

Git Configuration and Tokens

This section covers the setup and configuration of Git repository integrations with RoostGPT. To enable seamless integration between RoostGPT and your code repositories, you'll need to configure access tokens with appropriate permissions for your Git provider. These tokens allow RoostGPT to access your repositories, create webhooks, create PRs and manage test-related operations.

- [GitHub \(Cloud & Enterprise\)](#)
- [Gitlab \(Cloud & Self-hosted\)](#)
- [Azure DevOps](#)
- [Bitbucket \(Cloud & Server\)](#)

GitHub (Cloud & Enterprise)

GitHub Cloud (github.com)

- **Token Type:** Personal Access Token (Classic)
- **Generation Steps:**

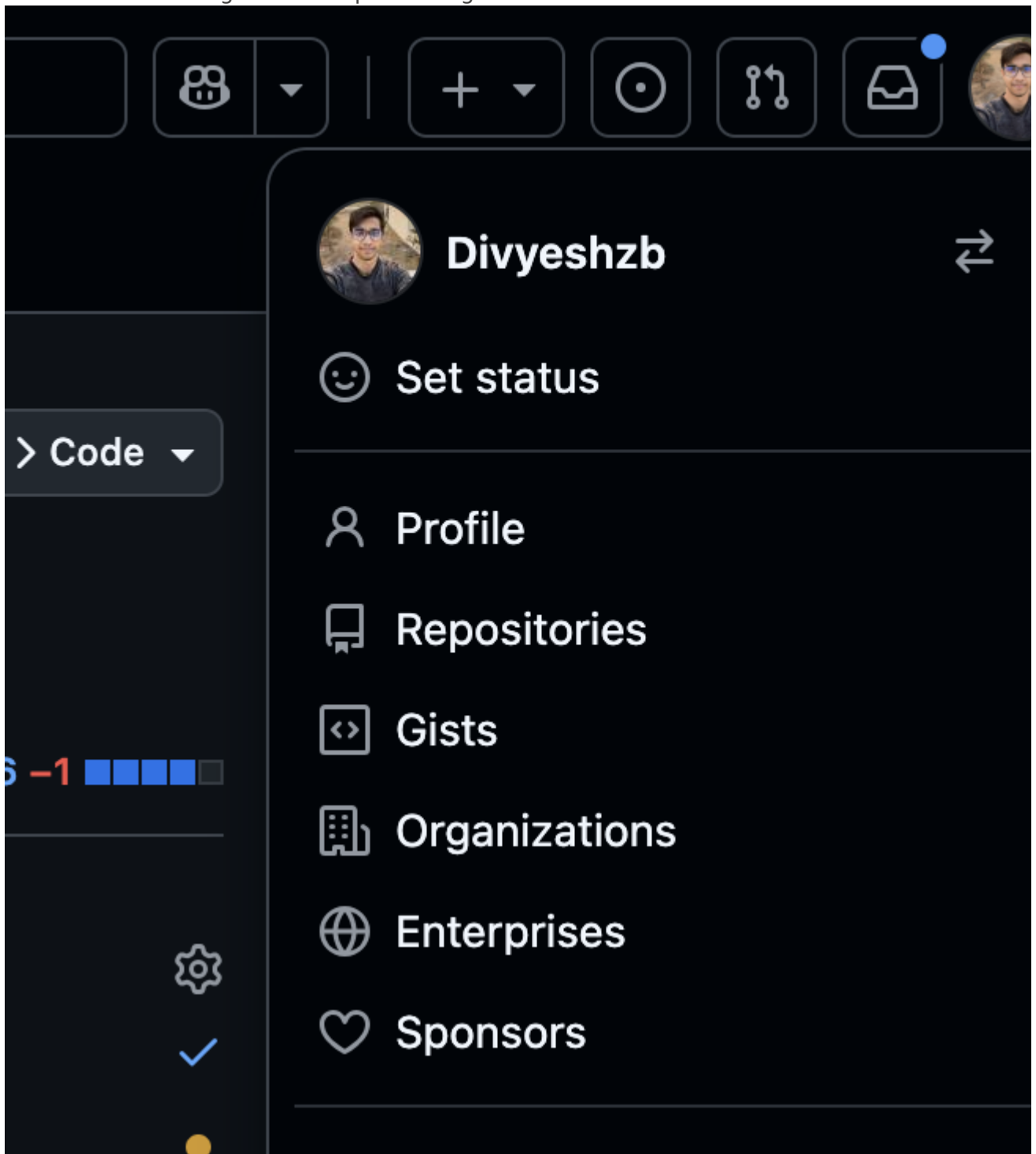
You can visit this site to quickly generate token

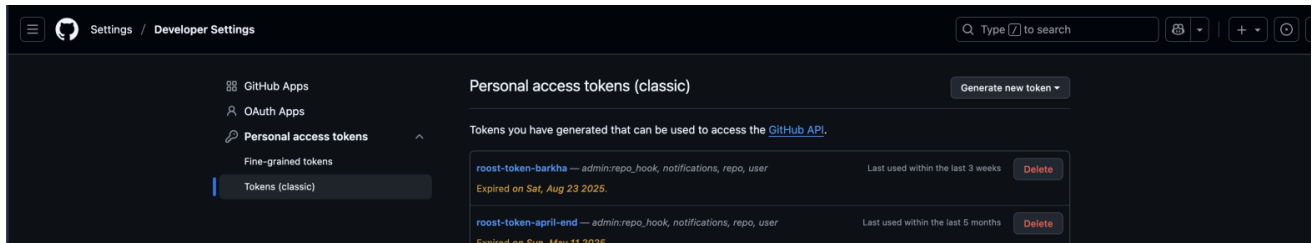
https://github.com/settings/tokens/new?description=roost-token&scopes=repo,admin:repo_hook,notifications,user

OR

Follow this steps

1. Go to GitHub Settings → Developer settings → Personal access tokens





2. Click "Generate new token" → "Generate new token (classic)"
3. Set expiration and select required scopes
4. Copy the generated token immediately

GitHub Enterprise Server

- **Token Type:** Personal Access Token
- **Generation Steps:** Same as GitHub Cloud but on your enterprise instance
- **Base URL:** Configure your enterprise server URL (e.g., `https://github.yourcompany.com`)

Required Permissions:

- `repo` - Full repository access (read/write to code, issues, PRs)
- `admin:repo_hook` - Repository webhook management
- `notification` - Access notifications
- `user` - Update ALL user data

Gitlab (Cloud & Self-hosted)

GitLab Cloud (gitlab.com)

- **Token Type:** Personal Access Token or Project Access Token
- **Generation Steps:**

You can visit this url directly and generate access Token

<https://gitlab.com/>

[/user settings/personal access tokens?page=1&state=active&sort=expires_asc](https://gitlab.com/user-settings/personal-access-tokens?page=1&state=active&sort=expires_asc)

OR

Follow this steps

1. Go to GitLab Settings → Access Tokens

The screenshot shows the GitLab 'Personal access tokens' configuration page. On the left is a sidebar with 'Access tokens' highlighted. The main area has a search bar and an 'Add new token' button. Below is a form with the following fields:

- Token name:** A text input field containing 'roost-token'.
- Description (optional):** A large text area for an optional description.
- Expiration date:** A date picker set to '2025-10-08'. A note below states: 'An administrator has set the maximum expiration date to 2026-09-08. Learn more.'
- Select scopes:** A section with two checked options:
 - read_user**: Grants read-only access to your profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
 - read_repository**: Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.

2. Enter token name and expiration date
3. Select required scopes
4. Click "Create token"

GitLab Self-hosted

- **Token Type:** Personal Access Token
- **Base URL:** Configure your GitLab instance URL

Required Permissions:

- `api` - Complete API access
- `read_repository` - Read repository content
- `write_repository` - Write repository content
- `read_user` - Read user information
- `read_api` - Read api information

Azure DevOps

Bitbucket (Cloud & Server)

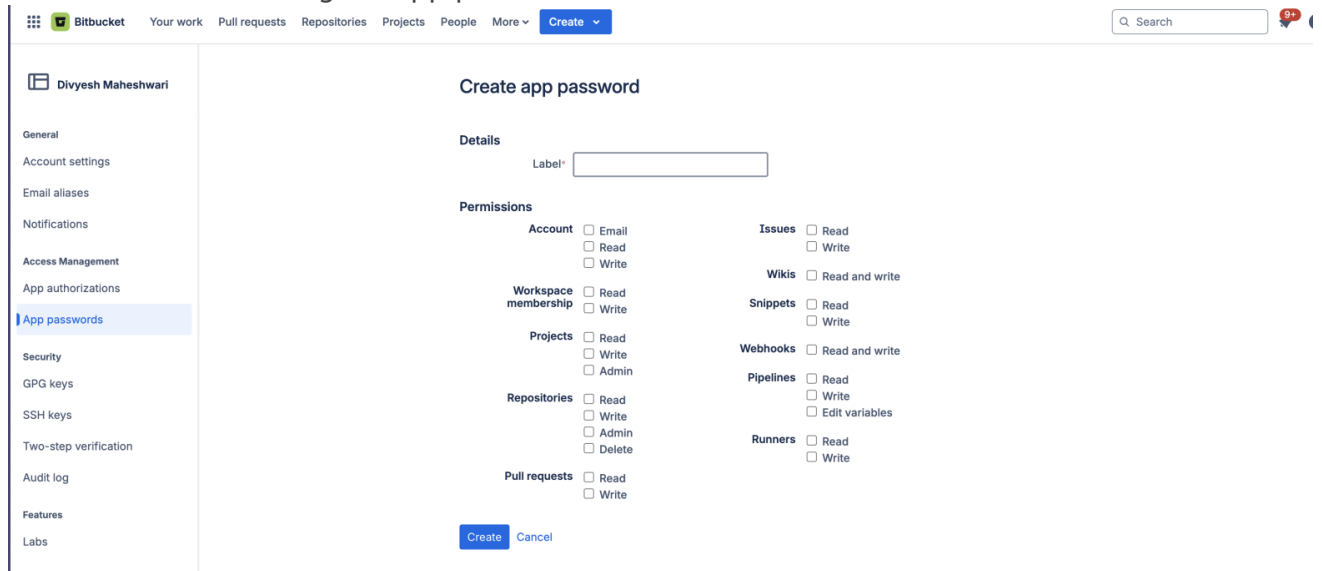
Bitbucket Cloud (bitbucket.org)

- **Token Type:** App Password or Repository Access Token
- **Generation Steps:**
Use this link to directly generate App Password
<https://bitbucket.org/account/settings/app-passwords/new>

OR

Follow this steps

1. Go to Bitbucket Settings → App passwords



The screenshot shows the Bitbucket 'Create app password' interface. On the left is a sidebar with navigation options: General, Account settings, Email aliases, Notifications, Access Management, App authorizations, App passwords (highlighted), Security, GPG keys, SSH keys, Two-step verification, Audit log, Features, and Labs. The main content area is titled 'Create app password' and includes a 'Details' section with a 'Label*' input field. Below this is a 'Permissions' section with checkboxes for various categories: Account (Email, Read, Write), Workspace membership (Read, Write), Projects (Read, Write, Admin), Repositories (Read, Write, Admin, Delete), Pull requests (Read, Write), Issues (Read, Write), Wikis (Read and write), Snippets (Read, Write), Webhooks (Read and write), Pipelines (Read, Write, Edit variables), and Runners (Read, Write). At the bottom of the permissions section are 'Create' and 'Cancel' buttons.

2. Create label and select permissions
3. Click Create

Bitbucket Server/Data Center

- **Token Type:** Personal Access Token
- **Base URL:** Configure your Bitbucket server URL

Required Permissions:

- Account (Read) - Account Information
- Repositories (Read & Write) - Repository access
- Webhooks (Read & Write) - Webhook management
- Pull Requests (Read & Write) - Pull request management
- Projects (Read & Write) - Project information access