WHITEPAPER

Veeva Vault Environment Management

A Practitioner's Guide to Sandbox Administration, Refresh Cycles, Snapshots and Configuration Deployment

Wolvio Solutions Private Limited | March 2026
www.wolviosolutions.com

Table of Contents

Executive Summary

For life sciences organisations running Veeva Vault, environment management is not a peripheral concern — it sits at the heart of platform reliability, regulatory compliance, and delivery velocity. Yet it is frequently underinvested in until something goes wrong: a misconfigured production deployment, a sandbox that has drifted so far from production that its test results are meaningless, or a refresh cycle that consumes more effort than it should.

This white paper, authored by the Wolvio Solutions platform practice, provides a comprehensive operational guide for managing the Veeva Vault environment. It covers the full lifecycle of sandbox administration — from sizing and provisioning through refresh cycles, snapshot management, and configuration migration — with the depth required by Vault Administrators and IT Platform teams responsible for keeping these environments running correctly.

Key topics covered in this guide:

- Sandbox types, sizes, data limits, and default entitlements
- Self-service sandbox creation and administration by Vault Administrators
- Sandbox refresh cycles — frequency rules, the mechanics of refresh, and why regular refresh is essential
- Snapshot management — types, allocations, size limits, and practical use cases
- Configuration Migration Packages (VPKs) — creation, validation, and production deployment
- A complete 20-step operational refresh checklist, ready to use as a change record
- Environment strategy, change control, and release management best practices

> **Practitioner Insight:** Organisations that invest in a documented environment management strategy consistently see fewer production incidents, faster configuration delivery, and stronger audit readiness. This guide gives your team the operational foundation to build that strategy.

1. Introduction to Veeva Vault Environments

Veeva Vault is a cloud-based content management and collaboration platform purpose-built for the life sciences industry. It serves as the backbone for managing regulated documents, clinical data, quality processes, and commercial content. Effective environment management is fundamental to any successful Vault implementation. It protects live data from development activity, enables safe validation of configuration changes before production, and supports the regulatory compliance obligations of GxP organisations.

A critical point that is often misunderstood in the field: all routine environment management operations in Veeva Vault, creating sandboxes, refreshing them, managing snapshots, and deploying configuration packages, are fully self-service operations performed by Vault Administrators through the Admin UI. No Veeva Support involvement is required for these day-to-day activities.

1.1. Environment Types

Veeva Vault organises environments into two primary operational categories:

- **Production Vault —** The live environment where real business operations occur. Regulated documents are managed, workflows are executed, and approved data is stored here. Stability and data integrity are paramount.
- **Sandbox Vaults —** Non-production copies of the production Vault used for development, configuration testing, user acceptance testing (UAT), integration development, and training. Sandbox Vaults are created and managed entirely by Vault Administrators, without Veeva involvement.

Veeva also provisions a temporary Prerelease Vault before each major platform release, allowing administrators to preview new features and test upgrade readiness ahead of the production upgrade. Prerelease Vaults do not count against sandbox entitlements.

1.2. Domain Architecture

Every Veeva Vault customer is provisioned with domains that separate live and non-live environments. Understanding this architecture is important for planning cross-environment user access and SSO configuration:

- **Production Domain —** Contains the live production Vault. All production data, user records, and configuration reside here.
- **Sandbox Domain —** Contains all non-production sandbox Vaults. Sandboxes are always isolated within the sandbox domain. UAT Vaults should also reside in the sandbox domain.

- **Prerelease Domain —** Contains the Prerelease Vaults made available by Veeva ahead of each major platform release. Prerelease Vaults are automatically deactivated once the release window closes.

Note: Domain-level settings such as SSO configuration and password policies are configured by Domain Admins. If your organisation needs to test domain-level settings in isolation, this should be handled within the domain admin configuration. Prerelease Vaults are available during the Veeva release cycle and are made inactive after the window closes.

Within a domain, Vault Administrators have full control over the sandbox lifecycle. Users can navigate between Vaults across domains using the Vault Selector, provided they hold a common user account in each domain.

Practitioner Insight: In practice, many organisations provision a second sandbox domain specifically for regression and release testing. This gives teams the ability to test domain-level configurations (SSO, IP ranges) without risk to the primary sandbox estate. If you require an additional domain, contact Veeva Support — this is one of the few environment operations that requires a support engagement.

2. Sandbox Sizes and Limits

Sandbox Vaults are available in six sizes — Small, Medium, Large, Very Large, Extra Large, and Full. Each has defined limits on object records and document versions. These limits are enforced at runtime: once a sandbox reaches its limit, creation of new records and documents is blocked until data is removed. Administrators can check current sandbox usage at any time via Admin > Deployment > Sandbox Vaults.

2.1. Sandbox Size Reference

The table below summarises all available sandbox sizes, their data limits, included snapshots, and default entitlements per production Vault:

Size	Max Object Records	Max Doc Versions	Snapshots Included	Included in Production
Small	100,000	10,000	2	4 per production Vault
Medium	1,000,000	100,000	2	2 per production Vault
Large	10,000,000	1,000,000	2	0 — add-on purchase
Very Large	100,000,000	10,000,000	2	0 — add-on purchase
Extra Large	500,000,000	50,000,000	2	0 — add-on purchase
Full	No Limit	No Limit	2	1 per production Vault

Note: The object record limit excludes system-managed records and internal configuration data. Sandbox Vaults that exceed their limits are blocked from creating new object records or document versions. Admins can delete records to return the Vault to compliance and recheck usage via the Sandbox Vaults admin page to lift the block.

Practitioner Insight: In our experience, Small sandboxes are sufficient for the majority of active development work — configuration metadata is lightweight. Where size starts to matter is in integration testing and UAT environments that need to hold representative volumes of object records and documents. Always right-size your sandboxes against actual test data requirements rather than defaulting to the largest available size.

2.2. Document Template Limits

Regardless of sandbox size, each sandbox Vault has a hard limit of 5 GB for document templates (both active and inactive). If a sandbox is created or refreshed from a source Vault containing more than 5 GB of document templates, no document templates will be copied to the resulting sandbox. Administrators managing large template libraries should be aware of this constraint and plan accordingly.

2.3. Sandbox Expiration

Small sandbox Vaults automatically expire after 30 days of inactivity. Vault sends email warnings to Vault Owners twice before the sandbox is deleted. Blocked or inactive sandbox Vaults display a warning in the Name field on the Sandbox Vaults page of the parent Vault.

> **Note:** Prerelease Vaults are provisioned separately by Veeva and do not count against sandbox entitlements. Large, Very Large, and Extra Large sandboxes are available as add-on purchases through Veeva's account team.

3. Creating and Administering Sandbox Vaults

Creating and administering sandbox Vaults is entirely within the control of Vault Administrators. No Veeva Support involvement is required for any standard sandbox management operation.

3.1. Creating a Sandbox

Vault Administrators with the Vault Owner security profile can create new sandboxes at any time from the production Vault or from another sandbox Vault that has sandbox allowances granted to it. The process is as follows:

- Navigate to **Admin > Deployment > Sandbox Vaults** in the source Vault.
- Under Active Sandbox Vaults, click **Create**.
- Select **From Vault** (copies current source configuration) or **From Snapshot** (restores from a saved snapshot).
- Enter a **Name** following your organisation's naming convention (e.g., ACME-DEV, ACME-UAT).
- **Select a Size** based on expected data volume and test scope.
- **Select a Release** — typically the current general release, or the upcoming prerelease version.
- **Select the Domain** (your sandbox domain).
- Optionally check **Add Current User as Vault Owner** for immediate admin access.
- Click **Save**. Vault provisions the sandbox asynchronously and sends an email notification on completion.

> **Note:** Sandbox Vaults may appear unavailable immediately after creation while initialisation completes. This is expected — wait a few minutes and refresh the page. If the sandbox remains unavailable after 60 minutes, check the Vault Notifications log for error details.

3.2. What Is Copied During Creation

When creating a sandbox from Vault, the following configuration elements are copied from the source:

- Document types, subtypes, and classification hierarchies
- Lifecycle definitions, states, and transition rules
- Workflow definitions and participant groups

- Object definitions, fields, relationships, and picklists
- User security profiles, permission sets, and groups
- Application settings and integration configuration
- Vault Java SDK code — triggers, actions, services

The following are excluded from sandbox creation and refresh:

- Document content files (unless creating a Full configuration sandbox that includes content)
- Document templates exceeding the 5 GB limit
- User passwords and active session data
- File staging server contents (unless linked staging is configured)
- Audit trail history and notification history

3.3. Managing User Access

Access to sandbox Vaults works identically to access to production Vaults. Administrators can add new users, grant access to existing domain users, or add cross-domain users. When creating a sandbox, the Add Current User as Vault Owner option automatically provisions the creating admin with full administrative access, enabling immediate configuration without a separate access setup step.

3.4. Granting Sandbox Allowances

Administrators can grant sandbox allowances from one Vault to another, enabling a cascading sandbox model where a staging sandbox can provision its own lower-level test environments. This is configured under Admin > Deployment > Sandbox Vaults using the Grant Allowances option. This approach is particularly useful for organisations running parallel workstreams where each team needs its own isolated environment.

3.5. Deleting Sandbox Vaults

Sandbox Vaults are deleted by Vault Administrators via the Actions menu on the Sandbox Vaults page. If the sandbox has linked file staging, administrators can manage the file staging link directly in the Vault Admin UI: navigate to Admin > Settings > File Staging Server, then remove the linkage before proceeding with deletion. No Veeva Support involvement is required for file staging, relinking, or unlinking.

4. Sandbox Refresh Cycles

Refreshing a sandbox is one of the most important routine operations in Veeva Vault environment management. A refresh overwrites the sandbox's configuration with the latest configuration from its source Vault, re-synchronising the sandbox to accurately reflect the current state of production. Like all standard sandbox operations, refresh is fully self-service and performed directly by Vault Administrators.

4.1. What a Refresh Does

- Overwrites all sandbox configuration — document types, lifecycles, objects, fields, workflows, security settings, SDK code — with the latest from the source Vault.
- Clears all custom data, test object records, and document versions created in the sandbox. Sandbox-specific data is not preserved.
- Does NOT delete or overwrite existing snapshots. Snapshots for the sandbox remain intact after a refresh.
- Has no effect on the production Vault or any other environment.

> **Note:** Refreshing a sandbox permanently replaces its configuration and data with the source. Always create a snapshot before refreshing if you need to preserve the current sandbox state for rollback purposes.

4.2. Refresh Frequency

The frequency at which a sandbox can be refreshed is governed by its size:

Sandbox Size	Maximum Refresh Frequency
Small	Up to five (5) times in a 24-hour period
Medium	Once in a 24-hour period
Large	Once in a 24-hour period
Very Large	Once in a 24-hour period
Extra Large	Once in a 24-hour period
Full	Once in a 24-hour period

If you exceed your sandbox allowance, your allowance will display as a negative number, and you will be unable to refresh any sandbox Vault until one or more sandboxes are deleted. If you need to reset a sandbox more frequently than the standard quota allows, use Refresh from Snapshot — this does not count against the refresh limit.

4.3. How to Refresh a Sandbox

- Navigate to **Admin > Deployment > Sandbox Vaults** in your production Vault.
- Under Active Sandbox Vaults, locate the sandbox to refresh.
- Click the **Actions** icon and select **Refresh from Vault** or **Refresh from Snapshot**.
- Confirm the operation. Vault begins the refresh asynchronously.
- A notification email is sent to the requesting administrator when the refresh is complete.

> **Note:** During a Veeva platform maintenance window ahead of a new release, a warning banner appears on the Sandbox Vaults page indicating that refresh operations may take longer than usual. All other Vault features remain fully operational. The refresh will complete — it simply may take longer than normal.

4.4. Why Regular Refresh Is Necessary

Sandbox refresh is not just a reset button. It is a foundational practice that directly determines the reliability of your development and testing operations. Below are the key reasons to maintain a disciplined refresh cadence:

4.4.1. Configuration Drift

Production Vaults evolve constantly. Document types are added, lifecycles are updated, workflows are modified, and security profiles are adjusted. A sandbox created months ago without subsequent refreshes no longer accurately reflects production. Testing against a drifted sandbox produces results that do not predict production behaviour, introducing deployment risk and eroding stakeholder confidence in the testing process.

4.4.2. Release Alignment

Veeva releases three major platform updates per year. Each release can change system object behaviour, introduce new configuration options, or deprecate existing features. Refreshing sandboxes at the start of each prerelease period ensures that testing is conducted against the correct version baseline and that any compatibility issues with existing SDK code or configuration are identified before the production upgrade.

4.4.3. Regulatory Compliance and Validated Environments

Organisations operating under GxP regulations must maintain validated environments. Refreshing a sandbox and re-executing qualification protocols (IQ/OQ/PQ) from a clean baseline is standard practice for maintaining sandbox environments in compliance with validation standards. Many organisations schedule sandbox refreshes at the start of every formal validation cycle to ensure an auditable, documented baseline.

4.4.4. Developer Productivity

Development teams inevitably introduce experimental configurations that diverge from production intent. A refresh provides a clean slate, enabling the next sprint to begin from a known-good baseline without the overhead of manually reverting sandbox-specific changes or debugging configuration conflicts introduced in previous iterations.

4.4.5. Data Hygiene

Sandbox environments accumulate test data over time — synthetic documents, dummy object records, test users, trial workflow executions. This accumulation pushes sandboxes toward their size limits, degrades performance, and creates confusion for testers. A refresh clears all accumulated test data and restores a clean, production-aligned environment ready for the next testing cycle.

> **Practitioner Insight:** One of the most common mistakes we see in the field is treating sandbox refresh as a disruptive event rather than a routine operation. Teams that schedule regular, predictable refreshes — aligned to their sprint cadence or release calendar — consistently experience fewer 'environment issues' during testing, and their test results are far more credible when presented for business sign-off.

5. Sandbox Refresh Checklist

The following checklist provides a complete operational record for every sandbox refresh. Administrators should complete and retain this checklist for each refresh, referencing the associated change ticket for audit and traceability purposes. For GxP-regulated environments, completed checklists form part of the operational qualification evidence.

5.1. Refresh Record

Field	Details
Vault Name	
Vault ID (Pre-Refresh)	
Vault ID (Post-Refresh)	
Source Vault Name	
Suite / Application	
Date of Refresh	
Jira / Change Ticket Reference	
Refresh Performed by	
Approved by	

5.2. Pre-Refresh

#	Checklist Step	Guidance	Done
1	Business / change approval obtained	Confirm the refresh is approved by the business owner and logged in the change management system. Ensure the Jira or change ticket is raised and approved before proceeding.	□
2	Impacted users and teams notified	Communicate the refresh window, expected downtime, and period of sandbox unavailability to all teams currently using the environment.	□
3	Sandbox size limits verified	Check current sandbox data usage is within its defined limits (Admin > Deployment > Sandbox Vaults). Resolve any over-limit conditions before initiating the refresh.	□

#	Checklist Step	Guidance	Done
4	Target release version confirmed	Confirm whether the sandbox will be refreshed to the current general release or the upcoming prerelease version. Document the intended version.	□
5	Snapshot taken from source Vault (including data)	Take a Snapshot from the source Vault to preserve a rollback point. Record the snapshot name and creation timestamp before proceeding.	□
6	Data backup completed via Vault Loader	Export critical runtime data via Vault Loader — at minimum: User Role Setup, User Roles, Group Membership, and any object records that will not be reloaded from a standard data source post-refresh.	□
7	Active integrations and scheduled jobs paused	Disable or pause any integration connections, scheduled SDK jobs, or external system processes that read from or write to the sandbox, to prevent errors during the refresh window.	□
8	Sandbox-specific settings documented	Record all sandbox-specific configuration (API endpoints, integration credentials, test user lists, hardcoded Vault IDs) that will need to be reconfigured after the refresh.	□

5.3. Refresh Execution

#	Checklist Step	Guidance	Done
1	Refresh initiated in Vault Admin	Navigate to Admin > Deployment > Sandbox Vaults. Select Refresh from Vault or Refresh from Snapshot. Confirm the action and record the time initiated.	□
2	Refresh completion notification received	Wait for the Vault email notification confirming the refresh is complete. Do not proceed to post-refresh steps until the notification is received.	□
3	Vault ID post-refresh recorded	Log the Vault ID displayed after refresh in the Refresh Record table above. Confirm it matches the expected sandbox.	□

5.4. Post-Refresh

#	Checklist Step	Guidance	Done
1	Integration credentials and API endpoints updated	Refresh overwrites integration configuration from production. Update all sandbox-specific API endpoints, credentials, and connection settings. Ensure no integration points inadvertently at production.	☐
2	Integration connections re-enabled and tested	Re-enable integration connections with the correct sandbox-specific URLs and credentials. Run connection tests where available to confirm connectivity.	☐
3	Sandbox-only test users re-invited or re-enabled	Re-invite or re-enable any sandbox-specific test users (UAT participants, external users, service accounts) who do not exist in production and were removed by the refresh.	☐
4	Hardcoded Vault IDs and URLs updated in connected systems	Update any external systems, scripts, CI/CD pipelines, or configuration files that reference the sandbox Vault ID or URL, where these values may have changed post-refresh.	☐
5	Vault Loader backup data reloaded	Reload all Vault Loader extracts captured pre-refresh: User Role Setup, User Roles, Group Membership, and any additional object records backed up prior to the refresh.	☐
6	Additional test data loaded	Load any test data packs, Test Data Packages, or additional reference data required for the current development or testing cycle.	☐
7	Configuration smoke test completed	Perform a targeted smoke test: verify key document types load correctly, lifecycle state transitions execute as expected, workflows can be initiated, and object records can be created.	☐
8	SDK and custom code validated	Confirm Vault Java SDK triggers, actions, and services are operating correctly by running functional checks across any custom-coded behaviour in scope for this environment.	☐

#	Checklist Step	Guidance	Done
9	Team notified — sandbox ready for use	Communicate to all impacted users and teams that the refresh is complete and the sandbox is available for use.	□

> **Note:** Complete and retain this checklist as part of the change record for every sandbox refresh. For GxP-regulated environments, completed checklists contribute to the operational qualification evidence for the environment.

6. Sandbox Snapshots

Snapshots are point-in-time copies of a sandbox Vault's configuration and data. They allow administrators to preserve a known-good state that can be used to rapidly create new sandboxes or restore an existing sandbox — without waiting for a full refresh from production or manually reloading test data.

6.1. Snapshot Types

- **Configuration Snapshot —** Captures Vault configuration only (metadata, document types, lifecycles, objects, fields, etc.) with no data records or document content. Smaller and faster to create. Ideal for preserving a configuration baseline before starting a new development sprint.
- **Data Snapshot —** Captures configuration plus object records and document versions. Ideal for preserving test environments pre-loaded with reference or regression data, avoiding the need to reload data after every sandbox refresh. The following exceptions apply to data snapshots:
    - Audit logs are not copied
    - Vault Connections are reset (similar to sandbox creation)
    - Vault Token records are reset
    - The Scheduled Data Export job is reset
    - Legacy Collaborative Authoring settings are cleared. Only enhanced Collaborative Authoring configurations are copied.

6.2. Snapshot Allocations and Size Limits

Veeva provides two (2) snapshots per sandbox, regardless of sandbox size. Data Snapshots are subject to the following size limits (content size limits exclude file staging server data):

Sandbox Size	Max Object Records	Max Doc Versions	Max Content Size
Small	100,000	10,000	50 GB
Medium	1,000,000	100,000	50 GB
Large	10,000,000	100,000	50 GB
Very Large	10,000,000	100,000	50 GB
Extra Large	10,000,000	100,000	50 GB
Full	10,000,000	100,000	50 GB

Note: A Data Snapshot cannot be created if the source sandbox already exceeds the size limits above. Ensure the sandbox is within limits before attempting to take a data snapshot.

6.3. Snapshot Versioning and Expiration

Vault assigns a snapshot the same release version as its source sandbox Vault when created. Over time, the snapshot's release version will lag behind that of the source sandbox as the platform is upgraded. Automatic snapshot expiration eliminates the need to manually upgrade snapshots across multiple releases.

Snapshots automatically expire if unused for a certain period. Updating or upgrading a snapshot resets its Expires On date. Vault Owners receive an email notification seven (7) days before a snapshot expires. This warning is also displayed on the Sandbox Snapshots page.

6.4. Practical Use Cases for Snapshots

- **Pre-Development Baseline —** Take a snapshot of a freshly refreshed sandbox before the start of a development sprint. If development work needs to be abandoned or rolled back, restore from the snapshot instantly without consuming the refresh quota.
- **UAT Environment Preparation —** Instead of using Vault Loader to populate a UAT sandbox with test data, create a data snapshot from a development sandbox and use it to spin up the UAT environment with pre-loaded test data in minutes.
- **Regression Test Baseline —** Preserve a snapshot after a successful regression test run. Each subsequent regression cycle starts from an identical baseline, ensuring consistent, repeatable, and auditable results.
- **Parallel Development Streams —** When multiple teams need isolated environments simultaneously, use a shared baseline snapshot to rapidly clone separate sandboxes for each team without repeated Vault Loader data loads.
- **Frequent Intra-Cycle Resets —** Use Refresh from Snapshot to reset a sandbox multiple times per day during intensive testing without consuming the daily refresh quota.

Practitioner Insight: Snapshots are one of the most underutilised capabilities in Veeva Vault environment management. Teams that build snapshot-taking into their sprint rituals — taking a clean snapshot before every sprint and after every successful UAT cycle — dramatically reduce the time spent resetting environments and improve the consistency of their testing baselines.

7. Configuration Migration Packages (VPKs)

Configuration Migration Packages (VPKs) are the primary mechanism for promoting configuration changes from sandbox environments into production. VPKs bundle selected configuration components, data, and SDK code into a portable archive that can be imported and deployed across Vaults. A disciplined VPK workflow is the single most important safeguard against uncontrolled production configuration changes.

7.1. What Can Be Migrated

- Configuration components: document types, lifecycles, workflows, objects, fields, picklists, security profiles, roles, user groups
- Reference data and test data records (when using data packages)
- Vault Java SDK custom code — triggers, actions, services, extensions
- MDL-based schema definitions for bulk metadata operations

7.2. Creating and Exporting a VPK

- **Vault Compare (recommended) —** Automatically identifies configuration differences between two Vaults and generates an outbound package. This ensures only intended changes are included and reduces the risk of over-deploying.
- **Manual Assembly —** Administrators manually select specific components, useful when promoting a targeted subset of changes rather than all differences between environments.

Once created, the outbound package is exported as a .vpk file. Vault sends an email notification with a download link when the export is complete. Store VPK files in a source control repository with descriptive commit messages for full traceability.

7.3. Importing and Deploying a VPK

- Navigate to **Admin > Deployment > Inbound Packages** in the target Vault and upload the VPK file.
- Vault asynchronously validates the package and sends an email notification when complete.
- Review the **Validation Log**. Address any components flagged as **Missing – Block** before proceeding. Components flagged as **Warning** should be reviewed.
- Enable **Configuration Mode** on the target Vault (Admin > Settings > General Settings).
- Deploy the package from the **Actions** menu on the Inbound Packages page. Vault determines the correct internal deployment order automatically.

- After deployment, disable Configuration Mode and validate key areas of the target Vault configuration.

Note: When deploying multiple sequential VPKs, re-validate each subsequent package after the previous one is deployed. This ensures dependency resolution reflects the updated target Vault state and prevents unexpected blocking errors.

7.4. VPK Deployment Best Practices

- Extract the Configuration Report before and after every deployment to maintain a full record of configuration state at each stage.
- Always run Vault Compare before creating a production VPK to confirm only intended changes are included.
- Validate VPKs in a QA or staging Vault before deploying to production.
- Export the Package Component Comparison (XLSX) and share with stakeholders before production deployment.
- For circular dependencies between objects, use a staged approach: remove the circular reference, deploy, then reintroduce it in a second package.
- Store all exported VPK files in source control cross-referenced to change management records.
- Deploy to production during low-traffic windows and maintain a documented rollback plan for every deployment.

Practitioner Insight: Never promote configuration changes directly in production — not even 'small' changes. The Vault Audit Trail will record the change, but without a corresponding VPK, you lose the ability to reproduce or roll back the change reliably. Every change that matters belongs in a VPK.

8. Environment Strategy and Best Practices

A well-defined environment strategy is what separates reactive platform management from proactive, governed delivery. The investment required is modest — a documented environment plan, a consistent naming convention, and a clear promotion pathway — but the operational benefits compound significantly over time.

8.1. Recommended Environment Flow

Stage	Environment	Sandbox Size	Refresh Cadence	Purpose
Development	Sandbox	Small	Per sprint / as needed	Active configuration build, SDK development, unit testing
QA / Test	Sandbox	Small–Medium	Per release cycle	Functional testing, regression, defect validation
UAT / Staging	Sandbox	Medium	Before each UAT window	Business user acceptance testing and sign-off
Prerelease	Veeva-provisioned	Veeva managed	Veeva-managed	Upgrade readiness, new feature validation
Production	Production	N/A	N/A	Live operations, regulated documents, approved data

> **Note:** For migration activities, select sandbox sizing based on the data volume involved. Migration environments typically require larger sandboxes than standard development or testing workloads.

8.2. Sandbox Naming and Management

- Adopt a consistent naming convention: [ORG]-[TYPE] (e.g., ACME-DEV, ACME-QA, ACME-UAT).
- Document each sandbox's owner, purpose, and intended refresh schedule in an Environment Management Plan.
- Dedicate sandboxes to specific purposes — avoid using a single sandbox for both active development and formal UAT.
- Monitor sandbox data usage regularly to avoid unexpectedly hitting size limits mid-sprint.

8.3. Refresh Strategy

- Refresh development sandboxes at the start of each sprint or release cycle to align with production.
- Refresh UAT sandboxes before each formal UAT window to ensure testers work against current production configuration.
- Immediately after a fresh refresh, take a Configuration Snapshot to capture the clean baseline before development begins.
- Use Refresh from Snapshot for quick resets within a development cycle to preserve the live refresh quota.

8.4. Change Control

- Never make configuration changes directly in production. All changes must originate in a development sandbox, be validated, and be promoted via VPK.
- Use Configuration Mode during deployment windows in production.
- Maintain a deployment log recording each VPK, its contents, the deploying administrator, and the deployment date.
- Use the Vault Audit Trail to review all configuration changes applied post-deployment.
- For GxP environments, maintain qualification documentation (IQ/OQ/PQ) aligned to each production configuration change.

8.5. Compliance and GxP

- Ensure all sandboxes used for validation testing are refreshed and documented as part of the validation lifecycle.
- Leverage Veeva's IQ/OQ documentation delivered with each release to reduce the qualification burden on platform-level changes.
- Track all configuration changes destined for production in your change management system, cross-referenced to Vault audit trail records.

9. Release Management

Veeva Vault follows a three-release-per-year cadence (R1, R2, R3) in General Release cycle. Each major release introduces platform enhancements, new features, and occasionally deprecates older capabilities. Coordinating sandbox environments with the Veeva release cycle is a critical part of maintaining a healthy platform estate.

9.1. Prerelease Vaults

Four weeks before each general release, Veeva makes a prerelease sandbox available. Prerelease sandboxes are fully self-service: administrators with the Vault Owner security profile can create them via Admin > Deployment > Sandbox Vaults by selecting the Prerelease Available option.

Prerelease Vaults enable organisations to:

- Test that existing configurations and SDK code behave correctly on the new release version
- Evaluate new features before deciding which to activate in production
- Prepare updated qualification documentation for GxP environments ahead of the production upgrade

Prerelease Vaults remain available for four weeks after the general release, do not count against sandbox entitlements, and require no Veeva Support involvement to create.

9.2. Feature Enablement After Upgrade

When the production Vault is upgraded, most new features are not automatically active. Administrators should review the release notes and the About the Release page to identify what requires explicit enablement:

- **Simple toggle —** Some features are enabled via a checkbox in Admin > Settings.
- **Configuration required —** Other features require lifecycle, workflow, or document field configuration before they are active.
- **Automatic —** Some platform enhancements are automatically applied by Veeva and are clearly marked in the release notes.

> **Practitioner Insight:** Build prerelease testing into your standard release calendar. Allocating two weeks of prerelease sandbox testing before each general release — covering configuration smoke tests, SDK regression tests, and a review of the release notes — is sufficient for most organisations to catch compatibility issues and plan feature enablement decisions before the production upgrade.

9.3. Limited Release Vaults

Veeva provides Limited Release vaults (called Fast Train Vaults) where releases are more frequent, and features are pushed before the General Release cycle. These Limited Release vaults can be used for Release Impact Assessments before the PreRelease vaults are made available.

10. Troubleshooting Common Issues

Sandbox Creation or Refresh Takes Longer Than Expected

Check whether a Veeva platform maintenance window is in progress — a warning banner will appear on the Sandbox Vaults page if so. During maintenance windows, operations may take longer, but will be completed. Wait for the email notification. If the operation has not completed after several hours, review the Vault Notifications log for error details.

Sandbox Appears Unavailable After Creation

Sandbox Vaults may appear unavailable immediately after creation or during refresh while initialisation completes. Refresh the admin page after a few minutes. If the sandbox remains unavailable after 60–90 minutes, check the Vault Notifications log for error details.

Sandbox is Over Limit — Cannot Create Records or Documents

Navigate to Admin > Deployment > Sandbox Vaults and review the usage indicators for the affected sandbox. Delete unnecessary object records or document versions to bring usage below the size limit. After deletion, use the Recheck Usage function to confirm compliance and lift the creation block.

Configuration Cloning Blocked Error

If a 'configuration cloning blocked' message persists for more than 90 minutes, this may indicate a feature is being upgraded on your Vault as part of a platform enhancement. Contact Veeva Support with the error details and the timestamp of when the issue began.

VPK Deployment Blocked — Missing Dependencies

Review the Validation Log on the Inbound Packages page. Identify components listed as Missing – Block. Add the missing components to the source outbound package, re-export the VPK, and re-import to the target. Re-validate the updated package before deploying. Never deploy a package with blocking validation errors.

Cannot delete a Sandbox

If sandbox deletion fails, check whether the sandbox has linked file staging. Navigate to Admin > Settings > File Staging Server and remove the file staging linkage in the Vault Admin UI. Once unlinked, retry the deletion. No Veeva Support involvement is required for this operation.

Conclusion

Effective Veeva Vault environment management is a discipline that combines platform knowledge with sound operational process. The good news for Vault Administrators and IT Platform teams is that Veeva has built comprehensive self-service capabilities into the platform: creating, refreshing, snapshotting, and decommissioning sandbox environments is entirely within the control of your team, without requiring Veeva Support engagement for routine operations.

The organisations that get the most value from their Vault environment estate are those that treat environment management as a first-class operational discipline — with documented strategies, predictable refresh cadences, disciplined use of snapshots, and rigorous VPK-based deployment governance. These practices do not require large teams or complex tooling. They require consistency and intention.

Wolvio Solutions works with life sciences organisations globally to design, implement, and optimise their Veeva Vault platform operations. If your team would benefit from an environment management assessment, a sandbox strategy review, or hands-on support with your deployment pipeline, we would welcome the opportunity to discuss.

About Wolvio Solutions

Wolvio Solutions is a modern IT consulting and services company specializing in digital transformation across Life Sciences, Healthcare, and Regulated Industries. With deep expertise in Veeva Vault, Cloud Ops, and intelligent automation, we help organizations implement scalable, compliant, and user-friendly solutions that drive business efficiency and innovation.

Let’s Connect

Wolvio Solutions Private Limited

📧 contact@wolviosolutions.com

🌐 www.wolviosolutions.com

📍 Sholinganallur, Chennai, India

🔗 LinkedIn: https://www.linkedin.com/company/wolvio-solutions

Let’s collaborate to shape your digital transformation journey!

Copyright & Usage

🔗 *"When precision, compliance, and scalability matter—Wolvio delivers."*

WOLVIO