Security and Authentication

Overview

GitLab Security

Git CLI

- HTTPS-based
- SSH-based

Personal access tokens

Manage users

Securing code contributions

- Code scans
- Secrets

Modify Git history

HTTPS vs SSH

Communication Protocol

Git uses HTTPS or SSH protocol to connect with repositories

HTTPS is a go to protocol for many public open-source projects

SSH is the secure and private alternative to HTTPS

Why SSH?

Secure

No need to use the username and password

Private

Each SSH key is stored on your own device

Cross-platform

SSH libraries are available on all operating systems

Create an SSH Key

Demo

Create an SSH key

Read the key values

Setup SSH Key in GitLab

Demo

Configure SSH key in GitLab

Demo

Push a commit using SSH

Personal Access Tokens

Personal Access Tokens

Secure tokens that are alternative to your password

Access can be granted based on what the token needs to do

Tokens can be rotated at any time in an account

Demo

Create a Personal Access Token

Use an Access Token with Git CLI

Demo

Use Personal Access Token

Manage Access and Users

Access Management

Integrated

User management is available with GitLab out of the box

Groups

Manage your users as a group with role-based permissions

Inherited

Everyone in the group can access the projects defined in a group

Inviting Users

GitLab Users
Invite members using their GitLab username or their email address

Permissions
GitLab uses role-based permissions

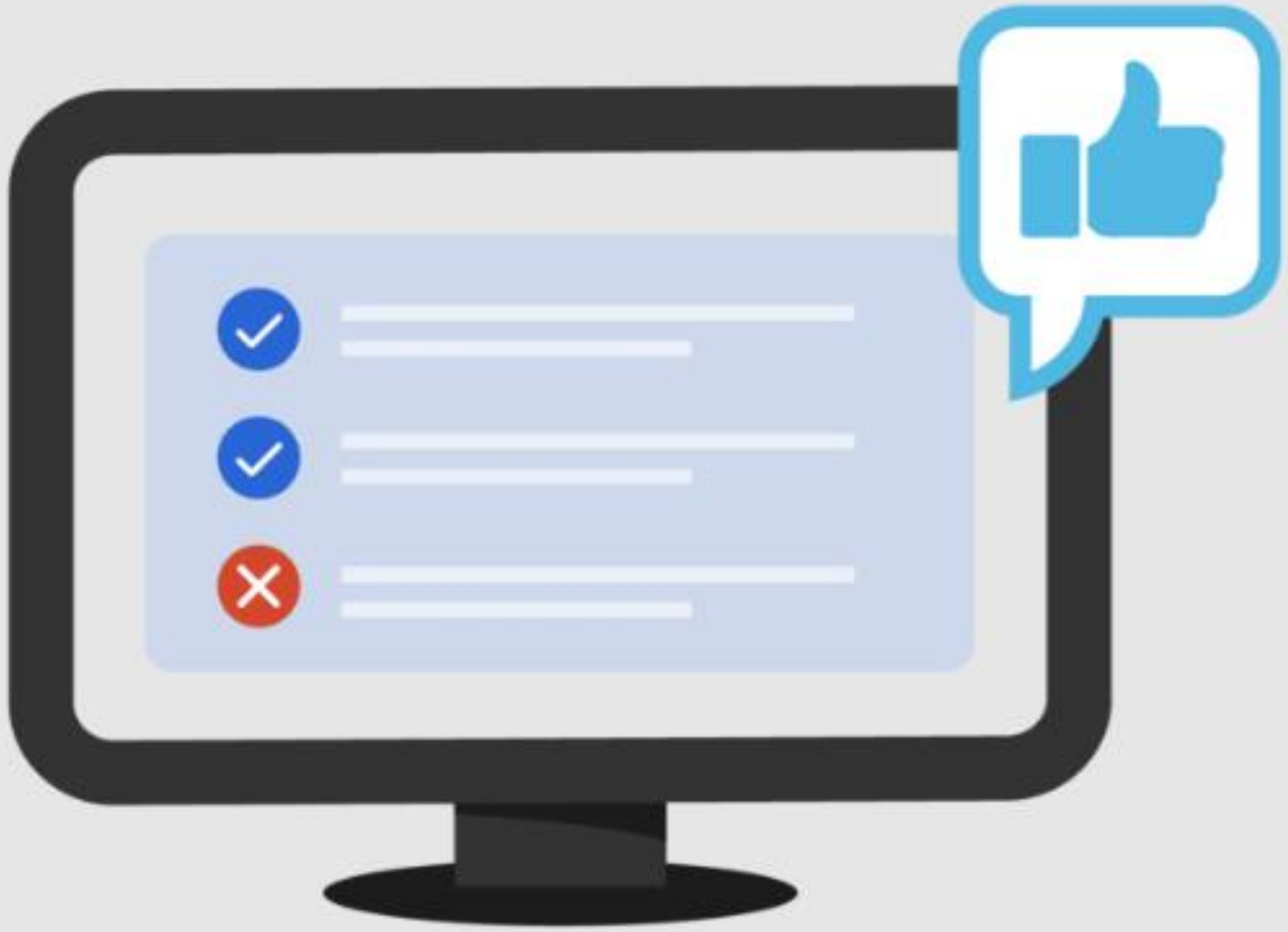
Expiration
Access can be automatically revoked after an expiry date

Authentication in Gitlab

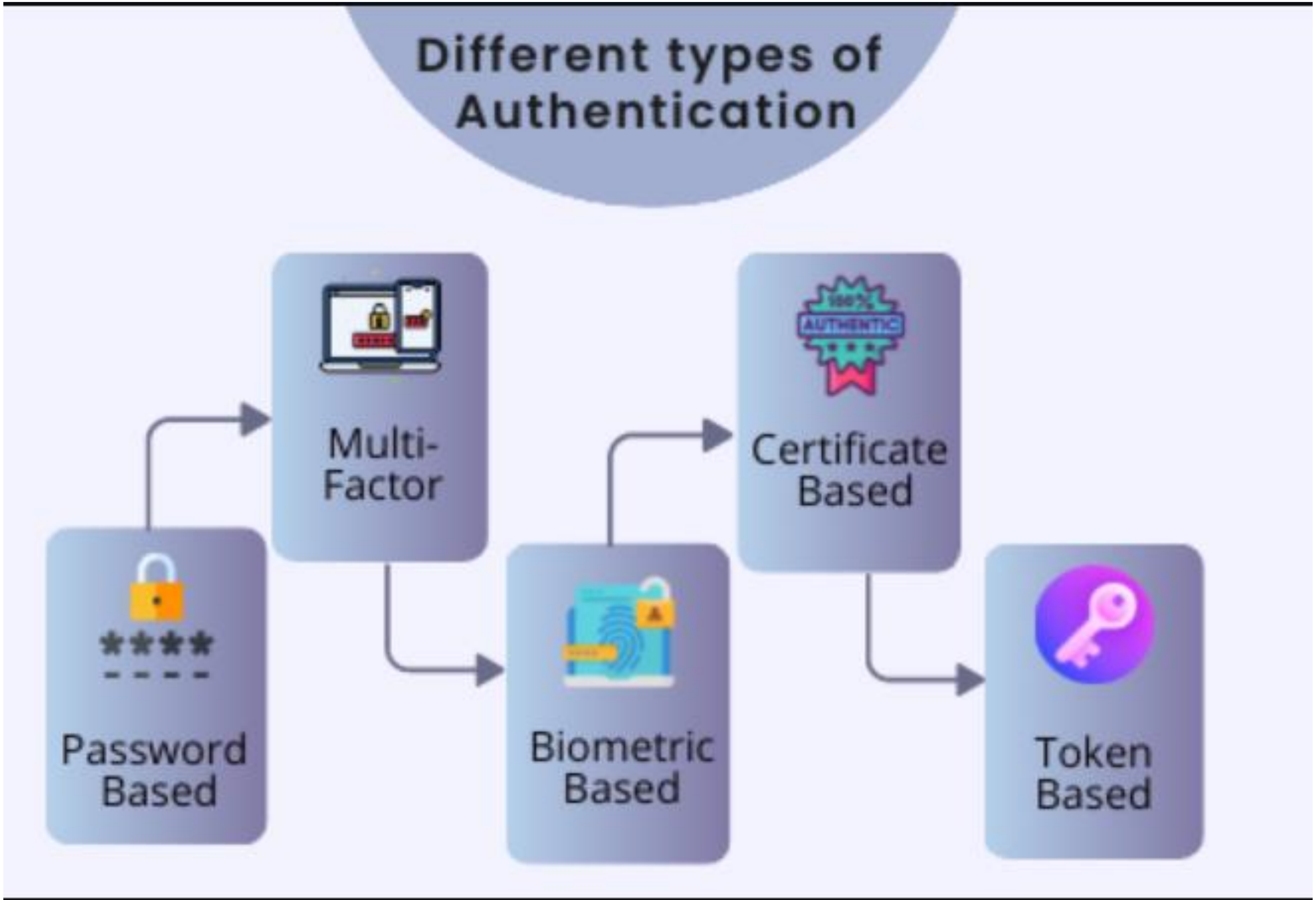
Authentication

Confirms users
are who they say they are.

Authorization



Gives users permission
to access a resource.



Different types of Authentication

Password Based

Multi-Factor

Biometric Based

Certificate Based

Token Based

Authentication: Gitlab Users & Administrator

1. Username & Password (Built-in)
2. SSH
3. LDAP
4. Two-factor authentication (2FA)
5. GitLab as OAuth2 authentication service provider
6. GitLab as OpenID Connect identity provider

Authentication: Gitlab Users & Administrator

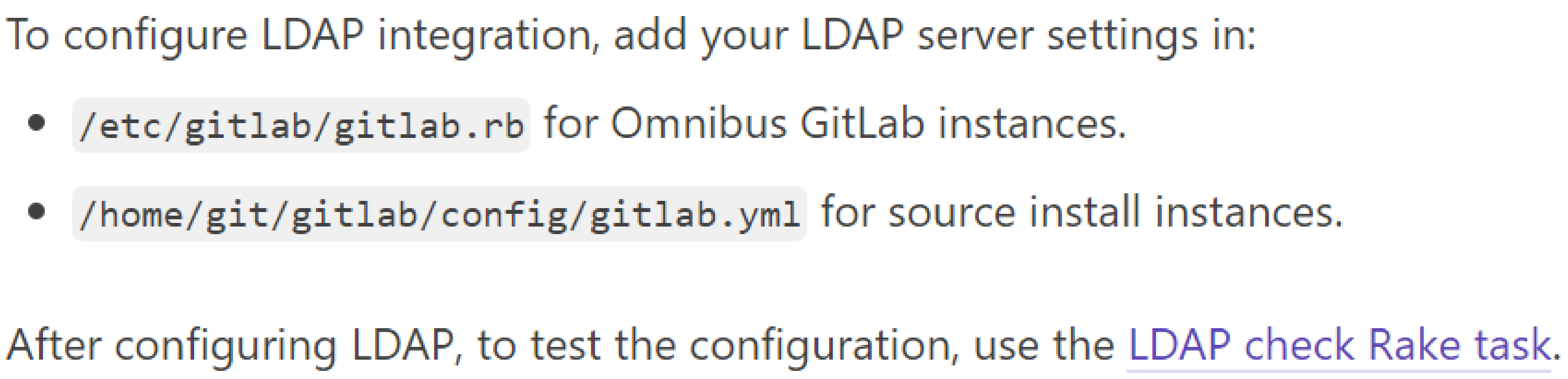
Integrations:

1. OmniAuth
2. Authentiq OmniAuth Provider
3. Atlassian Crowd OmniAuth Provider
4. CAS OmniAuth Provider
5. SAML OmniAuth Provider
6. SAML for GitLab.com Groups
7. SCIM user provisioning for GitLab.com Groups
8. Kerberos integration (GitLab EE)

Authentication: API

- OAuth 2 Tokens
- Personal access tokens
- Project access tokens
- Group access tokens
- Impersonation tokens
- OAuth 2.0 identity provider API

Authentication: Configure LDAP



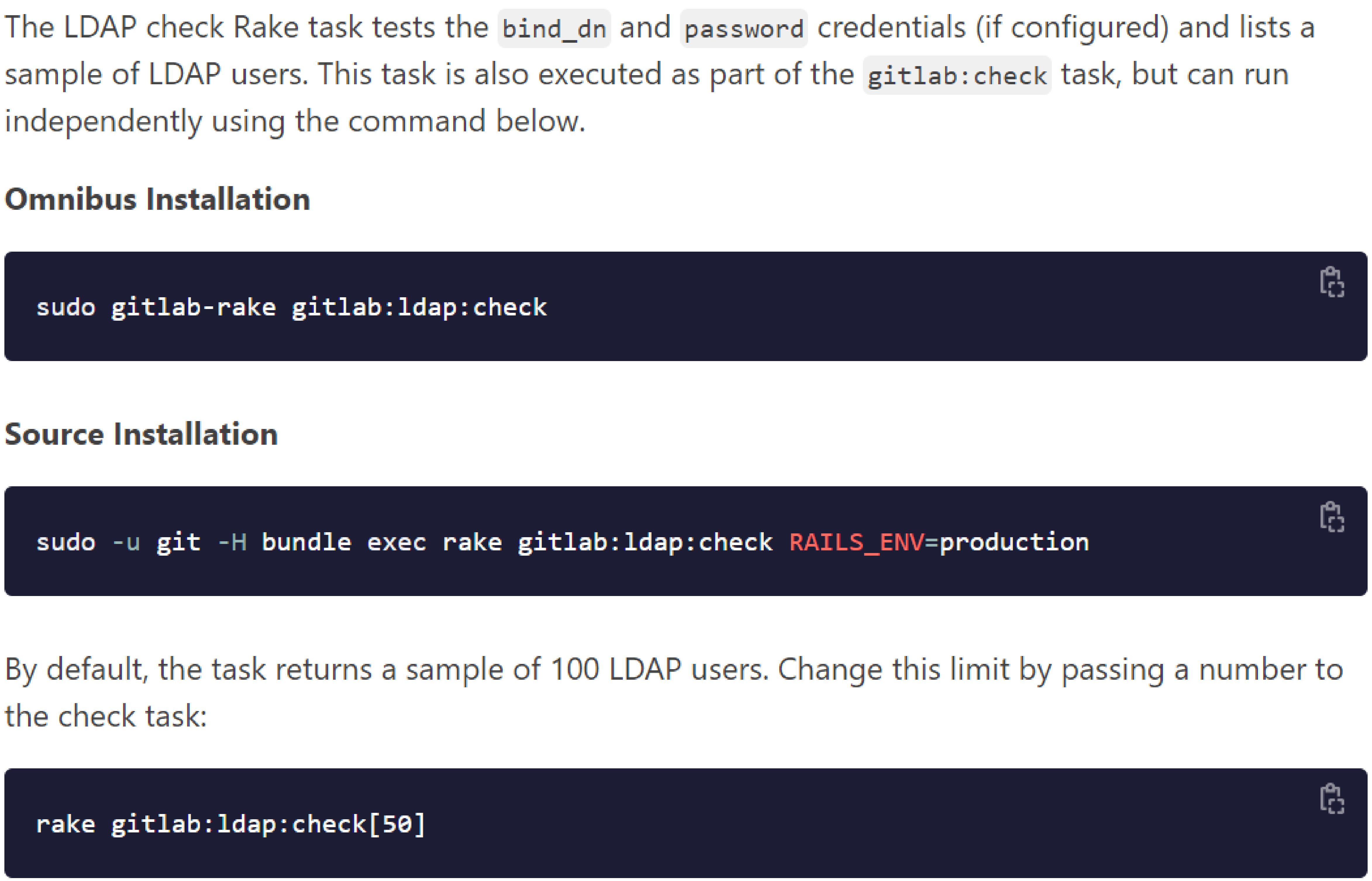
To configure LDAP integration, add your LDAP server settings in:

- `/etc/gitlab/gitlab.rb` for Omnibus GitLab instances.
- `/home/git/gitlab/config/gitlab.yml` for source install instances.

After configuring LDAP, to test the configuration, use the LDAP check Rake task.

https://docs.gitlab.com/ee/administration/auth/ldap/index.html

Authentication: LDAP Rake tasks



The LDAP check Rake task tests the `bind_dn` and `password` credentials (if configured) and lists a sample of LDAP users. This task is also executed as part of the `gitlab:check` task, but can run independently using the command below.

Omnibus Installation

```
sudo gitlab-rake gitlab:ldap:check
```

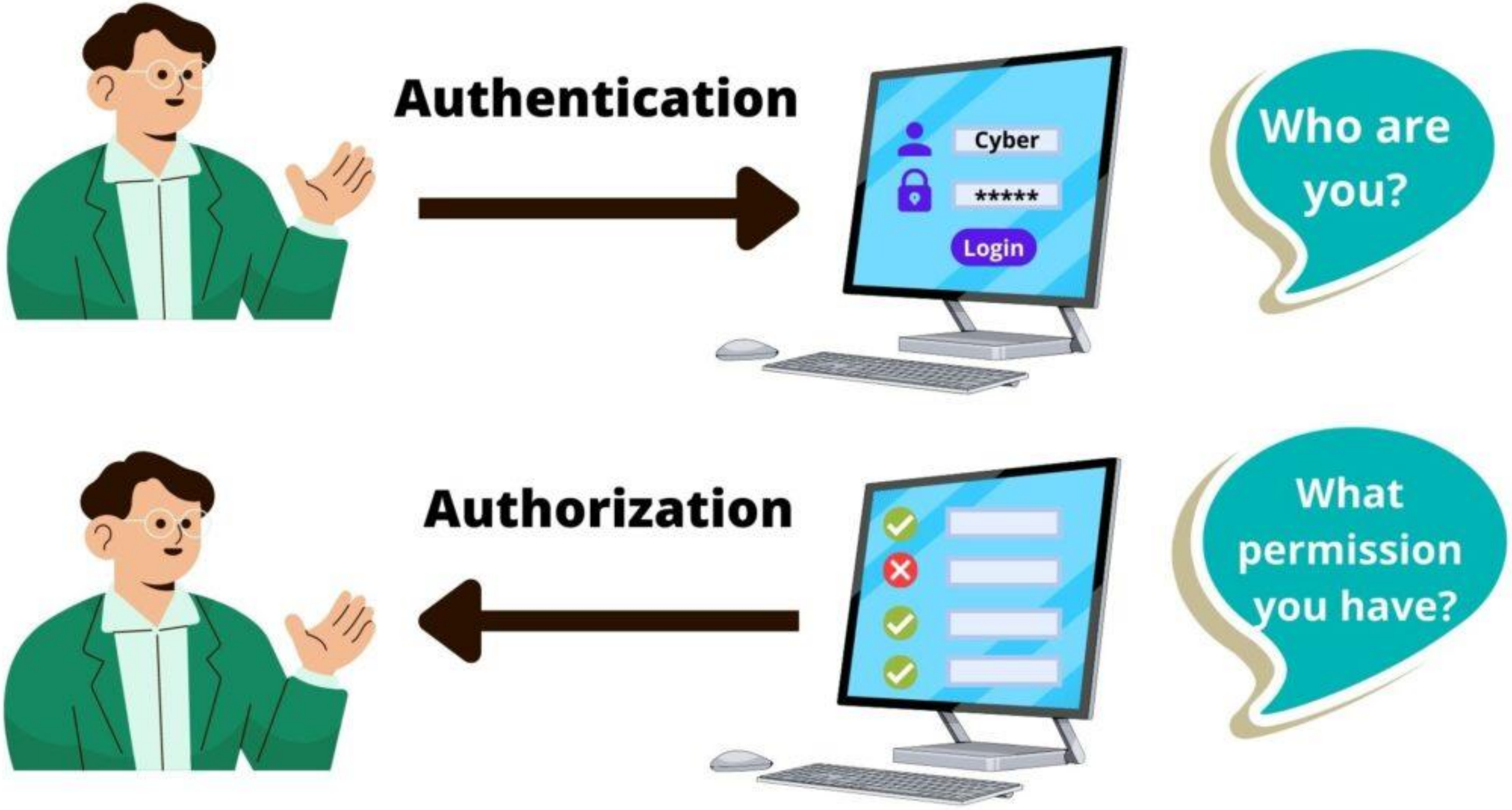
Source Installation

```
sudo -u git -H bundle exec rake gitlab:ldap:check RAILS_ENV=production
```

By default, the task returns a sample of 100 LDAP users. Change this limit by passing a number to the check task:

```
rake gitlab:ldap:check[50]
```

Authorization in Gitlab



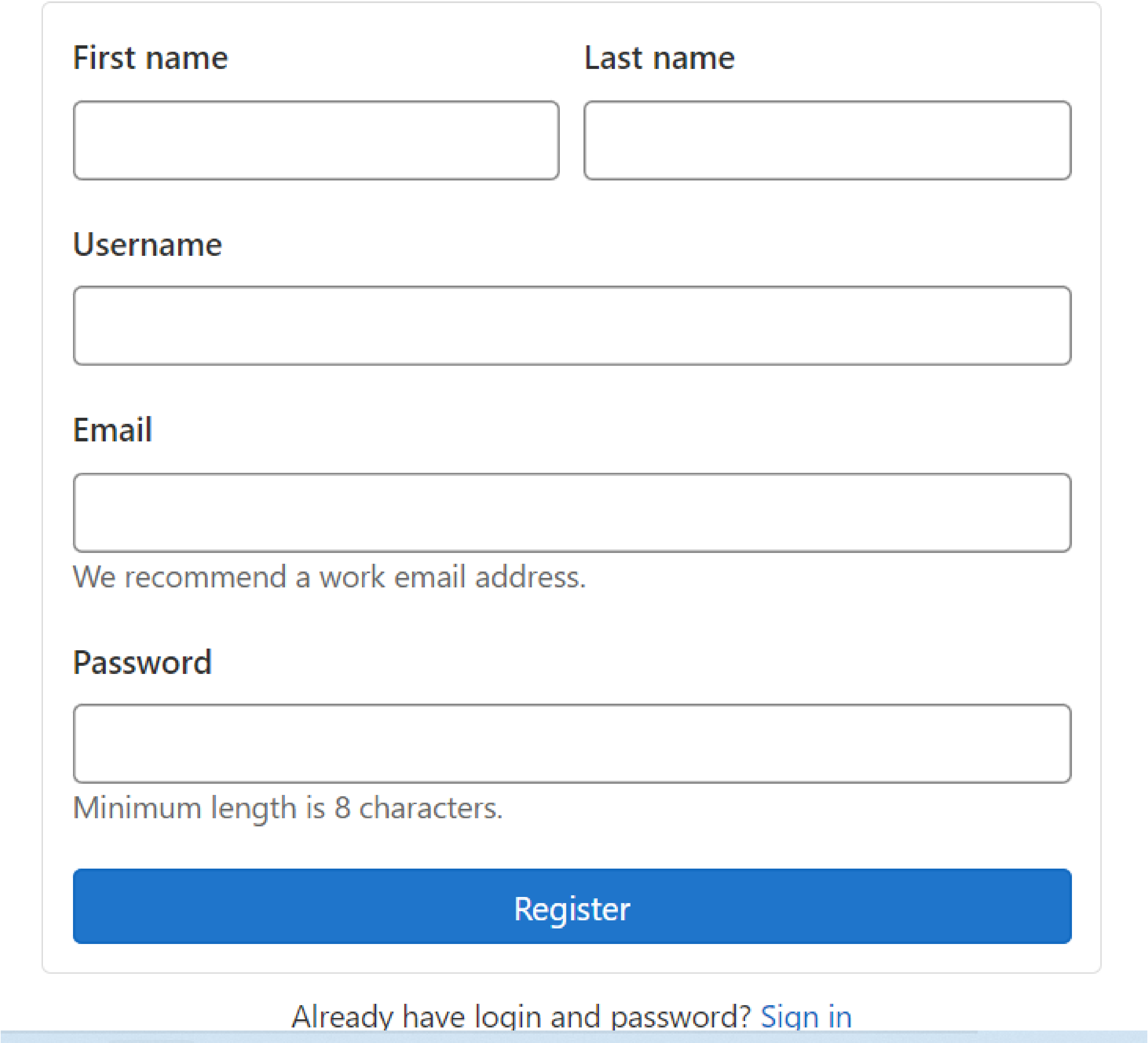
Authentication

Who are you?

Authorization

What permission you have?

Authorization: Step 1 - Register New Account

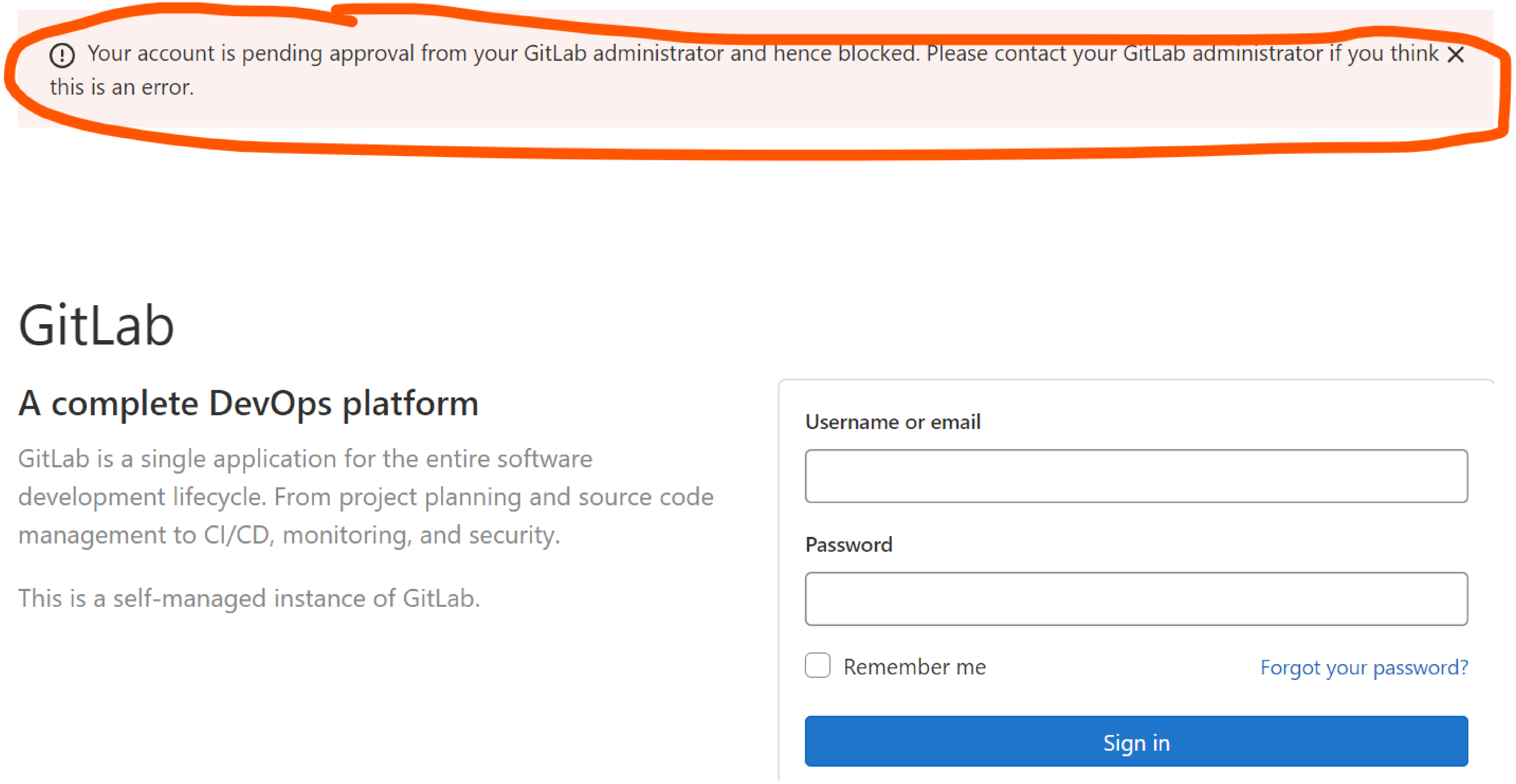


We recommend a work email address.

Minimum length is 8 characters.

Already have login and password? Sign in

Authorization: Step 2 - Pending Approval



ⓘ Your account is pending approval from your GitLab administrator and hence blocked. Please contact your GitLab administrator if you think this is an error. ✕

GitLab

A complete DevOps platform

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

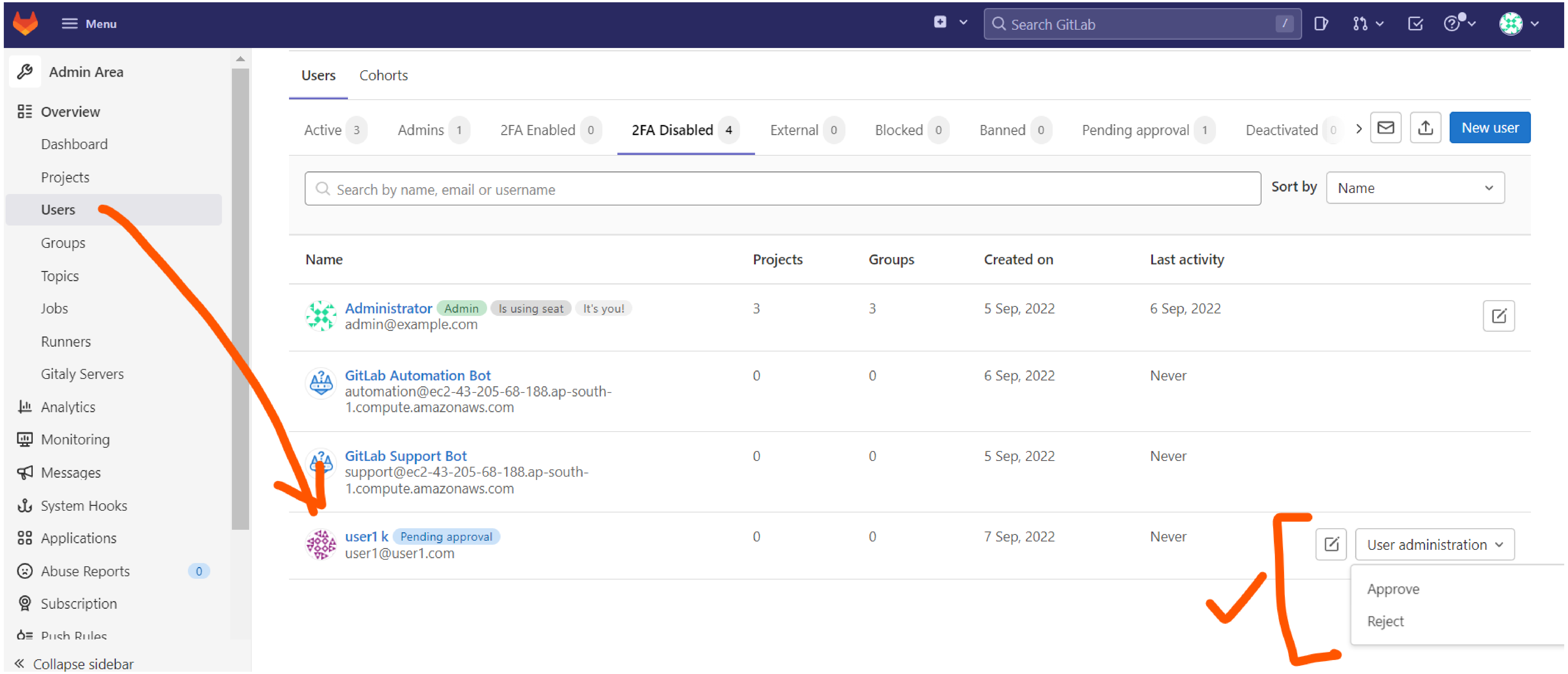
This is a self-managed instance of GitLab.

Username or email

Password

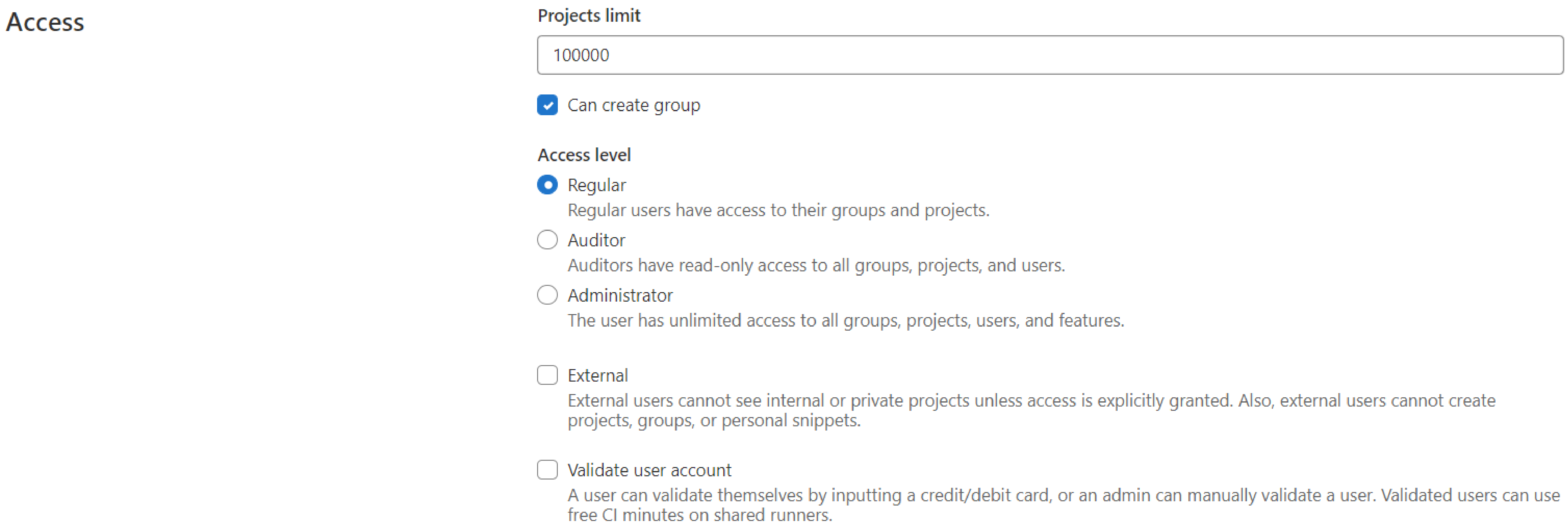
□ Remember me

Authorization: Step 2 - Approval



Name	Projects	Groups	Created on	Last activity
Administrator Admin Is using seat It's you! admin@example.com	3	3	5 Sep, 2022	6 Sep, 2022
GitLab Automation Bot automation@ec2-43-205-68-188.ap-south-1.compute.amazonaws.com	0	0	6 Sep, 2022	Never
GitLab Support Bot support@ec2-43-205-68-188.ap-south-1.compute.amazonaws.com	0	0	5 Sep, 2022	Never
user1 k Pending approval user1@user1.com	0	0	7 Sep, 2022	Never

Authorization: Gitlab Built-in roles



Access

Projects limit

100000

☑ Can create group

Access level

● Regular
Regular users have access to their groups and projects.

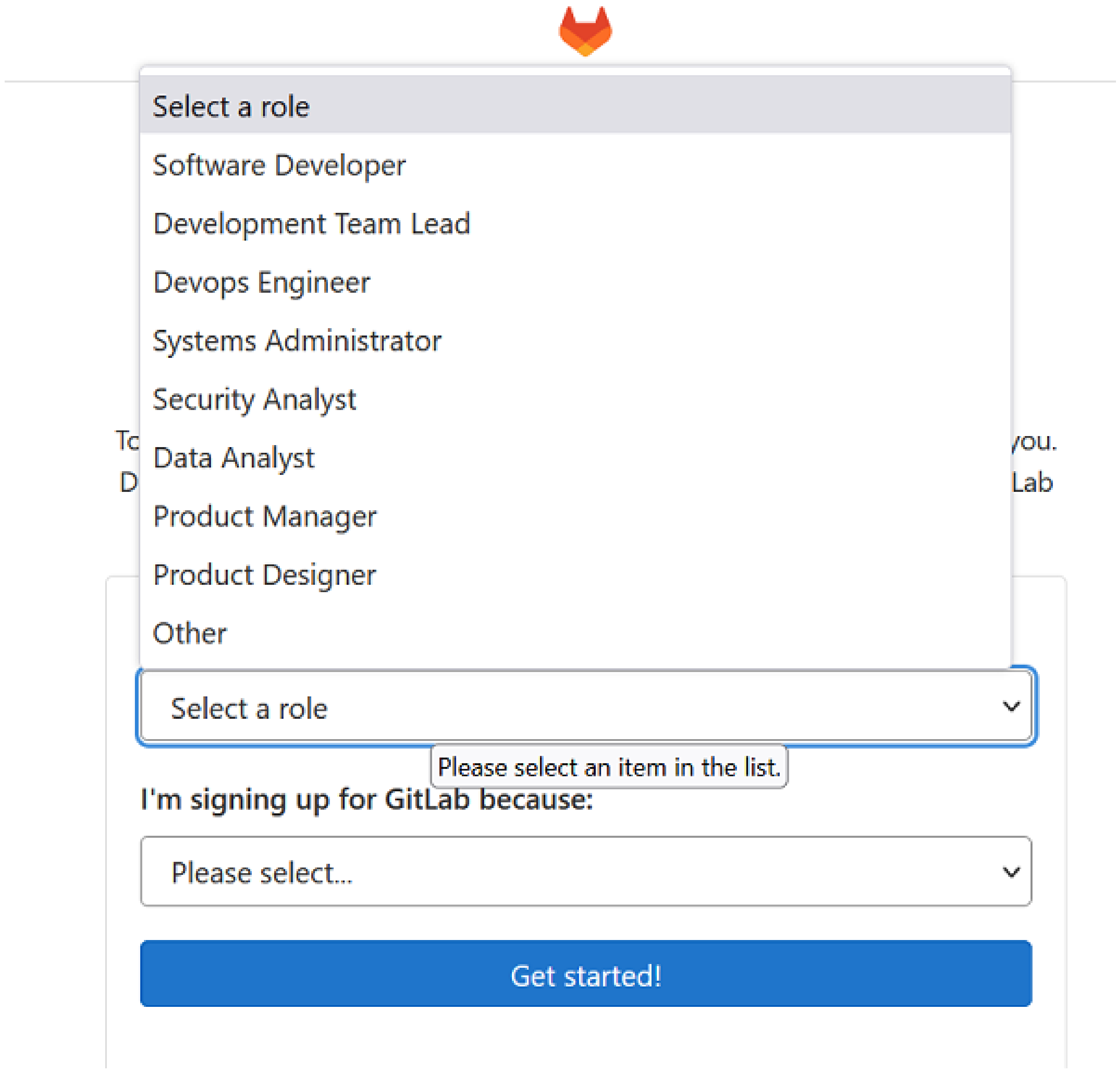
○ Auditor
Auditors have read-only access to all groups, projects, and users.

○ Administrator
The user has unlimited access to all groups, projects, users, and features.

□ External
External users cannot see internal or private projects unless access is explicitly granted. Also, external users cannot create projects, groups, or personal snippets.

□ Validate user account
A user can validate themselves by inputting a credit/debit card, or an admin can manually validate a user. Validated users can use free CI minutes on shared runners.

Authorization: Users can opt for role



Please select an item in the list.

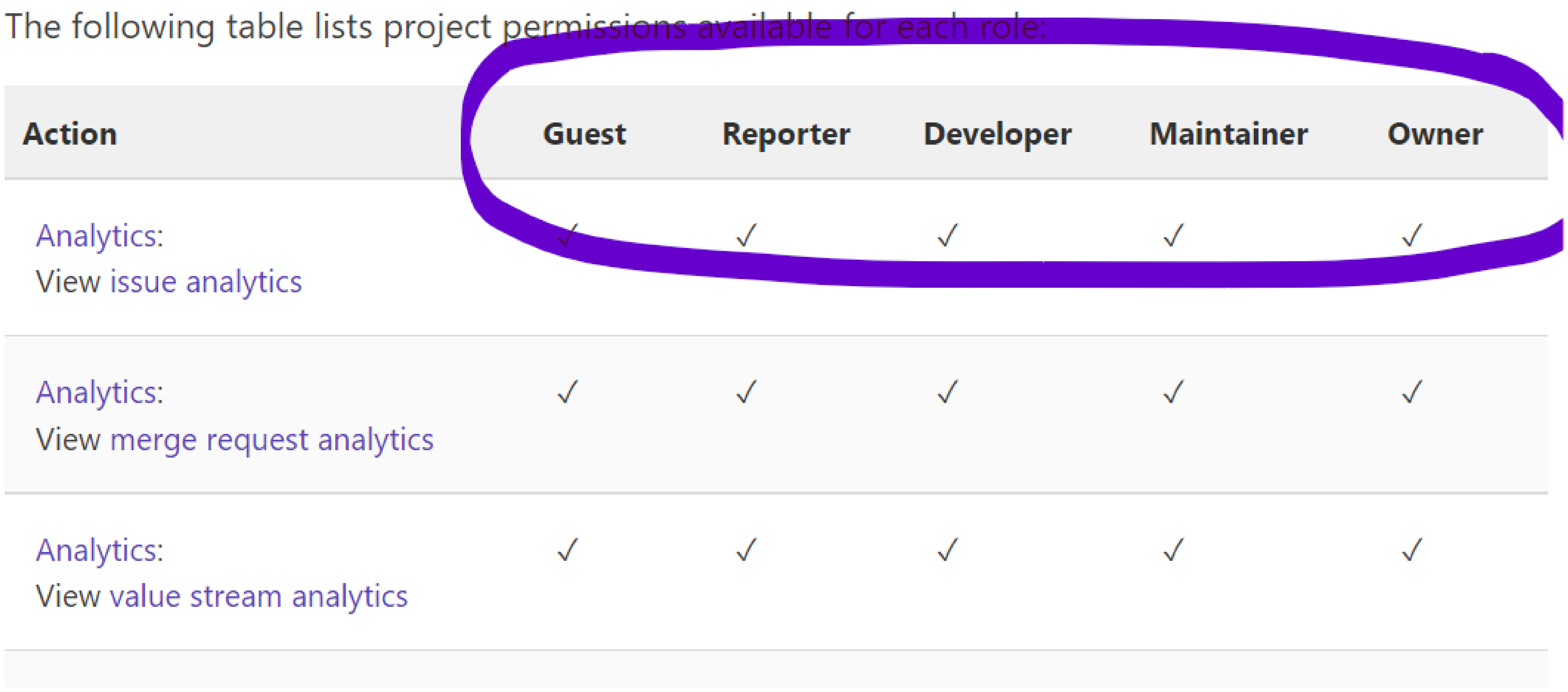
I'm signing up for GitLab because:

Authorization: Permission

A user’s role determines what permissions they have on a project. The Owner role provides all permissions but is available only:

- For group and project Owners. In GitLab 14.8 and earlier, the role is inherited for a group’s projects.
- For Administrators.

Authorization: Permission



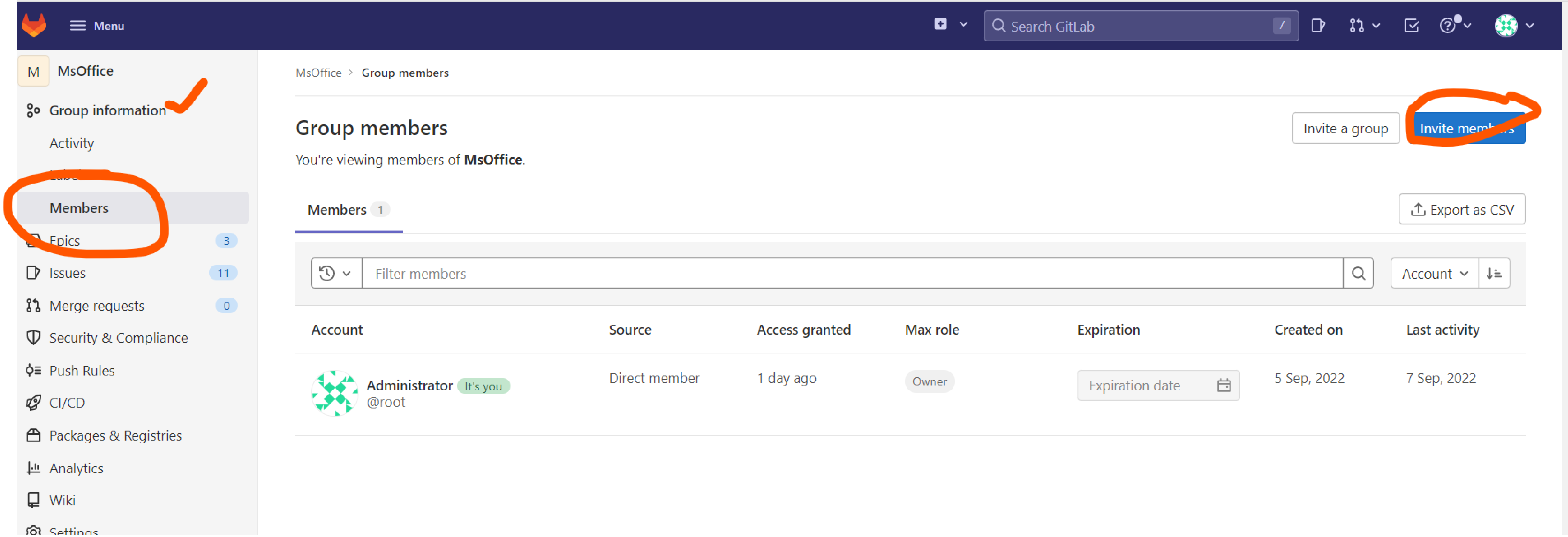
The following table lists project permissions available for each role:

Action	Guest	Reporter	Developer	Maintainer	Owner
Analytics: View issue analytics	✓	✓	✓	✓	✓
Analytics: View merge request analytics	✓	✓	✓	✓	✓
Analytics: View value stream analytics	✓	✓	✓	✓	✓

Authorization: Permission

https://docs.gitlab.com/ee/user/permissions.html

Authorization: Group Level Permission

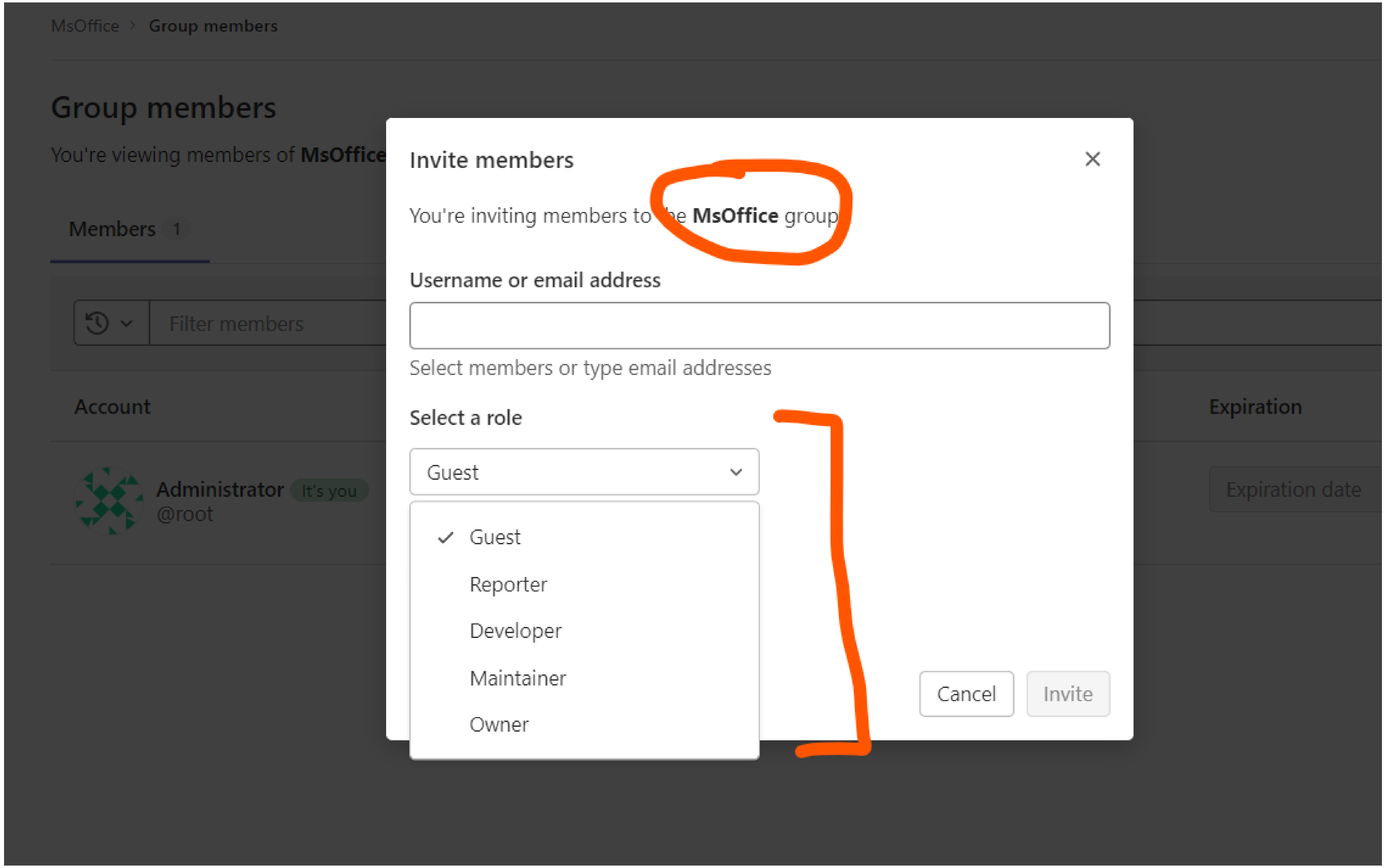


Group members

You're viewing members of **MsOffice**.

Account	Source	Access granted	Max role	Expiration	Created on	Last activity
Administrator It's you @root	Direct member	1 day ago	Owner	Expiration date	5 Sep, 2022	7 Sep, 2022

Authorization: Group Level Permission



MsOffice > Group members

Group members

You're viewing members of MsOffice

Members 1

Filter members

Account

Expiration

Administrator It's you
@root

Expiration date

Invite members

You're inviting members to the MsOffice group

Username or email address

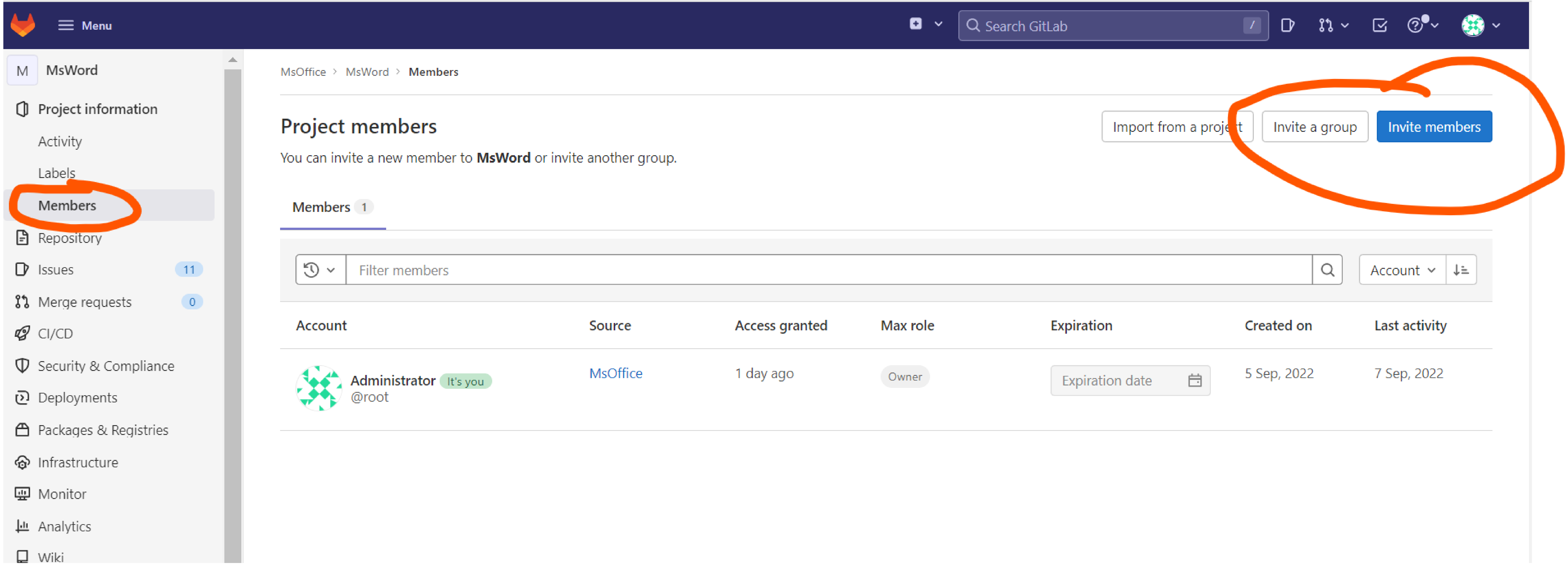
Select members or type email addresses

Select a role

Guest

✓ Guest
Reporter
Developer
Maintainer
Owner

Authorization: Project Level Permission



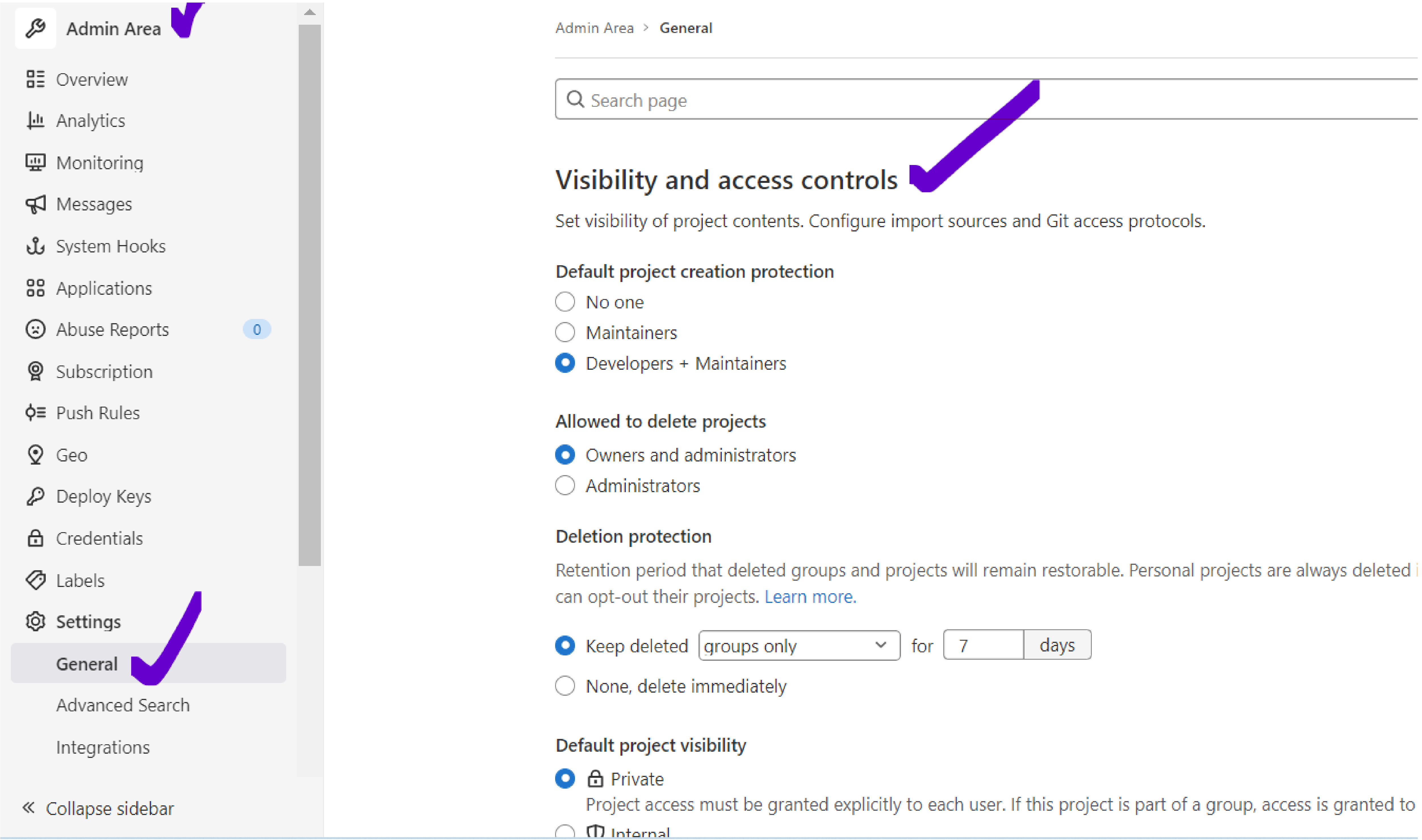
Project members

You can invite a new member to **MsWord** or invite another group.

Account	Source	Access granted	Max role	Expiration	Created on	Last activity
Administrator It's you @root	MsOffice	1 day ago	Owner	Expiration date	5 Sep, 2022	7 Sep, 2022

Authorization: Default System Setting

Authorization: Default System Setting



Visibility and access controls

Set visibility of project contents. Configure import sources and Git access protocols.

Default project creation protection

○ No one
○ Maintainers
● Developers + Maintainers

Allowed to delete projects

● Owners and administrators
○ Administrators

Deletion protection

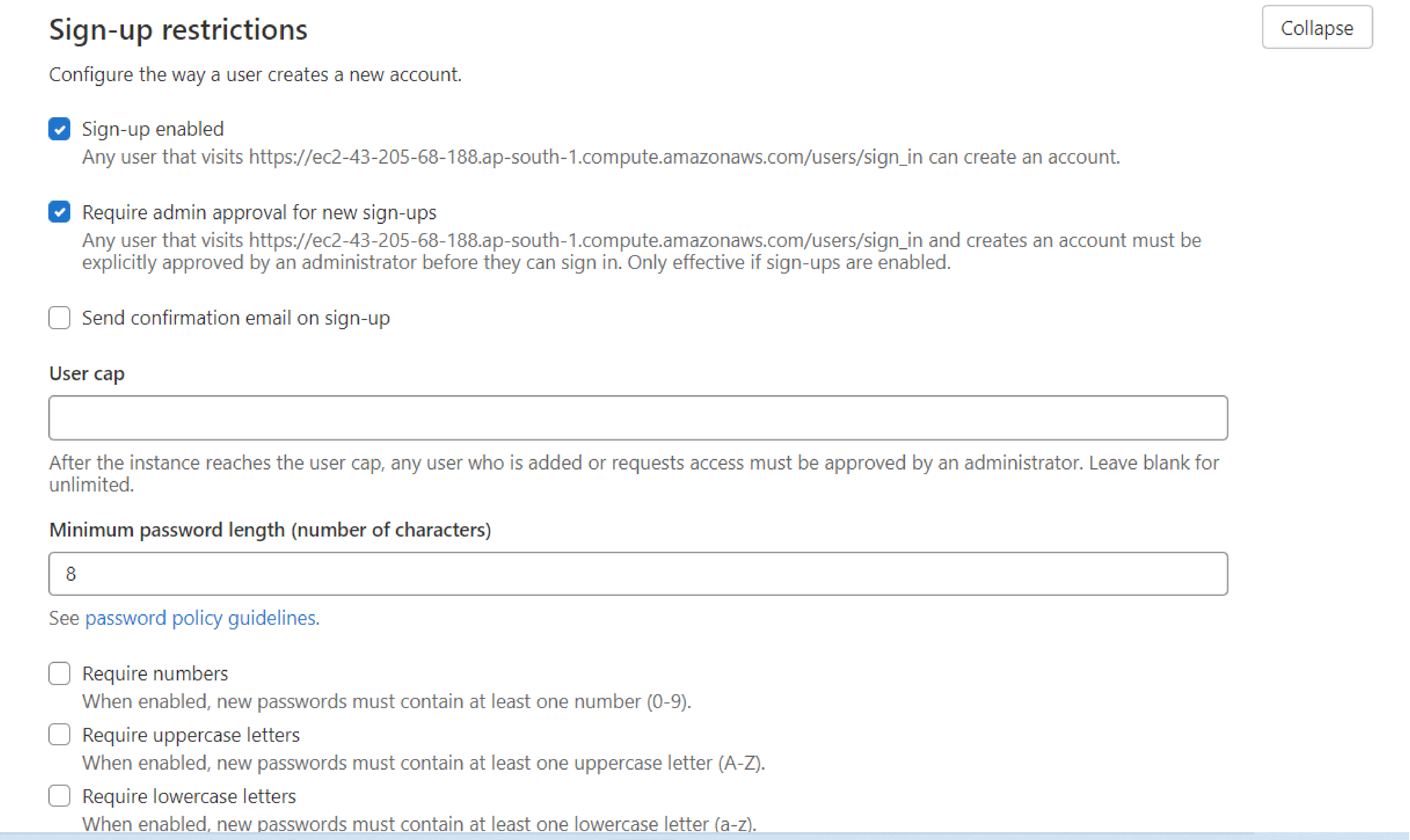
Retention period that deleted groups and projects will remain restorable. Personal projects are always deleted i can opt-out their projects. Learn more.

● Keep deleted groups only for 7 days
○ None, delete immediately

Default project visibility

● Private
Project access must be granted explicitly to each user. If this project is part of a group, access is granted to
○ Internal

Authorization: Default System Setting



Sign-up restrictions

Configure the way a user creates a new account.

☑ Sign-up enabled
Any user that visits https://ec2-43-205-68-188.ap-south-1.compute.amazonaws.com/users/sign_in can create an account.

☑ Require admin approval for new sign-ups
Any user that visits https://ec2-43-205-68-188.ap-south-1.compute.amazonaws.com/users/sign_in and creates an account must be explicitly approved by an administrator before they can sign in. Only effective if sign-ups are enabled.

☐ Send confirmation email on sign-up

User cap

After the instance reaches the user cap, any user who is added or requests access must be approved by an administrator. Leave blank for unlimited.

Minimum password length (number of characters)

8

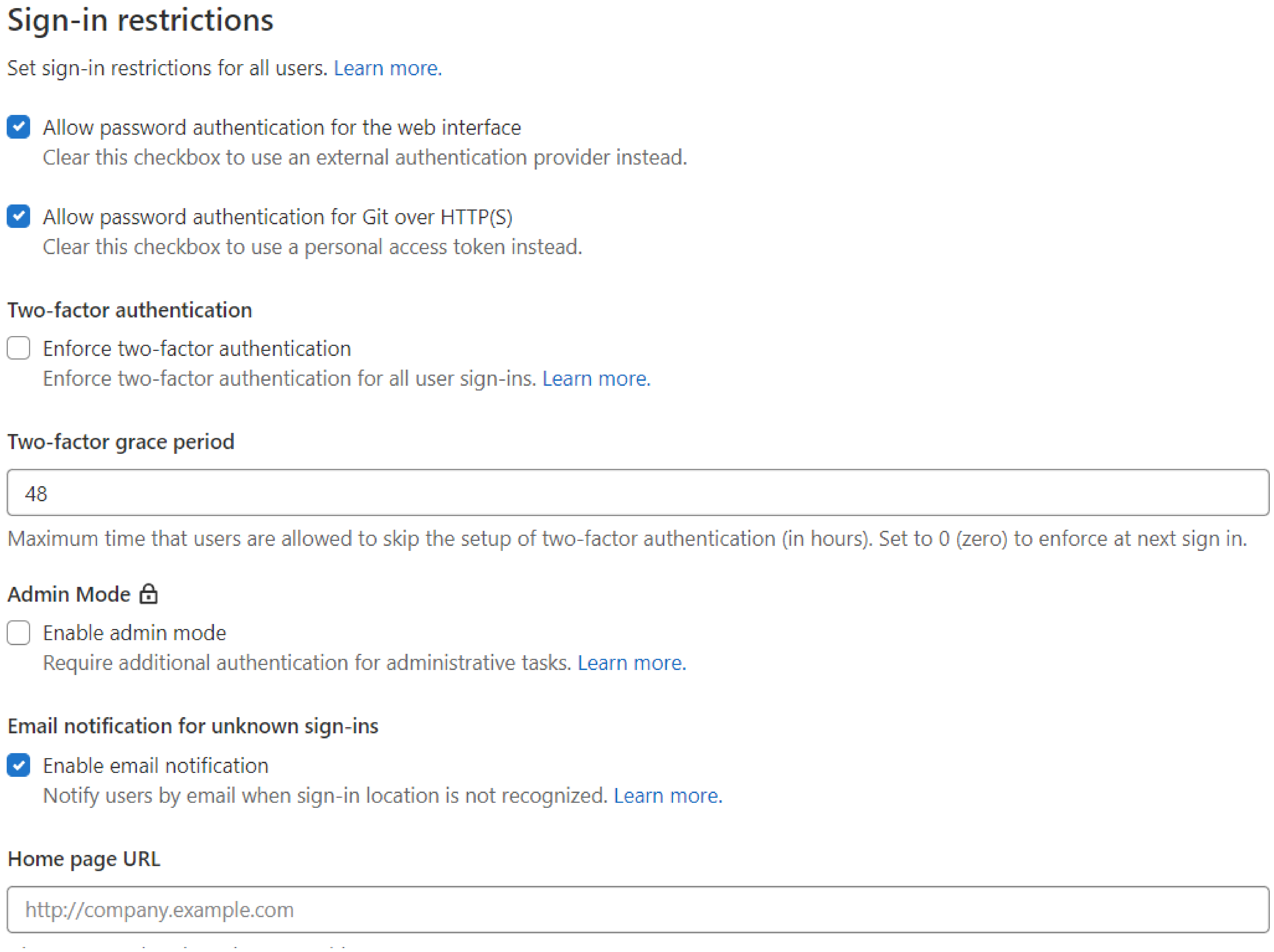
See password policy guidelines.

☐ Require numbers
When enabled, new passwords must contain at least one number (0-9).

☐ Require uppercase letters
When enabled, new passwords must contain at least one uppercase letter (A-Z).

☐ Require lowercase letters
When enabled, new passwords must contain at least one lowercase letter (a-z).

Authorization: Default System Setting



Sign-in restrictions

Set sign-in restrictions for all users. Learn more.

☑ Allow password authentication for the web interface
Clear this checkbox to use an external authentication provider instead.

☑ Allow password authentication for Git over HTTP(S)
Clear this checkbox to use a personal access token instead.

Two-factor authentication

□ Enforce two-factor authentication
Enforce two-factor authentication for all user sign-ins. Learn more.

Two-factor grace period

48

Maximum time that users are allowed to skip the setup of two-factor authentication (in hours). Set to 0 (zero) to enforce at next sign in.

Admin Mode 🔒

□ Enable admin mode
Require additional authentication for administrative tasks. Learn more.

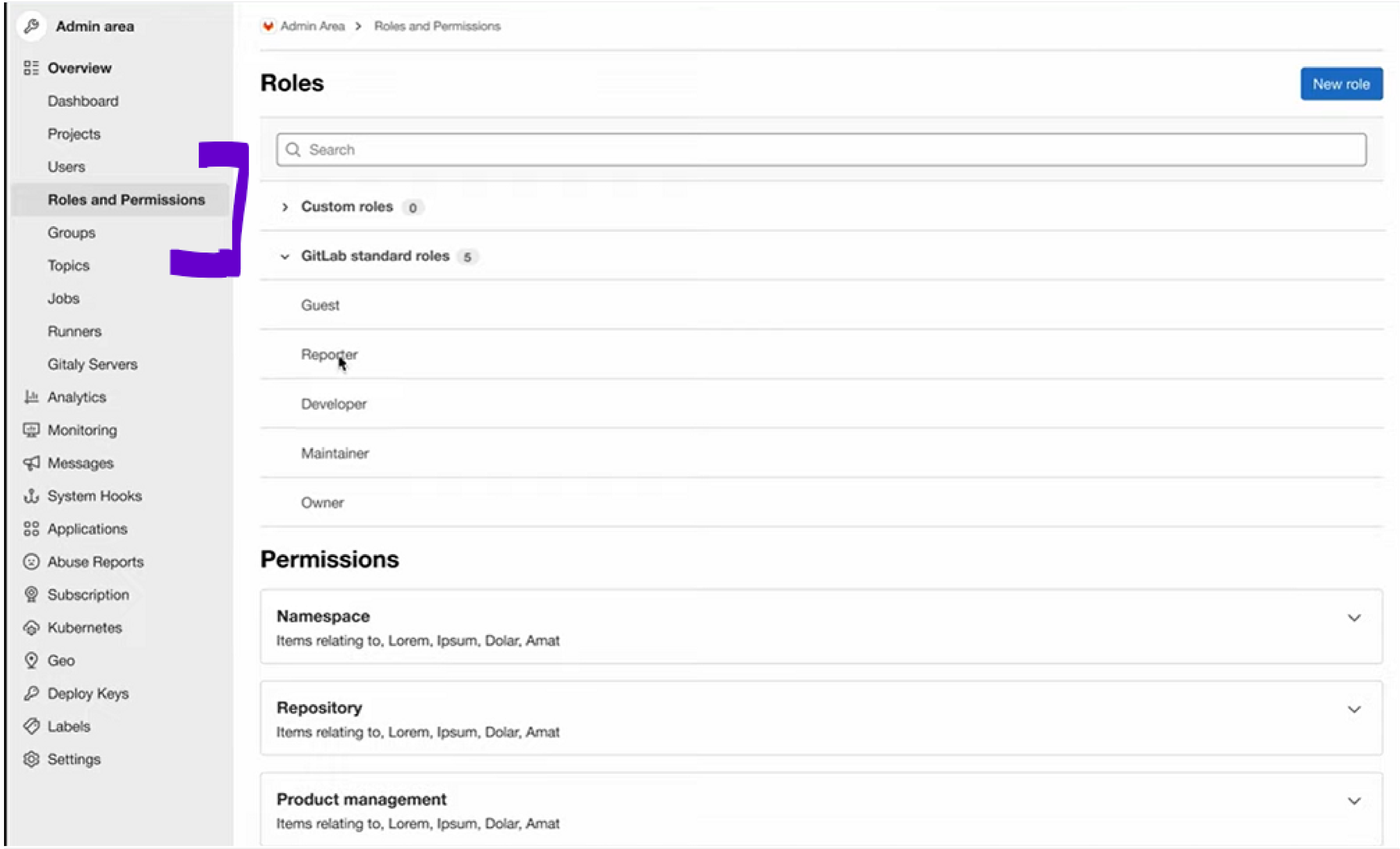
Email notification for unknown sign-ins

☑ Enable email notification
Notify users by email when sign-in location is not recognized. Learn more.

Home page URL

http://company.example.com

Limitation - Custom Roles - Upcoming



Roles

Custom roles 0

GitLab standard roles 5

- Guest
- Reporter
- Developer
- Maintainer
- Owner

Permissions

Namespace
Items relating to, Lorem, Ipsum, Dolar, Amat

Repository
Items relating to, Lorem, Ipsum, Dolar, Amat

Product management
Items relating to, Lorem, Ipsum, Dolar, Amat

Securing the Code

Demo

Run static code analysis

Demo

Detecting secrets

Summary

HTTPS and SSH

Create an SSH key

Setup SSH key in GitLab

Personal Access Tokens

Manage users and access

Detecting secrets

Up Next:
Delivery and Deployment