



Morris & Opazo

Centralize and automate your data on AWS Backup

We Love the Cloud

aws

PARTNER
Advanced Tier
Services

- Public Sector
- Immersion Day
- Solution Provider
- AWS Lambda Delivery
- Amazon Kinesis Delivery
- Amazon Redshift Delivery
- Amazon API Gateway Delivery
- Data & Analytics Services Competency
- Amazon EC2 for Windows Server Delivery

[morrisopazo.com](https://www.morrisopazo.com)

contacto@morrisopazo.com



A service that significantly simplifies the centralized management and automation of your data protection. Learn how to back up your services automatically and how to restore them in just a couple of clicks.

Whether you are using cloud services or on-premises components, AWS Backup is capable of generating backups of these as a safe and stable point with which to recover them easily and quickly in case the service in question suffers any inconvenience.

One of the biggest needs customers had when working in depth with AWS was to have one place where they can manage, configure, and regulate all their backup activity across all accounts and resources in their organization.

In the past, this was only possible if the client made backups on their own and restored their services manually, so when they had to do the same with several databases, instances, storage devices, etc., it became an extremely tedious task and in general impractical in terms of time and money used.

AWS Backup was born to provide a solution to the needs of customers in a secure, centralized and cost-effective way.

Through the following example you will learn how to make your backups automatically periodically according to your needs, as well as how to restore them in case of an emergency.

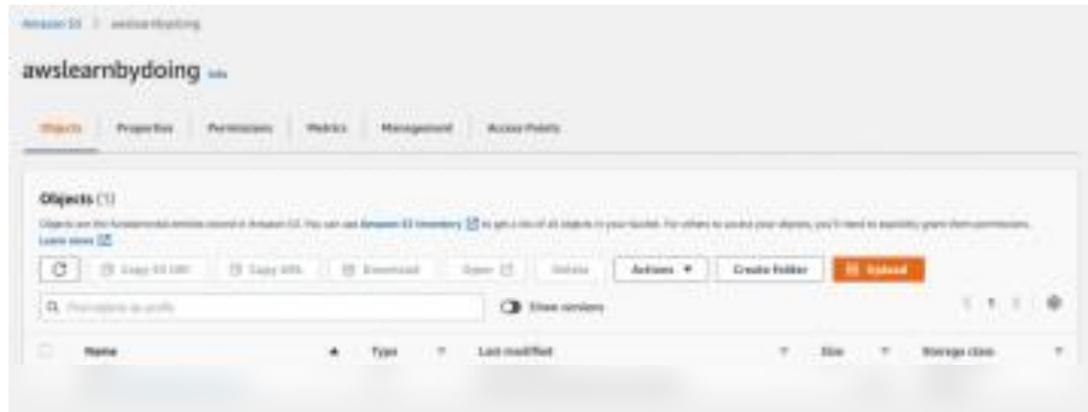
In essence, the process consists of creating a Security Backup Plan in which we will define the rules that it must follow and the permissions that it will have to carry out these actions and in case of wanting to make a restoration, select the point to which we want to return and define if We make changes at the component level and availability of our service.

Currently AWS Backup natively supports the following services:

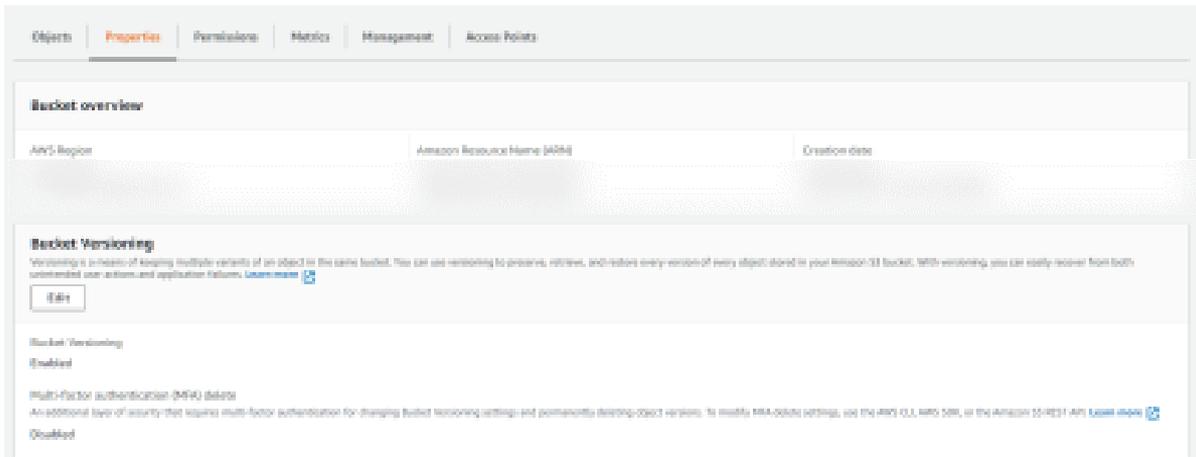
Amazon EC2, Amazon RDS, DocumentDB, VSS, Amazon Aurora, Amazon Neptune, Amazon S3, Amazon EFS, VMware, Amazon EBS, FSx (Luster & WFS), Dynamo DB, Volume Gateway.

Being the newest in this list VMware and Amazon S3, we will use the latter to restore a deleted object

First, we locate the Bucket or service to use and the elements it contains.



We make sure that versioning is activated



The screenshot shows the AWS console interface for a bucket. The 'Properties' tab is selected. Under the 'Bucket overview' section, there is a table with columns for 'AWS Region', 'Amazon Resource Name (ARN)', and 'Creation date'. Below this, the 'Bucket Versioning' section shows a description of versioning and an 'Edit' button. At the bottom, the 'Bucket Versioning' status is shown as 'Enabled', and the 'Multi-factor authentication (MFA) delete' status is shown as 'Disabled'.

Edit Bucket Versioning Info

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Suspend
This suspends the creation of object versions for all operations but preserves any existing object versions.

Enable

Multi-factor authentication (MFA) delete

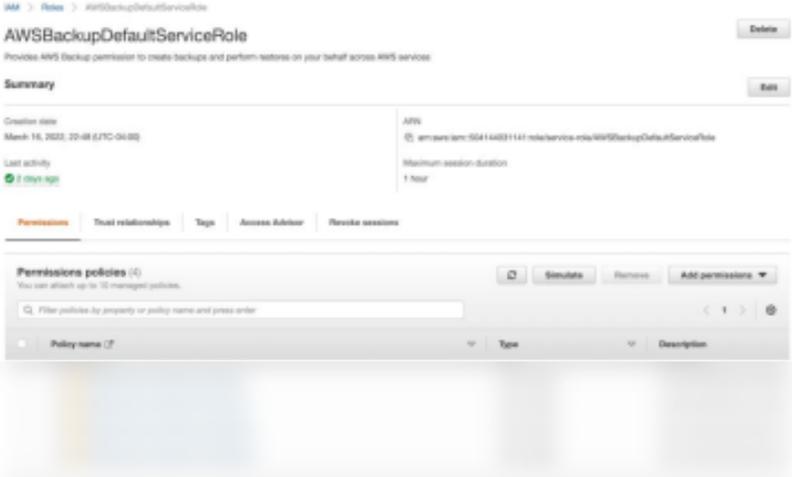
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

Disabled

Cancel

Save changes

In order for our Backup Plan to carry out all the necessary actions, we must grant the pertinent permissions to the services on which it is going to act, for this we will create an IAM role that we will later assign to our Backup Plan. If we are going to use a specific service, AWS also provides us with a role called “Default Role” that includes (in the case of S3) the following permissions:



AWSCredentials > Roles > AWSBackupDefaultServiceRole

AWSBackupDefaultServiceRole

Provides AWS Backup permission to create backups and perform actions on your behalf across AWS services

Summary

Creation date: March 16, 2022, 22:48 (UTC-04:00)

Last activity: 2 days ago

ARN: arn:aws:iam::554144211441:role/service-role/AWSBackupDefaultServiceRole

Maximum session duration: 1 hour

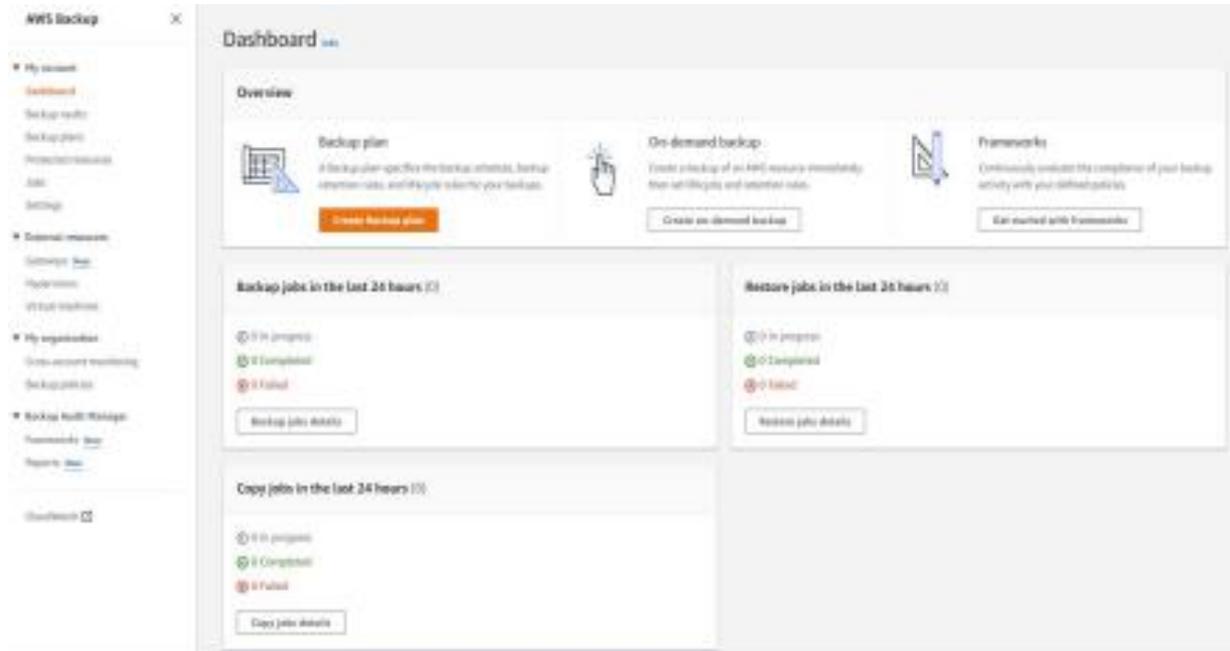
Permissions policies (0)

You can attach up to 12 managed policies.

Filter policies by property or policy name and press enter

Policy name (F)	Type	Description
-----------------	------	-------------

To create our Backup Plan we go to our AWS Backup dashboard and select “Create Backup Plan”



Once here we have the possibility to choose a pre-made template or create one according to our needs. In this case we will create a new plan

Start options

Choose how you want to begin. [Info](#)

Start with a template
Create a Backup plan based on a template provided by AWS Backup.

Build a new plan
Configure a new Backup plan from scratch.

Define a plan using JSON
Modify the JSON expression of an existing backup plan or create a new expression.

Backup plan name
Name your backup plan

Backup plan name is case sensitive. Must contain from 1 to 50 alphanumeric or '-' characters.

► **Tags added to backup plan**

Backup rule configuration [Info](#)

Add a Backup rule by defining a backup schedule, backup window, and lifecycle rules. You can add additional Backup rules to this Backup plan later. The backup cost depends on your backup configurations.

Backup rule name

Backup rule name is case sensitive. Must contain from 1 to 50 alphanumeric or `"/_` characters.

Backup vault [Info](#)

Default

Backup frequency [Info](#)

Daily

Enable continuous backups for point-in-time recovery (PITR) [Info](#)
Available for RDS and S3 resources.

Backup window

Use backup window defaults - recommended [Info](#)
5 AM UTC, starts within 8 hours.

Customize backup window

Transition to cold storage [Info](#)

Never

Retention period [Info](#)

Always

Copy to destination [Info](#)

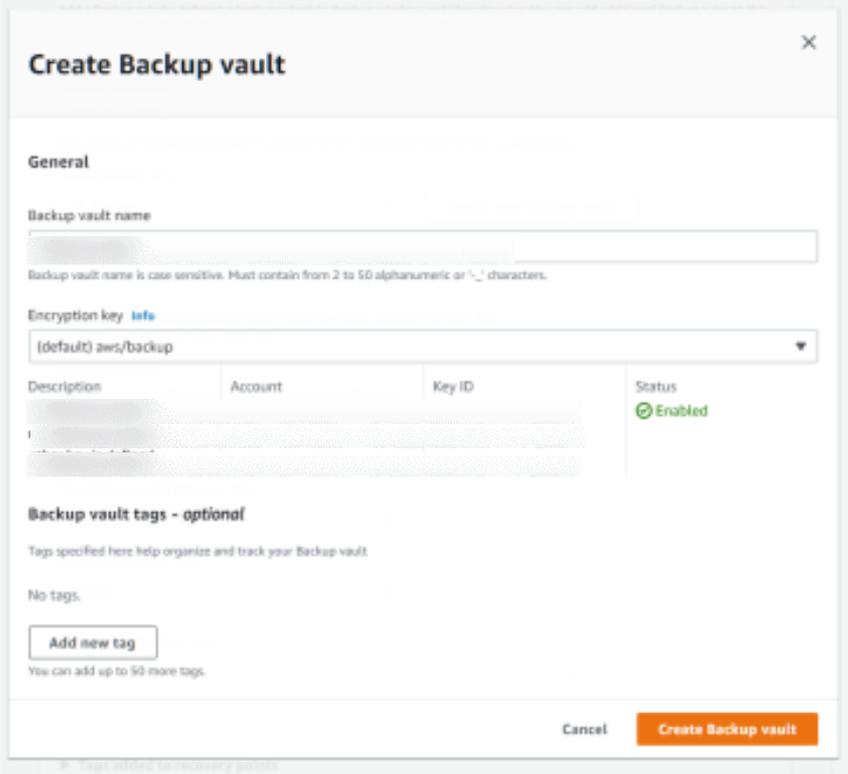
Choose a Region

► **Tags added to recovery points**

AWS Backup copies tags from the protected resource to the recovery point upon creation. You can specify additional tags to add to the recovery point.

The rules define how often backups are generated, what time they are done, how long the data is retained, and where it is stored. With cold storage we can also choose whether our data is going to be stored long-term and choose when our data is moved to this type of storage.

Vault refers to the space that AWS Backup gives us to store our backups, we can use the "Default Vault" or create our own and organize them as we please. We have 2 ways to do it, the first would be to create the vault previously from the Backup Vaults tab and the second is to create it when creating our Backup Plan, for the purposes of this blog we will use the latter.



Create Backup vault

General

Backup vault name

Backup vault name is case sensitive. Must contain from 2 to 50 alphanumeric or '-', '_' characters.

Encryption key [info](#)

Description	Account	Key ID	Status
			Enabled

Backup vault tags - optional

Tags specified here help organize and track your Backup vault.

No tags.

You can add up to 50 more tags.

Once the rules and the Vault where we will store our backups have been defined, we proceed to create the Backup Plan.

Backup rule configuration [Info](#)

Add a Backup rule by defining a backup schedule, backup window, and lifecycle rules. You can add additional Backup rules to this Backup plan later. The backup cost depends on your backup configurations.

Backup rule name

Backup rule name is case sensitive. Must contain from 1 to 50 alphanumeric or '.', '_' characters.

Backup vault [Info](#)

Backup frequency [Info](#)
Daily

Enable continuous backups for point-in-time recovery (PITR) [Info](#)
Available for RDS and S3 resources.

Backup window
 Use backup window defaults - recommended [Info](#)
5 AM UTC, starts within 8 hours.
 Customize backup window

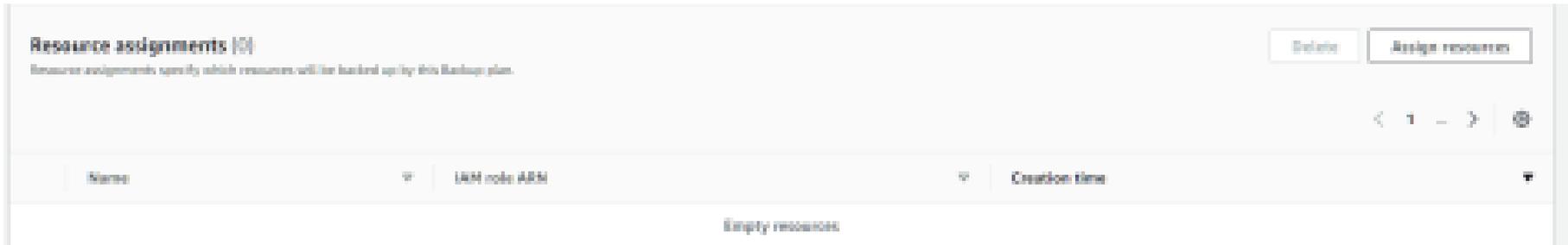
Transition to cold storage [Info](#)
Never

Retention period [Info](#)
Days 1

Copy to destination [Info](#)

► **Tags added to recovery points**
AWS Backup copies tags from the protected resource to the recovery point upon creation. You can specify additional tags to add to the recovery point.

After creating our Backup plan we will assign the corresponding resources so that you can work with them.



Resource assignments (0)

Resource assignments specify which resources will be backed up by this Backup plan.

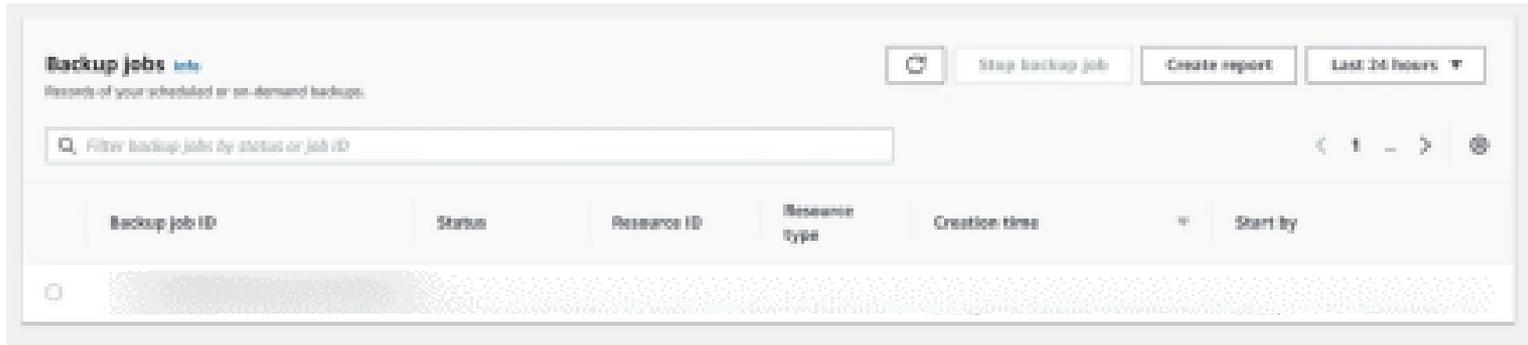
Delete Assign resources

< 1 - > ⚙

Name	IAM role ARN	Creation time
Empty resources		

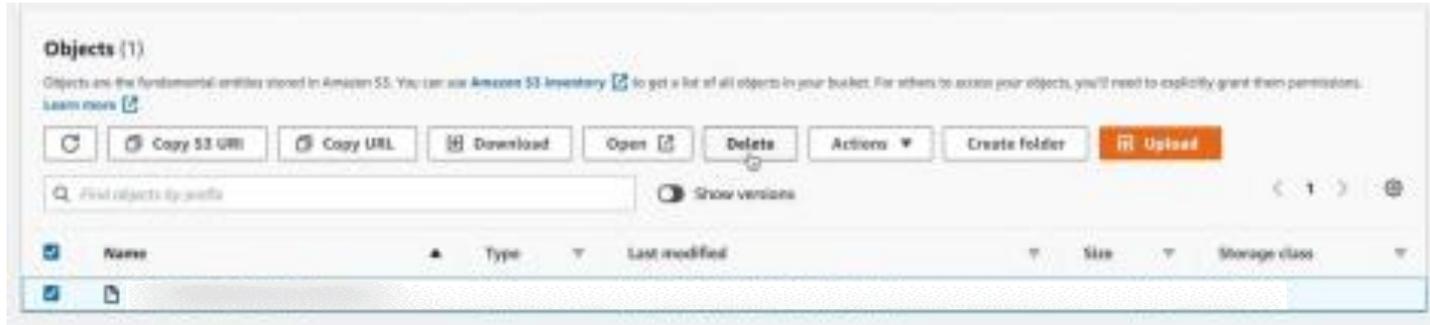
We select our bucket and wait for the first job to be created automatically (we can also configure ourselves the time we want it to be done)

We can monitor the effects of this in the Jobs tab, which once completed would look something like this (keep in mind that depending on the service, the creation process could take more or less time):

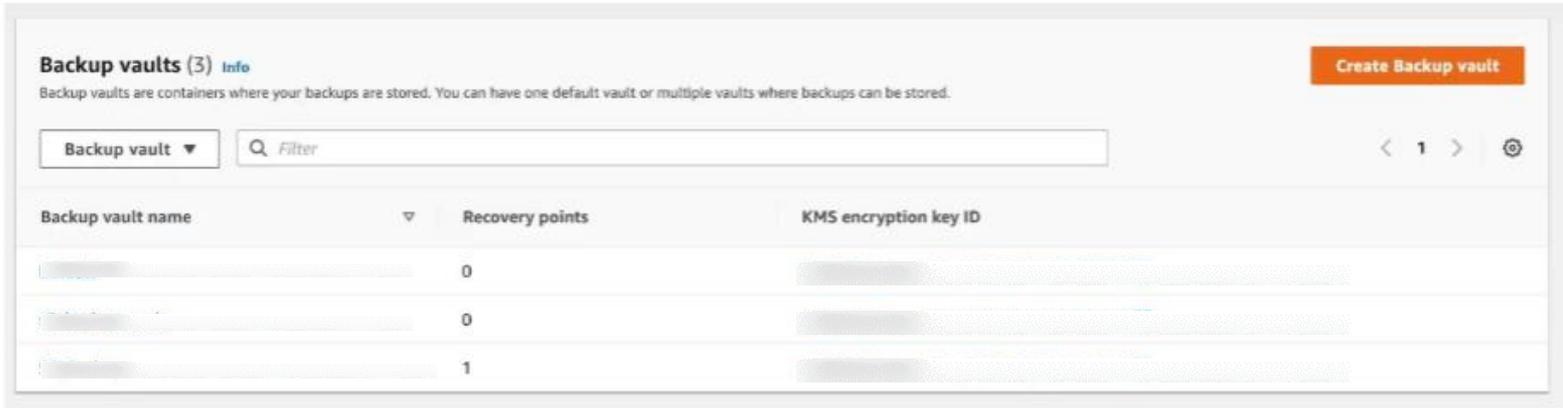


Now we have our backup created.

For the purposes of this example we will simulate the loss of an object in our S3 bucket (To edit the access rules see “*”)



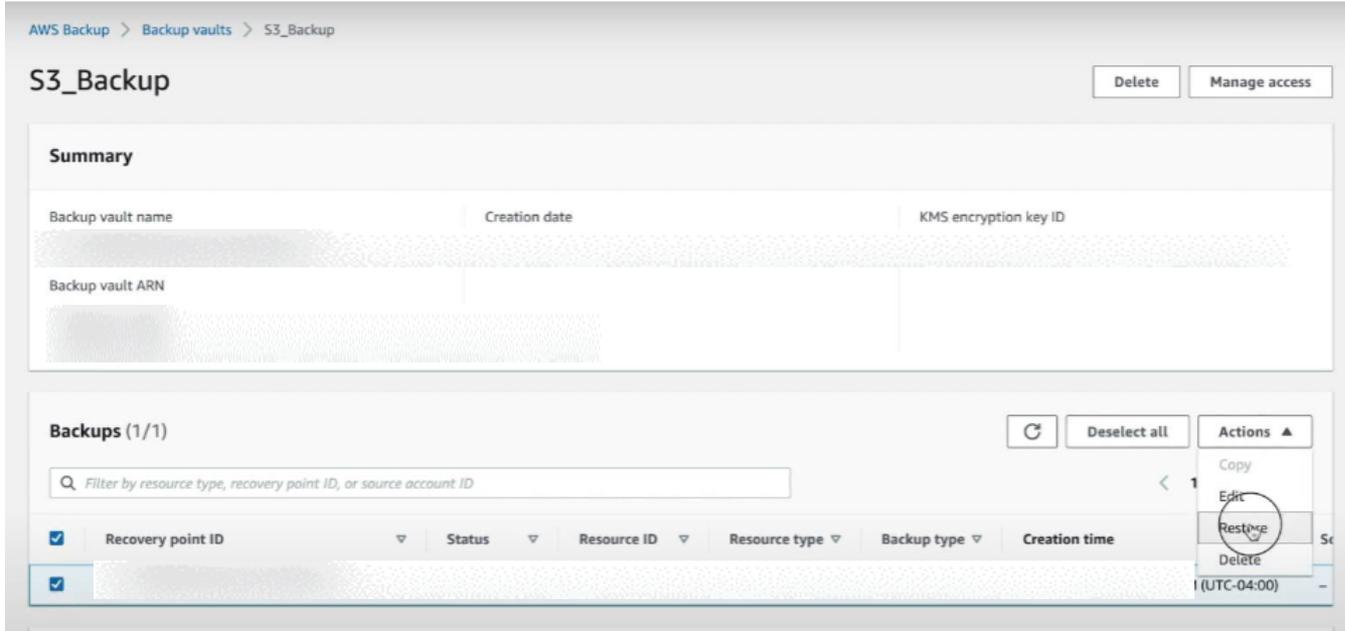
Now that there is nothing in our bucket we will try to restore the object in question using the recovery point from our Backup Vault created by our Backup plan, we select the Vault that we have created (S3_backup)



The screenshot shows the AWS Backup console interface. At the top, it says "Backup vaults (3) Info" with a "Create Backup vault" button. Below this is a description: "Backup vaults are containers where your backups are stored. You can have one default vault or multiple vaults where backups can be stored." There is a search filter box and a table with three columns: "Backup vault name", "Recovery points", and "KMS encryption key ID". The table contains three rows of data, with the third row showing 1 recovery point.

Backup vault name	Recovery points	KMS encryption key ID
[Redacted]	0	[Redacted]
[Redacted]	0	[Redacted]
[Redacted]	1	[Redacted]

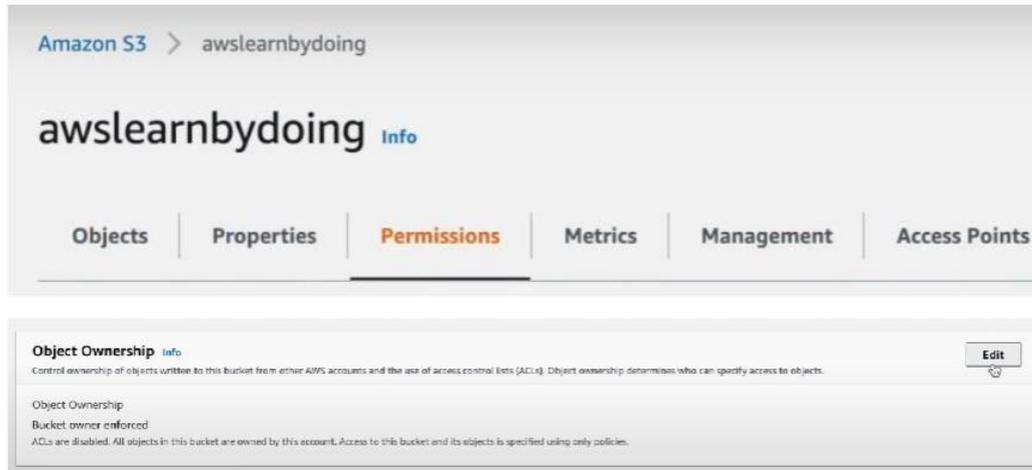
From here we choose the point to which we want to return where our object was still inside the bucket, once selected we display the actions button and press "Restore"



The screenshot shows the AWS Backup console interface. At the top, the breadcrumb navigation reads "AWS Backup > Backup vaults > S3_Backup". The main heading is "S3_Backup" with "Delete" and "Manage access" buttons to its right. Below this is a "Summary" section with a table containing fields for "Backup vault name", "Creation date", "KMS encryption key ID", and "Backup vault ARN". The "Backups (1/1)" section features a search filter, a refresh button, a "Deselect all" button, and an "Actions" dropdown menu. The dropdown menu is open, showing options for "Copy", "Edit", "Restore", and "Delete". The "Restore" option is circled in red. Below the menu is a table with columns for "Recovery point ID", "Status", "Resource ID", "Resource type", "Backup type", and "Creation time". A single backup entry is visible, with its "Recovery point ID" selected by a checkbox.

It is very important to remember that for this process to take effect in our S3 bucket we must grant the necessary permissions to both the same bucket and the Backup plan, so we must enable the ACLs and thus allow them to interact with each other.

For this we go to the permissions tab of our bucket and go down to the object ownership section (Object Ownership) and press edit:



The screenshot shows the AWS S3 console interface for the bucket 'awslearnbydoing'. The breadcrumb navigation at the top reads 'Amazon S3 > awslearnbydoing'. Below this, the bucket name 'awslearnbydoing' is displayed with an 'Info' link. A horizontal menu contains several tabs: 'Objects', 'Properties', 'Permissions' (which is highlighted with an orange underline), 'Metrics', 'Management', and 'Access Points'. Below the menu, the 'Object Ownership' section is visible, featuring an 'Info' link and an 'Edit' button. The text in this section reads: 'Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects. Object Ownership: Bucket owner enforced. ACLs are disabled. All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.'

We enable the ACLs and save

Edit Object Ownership [Info](#)

Object Ownership
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

 **Enabling ACLs turns off the bucket owner enforced setting for Object Ownership**
Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket policy.

I acknowledge that ACLs will be restored.

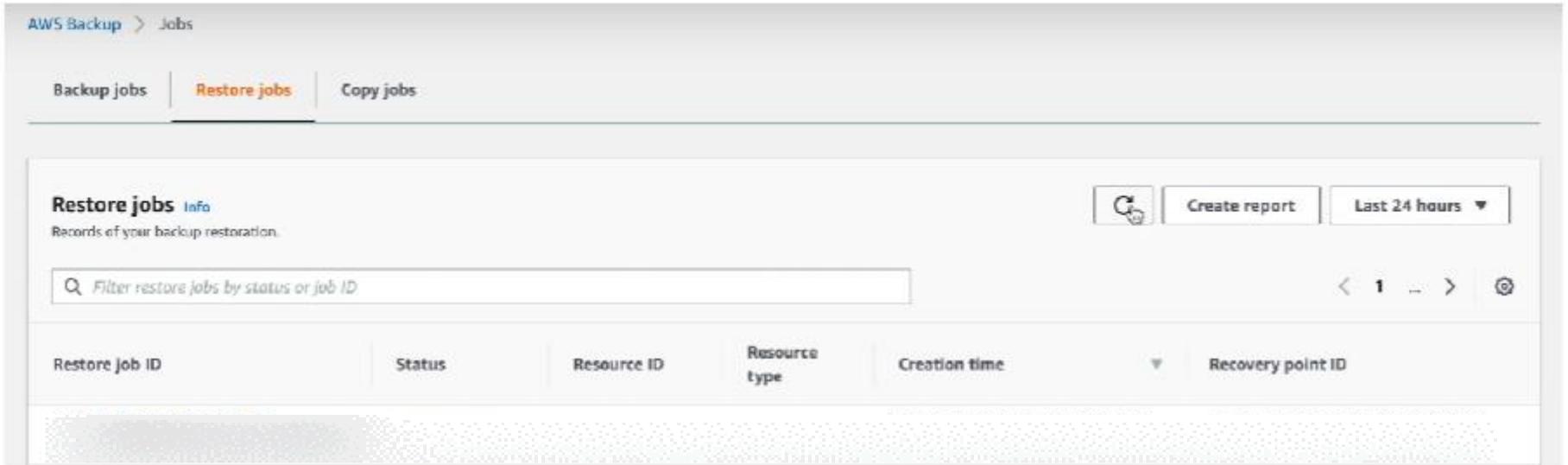
Object Ownership

Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

Object writer
The object writer remains the object owner.

 If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control ACL is required for object uploads. [Learn more](#) 

To confirm that our restoration has been successful we go to the Restore Jobs tab under the Jobs section in AWS Backup



The screenshot shows the AWS Backup console interface. At the top, there is a breadcrumb trail: "AWS Backup > Jobs". Below this, there are three tabs: "Backup jobs", "Restore jobs" (which is selected and highlighted in orange), and "Copy jobs". The main content area is titled "Restore jobs" with a blue "Info" link. Below the title, it says "Records of your backup restoration." To the right of this text are three buttons: a refresh icon, "Create report", and "Last 24 hours" with a dropdown arrow. Below these buttons is a search bar with the placeholder text "Filter restore jobs by status or job ID". At the bottom of the screenshot, the top of a table is visible, with columns for "Restore Job ID", "Status", "Resource ID", "Resource type", "Creation time", and "Recovery point ID".

Keep in mind that when doing a restore you can not only go back to a previous state of your resource or service, but you can also change configurations as needed, such as instances for a database or change the availability zone, even change of region on certain occasions (review documentation according to the service to confirm availability). And that's how we fully automatically backed up a service and learned how to restore it to a previous point at our convenience.

Try it for yourself and see all the features that AWS Backup has for you and your business!



Morris & Opazo

Expertos en Tecnología de la Nube

We Love the Cloud

aws

PARTNER
Advanced Tier
Services

- Public Sector
- Immersion Day
- Solution Provider
- AWS Lambda Delivery
- Amazon Kinesis Delivery
- Amazon Redshift Delivery
- Amazon API Gateway Delivery
- Data & Analytics Services Competency
- Amazon EC2 for Windows Server Delivery

Morrisopazo.com

contacto@Morrisopazo.com