



K2cloud Roles and Identity Mapping

Technical Note

Date: 19 September 2025

Version: 1.0

Abstract

This technical note explains how K2cloud implements Role-Based Access Control (RBAC) across projects, spaces, and Test Data Management (TDM) environments. It describes how customer-managed identity providers (IDPs) work with the K2cloud CyberArk federation service to enforce consistent authentication and authorization. The document details the layered approach to access control at the cloud and space levels, the process of mapping IDP groups to Fabric roles, and the creation and assignment of roles and permissions.

Practical usage scenarios show how access can be limited to specific areas and how deployment permissions are assigned to authorized groups, ensuring least-privilege access and separation of duties. The note also emphasizes TDM-specific roles and permissions, which expand RBAC controls to data provisioning, masking, and synthetic data generation workflows.

Together, these guidelines offer a framework for customers to align K2cloud access management with enterprise identity policies while maintaining flexibility and security across environments.

This document contains copyrighted work and proprietary information belonging to K2view. All intellectual property rights (including, without limitation, copyrights, trade secrets, trademarks, etc.) evidenced by or embodied in and/or attached, connected, or related to this document, as well as any information contained herein, are and shall be owned solely by K2view. K2view does not convey to you an interest in or to this document, the information contained herein, or its intellectual property rights, but only a personal, limited, fully revocable right to use the document solely for review purposes. Unless explicitly set forth otherwise, you may not reproduce by any means any document and/or copyright contained herein.

Information in this document is subject to change without notice. Corporate and individual names and data used in the examples herein are fictitious unless otherwise noted.

Copyright © 2025 K2view Ltd. / K2VIEW LLC. All rights reserved.

The following are trademarks of K2view: K2view logo, K2view's platform.

K2view reserves the right to update this list document from time to time.

Table of Contents

1.	Introduction	3
2.	Cloud and Space-Level Access Control	3
2.1.	Cloud-Level (Project Scope)	3
2.2.	Space-Level	3
3.	Delegated Authentication and Group Mapping	4
3.1.	Customer-Managed Roles	4
3.2.	Delegated Authentication	4
3.3.	Mapping Security Groups to Fabric Roles	4
3.4.	Key Concepts	4
3.5.	How It Works	5
4.	Role Creation and Permission Assignment	5
5.	Usage Scenarios	6
5.1.	Controlling Access to a Space	6
5.2.	Deployment Authorization Workflow	7
6.	TDM-Specific Access Control	8
6.1.	Roles and Permissions	8
6.2.	TDM Permissions Examples	8
6.3.	Integration with Fabric Roles	8

1. Introduction

K2view employs a Role-Based Access Control (RBAC) model to govern access across its cloud platform, Fabric environments, and TDM (Test Data Management) functionality. Access control is enforced through customer-managed identity providers (IDPs) using delegated authentication via the K2cloud CyberArk Identity Federation service. This ensures that user identities and group memberships defined in the enterprise directory are consistently mapped to Fabric roles and permissions.

2. Cloud and Space-Level Access Control

Access within K2cloud is organized at two levels: the **project (cloud)** level and the **space** level.

At the **cloud level**, Project Managers are assigned the **Cloud User** role. This role provides full CRUD (Create, Read, Update, Delete) privileges across projects and spaces, including the ability to manage project space profiles and assign roles to users. When a Project Manager creates a new space, they are automatically designated as its **Space Admin**, effectively becoming the owner of that space.

At the **space level**, the **Space Admin** has complete control within their space, including configuration, role assignment, and inviting collaborators. This separates space-specific administration from broader project management responsibilities.

For end users, the predefined **Space User** role provides access to the space without administrative privileges. The permissions of this role are determined by the Space Admin, based on Fabric's permission set. Additional custom roles can also be defined within a space to support more granular access models based on the [list of Fabric permissions](#).

This structure ensures clear separation of responsibilities: Cloud Users govern projects, Space Admins manage individual spaces, and Space Users (or custom roles) operate with least-privilege access.

2.1. Cloud-Level (Project Scope)

At the cloud level, access is governed by the **Cloud User** role. Users with this role have full administrative privileges, including the ability to create and manage projects and the spaces within them. Because this role effectively grants unrestricted control across the environment, it should be assigned only to **trusted project managers** or equivalent personnel with broad operational responsibility.

2.2. Space-Level

Each Fabric space is created with a designated **Space Admin**, who assumes ownership of that space. The Space Admin is responsible for granting access to additional users by mapping IDP-managed security groups either to the predefined **Space User** role or to custom roles defined within the space.

Permissions assigned at this level are **scoped exclusively to the individual space**, ensuring isolation between environments. Importantly, access to a space is available only through its unique URL, reinforcing separation of duties and limiting visibility beyond the space to which a user has been explicitly granted access.

3. Delegated Authentication and Group Mapping

3.1. Customer-Managed Roles

Cloud and Space users can be governed by the customer's IDP, which assigns users to security groups (e.g., within Microsoft Entra ID). K2cloud – via identity federation – maps security groups to corresponding Fabric or K2cloud roles.

3.2. Delegated Authentication

K2view enables Single Sign-On (SSO) integration, simplifying the login process while maintaining secure access across different K2view Spaces. Once integrated with the K2cloud CyberArk service, all authentication is delegated to the customer's IDP, including the management of security groups within the IDP.

Users can access K2view Spaces and Projects only based on their assigned roles and permissions, ensuring that access is strictly controlled and aligned with their responsibilities. This is accomplished by mapping IDP security groups to Fabric roles through the K2cloud CyberArk federation service.

Space administrators are responsible for creating roles within the space and assigning permissions to them. Newly created space roles need to be created within the customer's IDP and then mapped by K2view in the K2cloud CyberArk service to allow the IDP's role to be used in Fabric authorization.

Security groups managed by the IDP assign users to the privileges associated with the groups it manages. These groups are mapped to Fabric roles by K2view's CyberArk to the corresponding Fabric roles.

3.3. Mapping Security Groups to Fabric Roles

Fabric does not assign permissions directly to individual users. Instead, it relies on **group-to-role mapping** to enforce access consistently across large organizations.

The mapping follows a layered model:

IDP Security Group → K2cloud CyberArk Group → Fabric Role

- **IDP Security Groups** are created and managed by the customer in their identity provider (e.g., Entra ID, Okta, Ping).
- **K2cloud CyberArk Groups** are created and maintained by K2view. They act as the bridge, linking customer-managed security groups to Fabric roles.
- **Fabric Roles** are defined by the customer in Fabric and represent specific sets of permissions. These roles determine what actions a user can perform (e.g., deploy, read/write to web services, manage spaces).

By using this mapping approach, customers maintain control over **who belongs to which group**, while K2view ensures that group memberships translate into appropriate Fabric permissions. This model provides both flexibility and governance, aligning with enterprise RBAC policies.

3.4. Key Concepts

To support delegated authentication and group mapping, several core concepts must be understood:

- **Identity Provider (IDP):** The customer's directory system (e.g., Microsoft Entra ID, Okta, Ping, or any SAML-compliant provider). The IDP manages user accounts and their group memberships, acting as the source of truth for enterprise identity.

- **K2cloud CyberArk Service:** A federation layer operated by K2view. It receives authentication assertions from the IDP, maps group memberships to Fabric roles, and enforces identity across K2cloud services. It ensures that role assignments are consistent and centrally managed.
- **Fabric Roles:** Logical roles defined within Fabric that represent sets of permissions (e.g., deployment, web service access, space administration). These roles are assigned to users indirectly through IDP group mapping, never by direct user-to-role assignment.

3.5. How It Works

The overall workflow is as follows:

1. A user signs in via SSO using the customer's IDP.
2. The user's group memberships are included in the authentication assertion passed to K2cloud CyberArk.
3. K2cloud CyberArk maps those groups to Fabric Roles as defined in its configuration.
4. Fabric enforces access and permissions based on the mapped role(s) when the user interacts with a space or project.

This workflow ensures that authentication and authorization remain tightly coupled to the customer's enterprise identity system, while Fabric enforces permissions consistently at runtime.

4. Role Creation and Permission Assignment

Role-based access control in K2cloud requires coordination across three layers: **Fabric**, **K2cloud CyberArk**, and the **customer's identity provider (IDP)**. Each plays a distinct role in defining, mapping, and enforcing [permissions](#).

In Fabric

Roles are created through the Fabric Admin Web UI. For example, a customer might create a role like `_k2v_deployers` to assign deployment privileges. Each role receives [permissions](#) from the standard Fabric list (such as `Deploy`, `ALL_WS`, or more specific permissions). Scopes can also be applied to limit access to particular web services or LUTs, enabling detailed control.

In K2cloud CyberArk

K2cloud personnel configure the mapping between customer-managed IDP groups and Fabric roles. CyberArk acts as the federation layer, ensuring that a user's group membership in the IDP is consistently translated into the correct Fabric role during login.

In the IDP

The customer creates and manages IDP security groups that align with the organization's structure (e.g., developers, testers, deployment engineers). These groups are the source of truth for who should have which privileges. For IDPs such as Microsoft Entra ID, group GUIDs are also shared with K2view to complete the mapping process.

Key Point: Role creation is a **joint responsibility**. The customer defines the roles and groups, K2cloud CyberArk ensures proper mapping, and Fabric enforces permissions. This layered approach ensures that access control remains consistent with enterprise RBAC policies while allowing fine-grained operational control within Fabric.

5. Usage Scenarios

5.1. Controlling Access to a Space

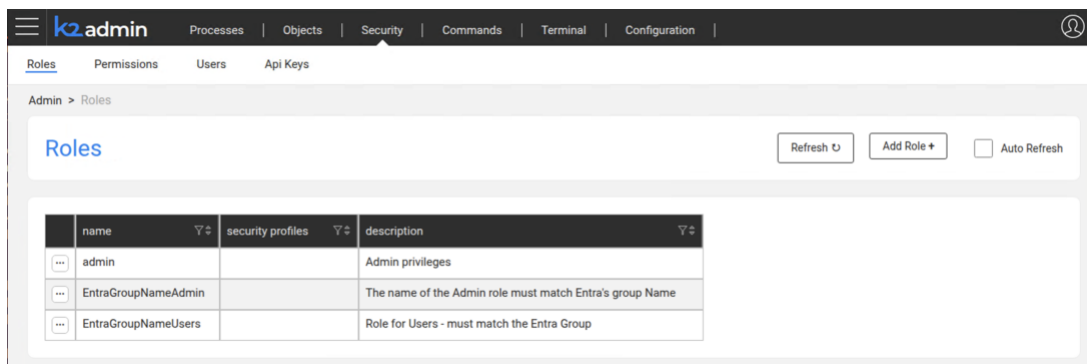
Use Case: Restricting certain users or roles from accessing specific spaces.

K2cloud's RBAC model supports this. However, it is essential not to assign the **Cloud User** role unless specifically necessary. Giving this role effectively promotes the user to a Project Manager, granting administrative access to all project spaces.

Instead, access should be managed at the **space level** by the Space Admin. Users who need access to a specific space should be assigned to an IDP-managed security group that is mapped, through CyberArk, to a corresponding **Space User** role or other space-specific roles created by the Space Admin.

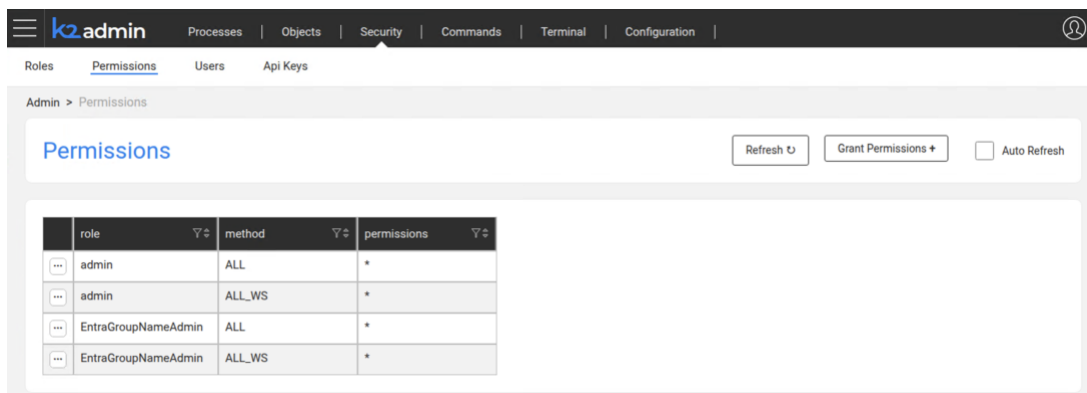
Implementation Example:

- Create an IDP group such as EntraGroupNameAdmin for administrators and EntraGroupNameUsers for standard users.
- Map these groups to Fabric roles with appropriate permissions (e.g., ALL and ALL_WS for administrative access).
- Restrict test users by not adding them to the group governing access to development spaces.



The screenshot shows the 'Roles' page in the K2admin interface. The page has a header with navigation links: Roles, Permissions, Users, and Api Keys. Below the header, there is a 'Roles' section with a table listing roles. The table has columns: name, security profiles, and description. The roles listed are 'admin', 'EntraGroupNameAdmin', and 'EntraGroupNameUsers'.

name	security profiles	description
admin		Admin privileges
EntraGroupNameAdmin		The name of the Admin role must match Entra's group Name
EntraGroupNameUsers		Role for Users - must match the Entra Group



The screenshot shows the 'Permissions' page in the K2admin interface. The page has a header with navigation links: Roles, Permissions, Users, and Api Keys. Below the header, there is a 'Permissions' section with a table listing permissions. The table has columns: role, method, and permissions. The permissions listed are 'admin', 'admin', 'EntraGroupNameAdmin', and 'EntraGroupNameAdmin'.

role	method	permissions
admin	ALL	*
admin	ALL_WS	*
EntraGroupNameAdmin	ALL	*
EntraGroupNameAdmin	ALL_WS	*

Key Point: Users should only gain access to spaces through the space's URL, and access should be scoped by the roles defined at the space level. This ensures least-privilege access while preserving administrative control.

5.2. Deployment Authorization Workflow

Use Case: Assign deployment permissions only to individuals directly involved in the SDLC, ensuring controlled and auditable changes across environments.

Fabric roles can be tailored with specific permissions, such as those listed in the [Fabric Credentials Overview](#). A key permission is **Deploy**, which grants the ability to run the deploy command on either a project or an entire environment.

Ultimately, this requires creating a Fabric Role with specific permissions. While Fabric Roles define the authorization, user assignment to these roles is managed outside Fabric by the customer's identity system. This ensures centralized governance and alignment with enterprise RBAC policies.

Workflow:

1. The customer defines a **deployment group** (e.g., *Deployers*) in the customer's IDP.
2. A Fabric Role is created with the Deploy permission.
3. K2cloud CyberArk maps the IDP group to the Fabric Role.
4. Members of this group can now:
 - Deploy from Fabric Studio.
 - Execute deployments via API or CI/CD pipelines.
 - Deploy only to allowed environments based on scope.
 - Deploy only to the environments allowed by the role's scope.

Key Point: Deployment authorization ensures separation of duties and prevents unauthorized promotion of code or data into controlled environments.

6. TDM-Specific Access Control

TDM (Test Data Management) extends the Fabric RBAC framework with permissions tailored to managing test data provisioning, masking, and synthetic generation. While authorization remains enforced through Fabric roles and customer-managed IDP groups, TDM adds a layer of granularity specific to test data environments.

6.1. Roles and Permissions

TDM introduces role categories within a Fabric space that align with typical responsibilities in test data operations:

- **TDM Admin:** Has full control of TDM configurations, including creation of environments, assignment of roles, definition of masking/synthetic rules, and approval of provisioning workflows.
- **Environment Owner:** Responsible for a specific test environment. This role manages environment variables, data provisioning rules, and lifecycle operations (e.g., refresh, reset).
- **Tester:** Limited to executing provisioning, masking, or synthetic data generation tasks within the environments they are assigned. They cannot alter global TDM settings or access environments outside their scope.

These roles inherit their base access from Fabric roles but introduce TDM-specific permissions to constrain operations within the TDM portal.

6.2. TDM Permissions Examples

Examples of how permissions can be scoped within TDM include:

- Executing **test data masking tasks** to anonymize sensitive fields.
- **Provisioning test data** into designated sandbox environments without impacting production or other spaces.
- Managing **synthetic data generation jobs** (rule-based or AI-driven) limited to approved environments.
- Restricting access to **sensitive datasets** so that only TDM Admins or Environment Owners can configure masking or cloning strategies.
- Controlling **environment refresh operations**, ensuring only Environment Owners or Admins can reset data to a baseline state.

6.3. Integration with Fabric Roles

TDM access is not isolated from Fabric. Instead, it builds upon Fabric roles defined and mapped via the IDP and K2cloud CyberArk federation service. For example:

- An IDP group such as TDM_Provisioners may be mapped to a Fabric role with restricted TDM permissions.
- A group TDM_Admin may be mapped to a Fabric role granting full TDM access.
- Environment-scoped roles ensure that testers can only execute tasks within specific spaces or datasets.

This layered approach ensures that TDM environments remain consistent with enterprise RBAC, while addressing the specialized requirements of test data provisioning and governance.