Amazon EC2 (Linux Edition)

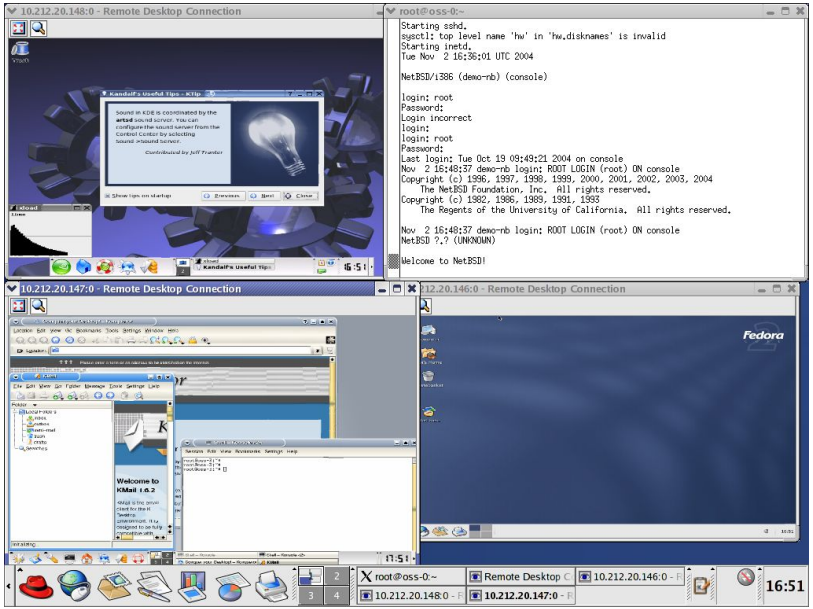
What Is Amazon EC2?

Amazon EC2 is AWS primary web service that **provides resizable compute capacity in the cloud**.

It allows users to **rent virtual computers** on which to run their own computer applications

EC2 uses Xen hypervisor which providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently. It was developed by the Linux Foundation and is supported by Intel.

The **hypervisor takes care of CPU scheduling and memory partitioning**, but it is unaware of networking, external storage devices, video, or any other common I/O functions found on a computing system.



Features of EC2

- Virtual computing environments, known as ***instances***
- Preconfigured templates for your instances, known as ***Amazon Machine Images (AMIs)***, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as ***instance types***
- Secure login information for your instances using ***key pairs*** (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as ***instance store volumes***
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as ***Amazon EBS volumes***
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as ***regions*** **and** ***Availability Zones***
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using ***security groups***
- Static IPv4 addresses for dynamic cloud computing, known as ***Elastic IP addresses***
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as *virtual private clouds* (**VPCs**)

Amazon EC2 Instance Types

There are dozens of instance types available, varying in the following dimensions:

- Virtual CPUs (vCPUs)
- Memory
- Storage (size and type)
- Network performancE

Use Case	Instance Types
General Purpose	T2, M3, M4
Compute Optimized	C4, C3
Memory Optimized	X1, R4, R3
Accelerated Computing	P2, G3, F1
Storage Optimized	I3, D2

Get Started with Amazon EC2

- Get Up and Running
  - Setting Up with Amazon EC2
  - Getting Started with Amazon EC2 Linux Instances
- Basics
  - Instances and AMIs
  - Regions and Availability Zones
  - Instance Types
  - Tags
- Networking and Security
  - Amazon EC2 Key Pairs
  - Security Groups
  - Elastic IP Addresses
  - Amazon EC2 and Amazon VPC
- Storage
  - Amazon EBS
  - Instance Store

Practical : Setting Up with Amazon EC2

Setting Up with Amazon EC2

- Sign Up for AWS
- Create an IAM User
  - Policy / AdministratorAccess / User Group
- Create a Key Pair

Create a Key Pair

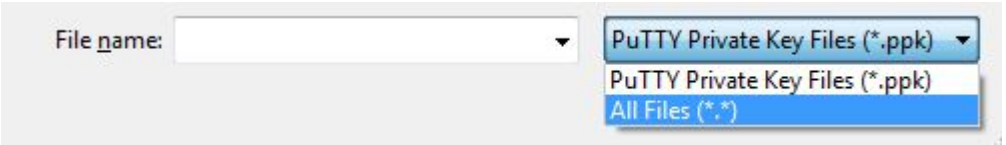
AWS uses public-key cryptography to secure the login information for your instance

1. From the AWS dashboard, choose **EC2** to open the Amazon EC2 console.
2. From the navigation bar, select a region for the key pair. **Key pairs are specific to a region**; for example, if you plan to launch an instance in the US West (Oregon) Region, you must create a key pair for the instance in the US West (Oregon) Region.
3. In the navigation pane, under NETWORK & SECURITY, click Key Pairs.
4. Click Create Key Pair
5. Enter a name for the new key pair in the Key pair name field of the Create Key Pair dialog box, and then click Create
6. The private key file is automatically downloaded by your browser. **The file name extension is .pem**.
7. ***chmod 400 your_user_name-key-pair-region_name.pem***
8. ssh -i ***our_user_name-key-pair-region_name.pem*** ubuntu@publicipaddress

Windows Users

1. If you plan to use PuTTY, you'll need to install it and use the following procedure to convert the `.pem` file to a `.ppk` file.
2. Download and install PuTTY from http://www.chiark.greenend.org.uk/~sgtatham/putty/. Be sure to install the entire suite.
3. Start PuTTYgen (for example, from the **Start** menu, click **All Programs > PuTTY > PuTTYgen**).
4. Under **Type of key to generate**, select **SSH-2 RSA**.

5. Choose **Load**. By default, PuTTYgen displays only files with the extension `.ppk`. To locate your `.pem` file, select the option to display files of all types.



Windows Users

6. Select the private key file that you created in the previous procedure and choose **Open**. Choose **OK** to dismiss the confirmation dialog box.
7. Choose **Save private key**. PuTTYgen displays a warning about saving the key without a passphrase. Click **Yes**.
8. Specify the same name for the key that you used for the key pair. PuTTY automatically adds the `.ppk` file extension.

Create a Virtual Private Cloud (VPC)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation bar, select a region for the VPC. VPCs are specific to a region, so you should select the same region in which you created your key pair.
3. On the VPC dashboard, click **Start VPC Wizard**.
4. On the **Step 1: Select a VPC Configuration** page, ensure that **VPC with a Single Public Subnet** is selected, and click **Select**.
5. On the **Step 2: VPC with a Single Public Subnet** page, enter a friendly name for your VPC in the **VPC name**field. Leave the other default configuration settings, and click **Create VPC**. On the confirmation page, click **OK**.

Create a Security Group

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level.

Note that if you plan to launch instances in multiple regions, you'll need to create a security group in each region.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select a region for the security group. Security groups are specific to a region, so you should select the same region in which you created your key pair.
3. Click **Security Groups** in the navigation pane.
4. Click **Create Security Group**.
5. Enter a name for the new security group and a description. Choose a name that is easy for you to remember, such as your IAM user name, followed by _SG_, plus the region name. For example, *me*_SG_*uswest2*.
6. In the **VPC** list, select your VPC. If you have a default VPC, it's the one that is marked with an asterisk (*).

Create a Security Group

7. On the Inbound tab, create the following rules (click Add Rule for each new rule), and then click Create:
    a. Select HTTP from the Type list, and make sure that Source is set to Anywhere (0.0.0.0/0).
    b. Select HTTPS from the Type list, and make sure that Source is set to Anywhere (0.0.0.0/0).
    c. Select SSH from the Type list. In the Source box, choose My IP to automatically populate the field with the public IPv4 address of your local computer. Alternatively, choose Custom and specify the public IPv4 address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing suffix /32, for example, 203.0.113.25/32. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24.

Launching an Instance

- From https://console.aws.amazon.com/ec2/ choose Launch Instance.
- Choose an Amazon Machine Image (AMI) page displays a list of basic configurations, called Amazon Machine Images (AMIs). Select one eligible for "Free tier eligible."
- On the Choose an Instance Type page, you can select the hardware configuration of your instance. Select the t2.micro type, which is selected by default.
- Next: Configure Instance Details and follow the directions to select a subnet
- Choose Review and Launch
- On the Review Instance Launch page, under Security Groups **Select an existing security group**
- Review Instance Launch page, choose Launch
- When prompted for a key pair, select Choose an existing key pair

Connect to Your Instance

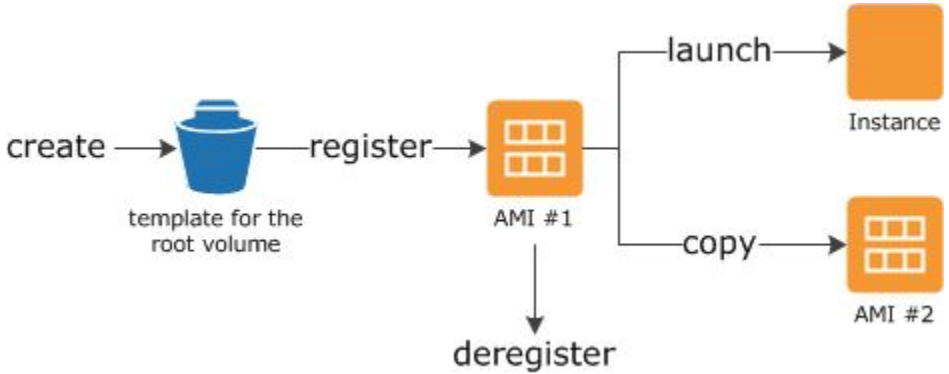
Amazon Machine Images (AMI)

An Amazon Machine Image (AMI) **provides the information required to launch an instance**, which is a virtual server in the cloud.

An AMI includes the following:

- A **template for the root volume for the instance** (for example, an operating system, an application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched

AMI Lifecycle

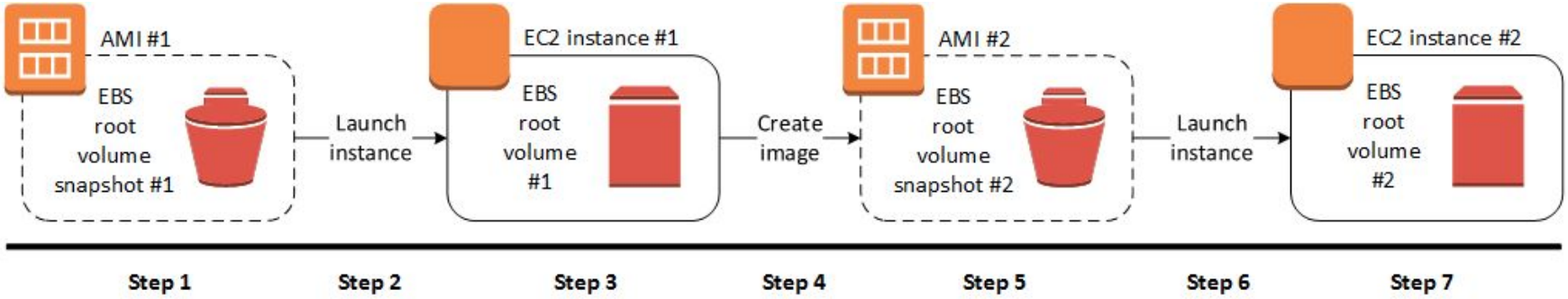


Creating Your Own AMI

You can customize the instance that you launch from a public AMI and then save that configuration as a custom AMI for your own use. Instances that you launch from your AMI use all the customizations that you've made.

The root storage device of the instance determines the process you follow to create an AMI. The root volume of an instance is either an Amazon EBS volume or an instance store volume.

In the navigation pane, choose Instances and select your instance. Choose Actions, Image, and Create



Tip

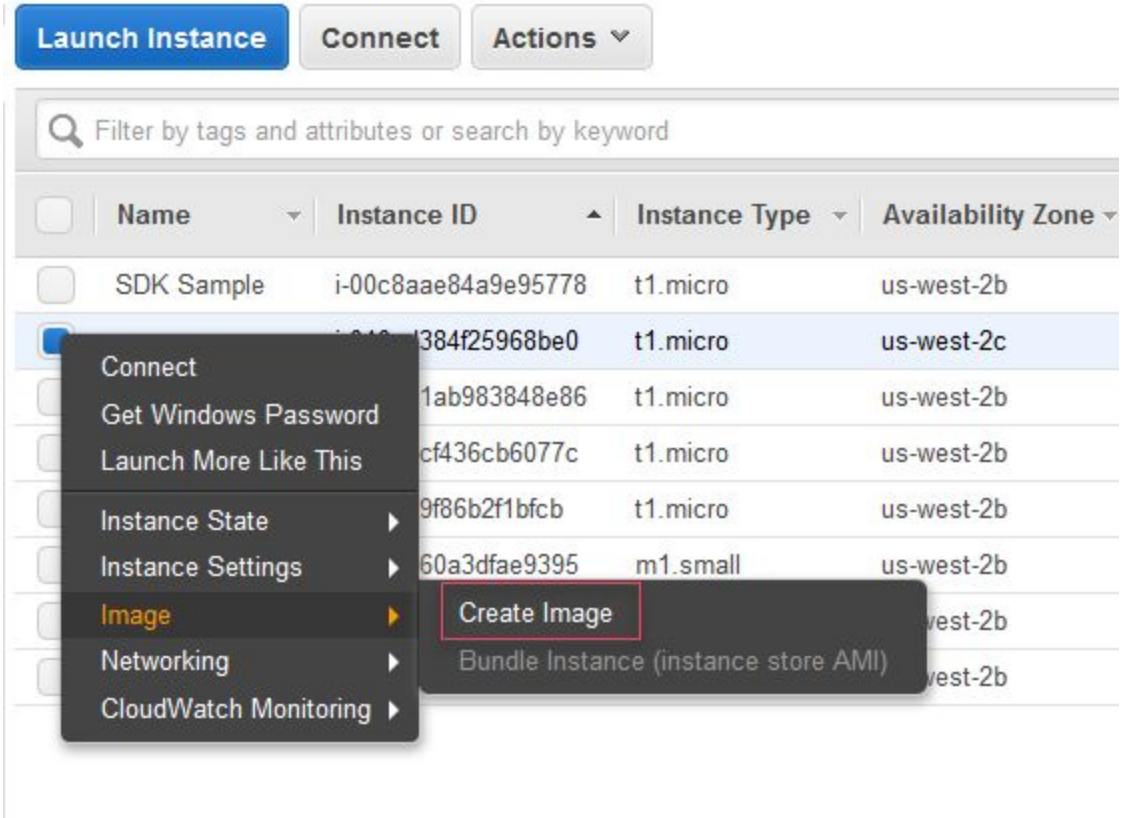
- If this option is disabled, your instance isn't an Amazon EBS-backed instance.
- By default, **Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance**
- If you choose No reboot, AWS can't guarantee the file system integrity of the created image.
- While your AMI is being created, you can choose **AMIs** in the navigation pane to view its status. Initially this will be `pending`. After a few minutes the status should change to `available`
- Choose **Snapshots** in the navigation pane to view the snapshot that was created for the new AMI. When you launch an instance from this AMI, AWS uses this snapshot to create its root device volume

AMI Sources

- Published by AWS
- The AWS Marketplace
- Generated from Existing Instances
- Uploaded Virtual Servers - Can import from existing VMs and launch it in EC2

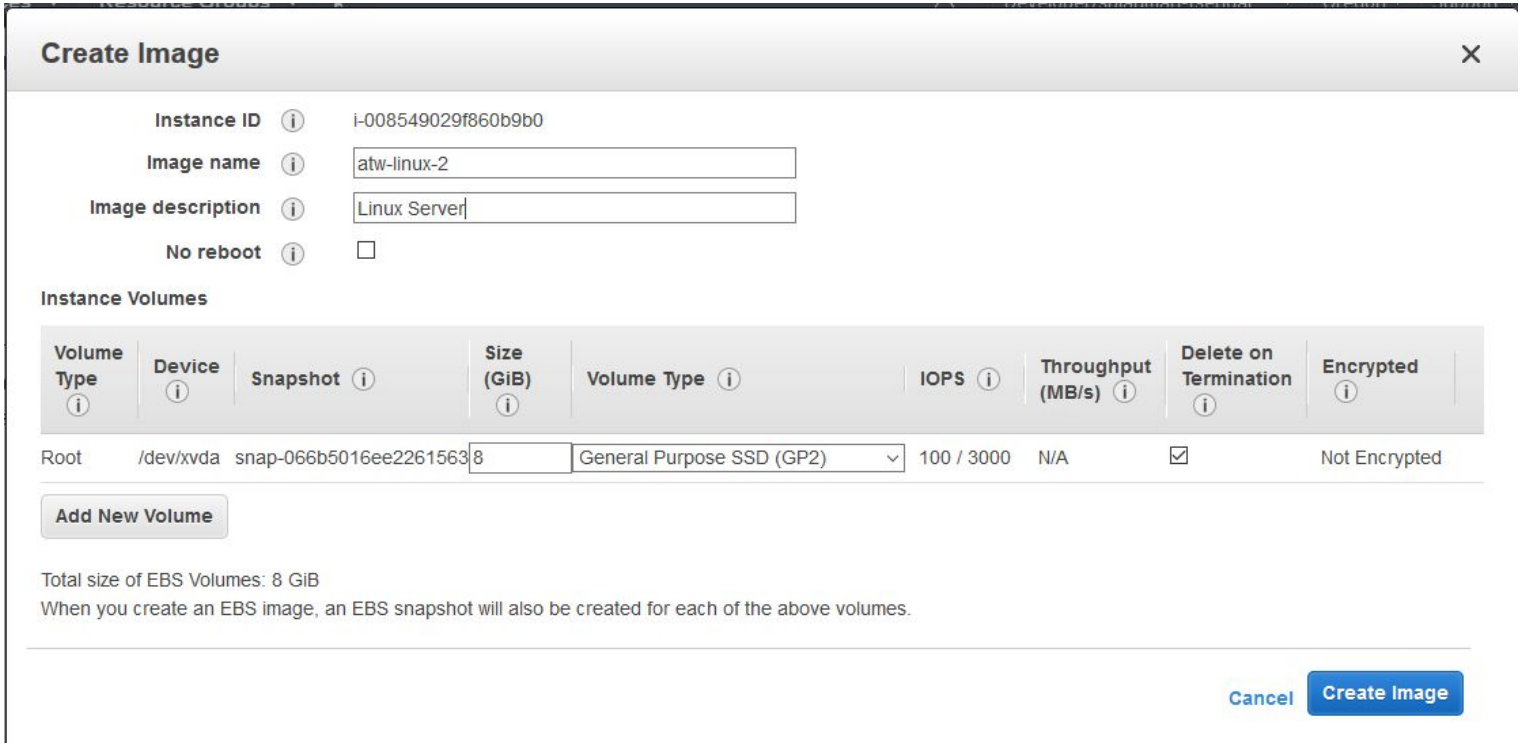
Practical: Create an AMI

Right-click the instance you want to use as the basis for your AMI, and choose **Create Image** from the context menu.



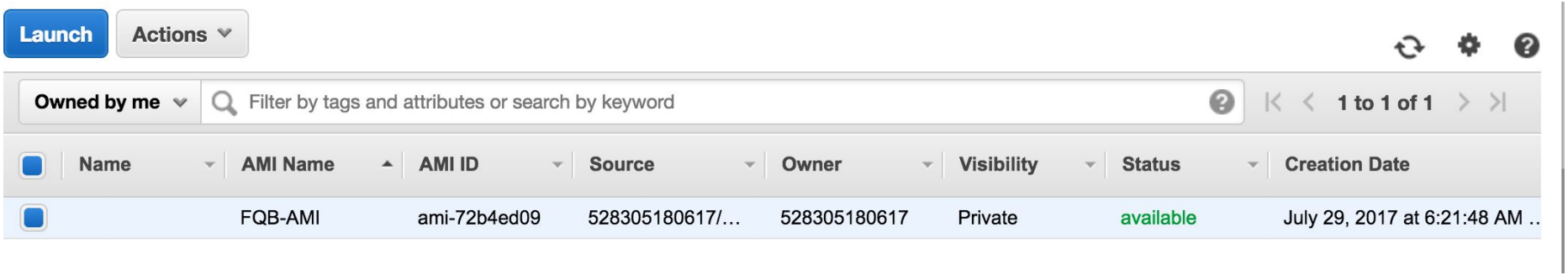
Click on **Create Image** context menu

In the **Create Image** dialog box, type a unique name and description, and then choose **Create Image**.



Click on **Create Image** dialog box. May take a few minutes for the AMI to be created.

After it is created, it will appear in the **AMIs** view in AWS Explorer. To display this view, double-click the **Amazon EC2** | **AMIs** node in AWS Explorer. To see your AMIs, from the **Viewing** drop-down list, choose **Owned By Me**. You may need to choose **Refresh** to see your AMI. When the AMI first appears, it may be in a pending state, but after a few moments, it transitions to an available state.



Securely Managing an Instance

Addressing an Instance

- **Public Domain Name System (DNS) Name**: AWS creates a DNS name that can be used to access the instance. This DNS name is generated automatically. **persists** only while the instance is running, **cannot be transferred**
- **Public IP**: May have a public IP address assigned. This IP address is assigned from the addresses reserved **by AWS** and cannot be specified. **Unique** on Internet, **persists while the instance is running**, **cannot be transferred**
- **Elastic IP**: **Unique** on the Internet that you reserve independently and associate with an Amazon EC2 instance. IP address **persists** until the customer releases it and is **not tied to the lifetime** or state of an individual instance, **can be transferred**.

Initial Access

Amazon EC2 uses two keys together, called as **key pair** to secure login information. Key pairs can be created through the AWS Management Console, CLI, or API, or customers can upload their own key pairs. **AWS stores the public key, and the private key is kept by the customer**. The private key is essential to acquiring secure access to an instance for the first time.

Virtual Firewall Protection

AWS allows you to control traffic in and out of your instances through virtual firewalls called security groups. Security groups allow you to control traffic based on port, protocol, and source/destination. Security groups have different capabilities depending on whether they are associated with an Amazon VPC or Amazon EC2-Classic.

Security groups are associated with instances when they are launched. Every instance must have at least one security group but can have more.

Type of Security Group	Capabilities
EC2-Classic Security Groups	Control outgoing instance traffic
VPC Security Groups	Control outgoing and incoming instance traffic

Virtual Firewall Protection

A **security group is default deny**; that is, it does not allow any traffic that is not explicitly allowed by a security group rule. A rule is defined by the three attributes. When an instance is associated with multiple security groups, the rules are aggregated and all traffic allowed by each of the individual groups is allowed

- **Port**
- **Protocol**
- **Source/Destination**

Type	Protocol	Port Range	Source
HTTP	TCP	80	0.0.0.0/0
HTTP	TCP	80	::/0
SSH	TCP	22	0.0.0.0/0

Source/Destination: Identifies the other end of the communication, source or destination. Defined in 2 ways

- CIDR block (x.x.x.x/x) - Ip Address range
- Security group - Includes any instance that is associated with the given security group. Prevent coupling security group rules with specific IP addresses

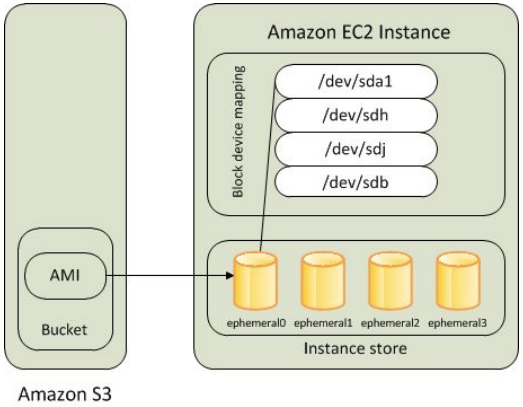
Tip: Security Group on Instance

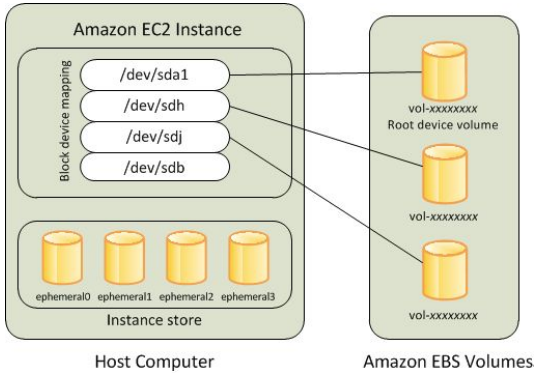
Security groups are applied at the instance level, as opposed to a traditional on-premises firewall that protects at the perimeter. The effect of this is that instead of having to breach a single perimeter to access all the instances in your security group, an attacker would have to breach the security group repeatedly for each individual instance !!!!

Amazon EC2 Root Device Volume

EC2 Root Device Volume types

When you launch an instance, **the *root device volume* contains the image used to boot the instance**. When Amazon introduced Amazon EC2, all AMIs were backed by Amazon EC2 instance store, which means the root device for an instance launched from the AMI is an **instance store volume created from a template stored in Amazon S3**. After Amazon introduced Amazon EBS, Amazon introduced AMIs that are backed by Amazon EBS. This means that the **root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot**.
You can choose between AMIs backed by Amazon EC2 instance store and AMIs backed by Amazon EBS. It is recommended that you use AMIs backed by Amazon EBS, because they launch faster and use persistent storage





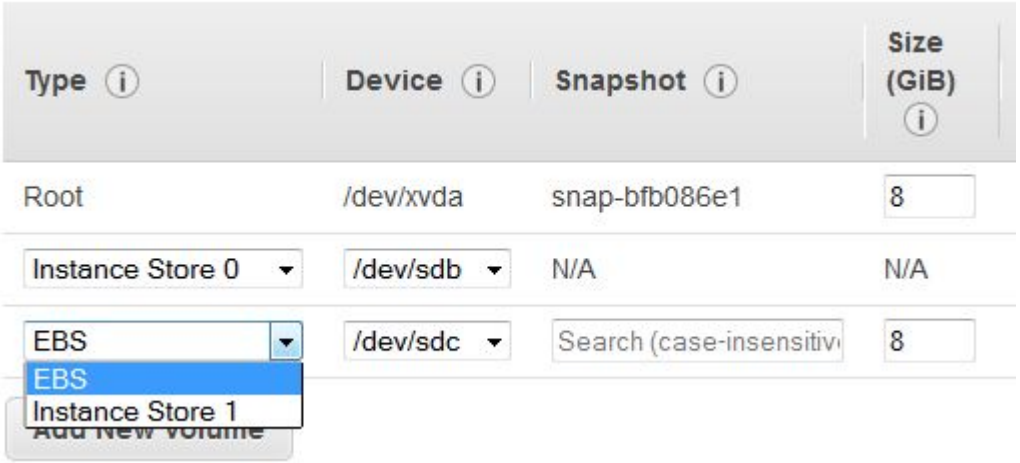
Instance Store (S3) vs EBS AMIS

Two types of EC2 AMIs and Instances

- Instance Store root volume (S3 backed)
  - Original EC2 root volume, boot from ephemeral storage
  - Can start and terminate only
  - All Data is ephemeral unless separate EBS volumes attached
- EBS root volume (EBS backed)
  - Newer, boot from EBS volume
  - Can start, stop, terminate, create image and terminate
  - Ephemeral storage still available but not exposed by default
- Additional EBS volumes can be attached to both

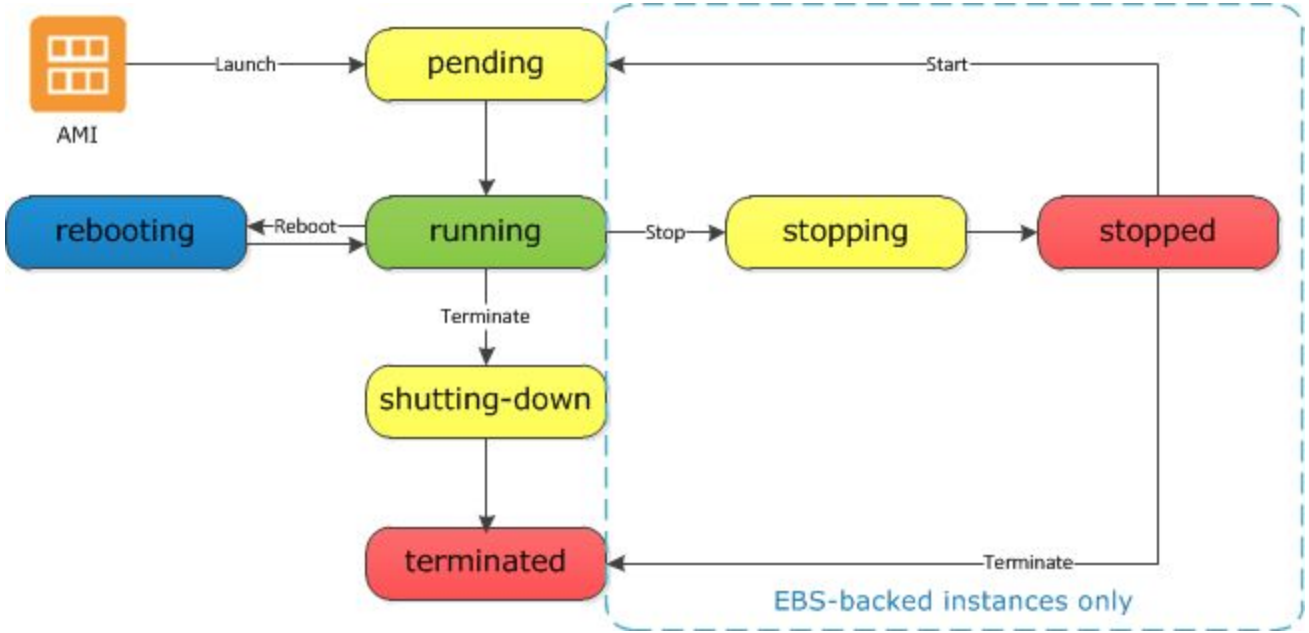
Instance Stores

An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.



EC2 Instance Lifecycle

EC2 Instance Lifecycle



Start - Stop - Terminate Effect on EC2 characteristics

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer	The instance runs on a new host computer	None
Private and public IPv4 addresses	These addresses stay the same	EC2-Classic: The instance gets new private and public IPv4 addresses EC2-VPC: The instance keeps its private IPv4 address. The instance gets a new public IPv4 address, unless it has an Elastic IP address (EIP), which doesn't change during a stop/start.	None
Elastic IP addresses (IPv4)	The Elastic IP remains associated with the instance	EC2-Classic: The Elastic IP is disassociated from the instance EC2-VPC: The Elastic IP remains associated with the instance	The Elastic IP is disassociated from the instance
IPv6 address (EC2-VPC only)	The address stays the same	The instance keeps its IPv6 address	None
Instance store volumes	The data is preserved	The data is erased	The data is erased
Root device volume	The volume is preserved	The volume is preserved	The volume is deleted by default
Billing	The instance billing hour doesn't change.	You stop incurring charges for an instance as soon as its state changes to stopping. Each time an instance transitions from stopped to running, we start a new instance billing hour.	You stop incurring charges for an instance as soon as its state changes to shutting-down.

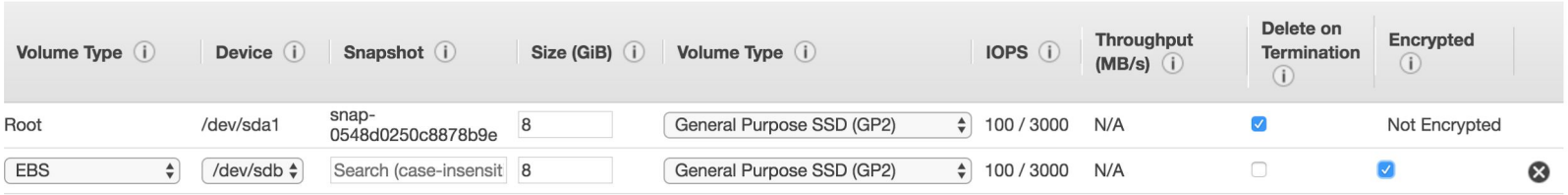
Amazon EBS

What is EBS?

Amazon Elastic Block Store (Amazon EBS) **provides block level storage volumes for use with EC2 instances**. EBS volumes are **highly available and reliable storage volumes** that can be attached to any running instance that is in the **same Availability Zone**.

Facilitates

- **Quickly accessible**
- **Long-term persistence**
- **Data encryption**



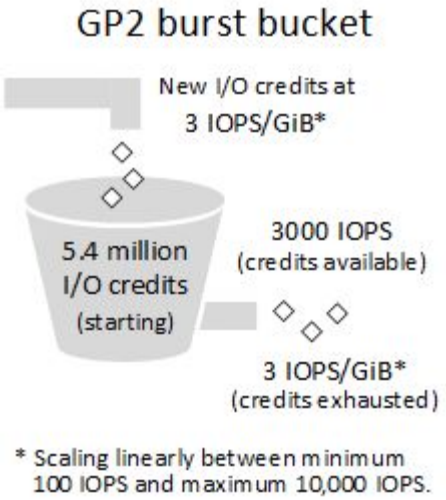
Use Cases

- Primary storage for file systems, databases, application servers

EBS Types

- EBS General Purpose SSD (gp2), Provisioned IOPS SSD (io1), Throughput Optimized HDD (st1), and Cold HDD (sc1) volumes up to 16 TiB in size
- General Purpose SSD (gp2) volumes, you can expect base performance of 3 IOPS/GiB, with the ability to burst to 3,000 IOPS for extended periods of time. (Small DB / Boot / Dev-Test Env)
- Provisioned IOPS SSD (io1) volumes, you can provision a specific level of I/O performance. Io1 volumes support up to 20,000 IOPS and 320 MB/s of throughput . (MongoDB, MySQL)
- Throughput Optimized HDD (st1) volumes provide low-cost magnetic storage.
- Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. Best for data that is infrequently accessed

Understanding Burst vs. Baseline Performance (GP2)



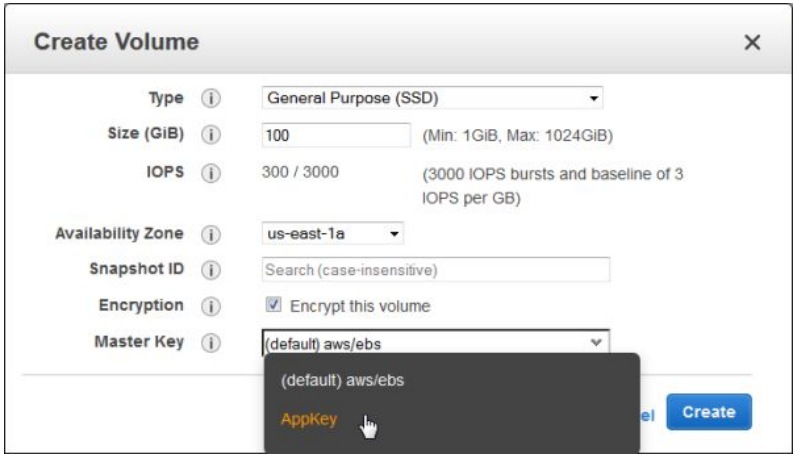
GP2 volumes start with 5.4M I/O credit that, if fully used, works out to 3000 IOPS for 30 minutes. The burst credit is always being replenished at the rate of 3 IOPS per GiB per second

Encryption Options

Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build, maintain, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume

A new master key will be created unless you select a master key that you created separately in the service. Your data and associated keys are encrypted using the industry-standard AES-256 algorithm



Benefits of EBS

- Data availability
  - AZ replication
- Data persistence
  - Pay for the volume usage
  - Attach to single instance
  - Safe on instance termination
- Data encryption
  - encrypted EBS volumes with the Amazon EBS encryption
- Snapshots
  - Backup to S3
- Flexibility
  - Modify volume type, volume size, and IOPS capacity without service interruption

	Solid-State Drives (SSD)		Hard disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of transactional workloads	Highest-performance SSD volume designed for mission-critical applications	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	Recommended for most workloads System boot volumes Virtual desktops Low-latency interactive apps Development and test environments	Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume Large database workloads, such as: MongoDB Cassandra Microsoft SQL Server MySQL PostgreSQL Oracle	Streaming workloads requiring consistent, fast throughput at a low price Big data Data warehouses Log processing Cannot be a boot volume	Throughput-oriented storage for large volumes of data that is infrequently accessed Scenarios where the lowest storage cost is important Cannot be a boot volume
API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS/Volume**	10,000	20,000	500	250
Max. Throughput/Volume†	160 MiB/s	320 MiB/s	500 MiB/s	250 MiB/s
Max. IOPS/Instance	75,000	75,000	75,000	75,000
Max. Throughput/Instance	1,750 MB/s	1,750 MB/s	1,750 MB/s	1,750 MB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

Practical : Creating an Amazon EBS Volume

Creating an Amazon EBS Volume

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the region in which you would like to create your volume.
3. In the navigation pane, under ELASTIC BLOCK STORE, choose Volumes.
4. Above the upper pane, choose Create Volume.
5. In the Create Volume dialog box, for Volume Type, choose General Purpose SSD (GP2), Provisioned IOPS SSD (IO1), Throughput Optimized HDD (ST1), Cold HDD (SC1), or Magnetic.
6. For Size, enter the size of the volume, in GiB.
7. For Availability Zone, select the Availability Zone in which to create the volume.
8. (Optional) To create an encrypted volume, select the Encrypted box and choose the master key you want to use when encrypting the volume. You can choose the default master key for your account,
9. Choose Yes, Create.

Practical: Restoring an Amazon EBS Volume from a Snapshot

Restoring an Amazon EBS Volume from a Snapshot

1. From the navigation bar, select the region that your snapshot is located in
2. In the navigation pane, choose Volumes, Create Volume.
3. In the Create Volume dialog box, for Volume Type, choose General Purpose SSD, Provisioned IOPS SSD, or Magnetic.
4. For Snapshot, start typing the ID or description of the snapshot from which you are restoring the volume, and select it from the list of suggested options.
5. For Size, enter the size of the volume in GiB,
6. In the Availability Zone list, select the Availability Zone in which to create the volume. EBS volumes can only be attached to EC2 instances within the same Availability Zone.
7. Choose Yes, Create

Practical : Attaching an Amazon EBS Volume to an Instance

Attaching an Amazon EBS Volume to an Instance

- Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
- In the navigation pane, choose Volumes.
- Select a volume and choose Actions, Attach Volume.
- In the Attach Volume dialog box, start typing the name or ID of the instance to attach the volume to for Instance, and select (only instances in the same AZ as the volume are displayed).
- You can keep the suggested device name, or enter a different supported device name.
- Choose Attach.

Practical: Making an Amazon EBS Volume Available for Use

Making an Amazon EBS Volume Available for Use

- Connect to your instance using SSH
- Use the lsblk command to view your available disk devices and their mount points (if applicable) to help you determine the correct device name to use. The output of lsblk removes the /dev/ prefix from full device paths

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0  100G  0 disk
xvda1 202:1    0    8G  0 disk /
```

- Determine if file system exists.

```
[ec2-user ~]$ sudo file -s /dev/xvdf
/dev/xvdf: data
```

Making an Amazon EBS Volume Available for Use

```
sudo file -s /dev/xvda1
 /dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=1701d228-e1bd-4094-a14c-8c64d6819362 (needs
 journal recovery) (extents) (large files) (huge files)
```

- (Conditional) Use the following command to create an ext4 file system on the volume.

```
[ec2-user ~]$ sudo mkfs -t ext4 device_name
```

- Use the following command to create a mount point directory for the volume. The mount point is where the volume is located in the file system tree and where you read and write files to after you mount the volume. Substitute a location for mount_point, such as /data.

```
[ec2-user ~]$ sudo mkdir mount_point
```

Making an Amazon EBS Volume Available for Use

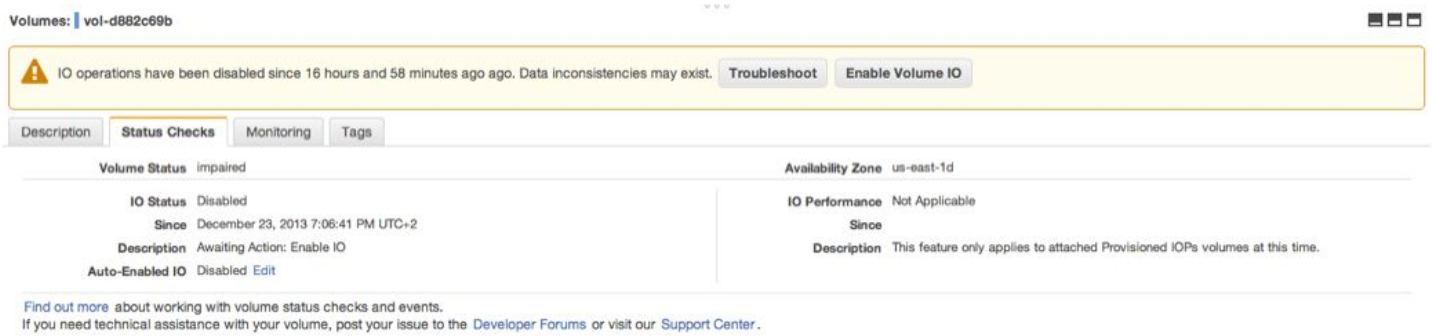
- Use the following command to mount the volume at the location you just created.

```
[ec2-user ~]$ sudo mount device_name mount_point
```

- Check df -h

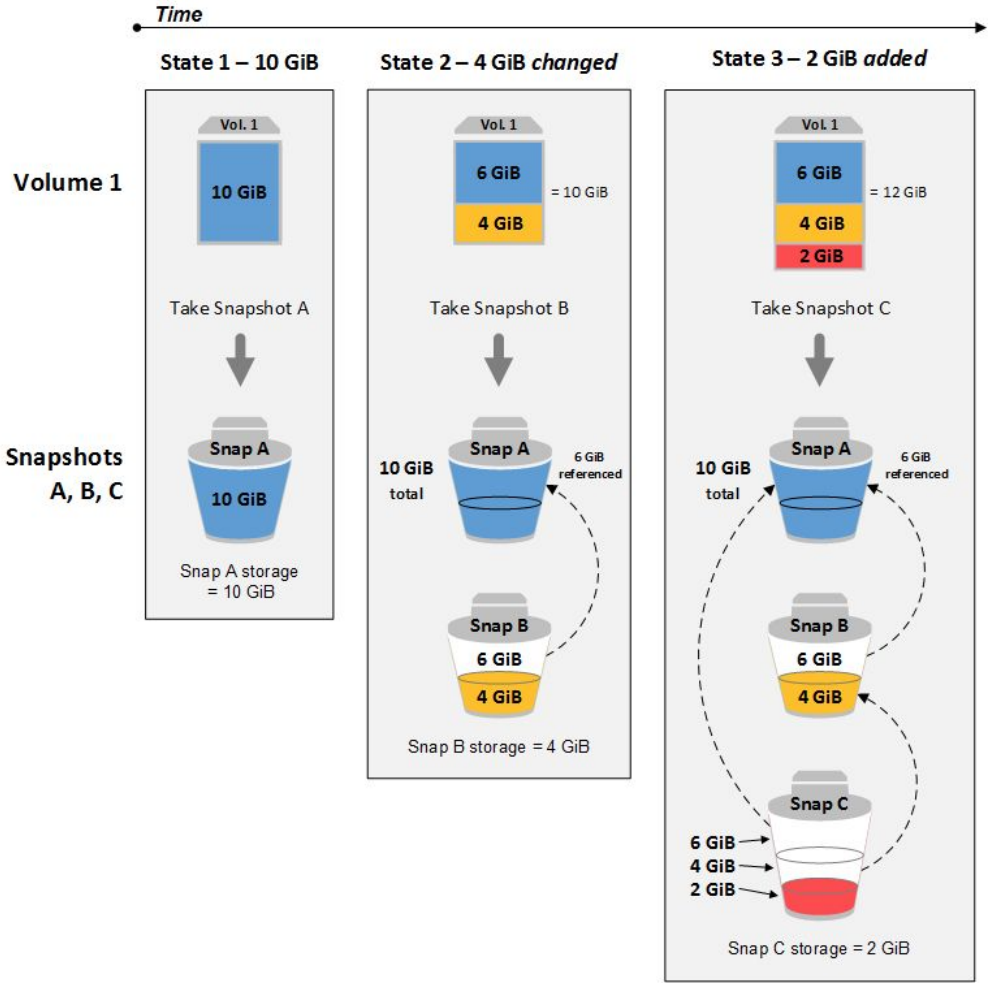
Monitoring the Status of Your Volumes

Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
ok	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
warning	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations) Severely Degraded (Volume performance is well below expectations)
impaired	Enabled (I/O Enabled or I/O Auto-Enabled) Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted) Not Available (Unable to determine I/O performance because I/O is disabled)
insufficient-data	Enabled (I/O Enabled or I/O Auto-Enabled) Insufficient Data	Insufficient Data



Amazon EBS Snapshots

- You can backup the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots.
- Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved.
- Minimizes the time required to create the snapshot and saves on storage costs.



Creating an Amazon EBS Snapshot

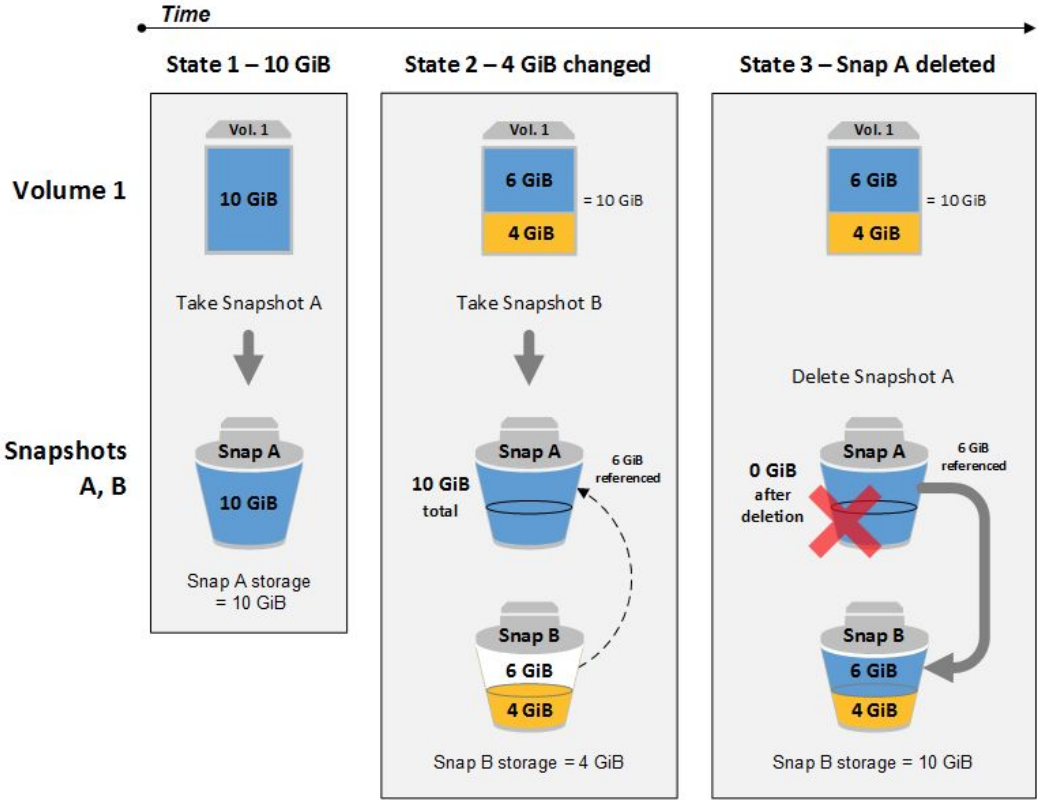
- Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
- Choose Snapshots in the navigation pane.
- Choose Create Snapshot.
- In the Create Snapshot dialog box, select the volume to create a snapshot for, and then choose Create.

Amazon EBS–Optimized Instances

An Amazon EBS–optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

EBS–optimized instances deliver dedicated bandwidth to Amazon EBS, with options between 500 Mbps and 12,000 Mbps, depending on the instance type you use. When attached to an EBS–optimized instance, General Purpose SSD (gp2) volumes are designed to deliver within 10% of their baseline and burst performance 99% of the time in a given year, and Provisioned IOPS SSD (io1) volumes are designed to deliver within 10% of their provisioned performance 99.9% of the time in a given year. Both Throughput Optimized HDD (st1) and Cold HDD (sc1) guarantee performance consistency of 90% of burst throughput 99% of the time in a given year.

Deleting an Amazon EBS Snapshot



Exam Essentials

- Know the properties of the Amazon EC2 pricing options.
- Know what determines network performance
  - Every instance type is rated for low, moderate, high, or 10 Gbps network performance
- Know what instance metadata is and how it's obtained
  - instance ID, instance type, and security groups
- Know how security groups protect instances
  - Security Group deny by default / Inbound / Outbound / Port / Protocol
- Know how to interpret the effect of security groups
  - Union of all the rules in multiple groups
- Know the different Amazon ebs volume types, their characteristics, and their appropriate workloads
- Know how to encrypt an Amazon ebs volume.
- Understand the concept and process of snapshots
- Know how Amazon ebs-optimized instances affect Amazon ebs performance

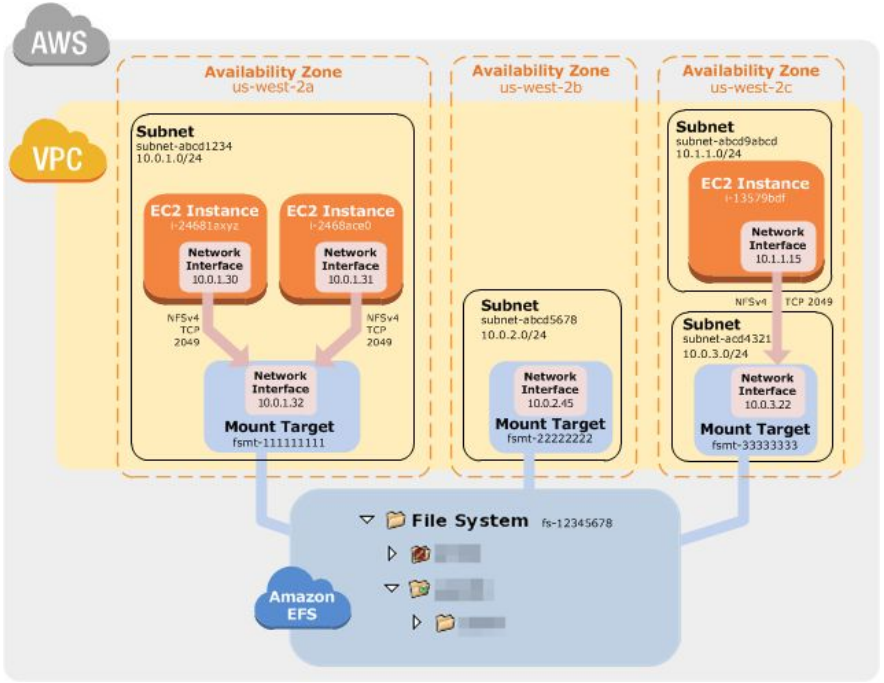
Amazon Elastic File System (Amazon EFS)

Amazon Elastic File System (Amazon EFS)

Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances

Important

Amazon EFS is not supported on Windows instances.



Exam Essentials

- Know the basics of launching an Amazon ec2 instance.
  - AMI + Hardware (CPU, RAM etc)
- Know what architectures are suited for what Amazon ec2 pricing options
  - Spot vs Reserved vs Dedicated
- Know how to combine multiple pricing options that result in cost optimization and scalability.
  - Combine Spot / Reserved / Dedicated to suit your needs
- Know the benefits of enhanced networking.
  - Network performance for different instance types
- Know the capabilities of vm import/export
  - Migrate Your Existing Applications and Workloads to Amazon EC2
- Know the methods for accessing an instance over the internet
  - Public IP address, elastic IP address, or public DNS name, VPC
- Know the lifetime of an instance store
  - Data on an instance store is lost when the instance is stopped or terminated. Instance store data survives an OS reboot.

Test

Test