



Amazon Web Services

Storage Services

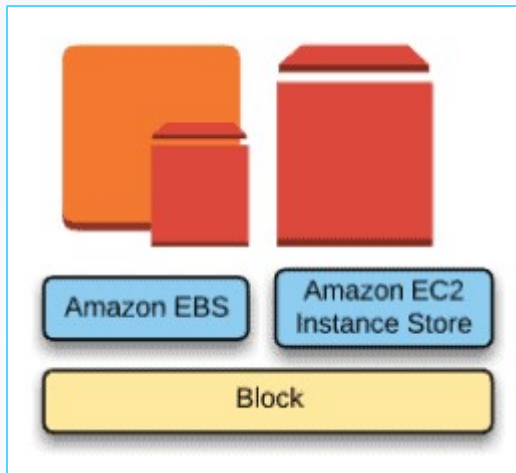
EBS, EFS & S3

Contents

- ✓ Storage Services – EBS, EFS & S3
- ✓ Elastic Block Storage – Volumes, Snapshots, AMIs
- ✓ Elastic File System
- ✓ Simple Storage Service (S3)



EBS, EFS & S3



Elastic Block Store



Elastic File System



Simple Storage Service



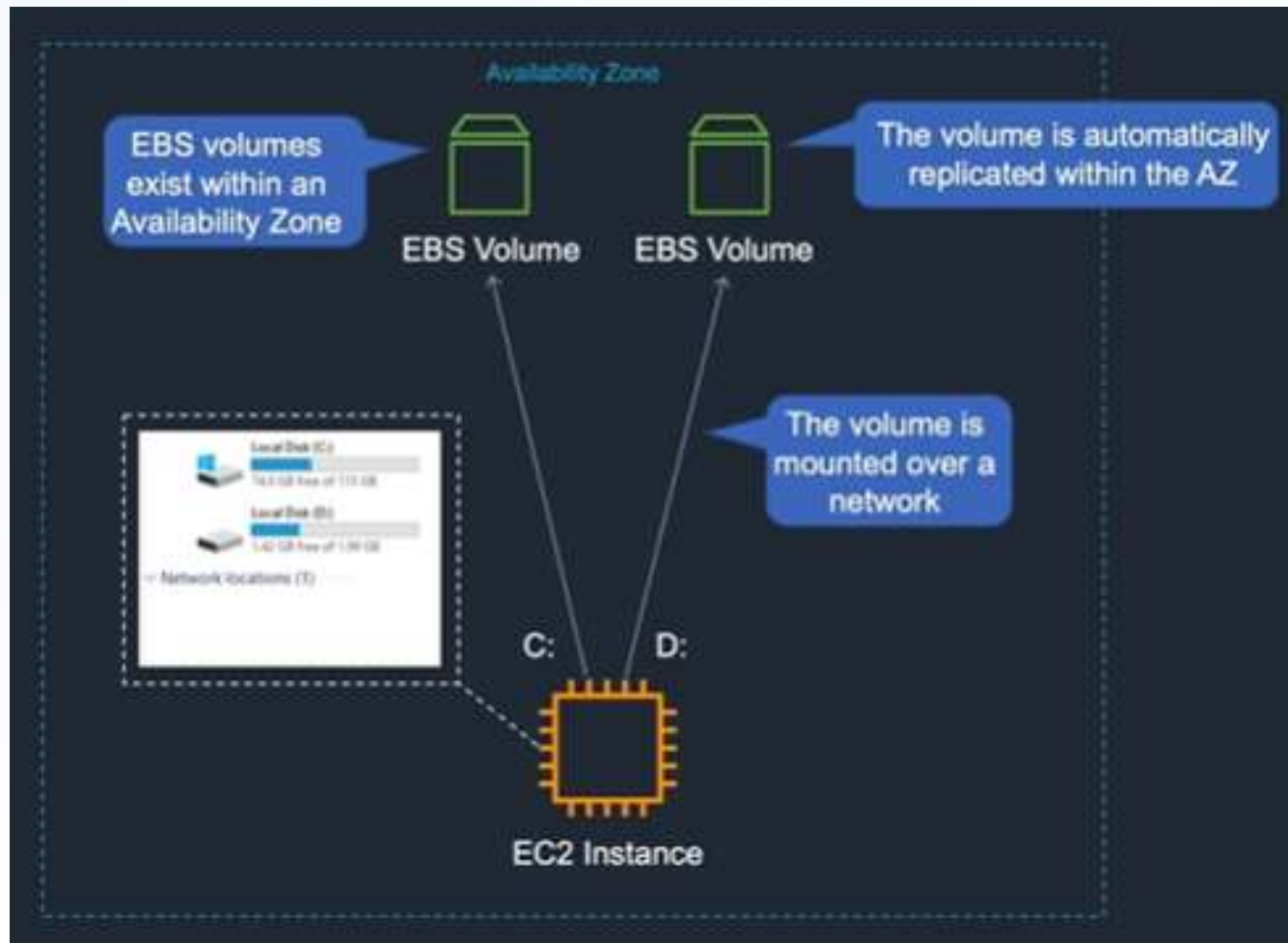


Amazon Elastic Block Store (EBS)

EBS Volumes, Snapshots & AMLs



Elastic Block Store (EBS)



Elastic Block Store (EBS)

Volume Type	Solid State Drives (SSD)		Hard Disk Drives (HDD)	
	EBS Provisioned IOPS SSD (io1)	EBS General Purpose SSD (gp2)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Short Description	Highest performance	General Purpose SSD volume	Low cost HDD volume	Lowest cost HDD
Use Cases	I/O-Intensive NoSQL and relational databases	Boot volumes, low-latency interactive apps, dev & test	Big data, data warehouses, log processing	Colder data requiring fewer scans per day
Volume Size	4GB – 16TB	1 GB – 16 TB	500 GB – 16 TB	500 GB – 16 TB
Max IOPS/Volume	64,000	16,000	500	250
Max Throughput/Volume	1,000 MB/s	250 MB/s	500 MB/s	250 MB/s

➤ IOPS – input/output operations per second



Elastic Block Store (EBS)

- EBS is durable block-level storage to be used with EC2 instances in the AWS cloud. They are accessed via NFS (network file systems).
 - Volumes get automatically replicated within Availability Zones for high availability and durability.
- EBS volumes are mounted onto EC2 instance similar to a physical hard drive and then format the EBS volume to the desired file system.
- EBS allows for dynamically increasing capacity, performance tuning and you can even change the type of volume with any downtime or performance impact.



Creating an EBS Volume

- Connect to **EC2 Service** and select **Elastic Block Store -> Volumes**.
- Click **Create Volume** button.
 - Select Volume Type (ex: Provisioned IOPS SSD (io1))
 - Select other details as appropriate.
- Click on **Create Volume**.

Volume type [Info](#)

Provisioned IOPS SSD (io1) ▼

Size (GiB) [Info](#)

100

Min: 4 GiB, Max: 16384 GiB. The value must be an integer.

IOPS [Info](#)

3000

Min: 100 IOPS, Max: 5000 IOPS (up to 50 IOPS per GiB)


Throughput (MiB/s) [Info](#)

Not applicable

Availability Zone [Info](#)

us-east-1a ▼

Snapshot ID - optional [Info](#)

Don't create volume from a snapshot ▼ 



Create a Windows EC2 instance

- Select EC2 Service and click on **Launch Instances**
- Select a Windows EC2 instance (Ex: Microsoft Windows Server 2019 Base)
- Make sure your security group has RDP inbound rules added.
- Review and launch the instance.

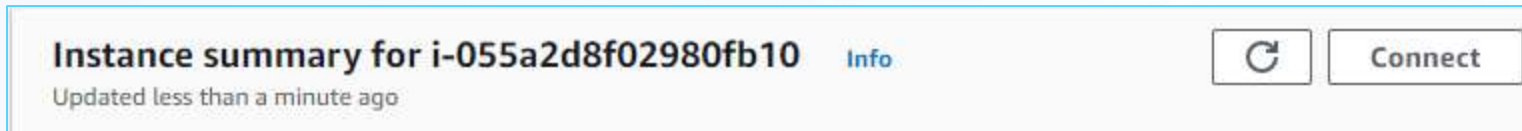
Number of instances	1	Launch into Auto Scaling
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-02e3d6afcaccb1efc (default)	
Subnet	subnet-07adcb7ea98547696 Default in us-east-1a 4091 IP Addresses available	
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	
Domain join directory	No directory	

sg-09b7afc3b0bd81e06		Web Access		Web Access
HTTP	TCP	80	...	::/0
RDP	TCP	3389		0.0.0.0/0
RDP	TCP	3389		::/0

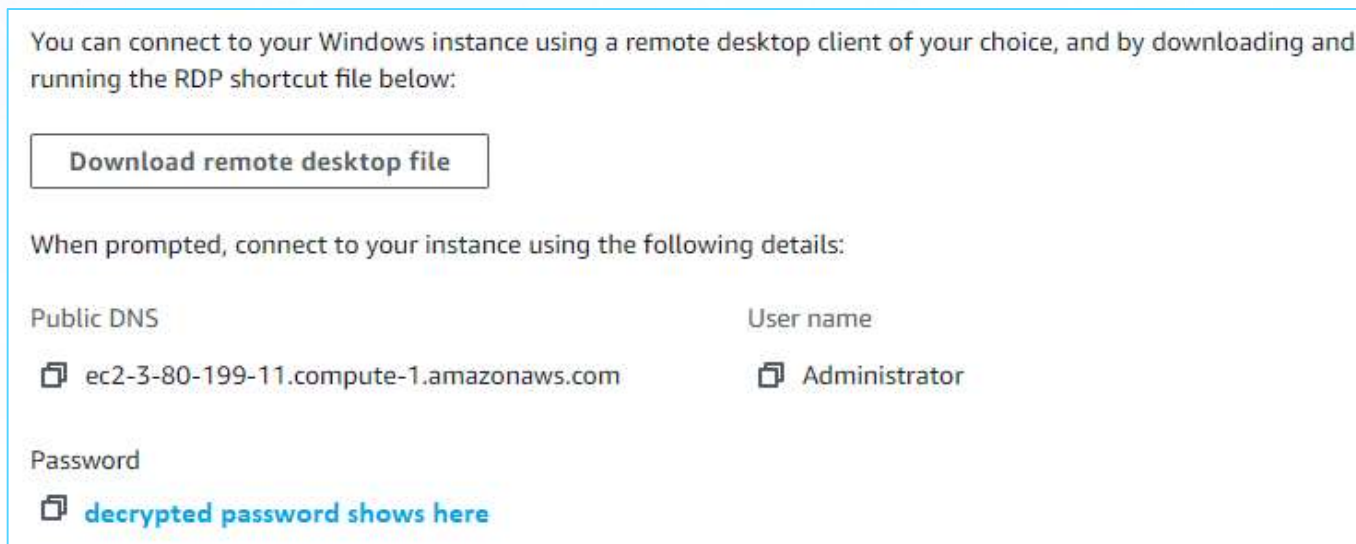


Connect to the Windows EC2 instance via RDP

- Open the windows EC2 instance and click on **Connect** button.

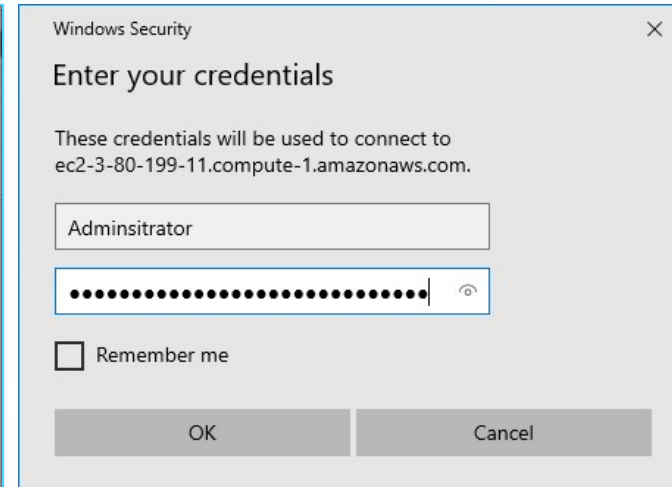
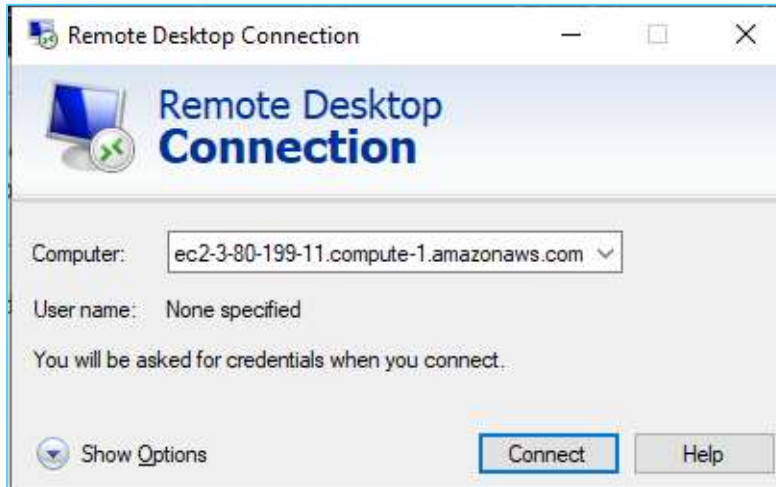
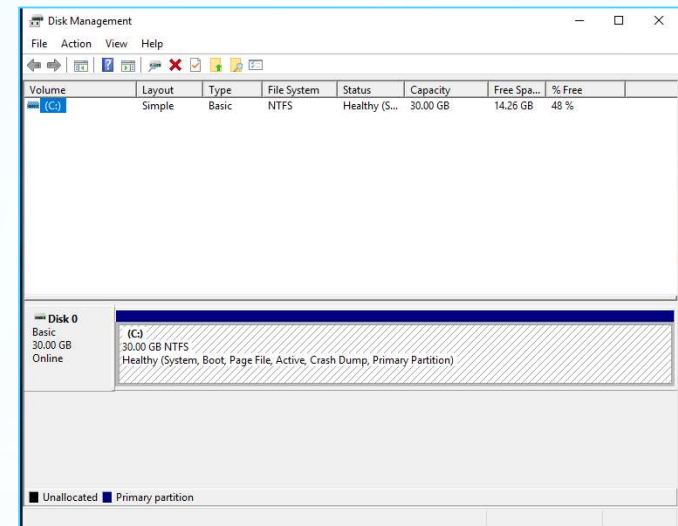


- Select **RDP Client** option and click on **Get Password**
- Browse and select the **private key (.pem file)** of the key-pair that you used.
- Click on **Decrypt password** button.
- Note down the **Public DNS**, **User name** and **password**.



Connect to the Windows EC2 instance via RDP

- Open RDP Client from windows and add your **Public DNS** as the computer name. Enter your credentials – **User name** and **Password**. This connects the EC2 instance.
- In the Windows server type 'disk management'; in the search and launch 'Disk Management' console.



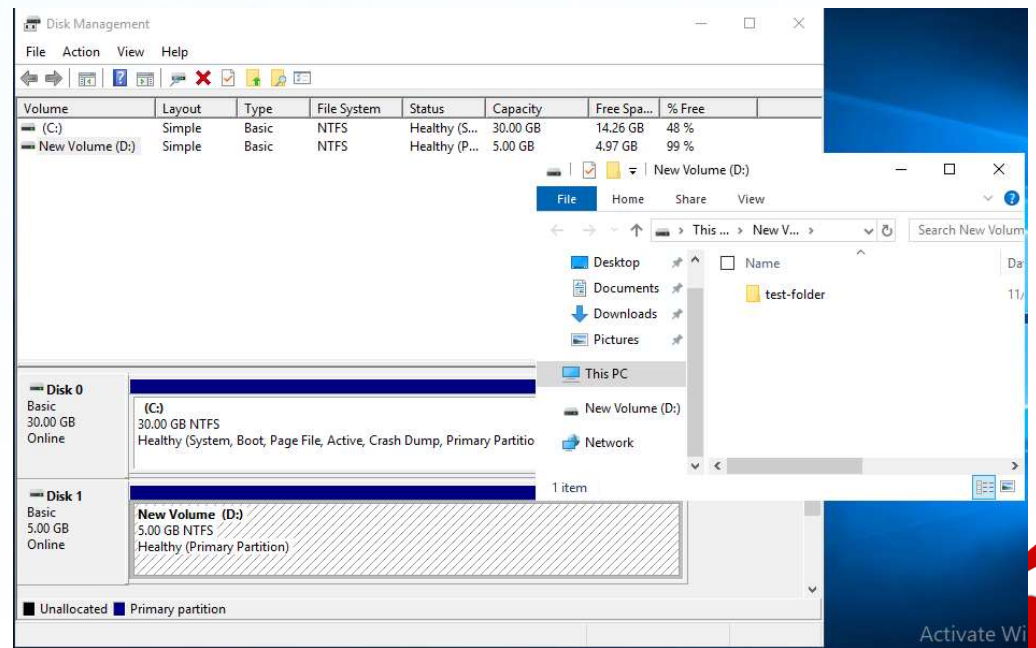
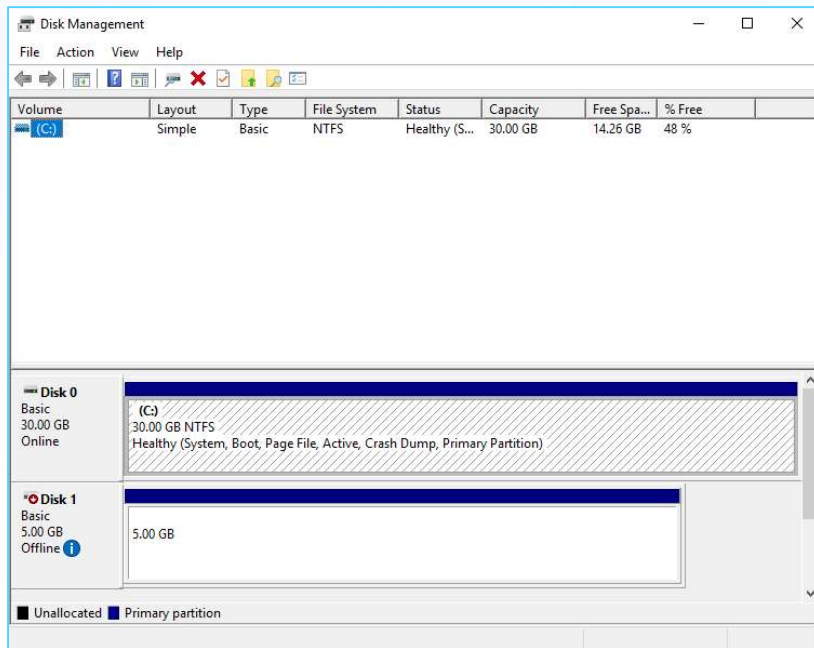
Attach the EBS volume to the EC2 instance

- Connect to **EC2 Service** and select **Elastic Block Store** -> **Volumes**.
- Right click on the volume and select **Attach volume** option.

The screenshot displays the AWS Management Console interface for the Elastic Block Store (EBS) Volumes page. On the left, the 'Volumes (1/1)' section shows a table with one volume selected. A context menu is open over this volume, listing actions: 'Create volume', 'Modify volume', 'Create snapshot', 'Create snapshot lifecycle policy', 'Delete volume', and 'Attach volume'. The 'Attach volume' option is highlighted. On the right, the 'Basic details' panel for the selected volume 'vol-039fc1a8869ac208c' is shown. It includes the 'Volume ID', 'Availability Zone' (us-east-1a), and 'Instance' (i-055a2d8f02980fb10). The 'Device name' is set to 'xvdf'. At the bottom right, there are 'Cancel' and 'Attach volume' buttons. A red dart is shown hitting the bullseye of a target in the bottom right corner of the image.

Attach the EBS volume to the EC2 instance

- Now, open the RDP client again. Now you should find the new volume attached.
- Right click on the new volume. Select **online**.
- Right click one more time. Select **Initialize Disk**.
- Right Click again and **Create a Simple Volume**. You can now add any content to this volume.



Instance Stores

- An instance store provides temporary block-level storage for your EC2 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.

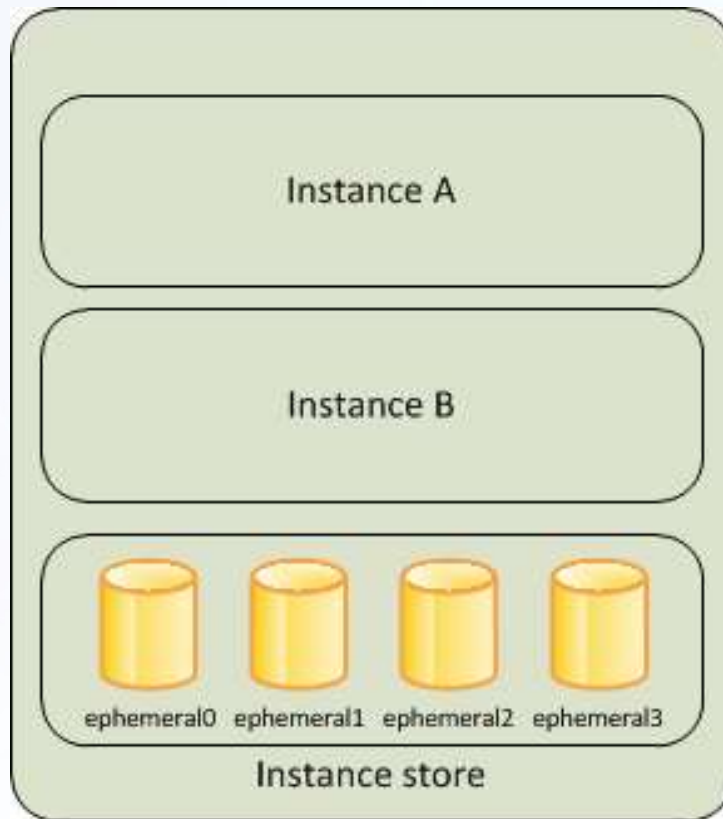


Instance Stores

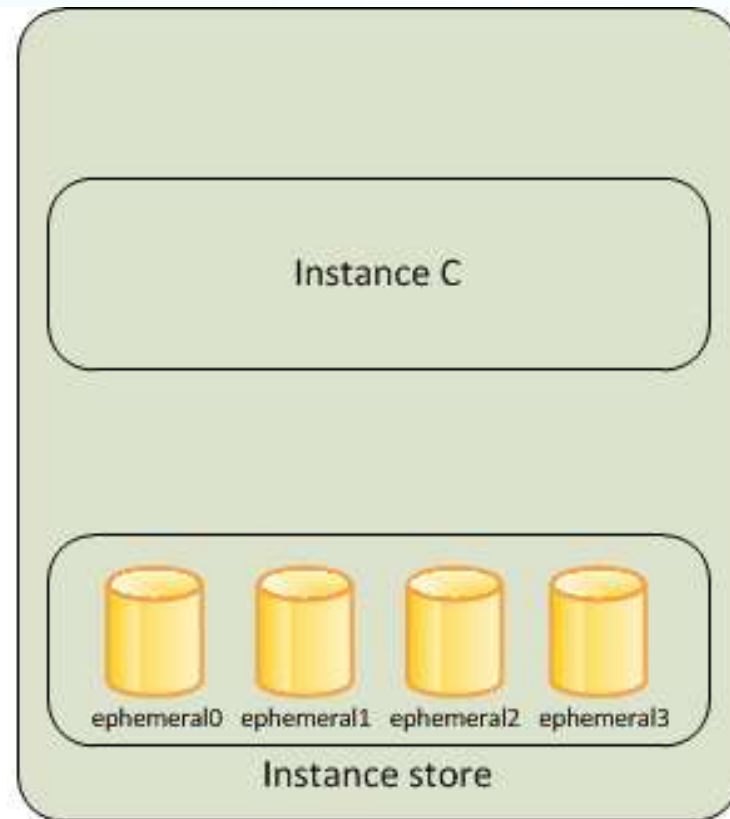
- An instance store consists of one or more instance store volumes exposed as block devices.
 - The size of an instance store as well as the number of devices available varies by instance type.
- The virtual devices for instance store volumes are ephemeral[0-23].
 - Instance types that support one instance store volume have ephemeral0.
 - Instance types that support two instance store volumes have ephemeral0 and ephemeral1, and so on.



Instance Stores



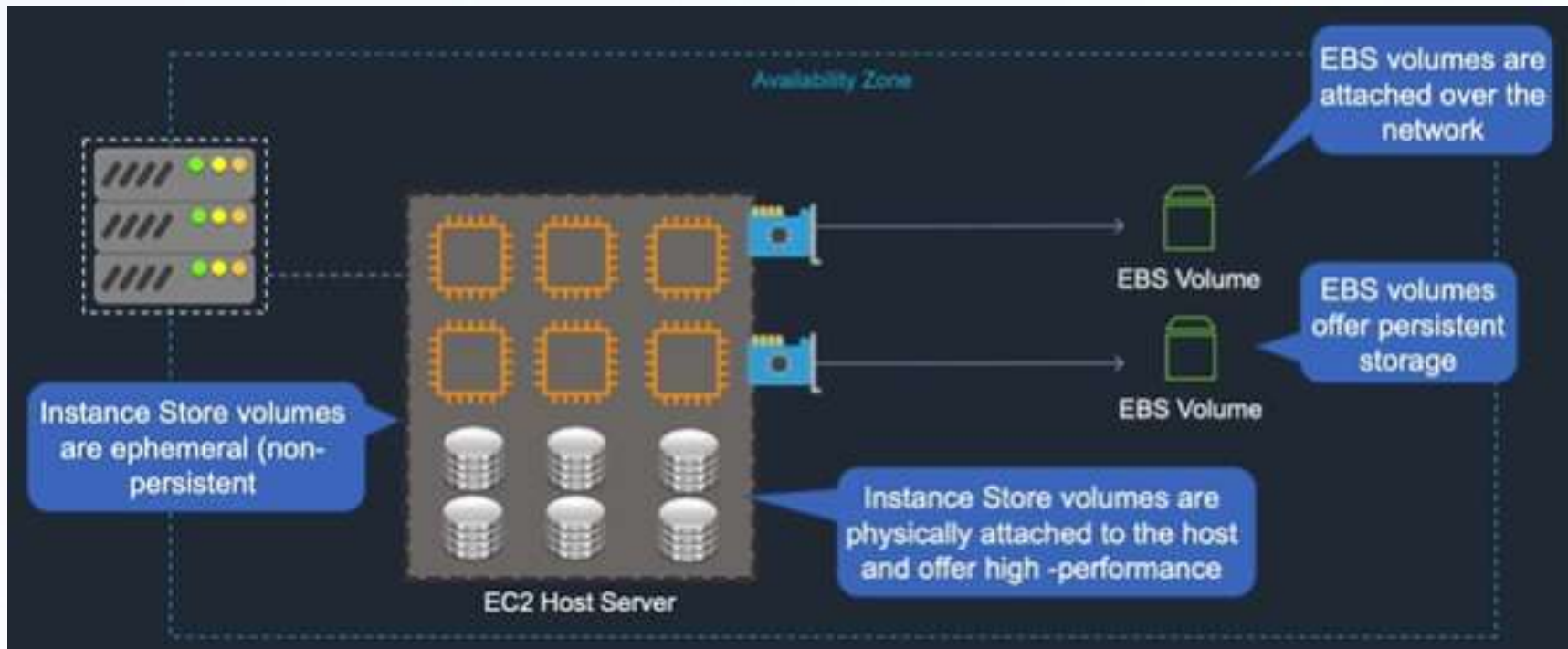
Host Computer 1



Host Computer 2



EBS Volumes & Instance Stores



EBS Volumes & Instance Stores

Step 2: Choose an Instance Type

<input type="checkbox"/>	c5a	c5a.16xlarge	64	128	EBS only	Yes
<input type="checkbox"/>	c5a	c5a.24xlarge	96	192	EBS only	Yes
<input checked="" type="checkbox"/>	c5ad	c5ad.large	2	4	1 x 75 (SSD)	Yes
<input type="checkbox"/>	c5ad	c5ad.xlarge	4	8	1 x 150 (SSD)	Instance Store Volumes Yes

Step 4: Add Storage

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ
Root EBS Volume	/dev/xvda	snap-08e7398ef8a6052df	8	General Purpose SSD (gp2) ▼
ephemeral0 Instance Store	nvme0n1	N/A	75	NVMe SSD
Add New Volume				

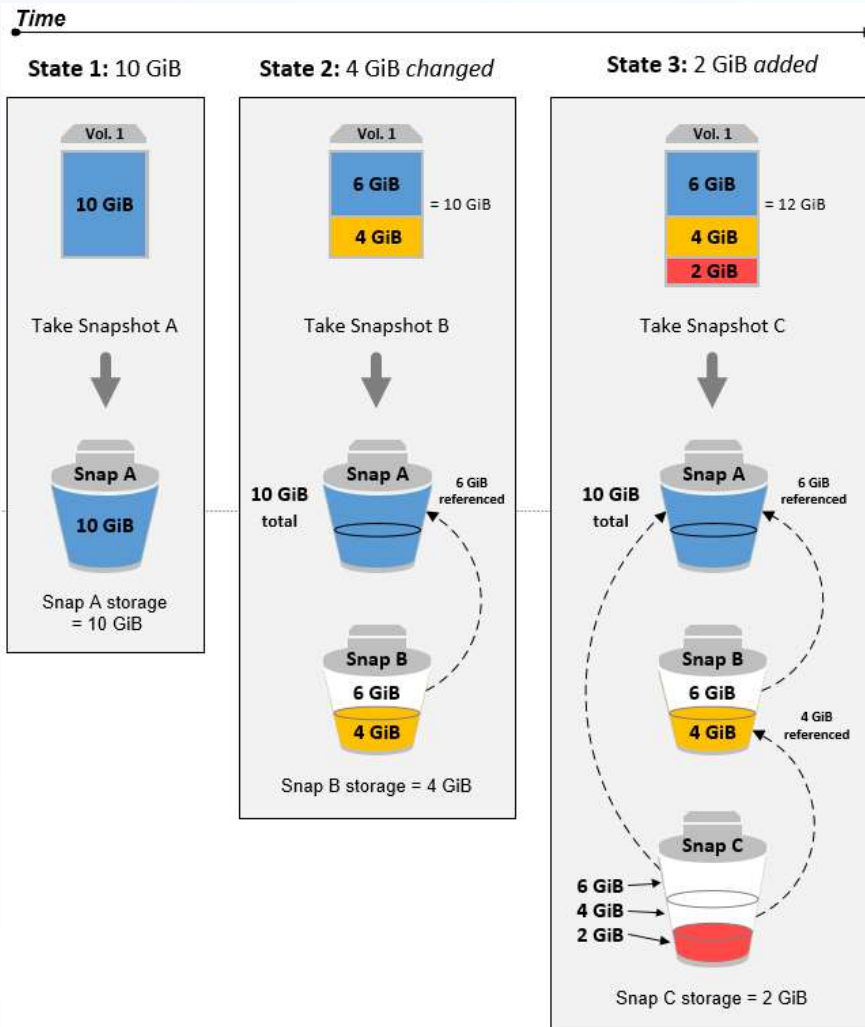


Amazon EBS Snapshots

- You can back up the data on your EBS volumes to 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.
- Each snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.
- When you create an EBS volume based on a snapshot, the new volume begins as an exact replica of the original volume that was used to create the snapshot.
- The replicated volume loads data in the background so that you can begin using it immediately.
 - If you access data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background.



Amazon EBS Snapshots



- A snapshot is stored in S3.
- It is in a Region, not in any specific AZ.
- We can create volumes from this snapshot in any desired AZ and we can then attach those volumes to an instance running in that AZ.



Amazon EBS Snapshots



Create a Snapshot of a EBS Volume

- Connect to **EC2 Service** and select **Elastic Block Store** -> **Snapshots**.
- Click on **Create Snapshot** button.


[Snapshots](#) > Create Snapshot

Create Snapshot

Select resource type ☒ Volume
☐ Instance

Volume*  

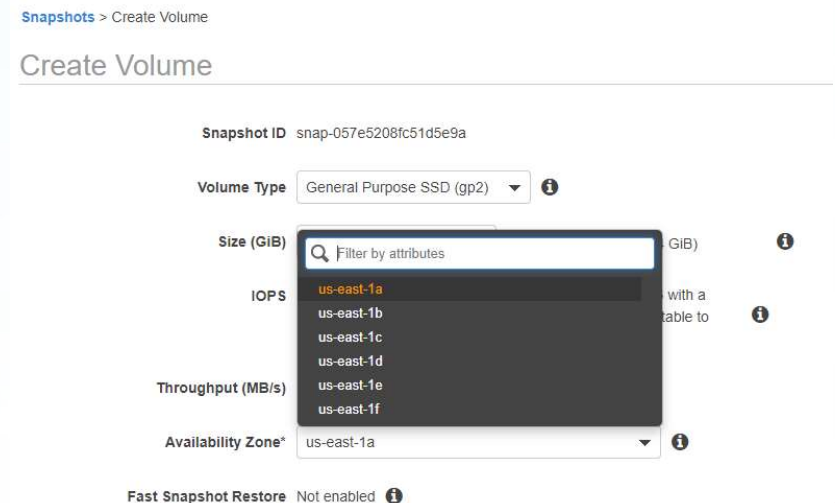
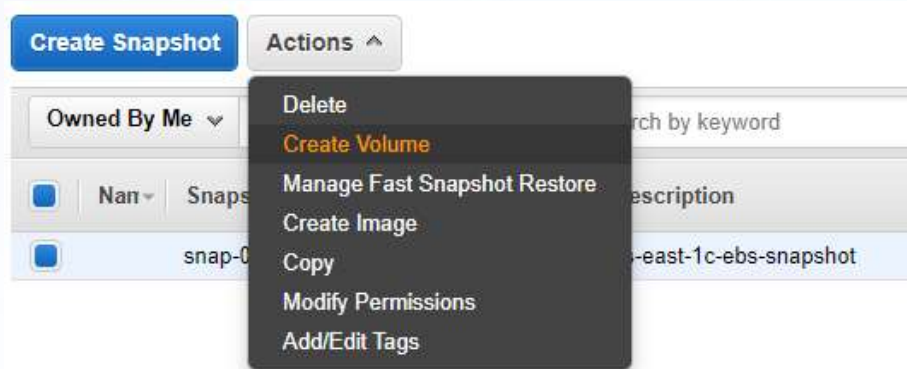
Description 

Encrypted ☐ Not Encrypted 



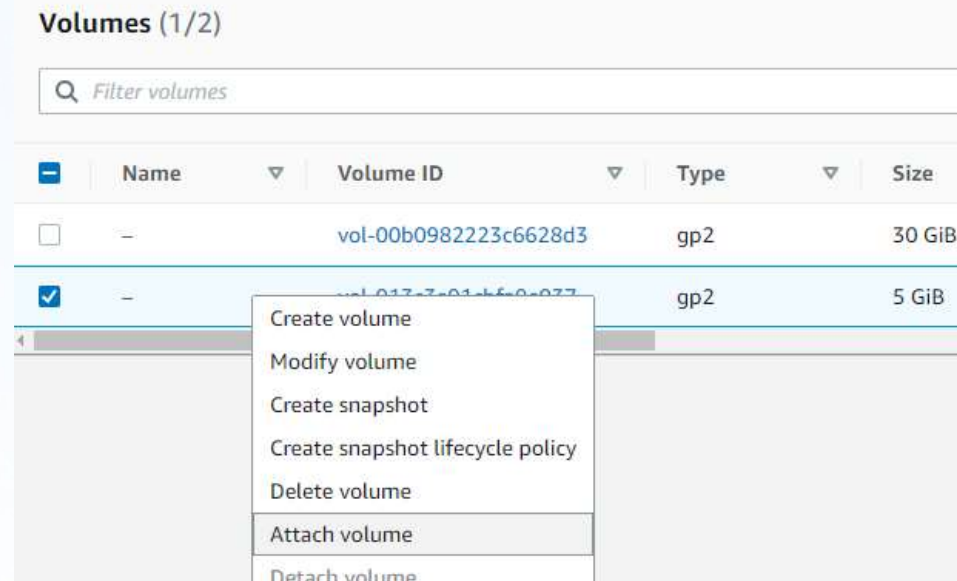
Create a new EBS volume from Snapshot

- Select the Snap shot you created in the previous step
- Click on **Actions >> Create Volume** option.
- Select the Availability zone in which you want to create this volume, You can attach this volume to any EC2 running in that AZ.



Attach the new EBS volume to an EC2 instance

- Connect to **EC2 Service** and select **Elastic Block Store** -> **Volumes**.
- Right click on the volume and select **Attach volume** option.



- Here, we moved an EBS volume from one AZ to another, and attached to an instance running on a different AZ.



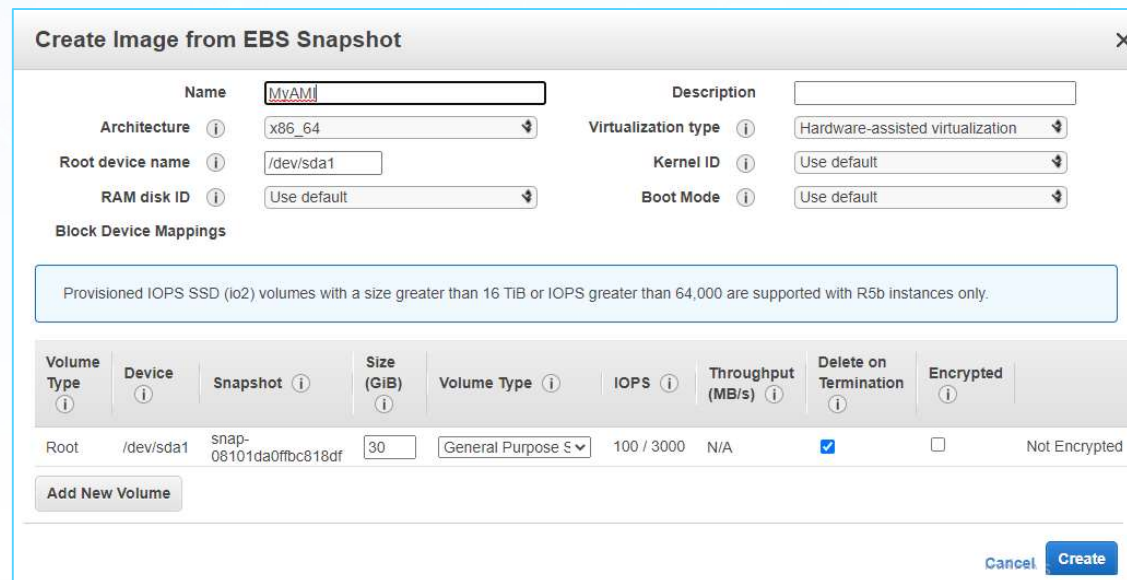
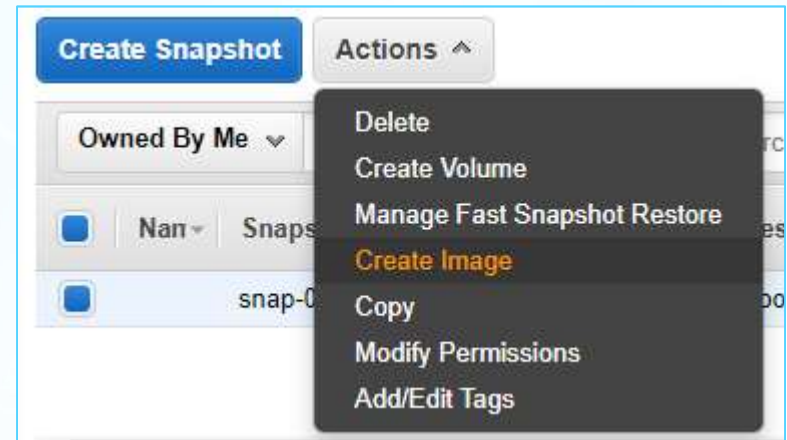
Amazon Machine Image (AMI)

- An AMI provides the information required to launch an instance.
 - You can launch multiple instances from a single AMI when you need multiple instances with the same configuration.
 - You can use different AMIs to launch instances when you need instances with different configurations.
- An AMI includes the following:
 - One or more Amazon Elastic Block Store (Amazon EBS) snapshots, or, for instance-store-backed AMIs, a template for the root volume of 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.



Create an AMI from snapshot

- Create a snapshot of a root volume for which you want to create an AMI. (This might take a few minutes)
- Select the snapshot and select “**Create Image**” option from the **Actions** menu

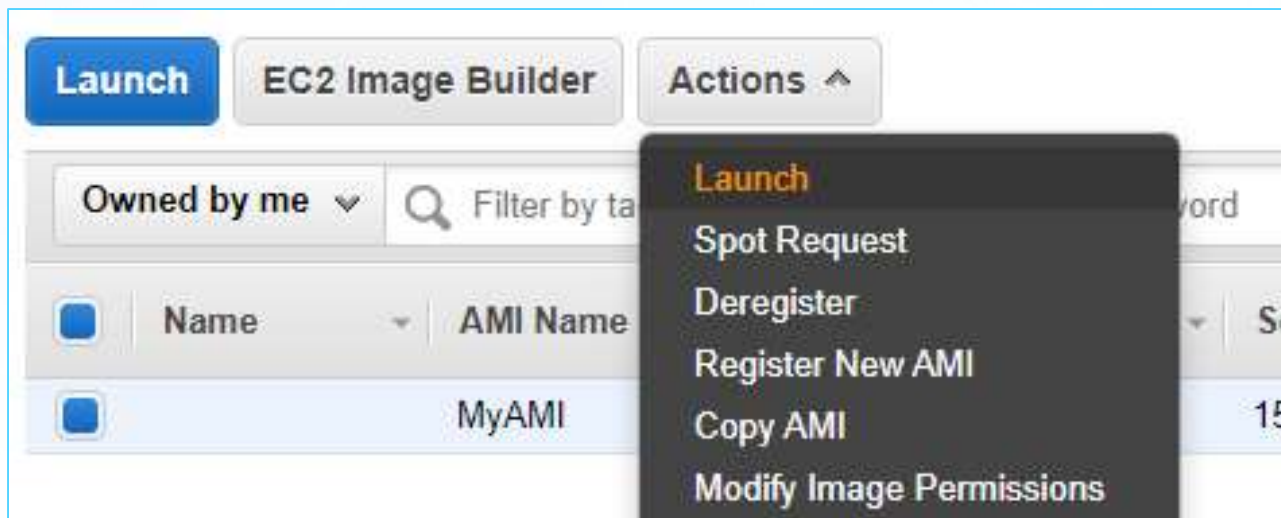
A screenshot of the 'Create Image from EBS Snapshot' dialog box. The dialog has a title bar with a close button. It contains several input fields and dropdown menus for configuring the AMI. The 'Name' field is filled with 'MyAMI'. The 'Architecture' dropdown is set to 'x86_64'. The 'Root device name' is '/dev/sda1'. The 'RAM disk ID' is set to 'Use default'. The 'Virtualization type' is 'Hardware-assisted virtualization'. The 'Kernel ID' is 'Use default'. The 'Boot Mode' is 'Use default'. Below these fields is a section for 'Block Device Mappings' with a table. The table has columns for 'Volume Type', 'Device', 'Snapshot', 'Size (GiB)', 'Volume Type', 'IOPS', 'Throughput (MB/s)', 'Delete on Termination', and 'Encrypted'. There is one row for the root volume. At the bottom of the dialog are 'Cancel' and 'Create' buttons.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-08101da0ffbc818df	30	General Purpose S	100 / 3000	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/> Not Encrypted



Launch a new EC2 instance from the AMI

- Select the AMI you just created.
- Select “**Launch**” option from the **Actions** menu.
- Select other configuration options and create the instance.



Why AMIs ?

- Amazon Machine Images (AMIs) allows us to easily create instances with preconfigured software.
- You first create one instance, install all the desired software and applications.
- You can then create a snapshot of the Root volume of this instance.
- You can then create an AMI from this snapshot
- You can then create any number of instances in any AZs using this AMI. That way, the new instances you create will have all the preinstalled software and applications that the original root volume has,

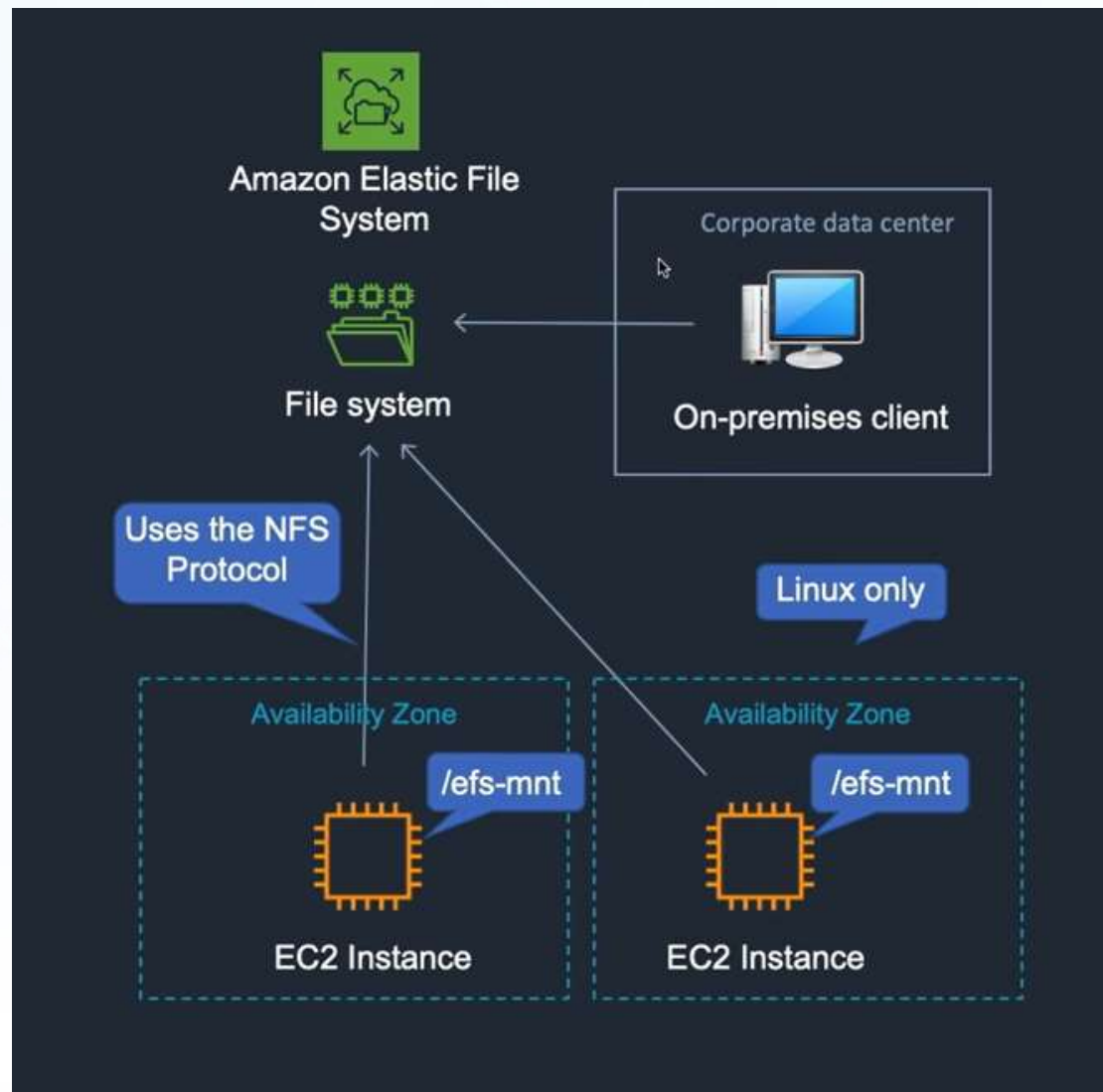




Amazon Elastic File System (EFS)



Amazon Elastic File System (EFS)



Amazon Elastic File System (EFS)

- Amazon EFS provides a simple, server-less, set-and-forget elastic file system.
- With Amazon EFS, you can create a file system, mount the file system on an Amazon EC2 instance, and then read and write data to and from your file system.
- You can mount an Amazon EFS file system in your virtual private cloud (VPC), through the Network File System versions 4.0 and 4.1 (NFSv4) protocol.



Amazon Elastic File System (EFS)

- You can access your Amazon EFS file system concurrently from multiple NFS clients, so applications that scale beyond a single connection can access a file system.
- Amazon EC2 and other AWS compute instances running in multiple Availability Zones within the same AWS Region can access the file system, so that many users can access and share a common data source.



Create an EFS

- Go to EFS landing page (Services >> Storage >> EFS) and click on **Create File System** button.
- Provide a name for the file system and click on **Create** button
- Select the file system and click on **View details** button

File systems (1)							
Filter by property values							
Name	File system ID	Encrypted	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA	Provisioned Throughput (MiB/s)	
MyEFS	fs-00263a8ea6f306bd4	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	-	

Amazon Elastic File System

Scalable, elastic, cloud-native NFS file system

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, elastic file system for general purpose workloads for use with AWS Cloud services and on-

Create file system

Create an EFS file system with service recommended settings.

Create file system

Create file system

Create an EFS file system with service recommended settings. [Learn more](#)

Name - optional
Name your file system.

MyEFS

Name must not be longer than 256 characters, and must only contain letters, numbers, and these characters: + - = . _ : /

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system. [Learn more](#)

vpc-02e3d6afcaccb1efc
default

Availability and Durability
Choose Regional (recommended) to create a file system using regional storage classes. Choose One Zone to create a file system using One Zone storage classes. [Learn more](#)

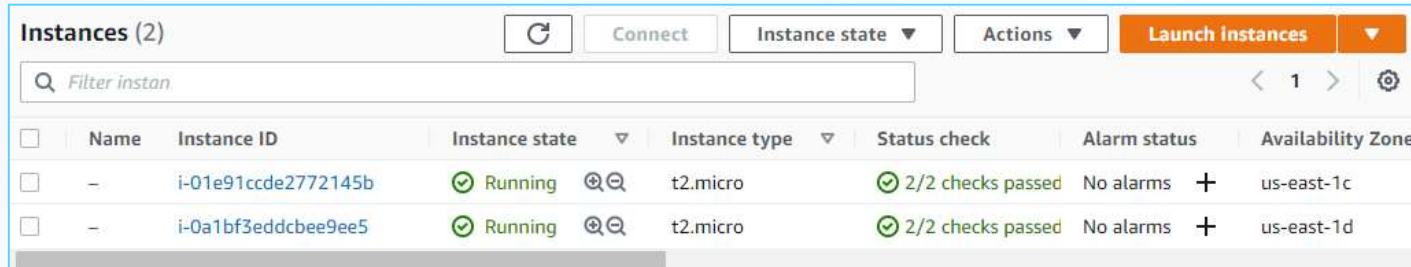
☒ Regional
Stores data redundantly across multiple AZs

☐ One Zone
Stores data redundantly within a single AZ

Cancel Customize Create

Attach an EFS to multiple EC2 instances

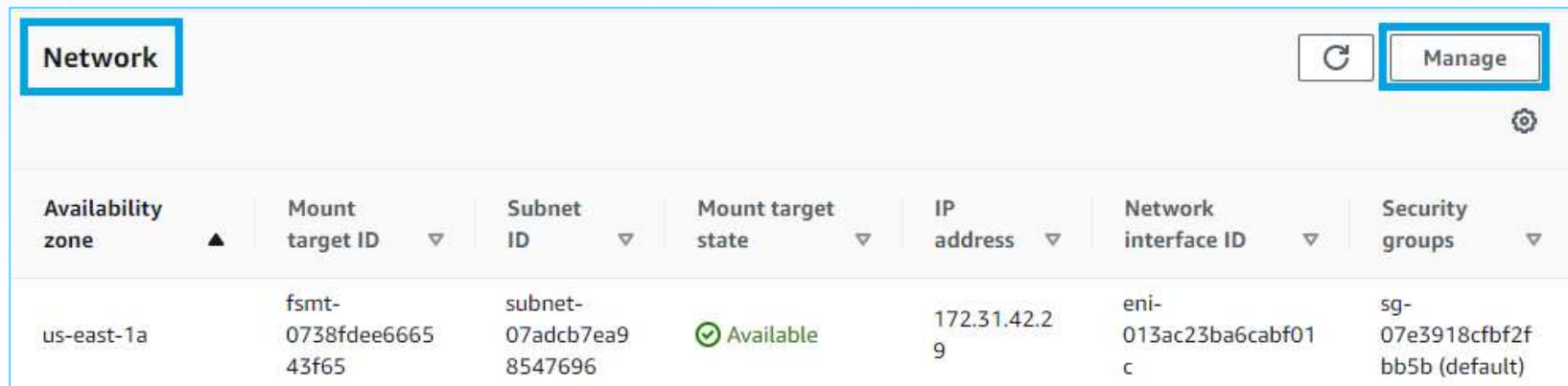
- Create two different EC2 instances in different AZs



The screenshot shows the 'Instances (2)' page in the AWS Management Console. It displays two EC2 instances, both in a 'Running' state. The first instance is in the 'us-east-1c' Availability Zone, and the second is in the 'us-east-1d' Availability Zone. Both are t2.micro instances with 2/2 status checks passed.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	-	i-01e91ccde2772145b	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c
<input type="checkbox"/>	-	i-0a1bf3eddcbee9ee5	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d

- In the details page of the EFS, click on **Network** tab and then **Manage** button



The screenshot shows the 'Network' tab of an EFS mount target in the AWS Management Console. The 'Manage' button is highlighted. The table below shows the network configuration for the mount target.

Availability zone	Mount target ID	Subnet ID	Mount target state	IP address	Network interface ID	Security groups
us-east-1a	fsmt-0738fdee666543f65	subnet-07adcb7ea98547696	Available	172.31.42.29	eni-013ac23ba6cabf01c	sg-07e3918cfbf2fbb5b (default)



Attach an EFS to multiple EC2 instances

- Make sure all the Mount targets of the EFS in all AZs have the same Security groups. You can delete and add Security groups to match with that of your EC2 instances.

Amazon EFS > File systems > fs-00263a8ea6f306bd4 > Network access

Availability zone

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system. [Learn more](#)

vpc-02e3d6afcaccb1efc
default

You must delete all existing mount targets in order to change the VPC of your file system.

Mount targets

A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone.

Availability zone	Subnet ID	IP address	Security groups	
us-east-1a	subnet-07adcb7ea98f	172.31.42.29	Choose security group...	Remove
			sg-09b7afc3b0bd81e06 Web Access	



Attach an EFS to multiple EC2 instances

- Make sure the security group you have added to the EFS mount points allows traffic from NFS (Network File System) from all IPs or at least from the security to which your EC2 instances belong.

Inbound rules (8)								Manage tags	Edit inbound rules
<input type="text" value="Filter security group rules"/>							< 1 > 		
<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol				
<input type="checkbox"/>	–	sgr-01d558cc39149d6f6	IPv4	HTTP	TCP				
<input type="checkbox"/>	–	sgr-00ad3ca092df49488	–	NFS	TCP				

- Now, open the EFS, and click on Attach button

Amazon EFS > File systems > fs-00263a8ea6f306bd4

MyEFS (fs-00263a8ea6f306bd4)


Delete Attach



Attach an EFS to multiple EC2 instances


- The **Attach** page shows the command that is you can run on a connected EC2 instance to mount a NFS file system in that EC2 instance.

Attach

Mount your Amazon EFS file system on a Linux instance. [Learn more](#) 

☒ Mount via DNS

Using the EFS mount helper:

 `sudo mount -t efs -o tls fs-00263a8ea6f306bd4:/ efs`



Connect to the EFS from EC2 instances

- Launch two different EC2 instances from two different Availability zones.
 - Refer to slides 56 & 57 for details on how to connect to EC2 instance using SSH / PuTTY

- Run the following commands on both EC2 terminals:

```
• sudo yum -y install amazon-efs-utils
• sudo mkdir /mnt/efs
• sudo mount -t efs -o tls <EFS-NAME>:/ /mnt/efs (refer prev. slide)
```

- Now cd to /mnt/efs and create any files and directories. Those will be created in the EFS. Any EC2 instance to which you mount this EFS will have access to those files and directories
- NOTES:
 - Make sure the security groups that these EFS mount points are attached to are same as that of EC2 instances and these security group allow inbound traffic from NFS (see slide 95)





Amazon Simple Storage Service (S3)



Amazon S3

- Amazon **Simple Storage Service** (Amazon **S3**) is an **object storage service** that offers industry-leading scalability, data availability, security, and performance.
- S3 is a web-based service designed for online backup and archiving of data and application programs. It allows to upload, store, and download any type of files up to 5 TB in size.
- You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere via an internet URL.
- In S3 we store objects in **buckets**
 - Each bucket can have any number of objects.
 - Bucket name must be globally unique (as it is accessed via a URL).



Amazon S3



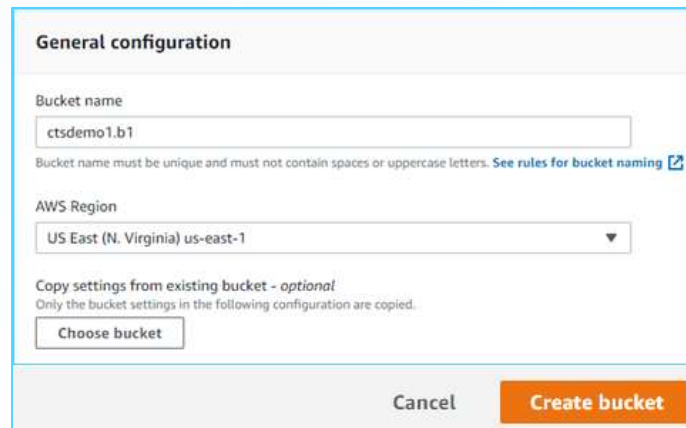
AWS S3 Buckets

- A bucket is a container on objects with a globally unique name and is accessed via an internet URL.
- A bucket is created in a region.
 - You can create the replicas of the bucket in other regions



Creating an S3 bucket

- Open Services > Storage > S3 and click on **Create bucket** button.
- General Configurations:
 - Bucket Name: Choose a globally unique name
 - AWS Region: Region in which to create the bucket
 - Block Public Access: By default, public access is blocked. Unblock if required.
 - Bucket Versioning: disabled by default. Enable if required.
 - Tag: Add if required
 - Encryption: Server-side encryption is disabled by default. Enable if required.
- Click on **Create bucket** button



General configuration

Bucket name

ctsdemo1.b1

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

Choose bucket

Cancel Create bucket



Upload objects to a bucket

- Click the bucket that you created from the list of buckets.
- Click on upload button.
 - Create new folders or upload files
- Here you can specify other options for the objects you uploaded, such as, ACL, Storage class, Encryption, Tags, Meta data etc.

Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (0)

[Remove](#) [Add files](#) [Add folder](#)

All files and folders in this table will be uploaded.

< 1 >

	Name	Folder	Type	Size
No files or folders				
You have not chosen any files or folders to upload.				

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#)

[Create folder](#) [Upload](#)

< 1 > [Settings](#)

	Name	Type	Last modified	Size	Storage class
Upload					



Upload objects to a bucket

Amazon S3 > ctsdemo1.b1

ctsdemo1.b1 [Info](#)



Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	 addresses.csv	csv	November 4, 2021, 11:03:42 (UTC+05:30)	210.0 B	Standard
<input type="checkbox"/>	 Baby_Names.csv	csv	November 4, 2021, 11:03:44 (UTC+05:30)	1.3 MB	Standard



Access an object in the S3 bucket

Properties	Permissions	Versions
Object overview		
Owner komaladeviquiz		S3 URI s3://ctsdemo1.b1/flowers1.jpeg
AWS Region US East (N. Virginia) us-east-1		Amazon Resource Name (ARN) arn:aws:s3:::ctsdemo1.b1/flowers1.jpeg
Last modified November 4, 2021, 11:10:03 (UTC+05:30)		Entity tag (Etag) 20f3d046baf02713efdc23b254c824ef
Size 62.0 KB		Object URL https://s3.amazonaws.com/ctsdemo1.b1/flo
Type jpeg		
Key flowers1.jpeg		

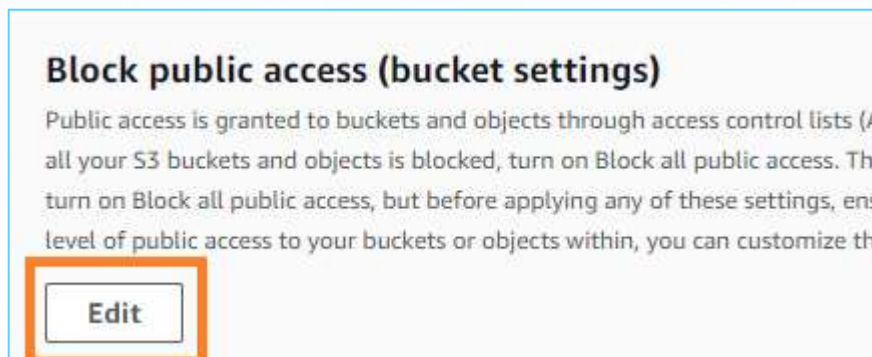


Access an object in the S3 bucket - Public Access

- If we try to paste the Object URL on a browser, we would get an access denied exception. This is because by default, the public access is blocked on a bucket.
- Open the bucket and go to **Permissions** tab.



- Go to **Block public access** option and click on **Edit**



Access an object in the S3 bucket- Bucket Policy

- Edit the Bucket Policy – Click on edit button in the Bucket Policy option.



- Paste the following code in the bucket policy box. (bucket-name mentioned in red)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::ctsdemo1.b1/*"
    }
  ]
}
```



Working with S3 from AWS CLI

- Make sure you installed AWS CLI
- Create an AWS user profile using Access keys:

```
C:\Users\USER>aws configure --profile Administrator
AWS Access Key ID [None]: AKIASJLVRN55RCQ2XAXY
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

How to get access keys ?

1. Open the IAM service and select a User
2. Open the **Security Credentials** tab
3. Click on the **Create access key** button
4. Download the Access keys CSV file



Working with S3 from AWS CLI

Run the following commands from AWS CLI:

- `aws s3 ls`
- `aws s3 ls --profile Administrator`
- `aws ls cp data\file1.txt s3://ctsdemo1.b1/ --profile Administrator`



AWS S3 Versioning

- In S3, when you create a bucket, versioning is suspended by default.
- To enable versioning:
 - Open the bucket
 - Go to properties tab
 - Click on **Edit** button in the **Bucket versioning**
 - Enable the versioning
- If versioning is enabled, when you upload the same object multiple times to the same bucket, different versions are create.
 - Open the object, click on **versions tab** to see these versions
- When you delete an object, a delete marker is created on the object.
 - You an deleted some of the delete marked files to go back to older versions.
- To delete an object from a bucket, you should not only delete the object itself, but also delete all the versions as well.





**THANK
YOU**