Day6 of learning AWS

06 February 2025 05:36 AM

Create First Snapshot

Follow the steps below to generate a snapshot from an EBS volume

1. Go to the Amazon EC2 console.
2. In the side menu, click on "Snapshots" and then select "Create snapshot."
3. Choose "Volume" as the resource type.
4. Select the specific volume for which you want to create the snapshot.
5. The Encryption field shows if the volume is encrypted. If it is, the snapshot will be automatically encrypted using the same key. If not, the snapshot won't be encrypted.
6. (Optional) Provide a brief description for the snapshot.
7. (Optional) Add custom tags in the Tags section by clicking "Add tag" and entering key-value pairs. You can add up to 50 tags.
8. Click on "Create snapshot" to complete the process.

To create a volume from an EBS snapshot, follow these steps

Here's a simpler version:

1. Go to the Amazon EC2 console.
2. In the menu, click on "Volumes."
3. Choose "Create volume."
4. Select the type of volume you want to create (the default is General Purpose SSD gp3).
5. Enter the size of the volume in gigabytes (GiB).
6. For certain types (io1, io2, and gp3), you can specify the maximum number of input/output operations per second (IOPS) or throughput.
7. Choose the Availability Zone where you want to create the volume. It must be the same as the instances you plan to attach it to.
8. Pick the snapshot you want to use to create the volume.
9. Set the encryption status. If the snapshot is encrypted, it's automatic; if not, you can choose to encrypt it and select the encryption key.
10. (Optional) Add custom tags to the volume for easier identification.
11. Click on "Create Volume."

Remember, your volume is ready when the status is "available." Attach it to an instance to start using it.
Click Here for Demo Video

Automate EBS Volume Snapshot

Considerations for Snapshot Lifecycle Policies:

1. **Region-Specific Targeting:**
   - Snapshot lifecycle policies work within the same AWS Region as the policy.
2. **Snapshot Creation Timing:**
   - The first snapshot starts within an hour of the specified time, followed by subsequent snapshots at their scheduled times.
3. **Multiple Policies for a Volume or Instance:**
   - You can use multiple policies for one volume or instance, each with its own schedule.
   - Tags are case-sensitive in targeting resources.
4. **Managing Existing Snapshots:**
   - If you remove target tags, existing snapshots are no longer managed, and manual deletion is required.
5. **Policy Impact on New Volumes:**
   - If new volumes are attached to a target instance, they are included in the backup during the next policy run.
6. **Custom Cron-Based Schedule:**
   - Custom cron-based schedules creating a single snapshot won't auto-delete when the retention threshold is met.
7. **Age-Based Policy Consideration:**
   - In age-based policies, if the retention period is shorter than the creation frequency, the last snapshot is retained until the next one is created. **Snapshot Archiving Considerations:**
8. **Snapshot Archiving Restrictions:**
   - Archiving is allowed only for volume-targeting snapshot policies.
   - One archiving rule per schedule is specified.
9. **Archiving Minimum Retention:**
   - The minimum retention period in the archive tier is 90 days.
10. **Snapshot Archiving Impact:**
    - Archiving converts snapshots to full snapshots, potentially increasing storage costs.
11. **Snapshot Sharing and Archiving:**
    - Fast snapshot restore and snapshot sharing are disabled for archived snapshots.
12. **Snapshot Archiving Retry:**

- If archiving fails for 24 hours, the snapshot remains in the standard tier, scheduled for deletion as if it had been archived.

13. **Tagging of Archived Snapshots:**
    - Archived snapshots are tagged for identification.
14. **Excluding Root Volumes and Data Volumes:**
    - Excluding root volumes impacts snapshot creation for the entire instance.
15. **Deleting Volumes or Terminating Instances:**
    - Deleting volumes or terminating instances affects snapshots based on the retention schedule.
16. **Fast Snapshot Restore:**
    - Enabled only for snapshots 16 TiB or less, and charges apply per minute.
17. **Multi-Attach Enabled Volumes:**
    - When targeting instances with Multi-Attach enabled volumes, separate snapshots are initiated for each attached instance.
18. **Snapshot Sharing Across Accounts:**
    - Encrypted snapshots require sharing the KMS key; default encryption KMS key snapshots can't be shared.
19. **Snapshot Archiving and Recycle Bin:**
    - Manually archived snapshots in the Recycle Bin must be managed manually.
20. **Policies in Error State:**
    - Policies in error state affect snapshot retention; manual deletion may be required.
21. **Snapshot Lock Considerations:**
    - Manually locked snapshots need manual deletion if still locked when their retention threshold is reached.

These considerations provide insights into managing snapshots effectively using lifecycle policies in AWS. Sure, here are simplified steps to create a snapshot policy in Amazon EC2:

1. Go to the Amazon EC2 website: https://console.aws.amazon.com/ec2/.
2. In the menu, select Elastic Block Store, then Lifecycle Manager, and click on Create lifecycle policy.
3. Choose EBS snapshot policy and click Next.
4. Specify the type of resource to back up (Volume or Instance) in the Target resources section.
5. If using AWS Outpost, choose the location of the target resources.
6. Choose the resource tags to identify volumes or instances for backup.
7. Enter a brief description for the policy.
8. Choose the IAM role that has permissions for managing snapshots.
9. Add tags for identification and categorization.
10. Choose to enable or disable the policy for immediate or manual start.
11. If targeting instances, decide whether to exclude volumes from multi-volume snapshot sets.
12. Configure the policy schedules, including name, frequency, start time, and retention type.
13. Specify the snapshot destination, tagging, and pre/post scripts if needed.
14. For volume-targeting policies, configure snapshot archiving.
15. Enable fast snapshot restore if desired, and choose the Availability Zones.
16. Configure cross-Region copy if needed.
17. Configure cross-account sharing if required.
18. Add additional schedules if necessary.
19. Review the policy summary.
20. Finally, choose to create the policy.

https://www.youtube.com/watch?v=s3gMJU9Nc7U&list=PL6XT0grm_TfqtwtwUit305qS-HhDvb4du&index=32

Recycle Bin Retention Rules

The Recycle Bin is like a safety net for your data in Amazon EBS. If you accidentally delete snapshots or AMIs, they aren't immediately gone. Instead, they go to the Recycle Bin for a specific time before being permanently deleted.
You have the power to bring back any deleted resource from the Recycle Bin as long as it's within the set time. Once you restore something, it leaves the Recycle Bin and becomes a fully functional part of your resources again. However, if you don't restore it in time, after the specified period, it's gone for good.
By using the Recycle Bin, you're adding an extra layer of protection to make sure important data doesn't get lost due to accidental deletions, helping to keep your business running smoothly.

How it's works

1. **Setting it Up:**
    - You make rules in the AWS Regions to decide what kind of stuff you want to protect.
    - Rules say which types of things to keep in the Recycle Bin when they get deleted.
    - You also set how long to keep things in the Recycle Bin before they vanish for good.
2. **Types of Rules:**
    - Tag-level rules: If you want to protect specific things based on their tags (labels), you use this. You pick certain tags, and anything with those tags goes to the Recycle Bin when deleted.
    - Region-level rules: If you want to protect all things of a certain type in a specific place (Region), you use this. It doesn't need tags; it covers everything of that type in that area.
3. **What Happens in the Recycle Bin:**
    - When something is in the Recycle Bin, you can bring it back anytime.
    - It stays there until either:
    - You bring it back, and then it's ready to use.
    - The set time passes, and if you didn't bring it back, it's gone for good.
4. **What it Protects:**

- It looks after Amazon EBS snapshots (like saving a copy of your data at a certain time).
- Also, it keeps an eye on Amazon EBS-backed Amazon Machine Images (AMIs – like a saved version of a computer setup).

Important Notes:

- The rules also apply to stored snapshots, and if you delete one that matches a rule, it stays in the Recycle Bin for the rule's time.
- Rules work even for disabled AMIs (ones not actively in use).

When it comes to safeguarding your valuable AWS resources, setting up retention rules in the Recycle Bin is a crucial step. This guide will walk you through the process of creating retention rules, ensuring that your deleted resources are securely stored before permanent deletion. Let's simplify this process step by step.

Step 1: Understand Required Parameters: To start, you need to specify the resource type you want to protect and the resources that fall under this rule. There are two levels to consider:

- **Tag-level Retention Rule:** Identify resources based on tags. You can assign up to 50 tags to each rule, making it highly customizable.
- **Region-level Retention Rule:** Protect all resources of a specific type in a particular Region without the need for tags.

Step 2: Set the Retention Period: Decide how long you want to retain the resources in the Recycle Bin after deletion. This period can range up to 1 year (365 days), providing flexibility based on your specific needs.

Step 3: Optional Parameters: You have the option to add more details:

- **Name and Description:** Make it easier to manage by giving your retention rule a descriptive name and a brief description.
- **Retention Rule Tags:** Use custom tags to organize and identify your retention rules effectively.
- **Locking Rules:** Optionally, you can lock retention rules on creation, providing an extra layer of security. If you choose to lock, specify the unlock delay period (7 to 30 days).

Step 4: Creation Methods: You can create a Recycle Bin retention rule using either the Recycle Bin console or the AWS CLI. Choose the method that suits your preference.

Step 5: Walkthrough Using the Recycle Bin Console:

1. Open the Recycle Bin console here.
2. In the navigation pane, select "Retention rules" and click "Create retention rule."
3. Fill in the rule details and settings as per your requirements.
4. Choose to lock or leave the retention rule unlocked based on your security preferences.
5. Optionally, add custom tags for better organization.
6. Click "Create retention rule" to complete the process.

By following these simplified steps, you can efficiently create Recycle Bin retention rules, ensuring the safety and recoverability of your AWS resources.

Conclusion: Securing your AWS resources doesn't have to be complicated. With Recycle Bin retention rules, you gain control over the fate of your deleted resources. Follow this guide, and you'll be on your way to a more organized and secure AWS environment.

Recover Snapshot from Recycle Bin

Title: Restoring Snapshots from the Recycle Bin - Step-by-Step Guide

Instructions:

1. **Accessing the Recycle Bin Console:** Open your web browser and go to https://console.aws.amazon.com/rbin/home/.
2. **Navigating to the Recycle Bin:** In the navigation pane on the left, locate and click on "Recycle Bin."
3. **Locating Snapshots:** The grid displayed on this page provides an overview of all snapshots currently residing in the Recycle Bin. Identify the specific snapshot you want to restore.
4. **Initiating the Restore Process:**
   - Select the snapshot you wish to recover.
   - Click on "Recover" to begin the restoration process.
5. **Confirmation Prompt:** When prompted, affirm your decision by choosing "Recover" once again.

Congratulations, you have successfully restored a snapshot from the Recycle Bin using the console!
https://www.youtube.com/watch?v=x2T5egBIdQg&list=PL6XT0grm_TfqtwtwUit305qS-HhDvb4du&index=33

Copy Snapshot From One Region to Another- Copy Snapshot Cross Region/Account

In Amazon EBS, you can easily create snapshots of your volumes, capturing a moment in time for your data. These snapshots are securely stored in Amazon S3. Once a snapshot is complete, you have the flexibility to copy it to another AWS Region or within the same Region. During this copy process, Amazon S3 ensures the data's security with server-side encryption.

Key Points

- Every copied snapshot receives a unique ID, different from the original snapshot.
- To copy multi-volume snapshots across Regions, identify them using the applied tag and copy each snapshot individually.
- If you want another account to copy your snapshot, adjust the snapshot permissions accordingly or make it public.
- Launch applications in a new AWS Region effortlessly.
- Move applications to enhance availability and reduce costs.
- Safeguard data across regions, enabling quick application restoration in case of a disaster.
- Modify encryption, change encryption keys, or create a copy for creating volumes from shared, encrypted snapshots.
- Copy encrypted EBS snapshots to preserve data for auditing or retention, adding an extra layer of security.

Considerations to Keep in Mind:

- A limit of 20 concurrent snapshot copy requests per destination Region exists. Wait for ongoing requests to finish before initiating new ones.
- Tags from the source snapshot aren't automatically copied. Add user-defined tags during or after the copy operation as needed.

- Snapshots created during a copy operation have arbitrary volume IDs (e.g., vol-ffff), which shouldn't be used for any specific purpose.
- Permissions specified for the copy operation apply only to the new snapshot, not the source snapshot.

Copying a snapshot is a straightforward process, and here's a guide to help you through it:

1. Open the Amazon EC2 console by navigating to https://console.aws.amazon.com/ec2/.
2. In the navigation pane, select "Snapshots."
3. Pick the snapshot you want to copy, then go to "Actions" and select "Copy snapshot."
4. Enter a brief description for the snapshot copy. The default description includes details about the source snapshot, aiding in easy identification. Modify it if necessary.
5. Specify the Region where you want to create the snapshot copy.
6. Determine the encryption status for the snapshot copy.
   - If the source snapshot is already encrypted or your account defaults to encryption, the copy is automatically encrypted, and this setting cannot be changed.
   - For unencrypted source snapshots and accounts without default encryption, you have the option to encrypt the copy. Choose "Encrypt this snapshot," and select the desired KMS key for encryption in the destination Region.
7. Click on "Copy snapshot" to initiate the copying process.

After completing the process, you'll find a newly created snapshot in the selected region, ready for your use.
https://www.youtube.com/watch?v=UJvKOo00xvU&list=PL6XT0grm_TfgtwtwUit305qS-HhDvb4du&index=34

AWS EBS Encryption

Encrypting an Amazon Elastic Block Store (EBS) volume has several implications and effects on the data stored in the volume. Here are the key points to consider:

1. **Encryption of Data in Transit:**
   - When you enable encryption for an EBS volume, all data moving between the volume and the instance is encrypted. This helps secure the data while it is being transferred.
2. **Encryption at Rest:**
   - Encryption at rest ensures that the data stored on the EBS volume is encrypted. This includes both the data blocks and any snapshots created from the volume.
3. **KMS Key Usage:**
   - EBS volume encryption uses AWS Key Management Service (KMS) keys. You can choose to use the default AWS managed key for EBS or specify a customer-managed key (CMK) in AWS KMS.
4. **Impact on Snapshots:**
   - If you create a snapshot of an encrypted EBS volume, the snapshot is also encrypted. When you launch an instance from an encrypted snapshot, the resulting EBS volumes will be encrypted as well.
5. **Performance Considerations:**
   - Encrypting and decrypting data does introduce some overhead. While modern hardware and AWS infrastructure minimize this impact, it's worth considering if you have performance-sensitive workloads.
6. **Changing Encryption Status:**
   - Once an EBS volume is encrypted, you cannot change its encryption status. Similarly, you cannot modify the encryption status of an existing snapshot. If you want to change the encryption status, you need to create a new encrypted volume or snapshot.
7. **Sharing Encrypted Snapshots:**
   - If you share an encrypted snapshot with another AWS account, they must have the necessary permissions to use the KMS key associated with the snapshot.
8. **Limitations on Snapshot Copying:**
   - When copying an unencrypted snapshot, you can choose to encrypt the copy. However, when copying an encrypted snapshot, the copy will always be encrypted, and you cannot choose to create an unencrypted copy.

https://www.youtube.com/watch?v=0MltSofWHTs&list=PL6XT0grm_TfgtwtwUit305qS-HhDvb4du&index=35

Amazon EC2 snapshot deletion

To remove a snapshot using the console, follow these steps:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click on "Snapshots."
3. Choose the snapshot you want to delete.
4. Click on "Actions" and then select "Delete snapshot."
5. Confirm the deletion by selecting "Delete."