GitLab Provider

The GitLab provider is used to interact with GitLab group or user resources.

It needs to be configured with the proper credentials before it can be used.

Use the navigation to the left to read about the available resources.

Example Usage

```
# Configure the GitLab Provider
provider "gitlab" {
    token = "${var.gitlab_token}"
}

# Add a project owned by the user
resource "gitlab_project" "sample_project" {
    name = "example"
}

# Add a hook to the project
resource "gitlab_project_hook" "sample_project_hook" {
    project = "${gitlab_project.sample_project.id}"
    url = "https://example.com/project_hook"
}

# Add a deploy key to the project
resource "gitlab_deploy_key" "sample_deploy_key" {
    project = "${gitlab_project.sample_project.id}"
    title = "terraform example"
    key = "ssh-rsa AAAA..."
}

# Add a group
resource "gitlab_group" "sample_group" {
    name = "example"
    path = "example"
    description = "An example group"
}

# Add a project to the group - example/example
resource "gitlab_project" "sample_group_project" {
    name = "example"
    namespace_id = "${gitlab_group.sample_group.id}"
}
```

Argument Reference

The following arguments are supported in the `provider` block:

- `token` - (Optional) This is the GitLab personal access token. It must be provided, but it can also be sourced from the `GITLAB_TOKEN` environment variable.

- `base_url` - (Optional) This is the target GitLab base API endpoint. Providing a value is a requirement when working with GitLab CE or GitLab Enterprise e.g. https://my.gitlab.server/api/v3/ (https://my.gitlab.server/api/v3/). It is optional to provide this value and it can also be sourced from the `GITLAB_BASE_URL` environment variable. The value must end with a slash.

- `cacert_file` - (Optional) This is a file containing the ca cert to verify the gitlab instance. This is available for use when working with GitLab CE or Gitlab Enterprise with a locally-issued or self-signed certificate chain.

- `insecure` - (Optional; boolean, defaults to false) When set to true this disables SSL verification of the connection to the GitLab instance.

gitlab_deploy_key

This resource allows you to create and manage deploy keys for your GitLab projects.

Example Usage

```
resource "gitlab_deploy_key" "example" {
  project = "example/deploying"
  title   = "Example deploy key"
  key     = "ssh-rsa AAAA..."
}
```

Argument Reference

The following arguments are supported:

- `project` - (Required, string) The name or id of the project to add the deploy key to.
- `title` - (Required, string) A title to describe the deploy key with.
- `key` - (Required, string) The public ssh key body.
- `can_push` - (Optional, boolean) Allow this deploy key to be used to push changes to the project. Defaults to `false`. **NOTE::** this cannot currently be managed.

gitlab_group

This resource allows you to create and manage GitLab groups. Note your provider will need to be configured with admin-level access for this resource to work.

Example Usage

```
resource "gitlab_group" "example" {
  name        = "example"
  path        = "example"
  description = "An example group"
}

// Create a project in the example group
resource "gitlab_project" "example" {
  name         = "example"
  description  = "An example project"
  namespace_id = "${gitlab_group.example.id}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of this group.
- `path` - (Required) The path of the group.
- `description` - (Optional) The description of the group.
- `lfs_enabled` - (Optional) Boolean, defaults to true. Whether to enable LFS support for projects in this group.
- `request_access_enabled` - (Optional) Boolean, defaults to false. Whether to enable users to request access to the group.
- `visibility_level` - (Optional) Set to `public` to create a public group. Valid values are `private`, `internal`, `public`. Groups are created as private by default.
- `parent_id` - (Optional) Integer, id of the parent group (creates a nested group).

Attributes Reference

The resource exports the following attributes:

- `id` - The unique id assigned to the group by the GitLab server. Serves as a namespace id where one is needed.

gitlab_label

This resource allows you to create and manage labels for your GitLab projects. For further information on labels, consult the gitlab documentation (https://docs.gitlab.com/ee/user/project/labels.htm).

Example Usage

```
resource "gitlab_label" "fixme" {
  project     = "example"
  name        = "fixme"
  description = "issue with failing tests"
  color       = "#ffcc00"
}
```

Argument Reference

The following arguments are supported:

- `project` - (Required) The name or id of the project to add the label to.

- `name` - (Required) The name of the label.

- `color` - (Required) The color of the label given in 6-digit hex notation with leading '#' sign (e.g. #FFAABB) or one of the CSS color names (https://developer.mozilla.org/en-US/docs/Web/CSS/color_value#Color_keywords).

- `description` - (Optional) The description of the label.

Attributes Reference

The resource exports the following attributes:

- `id` - The unique id assigned to the label by the GitLab server (the name of the label).

gitlab_project

This resource allows you to create and manage projects within your GitLab group or within your user.

Example Usage

```
resource "gitlab_project" "example" {
  name        = "example"
  description = "My awesome codebase"

  visibility_level = "public"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the project.
- `path` - (Optional) The path of the repository.
- `namespace_id` - (Optional) The namespace (group or user) of the project. Defaults to your user. See `gitlab_group` (/docs/providers/gitlab/r/group.html) for an example.
- `description` - (Optional) A description of the project.
- `default_branch` - (Optional) The default branch for the project.
- `issues_enabled` - (Optional) Enable issue tracking for the project.
- `merge_requests_enabled` - (Optional) Enable merge requests for the project.
- `wiki_enabled` - (Optional) Enable wiki for the project.
- `snippets_enabled` - (Optional) Enable snippets for the project.
- `visibility_level` - (Optional) Set to `public` to create a public project. Valid values are `private`, `internal`, `public`. Repositories are created as private by default.

Attributes Reference

The following additional attributes are exported:

- `id` - Integer that uniquely identifies the project within the gitlab install.
- `ssh_url_to_repo` - URL that can be provided to `git clone` to clone the repository via SSH.
- `http_url_to_repo` - URL that can be provided to `git clone` to clone the repository via HTTP.
- `web_url` - URL that can be used to find the project in a browser.

gitlab_project_hook

This resource allows you to create and manage hooks for your GitLab projects. For further information on hooks, consult the gitlab documentation (https://docs.gitlab.com/ce/user/project/integrations/webhooks.html).

Example Usage

```
resource "gitlab_project_hook" "example" {
  project               = "example/hooked"
  url                   = "https://example.com/hook/example"
  merge_requests_events = true
}
```

Argument Reference

The following arguments are supported:

- `project` - (Required) The name or id of the project to add the hook to.
- `url` - (Required) The url of the hook to invoke.
- `token` - (Optional) A token to present when invoking the hook.
- `enable_ssl_verification` - (Optional) Enable ssl verification when invoking the hook.
- `push_events` - (Optional) Invoke the hook for push events.
- `issues_events` - (Optional) Invoke the hook for issues events.
- `merge_requests_events` - (Optional) Invoke the hook for merge requests.
- `tag_push_events` - (Optional) Invoke the hook for tag push events.
- `note_events` - (Optional) Invoke the hook for notes events.
- `job_events` - (Optional) Invoke the hook for job events.
- `pipeline_events` - (Optional) Invoke the hook for pipeline events.
- `wiki_page_events` - (Optional) Invoke the hook for wiki page events.

Attributes Reference

The resource exports the following attributes:

- `id` - The unique id assigned to the hook by the GitLab server.

gitlab_user

This resource allows you to create and manage GitLab users. Note your provider will need to be configured with admin-level access for this resource to work.

Example Usage

```
resource "gitlab_user" "example" {
  name             = "Example Foo"
  username         = "example"
  password         = "superPassword"
  email            = "gitlab@user.create"
  is_admin         = true
  projects_limit   = 4
  can_create_group = false
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the user.
- `username` - (Required) The username of the user.
- `password` - (Required) The password of the user.
- `email` - (Required) The e-mail address of the user.
- `is_admin` - (Optional) Boolean, defaults to false. Whether to enable administrative priviledges for the user.
- `projects_limit` - (Optional) Integer, defaults to 0. Number of projects user can create.
- `can_create_group` - (Optional) Boolean, defaults to false. Whether to allow the user to create groups.
- `skip_confirmation` - (Optional) Boolean, defaults to true. Whether to skip confirmation.

Attributes Reference

The resource exports the following attributes:

- `id` - The unique id assigned to the user by the GitLab server.