

# Zerto

Deploy & Configure  
Zerto Long-Term  
Retention for  
Amazon S3

---



# Table of Contents

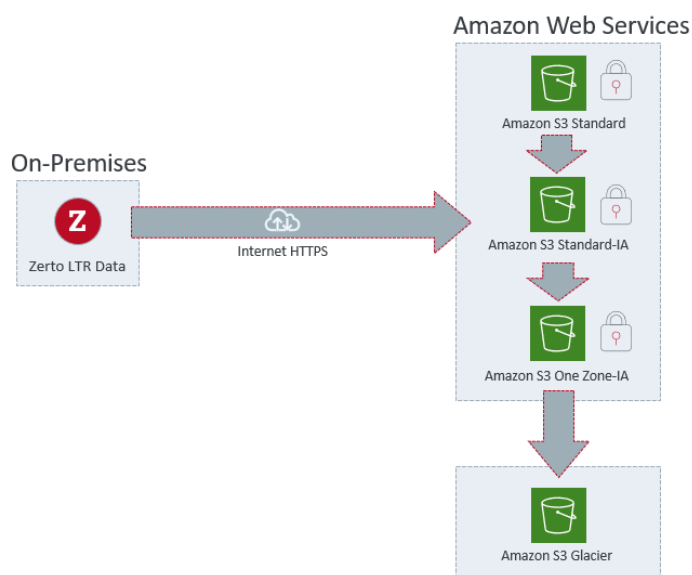
ABOUT ZERTO 9.0 BACKUP TO AMAZON S3.....	2
PRE-REQUISITES .....	3
AWS Pre-Requisites.....	3
On-Premises Pre-Requisites.....	3
REQUIREMENTS IN AWS SUBSCRIPTION .....	3
WORKFLOW .....	4
SETUP AMAZON S3 BUCKET, IAM POLICY, AND IAM USER .....	5
Create the AMAZON S3 Bucket.....	5
Create the IAM Policy.....	8
Create the IAM User and Attach to Policy .....	11
CREATE THE AMAZON S3 REPOSITORY IN ZERTO .....	15
NEXT STEPS.....	17
ADDITIONAL TIPS .....	17
Use JSON for the IAM Policy Creation.....	18
Re-Creating an Access Key ID to Obtain the Secret Key.....	18

## About Zerto 9.0 Backup to AMAZON S3

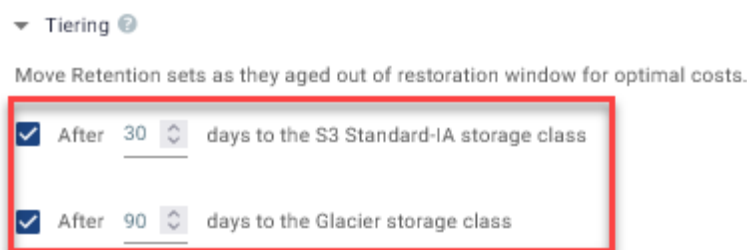
Zerto 8.5 introduced the ability to use Amazon S3 as a direct target for long-term retention via HTTPs with encryption at rest.

Zerto 9.0 delivers additional enhancements that allow automatic tiering of backup data to take advantage of more infrequently accessed storage types and increase cost efficiency as data ages.

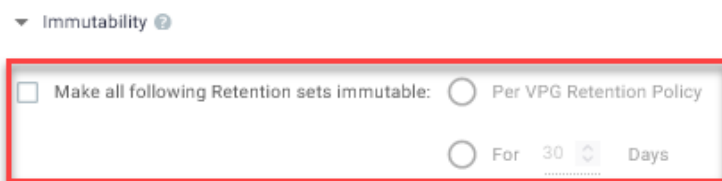
In addition to the auto-tiering capabilities, retention set immutability via Amazon S3 Object Lock has also been added to provide additional protection from data overwrites for a period of time set by the user. This new feature provides additional security to protect backup data from being overwritten or destroyed as in the case of ransomware. This setting is disabled by default and can be set per virtual protection group (VPG) or at the repository level.



Enabled by default (configurable by user), Zerto 9.0 will now automatically move retention sets based on age as portrayed below.



Immutability is disabled by default, however, for maximum protection from accidental data overwrites, deletion, or ransomware manipulation, enabling immutability on retention sets is recommended. This can be performed either at the VPG level or at the repository level.



**Note:** For repositories created prior to Zerto 9.0, the new feature will appear after upgrading, however, will not be enabled by default.

Setting up an Amazon S3 LTR repository is similar to any other repository on-premises or in the cloud, and this document will walk you through that procedure. For more information about Zerto, visit <https://www.zerto.com>.

Please refer to the AWS documentation to understand the costs associated with the use of the Simple Storage Service (S3), which may involve ingress, egress, and storage consumption based on GB/month. Please also note that for any questions or issues with the AWS management console, S3, or IAM, you may need to contact AWS support for assistance.

## Pre-Requisites

This guide assumes you are familiar with deploying solutions within the AWS management console as well as backup repositories in Zerto. For long-term retention considerations and known issues, see the **Zerto 9.0 (or later)** release notes as well as the online help by visiting <https://www.zerto.com/myzerto/>.

This document only refers to setup and configuration in a vSphere environment. If you are using Hyper-V, the steps are the same, as all work is in AWS and Zerto.

### AWS Pre-Requisites

- Minimum 10Mb/s bandwidth between your on-premises datacenter and AWS.
- Designated AWS region where the AMAZON S3 bucket will be deployed. It is recommended that the region chosen is closest to your on-premises datacenter.
- Understanding of AMAZON S3 pricing can be found here: <https://aws.amazon.com/s3/pricing/>.

### On-Premises Pre-Requisites

- In the on-premises Zerto Virtual Manager, you will need to be an administrator to perform the steps within this document.

## Requirements in AWS Subscription

The following requirements are needed in your designated AWS region:

- Permissions to create and manage S3 buckets
- Permissions to create and manage IAM policies and assign to services and resources
- Permissions to create and manage IAM users
- Permissions to attach/detach IAM policies to users

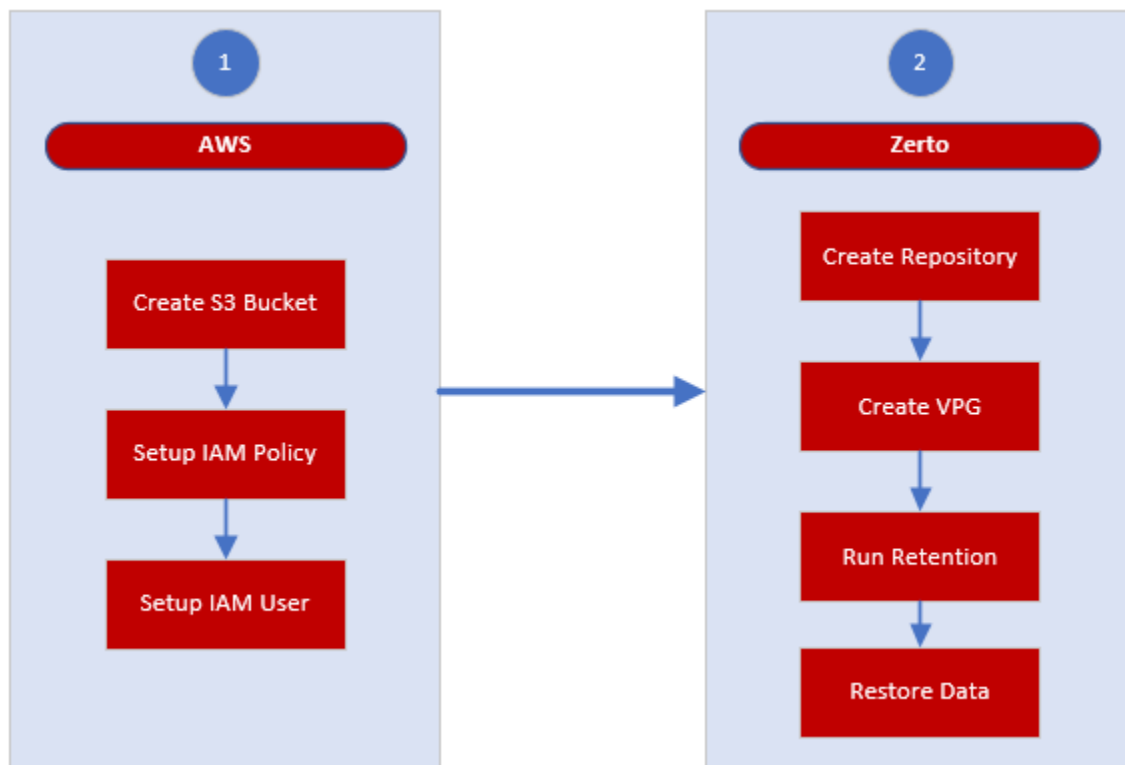
For information about AWS Identity and Access Management, see:

<https://docs.aws.amazon.com/iam/index.html>

For information about AMAZON S3, see: <https://docs.aws.amazon.com/s3/index.html>

## Workflow

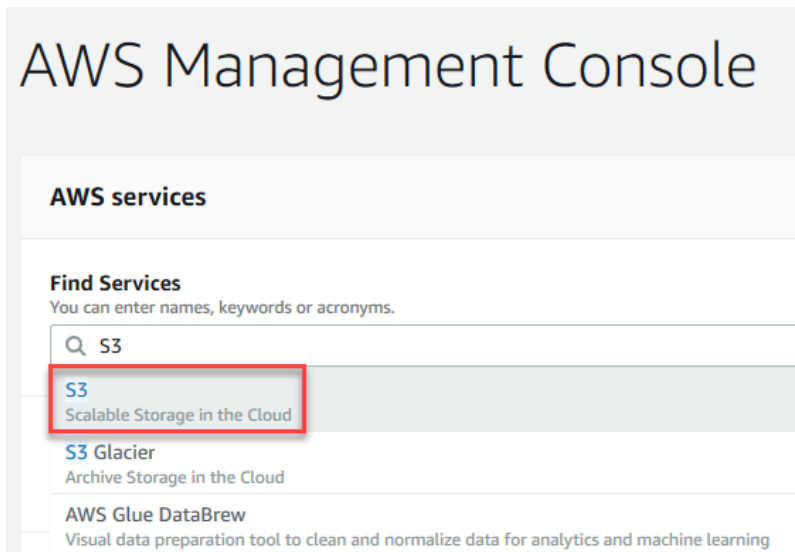
At a high-level, the steps in this document will walk you through the following workflow:



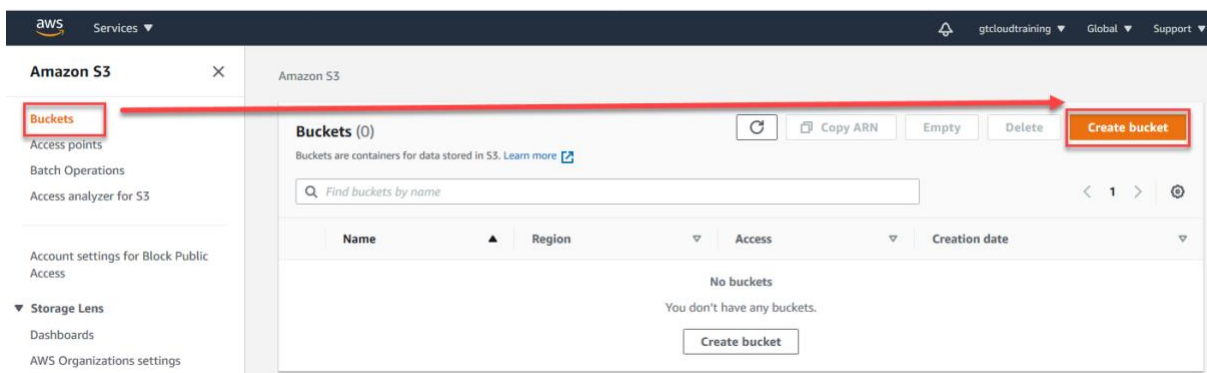
## Setup AMAZON S3 Bucket, IAM Policy, and IAM User

### Create the AMAZON S3 Bucket

1. Log onto AWS the AWS Management Console

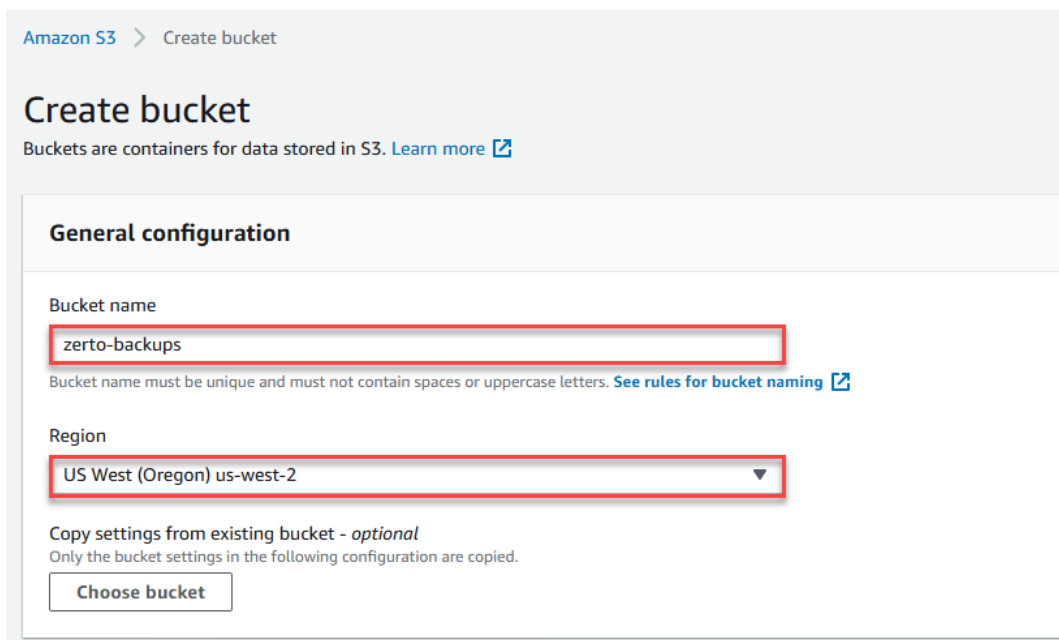


2. Under AWS Services, search for **S3** and select it to access the Amazon S3 service.
3. Click the **Create bucket** button.



4. Provide a **bucket name**.

5. Select a **region** close to your on-premises datacenter.



Amazon S3 > Create bucket

## Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

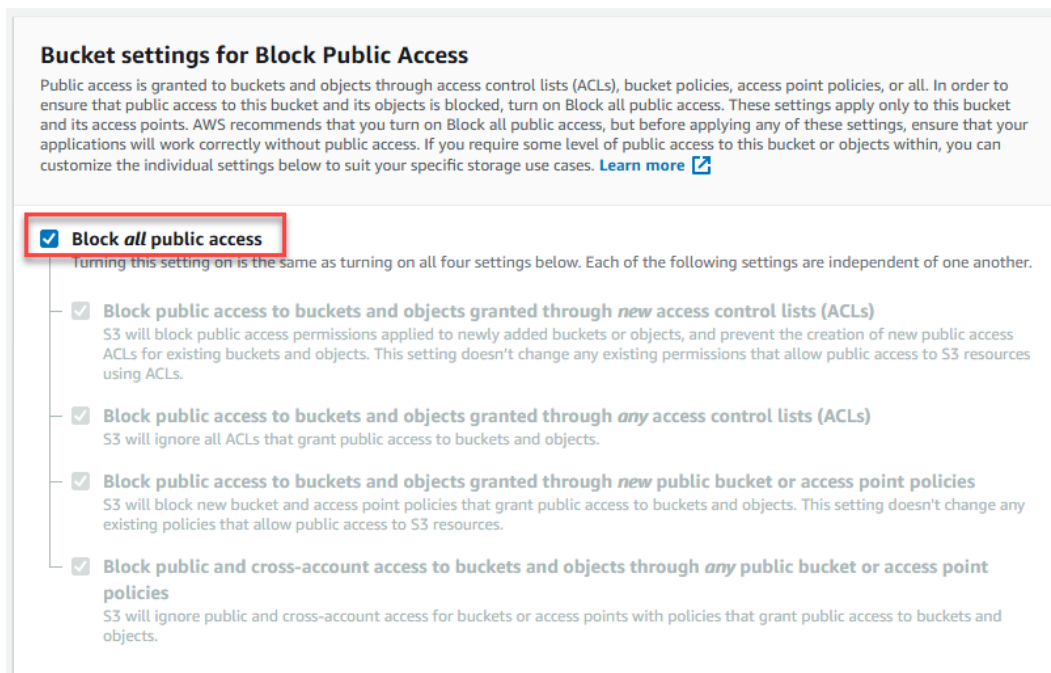
### General configuration

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

Region  
US West (Oregon) us-west-2

Copy settings from existing bucket - *optional*  
Only the bucket settings in the following configuration are copied.

6. Under bucket settings for Block Public Access, leave the default of **Block all public access**.



### Bucket settings for Block Public Access

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through *new* access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through *any* access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through *new* public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

7. **IMPORTANT:** If you are going to be enabling immutability on Amazon S3 retention sets protected by Zerto, **enable** bucket versioning. If not, leave as default (disabled).

**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

Disable

Enable

8. **Optional:** Add a tag.

**Tags (0) - optional**

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

9. Under Default encryption, enable Server-side encryption, and select **Amazon S3 key (SSE-E3)**. By using this encryption key type, costs for encryption services with S3 buckets can be greatly reduced as opposed to the AWS Key Management Service Key (SSE-KMS) option. See “Amazon S3 Example” in the following documentation for more information: <https://aws.amazon.com/kms/pricing/>.

**Default encryption**

Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption

Disable

Enable

Encryption key type

To upload an object with a customer-provided encryption key (SSE-C), use the AWS CLI, AWS SDK, or Amazon S3 REST API.

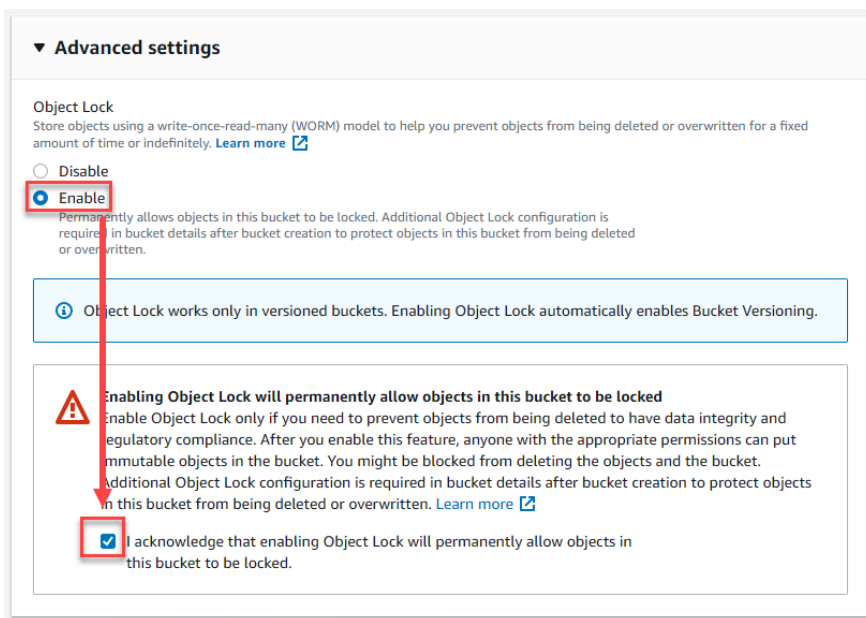
**Amazon S3 key (SSE-S3)**  
An encryption key that Amazon S3 creates, manages, and uses for you. [Learn more](#)

AWS Key Management Service key (SSE-KMS)  
An encryption key protected by AWS Key Management Service (AWS KMS). [Learn more](#)

Reduces cost of encryption by up to 99%

10. **IMPORTANT:** If you are going to be enabling immutability do not skip this step, because Object Lock needs to be enabled prior to creating the bucket. Expand the **Advanced Settings**.

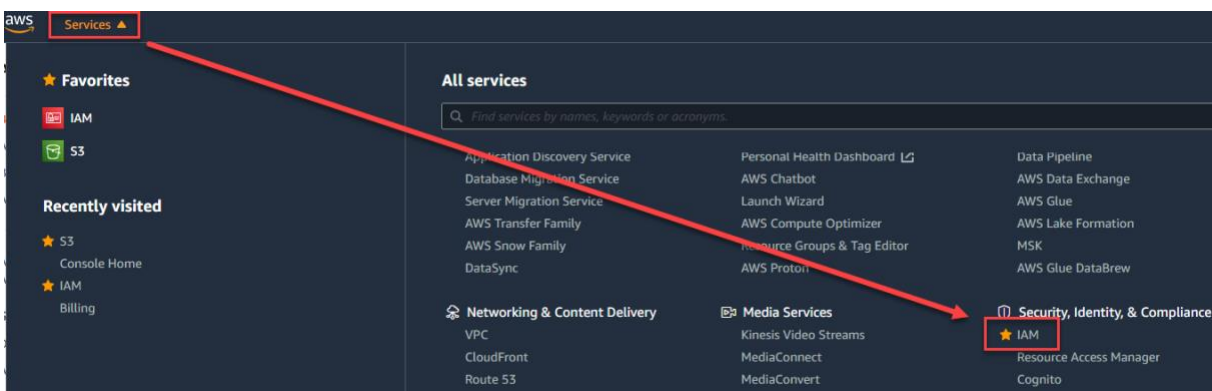
11. Select **Enable** for Object Lock, and check the box to acknowledge that enabling Object Lock will permanently allow objects in this bucket to be locked.



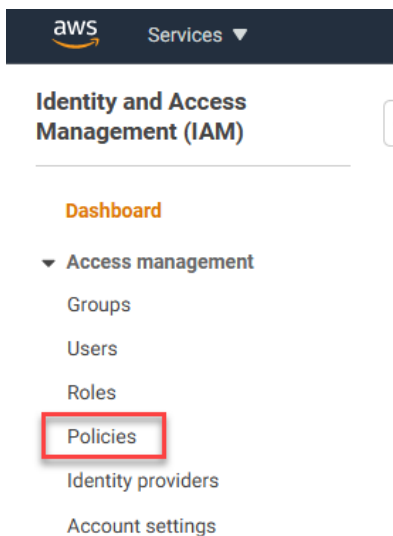
12. Click **Create** bucket.

### Create the IAM Policy

13. Click **Services**, then on the right, under Security, Identity, & Compliance, select **IAM**.



14. On the left navigation bar, click **Policies**.



15. Click **Create Policy**.
16. For Service, select **S3**.
17. Under Actions, select the following:
  - a. Access Level → List
    - i. ListAllMyBuckets
    - ii. ListBucket
    - iii. ListBucketVersions
  - b. Access Level → Read
    - i. GetBucketObjectLockConfiguration
    - ii. GetObject
    - iii. GetObjectACL
    - iv. GetObjectVersion
  - c. Access Level → Write
    - i. DeleteObject
    - ii. DeleteObjectVersion
    - iii. PutObject
    - iv. PutObjectRetention
    - v. RestoreObject
  - d. Access Level → Permissions management
    - i. PutObjectAcl
    - ii. DeleteBucketPolicy
18. Under Resources → bucket, click Add ARN.

▼ Resources  Specific  All resources  
[close](#)

**bucket** ⓘ Specify **bucket** resource ARN for the **DeleteBucketPolicy** and 1 more action. ⓘ  Any  
[Add ARN](#) to restrict access

**object** ⓘ Specify **object** resource ARN for the **PutObject** and 4 more actions. ⓘ  Any  
[Add ARN](#) to restrict access

19. In the **Bucket name** field, type the name of the S3 bucket you just created. You will see the ARN above auto-populate as you are typing. After you have typed the name of the bucket, click **Add**.

Add ARN(s) ×

Amazon Resource Names (ARNs) uniquely identify AWS resources. Resources are unique to each service. [Learn more](#) ↗

Specify ARN for bucket List ARNs manually

arn:aws:s3:::zerto-backups

Bucket name \*   Any

[Cancel](#) [Add](#)

20. Under **Resources** → **object**, click to enable the checkbox for **Any**.

▼ Resources  Specific  All resources  
[close](#)

**bucket** ⓘ  [EDIT](#) +  Any  
[Add ARN](#) to restrict access

**object** ⓘ Any resource of type = object  Any

21. Your configuration should look like the image below. When done, click **Review policy**.

▼ S3 (13 actions) ⚠ 2 warnings Clone Remove

► Service S3

► Actions **List**

- ListBucket
- ListBucketVersions

**Read**

- GetBucketObjectLockConfiguration
- GetObject
- GetObjectAcl
- GetObjectVersion

**Write**

- DeleteObject
- DeleteObjectVersion
- PutObject
- PutObjectRetention
- RestoreObject

**Permissions management**

- DeleteBucketPolicy
- PutObjectAcl

► Resources Specify object resource ARN for the PutObjectRetention and 8 more actions.

- arn:aws:s3::zerto-backups
- One or more actions may not support this resource.
- arn:aws:s3::/\*

► Request conditions Specify request conditions (optional)

22. Provide a name and description for your policy, then click **Create policy**.

## Create policy

1 2

### Review policy

Name\*

Use alphanumeric and '+, @, \_' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+, @, \_' characters.

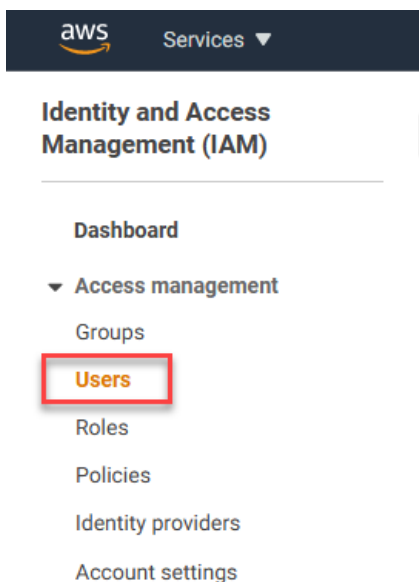
#### Summary

Service	Access level	Resource	Request condition
Allow (1 of 255 services) <a href="#">Show remaining 254</a>			
S3	Limited: List, Read, Write, Permissions management	Multiple	None

## Create the IAM User and Attach to Policy

Now, we will create the IAM user for Zerto backups and attach it to the policy you have just created. As part of this, you will also end up with an access key ID and secret access key.

23. On the left navigation, click on **Users**.



24. Click **Add user**.

25. Provide a **Username** and select **Programmatic access** as the **Access type**. By selecting Programmatic access, you will be provided an **access key ID** and a **secret access key** for the AWS API, which is required when configuring the Zerto repository.

## Add user



### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[+ Add another user](#)

### Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type\*  **Programmatic access**  
 Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- AWS Management Console access**  
 Enables a **password** that allows users to sign-in to the AWS Management Console.

26. Click **Next: Permissions**.

27. Under Set permissions, select **Attach existing policies directly**.

## Add user



### Set permissions

Add user to group

Copy permissions from existing user

Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

- In the search box beside **Filter policies**, type the name of the policy you created in the previous section, and when the result is returned, **check** the box beside the policy.

## Add user



### Set permissions

Add user to group

Copy permissions from existing user

Attach existing policies directly

Create policy



Filter policies		zerto-backup-policy	Showing 1 result	
	Policy name	Type	Used as	
<input checked="" type="checkbox"/>	zerto-backup-policy	Customer managed	None	

- Click **Next: Tags**.
- If you need to add a tag for billing and/or identification purposes, go ahead and add it now, then click **Next: Review**.

## Add user



### Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	

You can add 50 more tags.

31. Review user details, permissions boundary, and tags to verify settings, then click **Create user**.

## Add user



### Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details	
<b>User name</b>	zerto-backup-user
<b>AWS access type</b>	Programmatic access - with an access key
<b>Permissions boundary</b>	Permissions boundary is not set

Permissions summary	
The following policies will be attached to the user shown above.	
Type	Name
Managed policy	<a href="#">zerto-backup-policy</a>

Tags
No tags were added.

[Cancel](#)

[Previous](#)

[Create user](#)

32. **IMPORTANT:** When you see the success screen, click **Download .csv** to save a copy of the **Access key ID** and **Secret access key**. You will need it when configuring the repository in Zerto.

If you do not collect the file now, or copy the Secret Key, see the “**Additional Tips**” section at the end of this document to see how you can re-create an access key, which will provide you with a new secret key without having to create a new user.

## Add user



✔ **Success**

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://gtcloudtraining2019.signin.aws.amazon.com/console>

**Download .csv** ← **IMPORTANT!**

User	Access key ID	Secret access key
zerto-backup-user	AKIA[REDACTED]7W	***** Show

## Create the AMAZON S3 Repository in Zerto

Now that we have completed configuration of the AMAZON S3 bucket, IAM user and policy, we can create the repository in Zerto.

1. Log in to the Zerto UI (<https://yourServerIP:9669>).
2. Once logged in, click **Setup** → **Repositories** → + New Repository.

The screenshot shows the Zerto web interface. On the left is a navigation sidebar with 'Setup' highlighted. The main content area shows 'REPOSITORIES' with 'No Repositories' and a '+ New Repository' button. A red arrow points from the 'Setup' button in the sidebar to the 'REPOSITORIES' section, and another red arrow points from the 'REPOSITORIES' section to the '+ New Repository' button.

3. In the New Repository window, provide the following:
  - a. General:
    - i. Enter a name for the repository.
    - ii. Select **Amazon S3** as the storage type.
    - iii. **Optional:** Set as default repository.

▼ General ⓘ

Name

Storage Type

Connection Type

Set as default repository

b. Settings:

- i. Select the **Region** you created the S3 bucket in.
- ii. **Optional:** Enter the Endpoint URL (only use this if you **are not** selecting a region. If you selected a region, this field becomes optional. (See [AMAZON S3 Service Endpoints](#) for more information).
- iii. Provide the exact name of the bucket you created.
- iv. Enter the **Access Key ID** (You can retrieve this from the .csv file downloaded when creating the IAM user).
- v. Enter the **Secret Key** (You can retrieve this from the .csv file downloaded when creating the IAM user).
- vi. Select the storage class you want to use. (See [AMAZON S3 Storage Classes](#) for more information).

▼ Settings ⓘ

Region

Endpoint URL

Bucket Name

Access Key

Secret Key

Storage Class

c. Immutability:

- i. **Optional:** Enable immutability by enabling the checkbox and configuring your policy.

▼ Immutability ⓘ

Make all following Retention sets immutable:  Per VPG Retention Policy

For  Days

d. Tiering (enabled by default):

- i. Configure tiering as per your requirements.

▼ Tiering ⓘ

Move Retention sets as they aged out of restoration window for optimal costs.

After 30 days to the S3 Standard-IA storage class

After 90 days to the Glacier storage class

4. Click **Save**.
5. You now have a repository for Zerto backups:

	Repository Name	Storage Type	Connection Type	Connectivity	Path	Usage / Capacity (GB)	VPGs
<input checked="" type="checkbox"/>	AWS S3 Repository	Amazon S3	S3	Connected	s3://zerto-backups/	N/A	0

## Next Steps

Now you are ready to configure virtual protection groups (VPGs) to protect your data and enable backups to use this repository.

## Additional Tips

This section will be updated with additional tips & tricks that may help make the process a little easier.

## Use JSON for the IAM Policy Creation

If you are familiar with creating and formatting JSON files for use in AWS, you can save some time when setting up the policy by creating a JSON file that contains the IAM policy requirements. If you are creating one from scratch, you can refer to the image below, otherwise, here is a link to one that has already been created:

<https://github.com/ZertoPublic/Zerto9-LTR-AWS-IAM-JSON>

**Note:** If you decide to use the already-created JSON file from GitHub, be sure to update the **Resource ARN** for your **S3 bucket** (see below).

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "VisualEditor0",
6       "Effect": "Allow",
7       "Action": [
8         "s3:PutObject",
9         "s3:GetObjectAcl",
10        "s3:GetObject",
11        "s3:ListBucketVersions",
12        "s3:PutBucketAcl",
13        "s3:ListBucket",
14        "s3>DeleteObject",
15        "s3>DeleteBucketPolicy"
16      ],
17      "Resource": [
18        "arn:aws:s3:::zerto-backups",
19        "arn:aws:s3::*/*"
20      ]
21    },
22    {
23      "Sid": "VisualEditor1",
24      "Effect": "Allow",
25      "Action": "s3:ListAllMyBuckets",
26      "Resource": "*"
27    }
28  ]

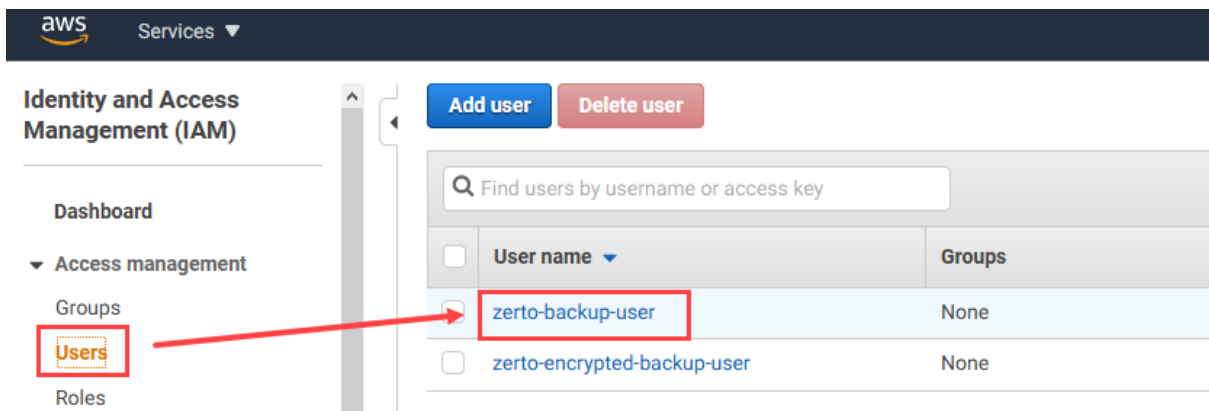
```

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

## Re-creating an Access Key ID to Obtain the Secret Key


If you by chance did not save the .csv file or copy the secret key when you first created the IAM user in section 5.3 above, you can still use the same IAM user, however, you will need to generate a new Access Key ID and get a new Secret Key by following the steps below:

1. In the AWS Management Console, go to **Identity and Access Management (IAM)**.
2. Under Access management, click **Users**, and then in your user list, click on the account you created for the Zerto backups.



3. Click on the **Security credentials** tab.

### Summary

**User ARN**    arn:aws:iam::073459543987:user/zerto-backup-user   
**Path**        /  
**Creation time**    2020-12-09 15:24 PST

**Permissions**    **Groups**    **Tags**    **Security credentials**    **Access Advisor**

▼ Permissions policies (1 policy applied)

[Add permissions](#)


Policy name ▼	Policy type ▼
<b>Attached directly</b>	
▶ <a href="#">zerto-backup-policy</a>	Managed policy

▶ Permissions boundary (not set)

4. Under Access Keys, click **Create access key**.

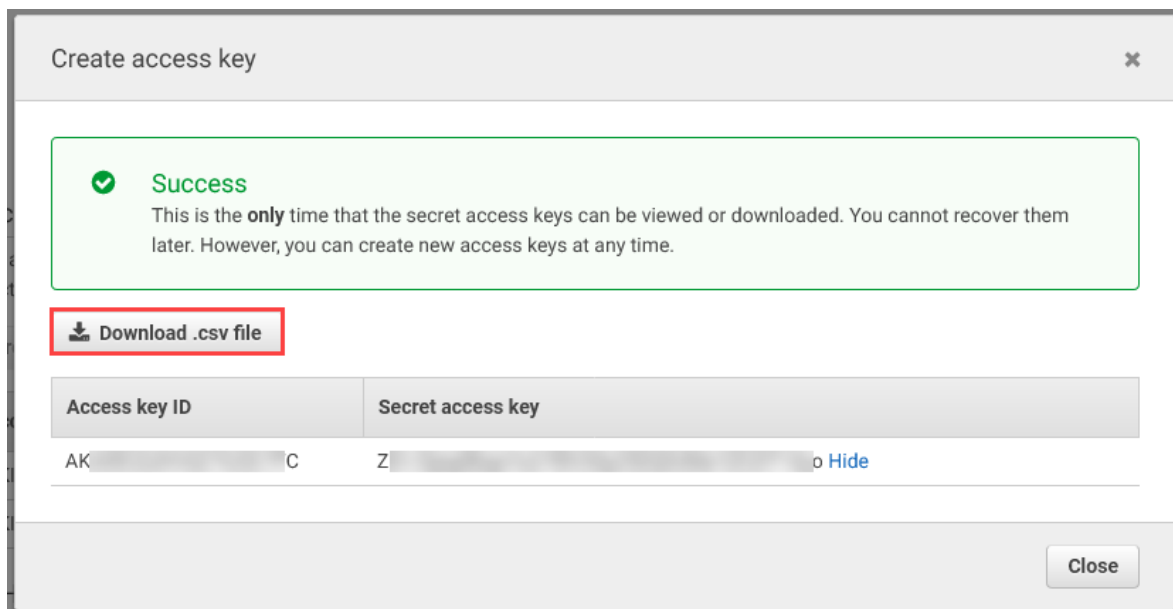
**Access keys**  
 Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. [Learn more](#)

[Create access key](#)

Access key ID	Created	Last used	Status
AKI[REDACTED]	2020-12-09 15:24 PST	2020-12-10 14:08 PST with s3 in us-west-2	Active   <a href="#">Make inactive</a> 

5. A new box will pop up, displaying the new Access Key ID and Secret Key. You will also get another opportunity to download the newly generated .csv file. Do so now and store the .csv file in a safe

place. You can now use this Access Key ID and Secret Key when you create the repository in Zerto.



6. If you are no longer using the original Access Key ID that was created, you can also mark that one as inactive, or even delete it.

## About Zerto

Zerto helps customers accelerate IT transformation through a single, scalable platform for cloud data management and protection. Built for enterprise scale, Zerto's simple, software-only platform uses continuous data protection to converge disaster recovery, backup, and data mobility and eliminate the risks and complexity of modernization and cloud adoption. Zerto enables an always-on customer experience by simplifying the protection, recovery, and mobility of applications and data across private, public, and hybrid clouds. Zerto is trusted by over 9,000 customers globally [www.zerto.com](http://www.zerto.com).