Folder access control

- Before you begin
- Default permissions for new folders
- How folder permissions work
  - Permission levels
    - Detailed action breakdown
  - Permission inheritance
- Resources that use folder permissions
  - Plugin permissions
- Manage folder permissions
  - View current permissions
  - Add a permission
  - Change a permission
  - Remove a permission
- Design your folder structure to manage permissions
- Related documentation

Folders are the primary way to organize and control access to resources in Grafana. When you assign permissions to a folder, those permissions apply to all resources within that folder, including dashboards, alert rules, SLOs, and more.

This page explains how folder permissions work and how to use them effectively to manage access for teams and users.

Before you begin

- You need Admin permission on a folder to manage its permissions
- To create folders, you need the Folder Creator role or appropriate organization permissions
- Folder permissions are available in all Grafana editions (OSS, Enterprise, and Cloud)

{{< admonition type="caution" >}} **Folder limitations:**

- Folders can be nested up to **4 levels deep**
- Folder names cannot contain underscores (`_`) or percent signs (`%`) {{< /admonition >}}

Default permissions for new folders

How a folder is created determines its initial permissions:

- **Folders created in the UI** are automatically granted: Admin role gets Admin permission, Editor role gets Edit permission, Viewer role gets View permission
- **Folders created via as-code** (Terraform, API, provisioning) only have permissions explicitly defined in the configuration, plus Admin role gets Admin permission by default

How folder permissions work

Folder permissions follow a simple principle: **a user's effective access to a resource is determined by their folder permission level**.

When you grant a user or team permission on a folder:

- The permission applies to the folder itself
- The permission cascades to all subfolders
- The permission applies to all resources in those folders (dashboards, alert rules, etc.)

Permission levels

Folders support three permission levels. Each level includes all capabilities of the levels below it.

Permission	Folder capabilities	Resource capabilities
View	View folder and navigate its contents	View dashboards, read alert rules, read silences, read annotations, read library panels
Edit	Create subfolders, modify folder properties, delete folder	Create and edit dashboards, create and write alert rules, create and write silences, create and write annotations, create and manage library panels

Permission	Folder capabilities	Resource capabilities
Admin	Manage folder permissions	All Edit capabilities plus manage dashboard permissions

Detailed action breakdown

View permission grants:

- `folders:read` - View the folder in navigation
- `dashboards:read` - View dashboards in the folder
- `alert.rules:read` - View alert rules in the folder
- `alert.silences:read` - View alert silences
- `annotations:read` - View annotations
- `library.panels:read` - View library panels

Edit permission grants (in addition to View):

- `folders:write` - Modify folder name and properties
- `folders:delete` - Delete the folder
- `folders:create` - Create subfolders
- `dashboards:create` - Create new dashboards
- `dashboards:write` - Edit existing dashboards
- `dashboards:delete` - Delete dashboards
- `alert.rules:create` - Create new alert rules
- `alert.rules:write` - Edit alert rules
- `alert.rules:delete` - Delete alert rules
- `alert.silences:create` - Create silences
- `alert.silences:write` - Edit silences
- `annotations:create` - Create annotations
- `annotations:write` - Edit annotations
- `annotations:delete` - Delete annotations
- `library.panels:create` - Create library panels
- `library.panels:write` - Edit library panels
- `library.panels:delete` - Delete library panels

Admin permission grants (in addition to Edit):

- `folders.permissions:read` - View folder permissions
- `folders.permissions:write` - Modify folder permissions
- `dashboards.permissions:read` - View dashboard permissions
- `dashboards.permissions:write` - Modify dashboard permissions

Permission inheritance

Permissions cascade from parent folders to child folders. A user with Edit permission on a parent folder automatically has Edit permission on all subfolders.

```
Production/                  # User has Edit permission here
├── Team A Dashboards/       # User inherits Edit permission
│   └── Service Metrics/     # User inherits Edit permission
└── Team B Dashboards/       # User inherits Edit permission
```

{{< admonition type="note" >}} You cannot grant a user lower permissions on a subfolder than they have on a parent folder. Permission inheritance always flows downward. {{< /admonition >}}

Resources that use folder permissions

The following resources are stored in folders and respect folder permissions:

Resource	How folder permissions apply
Dashboards	View/Edit/Admin directly controlled by folder permission
Subfolders	Inherit parent folder permissions, can add additional permissions
Alert rules	View/Edit controlled by folder permission
Alert silences	View/Create/Edit controlled by folder permission
Library panels	View/Create/Edit/Delete controlled by folder permission
SLOs	Folder permission + SLO role determines effective access (refer to Plugin permissions)
Annotations	Stored on dashboards; inherit permissions from the dashboard's parent folder

Plugin permissions

Some Grafana Cloud plugins use a two-layer permission model where effective access is determined by both folder permissions and a plugin-specific role:

```
Effective plugin access = minimum(folder permission, plugin role)
```

For example, with SLOs:

- A user with folder **Edit** permission and **SLO Writer** role can create and edit SLOs in that folder
- A user with folder **View** permission and **SLO Writer** role can only view SLOs (limited by folder)
- A user with folder **Edit** permission and **SLO Reader** role can only view SLOs (limited by plugin role)

For more information, refer to SLO access control.

Manage folder permissions

Folders are accessed through the **Dashboards** section in Grafana. You can manage permissions from the folder's context menu or from within the folder view.

View current permissions

1. In the left navigation, click **Dashboards**.
2. Navigate to the folder you want to inspect.
3. Click the folder name to open it, or click the ⋮ menu next to the folder.
4. Select **Manage permissions**.

The permissions dialog shows all users, teams, and roles with access to this folder.

Add a permission

1. Open the folder's **Manage permissions** dialog.
2. Click **Add a permission**.
3. Select who to grant access to:
   - **User** - A specific user account
   - **Team** - All members of a team
   - **Service Account** - A service account for API or automation access
   - **Role** - Users with a specific organization role (Viewer, Editor, Admin)
4. Select the permission level (View, Edit, or Admin).
5. Click **Save**.

Change a permission

1. Open the folder's **Manage permissions** dialog.
2. Find the user, team, or role in the list.
3. Use the permission dropdown to select a new level.

4. The change saves automatically.

Remove a permission

1. Open the folder's **Manage permissions** dialog.
2. Find the user, team, or role in the list.
3. Click the ✕ button to remove their access.
4. Confirm the removal.

{{< admonition type="warning" >}} Removing a permission removes access to the folder and all its contents. The user or team will no longer see dashboards, alert rules, or other resources in that folder. {{< /admonition >}}

Design your folder structure to manage permissions

A well-designed folder structure makes permission management simpler. The recommended pattern is to create top-level folders for each team, with subfolders for organizing content:

```
SRE Team/                    # SRE team has Admin
├── Production Monitoring/   # Inherited Admin
├── On-Call Dashboards/      # Inherited Admin
└── Runbooks/                # Inherited Admin

Platform Team/               # Platform team has Admin
├── Infrastructure/          # Inherited Admin
└── Cost Tracking/           # Inherited Admin

Shared/                      # Everyone has View, specific teams have Edit
├── Company KPIs/            # Marketing team has Edit
└── Executive Dashboards/    # Leadership has View
```

This pattern provides:

- **Team ownership**: Each team manages their own folder and contents
- **Clear boundaries**: Resources are organized by responsibility
- **Collaboration**: Shared folders enable cross-team visibility

For detailed guidance on folder organization patterns, including automation with Terraform and SCIM provisioning, refer to:

- [Configure multi-team access]({{< relref "../../../setup-grafana/configure-access/multi-team-access" >}})
- Managing access in Grafana: A single-stack journey with teams, roles, and real-world patterns

Related documentation

- [Manage dashboard permissions]({{< relref "../../user-management/manage-dashboard-permissions" >}}) - Dashboard-level permission management
- [Configure multi-team access]({{< relref "../../../setup-grafana/configure-access/multi-team-access" >}}) - Recommended folder structures for organizations
- [RBAC fixed roles]({{< relref "./access-control/rbac-fixed-basic-role-definitions" >}}) - Enterprise folder-related RBAC roles
- [Alerting folder permissions]({{< relref "../../../alerting/set-up/configure-rbac/access-folders" >}}) - How folders control alert rule access
- [Team management]({{< relref "../../team-management" >}}) - Create and manage teams for folder access

---

□□□ #2
□ □□□ □□□ 2026-10-31 15:37:49 UTC
□ □□□ □□□ 2026-10-31 15:37:51 UTC