Day5 of Learning AWS

05 February 2025 05:16 PM

Modify EBS Volume and Extend Linux File System on AWS

Modifying EBS Volume on AWS

If you need to modify an Elastic Block Store (EBS) volume on Amazon Web Services (AWS), follow these steps using the AWS Management Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose "Volumes."
3. Select the volume you want to modify and choose "Actions," then "Modify volume."
4. On the "Modify volume" screen, you'll see the current configuration, including type, size, IOPS, and throughput. Adjust the settings as needed:

- To modify the type, select a value for Volume type.
- To modify the size, enter a new value for Size.
- For gp3, io1, and io2 types, modify IOPS by entering a new value.
- For gp3 type, modify Throughput by entering a new value.

5. After making changes, choose "Modify." Confirm the modifications when prompted.

Why Can't You Decrease the Size of an EBS Volume?

EBS volumes cannot be decreased in size due to technical constraints. When you modify a volume, AWS only allows increasing its size, not decreasing. This limitation is inherent in the architecture and ensures data integrity and consistent performance.

Extending a Linux File System After Resizing an EBS Volume

To extend a file system on Linux after resizing an EBS volume, follow these steps:

1. Connect to your instance.
2. If your volume has a partition, extend it. Use the `lsblk` command to check for a partition.

```
[ec2-user ~]$ sudo lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  8G  0 disk
└─xvda1 202:1    0   7.2G  0 part /
xvdf    202:80   0  24G  0 disk
```

   If there's a partition, proceed to the next step. If not, skip to step 3.

3. Check if the partition needs extension by comparing sizes in the `lsblk` output. If needed, extend the partition using the `resize2fs` command:

```
[ec2-user ~]$ sudo resize2fs /dev/xvdf
```

4. Verify the partition extension with `lsblk`.
5. Extend the file system. Use `df -hT` to get information about the file system.

```
[ec2-user ~]$ df -hT
Filesystem      Type   Size    Used   Avail   Use%   Mounted on
/dev/xvdf      ext4   24.0G    1.9G   22.1G    10%   /data
```

6. Depending on the file system type:

- For XFS, use `sudo xfs_growfs -d /`.
- For Ext4, use `sudo resize2fs /dev/xvda1`.

7. Verify the file system extension with `df -hT`.

These steps ensure your Linux file system is extended after resizing an EBS volume on AWS.

Click Here for Demo Video

Resize EBS Root Volume

If you need to modify an Elastic Block Store (EBS) volume on Amazon Web Services (AWS), follow these steps using the AWS Management Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose "Volumes."
3. Select the volume you want to modify and choose "Actions," then "Modify volume."
4. On the "Modify volume" screen, you'll see the current configuration, including type, size, IOPS, and throughput. Adjust the settings as needed:

- To modify the type, select a value for Volume type.
- To modify the size, enter a new value for Size.
- For gp3, io1, and io2 types, modify IOPS by entering a new value.
- For gp3 type, modify Throughput by entering a new value.

5. After making changes, choose "Modify." Confirm the modifications when prompted.

Why Can't You Decrease the Size of an EBS Volume?

EBS volumes cannot be decreased in size due to technical constraints. When you modify a volume, AWS only allows increasing its size, not decreasing. This limitation is inherent in the architecture and ensures data integrity and consistent performance.

Extending a Linux File System After Resizing an EBS Volume

To extend a file system on Linux after resizing an EBS volume, follow these steps:

1. Connect to your instance.
2. If your volume has a partition, extend it. Use the `lsblk` command to check for a partition.

   ```
   [ec2-user ~]$ sudo lsblk
   NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   xvda    202:0    0  16G  0 disk
   └─xvda1 202:1    0   8G  0 part /
   xvdf    202:80   0  24G  0 disk
   ```

   If there's a partition, proceed to the next step. If not, skip to step 3.
3. Check if the partition needs extension by comparing sizes in the `lsblk` output. If needed, extend the partition using the `growpart` command:

   ```
   [ec2-user ~]$ sudo growpart /dev/xvda 1
   ```
4. Verify the partition extension with `lsblk`.
5. Extend the file system. Use `df -hT` to get information about the file system.

   ```
   [ec2-user ~]$ df -hT
   Filesystem     Type  Size   Used  Avail  Use%  Mounted on
   /dev/xvda1     ext4  8.0G   1.9G  6.2G   24%   /
   ```
6. Depending on the file system type:
   - For XFS, use `sudo xfs_growfs -d /`.
   - For Ext4, use `sudo resize2fs /dev/xvda1`.
7. Verify the file system extension with `df -hT`.

These steps ensure your Linux file system is extended after resizing an EBS volume on AWS.

Click Here for Demo Video

Attach One EBS Volume to Multiple EC2 Instance

Amazon EBS Multi-Attach simplifies attaching a single Provisioned IOPS SSD (io1 or io2) volume to multiple instances in the same Availability Zone. This feature enhances application availability, especially for applications managing concurrent write operations.

Considerations and Limitations

Before diving in, let's go through some key points:

1. **Number of Instances:** Up to 16 Linux instances on the Nitro System can share a Multi-Attach enabled volume in the same Availability Zone.
2. **Volume Types:** Multi-Attach is exclusive to Provisioned IOPS SSD (io1 and io2) volumes.
3. **Regions:** Multi-Attach for io1 volumes is limited to specific regions, while io2 supports it across all regions.
4. **File System Recommendation:** Standard file systems like XFS and EXT4 aren't designed for simultaneous access by multiple servers. Consider a clustered file system for data resiliency.
5. **I/O Fencing:** io2 volumes support I/O fencing for data consistency, while io1 volumes do not.

Performance

Performance considerations include:

- Each attached instance can reach its maximum IOPS performance.
- Aggregate performance of all attached instances cannot exceed the volume's maximum provisioned performance.

Working with Multi-Attach

To manage Multi-Attach enabled volumes:

1. **Enable Multi-Attach During Volume Creation:**
   - Use the AWS Console or Command Line during volume creation.
   - Choose "Provisioned IOPS SSD (io1/io2)" as the volume type and enable Multi-Attach in the options.
2. **Enable Multi-Attach for io2 After Creation:**
   - Use the AWS Console or Command Line for io2 volumes not attached to any instances.
3. **Disable Multi-Attach:**
   - Available through the AWS Console or Command Line.
4. **Attach Volume to Instances:**
   - Attach Multi-Attach enabled volumes to instances following standard procedures.
5. **Delete on Termination:**
   - Configure Multi-Attach volumes to be deleted on instance termination.

Click Here for Demo Video

Amazon EBS Volume types

Introduction:

Understanding Amazon Elastic Block Store (EBS) volume types is crucial for optimizing performance and costs in your AWS environment. In this guide, we'll break down the three main EBS volume types

- General Purpose SSD
- Provisioned IOPS SSD
- Throughput Optimized HDD/Cold HDD

1. General Purpose SSD Volumes:

Overview

General Purpose SSD volumes, represented by the gp3 type, offer a balanced mix of performance and cost-effectiveness for a variety of workloads.

Use Cases:

1. **Virtual Servers Boot Volumes:**
- Ideal for launching virtual servers quickly.
- Balanced I/O performance for boot volumes.

2. **Small to Medium-sized Databases:**
- Cost-effective storage for databases with moderate I/O needs.
- Suitable for applications with diverse workloads.

3. **Development and Test Environments:**
- Affordable storage for non-production environments.
- Balanced performance for testing applications.

2. Provisioned IOPS SSD Volumes:

Overview

Provisioned IOPS SSD volumes, represented by io2 and io1 types, are designed for applications demanding consistent and predictable I/O performance.

Use Cases:

1. **High-performance Databases:**
- Ensures low-latency and high-throughput for transactional databases.
- Critical for applications requiring intensive I/O operations.

2. **Business-critical Applications:**
- Predictable performance for applications sensitive to I/O latency.
- Customizable IOPS settings for specific performance requirements.

3. **Big Data and Analytics Applications:**
- High-performance storage for data-intensive analytics workloads.
- Consistent I/O throughput for big data processing.

3. Throughput Optimized HDD and Cold HDD Volumes:

Overview:

Throughput Optimized HDD and Cold HDD volumes are designed for large, sequential I/O workloads, with Throughput Optimized HDD suitable for frequent access and Cold HDD for infrequent access.

Use Cases

1. **Throughput Optimized HDD - Big Data Processing:**
- Ideal for Hadoop and Spark environments.
- High throughput for large-scale data processing.

2. **Throughput Optimized HDD - Data Warehousing:**
- Well-suited for data warehousing applications.
- Supports large sequential data access patterns.

3. **Cold HDD - Data Archiving and Backup:**
- Cost-effective storage for infrequently accessed data.
- Perfect for long-term storage compliance needs.

Conclusion:

Choosing the right Amazon EBS volume type is pivotal for optimizing your AWS storage. Whether you need balanced performance, high I/O predictability, or throughput for specific workloads, understanding these EBS types empowers you to make informed decisions. Consider the use cases outlined above to guide your selection process and strike the perfect balance between performance and cost-efficiency.

Sure, here is the information presented in a table:

Feature	**General Purpose SSD volumes (gp3)**	**Provisioned IOPS SSD volumes (io1, io2)**
Volume type	gp3	io1, io2, Block Express 3
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)
Use cases	- Transactional workloads	- Workloads that require Sub-millisecond latency
	- Virtual desktops	- Sustained IOPS performance
	- Medium-sized, single-instance databases	- More than 64,000 IOPS or 1,000 MiB/s of throughput
	- Low-latency interactive applications	- Workloads that require sustained IOPS performance

	- Boot volumes	- I/O-intensive database workloads
	- Development and test environments	
	- Workloads that require:	
	Sub-millisecond latency	
	Sustained IOPS performance	
	More than 64,000 IOPS or 1,000 MiB/s of throughput	
	- Workloads that require sustained IOPS performance	
	- I/O-intensive database workloads	
Volume size	1 GiB - 16 TiB	4 GiB - 64 TiB (io1, io2), 4 GiB - 16 TiB (Block Express 3)
Max IOPS per volume	16,000 (64 KiB I/O)	16,000 (16 KiB I/O) (io1, io2), 256,000 (16 KiB I/O) (Block Express 3)
Max throughput per volume	1,000 MiB/s	250 MiB/s (io1, io2), 4,000 MiB/s (Block Express 3)
Amazon EBS Multi-attach	Not supported	Supported
NVMe reservations	Not supported	Supported (io1, io2)
Boot volume	Supported	Supported

Please note that the information is based on the latest available data and may be subject to changes. Always refer to the official AWS documentation for the most up-to-date information.

Click Here for Demo Video

For more in-depth information and updates, refer to the official AWS documentation.

Amazon EBS Snapshots

Introduction:

In the vast landscape of AWS services, Amazon Elastic Block Store (EBS) provides a robust solution for storing data. One key aspect of ensuring data resiliency and disaster recovery is the creation of point-in-time copies known as Amazon EBS snapshots. Snapshots are like snapshots in a camera—they capture changes in your EBS volume over time. When you take periodic snapshots, only the blocks that have changed since the last snapshot are saved. It's like taking a picture of just the things that have changed since the last photo.

Features of Amazon EBS Snapshots:

- You can take a snapshot of a volume even while it's in use. However, there's a catch. The snapshot only includes data written to the volume at the time of the snapshot. If you can pause file writes, great! If not, unmount the volume, take the snapshot, and remount it for a complete picture.
- Encrypted volumes and their snapshots are automatically encrypted, providing extra security for your data. You can also share your snapshots with others, but for encrypted ones, they need the encryption key to access and use it.
- You can create snapshots for all or some volumes attached to an instance. Tagging makes it easier to manage multiple volume snapshots collectively during restore, copy, or retention.
- Automate snapshot creation and retention with snapshot lifecycle policies using Amazon Data Lifecycle Manager. This simplifies the snapshot management process.

Understanding Incremental Backups

EBS snapshots serve as incremental backups, capturing only the changed blocks on the device since the last snapshot. This efficiency minimizes snapshot creation time and optimizes storage costs by avoiding unnecessary duplication of data.

User Responsibility

It's crucial to note that AWS doesn't automatically back up data on EBS volumes. Users are responsible for regular EBS snapshot creation or can opt for automatic snapshot creation using tools like Amazon Data Lifecycle Manager or AWS Backup for enhanced data resiliency and disaster recovery.

Snapshot Management:

Storage in Amazon S3

EBS snapshots are stored in Amazon S3, offering durability and accessibility. However, users can't directly access the snapshots through the Amazon S3 console or API. Instead, snapshot creation and management can be performed through the Amazon EC2 console or API.

Snapshot Restoration

Each snapshot carries all the information needed to restore data to a new EBS volume. Upon creating a volume based on a snapshot, it becomes an exact replica of the original volume. The new volume loads data in the background, enabling immediate utilization. If accessing data not yet loaded, the volume seamlessly fetches it from Amazon S3.

Deletion Clarification

Deleting a snapshot removes only the data unique to that snapshot. Data referenced by other snapshots remains intact. This ensures efficient storage management while eliminating unnecessary data redundancy.

Snapshot Events and Multi-Volume Snapshots:

CloudWatch Events Integration: Snapshot status tracking is facilitated through CloudWatch Events, providing users with insights into the health and status of their EBS snapshots.

Multi-Volume Snapshots: For complex workloads spanning multiple EBS volumes, multi-volume snapshots offer exact point-in-time, data-coordinated, and crash-consistent backups without the need to stop instances or coordinate between volumes. This simplifies snapshot management for critical workloads.

Snapshot Pricing:

Incremental Storage Charges: Snapshot charges are based on stored data amount. As snapshots are incremental, deleting a snapshot may not immediately reduce storage costs. Data exclusive to a snapshot is removed upon deletion, while data referenced by other snapshots is retained.

Billing Considerations: For detailed information on snapshot pricing, users can refer to the AWS Billing User Guide.

Easily Copy Amazon EBS Snapshots

1. Copy Snapshots Anywhere

Once you create a snapshot and it's safely stored in Amazon S3, you can easily copy it to another place. This can be another AWS Region or even within the same place. Your data is protected during this process with strong encryption.

2. Copying Multi-Volume Snapshots

If you have multiple snapshots, you can tag them to make it easier to find and copy them. Copying them individually to another place is a breeze.

3. Sharing Snapshots

If you want someone else to copy your snapshot, you can allow access or make it public. It's like sharing a file with a friend.

4. Use Cases

- **Geographic Expansion:** Start using your applications in a new AWS Region easily.
- **Migration:** Move your application to a new place for better availability and cost.
- **Disaster Recovery:** Back up your data in different places. If something goes wrong, you can quickly restore your applications using these backups.

5. Encryption and Security

- **Encrypt Snapshots:** Keep your snapshots safe by encrypting them.
- **Change Encryption Key:** Change the key used to encrypt a snapshot for added security.
- **Data Retention:** Copy your data to another AWS account for safekeeping. This protects you in case of accidental deletions or if your main account is compromised.

In a nutshell, copying EBS snapshots is a smart way to keep your data safe, whether you're expanding, moving, or just being cautious. Start protecting your data today!

Click Here for Demo Video

For more detailed insights, visit the Amazon EBS Snapshots product page.