Keepit for okta

Coverage details

keepit

Contents

To ensure a smooth experience, restore job logic is designed in a way to prevent unwanted emails, avoid unusable accounts, and stay compliant with Okta's platform limitations.

User restore rules

The Okta connector restores every user in a working state (active or suspended).

Restore scenario	Status in snapshot	Behavior
Recreating user	Active	Set as active (active-like)*
	Provisioned	
	Recovery	
	Password expired	
	Suspended	Set as staged (negative-like)*
	Locked out	
	Staged	
	Deprovisioned	
Updating existing user	Active	Set as active (active-like)*
	Provisioned	
	Recovery	
	Password expired	
	Suspended	Set as suspended
	Locked out	Set as suspended (Okta can't set these states directly)
	Staged	
	Deprovisioned	Set as suspended**

* See below for an explanation of active-like vs negative-like

** Okta permanently deletes all user metadata when a user is marked as "Deprovisioned." To prevent data loss during updates to existing users, Keepit sets the status to "Suspended" instead of "Deprovisioned."

Active-like vs negative-like: Status categories explained

To simplify and explain our internal restore logic, we group user statuses into two conceptual categories: Active-like and Negative-like.

These categories are not part of official Okta terminology—they're used only within our internal context to make reasoning about restore behavior more intuitive.

- **Active-like:** includes statuses such as Active, Password Expired, Provisioned, or Recovery
- **Negative-like:** includes statuses such as Deprovisioned, Suspended, Locked out, or Staged

This classification helps clarify how the restore process should behave. Logically, Active-like statuses indicate users who are either currently usable or were recently in an active state. In contrast, Negative-like statuses represent users who were intentionally disabled or no longer active at the time of the snapshot.

For example, when recreating a user during restore, the handling may vary depending on whether their status falls into the Active-like or Negative-like category. This distinction allows us to tailor restore behavior in a way that respects the user's original state and intent.

Restored user status depends on status in snapshot
Users with an Active-like status are always recreated with the endpoint URL query parameter active=true. This typically results in the user's status being set to "Provisioned."

This behavior is necessary because user passwords are not included in the Okta response payload during backup - Okta does not expose them. Therefore, we are unable to restore the user's password.

As a result, even though we recreate the user with active=true, the status remains "Provisioned" instead of "Active," because the user has no password set.
If we were able to provide a password during user creation, the resulting status would be "Active."

Users with a Negative-like status are always recreated with the endpoint URL query parameter active=false. This typically results in the user's status being set to "Staged." Advancing these users beyond this status could trigger unwanted email notifications, so this behavior is intentionally avoided.

In conclusion, user status transitions beyond the initial restored state are not performed during the job in which the user is recreated.

Recreating a user

- If the user type exists: The user is recreated with the same user type as in the snapshot (e.g., contractor, employee).
- If the user type is missing: The user is recreated with Okta's default user type (e.g., Default).

- When restoring default user type from a snapshot, the user is recreated in Okta with that same type. However, the default user type may include custom properties marked as "required," which introduces a risk: the version may differ from the one in the snapshot. Currently, Keepit does not compare default user types or handle such differences. It does not disable or remove conflicting properties, which may result in errors during the restore process.

Updating an existing user

- If the snapshot's user type exists: The user's profile is updated.
- If the snapshot's user type is missing: The user's profile cannot be updated.
- If the snapshot's user type differs from the current one: Changing user types during an update is not supported.

User links

- Due to Okta's enforcement of uniqueness for role name combinations, a linked object definition can only be restored if the exact same pair of role names is not already used in another definition.
- If the required linked object definition does not exist in Okta, user relationships cannot be restored.
- A mismatch occurs if the role names differ in any way (e.g., name change, swapped roles, etc.). In such cases, the associated links cannot be restored.

Group restore rules

Definition-only restore

- When a group definition is selected for restore, only the group's basic properties (such as name, description, and profile attributes) are restored. In this case, app assignments, role assignments, and group members are not restored.
- When a group folder is selected for restore, both the group definition and its members are restored, but app assignments and role assignments are not restored.

Handling existing groups

- If a group with the same ID already exists in the target Okta environment and has a different modification time in the snapshot, the group's definition (such as name, description, and profile attributes) will be replaced based on the snapshot. This behavior follows the Okta API design, where updates fully replace the resource definition rather than applying partial changes.
- Group membership is not fully replaced – members present in the snapshot but missing in the target environment will be added, while any existing members not included in the snapshot will be left unchanged.

Missing required schema properties

If the group schema at the time of restore has required properties that were not present or populated in the snapshot, such properties will be temporary marked as optional during the restore. After the restore, they will be reverted back to required to maintain schema validity.

Removed custom schema properties

If custom properties existed in the snapshot but no longer exist in the current schema at the time of restore, these properties are simply ignored. They are excluded from the restore payload and will not be restored.

Application restore rules

Application users

- Individual user-to-app assignments take priority over those inherited from group-to-app assignments.
- The application must exist in Okta for user assignments to be restored.
- If the entire application is included in the restore scope:
    - If the application definition is included in the restore scope (i.e., the whole application is restored), its status in the snapshot must be ACTIVE.
- If only the application users folder is selected for restore:
    - The application status must be "active" in Okta.
- The application user schema may define required properties that are either missing or unpopulated in the snapshot.
    - In such cases, these properties are temporarily marked as optional during restore and are reset to required upon completion.
- The schema may contain custom properties that were present during the snapshot but deleted by the time of restore.
    - These properties are removed from the restore payload and not restored.
- When restoring individual application users or the entire application users folder, only users who were assigned individually to the application are restored.
- Group-to-app assignments are restored only when the corresponding application groups are restored.

Application groups

- The application must exist in Okta for group-related restore actions to proceed.
- If the entire application is included in the restore scope:
    - The application's status in the snapshot must be "active."
- If only the application groups folder is selected:
    - The application's status in Okta must be "active" at the time of restore.
- The application’s user schema may contain required properties that either did not exist at the time of the snapshot or were not populated.

    - In such cases, these properties will be temporarily marked as optional during restore and set back to required once the application restore completes.
- New custom properties may have existed in the user schema at the time of the snapshot but been deleted by the time of restore.
    - When this happens, any data for those non-existent properties is removed from the payload and not restored.
- Groups are re-assigned to the application with their original priority preserved.
- User-app assignments that were derived from group-app assignments are restored with overwritten property values.

Definition.json

- When selected for restore, only the definition of the application is restored. Groups, members, schema, grants, provisioning, client settings, and SSO SAML metadata will not be restored.
- If the application already exists in Okta and differs from the snapshot (based on modification time), it will be replaced rather than updated.
- The payload does not include a node for the Callback URI from the Email Verification Experience. This field is present only in OIDC applications:

App user schema

- The app user schema cannot exist without an associated application — it cannot be created independently.
- The application must be active during the restore. Otherwise, the operation will fail.
- The last updated timestamp does not change when updating the schema's title or description.
- It is not possible to update the data type of an existing profile schema attribute (e.g., from string to integer).
- The app user schema can only be restored when the provisioning state matches the backup and the target (remote) environment.
  If the provisioning states differ, the restore will fail.
- If the application does not exist in the target environment, provisioning will not be restored. Therefore, if the backup application had provisioning enabled, the app user schema restore will fail.

Linked objects

- Okta does not provide unique IDs for linked object definitions. Instead, each definition is identified by the combination of its primary and associated role names (e.g., Manager_Worker).
- A linked object definition is restorable only if both role names (primary and associated) from the snapshot do not exist in any form (either as primary or associated) within any linked object definition in Okta. If either the primary role name or the associated role name already exists in Okta (even in a different relation or reversed order), the definition restore will be skipped, because existing names cannot be reused.
- Okta API does not support updating existing linked object definitions, therefore:
    - There is no logic for updating existing definitions.
    - Restoration only occurs if the definition is completely missing and the role names are not used.

API service integrations

- API service integrations are immutable objects. They can only be created, not updated.
- Secrets associated with them can have their status updated (e.g., activated or deactivated), but their value cannot be changed.
- At least one active secret must exist for the integration to remain valid.

Profile mappings

- During restoration, existing profile attributes are not overwritten if they are missing in the snapshot.
- Mappings for properties that no longer exist in the schema are skipped and not restored.

Groups rules

- Group rules are restored in a deactivated state by default to prevent unintended automatic assignments.
- Only rules with valid Okta expression language conditions and existing target groups are restored.
- Users who were assigned through group rules are restored as static members; these must be manually removed if not desired.