

Reducing Object Storage Spend for Veeam Backup & Replication

A technical guide to optimising S3-compatible storage

Version 1.0 • ASI Solutions • New Zealand

Disclaimer

This document provides technical guidance for reducing object storage costs when using Veeam Backup & Replication with S3-compatible object storage (including public cloud object storage). It is not legal, financial, or licensing advice. Cost outcomes depend on provider pricing, region, workload characteristics, and configuration. Always validate changes in a non-production environment and confirm supportability with your vendors.

Disclaimer 1
1. Executive summary 3
2. Object storage cost anatomy 3
3. How Veeam uses object storage 4
4. Where costs blow out in Veeam environments 5
4.1 Request volume and object count 5
4.2 Immutability and the 'API tax' 5
4.3 Retention and immutability interaction 6
4.4 Unsupported bucket lifecycle policies 6
5. Optimisation playbook (prioritised) 6
5.1 Design for local restores first 6
5.2 Match retention to storage tier rules 6
5.3 Reduce request-driven spend 6
5.4 Stop accidental duplication 7
6. Build a cost model you can defend 7
6.1 Common cost levers and trade-offs 7
7. Implementation checklist 8
7.1 First 30 days (highest ROI) 8
7.2 Next 60-90 days 8
8. Predictable pricing option (call to action) 8
References 9

1. Executive summary

Object storage is marketed as low cost per TB, but most organisations do not actually pay just for storage. They pay for storage plus request-based charges, retrieval charges (for cold tiers), and data transfer out (egress). Veeam can unintentionally amplify these variable charges because backup data is composed of many objects and is managed continuously.

This paper focuses on practical, supportable ways to reduce spend without gambling on recoverability. If you want a predictable bill, you need to control three things: how much you store, how often you touch it, and how far you move it.

Key takeaways:

- Stop optimising for $/TB in isolation. Model storage, API requests, retrieval, and data transfer together.
- Keep the common restore window on fast local storage (Performance Tier) and push long retention to Capacity/Archive tiers.
- Avoid provider-side lifecycle rules or tiering on Veeam buckets. Veeam must manage these objects to keep backups recoverable.
- Treat immutability as a security control with a cost footprint. Understand how your Veeam version and settings affect API calls.
- Engineer the network path (private endpoints, gateway placement) to avoid paying hidden data-transfer taxes.
- Measure continuously: storage growth, request volume, and restore traffic. Costs only become predictable when usage is observable.

2. Object storage cost anatomy

Before touching configuration, get clarity on what you are being billed for. Object storage bills usually break down into:

- Capacity stored (GB/TB-month) by storage class or tier.
- Requests / API operations (for example PUT, GET, LIST, HEAD, and metadata operations).
- Data retrieval and rehydration charges (especially for archive tiers).
- Data transfer charges (egress to internet, cross-region, and sometimes cross-zone or private connectivity components).
- Optional features (replication, monitoring, inventory, analytics, key management).

If you only benchmark $/TB, you are ignoring the parts that typically swing month to month.

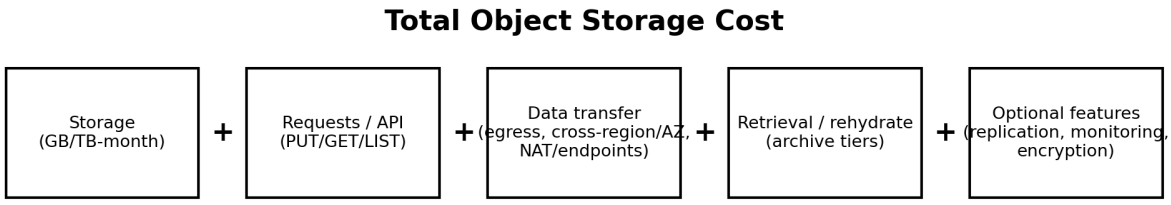


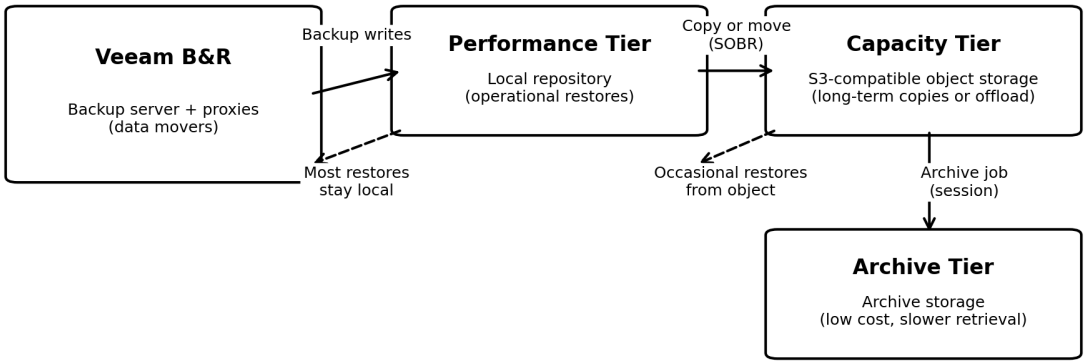
Figure 1 - A practical way to think about object storage cost.

3. How Veeam uses object storage

In Veeam Backup & Replication, object storage can be used directly as a backup repository, and it can also be attached to a Scale-Out Backup Repository (SOBR) to form storage tiers:

- Performance Tier: the primary landing zone for backups and the place you want most restores to come from.
- Capacity Tier: object storage used for long-term retention or secondary copies (copy mode) or offload (move mode).
- Archive Tier: an optional archive layer that can receive data from the performance tier or capacity tier.

When using move mode, Veeam runs an offload session automatically (by default every 4 hours) to collect eligible backup data and move it from the performance tier to the capacity tier. Copy mode behaves differently: it copies backups to object storage without reducing on-premises storage consumption.



Design goal for cost: keep common restores in the Performance Tier and push long retention to Object/Archive tiers.

Figure 2 - Simplified view of Veeam tiers and data movement.

Connection design matters. Veeam can connect to object storage directly (common for local or private connectivity) or via gateway servers (common when the object target is remote or requires a controlled egress point).

4. Where costs blow out in Veeam environments

Most cost surprises are not caused by a single setting. They come from a mismatch between how Veeam manages backup data and how cloud providers bill for object operations.

4.1 Request volume and object count

Veeam does not store one 'backup file' as one object. It breaks backup data into blocks and metadata, which increases the number of objects that need to be written, read, verified, retained, and deleted. More objects generally means more API operations.

4.2 Immutability and the 'API tax'

Immutability is non-negotiable for ransomware resilience, but it can increase API activity. Veeam has documented that Veeam Backup & Replication 12.x can generate higher immutability-related API calls when backing up directly to immutable object storage. Veeam Backup & Replication 13 includes improvements that reduce these immutability-related API calls.

If you are on 12.x and API charges matter in your environment, there is a real trade-off: you can reduce some API calls by increasing block size, but that typically increases storage consumption (incrementals grow). Do not treat this as a blanket tuning tip. Make it a measured decision based on your provider's pricing and your change rate.

4.3 Retention and immutability interaction

Retention deletes data. Immutability prevents deletion until a time window ends. That means you can have periods where you have logically expired backups that cannot be physically removed yet. If you do not model this, you will underestimate required capacity and cost.

4.4 Unsupported bucket lifecycle policies

Do not configure cloud provider lifecycle rules or tiering on buckets that Veeam uses as an object storage repository. Veeam needs to be the sole manager of objects for retention and data integrity. Provider-side lifecycle actions can lead to restore failures or data loss.

5. Optimisation playbook (prioritised)

If your objective is a lower bill and a better backup posture, tackle optimisation in this order. Most organisations do the opposite and waste time on micro-optimisations.

5.1 Design for local restores first

Egress is the silent killer in cloud storage economics. If you restore from cloud object storage back to on-premises, you are paying to pull your own data out. The fix is not a spreadsheet - it is architecture.

- Keep an operational restore window on the Performance Tier (fast, local). Size it based on how far back your real restores usually go.
- Use Capacity Tier for older restore points and compliance retention, where restores are rare.
- If you use cloud object storage for disaster recovery, plan to restore into the same cloud region where the data lives, not back over the internet.

5.2 Match retention to storage tier rules

Archive tiers are cheap because they assume long retention and low access. Many tiers enforce minimum storage durations and charge early deletion fees. If your retention is shorter than the minimum duration, you are paying archive prices without getting archive economics.

- Avoid archive tiers for backups that you test or restore regularly.
- Use archive tiers for long-term compliance where retrieval latency of hours is acceptable.
- Check minimum storage duration and early deletion periods before committing (these vary by provider and tier).

5.3 Reduce request-driven spend

Request charges are driven by object count, background maintenance, and immutability operations. Focus on changes that reduce high-frequency operations without compromising recoverability.

- Prefer Veeam Backup & Replication 13+ if immutability-related API costs are significant in your environment.
- On Veeam 12.x with direct-to-object immutable backups, only adjust block size or block generation intervals after measuring the impact on incremental growth and total stored TB.
- Use multiple buckets (where supported) to avoid throttling limits that cause retries and extra requests.
- Ensure network design avoids unnecessary hops that add costs (for example, NAT gateway processing or cross-zone traffic).

5.4 Stop accidental duplication

Redundancy is good. Paying twice for redundancy you do not need is not. The common culprits are copy mode everywhere, replication turned on by default, and multiple backup copies with overlapping retention.

- Use copy mode when you need a second copy for DR or immutability separation. Do not enable it just because it is available.
- Be deliberate about provider replication (cross-region replication is a cost multiplier).
- Consolidate overlapping backup copy policies and align retention across tiers.

6. Build a cost model you can defend

Decision makers do not need perfect modelling. They need modelling that is directionally correct and transparent about assumptions.

At minimum, track these inputs:

- Protected source data (TB) and daily change rate.
- Compression and deduplication outcomes (measured, not assumed).
- Retention policy (operational restore window vs long-term retention).
- Immutability period and how it overlaps with retention.
- Monthly restore and test-restore volume (TB) and where it is restored to (local vs cloud vs internet).
- Request volume trends (PUT/GET/DELETE/HEAD) and any periodic spikes.

If you cannot explain the bill using these inputs, you are not in control of object storage spend - you are hoping.

6.1 Common cost levers and trade-offs

Lever	Primary saving	Trade-off / risk	Owner
Increase operational restore window on performance tier	Less restore egress and retrieval spend	More local capacity required	Backup + infrastructure
Move mode for older restore points	Reduced local storage footprint	Restores from capacity tier may be slower and can incur egress	Backup

Archive tier for compliance retention	Lower $/TB for long retention	Higher retrieval latency and potential early deletion fees	Backup + risk/compliance
Immutability tuning (version-dependent)	Lower immutability-related API costs	May increase stored TB; must stay supportable	Backup + security
Network optimisation (private endpoints, gateway placement)	Lower data-transfer taxes and fewer failures	Design effort; may require cloud networking changes	Cloud networking

7. Implementation checklist

7.1 First 30 days (highest ROI)

- Baseline current costs by category: storage, requests, retrieval, and data transfer.
- Confirm Veeam version and where object storage is used (capacity tier, archive tier, or direct-to-object).
- Remove any bucket lifecycle rules or tiering policies on Veeam-managed buckets (unsupported).
- Set budgets/alerts for request spikes and data transfer out.
- Validate operational restore window matches real restore behaviour.

7.2 Next 60-90 days

- Evaluate Veeam upgrade path if you are on 12.x and paying an immutability-related API premium.
- Model archive tier suitability against retention and retrieval objectives.
- Optimise network path to object storage (private endpoints, gateway placement, avoid unnecessary NAT).
- Document a restore test plan that minimises egress (restore in-region when using cloud object storage).

8. Predictable pricing option (call to action)

If your analysis shows that API calls and egress dominate your spend, the problem is not that your backups are too large - it is that your pricing model is variable by design. At that point, cost optimisation becomes a never-ending exercise.

ASI InfiniStor is an S3-compatible object storage platform designed for predictable backup economics: a simple $ per TB price for hot or cold storage, with no API request fees and no ingress/egress charges. For organisations that want S3 compatibility without hyperscaler bill volatility, it can be a cost-effective local alternative.

Next step: run a side-by-side cost model using your real backup footprint and restore behaviour.

References

- Veeam Help Center - Scale-Out Backup Repositories (SOBR) - https://helpcenter.veeam.com/docs/vbr/userguide/backup_repository_sobr.html?ver=13
- Veeam Help Center - Moving Backups to Capacity Tier (offload session) - https://helpcenter.veeam.com/docs/vbr/userguide/capacity_tier_move.html?ver=13
- Veeam Help Center - Archive Tier - https://helpcenter.veeam.com/docs/vbr/userguide/archive_tier.html?ver=13
- Veeam Help Center - Object Storage Repositories (connection modes, multiple buckets) - https://helpcenter.veeam.com/docs/vbr/userguide/object_storage_repository.html?ver=13
- Veeam Best Practices - Object repositories (lifecycle rules not supported) - https://bp.veeam.com/vbr/2_Design_Structures/D_Veeam_Components/D_backup_repositories/object.html
- AWS Prescriptive Guidance - Back up and archive data to Amazon S3 with Veeam Backup & Replication - https://docs.aws.amazon.com/prescriptive-guidance/latest/patterns/back-up-and-archive-data-to-amazon-s3-with-veeam-backup-replication.html
- Veeam KB4470 - Increase in API Calls when Performing Backups Directly to Immutable Object Storage - https://www.veeam.com/kb4470
- Veeam Help Center - Capacity Tier Retention Policy and immutability behaviour - https://helpcenter.veeam.com/docs/vbr/userguide/capacity_tier_retention.html?ver=13
- AWS - Amazon S3 Pricing - https://aws.amazon.com/s3/pricing
- AWS Docs - Archival storage in S3 Glacier storage classes (minimum storage durations) - https://docs.aws.amazon.com/AmazonS3/latest/userguide/archival-storage.html
- Microsoft Azure - Blob Storage pricing (early deletion periods) - https://azure.microsoft.com/en-au/pricing/details/storage/blobs
- Microsoft Learn - Estimate the cost of archiving data (Azure Blob Storage) - https://learn.microsoft.com/en-us/azure/storage/blobs/archive-cost-estimation