

Grafana Provider

The Grafana provider configures data sources and dashboards in Grafana (<http://grafana.org/>), which is a web application for creating, viewing and sharing metrics dashboards.

The provider configuration block accepts the following arguments:

- `url` - (Required) The root URL of a Grafana server. May alternatively be set via the `GRAFANA_URL` environment variable.
- `auth` - (Required) The API token or username/password to use to authenticate to the Grafana server. If username/password is used, they are provided in a single string and separated by a colon. May alternatively be set via the `GRAFANA_AUTH` environment variable.

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

Example Usage

```
provider "grafana" {
  url = "http://grafana.example.com/"
  auth = "1234abcd"
}

resource "grafana_dashboard" "metrics" {
  config_json = "${file("grafana-dashboard.json")}"
}

resource "grafana_data_source" "influxdb" {
  type          = "influxdb"
  name          = "test_influxdb"
  url           = "http://influxdb.example.net:8086/"
  username      = "foo"
  password      = "bar"
  database_name = "mydb"
}

resource "grafana_alert_notification" "slack" {
  name = "My Slack"
  type = "slack"

  settings {
    "slack" = "https://myteam.slack.com/hook"
    "recipient" = "@someguy"
    "uploadImage" = "false"
  }
}

resource "grafana_organization" "org" {
  name          = "Grafana Organization"
  admin_user    = "admin"
  create_users = true
  admins        = [
    "admin@example.com"
  ]
  editors       = [
    "editor-01@example.com",
    "editor-02@example.com"
  ]
  viewers       = [
    "viewer-01@example.com",
    "viewer-02@example.com"
  ]
}
```

grafana_alert_notification

The alert notification resource allows an alert notification channel to be created on a Grafana server.

Example Usage

```
resource "grafana_alert_notification" "email_someteam" {
  name = "Email that team"
  type = "email"
  is_default = false

  settings {
    "addresses" = "foo@example.net;bar@example.net"
    "uploadImage" = "false"
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the alert notification channel.
- `type` - (Required) The type of the alert notification channel.
- `is_default` - (Optional) Is this the default channel for all your alerts.
- `settings` - (Optional) Additional settings, for full reference lookup Grafana HTTP API documentation (http://docs.grafana.org/http_api/alerting).

Note: In `settings` the strings `"true"` and `"false"` are mapped to boolean `true` and `false` when sent to Grafana.

Attributes Reference

The resource exports the following attributes:

- `id` - The ID of the resource

grafana_dashboard

The dashboard resource allows a dashboard to be created on a Grafana server.

Example Usage

```
resource "grafana_dashboard" "metrics" {
  config_json = "${file("grafana-dashboard.json")}"
}
```

Dashboards can be exported from Grafana's web UI in JSON format and provided via the `config_json` argument.

The exported JSON will include references to Grafana data sources, but the data source configuration is not exported in this way. In order to fully manage a dashboard with Terraform, necessary data sources can be created using the `grafana_data_source` resource. In order to ensure that a data source is created before a dashboard that refers to it, use the `depends_on` meta-parameter:

```
depends_on = ["grafana_data_source.metrics"]
```

Argument Reference

The following arguments are supported:

- `config_json` - (Required) The JSON configuration for the dashboard.

Attributes Reference

The resource exports the following attributes:

- `slug` - A URL "slug" for this dashboard, generated by Grafana by removing certain characters from the dashboard name given as part of the `config_json` argument. This can be used to generate the URL for a dashboard.

Import

Existing organizations can be imported using the dashboard "slug" which can be obtained from the url of the dashboard in Grafana (e.g. `https://grafana.mydomain.com/d/abcdef-gh/dashboard-slug`)

```
$ terraform import grafana_dashboard.dashboard_name {dashboard_slug}
```

grafana_data_source

The data source resource allows a data source to be created on a Grafana server.

Example Usage

The required arguments for this resource vary depending on the type of data source selected (via the `type` argument). The following example is for InfluxDB. See Grafana's *Data Sources Guides* (<http://docs.grafana.org/#data-sources-guides>) for more details on the supported data source types and the arguments they use.

For an InfluxDB datasource:

```
resource "grafana_data_source" "metrics" {
  type      = "influxdb"
  name      = "myapp-metrics"
  url       = "http://influxdb.example.net:8086/"
  username  = "myapp"
  password  = "foobarbaz"
  database_name = "${influxdb_database.metrics.name}"
}
```

For a CloudWatch datasource:

```
resource "grafana_data_source" "test_cloudwatch" {
  type = "cloudwatch"
  name = "cw-example"

  json_data {
    default_region = "us-east-1"
    auth_type      = "keys"
  }

  secure_json_data {
    access_key = "123"
    secret_key = "456"
  }
}
```

Argument Reference

The following arguments are supported:

- `type` - (Required) The data source type. Must be one of the data source keywords supported by the Grafana server.
- `name` - (Required) A unique name for the data source within the Grafana server.
- `url` - (Optional) The URL for the data source. The type of URL required varies depending on the chosen data source type.

- `is_default` - (Optional) If true, the data source will be the default source used by the Grafana server. Only one data source on a server can be the default.
- `basic_auth_enabled` - (Optional) - If true, HTTP basic authentication will be used to make requests.
- `basic_auth_username` - (Required if `basic_auth_enabled` is true) The username to use for basic auth.
- `basic_auth_password` - (Required if `basic_auth_enabled` is true) The password to use for basic auth.
- `username` - (Required by some data source types) The username to use to authenticate to the data source.
- `password` - (Required by some data source types) The password to use to authenticate to the data source.
- `json_data` - (Required by some data source types) The default region and authentication type to access the data source. `json_data` is documented in more detail below.
- `secure_json_data` - (Required by some data source types) The access and secret keys required to access the data source. `secure_json_data` is documented in more detail below.
- `database_name` - (Required by some data source types) The name of the database to use on the selected data source server.
- `access_mode` - (Optional) The method by which the browser-based Grafana application will access the data source. The default is "proxy", which means that the application will make requests via a proxy endpoint on the Grafana server.

JSON Data (`json_data`) supports the following:

- `auth_type` - (Required by some data source types) The authentication type used to access the data source.
- `default` - (Required by some data source types) The default region for the data source.
- `custom_metrics_namespaces` - (Optional, for the CloudWatch data source type) A comma-separated list of custom namespaces to be queried by the CloudWatch data source.
- `assume_role_arn` - (Optional, for the CloudWatch data source type) The role ARN to be assumed by Grafana when using the CloudWatch data source.

Secure JSON Data (`secure_json_data`) supports the following:

- `access_key` - (Required by some data source types) The access key required to access the data source.
- `secret_key` - (Required by some data source types) The secret key required to access the data source.

Attributes Reference

The resource exports the following attributes:

- `id` - The opaque unique id assigned to the data source by the Grafana server.

grafana_folder

The folder resource allows a folder to be created on a Grafana server.

Example Usage

```
resource "grafana_folder" "collection" {
  title = "Folder Title"
}

resource "grafana_dashboard" "dashboard_in_folder" {
  folder = "${grafana_folder.collection.id}"
  ...
}
```

Argument Reference

The following arguments are supported:

- `title` - (Required) The title of the folder.

Attributes Reference

The resource exports the following attributes:

- `id` - The internal id of the folder in Grafana (only guaranteed to be unique within this Grafana instance). The `id` is used by the `grafana_dashboard` resource to place a dashboard within a folder.
- `uid` - An external id of the folder in Grafana (stable when folders are migrated between Grafana instances). The `uid` is required by several Grafana Folder APIs.

Import

Folders cannot be imported.

grafana_organization

The organization resource allows Grafana organizations and their membership to be created and managed.

Example Usage

```
# Create a Grafana organization with defined membership, creating placeholder
# accounts for users that don't exist.
resource "grafana_organization" "test-org" {
  name          = "Test Organization"
  admin_user    = "admin"
  create_users  = true
  admins        = [
    "admin@example.com"
  ]
  editors       = [
    "editor-01@example.com",
    "editor-02@example.com"
  ]
  viewers       = [
    "viewer-01@example.com",
    "viewer-02@example.com"
  ]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The display name for the Grafana organization created.
- `admin_user` - (Optional) The login name of the configured default admin user (<http://docs.grafana.org/installation/configuration/#admin-user>) for the Grafana installation. If unset, this value defaults to `admin`, the Grafana default. Grafana adds the default admin user to all organizations automatically upon creation, and this parameter keeps Terraform from removing it from organizations.
- `create_users` - (Optional) Whether or not to create Grafana users specified in the organization's membership if they don't already exist in Grafana. If unspecified, this parameter defaults to `true`, creating placeholder users with the `name`, `login`, and `email` set to the email of the user, and a random password. Setting this option to `false` will cause an error to be thrown for any users that do not already exist in Grafana.

This option is particularly useful when integrating Grafana with external authentication services such as `auth.github` (<http://docs.grafana.org/installation/configuration/#auth-github>) and `auth.google` (<http://docs.grafana.org/installation/configuration/#auth-google>).

- `admins` - (Optional) A list of email addresses corresponding to users who should be given `admin` access to the organization. Note: users specified here must already exist in Grafana unless 'create_users' is set to true.

- `editors` - (Optional) A list of email addresses corresponding to users who should be given `editor` access to the organization. Note: users specified here must already exist in Grafana unless `'create_users'` is set to true.
- `viewers` - (Optional) A list of email addresses corresponding to users who should be given `viewer` access to the organization. Note: users specified here must already exist in Grafana unless `'create_users'` is set to true.

A user can only be listed under one role-group for an organization, listing the same user under multiple roles will cause an error to be thrown.

Note - Users specified for each role-group (`admins` , `editors` , `viewers`) should be listed in ascending alphabetical order (A-Z). By defining users in alphabetical order, Terraform is prevented from detecting unnecessary changes when comparing the list of defined users in the resource to the (ordered) list returned by the Grafana API.

Attributes Reference

The following attributes are exported:

- `org_id` - The organization id assigned to this organization by Grafana.

Import

Existing organizations can be imported using the organization id obtained from the Grafana Web UI under 'Server Admin'.

```
$ terraform import grafana_organization.org_name {org_id}
```