The immutability of backups in the Nuabee solution

Linux and Windows backup · https://nuabee.com/documents/immutability-backups-in-nuabee-solution/

The need for immutability of backups

In an ultimate context, the immutability of backup copies is essential. At Nuabee, we address this issue using a multi-pronged approach.

How Nuabee works

The Nuabee solution works as follows:

- Backing up to the cloud ensures environmental separation, as in the previous case. The encryption keys are held solely by the Customer; Nuabee has no knowledge of them.
- The storage access token (AK/SK) is secured in a Nuabee vault, and the Customer has no knowledge of it.
- The separation and isolation of keys and access tokens improves security.
- Encrypting data at source and transporting it securely provides protection in the event of an attack on cloud object storage.
  - Data encrypted at source remains encrypted in S3 object storage.
- The agent used for backups has no modification rights and only performs push/pull operations.
  - Therefore, even if the Client were to be attacked, no changes to the infrastructure or backup environment would be possible.
- As in the previous case, the retention period allows you to return to a state where the data is not corrupted.

The contribution of Object Lock to S3 object storage

In the Nuabee solution, it is possible to add an extra layer of protection by implementing Object Lock at the storage bucket level. By default, this bucket will have a lock duration that is generally equivalent to the backup retention period (minimum 1 week, configurable).

This means that, in addition to the protections explained above, if someone with access to cloud storage wanted to destroy certain backups, object lock would provide additional protection.

In governance mode, if an object is deleted during the immutability period, the object is not deleted; instead, a deletion marker with a unique version ID is created, which becomes the current version. If you specify a version ID when deleting a WORM-protected object, OBS prevents you from deleting that object and returns a 403 error.