiments

Spot Requests (0)

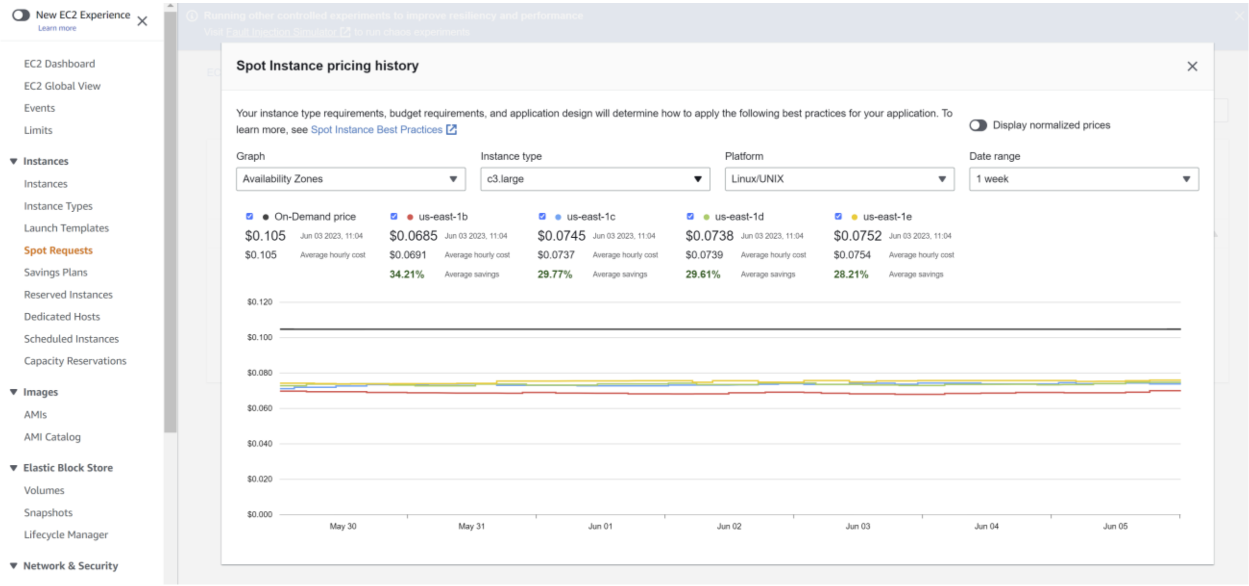
☐	Request ID	Request type	Instance type	State	Capacity	Status	Persistence	Created ▲

You currently have no resources in this region.

If you are new to EC2 Spot Instances, visit the Getting Started page

Click the Request Spot Instances button to launch a Spot Instance

Request Spot Instances



Spot Instance pricing history

Your instance type requirements, budget requirements, and application design will determine how to apply the following best practices for your application. To learn more, see Spot Instance Best Practices

Graph: Availability Zones
Instance type: c3.large
Platform: Linux/UNIX
Date range: 1 week

	☑ ● On-Demand price	☑ ● us-east-1b	☑ ● us-east-1c	☑ ● us-east-1d	☑ ● us-east-1e
Jun 03 2023, 11:04	$0.105	$0.0685	$0.0745	$0.0738	$0.0752
Average hourly cost	$0.105	$0.0691	$0.0737	$0.0739	$0.0754
Average savings		34.21%	29.77%	29.61%	28.21%

Amazon Machine Image(AMI):

In the context of Amazon Web Services (AWS), an AMI stands for Amazon Machine Image. An AMI is a pre-configured template that contains the necessary information to launch an EC2 instance. It includes the operating system, software packages, configuration settings, and any additional data required for the instance.

Here are some key points about AMIs in AWS:

1. Pre-built Templates: AMIs provide a starting point for launching instances. AWS offers a wide range of pre-built AMIs that are tailored for various use cases, including different operating systems (such as Amazon Linux, Ubuntu, Windows Server, etc.) and application-specific configurations.

2. Public and Private AMIs: AWS provides a public repository of AMIs that are available to all AWS users. These AMIs are created and shared by the AWS community or AWS itself. Additionally, you can create your own private AMIs by customizing and saving an existing EC2 instance as an image.

3. Customization and Configuration: You can customize an AMI to meet your specific requirements. This involves installing additional software, applying updates, configuring security settings, and making any desired changes to the base image. Once the modifications are made, you can save the customized instance as a new AMI.

4. Versioning and Lifecycle: AMIs support versioning, allowing you to manage and track different iterations of an image. You can create new versions of an AMI as you make updates or changes to the underlying configuration. Additionally, you can control the lifecycle of an AMI, including when to deprecate or deregister an image.

5. Marketplace AMIs: AWS Marketplace offers a wide selection of AMIs from third-party vendors. These AMIs may come with pre-installed software, databases, or complete application stacks, enabling you to quickly deploy complex solutions without having to manually configure them.

To launch an EC2 instance, you typically select an appropriate AMI, specify the instance type, configure networking, security settings, and storage options. By leveraging AMIs, you can easily replicate and deploy instances with the desired operating system and software configurations.

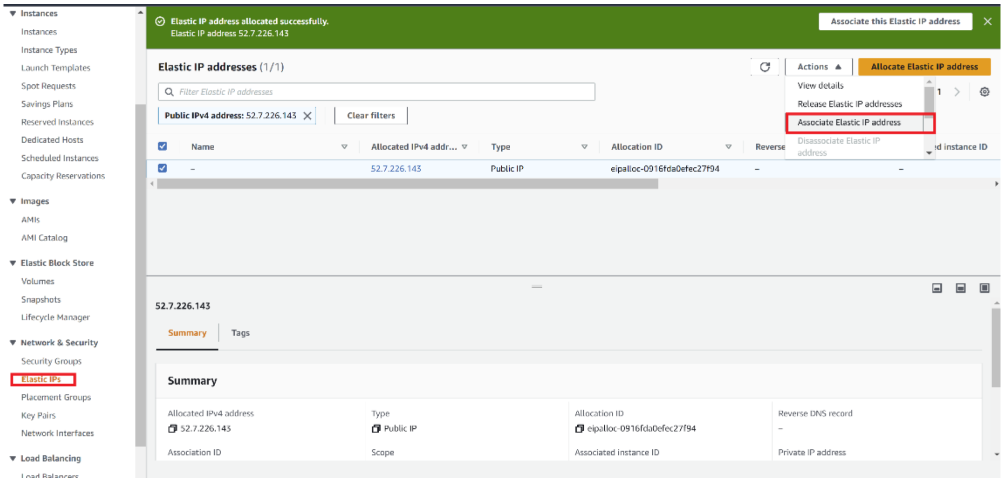
Elastic ip:

Add a new elastic ip

Ec2>Network & security>Elastic Ips>click allocate elastic ip button>click allocate

Attach this elastic ip to a ec2 instance

Ec2>Network & security>Elastic Ips>select the elastic ip>click Action >click Associate Elastic ip



Elastic IP address allocated successfully.
Elastic IP address 52.7.226.143

Elastic IP addresses (1/1)

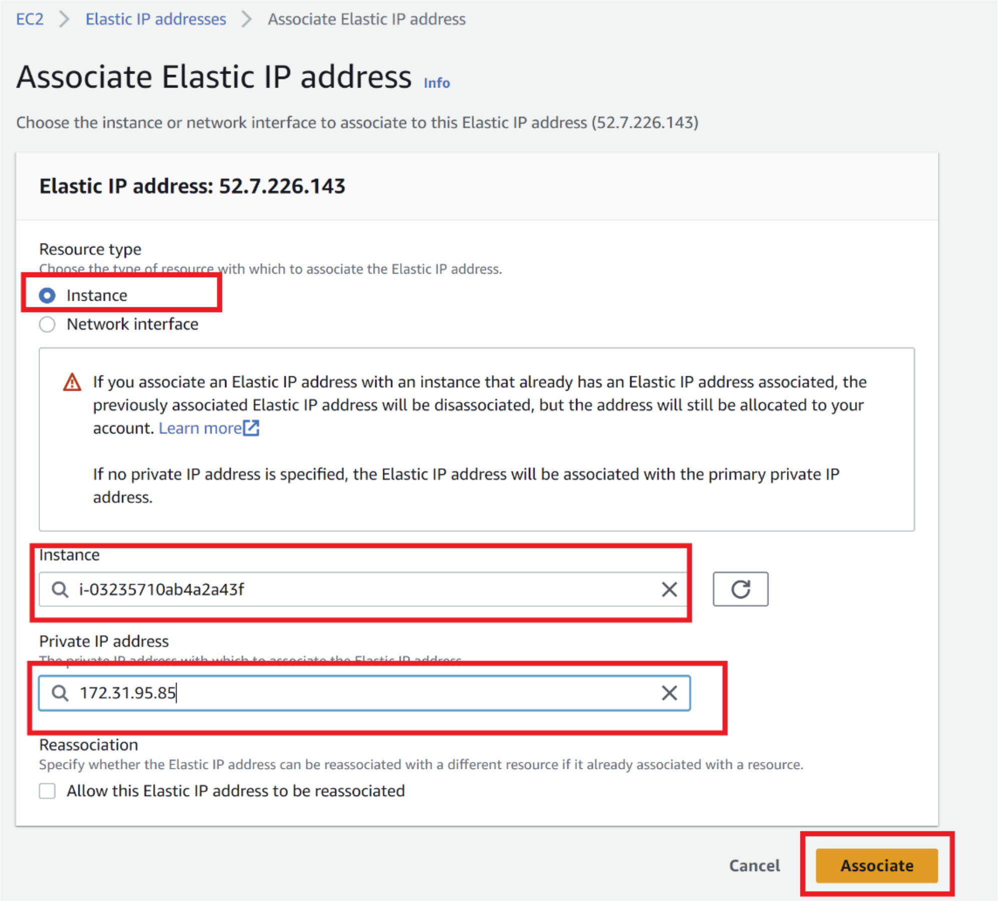
Public IPv4 address: 52.7.226.143

☑	Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse
☑	–	52.7.226.143	Public IP	eipalloc-0916fda0efec27f94	–

52.7.226.143

Summary

Allocated IPv4 address	Type	Allocation ID	Reverse DNS record
52.7.226.143	Public IP	eipalloc-0916fda0efec27f94	–
Association ID	Scope	Associated instance ID	Private IP address



EC2 > Elastic IP addresses > Associate Elastic IP address

Associate Elastic IP address Info

Choose the instance or network interface to associate to this Elastic IP address (52.7.226.143)

Elastic IP address: 52.7.226.143

Resource type

Choose the type of resource with which to associate the Elastic IP address.

● Instance

○ Network interface

⚠ If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. Learn more

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

Instance

i-03235710ab4a2a43f

Private IP address

The private IP address with which to associate the Elastic IP address.

172.31.95.85

Reassociation

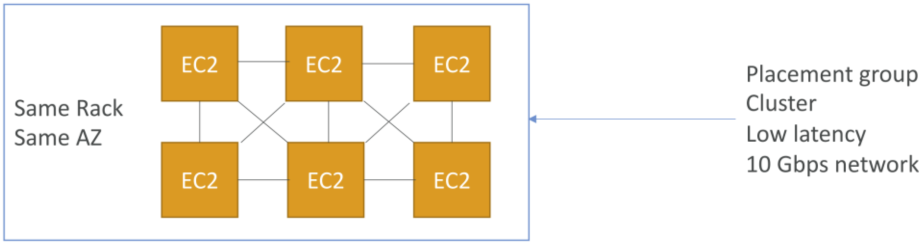
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

□ Allow this Elastic IP address to be reassociated

Placement Groups:

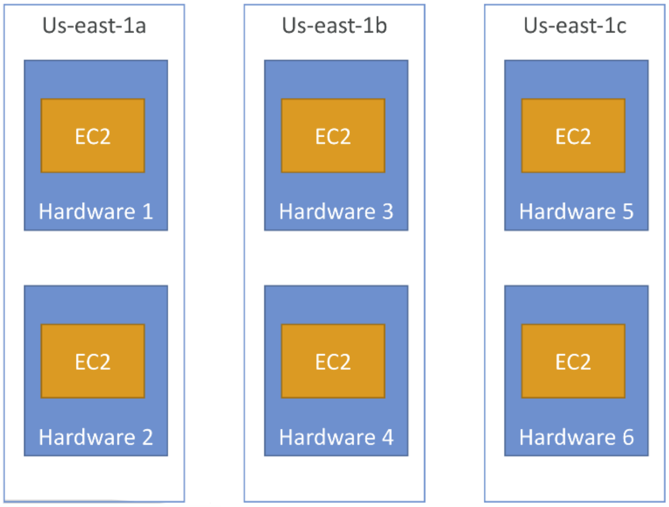
- Sometimes you want control over the EC2 Instance placement strategy
- That strategy can be defined using placement groups
- When you create a placement group, you specify one of the following strategies for the group:
  - *Cluster*—clusters instances into a low-latency group in a single Availability Zone
  - *Spread*—spreads instances across underlying hardware (max 7 instances per group per AZ)
  - *Partition*—spreads instances across many different partitions (which rely on different sets of racks) within an AZ. Scales to 100s of EC2 instances per group (Hadoop, Cassandra, Kafka)

Placement Groups-Cluster



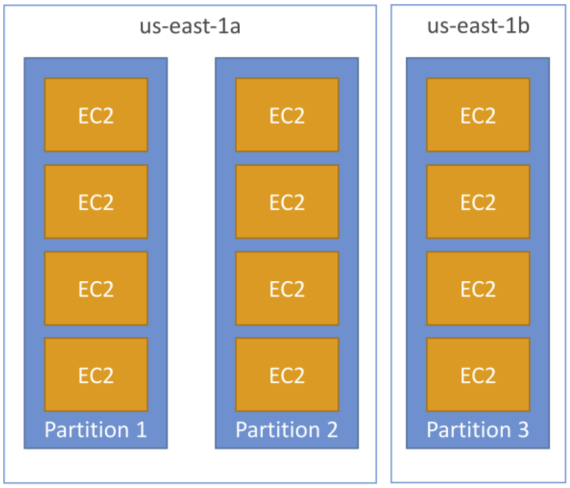
- Pros: Great network (10 Gbps bandwidth between instances with Enhanced Networking enabled - recommended)
- Cons: If the rack fails, all instances fails at the same time
- Use case:
  - Big Data job that needs to complete fast
  - Application that needs extremely low latency and high network throughput

Placement Groups- Spread



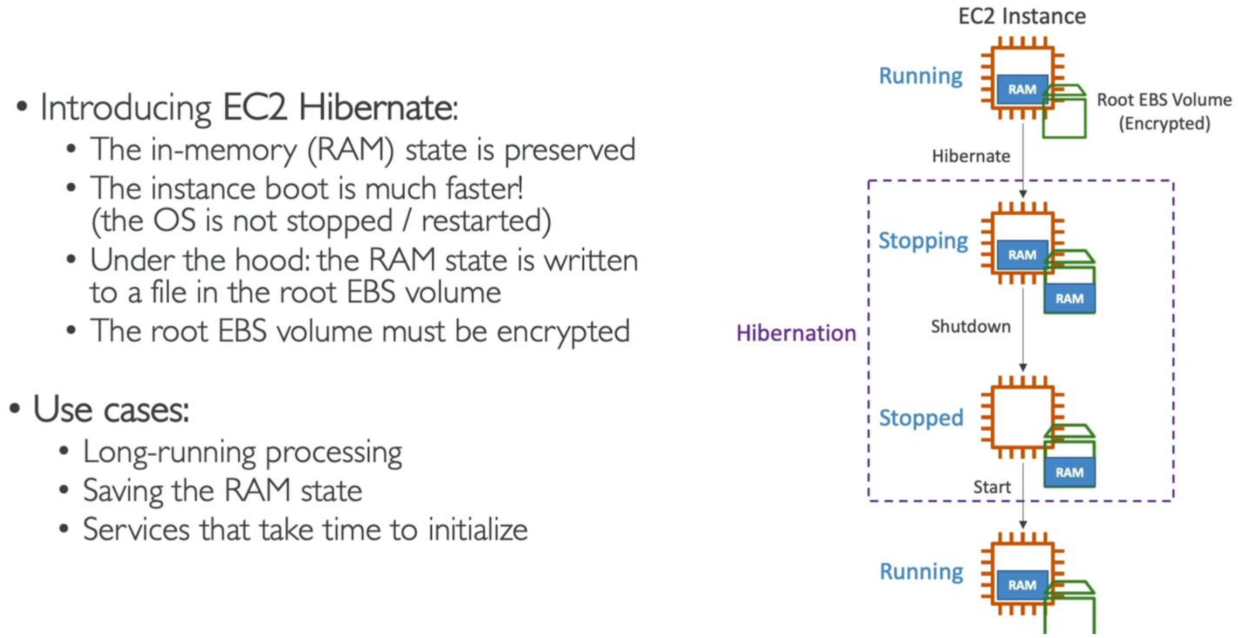
- Pros:
  - Can span across Availability Zones (AZ)
  - Reduced risk is simultaneous failure
  - EC2 Instances are on different physical hardware
- Cons:
  - Limited to 7 instances per AZ per placement group
- Use case:
  - Application that needs to maximize high availability
  - Critical Applications where each instance must be isolated from failure from each other

Placement Groups- Partition



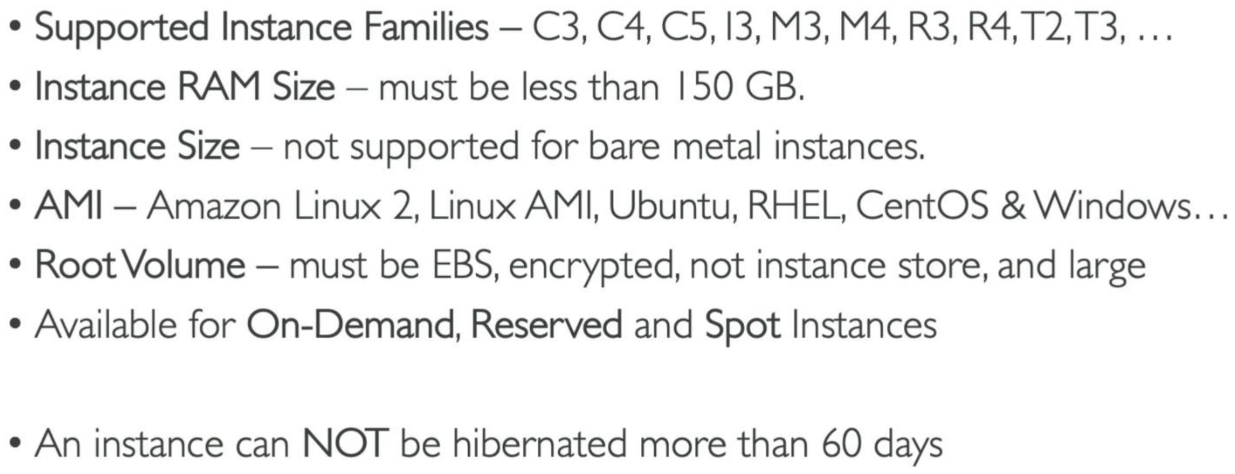
- Up to 7 partitions per AZ
- Can span across multiple AZs in the same region
- Up to 100s of EC2 instances
- The instances in a partition do not share racks with the instances in the other partitions
- A partition failure can affect many EC2 but won't affect other partitions
- EC2 instances get access to the partition information as metadata
- Use cases: HDFS, HBase, Cassandra, Kafka

Ec2 hibernate:



- Introducing **EC2 Hibernate:**
  - The in-memory (RAM) state is preserved
  - The instance boot is much faster! (the OS is not stopped / restarted)
  - Under the hood: the RAM state is written to a file in the root EBS volume
  - The root EBS volume must be encrypted
- Use cases:
  - Long-running processing
  - Saving the RAM state
  - Services that take time to initialize

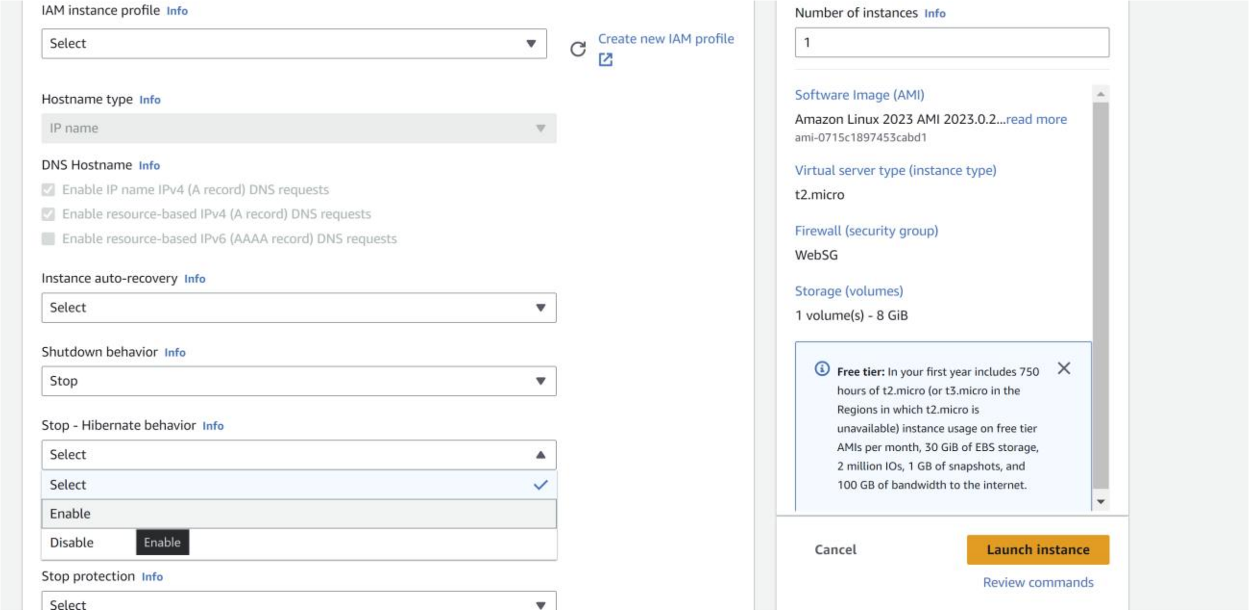
Hibernate- Good to know



- **Supported Instance Families** – C3, C4, C5, I3, M3, M4, R3, R4, T2, T3, …
- **Instance RAM Size** – must be less than 150 GB.
- **Instance Size** – not supported for bare metal instances.
- **AMI** – Amazon Linux 2, Linux AMI, Ubuntu, RHEL, CentOS & Windows…
- **Root Volume** – must be EBS, encrypted, not instance store, and large
- Available for **On-Demand**, **Reserved** and **Spot** Instances

- An instance can **NOT** be hibernated more than 60 days

Enable hibernate when launching an ec2 instance and follow showing the steps.
ec2>launch instance> scroll down and advanced details



IAM instance profile Info

Select

Create new IAM profile

Hostname type Info

IP name

DNS Hostname Info

☑ Enable IP name IPv4 (A record) DNS requests

☑ Enable resource-based IPv4 (A record) DNS requests

□ Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery Info

Select

Shutdown behavior Info

Stop

Stop - Hibernate behavior Info

Select

- Select ✓
- Enable
- Disable

Stop protection Info

Select

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.0.2...read more
ami-0715c1897453cabd1

Virtual server type (instance type)

t2.micro

Firewall (security group)

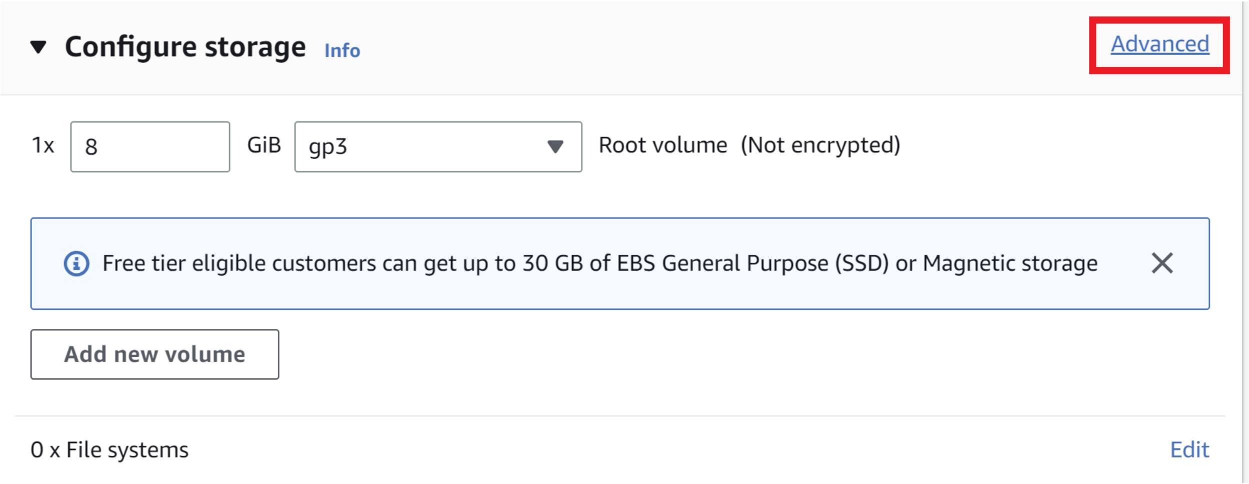
WebSG

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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

To enable hibernation, space is allocated on the root volume to store the instance memory (RAM). Make sure that the **root volume is large enough to store** the RAM contents and accommodate your expected usage, e.g. OS, applications. To use hibernation, the **root volume must be an encrypted** EBS volume.

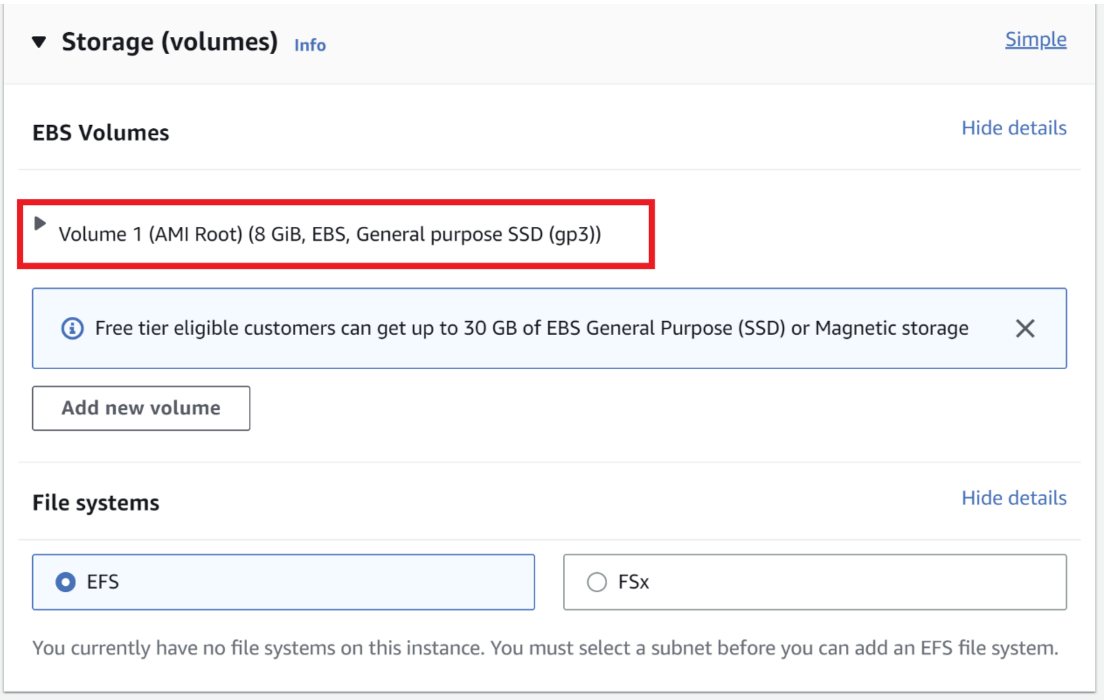


▼ Configure storage Info

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

0 x File systems



▼ Storage (volumes)

EBS Volumes

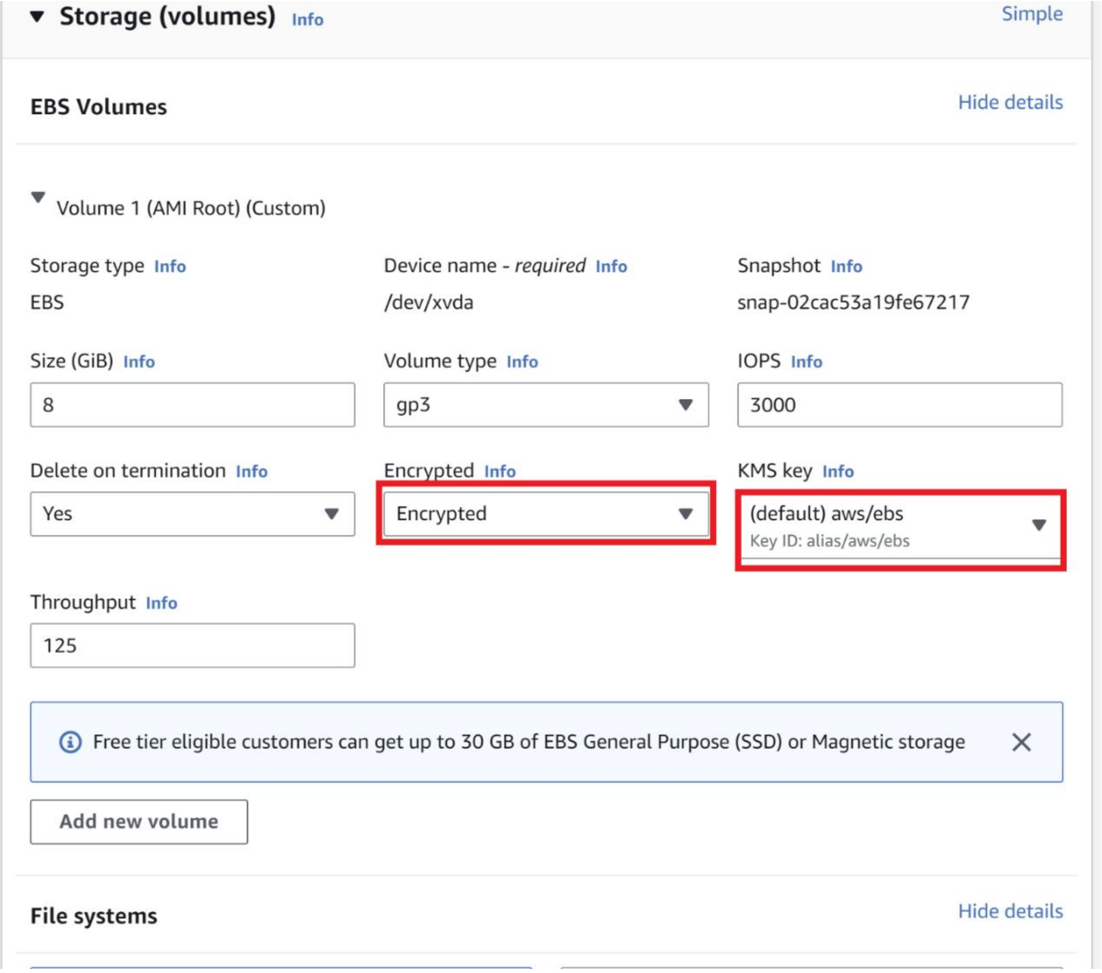
▶ Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

File systems

● EFS ○ FSx

You currently have no file systems on this instance. You must select a subnet before you can add an EFS file system.



▼ Storage (volumes)

EBS Volumes

▼ Volume 1 (AMI Root) (Custom)

Storage type: EBS

Device name - *required*: /dev/xvda

Snapshot: snap-02cac53a19fe67217

Size (GiB): 8

Volume type: gp3

IOPS: 3000

Delete on termination: Yes

Encrypted: Encrypted

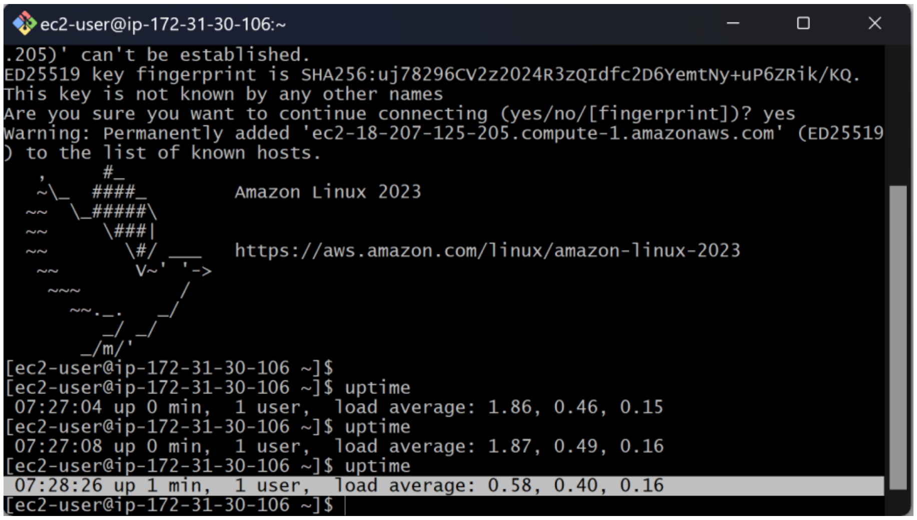
KMS key: (default) aws/ebs
Key ID: alias/aws/ebs

Throughput: 125

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

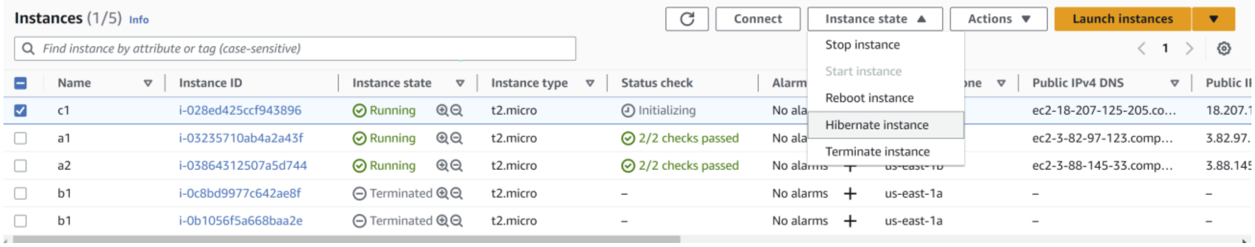
File systems

Now launch the instance and login and type the following command for checking the system uptime #uptime



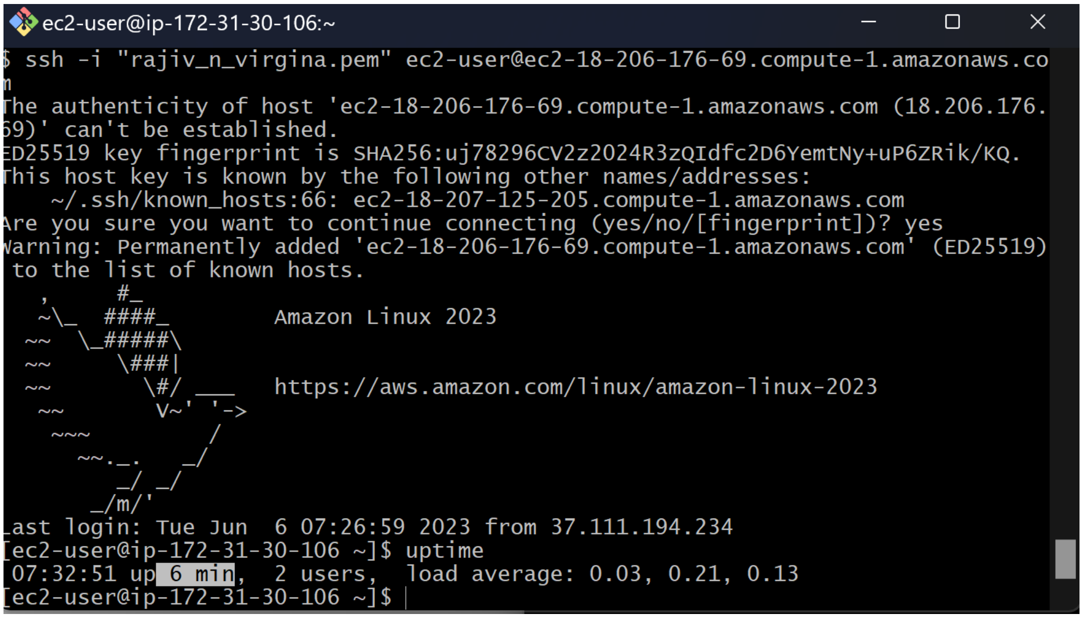
```
.205)' can't be established.
ED25519 key fingerprint is SHA256:uj78296CV2z2024R3zQIdfc2D6YemtNy+uP6ZRik/KQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-207-125-205.compute-1.amazonaws.com' (ED25519
) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-30-106 ~]$
[ec2-user@ip-172-31-30-106 ~]$ uptime
 07:27:04 up 0 min,  1 user,  load average: 1.86, 0.46, 0.15
[ec2-user@ip-172-31-30-106 ~]$ uptime
 07:27:08 up 0 min,  1 user,  load average: 1.87, 0.49, 0.16
[ec2-user@ip-172-31-30-106 ~]$ uptime
 07:28:26 up 1 min,  1 user,  load average: 0.58, 0.40, 0.16
[ec2-user@ip-172-31-30-106 ~]$
```

Now hibernate the instance as shown the following picture



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm	ne	Public IPv4 DNS	Public II
☑	c1	i-028ed425ccf943896	✓ Running	t2.micro	Initializing	No ala		ec2-18-207-125-205.co...	18.207.1
☐	a1	i-03235710ab4a2a43f	✓ Running	t2.micro	✓ 2/2 checks passed	No ala		ec2-3-82-97-123.comp...	3.82.97.
☐	a2	i-03864312507a5d744	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +		ec2-3-88-145-33.comp...	3.88.145
☐	b1	i-0c8bd9977c642ae8f	Terminated	t2.micro	–	No alarms +	us-east-1a	–	–
☐	b1	i-0b1056f5a668baa2e	Terminated	t2.micro	–	No alarms +	us-east-1a	–	–

Now start the instance and login and check the uptime and you can see it will showing more uptime



```
$ ssh -i "rajiv_n_virgina.pem" ec2-user@ec2-18-206-176-69.compute-1.amazonaws.co
m
The authenticity of host 'ec2-18-206-176-69.compute-1.amazonaws.com (18.206.176.
69)' can't be established.
ED25519 key fingerprint is SHA256:uj78296CV2z2024R3zQIdfc2D6YemtNy+uP6ZRik/KQ.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:66: ec2-18-207-125-205.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-206-176-69.compute-1.amazonaws.com' (ED25519)
 to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Jun  6 07:26:59 2023 from 37.111.194.234
[ec2-user@ip-172-31-30-106 ~]$ uptime
 07:32:51 up 6 min,  2 users,  load average: 0.03, 0.21, 0.13
[ec2-user@ip-172-31-30-106 ~]$
```