SCALITY

WHITE PAPER

Using S3 object lock

for immutable, ransomware-proof data

Table of contents

Introduction

In this era of information technology, there are a couple of near-certainties for organizations — they will be exposed to some form of cybersecurity threat (viruses, phishing and ransomware) and cybercriminals will try to profit from such attacks.

The rate of cybercrime exploded by staggering amounts during the COVID-19 pandemic, with some figures stating an increase of up to 600%. Luckily, we have backups, right? Well…yes and no.

According to a 2022 ransomware trends report, 76% of surveyed organizations suffered at least one ransomware attack within the past year, and 47% of their data, on average, was encrypted by ransomware. 94% of attacks targeted backup repositories! While 52% of companies with ransomware-encrypted data paid the ransom and were successful in recovery, one in four were unable to recover their data after paying up.

94%

of ransomware attacks attempt to infect backup repositories

47%

of data is encrypted in a ransomware attack

24%

of organizations who pay a ransom still cannot recover their data

Clearly, ransomware is pervasive and inevitable. Even meeting demands for ransom payments doesn't guarantee data recovery. These cold, hard facts should concern every IT leader. So, what can we do about it?

Prevention is always better than cure, but companies have very large attack surfaces these days (think email, phone, social media, websites, etc.) — and the more attack vectors, the higher the chance that something will get through. Of course, we should do everything possible to keep cybercriminals out, but we also have to consider the likelihood of a successful attack — and proactively lay plans for a surefire path to recovery.

Want to be able to recover all of your precious data without paying a ransom? An effective backup and recovery strategy employs the air gap technique — a security countermeasure based on the idea of creating an impenetrable barrier between a digital asset and malicious actors. At any given time, a copy of your organization's data is offline (disconnected) and cannot be accessed. One way to achieve an air gap is to have immutable (undeletable and unmodifiable) copies of your data. The key to data immutability is S3 object locking.

Utilizing S3 object locking on Scality RING or ARTESCA software protects your most valuable asset — your data — so that your backups remain free from corruption even if your backup application is compromised.

While tape is another option, it's, well...tape. Let's just say it's not ideal for everyone or for all dataset sizes.

What is S3 Object Locking?

Object locking adds WORM (Write Once, Read Many) capabilities to Scality's S3 storage offerings. Object locking is built on two key S3 foundational principles:

- **Object immutability:** Unlike traditional files, which are designed to be appended, pre-pended or otherwise edited, objects are designed to be unchangeable.
- **Object versioning:** Uploading an object with the same name will either overwrite the existing object (if no versioning) or create a new object version (if versioning is enabled). Used to preserve, retrieve and restore all versions of objects stored in your buckets, this super-handy feature helps recover objects after accidental deletion or overwrite.

Object locking allows us to protect business-critical data through WORM, as objects in the WORM state cannot be modified or deleted. There are two types of immutability (WORM) policies that can be applied:

- **Time-based retention mode:** Users can set policies to store data unaltered for a specific time period. There are two configurable time-based retention types:

  - **Governance mode:** Users cannot overwrite or delete an object version or alter the lock settings unless they have special permissions. In this mode, we protect the object from *most users*, but a select user can be given permissions to alter the retention settings and delete objects, if necessary. This mode is normally recommended if your organization is still in the process of formulating its data security strategy.

  - **Compliance mode:** No user can overwrite or delete a protected object version, including the root user for the account. This includes shortening the retention period (note: extending the retention period is allowed). If you enable compliance mode, make sure you **really** want that data locked in place for 7 years.

- **Legal hold:** When an object is in legal hold, it is immutable until the legal hold is explicitly cleared from the object. This can be only be done by the privileged user with required permissions.

It should be noted that time-based retention modes and legal hold are not mutually exclusive. For example, you can put a legal hold on an object that is currently in time-based retention mode. Once that retention mode expires, the object will still be protected from deletion or change until the legal hold is lifted.

One other aspect of S3 object locking is that it can be applied at the bucket level and also at the individual object level. This provides multiple ways for applications to utilize object locking. The retention set on a specific object (through the application) will supersede any default policies that exist on a bucket.

Sounds good, but is it really (REALLY) secure?

If it wasn't really (REALLY) secure, it would be a sheer fallacy to use object locking as a key to combating ransomware. But don't just take our word for it, download a copy of the compliance assessment report from Cohasset Associates here.

If you want to save yourself some reading, the document summarizes:

"Accordingly, Cohasset concludes that RING, when properly configured and utilized to retain time-based records, meets the five requirements of SEC Rule 17a-4(f) and FINRA Rule 4511(c), which relate to the recording and non-rewriteable, non-erasable storage of electronic records."

When or why would I use it?

You can utilize object locking for any and all data that you want to keep safe from tampering. This could be your backups or archives. It could range from a single object, file or even an NFT you want to keep secure in one or a few buckets, to the backups of your entire infrastructure.

- Object locking is the right tool for you if:
- Your business has sensitive data to protect
- Your business has a requirement to store in a manner that satisfies compliance regulations in finance, healthcare or other highly regulated sectors

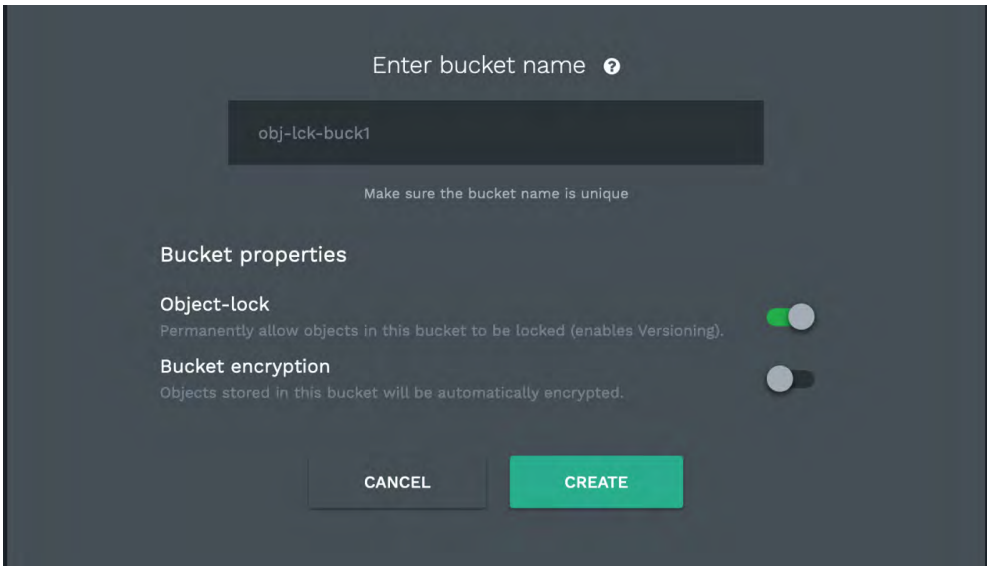
I’m convinced. How do I use it?

Basic object locking is super easy, but there are three rules we need to understand:

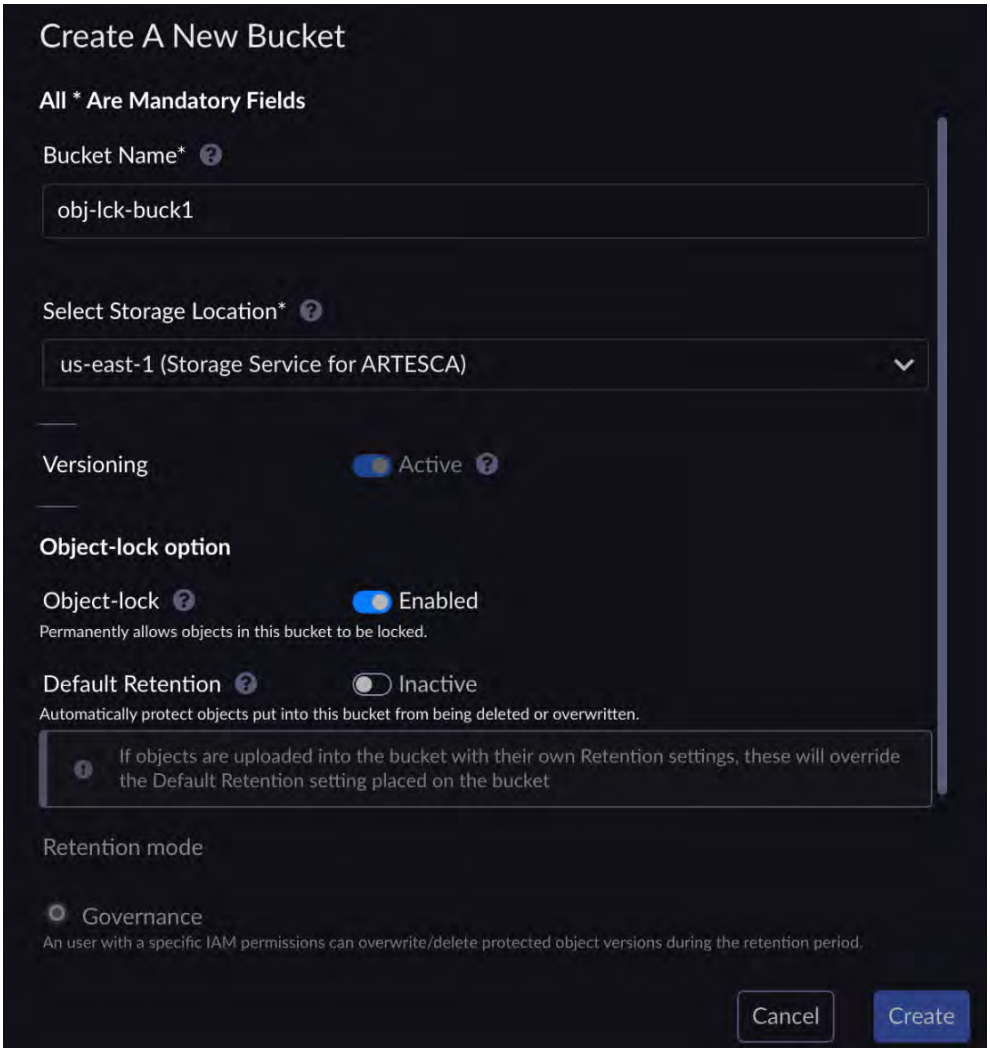
1. You can only set object locking on a bucket at creation time. Object locking needs to be set for every bucket that requires it.
2. When object locking is set on a bucket, versioning is automatically enabled.
3. Once a bucket is created with object locking, you cannot disable object lock or suspend versioning on the bucket.

Bucket creation and object locking configuration can be performed through the S3 APIs. For full details, review the RING and ARTESCA documentation. In this document, we will examine how to configure object locking through the RING and ARTESCA UIs.

Let’s start with RING. We will create a bucket called obj-lck-buck1 and toggle on *Object-lock* under “Bucket properties.”

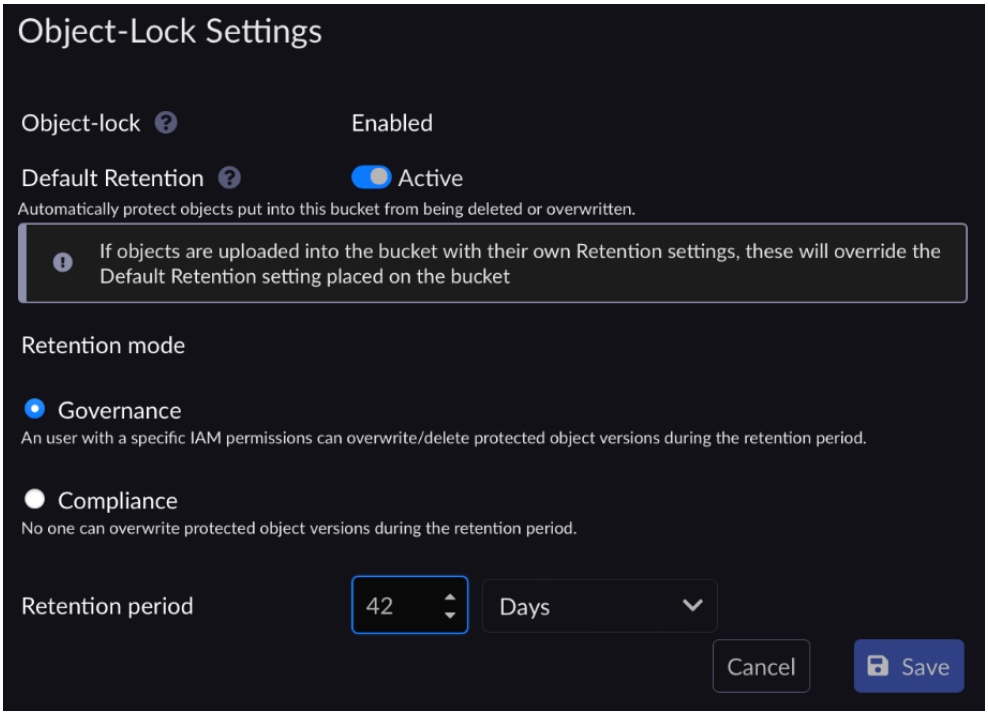


With ARTESCA, we create a bucket called obj-lck-buck1 and toggle on Object-lock under the *"Object-lock"* option. Versioning is simultaneously enabled

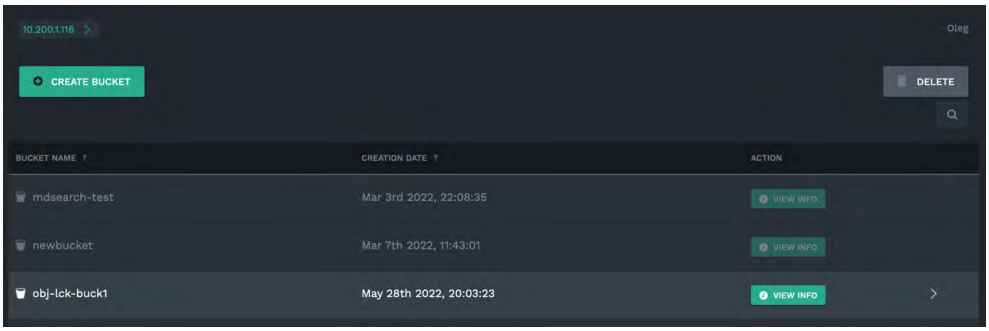


ARTESCA allows you to immediately configure time-based default retention in either governance or compliance mode as well as set a retention duration. In this example, we are going to configure governance mode and retain data for 42 days.

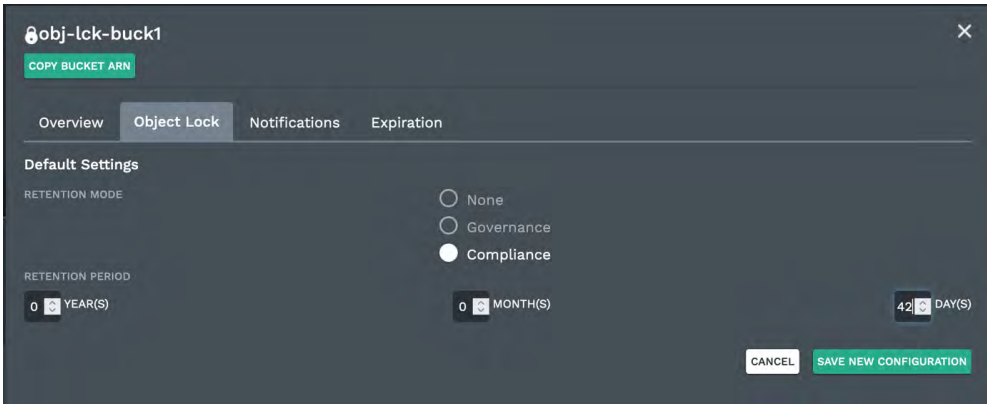
Even if we create a bucket without a default retention, it can always be added through the bucket tab in the UI.



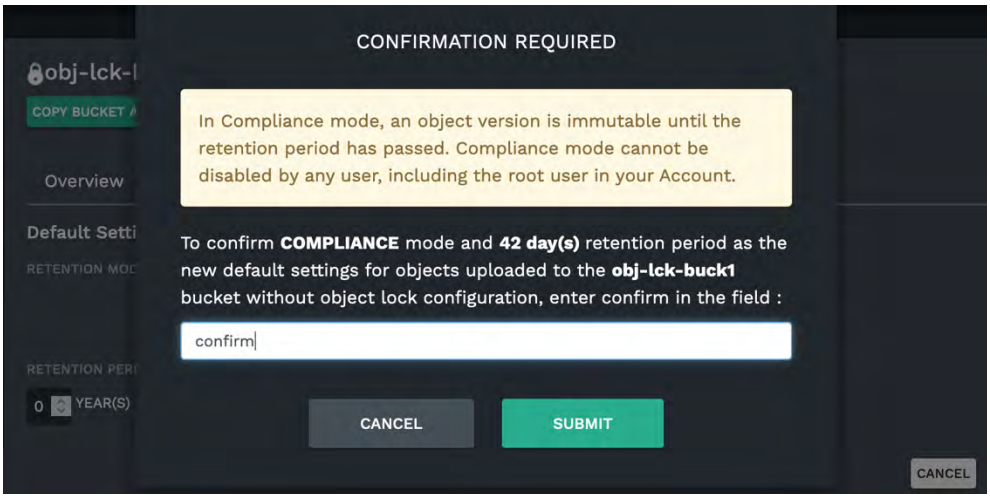
With RING, if we need to configure a default retention, it must be done after the bucket has been created with object locking switched on. To get to bucket options, first we need to “View Info” of the newly created bucket.



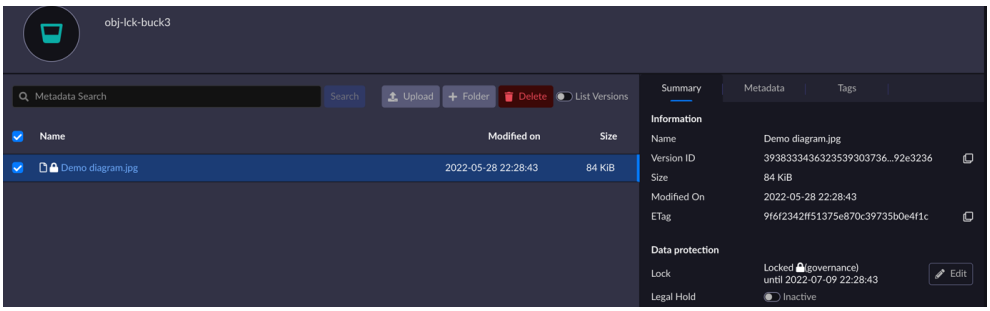
We now have access to the “Object Lock” tab, where we will configure the bucket to have a default retention period of 42 days, in Compliance mode.



To verify that we are indeed happy to enable Compliance mode for this bucket, we will need to enter “confirm” in the pop-up screen.

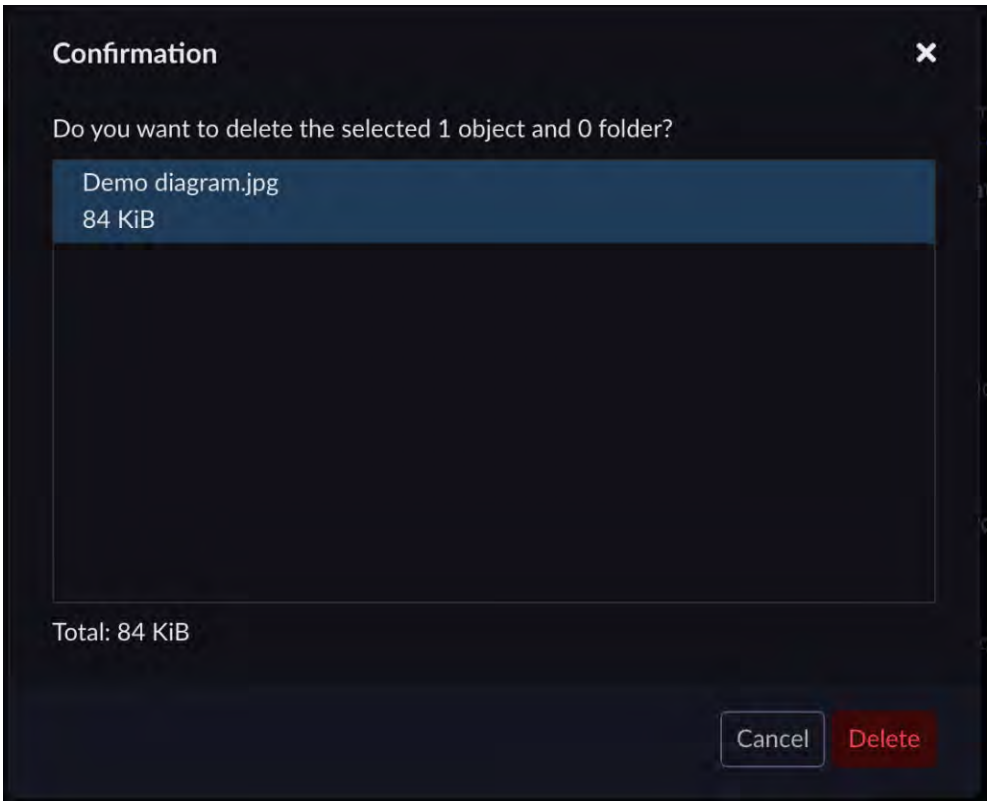


Great! We now have object locking set up. Let’s upload a file and examine the effects of the default time-retention policy we’ve applied.



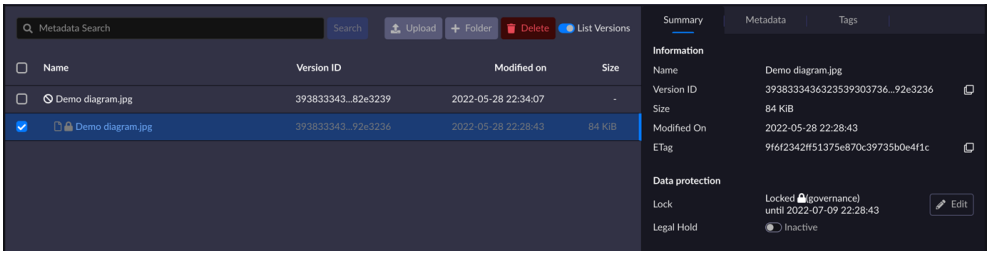
The padlock to the left of the object name denotes the fact that there is a lock in place. In the object summary pane (to the right), we also have the lock mode (Governance), the expiry of the lock, as well as the ability to add legal hold on this object (currently off).

Let's see what happens when we try to delete the object immediately after creation, well before the 42 days' expiry.



After clicking delete, we are faced with a concerningly empty screen — did object locking just fail us?

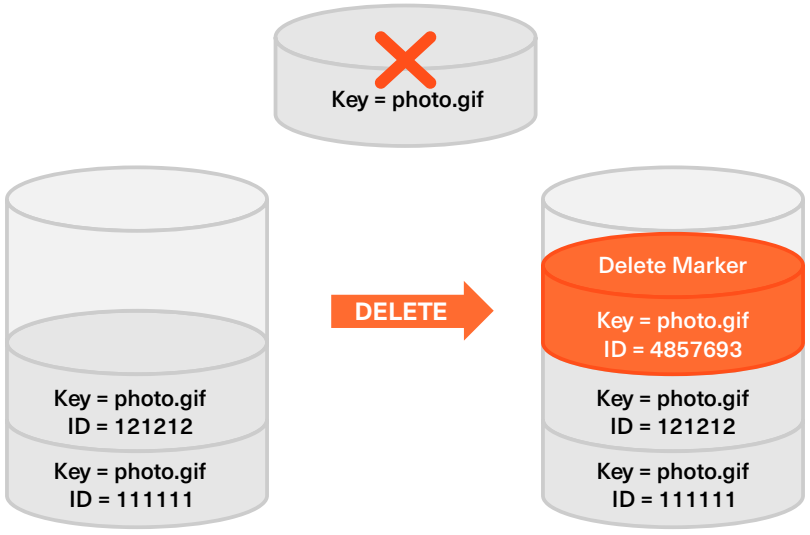
Rest assured, it hasn't. Let's switch on the listing of versioned objects. Here, we see that our object hasn't been deleted. It has simply been listed as a previous version and a delete marker has been added.



What we see here is the effect of object versioning, one of the enabling technologies behind object locking. As per the AWS S3 documentation:

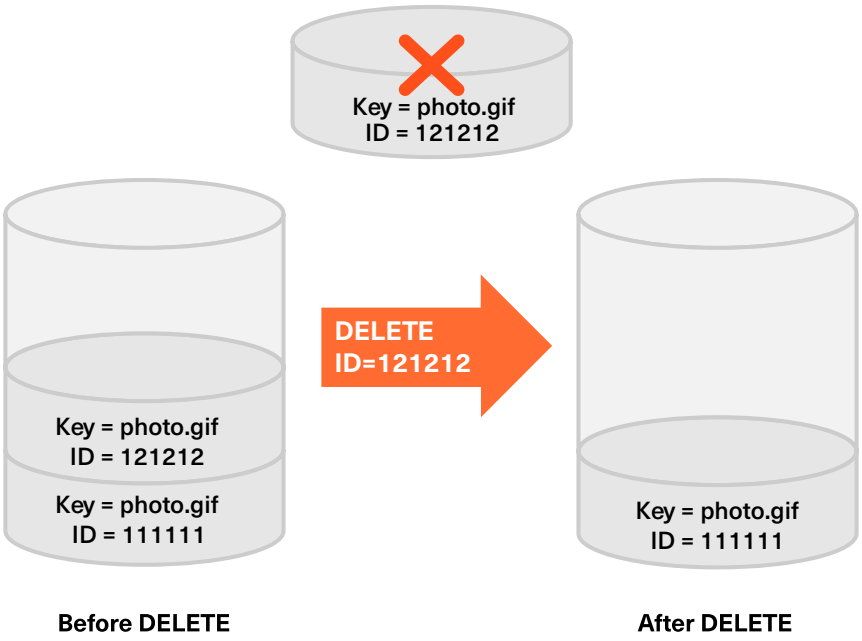
"When versioning is enabled, a simple DELETE cannot permanently delete an object. Instead, Amazon S3 inserts a delete marker in the bucket, and that marker becomes the current version of the object with a new ID."

- When you try to GET an object whose current version is a delete marker, Amazon S3 behaves as though the object has been deleted (even though it has not been erased) and returns a 404 error.

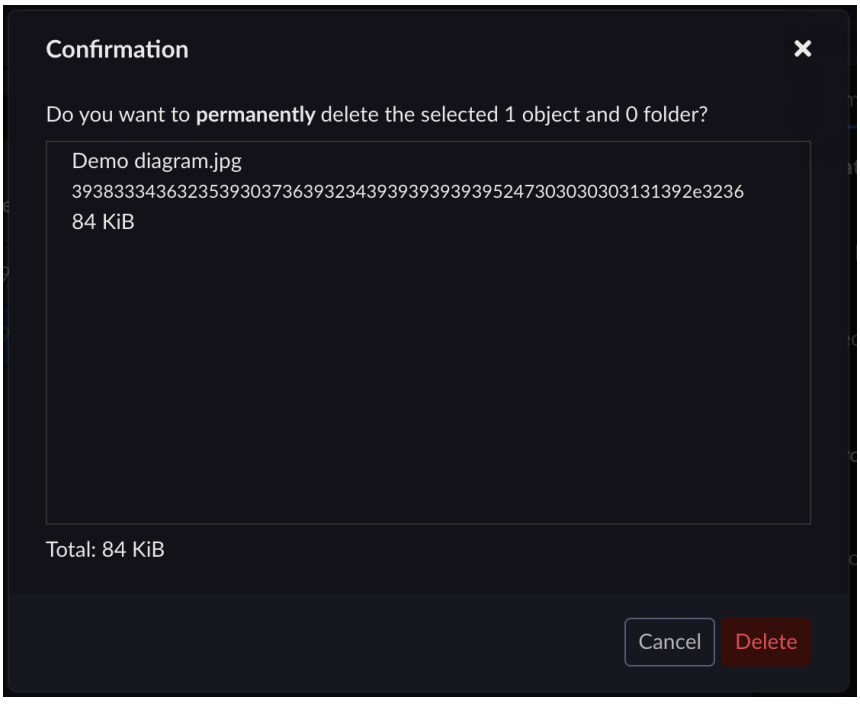


Note: In the above scenario, a lifecycle expiration policy can be used to delete the previous versions of the deleted objects and reclaim space.

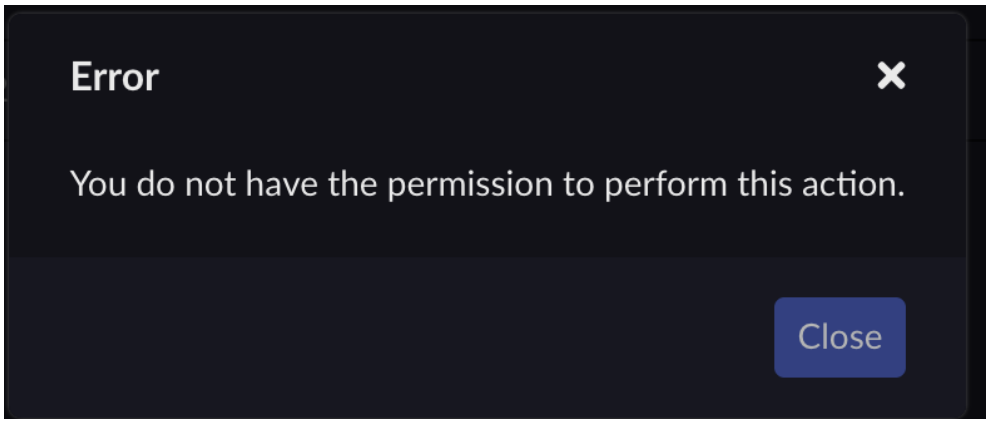
- To delete versioned objects permanently, you must use DELETE Object versionId.



OK, now we know how versioning works. Let's try to delete the specific version ID of the locked object. As we're deleting a specific version ID, this constitutes a permanent deletion of that version, so the system will ask us to confirm that's what we really want to do.



We are greeted with a rather reassuring message that our attempt at deletion has failed, as we don't have the required permissions



As a reminder, in Governance mode, if (and only if) you have the required permissions, you can indeed delete a locked object or reduce its retention period. In Compliance mode, no one has the permissions to delete or reduce the retention period of a locked object.

To restore the object to "regular" visibility, we simply delete the delete marker — and everything's back to normal.

You also mentioned ransomware...

Right — ransomware. The very nature of objects (their immutability) makes object storage a great starting point for ransomware protection. When combined with object versioning and object locking, we get a ransomware-proof S3 data repository, backed by validation from Cohasset Associates.

These ransomware-hardened capabilities are just a part of the reason why Scality gets the moniker "unbreakable storage." Our guarantees of 100% availability and up to 14 x 9s of durability set Scality apart as an industry leader in the object storage space.

How can you consume this ransomware-proof storage? Either through your applications, or you can store data directly into your object-locking enabled bucket(s).

It's important to realize that not all applications are made equal when it comes to supporting object lock. We could categorize them in two families:

- Object lock-**aware** applications: Applications that make the effort to tightly integrate with ObjectLock will typically use APIs such as "PUT object retention" and "DELETE Object versionId." With these applications, once ObjectLock is enabled at bucket creation time, everything from setting retention dates to deleting objects will be managed by the application and likely through the application's UI.
- Object lock-**compatible** applications: Some applications can work with ObjectLock (nothing will break) but were not necessarily designed to operate with ObjectLock- enabled buckets. Typically, such applications won't use "PUT object retention" and "DELETE Object versionId" APIs and will rely on default retention policies for setting retention times, standard DELETE operations, and lifecycle expiration policies to reclaim space by deleting previous versions of deleted objects.

Here's a list of popular backup, business continuity and archive applications that support object locking as of the publication date of this paper.

- Commvault
- HYCU
- Kasten
- Rubrik *
- Veeam
- Veritas NetBackup
- Veritas Enterprise Vault
- Zerto

* As of the publication date of this paper, ObjectLock is not yet certified with Scality S3 offerings.

See the Scality ISV compatibility list for the latest updates.

With these vendors and a Scality S3 object store configured with object locking, your data is protected against ransomware — even if cybercriminals manage to get into your network, past firewalls and onto your backup servers.

Conclusion

Encrypted threats are occurring every few seconds around the world. No business, large or small, is immune. Ransomware is the most common and potentially the costliest threat. While prevention is always smart, you need to ensure a fail-safe is in place in case of a breach. Data stored in a Scality S3 object store with object locking will ensure that your data remains intact, even if your backup applications are compromised.

We've examined how to create an object-locked bucket, how to set default retention policies of compliance or governance for each bucket, and how to identify objects with locks. You should now feel comfortable with object locking principles and Scality governing UIs. Remember that everything we did through the UI is also available through relevant S3 and IAM APIs.

Happy object locking!

Sources

Sonicwall. *Cyber Threat Report* (2022).

Veeam. *Ransomware Trends Report* (2022).

Author

Oleg Kokotović, *senior systems engineer*, Scality

ABOUT SCALITY Scality® storage propels companies to unify data management no matter where data lives — from edge to core to cloud. Our market-leading file and object storage software protects data on-premises and in hybrid and multi-cloud environments. With RING and ARTESCA, Scality's approach to managing data across the enterprise accelerates business insight for sound decision-making and maximum return on investment. To compete in a data-driven economy, IT leaders and application developers trust Scality to build sustainable, adaptable solutions. Scality is recognized as a leader by Gartner and IDC.

Follow us on Twitter and LinkedIn. Visit www.scality.com, or subscribe to our blog.
San Francisco. Paris. Washington, D.C. Tokyo. London.