# » google_billing_account

Use this data source to get information about a Google Billing Account.

```
data "google_billing_account" "acct" {
  display_name = "My Billing Account"
  open         = true
}

resource "google_project" "my_project" {
  name       = "My Project"
  project_id = "your-project-id"
  org_id     = "1234567"

  billing_account = "${data.google_billing_account.acct.id}"
}
```

## » Argument Reference

The arguments of this data source act as filters for querying the available billing accounts. The given filters must match exactly one billing account whose data will be exported as attributes. The following arguments are supported:

- `billing_account` (Optional) - The name of the billing account in the form `{billing_account_id}` or `billingAccounts/{billing_account_id}`.
- `display_name` (Optional) - The display name of the billing account.
- `open` (Optional) - `true` if the billing account is open, `false` if the billing account is closed.

**NOTE:** One of `billing_account` or `display_name` must be specified.

## » Attributes Reference

The following additional attributes are exported:

- `id` - The billing account ID.
- `name` - The resource name of the billing account in the form `billingAccounts/{billing_account_id}`.
- `project_ids` - The IDs of any projects associated with the billing account.

# » google_client_config

Use this data source to access the configuration of the Google Cloud provider.

## » Example Usage

```
data "google_client_config" "current" {}

output "project" {
  value = "${data.google_client_config.current.project}"
}
```

## » Argument Reference

There are no arguments available for this data source.

## » Attributes Reference

In addition to the arguments listed above, the following attributes are exported:

- `project` - The ID of the project to apply any resources to.
- `region` - The region to operate under.

# » google_cloudfunctions_function

Get information about a Google Cloud Function. For more information see the official documentation and API.

## » Example Usage

```
data "google_cloudfunctions_function" "my-function" {
  name = "function"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of a Cloud Function.

---

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) The region in which the resource belongs. If it is not provided, the provider region is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `name` - The name of the Cloud Function.
- `source_archive_bucket` - The GCS bucket containing the zip archive which contains the function.
- `source_archive_object` - The source archive object (file) in archive bucket.
- `description` - Description of the function.
- `available_memory_mb` - Available memory (in MB) to the function.
- `timeout` - Function execution timeout (in seconds).
- `entry_point` - Name of a JavaScript function that will be executed when the Google Cloud Function is triggered.
- `trigger_http` - If function is triggered by HTTP, this boolean is set.
- `trigger_bucket` - If function is triggered by bucket, bucket name is set here.
- `trigger_topic` - If function is triggered by Pub/Sub topic, name of topic is set here.
- `https_trigger_url` - If function is triggered by HTTP, trigger URL is set here.
- `labels` - A map of labels applied to this function.

# » google_compute_address

Get the IP address from a static address. For more information see the official API documentation.

## » Example Usage

```
data "google_compute_address" "my_address" {
  name = "foobar"
}

resource "google_dns_record_set" "frontend" {
  name = "frontend.${google_dns_managed_zone.prod.dns_name}"
  type = "A"
  ttl  = 300

  managed_zone = "${google_dns_managed_zone.prod.name}"

  rrdatas = ["${data.google_compute_address.my_address.address}"]
}
```

```
resource "google_dns_managed_zone" "prod" {
  name     = "prod-zone"
  dns_name = "prod.mydomain.com."
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE.

---

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) The Region in which the created address reside. If it is not provided, the provider region is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.
- `address` - The IP of the created resource.
- `status` - Indicates if the address is used. Possible values are: RESERVED or IN_USE.

# » google_compute_default_service_account

Use this data source to retrieve default service account for this project

## » Example Usage

```
data "google_compute_default_service_account" "default" { }

output "default_account" {
  value = "${google_compute_default_service_account.default.email}"
}
```

## » Argument Reference

The following arguments are supported:

- `project` - (Optional) The project ID. If it is not provided, the provider project is used.

## » Attributes Reference

The following attributes are exported:

- `email` - Email address of the default service account used by VMs running in this project

# » google_compute_image

Get information about a Google Compute Image. Check that your service account has the `compute.imageUser` role if you want to share custom images from another project. If you want to use public images, do not forget to specify the dedicated project. For more information see the official documentation and its API.

## » Example Usage

```
data "google_compute_image" "my_image" {
  name    = "debian-9"
  project = "debian-cloud"
}

resource "google_compute_instance" "default" {
  # ...

  boot_disk {
    initialize_params {
      image = "${data.google_compute_image.my_image.self_link}"
    }
  }
}
```

## » Argument Reference

The following arguments are supported:

- name or family - (Required) The name of a specific image or a family. Exactly one of name of family must be specified. If name is specified, it will fetch the corresponding image. If family is specified, it will returns the latest image that is part of an image family and is not deprecated.

---

- project - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used. If you are using a public base image, be sure to specify the correct Image Project.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- self_link - The URI of the image.
- name - The name of the image.
- family - The family name of the image.
- disk_size_gb - The size of the image when restored onto a persistent disk in gigabytes.
- archive_size_bytes - The size of the image tar.gz archive stored in Google Cloud Storage in bytes.
- image_id - The unique identifier for the image.
- image_encryption_key_sha256 - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this image.
- source_image_id - The ID value of the image used to create this image.
- source_disk - The URL of the source disk used to create this image.
- source_disk_encryption_key_sha256 - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this image.
- source_disk_id - The ID value of the disk used to create this image.
- creation_timestamp - The creation timestamp in RFC3339 text format.
- description - An optional description of this image.
- labels - A map of labels applied to this image.
- label_fingerprint - A fingerprint for the labels being applied to this image.
- licenses - A list of applicable license URI.
- status - The status of the image. Possible values are **FAILED**, **PENDING**, or **READY**.

# » google_compute_forwarding_rule

Get a forwarding rule within GCE from its name.

## » Example Usage

```
data "google_compute_forwarding_rule" "my-forwarding-rule" {
  name = "forwarding-rule-us-east1"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the forwarding rule.

---

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) The region in which the resource belongs. If it is not provided, the project region is used.

## » Attributes Reference

In addition to the arguments listed above, the following attributes are exported:

- `description` - Description of this forwarding rule.
- `network` - Network of this forwarding rule.
- `subnetwork` - Subnetwork of this forwarding rule.
- `ip_address` - IP address of this forwarding rule.
- `ip_protocol` - IP protocol of this forwarding rule.
- `ports` - List of ports to use for internal load balancing, if this forwarding rule has any.
- `port_range` - Port range, if this forwarding rule has one.
- `target` - URL of the target pool, if this forwarding rule has one.
- `backend_service` - Backend service, if this forwarding rule has one.
- `load_balancing_scheme` - Type of load balancing of this forwarding rule.
- `region` - Region of this forwarding rule.
- `self_link` - The URI of the resource.

# » google_compute_global_address

Get the IP address from a static address reserved for a Global Forwarding Rule which are only used for HTTP load balancing. For more information see the official API documentation.

## » Example Usage

```
data "google_compute_global_address" "my_address" {
  name = "foobar"
}

resource "google_dns_record_set" "frontend" {
  name = "lb.${google_dns_managed_zone.prod.dns_name}"
  type = "A"
  ttl  = 300

  managed_zone = "${google_dns_managed_zone.prod.name}"

  rrdatas = ["${data.google_compute_global_address.my_address.address}"]
}

resource "google_dns_managed_zone" "prod" {
  name     = "prod-zone"
  dns_name = "prod.mydomain.com."
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE.

---

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.
- `address` - The IP of the created resource.

- `status` - Indicates if the address is used. Possible values are: RESERVED or IN_USE.

# » google_compute_network

Get a network within GCE from its name.

## » Example Usage

```
data "google_compute_network" "my-network" {
  name = "default-us-east1"
}
```

## » Argument Reference

The following arguments are supported:

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

---

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following attributes are exported:

- `network` - The network name or resource link to the parent network of this network.
- `description` - Description of this network.
- `gateway_ipv4` - The IP address of the gateway.
- `subnetworks_self_links` - the list of subnetworks which belong to the network
- `self_link` - The URI of the resource.

# » google_bigquery_dataset

Creates a dataset resource for Google BigQuery. For more information see the official documentation and API.

## » Example Usage

```
resource "google_bigquery_dataset" "default" {
  dataset_id                  = "foo"
  friendly_name               = "test"
  description                 = "This is a test description"
  location                    = "EU"
  default_table_expiration_ms = 3600000

  labels {
    env = "default"
  }
}
```

## » Argument Reference

The following arguments are supported:

- `dataset_id` - (Required) A unique ID for the resource. Changing this forces a new resource to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `friendly_name` - (Optional) A descriptive name for the dataset.
- `description` - (Optional) A user-friendly description of the dataset.
- `location` - (Optional) The geographic location where the dataset should reside.

  Possible values include `EU` and `US`. The default value is `US`.

  Changing this forces a new resource to be created.
- `default_table_expiration_ms` - (Optional) The default lifetime of all tables in the dataset, in milliseconds. The minimum value is 3600000 milliseconds (one hour).

  Once this property is set, all newly-created tables in the dataset will have an expirationTime property set to the creation time plus the value in this property, and changing the value will only affect new tables, not existing ones. When the expirationTime for a given table is reached, that table will be deleted automatically. If a table's expirationTime is modified or removed before the table expires, or if you provide an explicit expirationTime when creating a table, that value takes precedence over the default expiration time indicated by this property.
- `labels` - (Optional) A mapping of labels to assign to the resource.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.
- `etag` - A hash of the resource.
- `creation_time` - The time when this dataset was created, in milliseconds since the epoch.
- `last_modified_time` - The date when this dataset or any of its tables was last modified, in milliseconds since the epoch.

## » Import

BigQuery datasets can be imported using the `project` and `dataset_id`, e.g.

```
$ terraform import google_bigquery_dataset.default gcp-project:foo
```

# » google_bigquery_table

Creates a table resource in a dataset for Google BigQuery. For more information see the official documentation and API.

## » Example Usage

```
resource "google_bigquery_dataset" "default" {
  dataset_id                  = "foo"
  friendly_name               = "test"
  description                 = "This is a test description"
  location                    = "EU"
  default_table_expiration_ms = 3600000

  labels {
    env = "default"
  }
}

resource "google_bigquery_table" "default" {
  dataset_id = "${google_bigquery_dataset.default.dataset_id}"
  table_id   = "bar"

  time_partitioning {
    type = "DAY"
  }

  labels {
    env = "default"
  }

  schema = "${file("schema.json")}"
}
```

## » Argument Reference

The following arguments are supported:

- `dataset_id` - (Required) The dataset ID to create the table in. Changing this forces a new resource to be created.
- `table_id` - (Required) A unique ID for the resource. Changing this forces a new resource to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `description` - (Optional) The field description.
- `expiration_time` - (Optional) The time when this table expires, in milliseconds since the epoch. If not present, the table will persist indefinitely. Expired tables will be deleted and their storage reclaimed.
- `friendly_name` - (Optional) A descriptive name for the table.
- `labels` - (Optional) A mapping of labels to assign to the resource.
- `schema` - (Optional) A JSON schema for the table.
- `time_partitioning` - (Optional) If specified, configures time-based partitioning for this table. Structure is documented below.
- `view` - (Optional) If specified, configures this table as a view. Structure is documented below.

The `time_partitioning` block supports:

- `expiration_ms` - (Optional) Number of milliseconds for which to keep the storage for a partition.
- `type` - (Required) The only type supported is DAY, which will generate one partition per day based on data loading time.

The `view` block supports:

- `query` - (Required) A query that BigQuery executes when the view is referenced.
- `use_legacy_sql` - (Optional) Specifies whether to use BigQuery's legacy SQL for this view. The default value is true. If set to false, the view will use BigQuery's standard SQL.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `creation_time` - The time when this table was created, in milliseconds since the epoch.
- `etag` - A hash of the resource.
- `last_modified_time` - The time when this table was last modified, in milliseconds since the epoch.
- `location` - The geographic location where the table resides. This value is inherited from the dataset.
- `num_bytes` - The size of this table in bytes, excluding any data in the streaming buffer.
- `num_long_term_bytes` - The number of bytes in the table that are considered "long-term storage".
- `num_rows` - The number of rows of data in this table, excluding any data in the streaming buffer.
- `self_link` - The URI of the created resource.
- `type` - Describes the table type.

## » Import

BigQuery tables can be imported using the `project`, `dataset_id`, and `table_id`, e.g.

```
$ terraform import google_bigquery_table.default gcp-project:foo.bar
```

# » google_bigtable_instance

Creates a Google Bigtable instance. For more information see the official documentation and API.

## » Example Usage

```
resource "google_bigtable_instance" "instance" {
  name         = "tf-instance"
  cluster_id   = "tf-instance-cluster"
  zone         = "us-central1-b"
  num_nodes    = 3
  storage_type = "HDD"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Bigtable instance.
- `cluster_id` - (Required) The name of the Bigtable instance's cluster.
- `zone` - (Required) The zone to create the Bigtable instance in. Zones that support Bigtable instances are noted on the Cloud Locations page.
- `num_nodes` - (Optional) The number of nodes in your Bigtable instance. Minimum of `3` for a `PRODUCTION` instance. Cannot be set for a `DEVELOPMENT` instance.
- `instance_type` - (Optional) The instance type to create. One of `"DEVELOPMENT"` or `"PRODUCTION"`. Defaults to `PRODUCTION`.
- `storage_type` - (Optional) The storage type to use. One of `"SSD"` or `"HDD"`. Defaults to `SSD`.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `display_name` - (Optional) The human-readable display name of the Bigtable instance. Defaults to the instance `name`.

## » Attributes Reference

Only the arguments listed above are exposed as attributes.

# » google_bigtable_table

Creates a Google Bigtable table inside an instance. For more information see the official documentation and API.

### » Example Usage

```
resource "google_bigtable_instance" "instance" {
  name         = "tf-instance"
  cluster_id   = "tf-instance-cluster"
  zone         = "us-central1-b"
  num_nodes    = 3
  storage_type = "HDD"
}

resource "google_bigtable_table" "table" {
  name          = "tf-table"
  instance_name = "${google_bigtable_instance.instance.name}"
  split_keys    = ["a", "b", "c"]
}
```

### » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the table.
- `instance_name` - (Required) The name of the Bigtable instance.
- `split_keys` - (Optional) A list of predefined keys to split the table on.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

### » Attributes Reference

Only the arguments listed above are exposed as attributes.

## » google_cloudfunctions_function

Creates a new Cloud Function. For more information see the official documentation and API.

### » Example Usage

```
resource "google_storage_bucket" "bucket" {
  name = "test-bucket"
}
```

```
resource "google_storage_bucket_object" "archive" {
  name   = "index.zip"
  bucket = "${google_storage_bucket.bucket.name}"
  source = "./path/to/zip/file/which/contains/code"
}

resource "google_cloudfunctions_function" "function" {
  name                  = "function-test"
  description           = "My function"
  available_memory_mb   = 128
  source_archive_bucket = "${google_storage_bucket.bucket.name}"
  source_archive_object = "${google_storage_bucket_object.archive.name}"
  trigger_http          = true
  timeout               = 60
  entry_point           = "helloGET"
  labels {
    my-label = "my-label-value"
  }
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A user-defined name of the function. Function names must be unique globally.
- `source_archive_bucket` - (Required) The GCS bucket containing the zip archive which contains the function.
- `source_archive_object` - (Required) The source archive object (file) in archive bucket.

---

- `description` - (Optional) Description of the function.
- `available_memory_mb` - (Optional) Memory (in MB), available to the function. Default value is 256MB. Allowed values are: 128MB, 256MB, 512MB, 1024MB, and 2048MB.
- `timeout` - (Optional) Timeout (in seconds) for the function. Default value is 60 seconds. Cannot be more than 540 seconds.
- `entry_point` - (Optional) Name of a JavaScript function that will be executed when the Google Cloud Function is triggered.

- `trigger_http` - (Optional) Boolean variable. Any HTTP request (of a supported type) to the endpoint will trigger function execution. Supported HTTP request types are: POST, PUT, GET, DELETE, and OPTIONS. Endpoint is returned as `https_trigger_url`. Cannot be used with `trigger_bucket` and `trigger_topic`.
- `trigger_bucket` - (Optional) Google Cloud Storage bucket name. Every change in files in this bucket will trigger function execution. Cannot be used with `trigger_http` and `trigger_topic`.
- `trigger_topic` - (Optional) Name of Pub/Sub topic. Every message published in this topic will trigger function execution with message contents passed as input data. Cannot be used with `trigger_http` and `trigger_bucket`.
- `labels` - (Optional) A set of key/value label pairs to assign to the function.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `https_trigger_url` - URL which triggers function execution. Returned only if `trigger_http` is used.
- `project` - Project of the function. If it is not provided, the provider project is used.
- `region` - Region of function. Currently can be only "us-central1". If it is not provided, the provider region is used.

## » Import

Functions can be imported using the `name`, e.g.

```
$ terraform import google_cloudfunctions_function.default function-test
```

# » google_folder

Allows management of a Google Cloud Platform folder. For more information see the official documentation and API.

A folder can contain projects, other folders, or a combination of both. You can use folders to group projects under an organization in a hierarchy. For example, your organization might contain multiple departments, each with its own set of Cloud Platform resources. Folders allows you to group these resources on a per-department basis. Folders are used to group resources that share common IAM policies.

Folders created live inside an Organization. See the Organization documentation for more details.

The service account used to run Terraform when creating a `google_folder` resource must have `roles/resourcemanager.folderCreator`. See the Access Control for Folders Using IAM doc for more information.

## » Example Usage

```
# Top-level folder under an organization.
resource "google_folder" "department1" {
  display_name = "Department 1"
  parent     = "organizations/1234567"
}

# Folder nested under another folder.
resource "google_folder" "team-abc" {
  display_name = "Team ABC"
  parent     = "${google_folder.department1.name}"
}
```

## » Argument Reference

The following arguments are supported:

- `display_name` - (Required) The folder's display name. A folder's display name must be unique amongst its siblings, e.g. no two folders with the same parent can share the same display name. The display name must start and end with a letter or digit, may contain letters, digits, spaces, hyphens and underscores and can be no longer than 30 characters.
- `parent` - (Required) The resource name of the parent Folder or Organization. Must be of the form `folders/{folder_id}` or `organizations/{org_id}`.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `name` - The resource name of the Folder. Its format is folders/{folder_id}.
- `lifecycle_state` - The lifecycle state of the folder such as `ACTIVE` or `DELETE_REQUESTED`.

- `create_time` - Timestamp when the Folder was created. Assigned by the server. A timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds. Example: "2014-10-02T15:01:23.045123456Z".

## » Import

Folders can be imported using the folder autogenerated `name`, e.g.

```
# Both syntaxes are valid
$ terraform import google_folder.department1 1234567
$ terraform import google_folder.department1 folders/1234567
```

# » google_folder_iam_binding

Allows creation and management of a single binding within IAM policy for an existing Google Cloud Platform folder.

**Note:** This resource *must not* be used in conjunction with `google_folder_iam_policy` or they will fight over what your policy should be.

## » Example Usage

```
resource "google_folder" "department1" {
  display_name = "Department 1"
  parent       = "organizations/1234567"
}

resource "google_folder_iam_binding" "admin" {
  folder  = "${google_folder.department1.name}"
  role    = "roles/editor"

  members = [
    "user:jane@example.com",
  ]
}
```

## » Argument Reference

The following arguments are supported:

- `folder` - (Required) The resource name of the folder the policy is attached to. Its format is folders/{folder_id}.

- `members` (Required) - An array of identites that will be granted the privilege in the `role`. Each entry can have one of the following values:
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role -` (Required) The role that should be applied. Only one `google_folder_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the folder's IAM policy.

## » Import

IAM binding imports use space-delimited identifiers; first the resource in question and then the role. These bindings can be imported using the `folder` and role, e.g.

```
$ terraform import google_folder_iam_binding.viewer "folder-name roles/viewer"
```

# » google_folder_iam_member

Allows creation and management of a single member for a single binding within the IAM policy for an existing Google Cloud Platform folder.

**Note:** This resource *must not* be used in conjunction with `google_folder_iam_policy` or they will fight over what your policy should be. Similarly, roles controlled by `google_folder_iam_binding` should not be assigned to using `google_folder_iam_member`.

## » Example Usage

```
resource "google_folder" "department1" {
  display_name = "Department 1"
  parent       = "organizations/1234567"
}

resource "google_folder_iam_member" "admin" {
  folder  = "${google_folder.department1.name}"
  role    = "roles/editor"
  member  = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `folder` - (Required) The resource name of the folder the policy is attached to. Its format is folders/{folder_id}.
- `member` - (Required) The identity that will be granted the privilege in the `role`. This field can have one of the following values:
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the folder's IAM policy.

### » Import

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account. This member resource can be imported using the `folder`, role, and account e.g.

```
$ terraform import google_folder_iam_member.my_project "folder-name roles/viewer foo@example
```

## » google_folder_iam_policy

Allows creation and management of the IAM policy for an existing Google Cloud Platform folder.

### » Example Usage

```
resource "google_folder_iam_policy" "folder_admin_policy" {
  folder      = "${google_folder.department1.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}

resource "google_folder" "department1" {
  display_name = "Department 1"
  parent       = "organizations/1234567"
}

data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"

    members = [
      "user:jane@example.com",
    ]
  }
}
```

### » Argument Reference

The following arguments are supported:

- `folder` - (Required) The resource name of the folder the policy is attached to. Its format is folders/{folder_id}.

- `policy_data` - (Required) The `google_iam_policy` data source that represents the IAM policy that will be applied to the folder. This policy overrides any existing policy applied to the folder.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the folder's IAM policy. `etag` is used for optimistic concurrency control as a way to help prevent simultaneous updates of a policy from overwriting each other.

# » google_folder_organization_policy

Allows management of Organization policies for a Google Folder. For more information see the official documentation and API.

## » Example Usage

To set policy with a boolean constraint:

```
resource "google_folder_organization_policy" "serial_port_policy" {
  folder     = "folders/123456789"
  constraint = "compute.disableSerialPortAccess"

  boolean_policy {
    enforced = true
  }
}
```

To set a policy with a list contraint:

```
resource "google_folder_organization_policy" "services_policy" {
  folder     = "folders/123456789"
  constraint = "serviceuser.services"

  list_policy {
    allow {
      all = true
    }
  }
}
```

Or to deny some services, use the following instead:

```
resource "google_folder_organization_policy" "services_policy" {
  folder     = "folders/123456789"
  constraint = "serviceuser.services"

  list_policy {
    suggested_values = "compute.googleapis.com"

    deny {
      values = ["cloudresourcemanager.googleapis.com"]
    }
  }
}
```

## » Argument Reference

The following arguments are supported:

- `folder` - (Required) The resource name of the folder to set the policy for. Its format is folders/{folder_id}.
- `constraint` - (Required) The name of the Constraint the Policy is configuring, for example, `serviceuser.services`. Check out the complete list of available constraints.

---

- `version` - (Optional) Version of the Policy. Default version is 0.
- `boolean_policy` - (Optional) A boolean policy is a constraint that is either enforced or not. Structure is documented below.
- `list_policy` - (Optional) A policy that can define specific values that are allowed or denied for the given constraint. It can also be used to allow or deny all values. Structure is documented below.

---

The `boolean_policy` block supports:

- `enforced` - (Required) If true, then the Policy is enforced. If false, then any configuration is acceptable.

The `list_policy` block supports:

- `allow` or `deny` - (Optional) One or the other must be set.
- `suggested_values` - (Optional) The Google Cloud Console will try to default to a configuration that matches the value specified in this field.

The `allow` or `deny` blocks support:

- `all` - (Optional) The policy allows or denies all values.

- `values` - (Optional) The policy can define specific values that are allowed or denied.

### » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the organization policy. `etag` is used for optimistic concurrency control as a way to help prevent simultaneous updates of a policy from overwriting each other.
- `update_time` - (Computed) The timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds, representing when the variable was last updated. Example: "2016-10-09T12:33:37.578138407Z".

## » google_organization_policy

Allows management of Organization policies for a Google Organization. For more information see the official documentation and API.

### » Example Usage

To set policy with a boolean constraint:

```
resource "google_organization_policy" "serial_port_policy" {
  org_id     = "123456789"
  constraint = "compute.disableSerialPortAccess"

  boolean_policy {
    enforced = true
  }
}
```

To set a policy with a list contraint:

```
resource "google_organization_policy" "services_policy" {
  org_id     = "123456789"
  constraint = "serviceuser.services"

  list_policy {
    allow {
      all = true
    }
  }
}
```

Or to deny some services, use the following instead:

```
resource "google_organization_policy" "services_policy" {
  org_id     = "123456789"
  constraint = "serviceuser.services"

  list_policy {
    suggested_values = "compute.googleapis.com"

    deny {
      values = ["cloudresourcemanager.googleapis.com"]
    }
  }
}
```

## » Argument Reference

The following arguments are supported:

- `org_id` - (Required) The numeric ID of the organization to set the policy for.
- `constraint` - (Required) The name of the Constraint the Policy is configuring, for example, `serviceuser.services`. Check out the complete list of available constraints.

---

- `version` - (Optional) Version of the Policy. Default version is 0.
- `boolean_policy` - (Optional) A boolean policy is a constraint that is either enforced or not. Structure is documented below.
- `list_policy` - (Optional) A policy that can define specific values that are allowed or denied for the given constraint. It can also be used to allow or deny all values. Structure is documented below.

---

The `boolean_policy` block supports:

- `enforced` - (Required) If true, then the Policy is enforced. If false, then any configuration is acceptable.

The `list_policy` block supports:

- `allow` or `deny` - (Optional) One or the other must be set.
- `suggested_values` - (Optional) The Google Cloud Console will try to default to a configuration that matches the value specified in this field.

The `allow` or `deny` blocks support:

- `all` - (Optional) The policy allows or denies all values.
- `values` - (Optional) The policy can define specific values that are allowed or denied.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the organization policy. `etag` is used for optimistic concurrency control as a way to help prevent simultaneous updates of a policy from overwriting each other.
- `update_time` - (Computed) The timestamp in RFC3339 UTC ”Zulu” format, accurate to nanoseconds, representing when the variable was last updated. Example: ”2016-10-09T12:33:37.578138407Z”.

## » Import

Organization Policies can be imported using the `org_id` and the `contraint`, e.g.

```
$ terraform import google_organization_policy.services_policy 123456789:constraints/serviceu
```

# » google_organization_iam_binding

Allows creation and management of a single binding within IAM policy for an existing Google Cloud Platform Organization.

**Note:** This resource **must not** be used in conjunction with `google_organization_iam_member` for the **same role** or they will fight over what your policy should be.

## » Example Usage

```
resource "google_organization_iam_binding" "binding" {
  org_id = "123456789"
  role    = "roles/browser"

  members = [
    "user:jane@example.com",
  ]
}
```

## » Argument Reference

The following arguments are supported:

- `org_id` - (Required) The numeric ID of the organization in which you want to create a custom role.
- `role` - (Required) The role that should be applied. Only one `google_organization_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `members` - (Required) A list of users that the role should apply to.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the organization's IAM policy.

## » Import

IAM binding imports use space-delimited identifiers; first the resource in question and then the role. These bindings can be imported using the `org_id` and role, e.g.

```
$ terraform import google_organization_iam_binding.my_org "your-org-id roles/viewer"
```

# » google_organization_iam_custom_role

Allows management of a customized Cloud IAM organization role. For more information see the official documentation and API.

## » Example Usage

This snippet creates a customized IAM organization role.

```
resource "google_organization_iam_custom_role" "my-custom-role" {
  role_id     = "myCustomRole"
  org_id      = "123456789"
  title       = "My Custom Role"
  description = "A description"
  permissions = ["iam.roles.list", "iam.roles.create", "iam.roles.delete"]
}
```

## » Argument Reference

The following arguments are supported:

- `role_id` - (Required) The role id to use for this role.
- `org_id` - (Required) The numeric ID of the organization in which you want to create a custom role.
- `title` - (Required) A human-readable title for the role.
- `permissions` (Required) The names of the permissions this role grants when bound in an IAM policy. At least one permission must be specified.
- `stage` - (Optional) The current launch stage of the role. Defaults to `GA`. List of possible stages is here.
- `description` - (Optional) A human-readable description for the role.
- `deleted` - (Optional) The current deleted state of the role. Defaults to `false`.

## » Import

Customized IAM organization role can be imported using their URI, e.g.

```
$ terraform import google_organization_iam_custom_role.my-custom-role organizations/12345678
```

# » google_organization_iam_member

Allows creation and management of a single member for a single binding within the IAM policy for an existing Google Cloud Platform Organization.

**Note:** This resource **must not** be used in conjunction with `google_organization_iam_binding` for the **same role** or they will fight over what your policy should be.

## » Example Usage

```
resource "google_organization_iam_member" "binding" {
  org_id = "0123456789"
  role    = "roles/editor"
  member  = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `org_id` - (Required) The numeric ID of the organization in which you want to create a custom role.
- `role` - (Required) The role that should be applied. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `member` - (Required) The user that the role should apply to.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the organization's IAM policy.

## » Import

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account. This member resource can be imported using the `org_id`, role, and account e.g.

```
$ terraform import google_organization_iam_member.my_org "your-org-id roles/viewer foo@examp
```

# » google_organization_iam_policy

Allows management of the entire IAM policy for an existing Google Cloud Platform Organization.

**Warning:** New organizations have several default policies which will, without extreme caution, be **overwritten** by use of this resource. The safest alternative is to use multiple `google_organization_iam_binding` resources. It is easy to use this resource to remove your own access to an organization, which will require a call to Google Support to have fixed, and can take multiple days to resolve. If you do use this resource, the best way to be sure that you are not making dangerous changes is to start by importing your existing policy, and examining the diff very closely.

**Note:** This resource **must not** be used in conjunction with `google_organization_iam_member` or `google_organization_iam_binding` or they will fight over what your policy should be.

### » Example Usage

```
resource "google_organization_iam_policy" "policy" {
  org_id = "123456789"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}

data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"

    members = [
      "user:jane@example.com",
    ]
  }
}
```

### » Argument Reference

The following arguments are supported:

- `org_id` - (Required) The numeric ID of the organization in which you want to create a custom role.
- `policy_data` - (Required) The `google_iam_policy` data source that represents the IAM policy that will be applied to the organization. This policy overrides any existing policy applied to the organization.

### » Import

```
$ terraform import google_organization_iam_policy.my_org your-org-id
```

## » google_project

Allows creation and management of a Google Cloud Platform project.

Projects created with this resource must be associated with an Organization. See the Organization documentation for more details.

The service account used to run Terraform when creating a `google_project` resource must have `roles/resourcemanager.projectCreator`. See the Access Control for Organizations Using IAM doc for more information.

Note that prior to 0.8.5, `google_project` functioned like a data source, meaning any project referenced by it had to be created and managed outside Terraform.

As of 0.8.5, `google_project` functions like any other Terraform resource, with Terraform creating and managing the project. To replicate the old behavior, either:

- Use the project ID directly in whatever is referencing the project, using the google_project_iam_policy to replace the old `policy_data` property.
- Use the import functionality to import your pre-existing project into Terraform, where it can be referenced and used just like always, keeping in mind that Terraform will attempt to undo any changes made outside Terraform.

It's important to note that any project resources that were added to your Terraform config prior to 0.8.5 will continue to function as they always have, and will not be managed by Terraform. Only newly added projects are affected.

## » Example Usage

```
resource "google_project" "my_project" {
  name = "My Project"
  project_id = "your-project-id"
  org_id     = "1234567"
}
```

To create a project under a specific folder

```
resource "google_project" "my_project-in-a-folder" {
  name = "My Project"
  project_id = "your-project-id"
  folder_id  = "${google_folder.department1.name}"
}

resource "google_folder" "department1" {
  display_name = "Department 1"
  parent     = "organizations/1234567"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The display name of the project.
- `project_id` - (Required) The project ID. Changing this forces a new project to be created.
- `org_id` - (Optional) The numeric ID of the organization this project belongs to. Changing this forces a new project to be created. Only one of

org_id or folder_id may be specified. If the org_id is specified then the project is created at the top level. Changing this forces the project to be migrated to the newly specified organization.

- folder_id - (Optional) The numeric ID of the folder this project should be created under. Only one of org_id or folder_id may be specified. If the folder_id is specified, then the project is created under the specified folder. Changing this forces the project to be migrated to the newly specified folder.
- billing_account - (Optional) The alphanumeric ID of the billing account this project belongs to. The user or service account performing this operation with Terraform must have Billing Account Administrator privileges (roles/billing.admin) in the organization. See Google Cloud Billing API Access Control for more details.
- skip_delete - (Optional) If true, the Terraform resource can be deleted without deleting the Project via the Google API.
- policy_data - (Deprecated) The IAM policy associated with the project. This argument is no longer supported, and will be removed in a future version of Terraform. It should be replaced with a google_project_iam_policy resource.
- labels - (Optional) A set of key/value label pairs to assign to the project.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- number - The numeric identifier of the project.
- policy_etag - (Deprecated) The etag of the project's IAM policy, used to determine if the IAM policy has changed. Please use google_project_iam_policy's etag property instead; future versions of Terraform will remove the policy_etag attribute

## » Import

Projects can be imported using the project_id, e.g.

```
$ terraform import google_project.my_project your-project-id
```

## » google_project_iam_binding

Allows creation and management of a single binding within IAM policy for an existing Google Cloud Platform project.

**Note:** This resource *must not* be used in conjunction with `google_project_iam_policy` or they will fight over what your policy should be.

### » Example Usage

```
resource "google_project_iam_binding" "project" {
  project = "your-project-id"
  role    = "roles/editor"

  members = [
    "user:jane@example.com",
  ]
}
```

### » Argument Reference

The following arguments are supported:

- `members` (Required) - An array of identites that will be granted the privilege in the `role`. Each entry can have one of the following values:
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_project_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `project` - (Optional) The project ID. If not specified, uses the ID of the project configured with the provider.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the project's IAM policy.

## » Import

IAM binding imports use space-delimited identifiers; first the resource in question and then the role. These bindings can be imported using the `project_id` and role, e.g.

```
$ terraform import google_project_iam_binding.my_project "your-project-id roles/viewer"
```

# » google_project_iam_member

Allows creation and management of a single member for a single binding within the IAM policy for an existing Google Cloud Platform project.

**Note:** This resource *must not* be used in conjunction with `google_project_iam_policy` or they will fight over what your policy should be. Similarly, roles controlled by `google_project_iam_binding` should not be assigned to using `google_project_iam_member`.

## » Example Usage

```
resource "google_project_iam_member" "project" {
  project = "your-project-id"
  role    = "roles/editor"
  member  = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `member` - (Required) The identity that will be granted the privilege in the `role`. This field can have one of the following values:
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.

  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.

- `role` - (Required) The role that should be applied. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `project` - (Optional) The project ID. If not specified, uses the ID of the project configured with the provider.

### » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the project's IAM policy.

### » Import

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account. This member resource can be imported using the `project_id`, role, and account e.g.

```
$ terraform import google_project_iam_member.my_project "your-project-id roles/viewer foo@ex
```

## » google_project_iam_policy

Allows creation and management of an IAM policy for an existing Google Cloud Platform project.

**Be careful!** You can accidentally lock yourself out of your project using this resource. Proceed with caution.

### » Example Usage

```
resource "google_project_iam_policy" "project" {
  project     = "your-project-id"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"

    members = [
      "user:jane@example.com",
    ]
  }
}
```

## » Argument Reference

The following arguments are supported:

- `project` - (Required) The project ID. Changing this forces a new resource to be created.
- `policy_data` - (Required) The `google_iam_policy` data source that represents the IAM policy that will be applied to the project. The policy will be merged with any existing policy applied to the project.

  Changing this updates the policy.

  Deleting this removes the policy, but leaves the original project policy intact. If there are overlapping `binding` entries between the original project policy and the data source policy, they will be removed.
- `authoritative` - (DEPRECATED) (Optional) A boolean value indicating if this policy should overwrite any existing IAM policy on the project. When set to true, **any policies not in your config file will be removed**. This can **lock you out** of your project until an Organization Administrator grants you access again, so please exercise caution. If this argument is `true` and you want to delete the resource, you must set the `disable_project` argument to `true`, acknowledging that the project will be inaccessible to anyone but the Organization Admins, as it will no longer have an IAM policy. Rather than using this, you should use `google_project_iam_policy_binding` and `google_project_iam_policy_member`.
- `disable_project` - (DEPRECATED) (Optional) A boolean value that must be set to `true` if you want to delete a `google_project_iam_policy` that is authoritative.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the project's IAM policy.
- `restore_policy` - (DEPRECATED) (Computed) The IAM policy that will be restored when a non-authoritative policy resource is deleted.

## » Import

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `project_id` e.g.

```
$ terraform import google_project_iam_policy.my_project your-project-id
```

# » google_project_iam_custom_role

Allows management of a customized Cloud IAM project role. For more information see the official documentation and API.

## » Example Usage

This snippet creates a customized IAM role.

```
resource "google_project_iam_custom_role" "my-custom-role" {
  role_id     = "myCustomRole"
  title       = "My Custom Role"
  description = "A description"
  permissions = ["iam.roles.list", "iam.roles.create", "iam.roles.delete"]
}
```

## » Argument Reference

The following arguments are supported:

- `role_id` - (Required) The role id to use for this role.
- `title` - (Required) A human-readable title for the role.
- `permissions` (Required) The names of the permissions this role grants when bound in an IAM policy. At least one permission must be specified.

- `project` - (Optional) The project that the service account will be created in. Defaults to the provider project configuration.
- `stage` - (Optional) The current launch stage of the role. Defaults to `GA`. List of possible stages is here.
- `description` - (Optional) A human-readable description for the role.
- `deleted` - (Optional) The current deleted state of the role. Defaults to `false`.

### » Import

Customized IAM project role can be imported using their URI, e.g.

```
$ terraform import google_project_iam_custom_role.my-custom-role projects/my-project/roles/m
```

## » google_project_service

Allows management of a single API service for an existing Google Cloud Platform project.

For a list of services available, visit the API library page or run `gcloud service-management list`.

**Note:** This resource *must not* be used in conjunction with `google_project_services` or they will fight over which services should be enabled.

### » Example Usage

```
resource "google_project_service" "project" {
  project = "your-project-id"
  service = "iam.googleapis.com"
}
```

### » Argument Reference

The following arguments are supported:

- `service` - (Required) The service to enable.
- `project` - (Optional) The project ID. If not provided, the provider project is used.

- `disable_on_destroy` - (Optional) If true, disable the service when the terraform resource is destroyed. Defaults to true. May be useful in the event that a project is long-lived but the infrastructure running in that project changes frequently.

# » google_project_services

Allows management of enabled API services for an existing Google Cloud Platform project. Services in an existing project that are not defined in the config will be removed.

For a list of services available, visit the API library page or run `gcloud service-management list`.

**Note:** This resource attempts to be the authoritative source on which APIs are enabled, which can lead to conflicts when certain APIs or actions enable other APIs. To just ensure that a specific API is enabled, use the google_project_service resource.

## » Example Usage

```
resource "google_project_services" "project" {
  project = "your-project-id"
  services   = ["iam.googleapis.com", "cloudresourcemanager.googleapis.com"]
}
```

## » Argument Reference

The following arguments are supported:

- `project` - (Required) The project ID. Changing this forces Terraform to attempt to disable all previously managed API services in the previous project.
- `services` - (Required) The list of services that are enabled. Supports update.

## » Import

Project services can be imported using the `project_id`, e.g.

```
$ terraform import google_project_services.my_project your-project-id
```

# » google_project_usage_export_bucket

Sets up a usage export bucket for a particular project. A usage export bucket is a pre-configured GCS bucket which is set up to receive daily and monthly reports of the GCE resources used.

For more information see the Docs and for further details, the API Documentation.

## » Example Usage

```
resource "google_project_usage_export_bucket" "export" {
  project      = "foo"
  bucket_name  = "bar"
}
```

## » Argument Reference

- `project`: (Required) The project to set the export bucket on.
- `bucket_name`: (Required) The bucket to store reports in.
- `prefix`: (Optional) A prefix for the reports, for instance, the project name.

## » Note

You should specify only one of these per project. If there are two or more they will fight over which bucket the reports should be stored in. It is safe to have multiple resources with the same backing bucket.

# » google_service_account

Allows management of a Google Cloud Platform service account

## » Example Usage

This snippet creates a service account, then gives it objectViewer permission in a project.

```
resource "google_service_account" "object_viewer" {
  account_id   = "object-viewer"
  display_name = "Object viewer"
}
```

## » Argument Reference

The following arguments are supported:

- `account_id` - (Required) The service account ID. Changing this forces a new service account to be created.
- `display_name` - (Optional) The display name for the service account. Can be updated without creating a new resource.
- `project` - (Optional) The ID of the project that the service account will be created in. Defaults to the provider project configuration.
- `policy_data` - (DEPRECATED, Optional) The `google_iam_policy` data source that represents the IAM policy that will be applied to the service account. The policy will be merged with any existing policy.

  This attribute has been deprecated. Use the google_service_account_iam_* resources instead.

  Deleting this removes the policy declared in Terraform. Any policy bindings associated with the project before Terraform was used are not deleted.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `email` - The e-mail address of the service account. This value should be referenced from any `google_iam_policy` data sources that would grant the service account privileges.
- `name` - The fully-qualified name of the service account.
- `unique_id` - The unique id of the service account.

## » Import

Service accounts can be imported using their URI, e.g.

```
$ terraform import google_service_account.my_sa projects/my-project/serviceAccounts/my-sa@my
```

# » IAM policy for service account

When managing IAM roles, you can treat a service account either as a resource or as an identity. This resource is to add iam policy bindings to a service account resource.

Three different resources help you manage your IAM policy for a service account. Each of these resources serves a different use case:

- `google_service_account_iam_policy`: Authoritative. Sets the IAM policy for the service account and replaces any existing policy already attached.
- `google_service_account_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the service account are preserved.
- `google_service_account_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the service account are preserved.

**Note:** `google_service_account_iam_policy` **cannot** be used in conjunction with `google_service_account_iam_binding` and `google_service_account_iam_member` or they will fight over what your policy should be.

**Note:** `google_service_account_iam_binding` resources **can be** used in conjunction with `google_service_account_iam_member` resources **only if** they do not grant privilege to the same role.

## » google_service_account_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"

    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_service_account_iam_policy" "admin-account-iam" {
    service_account_id = "your-service-account-id"
    policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_service_account_iam_binding

```
resource "google_service_account_iam_binding" "admin-account-iam" {
  service_account_id = "your-service-account-id"
  role        = "roles/editor"

  members = [
    "user:jane@example.com",
  ]
}
```

## » google_service_account_iam_member

```
resource "google_service_account_iam_member" "admin-account-iam" {
  service_account_id = "your-service-account-id"
  role        = "roles/editor"
  member      = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `service_account_id` - (Required) The service account id to apply policy to.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_service_account_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `policy_data` - (Required only by `google_service_account_iam_policy`) The policy data generated by a `google_iam_policy` data source.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the service account IAM policy.

## » Import

Service account IAM resources can be imported using the project, service account email, role and member.

```
$ terraform import google_service_account_iam_policy.admin-account-iam projects/{your-projec

$ terraform import google_service_account_iam_binding.admin-account-iam "projects/{your-proj

$ terraform import google_service_account_iam_member.admin-account-iam "projects/{your-proje
```

# » IAM policy for service account

When managing IAM roles, you can treat a service account either as a resource or as an identity. This resource is to add iam policy bindings to a service account resource.

Three different resources help you manage your IAM policy for a service account. Each of these resources serves a different use case:

- `google_service_account_iam_policy`: Authoritative. Sets the IAM policy for the service account and replaces any existing policy already attached.
- `google_service_account_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the service account are preserved.
- `google_service_account_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the service account are preserved.

**Note:** `google_service_account_iam_policy` **cannot** be used in conjunction with `google_service_account_iam_binding` and `google_service_account_iam_member` or they will fight over what your policy should be.

**Note:** `google_service_account_iam_binding` resources **can be** used in conjunction with `google_service_account_iam_member` resources **only if** they do not grant privilege to the same role.

## » google_service_account_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"

    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_service_account_iam_policy" "admin-account-iam" {
    service_account_id = "your-service-account-id"
    policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_service_account_iam_binding

```
resource "google_service_account_iam_binding" "admin-account-iam" {
  service_account_id = "your-service-account-id"
  role        = "roles/editor"

  members = [
    "user:jane@example.com",
  ]
}
```

## » google_service_account_iam_member

```
resource "google_service_account_iam_member" "admin-account-iam" {
  service_account_id = "your-service-account-id"
  role        = "roles/editor"
  member      = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `service_account_id` - (Required) The service account id to apply policy to.

- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_service_account_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `policy_data` - (Required only by `google_service_account_iam_policy`) The policy data generated by a `google_iam_policy` data source.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the service account IAM policy.

## » Import

Service account IAM resources can be imported using the project, service account email, role and member.

```
$ terraform import google_service_account_iam_policy.admin-account-iam projects/{your-proje

$ terraform import google_service_account_iam_binding.admin-account-iam "projects/{your-proj

$ terraform import google_service_account_iam_member.admin-account-iam "projects/{your-proje
```

# » IAM policy for service account

When managing IAM roles, you can treat a service account either as a resource or as an identity. This resource is to add iam policy bindings to a service account resource.

Three different resources help you manage your IAM policy for a service account. Each of these resources serves a different use case:

- `google_service_account_iam_policy`: Authoritative. Sets the IAM policy for the service account and replaces any existing policy already attached.
- `google_service_account_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the service account are preserved.
- `google_service_account_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the service account are preserved.

**Note:** `google_service_account_iam_policy` **cannot** be used in conjunction with `google_service_account_iam_binding` and `google_service_account_iam_member` or they will fight over what your policy should be.

**Note:** `google_service_account_iam_binding` resources **can be** used in conjunction with `google_service_account_iam_member` resources **only if** they do not grant privilege to the same role.

## » google_service_account_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"

    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_service_account_iam_policy" "admin-account-iam" {
    service_account_id = "your-service-account-id"
    policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_service_account_iam_binding

```
resource "google_service_account_iam_binding" "admin-account-iam" {
  service_account_id = "your-service-account-id"
  role        = "roles/editor"

  members = [
    "user:jane@example.com",
  ]
}
```

## » google_service_account_iam_member

```
resource "google_service_account_iam_member" "admin-account-iam" {
  service_account_id = "your-service-account-id"
  role        = "roles/editor"
  member      = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `service_account_id` - (Required) The service account id to apply policy to.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.

- `role` - (Required) The role that should be applied. Only one `google_service_account_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `policy_data` - (Required only by `google_service_account_iam_policy`) The policy data generated by a `google_iam_policy` data source.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the service account IAM policy.

## » Import

Service account IAM resources can be imported using the project, service account email, role and member.

```
$ terraform import google_service_account_iam_policy.admin-account-iam projects/{your-projec

$ terraform import google_service_account_iam_binding.admin-account-iam "projects/{your-proj

$ terraform import google_service_account_iam_member.admin-account-iam "projects/{your-proje
```

# » google_service_account_key

Creates and manages service account key-pairs, which allow the user to establish identity of a service account outside of GCP. For more information, see the official documentation and API.

## » Example Usage, creating a new Key Pair

```
resource "google_service_account" "acceptance" {
  account_id = "%v"
  display_name = "%v"
}

resource "google_service_account_key" "acceptance" {
  service_account_id = "${google_service_account.acceptance.id}"
  public_key_type = "TYPE_X509_PEM_FILE"
}
```

## » Example Usage, save key in Kubernetes secret

```
resource "google_service_account" "myaccount" {
  account_id   = "myaccount"
  display_name = "My Service Account"
}
resource "google_service_account_key" "mykey" {
  service_account_id = "${google_service_account.myaccount.id}"
}
resource "kubernetes_secret" "google-application-credentials" {
  metadata {
    name = "google-application-credentials"
  }
  data {
    credentials.json = "${base64decode(google_service_account_key.mykey.private_key)}"
  }
}
```

## » Create new Key Pair, encrypting the private key with a PGP Key

```
resource "google_service_account" "acceptance" {
  account_id = "%v"
  display_name = "%v"
}

resource "google_service_account_key" "acceptance" {
  service_account_id = "${google_service_account.acceptance.id}"
  pgp_key = "keybase:keybaseusername"
  public_key_type = "TYPE_X509_PEM_FILE"
}
```

## » Argument Reference

The following arguments are supported:

- `service_account_id` - (Required) The Service account id of the Key Pair.
- `key_algorithm` - (Optional) The algorithm used to generate the key. KEY_ALG_RSA_2048 is the default algorithm. Valid values are listed at ServiceAccountPrivateKeyType (only used on create)
- `public_key_type` (Optional) The output format of the public key requested. X509_PEM is the default output format.

- `private_key_type` (Optional) The output format of the private key. GOOGLE_CREDENTIALS_FILE is the default output format.
- `pgp_key` – (Optional) An optional PGP key to encrypt the resulting private key material. Only used when creating or importing a new key pair. May either be a base64-encoded public key or a `keybase:keybaseusername` string for looking up in Vault.

**NOTE:** a PGP key is not required, however it is strongly encouraged. Without a PGP key, the private key material will be stored in state unencrypted.

## » Attributes Reference

The following attributes are exported in addition to the arguments listed above:

- `name` - The name used for this key pair
- `public_key` - The public key, base64 encoded
- `private_key` - The private key in JSON format, base64 encoded. This is what you normally get as a file when creating service account keys through the CLI or web console. This is only populated when creating a new key, and when no `pgp_key` is provided.
- `private_key_encrypted` – The private key material, base 64 encoded and encrypted with the given `pgp_key`. This is only populated when creating a new key and `pgp_key` is supplied
- `private_key_fingerprint` - The MD5 public key fingerprint for the encrypted private key. This is only populated when creating a new key and `pgp_key` is supplied
- `valid_after` - The key can be used after this timestamp. A timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds. Example: "2014-10-02T15:01:23.045123456Z".
- `valid_before` - The key can be used before this timestamp. A timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds. Example: "2014-10-02T15:01:23.045123456Z".

# » google_compute_address

Creates a static IP address resource for Google Compute Engine. For more information see the official documentation for external and internal static IP reservations, as well as the API.

## » Example Usage

```
resource "google_compute_address" "default" {
  name = "test-address"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) The Region in which the created address should reside. If it is not provided, the provider region is used.
- `address_type` - (Optional) The Address Type that should be configured. Specify INTERNAL to reserve an internal static IP address EXTERNAL to specify an external static IP address. Defaults to EXTERNAL if omitted.
- `subnetwork` - (Optional) The self link URI of the subnetwork in which to create the address. A subnetwork may only be specified for INTERNAL address types.
- `address` - (Optional) The IP address to reserve. An address may only be specified for INTERNAL address types. The IP address must be inside the specified subnetwork, if any.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.
- `address` - The IP of the created resource.

## » Import

Addresses can be imported using the `project`, `region` and `name`, e.g.

```
$ terraform import google_compute_address.default gcp-project/us-central1/test-address
```

If `project` is omitted, the default project set for the provider is used:

```
$ terraform import google_compute_address.default us-central1/test-address
```

If `project` and `region` are omitted, the default project and region set for the provider are used.

```
$ terraform import google_compute_address.default test-address
```

Alternatively, addresses can be imported using a full or partial `self_link`.

```
$ terraform import google_compute_address.default https://www.googleapis.com/compute/v1/proj

$ terraform import google_compute_address.default projects/gcp-project/regions/us-central1/a
```

## » google_compute_autoscaler

A Compute Engine Autoscaler automatically adds or removes virtual machines from a managed instance group based on increases or decreases in load. This allows your applications to gracefully handle increases in traffic and reduces cost when the need for resources is lower. You just define the autoscaling policy and the autoscaler performs automatic scaling based on the measured load. For more information, see the official documentation and API

### » Example Usage

```
resource "google_compute_instance_template" "foobar" {
  name           = "foobar"
  machine_type   = "n1-standard-1"
  can_ip_forward = false

  tags = ["foo", "bar"]

  disk {
    source_image = "debian-cloud/debian-8"
  }

  network_interface {
    network = "default"
  }

  metadata {
    foo = "bar"
  }

  service_account {
    scopes = ["userinfo-email", "compute-ro", "storage-ro"]
  }
}

resource "google_compute_target_pool" "foobar" {
  name = "foobar"
}

resource "google_compute_instance_group_manager" "foobar" {
  name = "foobar"
  zone = "us-central1-f"

  instance_template  = "${google_compute_instance_template.foobar.self_link}"
  target_pools       = ["${google_compute_target_pool.foobar.self_link}"]
  base_instance_name = "foobar"
}

resource "google_compute_autoscaler" "foobar" {
  name   = "scaler"
  zone   = "us-central1-f"
  target = "${google_compute_instance_group_manager.foobar.self_link}"

  autoscaling_policy = {
    max_replicas    = 5
    min_replicas    = 1
    cooldown_period = 60

    cpu_utilization {
      target = 0.5
    }
  }
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the autoscaler.
- `target` - (Required) The full URL to the instance group manager whose size we control.
- `zone` - (Required) The zone of the target.
- `autoscaling_policy` - (Required) The parameters of the autoscaling algorithm. Structure is documented below.

- `description` - (Optional) An optional textual description of the instance group manager.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `autoscaling_policy` block contains:

- `max_replicas` - (Required) The group will never be larger than this.
- `min_replicas` - (Required) The group will never be smaller than this.
- `cooldown_period` - (Optional) Period to wait between changes. This should be at least double the time your instances take to start up.
- `cpu_utilization` - (Optional) A policy that scales when the cluster's average CPU is above or below a given threshold. Structure is documented below.
- `metric` - (Optional) A policy that scales according to Google Cloud Monitoring metrics Structure is documented below.
- `load_balancing_utilization` - (Optional) A policy that scales when the load reaches a proportion of a limit defined in the HTTP load balancer. Structure is documented below.

The `cpu_utilization` block contains:

- `target` - The floating point threshold where CPU utilization should be. E.g. for 50% one would specify 0.5.

The `metric` block contains (more documentation here):

- `name` - The name of the Google Cloud Monitoring metric to follow, e.g. `compute.googleapis.com/instance/network/received_bytes_count`
- `type` - Either "cumulative", "delta", or "gauge".
- `target` - The desired metric value per instance. Must be a positive value.

The `load_balancing_utilization` block contains:

- `target` - The floating point threshold where load balancing utilization should be. E.g. if the load balancer's `maxRatePerInstance` is 10 requests per second (RPS) then setting this to 0.5 would cause the group to be scaled such that each instance receives 5 RPS.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URL of the created resource.

### » Import

Autoscalers can be imported using the `name`, e.g.

```
$ terraform import google_compute_autoscaler.foobar scaler
```

## » google_compute_backend_bucket

A Backend Bucket defines a Google Cloud Storage bucket that will serve traffic through Google Cloud Load Balancer. For more information see the official documentation and API.

### » Example Usage

```
resource "google_compute_backend_bucket" "image_backend" {
  name        = "image-backend-bucket"
  description = "Contains beautiful images"
  bucket_name = "${google_storage_bucket.image_bucket.name}"
  enable_cdn  = true
}

resource "google_storage_bucket" "image_bucket" {
  name     = "image-store-bucket"
  location = "EU"
}
```

### » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the backend bucket.
- `bucket_name` - (Required) The name of the Google Cloud Storage bucket to be used as a backend bucket.

---

- `description` - (Optional) The textual description for the backend bucket.
- `enable_cdn` - (Optional) Whether or not to enable the Cloud CDN on the backend bucket.

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `self_link` - The URI of the created resource.

## » Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 4 minutes.
- `update` - Default is 4 minutes.
- `delete` - Default is 4 minutes.

## » Import

Backend buckets can be imported using the `name`, e.g.

```
$ terraform import google_compute_backend_bucket.image_backend image-backend-bucket
```

# » google_compute_backend_service

A Backend Service defines a group of virtual machines that will serve traffic for load balancing. For more information see the official documentation and the API.

For internal load balancing, use a google_compute_region_backend_service.

## » Example Usage

```
resource "google_compute_backend_service" "website" {
  name        = "my-backend"
  description = "Our company website"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10
  enable_cdn  = false
  backend {
    group = "${google_compute_instance_group_manager.webservers.instance_group}"
  }

  health_checks = ["${google_compute_http_health_check.default.self_link}"]
}

resource "google_compute_instance_group_manager" "webservers" {
  name               = "my-webservers"
  instance_template  = "${google_compute_instance_template.webserver.self_link}"
  base_instance_name = "webserver"
  zone               = "us-central1-f"
  target_size        = 1
}

resource "google_compute_instance_template" "webserver" {
  name         = "standard-webserver"
  machine_type = "n1-standard-1"

  network_interface {
    network = "default"
  }

  disk {
    source_image = "debian-cloud/debian-8"
    auto_delete  = true
    boot         = true
  }
}

resource "google_compute_http_health_check" "default" {
  name               = "test"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the backend service.
- `health_checks` - (Required) Specifies a list of HTTP/HTTPS health checks for checking the health of the backend service. Currently at most

one health check can be specified, and a health check is required.

---

- backend - (Optional) The list of backends that serve this BackendService. Structure is documented below.
- iap - (Optional) Specification for the Identity-Aware proxy. Disabled if not specified. Structure is documented below.
- cdn_policy - (Optional) Cloud CDN configuration for this BackendService. Structure is documented below.
- description - (Optional) The textual description for the backend service.
- enable_cdn - (Optional) Whether or not to enable the Cloud CDN on the backend service.
- port_name - (Optional) The name of a service that has been added to an instance group in this backend. See related docs for details. Defaults to http.
- project - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- protocol - (Optional) The protocol for incoming requests. Defaults to HTTP.
- session_affinity - (Optional) How to distribute load. Options are NONE (no affinity), CLIENT_IP (hash of the source/dest addresses / ports), and GENERATED_COOKIE (distribute load using a generated session cookie).
- timeout_sec - (Optional) The number of secs to wait for a backend to respond to a request before considering the request failed. Defaults to 30.
- connection_draining_timeout_sec - (Optional) Time for which instance will be drained (not accept new connections, but still work to finish started ones). Defaults to 300.

The backend block supports:

- group - (Required) The name or URI of a Compute Engine instance group (google_compute_instance_group_manager.xyz.instance_group) that can receive traffic.
- balancing_mode - (Optional) Defines the strategy for balancing load. Defaults to UTILIZATION
- capacity_scaler - (Optional) A float in the range [0, 1.0] that scales the maximum parameters for the group (e.g., max rate). A value of 0.0 will cause no requests to be sent to the group (i.e., it adds the group in a drained state). The default is 1.0.
- description - (Optional) Textual description for the backend.

- `max_rate` - (Optional) Maximum requests per second (RPS) that the group can handle.
- `max_rate_per_instance` - (Optional) The maximum per-instance requests per second (RPS).
- `max_utilization` - (Optional) The target CPU utilization for the group as a float in the range [0.0, 1.0]. This flag can only be provided when the balancing mode is `UTILIZATION`. Defaults to `0.8`.

The `cdn_policy` block supports:

- `cache_key_policy` - (Optional) The CacheKeyPolicy for this CdnPolicy. Structure is documented below.

The `cache_key_policy` block supports:

- `include_host` - (Optional) If true, requests to different hosts will be cached separately.
- `include_protocol` - (Optional) If true, http and https requests will be cached separately.
- `include_query_string` - (Optional) If true, include query string parameters in the cache key according to `query_string_whitelist` and `query_string_blacklist`. If neither is set, the entire query string will be included. If false, the query string will be excluded from the cache key entirely.
- `query_string_blacklist` - (Optional) Names of query string parameters to exclude in cache keys. All other parameters will be included. Either specify `query_string_whitelist` or `query_string_blacklist`, not both. '&' and '=' will be percent encoded and not treated as delimiters.
- `query_string_whitelist` - (Optional) Names of query string parameters to include in cache keys. All other parameters will be excluded. Either specify `query_string_whitelist` or `query_string_blacklist`, not both. '&' and '=' will be percent encoded and not treated as delimiters.

The `iap` block supports:

- `oauth2_client_id` - (Required) The client ID for use with OAuth 2.0.
- `oauth2_client_secret` - (Required) The client secret for use with OAuth 2.0.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `fingerprint` - The fingerprint of the backend service.

- `self_link` - The URI of the created resource.

## » Import

Backend services can be imported using the `name`, e.g.

```
$ terraform import google_compute_backend_service.website my-backend
```

# » google_compute_disk

Creates a new persistent disk within GCE, based on another disk. For more information see the official documentation and API.

**Note:** All arguments including the disk encryption key will be stored in the raw state as plain-text. Read more about sensitive data in state.

## » Example Usage

```
resource "google_compute_disk" "default" {
  name  = "test-disk"
  type  = "pd-ssd"
  zone  = "us-central1-a"
  image = "debian-8-jessie-v20170523"
  labels {
    environment = "dev"
  }
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `zone` - (Required) The zone where this disk will be available.

---

- `disk_encryption_key_raw` - (Optional) A 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to encrypt this disk.
- `image` - (Optional) The image from which to initialize this disk. This can be one of: the image's `self_link`, `projects/{project}/global/images/{image}`, `projects/{project}/global/images/family/{family}`, `global/images/{image}`,

global/images/family/{family}, family/{family}, {project}/{family}, {project}/{image}, {family}, or {image}. If referred by family, the images names must include the family name. If they don't, use the google_compute_image data source. For instance, the image centos-6-v20180104 includes its family name centos-6. These images can be referred by family name here.

- project - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- size - (Optional) The size of the image in gigabytes. If not specified, it will inherit the size of its base image.
- snapshot - (Optional) Name of snapshot from which to initialize this disk.
- type - (Optional) The GCE disk type.
- labels - (Optional) A set of key/value label pairs to assign to the image.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- disk_encryption_key_sha256 - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.
- self_link - The URI of the created resource.
- users - The Users of the created resource.
- label_fingerprint - The fingerprint of the assigned labels.

## » Timeouts

google_compute_disk provides the following Timeouts configuration options:

- create - (Default 5 minutes) Used for creating disks.
- update - (Default 5 minutes) Used for resizing a disk and setting labels on disks.
- delete - (Default 5 minutes) Used for destroying disks (not including time to detach the disk from instances).

## » Import

Disks can be imported using the name, e.g.

```
$ terraform import google_compute_disk.default test-disk
```

# » google_compute_firewall

Manages a firewall resource within GCE. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_firewall" "default" {
  name    = "test-firewall"
  network = "${google_compute_network.other.name}"

  allow {
    protocol = "icmp"
  }

  allow {
    protocol = "tcp"
    ports    = ["80", "8080", "1000-2000"]
  }

  source_tags = ["web"]
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `network` - (Required) The name or self_link of the network to attach this firewall to.

---

- `allow` - (Required) Can be specified multiple times for each allow rule. Each allow block supports fields documented below.
- `deny` - (Optional) Can be specified multiple times for each deny rule. Each deny block supports fields documented below. Can be specified instead of allow.
- `description` - (Optional) Textual description field.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

- `priority` - (Optional) The priority for this firewall. Ranges from 0-65535, inclusive. Defaults to 1000. Firewall resources with lower priority values have higher precedence (e.g. a firewall resource with a priority value of 0 takes effect over all other firewall rules with a non-zero priority).
- `source_ranges` - (Optional) A list of source CIDR ranges that this firewall applies to. Can't be used for `EGRESS`.
- `source_tags` - (Optional) A list of source tags for this firewall. Can't be used for `EGRESS`.
- `target_tags` - (Optional) A list of target tags for this firewall.
- `direction` - (Optional) Direction of traffic to which this firewall applies; One of `INGRESS` or `EGRESS`. Defaults to `INGRESS`.
- `destination_ranges` - (Optional) A list of destination CIDR ranges that this firewall applies to. Can't be used for `INGRESS`.
- `source_service_accounts` - (Optional) A list of service accounts such that the firewall will apply only to traffic originating from an instance with a service account in this list. Source service accounts cannot be used to control traffic to an instance's external IP address because service accounts are associated with an instance, not an IP address. `source_ranges` can be set at the same time as `source_service_accounts`. If both are set, the firewall will apply to traffic that has source IP address within `source_ranges` OR the source IP belongs to an instance with service account listed in `source_service_accounts`. The connection does not need to match both properties for the firewall to apply. `source_service_accounts` cannot be used at the same time as `source_tags` or `target_tags`.
- `target_service_accounts` - (Optional) A list of service accounts indicating sets of instances located in the network that may make network connections as specified in `allow`. `target_service_accounts` cannot be used at the same time as `source_tags` or `target_tags`. If neither `target_service_accounts` nor `target_tags` are specified, the firewall rule applies to all instances on the specified network.

The `allow` block supports:

- `protocol` - (Required) The name of the protocol to allow. This value can either be one of the following well known protocol strings (tcp, udp, icmp, esp, ah, sctp), or the IP protocol number, or `all`.
- `ports` - (Optional) List of ports and/or port ranges to allow. This can only be specified if the protocol is TCP or UDP.

The `deny` block supports:

- `protocol` - (Required) The name of the protocol to allow. This value can either be one of the following well known protocol strings (tcp, udp, icmp, esp, ah, sctp), or the IP protocol number, or `all`.

- `ports` - (Optional) List of ports and/or port ranges to allow. This can only be specified if the protocol is TCP or UDP.

### » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.

### » Import

Firewalls can be imported using the `name`, e.g.

```
$ terraform import google_compute_firewall.default test-firewall
```

## » google_compute_forwarding_rule

Manages a Forwarding Rule within GCE. This binds an ip and port range to a target pool. For more information see the official documentation and API.

### » Example Usage

```
resource "google_compute_forwarding_rule" "default" {
  name       = "website-forwarding-rule"
  target     = "${google_compute_target_pool.default.self_link}"
  port_range = "80"
}
```

### » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `backend_service` - (Optional) BackendService resource to receive the matched traffic. Only used for internal load balancing.
- `description` - (Optional) Textual description field.

- `ip_address` - (Optional) The static IP. (if not set, an ephemeral IP is used).
- `ip_protocol` - (Optional) The IP protocol to route, one of ”TCP” ”UDP” ”AH” ”ESP” or ”SCTP” for external load balancing, ”TCP” or ”UDP” for internal (default ”TCP”).
- `load_balancing_scheme` - (Optional) Type of load balancing to use. Can be set to ”INTERNAL” or ”EXTERNAL” (default ”EXTERNAL”).
- `network` - (Optional) Network name or self_link that the load balanced IP should belong to. Only used for internal load balancing. If it is not provided, the default network is used.
- `port_range` - (Optional) A range e.g. ”1024-2048” or a single port ”1024” (defaults to all ports!). Only used for external load balancing. Some types of forwarding targets have constraints on the acceptable ports:
  - Target HTTP proxy: 80, 8080
  - Target HTTPS proxy: 443
  - Target TCP proxy: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
  - Target SSL proxy: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
  - Target VPN gateway: 500, 4500
- `ports` - (Optional) A list of ports (maximum of 5) to use for internal load balancing. Packets addressed to these ports will be forwarded to the backends configured with this forwarding rule. Required for internal load balancing.
- `project` - (Optional) The ID of project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) The Region in which the created address should reside. If it is not provided, the provider region is used.
- `subnetwork` - (Optional) Subnetwork that the load balanced IP should belong to. Only used for internal load balancing. Must be specified if the network is in custom subnet mode.
- `target` - (Optional) URL of target pool. Required for external load balancing.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.

## » Import

Forwarding rules can be imported using the `name`, e.g.

```
$ terraform import google_compute_forwarding_rule.default website-forwarding-rule
```

# » google_compute_global_address

Creates a static IP address resource global to a Google Compute Engine project. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_global_address" "default" {
  name = "global-appserver-ip"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `ip_version` - (Optional) The IP Version that will be used by this address. One of `"IPV4"` or `"IPV6"`.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `address` - The assigned address.
- `self_link` - The URI of the created resource.

## » Import

Global addresses can be imported using the `name`, e.g.

```
$ terraform import google_compute_global_address.default global-appserver-ip
```

# » google_compute_global_forwarding_rule

Manages a Global Forwarding Rule within GCE. This binds an ip and port to a target HTTP(s) proxy. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_global_forwarding_rule" "default" {
  name       = "default-rule"
  target     = "${google_compute_target_http_proxy.default.self_link}"
  port_range = "80"
}

resource "google_compute_target_http_proxy" "default" {
  name        = "test-proxy"
  description = "a description"
  url_map     = "${google_compute_url_map.default.self_link}"
}

resource "google_compute_url_map" "default" {
  name            = "url-map"
  description     = "a description"
  default_service = "${google_compute_backend_service.default.self_link}"

  host_rule {
    hosts        = ["mysite.com"]
    path_matcher = "allpaths"
  }

  path_matcher {
    name            = "allpaths"
    default_service = "${google_compute_backend_service.default.self_link}"

    path_rule {
      paths   = ["/*"]
      service = "${google_compute_backend_service.default.self_link}"
    }
  }
}

resource "google_compute_backend_service" "default" {
  name        = "default-backend"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = ["${google_compute_http_health_check.default.self_link}"]
}

resource "google_compute_http_health_check" "default" {
  name               = "test"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `target` - (Required) URL of target HTTP or HTTPS proxy.

---

- `description` - (Optional) Textual description field.
- `ip_address` - (Optional) The static IP. (if not set, an ephemeral IP is used). This should be the literal IP address to be used, not the `self_link` to a `google_compute_global_address` resource. (If using a `google_compute_global_address` resource, use the `address` property instead of the `self_link` property.)
- `ip_protocol` - (Optional) The IP protocol to route, one of ”TCP” ”UDP” ”AH” ”ESP” or ”SCTP”. (default ”TCP”).
- `port_range` - (Optional) A range e.g. ”1024-2048” or a single port ”1024” (defaults to all ports!). Some types of forwarding targets have constraints on the acceptable ports:
  - Target HTTP proxy: 80, 8080
  - Target HTTPS proxy: 443
  - Target TCP proxy: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222

    - Target SSL proxy: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
    - Target VPN gateway: 500, 4500

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `ip_version` - (Optional) The IP Version that will be used by this resource's address. One of `"IPV4"` or `"IPV6"`.

---

- `labels` - (Optional, Beta) A set of key/value label pairs to assign to the resource.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.
- `label_fingerprint` - (Beta) The current label fingerprint.

## » Import

Global forwarding rules can be imported using the `name`, e.g.

```
$ terraform import google_compute_global_forwarding_rule.default default-rule
```

# » google_compute_health_check

Manages a health check within GCE. This is used to monitor instances behind load balancers. Timeouts or HTTP errors cause the instance to be removed from the pool. For more information, see the official documentation and API.

## » Example Usage

```
resource "google_compute_health_check" "default" {
  name = "internal-service-health-check"

  timeout_sec        = 1
  check_interval_sec = 1

  tcp_health_check {
    port = "80"
  }
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `check_interval_sec` - (Optional) The number of seconds between each poll of the instance instance (default 5).
- `description` - (Optional) Textual description field.
- `healthy_threshold` - (Optional) Consecutive successes required (default 2).
- `http_health_check` - (Optional) An HTTP Health Check. Only one kind of Health Check can be added. Structure is documented below.
- `https_health_check` - (Optional) An HTTPS Health Check. Only one kind of Health Check can be added. Structure is documented below.
- `ssl_health_check` - (Optional) An SSL Health Check. Only one kind of Health Check can be added. Structure is documented below.
- `tcp_health_check` - (Optional) A TCP Health Check. Only one kind of Health Check can be added. Structure is documented below.
- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `timeout_sec` - (Optional) The number of seconds to wait before declaring failure (default 5).
- `unhealthy_threshold` - (Optional) Consecutive failures required (default 2).

The `http_health_check` block supports:

- `host` - (Optional) HTTP host header field (default instance's public ip).
- `port` - (Optional) TCP port to connect to (default 80).
- `proxy_header` - (Optional) Type of proxy header to append before sending data to the backend, either NONE or PROXY_V1 (default NONE).
- `request_path` - (Optional) URL path to query (default /).

The `https_health_check` block supports:

- host - (Optional) HTTPS host header field (default instance's public ip).
- port - (Optional) TCP port to connect to (default 443).
- proxy_header - (Optional) Type of proxy header to append before sending data to the backend, either NONE or PROXY_V1 (default NONE).
- request_path - (Optional) URL path to query (default /).

The ssl_health_check block supports:

- port - (Optional) TCP port to connect to (default 443).
- proxy_header - (Optional) Type of proxy header to append before sending data to the backend, either NONE or PROXY_V1 (default NONE).
- request - (Optional) Application data to send once the SSL connection has been established (default "").
- response - (Optional) The response that indicates health (default "")

The tcp_health_check block supports:

- port - (Optional) TCP port to connect to (default 80).
- proxy_header - (Optional) Type of proxy header to append before sending data to the backend, either NONE or PROXY_V1 (default NONE).
- request - (Optional) Application data to send once the TCP connection has been established (default "").
- response - (Optional) The response that indicates health (default "")

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- self_link - The URI of the created resource.

## » Import

Health checks can be imported using the name, e.g.

```
$ terraform import google_compute_health_check.default internal-service-health-check
```

# » google_compute_http_health_check

Manages an HTTP health check within GCE. This is used to monitor instances behind load balancers. Timeouts or HTTP errors cause the instance to be removed from the pool. For more information, see the official documentation and API.

**Note:** HttpHealthChecks/HTTPSHealthChecks are legacy health checks. The newer google_compute_health_check should be preferred for all uses except except Network Load Balancers which still require the legacy HttpHealthChecks.

## » Example Usage

```
resource "google_compute_http_health_check" "default" {
  name         = "authentication-health-check"
  request_path = "/health_check"

  timeout_sec        = 1
  check_interval_sec = 1
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `check_interval_sec` - (Optional) The number of seconds between each poll of the instance instance (default 5).
- `description` - (Optional) Textual description field.
- `healthy_threshold` - (Optional) Consecutive successes required (default 2).
- `host` - (Optional) HTTP host header field (default instance's public ip).
- `port` - (Optional) TCP port to connect to (default 80).
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `request_path` - (Optional) URL path to query (default /).
- `timeout_sec` - (Optional) The number of seconds to wait before declaring failure (default 5).
- `unhealthy_threshold` - (Optional) Consecutive failures required (default 2).

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `self_link` - The URI of the created resource.

## » Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 4 minutes.
- `update` - Default is 4 minutes.
- `delete` - Default is 4 minutes.

## » Import

HTTP health checks can be imported using the `name`, e.g.

```
$ terraform import google_compute_http_health_check.default authentication-health-check
```

# » google_compute_https_health_check

Manages an HTTPS health check within GCE. This is used to monitor instances behind load balancers. Timeouts or HTTPS errors cause the instance to be removed from the pool. For more information, see the official documentation and API.

**Note:** HTTPSHealthChecks/HttpHealthChecks are legacy health checks. The newer google_compute_health_check should be preferred for all uses except except Network Load Balancers which still require the legacy HttpHealthChecks.

## » Example Usage

```
resource "google_compute_https_health_check" "default" {
  name         = "test"
  request_path = "/health_check"

  timeout_sec        = 1
  check_interval_sec = 1
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `check_interval_sec` - (Optional) How often to poll each instance (default 5).
- `description` - (Optional) Textual description field.
- `healthy_threshold` - (Optional) Consecutive successes required (default 2).
- `host` - (Optional) HTTPS host header field (default instance's public ip).
- `port` - (Optional) TCP port to connect to (default 443).
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `request_path` - (Optional) URL path to query (default /).
- `timeout_sec` - (Optional) How long before declaring failure (default 5).
- `unhealthy_threshold` - (Optional) Consecutive failures required (default 2).

## » Attributes Reference

The following attributes are exported:

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `self_link` - The URL of the created resource.

## » Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 4 minutes.
- `update` - Default is 4 minutes.
- `delete` - Default is 4 minutes.

## » Import

HTTPS health checks can be imported using the `name`, e.g.

```
$ terraform import google_compute_https_health_check.default test
```

# » google_compute_image

Creates a bootable VM image resource for Google Compute Engine from an existing tarball. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_image" "bootable-image" {
  name = "my-custom-image"

  raw_disk {
    source = "https://storage.googleapis.com/my-bucket/my-disk-image-tarball.tar.gz"
  }
}

resource "google_compute_instance" "vm" {
  name         = "vm-from-custom-image"
  machine_type = "n1-standard-1"
  zone         = "us-east1-c"

  boot_disk {
    initialize_params {
      image = "${google_compute_image.bootable-image.self_link}"
    }
  }

  network_interface {
    network = "default"
  }
}
```

## » Argument Reference

The following arguments are supported: (Note that one of either source_disk or raw_disk is required)

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `description` - (Optional) The description of the image to be created
- `family` - (Optional) The name of the image family to which this image belongs.
- `labels` - (Optional) A set of key/value label pairs to assign to the image.
- `source_disk` - (Optional) The URL of a disk that will be used as the source of the image. Changing this forces a new resource to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `raw_disk` - (Optional) The raw disk that will be used as the source of the image. Changing this forces a new resource to be created. Structure is documented below.
- `create_timeout` - (Deprecated) Configurable timeout in minutes for creating images. Default is 4 minutes.

The `raw_disk` block supports:

- `source` - (Required) The full Google Cloud Storage URL where the disk image is stored.
- `sha1` - (Optional) SHA1 checksum of the source tarball that will be used to verify the source before creating the image.
- `container_type` - (Optional) The format used to encode and transmit the block device. TAR is the only supported type and is the default.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.
- `label_fingerprint` - The fingerprint of the assigned labels.

## » Timeouts

`google_compute_image` provides the following Timeouts configuration options:

- `create` - Default `4 minutes`
- `update` - Default `4 minutes`

- `delete` - Default `4 minutes`

### » Import

VM image can be imported using the `name`, e.g.

```
$ terraform import google_compute_image.web-image my-custom-image
```

## » google_compute_instance

Manages a VM instance resource within GCE. For more information see the official documentation and API.

### » Example Usage

```
resource "google_compute_instance" "default" {
  name         = "test"
  machine_type = "n1-standard-1"
  zone         = "us-central1-a"

  tags = ["foo", "bar"]

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-8"
    }
  }

  // Local SSD disk
  scratch_disk {
  }

  network_interface {
    network = "default"

    access_config {
      // Ephemeral IP
    }
  }

  metadata {
    foo = "bar"
  }
  metadata_startup_script = "echo hi > /test.txt"

  service_account {
    scopes = ["userinfo-email", "compute-ro", "storage-ro"]
  }
}
```

## » Argument Reference

The following arguments are supported:

- `boot_disk` - (Required) The boot disk for the instance. Structure is documented below.
- `machine_type` - (Required) The machine type to create. To create a custom machine type, value should be set as specified here. **Note**: `allow_stopping_for_update` must be set to true in order to update this field.
- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `zone` - (Required) The zone that the machine should be created in.
- `network_interface` - (Required) Networks to attach to the instance. This can be specified multiple times. Structure is documented below.

---

- `allow_stopping_for_update` - (Optional) If true, allows Terraform to stop the instance to update its properties. If you try to update a property that requires stopping the instance without setting this field, the update will fail.
- `attached_disk` - (Optional) List of disks to attach to the instance. Structure is documented below.
- `can_ip_forward` - (Optional) Whether to allow sending and receiving of packets with non-matching source or destination IPs. This defaults to false.
- `create_timeout` - (Optional) Configurable timeout in minutes for creating instances. Default is 4 minutes. Changing this forces a new resource to be created.
- `description` - (Optional) A brief description of this resource.
- `deletion_protection` - (Optional) Enable deletion protection on this instance. Defaults to false. **Note:** you must disable deletion protection be-

fore removing the resource (e.g., via `terraform destroy`), or the instance cannot be deleted and the Terraform run will not complete successfully.

- `guest_accelerator` - (Optional) List of the type and count of accelerator cards attached to the instance. Structure documented below.
- `labels` - (Optional) A set of key/value label pairs to assign to the instance.
- `metadata` - (Optional) Metadata key/value pairs to make available from within the instance.
- `metadata_startup_script` - (Optional) An alternative to using the startup-script metadata key, except this one forces the instance to be recreated (thus re-running the script) if it is changed. This replaces the startup-script metadata key on the created instance and thus the two mechanisms are not allowed to be used simultaneously.
- `min_cpu_platform` - (Optional) Specifies a minimum CPU platform for the VM instance. Applicable values are the friendly names of CPU platforms, such as `Intel Haswell` or `Intel Skylake`. See the complete list here. **Note**: `allow_stopping_for_update` must be set to true in order to update this field.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `scheduling` - (Optional) The scheduling strategy to use. More details about this configuration option are detailed below.
- `scratch_disk` - (Optional) Scratch disks to attach to the instance. This can be specified multiple times for multiple scratch disks. Structure is documented below.
- `service_account` - (Optional) Service account to attach to the instance. Structure is documented below. **Note**: `allow_stopping_for_update` must be set to true in order to update this field.
- `tags` - (Optional) A list of tags to attach to the instance.

---

The `boot_disk` block supports:

- `auto_delete` - (Optional) Whether the disk will be auto-deleted when the instance is deleted. Defaults to true.
- `device_name` - (Optional) Name with which attached disk will be accessible under `/dev/disk/by-id/`
- `disk_encryption_key_raw` - (Optional) A 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to encrypt this disk.

- `initialize_params` - (Optional) Parameters for a new disk that will be created alongside the new instance. Either `initialize_params` or `source` must be set. Structure is documented below.

- `source` - (Optional) The name or self_link of the existing disk (such as those managed by `google_compute_disk`) to attach.

The `initialize_params` block supports:

- `size` - (Optional) The size of the image in gigabytes. If not specified, it will inherit the size of its base image.

- `type` - (Optional) The GCE disk type. May be set to pd-standard or pd-ssd.

- `image` - (Optional) The image from which to initialize this disk. This can be one of: the image's `self_link`, `projects/{project}/global/images/{image}`, `projects/{project}/global/images/family/{family}`, `global/images/{image}`, `global/images/family/{family}`, `family/{family}`, `{project}/{family}`, `{project}/{image}`, `{family}`, or `{image}`. If referred by family, the images names must include the family name. If they don't, use the google_compute_image data source. For instance, the image `centos-6-v20180104` includes its family name `centos-6`. These images can be referred by family name here.

The `scratch_disk` block supports:

- `interface` - (Optional) The disk interface to use for attaching this disk; either SCSI or NVME. Defaults to SCSI.

The `attached_disk` block supports:

- `source` - (Required) The name or self_link of the disk to attach to this instance.

- `device_name` - (Optional) Name with which the attached disk will be accessible under `/dev/disk/by-id/`

- `mode` - (Optional) Either "READ_ONLY" or "READ_WRITE", defaults to "READ_WRITE" If you have a persistent disk with data that you want to share between multiple instances, detach it from any read-write instances and attach it to one or more instances in read-only mode.

- `disk_encryption_key_raw` - (Optional) A 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to encrypt this disk.

The `network_interface` block supports:

- `network` - (Optional) The name or self_link of the network to attach this interface to. Either `network` or `subnetwork` must be provided.

- `subnetwork` - (Optional) The name or self_link of the subnetwork to attach this interface to. The subnetwork must exist in the same region this instance will be created in. Either network or subnetwork must be provided.

- subnetwork_project - (Optional) The project in which the subnetwork belongs. If the subnetwork is a self_link, this field is ignored in favor of the project defined in the subnetwork self_link. If the subnetwork is a name and this field is not provided, the provider project is used.
- address - (Optional) The private IP address to assign to the instance. If empty, the address will be automatically assigned.
- access_config - (Optional) Access configurations, i.e. IPs via which this instance can be accessed via the Internet. Omit to ensure that the instance is not accessible from the Internet (this means that ssh provisioners will not work unless you are running Terraform can send traffic to the instance's network (e.g. via tunnel or because it is running on another cloud instance on that network). This block can be repeated multiple times. Structure documented below.
- alias_ip_range - (Optional) An array of alias IP ranges for this network interface. Can only be specified for network interfaces on subnet-mode networks. Structure documented below.

The access_config block supports:

- nat_ip - (Optional) The IP address that will be 1:1 mapped to the instance's network ip. If not given, one will be generated.

The alias_ip_range block supports:

- ip_cidr_range - The IP CIDR range represented by this alias IP range. This IP CIDR range must belong to the specified subnetwork and cannot contain IP addresses reserved by system or used by other network interfaces. This range may be a single IP address (e.g. 10.2.3.4), a netmask (e.g. /24) or a CIDR format string (e.g. 10.1.2.0/24).
- subnetwork_range_name - (Optional) The subnetwork secondary range name specifying the secondary range from which to allocate the IP CIDR range for this alias IP range. If left unspecified, the primary range of the subnetwork will be used.

The service_account block supports:

- email - (Optional) The service account e-mail address. If not given, the default Google Compute Engine service account is used. **Note**: allow_stopping_for_update must be set to true in order to update this field.
- scopes - (Required) A list of service scopes. Both OAuth2 URLs and gcloud short names are supported. To allow full access to all Cloud APIs, use the cloud-platform scope. See a complete list of scopes here. **Note**:

allow_stopping_for_update must be set to true in order to update this field.

The scheduling block supports:

- preemptible - (Optional) Is the instance preemptible.
- on_host_maintenance - (Optional) Describes maintenance behavior for the instance. Can be MIGRATE or TERMINATE, for more info, read here
- automatic_restart - (Optional) Specifies if the instance should be restarted if it was terminated by Compute Engine (not a user).

The guest_accelerator block supports:

- type (Required) - The accelerator type resource to expose to this instance. E.g. nvidia-tesla-k80.
- count (Required) - The number of the guest accelerator cards exposed to this instance.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- instance_id - The server-assigned unique identifier of this instance.
- metadata_fingerprint - The unique fingerprint of the metadata.
- self_link - The URI of the created resource.
- tags_fingerprint - The unique fingerprint of the tags.
- label_fingerprint - The unique fingerprint of the labels.
- cpu_platform - The CPU platform used by this instance.
- network_interface.0.address - The internal ip address of the instance, either manually or dynamically assigned.
- network_interface.0.access_config.0.assigned_nat_ip - If the instance has an access config, either the given external ip (in the nat_ip field) or the ephemeral (generated) ip (if you didn't provide one).
- attached_disk.0.disk_encryption_key_sha256 - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.
- boot_disk.disk_encryption_key_sha256 - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.

- `disk.0.disk_encryption_key_sha256` - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.

## » Import

**Note:** The fields `boot_disk.0.disk_entryption_raw` and `attached_disk.*.disk_encryption_key_raw` cannot be imported automatically. The API doesn't return this information. If you are setting one of these fields in your config, you will need to update your state manually after importing the resource.

Instances can be imported using the `project`, `zone` and `name`, e.g.

```
$ terraform import google_compute_instance.default gcp-project/us-central1-a/test
```

# » google_compute_instance_group

Creates a group of dissimilar Compute Engine virtual machine instances. For more information, see the official documentation and API

## » Example Usage

### » Empty instance group

```
resource "google_compute_instance_group" "test" {
  name        = "terraform-test"
  description = "Terraform test instance group"
  zone        = "us-central1-a"
  network     = "${google_compute_network.default.self_link}"
}
```

### » With instances and named ports

```
resource "google_compute_instance_group" "webservers" {
  name        = "terraform-webservers"
  description = "Terraform test instance group"

  instances = [
    "${google_compute_instance.test.self_link}",
    "${google_compute_instance.test2.self_link}",
  ]

  named_port {
    name = "http"
    port = "8080"
  }

  named_port {
    name = "https"
    port = "8443"
  }

  zone = "us-central1-a"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the instance group. Must be 1-63 characters long and comply with RFC1035. Supported characters include lowercase letters, numbers, and hyphens.
- `zone` - (Required) The zone that this instance group should be created in.

---

- `description` - (Optional) An optional textual description of the instance group.
- `instances` - (Optional) List of instances in the group. They should be given as self_link URLs. When adding instances they must all be in the same network and zone as the instance group.
- `named_port` - (Optional) The named port configuration. See the section below for details on configuration.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `network` - (Optional) The URL of the network the instance group is in. If this is different from the network where the instances are in, the creation fails. Defaults to the network where the instances are in (if neither `network` nor `instances` is specified, this field will be blank).

The `named_port` block supports:

- `name` - (Required) The name which the port will be mapped to.
- `port` - (Required) The port number to map the name to.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.
- `size` - The number of instances in the group.

## » Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is `6 minutes`
- `update` - Default is `6 minutes`
- `delete` - Default is `6 minutes`

## » Import

Instance group can be imported using the `zone` and `name`, e.g.

```
$ terraform import google_compute_instance_group.webservers us-central1-a/terraform-webserve
```

# » google_compute_instance_group_manager

The Google Compute Engine Instance Group Manager API creates and manages pools of homogeneous Compute Engine virtual machine instances from a common instance template. For more information, see the official documentation and API

**Note:** Use google_compute_region_instance_group_manager to create a regional (multi-zone) instance group manager.

## » Example Usage

```
resource "google_compute_health_check" "autohealing" {
  name                = "autohealing-health-check"
  check_interval_sec  = 5
  timeout_sec         = 5
  healthy_threshold   = 2
  unhealthy_threshold = 10                         # 50 seconds

  http_health_check {
    request_path = "/healthz"
    port         = "8080"
  }
}

resource "google_compute_instance_group_manager" "appserver" {
  name = "appserver-igm"

  base_instance_name = "app"
  instance_template  = "${google_compute_instance_template.appserver.self_link}"
  update_strategy    = "NONE"
  zone               = "us-central1-a"

  target_pools = ["${google_compute_target_pool.appserver.self_link}"]
  target_size  = 2

  named_port {
    name = "customHTTP"
    port = 8888
  }

  auto_healing_policies {
    health_check      = "${google_compute_health_check.autohealing.self_link}"
    initial_delay_sec = 300
  }
}
```

## » Argument Reference

The following arguments are supported:

- `base_instance_name` - (Required) The base instance name to use for instances in this group. The value must be a valid RFC1035 name. Supported characters are lowercase letters, numbers, and hyphens (-). Instances are named by appending a hyphen and a random four-character string to the base instance name.
- `instance_template` - (Required) The full URL to an instance template from which all new instances will be created.
- `name` - (Required) The name of the instance group manager. Must be 1-63 characters long and comply with RFC1035. Supported characters include lowercase letters, numbers, and hyphens.
- `zone` - (Required) The zone that instances in this group should be created in.

---

- `description` - (Optional) An optional textual description of the instance group manager.
- `named_port` - (Optional) The named port configuration. See the section below for details on configuration.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `update_strategy` - (Optional, Default `"RESTART"`) If the `instance_template` resource is modified, a value of `"NONE"` will prevent any of the managed instances from being restarted by Terraform. A value of `"RESTART"` will restart all of the instances at once. `"ROLLING_UPDATE"` is supported as [Beta feature]. A value of `"ROLLING_UPDATE"` requires `rolling_update_policy` block to be set
- `target_size` - (Optional) The target number of running instances for this managed instance group. This value should always be explicitly set unless this resource is attached to an autoscaler, in which case it should never be set. Defaults to `0`.
- `target_pools` - (Optional) The full URL of all target pools to which new instances in the group are added. Updating the target pools attribute does not affect existing instances.

---

- `auto_healing_policies` - (Optional, Beta) The autohealing policies for this managed instance group. You can specify only one value. Structure is documented below. For more information, see the official documentation.
- `rolling_update_policy` - (Optional, Beta) The update policy for this managed instance group. Structure is documented below. For more information, see the official documentation and API

The **rolling_update_policy** block supports:

```
rolling_update_policy{
  type = "PROACTIVE"
  minimal_action = "REPLACE"
  max_surge_percent = 20
  max_unavailable_fixed = 2
  min_ready_sec = 50
}
```

- `minimal_action` - (Required) - Minimal action to be taken on an instance. Valid values are `"RESTART"`, `"REPLACE"`
- `type` - (Required) - The type of update. Valid values are `"OPPORTUNISTIC"`, `"PROACTIVE"`

- `max_surge_fixed` - (Optional), The maximum number of instances that can be created above the specified targetSize during the update process. Conflicts with `max_surge_percent`. If neither is set, defaults to 1
- `max_surge_percent` - (Optional), The maximum number of instances(calculated as percentage) that can be created above the specified targetSize during the update process. Conflicts with `max_surge_fixed`.
- `max_unavailable_fixed` - (Optional), The maximum number of instances that can be unavailable during the update process. Conflicts with `max_unavailable_percent`. If neither is set, defaults to 1
- `max_unavailable_percent` - (Optional), The maximum number of instances(calculated as percentage) that can be unavailable during the update process. Conflicts with `max_unavailable_fixed`.
- `min_ready_sec` - (Optional), Minimum number of seconds to wait for after a newly created instance becomes available. This value must be from range [0, 3600]

---

The **named_port** block supports: (Include a `named_port` block for each named-port required).

- `name` - (Required) The name of the port.
- `port` - (Required) The port number.

---

The **auto_healing_policies** block supports:

- `health_check` - (Required) The health check resource that signals autohealing.
- `initial_delay_sec` - (Required) The number of seconds that the managed instance group waits before it applies autohealing policies to new instances or recently recreated instances. Between 0 and 3600.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `fingerprint` - The fingerprint of the instance group manager.
- `instance_group` - The full URL of the instance group created by the manager.
- `self_link` - The URL of the created resource.

### » Import

Instance group managers can be imported using the `name`, e.g.

```
$ terraform import google_compute_instance_group_manager.appserver appserver-igm
```

## » google_compute_instance_template

Manages a VM instance template resource within GCE. For more information see the official documentation and API.

### » Example Usage

```
resource "google_compute_instance_template" "default" {
  name        = "appserver-template"
  description = "This template is used to create app server instances."

  tags = ["foo", "bar"]

  labels = {
    environment = "dev"
  }

  instance_description = "description assigned to instances"
  machine_type         = "n1-standard-1"
  can_ip_forward       = false

  scheduling {
    automatic_restart   = true
    on_host_maintenance = "MIGRATE"
  }

  // Create a new boot disk from an image
  disk {
    source_image = "debian-cloud/debian-8"
    auto_delete  = true
    boot         = true
  }

  // Use an existing disk resource
  disk {
    source      = "foo_existing_disk"
    auto_delete = false
    boot        = false
  }

  network_interface {
    network = "default"
  }

  metadata {
    foo = "bar"
  }

  service_account {
    scopes = ["userinfo-email", "compute-ro", "storage-ro"]
  }
}
```

## » Using with Instance Group Manager

Instance Templates cannot be updated after creation with the Google Cloud Platform API. In order to update an Instance Template, Terraform will destroy the existing resource and create a replacement. In order to effectively use an Instance Template resource with an Instance Group Manager resource, it's recommended to specify `create_before_destroy` in a lifecycle block. Either omit the Instance Template `name` attribute, or specify a partial name with `name_prefix`. Example:

```
resource "google_compute_instance_template" "instance_template" {
  name_prefix  = "instance-template-"
  machine_type = "n1-standard-1"
  region       = "us-central1"

  // boot disk
  disk {
    # ...
  }

  // networking
  network_interface {
    # ...
  }

  lifecycle {
    create_before_destroy = true
  }
}
```

```
resource "google_compute_instance_group_manager" "instance_group_manager" {
  name               = "instance-group-manager"
  instance_template  = "${google_compute_instance_template.instance_template.self_link}"
  base_instance_name = "instance-group-manager"
  zone               = "us-central1-f"
  target_size        = "1"
}
```

With this setup Terraform generates a unique name for your Instance Template and can then update the Instance Group manager without conflict before destroying the previous Instance Template.

## » Argument Reference

Note that changing any field for this resource forces a new resource to be created.

The following arguments are supported:

- `disk` - (Required) Disks to attach to instances created from this template. This can be specified multiple times for multiple disks. Structure is documented below.
- `machine_type` - (Required) The machine type to create.

---

- `name` - (Optional) The name of the instance template. If you leave this blank, Terraform will auto-generate a unique name.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `can_ip_forward` - (Optional) Whether to allow sending and receiving of packets with non-matching source or destination IPs. This defaults to false.
- `description` - (Optional) A brief description of this resource.
- `instance_description` - (Optional) A brief description to use for instances created from this template.
- `labels` - (Optional) A set of key/value label pairs to assign to instances created from this template,
- `metadata` - (Optional) Metadata key/value pairs to make available from within instances created from this template.
- `metadata_startup_script` - (Optional) An alternative to using the startup-script metadata key, mostly to match the compute_instance resource. This replaces the startup-script metadata key on the created

instance and thus the two mechanisms are not allowed to be used simultaneously.

- `network_interface` - (Required) Networks to attach to instances created from this template. This can be specified multiple times for multiple networks. Structure is documented below.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) An instance template is a global resource that is not bound to a zone or a region. However, you can still specify some regional resources in an instance template, which restricts the template to the region where that resource resides. For example, a custom `subnetwork` resource is tied to a specific region. Defaults to the region of the Provider if no value is given.
- `scheduling` - (Optional) The scheduling strategy to use. More details about this configuration option are detailed below.
- `service_account` - (Optional) Service account to attach to the instance. Structure is documented below.
- `tags` - (Optional) Tags to attach to the instance.
- `guest_accelerator` - (Optional) List of the type and count of accelerator cards attached to the instance. Structure documented below.
- `min_cpu_platform` - (Optional) Specifies a minimum CPU platform. Applicable values are the friendly names of CPU platforms, such as `Intel Haswell` or `Intel Skylake`. See the complete list here.

The `disk` block supports:

- `auto_delete` - (Optional) Whether or not the disk should be auto-deleted. This defaults to true.
- `boot` - (Optional) Indicates that this is a boot disk.
- `device_name` - (Optional) A unique device name that is reflected into the /dev/ tree of a Linux operating system running within the instance. If not specified, the server chooses a default device name to apply to this disk.
- `disk_name` - (Optional) Name of the disk. When not provided, this defaults to the name of the instance.
- `source_image` - (Required if source not set) The image from which to initialize this disk. This can be one of: the image's `self_link`, `projects/{project}/global/images/{image}`, `projects/{project}/global/images/family/{famil` `global/images/{image}`, `global/images/family/{family}`, `family/{family}`, `{project}/{family}`, `{project}/{image}`, `{family}`, or `{image}`.

- interface - (Optional) Specifies the disk interface to use for attaching this disk.
- mode - (Optional) The mode in which to attach this disk, either READ_WRITE or READ_ONLY. If you are attaching or creating a boot disk, this must read-write mode.
- source - (Required if source_image not set) The name of the disk (such as those managed by google_compute_disk) to attach.
- disk_type - (Optional) The GCE disk type. Can be either "pd-ssd", "local-ssd", or "pd-standard".
- disk_size_gb - (Optional) The size of the image in gigabytes. If not specified, it will inherit the size of its base image.
- type - (Optional) The type of GCE disk, can be either "SCRATCH" or "PERSISTENT".

The network_interface block supports:

- network - (Optional) The name or self_link of the network to attach this interface to. Use network attribute for Legacy or Auto subnetted networks and subnetwork for custom subnetted networks.
- subnetwork - (Optional) the name of the subnetwork to attach this interface to. The subnetwork must exist in the same region this instance will be created in. Either network or subnetwork must be provided.
- subnetwork_project - (Optional) The ID of the project in which the subnetwork belongs. If it is not provided, the provider project is used.
- address - (Optional) The private IP address to assign to the instance. If empty, the address will be automatically assigned.
- access_config - (Optional) Access configurations, i.e. IPs via which this instance can be accessed via the Internet. Omit to ensure that the instance is not accessible from the Internet (this means that ssh provisioners will not work unless you are running Terraform can send traffic to the instance's network (e.g. via tunnel or because it is running on another cloud instance on that network). This block can be repeated multiple times. Structure documented below.
- alias_ip_range - (Optional) An array of alias IP ranges for this network interface. Can only be specified for network interfaces on subnet-mode networks. Structure documented below.

The access_config block supports:

- nat_ip - (Optional) The IP address that will be 1:1 mapped to the instance's network ip. If not given, one will be generated.

The alias_ip_range block supports:

- `ip_cidr_range` - The IP CIDR range represented by this alias IP range. This IP CIDR range must belong to the specified subnetwork and cannot contain IP addresses reserved by system or used by other network interfaces. At the time of writing only a netmask (e.g. /24) may be supplied, with a CIDR format resulting in an API error.
- `subnetwork_range_name` - (Optional) The subnetwork secondary range name specifying the secondary range from which to allocate the IP CIDR range for this alias IP range. If left unspecified, the primary range of the subnetwork will be used.

The `service_account` block supports:

- `email` - (Optional) The service account e-mail address. If not given, the default Google Compute Engine service account is used.
- `scopes` - (Required) A list of service scopes. Both OAuth2 URLs and gcloud short names are supported. To allow full access to all Cloud APIs, use the `cloud-platform` scope. See a complete list of scopes here.

The `scheduling` block supports:

- `automatic_restart` - (Optional) Specifies whether the instance should be automatically restarted if it is terminated by Compute Engine (not terminated by a user). This defaults to true.
- `on_host_maintenance` - (Optional) Defines the maintenance behavior for this instance.
- `preemptible` - (Optional) Allows instance to be preempted. This defaults to false. Read more on this here.

The `guest_accelerator` block supports:

- `type` (Required) - The accelerator type resource to expose to this instance. E.g. `nvidia-tesla-k80`.
- `count` (Required) - The number of the guest accelerator cards exposed to this instance.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `metadata_fingerprint` - The unique fingerprint of the metadata.
- `self_link` - The URI of the created resource.
- `tags_fingerprint` - The unique fingerprint of the tags.

### » Import

Instance templates can be imported using the `name`, e.g.

```
$ terraform import google_compute_instance_template.default appserver-template
```

## » google_compute_network_peering

Manages a network peering within GCE. For more information see the official documentation and API.

**Note:** Both network must create a peering with each other for the peering to be functional.

**Note:** Subnets IP ranges across peered VPC networks cannot overlap.

### » Example Usage

```
resource "google_compute_network_peering" "peering1" {
  name = "peering1"
  network = "${google_compute_network.default.self_link}"
  peer_network = "${google_compute_network.other.self_link}"
}

resource "google_compute_network_peering" "peering2" {
  name = "peering2"
  network = "${google_compute_network.other.self_link}"
  peer_network = "${google_compute_network.default.self_link}"
}

resource "google_compute_network" "default" {
  name                    = "foobar"
  auto_create_subnetworks = "false"
}

resource "google_compute_network" "other" {
  name                    = "other"
  auto_create_subnetworks = "false"
}
```

### » Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the peering.
- `network` - (Required) Resource link of the network to add a peering to.
- `peer_network` - (Required) Resource link of the peer network.
- `auto_create_routes` - (Optional) If set to `true`, the routes between the two networks will be created and managed automatically. Defaults to `true`.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `state` - State for the peering.
- `state_details` - Details about the current state of the peering.

# » google_compute_network

Manages a network within GCE. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_network" "default" {
  name                    = "foobar"
  auto_create_subnetworks = "true"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `auto_create_subnetworks` - (Optional) If set to true, this network will be created in auto subnet mode, and Google will create a subnet for each region automatically. If set to false, a custom subnetted network will be created that can support `google_compute_subnetwork` resources. Defaults to true.

- `ipv4_range` - (Optional) If set to a CIDR block, uses the legacy VPC API with the specified range. This API is deprecated. If set, `auto_create_subnetworks` must be explicitly set to false.
- `routing_mode` - (Optional) Sets the network-wide routing mode for Cloud Routers to use. Accepted values are `"GLOBAL"` or `"REGIONAL"`. Defaults to `"REGIONAL"`. Refer to the Cloud Router documentation for more details.
- `description` - (Optional) A brief description of this resource.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `gateway_ipv4` - The IPv4 address of the gateway.
- `name` - The unique name of the network.
- `self_link` - The URI of the created resource.

## » Import

Networks can be imported using the `name`, e.g.

```
$ terraform import google_compute_network.default foobar
```

# » google_compute_project_metadata

Manages metadata common to all instances for a project in GCE. For more information see the official documentation and API.

**Note:** If you want to manage only single key/value pairs within the project metadata rather than the entire set, then use google_compute_project_metadata_item.

## » Example Usage

```
resource "google_compute_project_metadata" "default" {
  metadata {
    foo  = "bar"
    fizz = "buzz"
    "13" = "42"
  }
}
```

### » Argument Reference

The following arguments are supported:

- `metadata` - (Required) A series of key value pairs. Changing this resource updates the GCE state.

---

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

### » Attributes Reference

Only the arguments listed above are exposed as attributes.

## » google_compute_project_metadata_item

Manages a single key/value pair on metadata common to all instances for a project in GCE. Using `google_compute_project_metadata_item` lets you manage a single key/value setting in Terraform rather than the entire project metadata map.

### » Example Usage

```
resource "google_compute_project_metadata_item" "default" {
  key = "my_metadata"
  value = "my_value"
}
```

### » Argument Reference

The following arguments are supported:

- `key` - (Required) The metadata key to set.
- `value` - (Required) The value to set for the given metadata key.

---

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

Only the arguments listed above are exposed as attributes.

## » Import

Project metadata items can be imported using the `key`, e.g.

```
$ terraform import google_compute_project_metadata_item.default my_metadata
```

# » google_compute_region_autoscaler

A Compute Engine Regional Autoscaler automatically adds or removes virtual machines from a managed instance group based on increases or decreases in load. This allows your applications to gracefully handle increases in traffic and reduces cost when the need for resources is lower. You just define the autoscaling policy and the autoscaler performs automatic scaling based on the measured load. For more information, see the official documentation and API

## » Example Usage

```
resource "google_compute_instance_template" "foobar" {
  name           = "foobar"
  machine_type   = "n1-standard-1"
  can_ip_forward = false

  tags = ["foo", "bar"]

  disk {
    source_image = "debian-cloud/debian-8"
  }

  network_interface {
    network = "default"
  }

  metadata {
    foo = "bar"
  }
  service_account {
    scopes = ["userinfo-email", "compute-ro", "storage-ro"]
  }
}

resource "google_compute_target_pool" "foobar" {
  name = "foobar"
}

resource "google_compute_region_instance_group_manager" "foobar" {
  name   = "foobar"
  region = "us-central1"

  instance_template  = "${google_compute_instance_template.foobar.self_link}"
  target_pools       = ["${google_compute_target_pool.foobar.self_link}"]
  base_instance_name = "foobar"
}

resource "google_compute_region_autoscaler" "foobar" {
  name   = "scaler"
  region = "us-central1"
  target = "${google_compute_region_instance_group_manager.foobar.self_link}"

  autoscaling_policy = {
    max_replicas    = 5
    min_replicas    = 1
    cooldown_period = 60

    cpu_utilization {
      target = 0.5
    }
  }
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the autoscaler.
- `target` - (Required) The full URL to the instance group manager whose size we control.
- `region` - (Required) The region of the target.

- `autoscaling_policy` - (Required) The parameters of the autoscaling algorithm. Structure is documented below.

---

- `description` - (Optional) An optional textual description of the instance group manager.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `autoscaling_policy` block contains:

- `max_replicas` - (Required) The group will never be larger than this.
- `min_replicas` - (Required) The group will never be smaller than this.
- `cooldown_period` - (Optional) Period to wait between changes. This should be at least double the time your instances take to start up.
- `cpu_utilization` - (Optional) A policy that scales when the cluster's average CPU is above or below a given threshold. Structure is documented below.
- `metric` - (Optional) A policy that scales according to Google Cloud Monitoring metrics Structure is documented below.
- `load_balancing_utilization` - (Optional) A policy that scales when the load reaches a proportion of a limit defined in the HTTP load balancer. Structure is documented below.

The `cpu_utilization` block contains:

- `target` - The floating point threshold where CPU utilization should be. E.g. for 50% one would specify 0.5.

The `metric` block contains (more documentation here):

- `name` - The name of the Google Cloud Monitoring metric to follow, e.g. `compute.googleapis.com/instance/network/received_bytes_count`
- `type` - Either ”cumulative”, ”delta”, or ”gauge”.
- `target` - The desired metric value per instance. Must be a positive value.

The `load_balancing_utilization` block contains:

- `target` - The floating point threshold where load balancing utilization should be. E.g. if the load balancer's `maxRatePerInstance` is 10 requests per second (RPS) then setting this to 0.5 would cause the group to be scaled such that each instance receives 5 RPS.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URL of the created resource.

## » Import

Autoscalers can be imported using the `name`, e.g.

```
$ terraform import google_compute_region_autoscaler.foobar scaler
```

# » google_compute_region_backend_service

A Region Backend Service defines a regionally-scoped group of virtual machines that will serve traffic for load balancing. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_region_backend_service" "foobar" {
  name             = "blablah"
  description      = "Hello World 1234"
  protocol         = "TCP"
  timeout_sec      = 10
  session_affinity = "CLIENT_IP"

  backend {
    group = "${google_compute_region_instance_group_manager.foo.instance_group}"
  }

  health_checks = ["${google_compute_health_check.default.self_link}"]
}

resource "google_compute_region_instance_group_manager" "foo" {
  name               = "terraform-test"
  instance_template  = "${google_compute_instance_template.foobar.self_link}"
  base_instance_name = "foobar"
  region             = "us-central1"
  target_size        = 1
}
```

```
resource "google_compute_instance_template" "foobar" {
  name         = "terraform-test"
  machine_type = "n1-standard-1"

  network_interface {
    network = "default"
  }

  disk {
    source_image = "debian-cloud/debian-8"
    auto_delete  = true
    boot         = true
  }
}

resource "google_compute_health_check" "default" {
  name               = "test"
  check_interval_sec = 1
  timeout_sec        = 1

  tcp_health_check {
    port = "80"
  }
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the backend service.
- `health_checks` - (Required) Specifies a list of health checks for checking the health of the backend service. Currently at most one health check can be specified, and a health check is required.

---

- `backend` - (Optional) The list of backends that serve this BackendService. Structure is documented below.
- `description` - (Optional) The textual description for the backend service.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `protocol` - (Optional) The protocol for incoming requests. Defaults to `HTTP`.

- `session_affinity` - (Optional) How to distribute load. Options are `NONE` (no affinity), `CLIENT_IP`, `CLIENT_IP_PROTO`, or `CLIENT_IP_PORT_PROTO`. Defaults to `NONE`.
- `region` - (Optional) The Region in which the created address should reside. If it is not provided, the provider region is used.
- `timeout_sec` - (Optional) The number of secs to wait for a backend to respond to a request before considering the request failed. Defaults to `30`.
- `connection_draining_timeout_sec` - (Optional) Time for which instance will be drained (not accept new connections, but still work to finish started ones). Defaults to `0`.

The `backend` block supports:

- `group` - (Required) The name or URI of a Compute Engine instance group (`google_compute_region_instance_group_manager.xyz.instance_group`) that can receive traffic. Instance groups must contain at least one instance.
- `description` - (Optional) Textual description for the backend.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `fingerprint` - The fingerprint of the backend service.
- `self_link` - The URI of the created resource.

# » google_compute_region_instance_group_manager

The Google Compute Engine Regional Instance Group Manager API creates and manages pools of homogeneous Compute Engine virtual machine instances from a common instance template. For more information, see the official documentation and API

**Note:** Use google_compute_instance_group_manager to create a single-zone instance group manager.

## » Example Usage

```
resource "google_compute_health_check" "autohealing" {
  name                = "autohealing-health-check"
  check_interval_sec  = 5
  timeout_sec         = 5
  healthy_threshold   = 2
  unhealthy_threshold = 10                         # 50 seconds

  http_health_check {
    request_path = "/healthz"
    port         = "8080"
  }
}

resource "google_compute_region_instance_group_manager" "appserver" {
  name = "appserver-igm"

  base_instance_name         = "app"
  instance_template          = "${google_compute_instance_template.appserver.self_link}"
  region                     = "us-central1"
  distribution_policy_zones  = ["us-central1-a", "us-central1-f"]

  target_pools = ["${google_compute_target_pool.appserver.self_link}"]
  target_size  = 2

  named_port {
    name = "custom"
    port = 8888
  }

  auto_healing_policies {
    health_check      = "${google_compute_health_check.autohealing.self_link}"
    initial_delay_sec = 300
  }
}
```

## » Argument Reference

The following arguments are supported:

- `base_instance_name` - (Required) The base instance name to use for instances in this group. The value must be a valid RFC1035 name. Supported characters are lowercase letters, numbers, and hyphens (-). Instances are named by appending a hyphen and a random four-character string to the base instance name.
- `instance_template` - (Required) The full URL to an instance template from which all new instances will be created.
- `name` - (Required) The name of the instance group manager. Must be 1-63 characters long and comply with RFC1035. Supported characters include

lowercase letters, numbers, and hyphens.

- `region` - (Required) The region where the managed instance group resides.

---

- `description` - (Optional) An optional textual description of the instance group manager.
- `named_port` - (Optional) The named port configuration. See the section below for details on configuration.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `target_size` - (Optional) The target number of running instances for this managed instance group. This value should always be explicitly set unless this resource is attached to an autoscaler, in which case it should never be set. Defaults to `0`.
- `target_pools` - (Optional) The full URL of all target pools to which new instances in the group are added. Updating the target pools attribute does not affect existing instances.

---

- `auto_healing_policies` - (Optional, Beta) The autohealing policies for this managed instance group. You can specify only one value. Structure is documented below. For more information, see the official documentation.
- `distribution_policy_zones` - (Optional, Beta) The distribution policy for this managed instance group. You can specify one or more values. For more information, see the official documentation.

The `named_port` block supports: (Include a `named_port` block for each named-port required).

- `name` - (Required) The name of the port.
- `port` - (Required) The port number.

The `auto_healing_policies` block supports:

- `health_check` - (Required) The health check resource that signals autohealing.
- `initial_delay_sec` - (Required) The number of seconds that the managed instance group waits before it applies autohealing policies to new instances or recently recreated instances. Between 0 and 3600.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `fingerprint` - The fingerprint of the instance group manager.
- `instance_group` - The full URL of the instance group created by the manager.
- `self_link` - The URL of the created resource.

## » Import

Instance group managers can be imported using the `name`, e.g.

```
$ terraform import google_compute_region_instance_group_manager.appserver appserver-igm
```

# » google_compute_route

Manages a network route within GCE. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_network" "default" {
  name = "compute-network"
}

resource "google_compute_subnetwork" "default" {
  name          = "compute-subnetwork"
  ip_cidr_range = "10.0.0.0/16"
  network       = "${google_compute_network.default.self_link}"
  region        = "us-central1"
}

resource "google_compute_route" "default" {
  name        = "network-route"
  dest_range  = "15.0.0.0/24"
  network     = "${google_compute_network.foobar.name}"
  next_hop_ip = "10.0.1.5"
  priority    = 100
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `dest_range` - (Required) The destination IPv4 address range that this route applies to.
- `network` - (Required) The name or self_link of the network to attach this route to.
- `priority` - (Optional) The priority of this route, used to break ties. Defaults to 1000.

---

- `next_hop_gateway` - (Optional) The URL of the internet gateway to route to if this route is matched. The alias "default-internet-gateway" can also be used.
- `next_hop_instance` - (Optional) The name of the VM instance to route to if this route is matched.
- `next_hop_instance_zone` - (Required when `next_hop_instance` is specified) The zone of the instance specified in `next_hop_instance`.
- `next_hop_ip` - (Optional) The IP address of the next hop if this route is matched.
- `next_hop_vpn_tunnel` - (Optional) The name of the VPN to route to if this route is matched.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `tags` - (Optional) The tags that this route applies to.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `next_hop_network` - The name of the next hop network, if available.
- `self_link` - The URI of the created resource.

## » Import

Network routes can be imported using the `name`, e.g.

```
$ terraform import google_compute_route.default network-route
```

## » google_compute_router

Manages a Cloud Router resource. For more information see the official documentation and API.

### » Example Usage

```
resource "google_compute_network" "foobar" {
  name = "network-1"
}

resource "google_compute_subnetwork" "foobar" {
  name          = "subnet-1"
  network       = "${google_compute_network.foobar.self_link}"
  ip_cidr_range = "10.0.0.0/16"
  region        = "us-central1"
}

resource "google_compute_address" "foobar" {
  name   = "vpn-gateway-1-address"
  region = "${google_compute_subnetwork.foobar.region}"
}

resource "google_compute_vpn_gateway" "foobar" {
  name    = "vpn-gateway-1"
  network = "${google_compute_network.foobar.self_link}"
  region  = "${google_compute_subnetwork.foobar.region}"
}

resource "google_compute_forwarding_rule" "foobar_esp" {
  name        = "vpn-gw-1-esp"
  region      = "${google_compute_vpn_gateway.foobar.region}"
  ip_protocol = "ESP"
  ip_address  = "${google_compute_address.foobar.address}"
  target      = "${google_compute_vpn_gateway.foobar.self_link}"
}

resource "google_compute_forwarding_rule" "foobar_udp500" {
  name        = "vpn-gw-1-udp-500"
  region      = "${google_compute_forwarding_rule.foobar_esp.region}"
  ip_protocol = "UDP"
  port_range  = "500-500"
  ip_address  = "${google_compute_address.foobar.address}"
  target      = "${google_compute_vpn_gateway.foobar.self_link}"
}

resource "google_compute_forwarding_rule" "foobar_udp4500" {
  name        = "vpn-gw-1-udp-4500"
  region      = "${google_compute_forwarding_rule.foobar_udp500.region}"
  ip_protocol = "UDP"
  port_range  = "4500-4500"
  ip_address  = "${google_compute_address.foobar.address}"
  target      = "${google_compute_vpn_gateway.foobar.self_link}"
}

resource "google_compute_router" "foobar" {
  name    = "router-1"
  region  = "${google_compute_forwarding_rule.foobar_udp500.region}"
  network = "${google_compute_network.foobar.self_link}"

  bgp {
    asn = 64512
  }
}

resource "google_compute_vpn_tunnel" "foobar" {
  name               = "vpn-tunnel-1"
  region             = "${google_compute_forwarding_rule.foobar_udp4500.region}"
  target_vpn_gateway = "${google_compute_vpn_gateway.foobar.self_link}"
  shared_secret      = "unguessable"
  peer_ip            = "8.8.8.8"
  router             = "${google_compute_router.foobar.name}"
}

resource "google_compute_router_interface" "foobar" {
  name       = "interface-1"
  router     = "${google_compute_router.foobar.name}"
  region     = "${google_compute_router.foobar.region}"
  ip_range   = "169.254.1.1/30"
  vpn_tunnel = "${google_compute_vpn_tunnel.foobar.name}"
}

resource "google_compute_router_peer" "foobar" {
  name                      = "peer-1"
  router                    = "${google_compute_router.foobar.name}"
  region                    = "${google_compute_router.foobar.region}"
  peer_ip_address           = "169.254.1.2"
  peer_asn                  = 65513
  advertised_route_priority = 100
  interface                 = "${google_compute_router_interface.foobar.name}"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the router, required by GCE. Changing this forces a new router to be created.
- `network` - (Required) The name or resource link to the network this Cloud Router will use to learn and announce routes. Changing this forces a new router to be created.
- `bgp` - (Required) BGP information specific to this router. Changing this forces a new router to be created. Structure is documented below.

---

- `description` - (Optional) A description of the resource. Changing this forces a new router to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used. Changing this forces a new router to be created.
- `region` - (Optional) The region this router should sit in. If not specified, the project region will be used. Changing this forces a new router to be created.

---

The `bgp` block supports:

- `asn` - (Required) Local BGP Autonomous System Number (ASN). Must be an RFC6996 private ASN.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.

## » Import

Routers can be imported using the `region` and `name`, e.g.

```
$ terraform import google_compute_router.foobar us-central1/router-1
```

# » google_compute_router_interface

Manages a Cloud Router interface. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_router_interface" "foobar" {
  name       = "interface-1"
  router     = "router-1"
  region     = "us-central1"
  ip_range   = "169.254.1.1/30"
  vpn_tunnel = "tunnel-1"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the interface, required by GCE. Changing this forces a new interface to be created.
- `router` - (Required) The name of the router this interface will be attached to. Changing this forces a new interface to be created.
- `vpn_tunnel` - (Required) The name or resource link to the VPN tunnel this interface will be linked to. Changing this forces a new interface to be created.

---

- `ip_range` - (Optional) IP address and range of the interface. The IP range must be in the RFC3927 link-local IP space. Changing this forces a new interface to be created.
- `project` - (Optional) The ID of the project in which this interface's router belongs. If it is not provided, the provider project is used. Changing this forces a new interface to be created.

- `region` - (Optional) The region this interface's router sits in. If not specified, the project region will be used. Changing this forces a new interface to be created.

## » Attributes Reference

Only the arguments listed above are exposed as attributes.

## » Import

Router interfaces can be imported using the `region`, `router`, and `name`, e.g.

```
$ terraform import google_compute_router_interface.foobar us-central1/router-1/interface-1
```

# » google_compute_router

Manages a Cloud Router BGP peer. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_router_peer" "foobar" {
  name                      = "peer-1"
  router                    = "router-1"
  region                    = "us-central1"
  peer_ip_address           = "169.254.1.2"
  peer_asn                  = 65513
  advertised_route_priority = 100
  interface                 = "interface-1"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for BGP peer, required by GCE. Changing this forces a new peer to be created.
- `router` - (Required) The name of the router in which this BGP peer will be configured. Changing this forces a new peer to be created.
- `interface` - (Required) The name of the interface the BGP peer is associated with. Changing this forces a new peer to be created.

- peer_ip_address - (Required) IP address of the BGP interface outside Google Cloud. Changing this forces a new peer to be created.
- peer_asn - (Required) Peer BGP Autonomous System Number (ASN). Changing this forces a new peer to be created.

---

- advertised_route_priority - (Optional) The priority of routes advertised to this BGP peer. Changing this forces a new peer to be created.
- project - (Optional) The ID of the project in which this peer's router belongs. If it is not provided, the provider project is used. Changing this forces a new peer to be created.
- region - (Optional) The region this peer's router sits in. If not specified, the project region will be used. Changing this forces a new peer to be created.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- ip_address - IP address of the interface inside Google Cloud Platform.

## » Import

Router BGP peers can be imported using the region, router, and name, e.g.

```
$ terraform import google_compute_router_peer.foobar us-central1/router-1/peer-1
```

# » google_compute_shared_vpc_host_project

Enables the Google Compute Engine Shared VPC feature for a project, assigning it as a Shared VPC host project.

For more information, see, the Project API documentation, where the Shared VPC feature is referred to by its former name "XPN".

## » Example Usage

```
# A host project provides network resources to associated service projects.
resource "google_compute_shared_vpc_host_project" "host" {
  project = "host-project-id"
}

# A service project gains access to network resources provided by its
# associated host project.
resource "google_compute_shared_vpc_service_project" "service1" {
  host_project    = "${google_compute_shared_vpc_host_project.host.project}"
  service_project = "service-project-id-1"
}
resource "google_compute_shared_vpc_service_project" "service2" {
  host_project    = "${google_compute_shared_vpc_host_project.host.project}"
  service_project = "service-project-id-2"
}
```

## » Argument Reference

The following arguments are expected:

- `project` - (Required) The ID of the project that will serve as a Shared VPC host project

## » Import

Google Compute Engine Shared VPC host project feature can be imported using the `project`, e.g.

```
$ terraform import google_compute_shared_vpc_host_project.host host-project-id
```

# » google_compute_shared_vpc_service_project

Enables the Google Compute Engine Shared VPC feature for a project, assigning it as a Shared VPC service project associated with a given host project.

For more information, see, the Project API documentation, where the Shared VPC feature is referred to by its former name ”XPN”.

## » Example Usage

```
resource "google_compute_shared_vpc_service_project" "service1" {
  host_project    = "host-project-id"
  service_project = "service-project-id-1"
}
```

For a complete Shared VPC example with both host and service projects, see `google_compute_shared_vpc_host_project`.

## » Argument Reference

The following arguments are expected:

- `host_project` - (Required) The ID of a host project to associate.
- `service_project` - (Required) The ID of the project that will serve as a Shared VPC service project.

## » Import

Google Compute Engine Shared VPC service project feature can be imported using the `host_project` and `service_project`, e.g.

```
$ terraform import google_compute_shared_vpc_service_project.service1 host-project-id/servic
```

# » google_compute_snapshot

Creates a new snapshot of a disk within GCE. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_snapshot" "default" {
  name        = "test-snapshot"
  source_disk = "test-disk"
  zone        = "us-central1-a"

  labels {
    my-label = "my-label-value"
  }
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `zone` - (Required) The zone where the source disk is located.
- `source_disk` - (Required) The disk which will be used as the source of the snapshot.

- `source_disk_encryption_key_raw` - (Optional) A 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to decrypt the source disk.
- `snapshot_encryption_key_raw` - (Optional) A 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to encrypt this snapshot.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `labels` - (Optional) A set of key/value label pairs to assign to the snapshot.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `snapshot_encryption_key_sha256` - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.
- `source_disk_encryption_key_sha256` - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects the source disk.
- `source_disk_link` - The URI of the source disk.
- `self_link` - The URI of the created resource.
- `label_fingerprint` - The unique fingerprint of the labels.

# » google_compute_ssl_certificate

Creates an SSL certificate resource necessary for HTTPS load balancing in GCE. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_ssl_certificate" "default" {
  name_prefix = "my-certificate-"
  description = "a description"
  private_key = "${file("path/to/private.key")}"
  certificate = "${file("path/to/certificate.crt")}"
}
```

## » Using with Target HTTPS Proxies

SSL certificates cannot be updated after creation. In order to apply the specified configuration, Terraform will destroy the existing resource and create a replacement. To effectively use an SSL certificate resource with a Target HTTPS Proxy resource, it's recommended to specify `create_before_destroy` in a lifecycle block. Either omit the Instance Template `name` attribute, or specify a partial name with `name_prefix`. Example:

```
resource "google_compute_ssl_certificate" "default" {
  name_prefix = "my-certificate-"
  description = "a description"
  private_key = "${file("path/to/private.key")}"
  certificate = "${file("path/to/certificate.crt")}"

  lifecycle {
    create_before_destroy = true
  }
}

resource "google_compute_target_https_proxy" "my_proxy" {
  name             = "public-proxy"
  url_map          = # ...
  ssl_certificates = ["${google_compute_ssl_certificate.default.self_link}"]
}
```

## » Argument Reference

The following arguments are supported:

- `certificate` - (Required) A local certificate file in PEM format. The chain may be at most 5 certs long, and must include at least one intermediate cert. Changing this forces a new resource to be created.
- `private_key` - (Required) Write only private key in PEM format. Changing this forces a new resource to be created.

---

- `name` - (Optional) A unique name for the SSL certificate. If you leave this blank, Terraform will auto-generate a unique name.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.
- `description` - (Optional) An optional description of this resource. Changing this forces a new resource to be created.

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `certificate_id` - A unique ID for the certificate, assigned by GCE.
- `self_link` - The URI of the created resource.

## » Import

SSL certificate can be imported using the `name`, e.g.

```
$ terraform import compute_ssl_certificate.html.foobar foobar
```

# » google_compute_subnetwork

Manages a subnetwork within GCE. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_subnetwork" "default-us-east1" {
  name          = "default-us-east1"
  ip_cidr_range = "10.0.0.0/16"
  network       = "${google_compute_network.default.self_link}"
  region        = "us-east1"
}

resource "google_compute_network" "default" {
  name = "test"
}
```

## » Argument Reference

The following arguments are supported:

- `ip_cidr_range` - (Required) The IP address range that machines in this network are assigned to, represented as a CIDR block.

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `network` - (Required) The network name or resource link to the parent network of this subnetwork. The parent network must have been created in custom subnet mode.

---

- `description` - (Optional) Description of this subnetwork.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) The region this subnetwork will be created in. If unspecified, this defaults to the region configured in the provider.
- `private_ip_google_access` - (Optional) Whether the VMs in this subnet can access Google services without assigned external IP addresses.

---

- `secondary_ip_range` - (Optional, Beta) An array of configurations for secondary IP ranges for VM instances contained in this subnetwork. Structure is documented below.

The `secondary_ip_range` block supports:

- `range_name` - (Required) The name associated with this subnetwork secondary range, used when adding an alias IP range to a VM instance.
- `ip_cidr_range` - (Required) The range of IP addresses belonging to this subnetwork secondary range. Ranges must be unique and non-overlapping with all primary and secondary IP ranges within a network.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `gateway_address` - The IP address of the gateway.
- `self_link` - The URI of the created resource.

## » Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is `6 minutes`
- `update` - Default is `6 minutes`
- `delete` - Default is `6 minutes`

### » Import

Subnetwork can be imported using the `region` and `name`, e.g.

```
$ terraform import google_compute_subnetwork.default-us-east1 us-east1/default-us-east1
```

## » google_compute_target_http_proxy

Creates a target HTTP proxy resource in GCE. For more information see the official documentation and API.

### » Example Usage

```
resource "google_compute_target_http_proxy" "default" {
  name        = "test-proxy"
  description = "a description"
  url_map     = "${google_compute_url_map.default.self_link}"
}

resource "google_compute_url_map" "default" {
  name        = "url-map"
  description = "a description"

  default_service = "${google_compute_backend_service.default.self_link}"

  host_rule {
    hosts        = ["mysite.com"]
    path_matcher = "allpaths"
  }

  path_matcher {
    name            = "allpaths"
    default_service = "${google_compute_backend_service.default.self_link}"

    path_rule {
      paths   = ["/*"]
      service = "${google_compute_backend_service.default.self_link}"
    }
  }
}

resource "google_compute_backend_service" "default" {
  name        = "default-backend"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = ["${google_compute_http_health_check.default.self_link}"]
}

resource "google_compute_http_health_check" "default" {
  name               = "test"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `url_map` - (Required) The URL of a URL Map resource that defines the mapping from the URL to the BackendService.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

---

- `description` - (Optional) A description of this resource. Changing this forces a new resource to be created.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `proxy_id` - A unique ID assigned by GCE.
- `self_link` - The URI of the created resource.

## » Import

Target HTTP Proxy can be imported using the `name`, e.g.

```
$ terraform import compute_target_http_proxy.foobar foobar
```

# » google_compute_target_https_proxy

Creates a target HTTPS proxy resource in GCE. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_target_https_proxy" "default" {
  name             = "test-proxy"
  description      = "a description"
  url_map          = "${google_compute_url_map.default.self_link}"
  ssl_certificates = ["${google_compute_ssl_certificate.default.self_link}"]
}

resource "google_compute_ssl_certificate" "default" {
  name        = "my-certificate"
  description = "a description"
  private_key = "${file("path/to/private.key")}"
  certificate = "${file("path/to/certificate.crt")}"
}

resource "google_compute_url_map" "default" {
  name        = "url-map"
  description = "a description"

  default_service = "${google_compute_backend_service.default.self_link}"

  host_rule {
    hosts        = ["mysite.com"]
    path_matcher = "allpaths"
  }

  path_matcher {
    name            = "allpaths"
    default_service = "${google_compute_backend_service.default.self_link}"

    path_rule {
      paths   = ["/*"]
      service = "${google_compute_backend_service.default.self_link}"
    }
  }
}

resource "google_compute_backend_service" "default" {
  name        = "default-backend"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = ["${google_compute_http_health_check.default.self_link}"]
}

resource "google_compute_http_health_check" "default" {
  name               = "test"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `ssl_certificates` - (Required) The URLs or names of the SSL Certificate resources that authenticate connections between users and load balancing.
- `url_map` - (Required) The URL of a URL Map resource that defines the mapping from the URL to the BackendService.

---

- `description` - (Optional) A description of this resource. Changing this forces a new resource to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `proxy_id` - A unique ID assigned by GCE.
- `self_link` - The URI of the created resource.

### » Import

Target HTTPS Proxy can be imported using the `name`, e.g.

```
$ terraform import compute_target_https_proxy.foobar foobar
```

## » google_compute_target_ssl_proxy

Creates a target SSL proxy resource in GCE. For more information see the official documentation and API.

### » Example Usage

```
resource "google_compute_target_ssl_proxy" "default" {
  name = "test"
  backend_service = "${google_compute_backend_service.default.self_link}"
  ssl_certificates = ["${google_compute_ssl_certificate.default.self_link}"]
}

resource "google_compute_ssl_certificate" "default" {
  name = "default-cert"
  private_key = "${file("path/to/test.key")}"
  certificate = "${file("path/to/test.crt")}"
}

resource "google_compute_backend_service" "default" {
  name = "default-backend"
  protocol    = "SSL"
  health_checks = ["${google_compute_health_check.default.self_link}"]
}

resource "google_compute_health_check" "default" {
  name = "default-health-check"
  check_interval_sec = 1
  timeout_sec = 1
  tcp_health_check {
    port = "443"
  }
}
```

### » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `backend_service` - (Required) The URL of a Backend Service resource to receive the matched traffic.
- `ssl_certificates` - (Required) The URLs or names of the SSL Certificate resources that authenticate connections between users and load balancing.

---

- `proxy_header` - (Optional) Type of proxy header to append before sending data to the backend, either NONE or PROXY_V1 (default NONE).
- `description` - (Optional) A description of this resource. Changing this forces a new resource to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `proxy_id` - A unique ID assigned by GCE.
- `self_link` - The URI of the created resource.

## » Import

SSL proxy can be imported using the `name`, e.g.

```
$ terraform import google_compute_target_ssl_proxy.default test
```

# » google_compute_target_tcp_proxy

Creates a target TCP proxy resource in GCE. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_target_tcp_proxy" "default" {
  name = "test"
  description = "test"
  backend_service = "${google_compute_backend_service.default.self_link}"
}

resource "google_compute_backend_service" "default" {
  name        = "default-backend"
  protocol    = "TCP"
  timeout_sec = 10

  health_checks = ["${google_compute_health_check.default.self_link}"]
}

resource "google_compute_health_check" "default" {
  name = "default"
  timeout_sec        = 1
  check_interval_sec = 1

  tcp_health_check {
    port = "443"
  }
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `backend_service` - (Required) The URL of a Backend Service resource to receive the matched traffic.

---

- `proxy_header` - (Optional) Type of proxy header to append before sending data to the backend, either NONE or PROXY_V1 (default NONE).
- `description` - (Optional) A description of this resource. Changing this forces a new resource to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `proxy_id` - A unique ID assigned by GCE.
- `self_link` - The URI of the created resource.

## » Import

TCP proxy can be imported using the `name`, e.g.

```
$ terraform import google_compute_target_tcp_proxy.default test
```

# » google_compute_target_pool

Manages a Target Pool within GCE. This is a collection of instances used as target of a network load balancer (Forwarding Rule). For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_target_pool" "default" {
  name = "instance-pool"

  instances = [
    "us-central1-a/myinstance1",
    "us-central1-b/myinstance2",
  ]

  health_checks = [
    "${google_compute_http_health_check.default.name}",
  ]
}

resource "google_compute_http_health_check" "default" {
  name               = "default"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `backup_pool` - (Optional) URL to the backup target pool. Must also set failover_ratio.
- `description` - (Optional) Textual description field.
- `failover_ratio` - (Optional) Ratio (0 to 1) of failed nodes before using the backup pool (which must also be set).
- `health_checks` - (Optional) List of zero or one health check name or self_link. Only legacy `google_compute_http_health_check` is supported.
- `instances` - (Optional) List of instances in the pool. They can be given as URLs, or in the form of ”zone/name”. Note that the instances need not exist at the time of target pool creation, so there is no need to use the Terraform interpolators to create a dependency on the instances from the target pool.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) Where the target pool resides. Defaults to project region.
- `session_affinity` - (Optional) How to distribute load. Options are ”NONE” (no affinity). ”CLIENT_IP” (hash of the source/dest addresses / ports), and ”CLIENT_IP_PROTO” also includes the protocol (default ”NONE”).

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.

## » Import

Target pools can be imported using the `name`, e.g.

```
$ terraform import google_compute_target_pool.default instance-pool
```

# » google_compute_url_map

Manages a URL Map resource within GCE. For more information see the official documentation and API.

## » Example Usage

```
resource "google_compute_url_map" "foobar" {
  name        = "urlmap"
  description = "a description"

  default_service = "${google_compute_backend_service.home.self_link}"

  host_rule {
    hosts        = ["mysite.com"]
    path_matcher = "allpaths"
  }

  path_matcher {
    name            = "allpaths"
    default_service = "${google_compute_backend_service.home.self_link}"

    path_rule {
      paths   = ["/home"]
      service = "${google_compute_backend_service.home.self_link}"
    }

    path_rule {
      paths   = ["/login"]
      service = "${google_compute_backend_service.login.self_link}"
    }

    path_rule {
      paths   = ["/static"]
      service = "${google_compute_backend_bucket.static.self_link}"
    }
  }

  test {
    service = "${google_compute_backend_service.home.self_link}"
    host    = "hi.com"
    path    = "/home"
  }
}
```

```
resource "google_compute_backend_service" "login" {
  name        = "login-backend"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = ["${google_compute_http_health_check.default.self_link}"]
}

resource "google_compute_backend_service" "home" {
  name        = "home-backend"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = ["${google_compute_http_health_check.default.self_link}"]
}

resource "google_compute_http_health_check" "default" {
  name               = "test"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}

resource "google_compute_backend_bucket" "static" {
  name        = "static-asset-backend-bucket"
  bucket_name = "${google_storage_bucket.static.name}"
  enable_cdn  = true
}

resource "google_storage_bucket" "static" {
  name     = "static-asset-bucket"
  location = "US"
}
```

## » Argument Reference

The following arguments are supported:

- `default_service` - (Required) The backend service or backend bucket to use when none of the given rules match.
- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

- `description` - (Optional) A brief description of this resource.
- `host_rule` - (Optional) A list of host rules. Multiple blocks of this type are permitted. Structure is documented below.
- `path_matcher` - (Optional) A list of paths to match. Structure is documented below.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `test` - (Optional) The test to perform. Multiple blocks of this type are permitted. Structure is documented below.

The `host_rule` block supports:

- `hosts` (Required) - The list of host patterns to match.
- `description` - (Optional) An optional description of the host rule.
- `path_matcher` - (Required) The name of the `path_matcher` to apply this host rule to.

The `path_matcher` block supports:

- `name` - (Required) The name of the `path_matcher` resource.
- `default_service` - (Required) The backend service or backend bucket to use if none of the given paths match.
- `description` - (Optional) An optional description of the host rule.
- `path_rule` - (Optional) A list of path rules. Multiple blocks of this type are permitted. Structure is documented below.

The `path_rule` block supports:

- `paths` - (Required) The list of paths to match against.
- `service` - (Required) The backend service or backend bucket to use if any of the given paths match.

The `test` block supports:

- `service` - (Required) The backend service or backend bucket link that should be matched by this test.
- `host` - (Required) The host component of the URL being tested.
- `path` - (Required) The path component of the URL being tested.
- `description` - (Optional) An optional description of this test.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `fingerprint` - The unique fingerprint for this resource.
- `map_id` - The GCE assigned ID of the resource.
- `self_link` - The URI of the created resource.

## » Import

URL Map can be imported using the `name`, e.g.

```
$ terraform import google_compute_url_map.html.foobar foobar
```

Currently `host_rule`, `path_matcher` and `test` importing is not yet supported.

# » google_compute_vpn_gateway

Manages a VPN Gateway in the GCE network. For more info, read the documentation.

## » Example Usage

```
resource "google_compute_network" "network1" {
  name       = "network1"
  ipv4_range = "10.120.0.0/16"
}

resource "google_compute_vpn_gateway" "target_gateway" {
  name    = "vpn1"
  network = "${google_compute_network.network1.self_link}"
  region  = "${var.region}"
}

resource "google_compute_address" "vpn_static_ip" {
  name   = "vpn-static-ip"
  region = "${var.region}"
}

resource "google_compute_forwarding_rule" "fr_esp" {
  name        = "fr-esp"
  region      = "${var.region}"
  ip_protocol = "ESP"
  ip_address  = "${google_compute_address.vpn_static_ip.address}"
  target      = "${google_compute_vpn_gateway.target_gateway.self_link}"
}

resource "google_compute_forwarding_rule" "fr_udp500" {
  name        = "fr-udp500"
  region      = "${var.region}"
  ip_protocol = "UDP"
  port_range  = "500"
  ip_address  = "${google_compute_address.vpn_static_ip.address}"
  target      = "${google_compute_vpn_gateway.target_gateway.self_link}"
}

resource "google_compute_forwarding_rule" "fr_udp4500" {
  name        = "fr-udp4500"
  region      = "${var.region}"
  ip_protocol = "UDP"
  port_range  = "4500"
  ip_address  = "${google_compute_address.vpn_static_ip.address}"
  target      = "${google_compute_vpn_gateway.target_gateway.self_link}"
}

resource "google_compute_vpn_tunnel" "tunnel1" {
  name          = "tunnel1"
  region        = "${var.region}"
  peer_ip       = "15.0.0.120"
  shared_secret = "a secret message"

  target_vpn_gateway = "${google_compute_vpn_gateway.target_gateway.self_link}"

  depends_on = [
    "google_compute_forwarding_rule.fr_esp",
    "google_compute_forwarding_rule.fr_udp500",
    "google_compute_forwarding_rule.fr_udp4500",
  ]
}

resource "google_compute_route" "route1" {
  name       = "route1"
  network    = "${google_compute_network.network1.name}"
  dest_range = "15.0.0.0/24"
  priority   = 1000

  next_hop_vpn_tunnel = "${google_compute_vpn_tunnel.tunnel1.self_link}"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `network` - (Required) The name or resource link to the network this VPN gateway is accepting traffic for. Changing this forces a new resource to be created.

---

- `description` - (Optional) A description of the resource. Changing this forces a new resource to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) The region this gateway should sit in. If not specified, the project region will be used. Changing this forces a new resource to be created.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.

# » google_compute_vpn_tunnel

Manages a VPN Tunnel to the GCE network. For more info, read the documentation.

**Note:** All arguments including the `shared_secret` will be stored in the raw state as plain-text. Read more about sensitive data in state.

## » Example Usage

```
resource "google_compute_network" "network1" {
  name = "network1"
}

resource "google_compute_subnetwork" "subnet1" {
  name          = "subnet1"
  network       = "${google_compute_network.network1.self_link}"
  ip_cidr_range = "10.120.0.0/16"
  region        = "us-central1"
}

resource "google_compute_vpn_gateway" "target_gateway" {
  name    = "vpn1"
  network = "${google_compute_network.network1.self_link}"
  region  = "${google_compute_subnetwork.subnet1.region}"
}

resource "google_compute_address" "vpn_static_ip" {
  name   = "vpn-static-ip"
  region = "${google_compute_subnetwork.subnet1.region}"
}

resource "google_compute_forwarding_rule" "fr_esp" {
  name        = "fr-esp"
  ip_protocol = "ESP"
  ip_address  = "${google_compute_address.vpn_static_ip.address}"
  target      = "${google_compute_vpn_gateway.target_gateway.self_link}"
}

resource "google_compute_forwarding_rule" "fr_udp500" {
  name        = "fr-udp500"
  ip_protocol = "UDP"
  port_range  = "500-500"
  ip_address  = "${google_compute_address.vpn_static_ip.address}"
  target      = "${google_compute_vpn_gateway.target_gateway.self_link}"
}

resource "google_compute_forwarding_rule" "fr_udp4500" {
  name        = "fr-udp4500"
  ip_protocol = "UDP"
  port_range  = "4500-4500"
  ip_address  = "${google_compute_address.vpn_static_ip.address}"
  target      = "${google_compute_vpn_gateway.target_gateway.self_link}"
}

resource "google_compute_vpn_tunnel" "tunnel1" {
  name          = "tunnel1"
  peer_ip       = "15.0.0.120"
  shared_secret = "a secret message"

  target_vpn_gateway = "${google_compute_vpn_gateway.target_gateway.self_link}"
  local_traffic_selector  = ["${google_compute_subnetwork.subnet1.ip_cidr_range}"]
  remote_traffic_selector = ["172.16.0.0/12"]

  depends_on = [
    "google_compute_forwarding_rule.fr_esp",
    "google_compute_forwarding_rule.fr_udp500",
    "google_compute_forwarding_rule.fr_udp4500",
  ]
}

resource "google_compute_route" "route1" {
  name       = "route1"
  network    = "${google_compute_network.network1.name}"
  dest_range = "15.0.0.0/24"
  priority   = 1000

  next_hop_vpn_tunnel = "${google_compute_vpn_tunnel.tunnel1.self_link}"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.
- `peer_ip` - (Required) The VPN gateway sitting outside of GCE. Changing this forces a new resource to be created.
- `shared_secret` - (Required) A passphrase shared between the two VPN gateways. Changing this forces a new resource to be created.
- `target_vpn_gateway` - (Required) A link to the VPN gateway sitting inside GCE. Changing this forces a new resource to be created.

---

- `description` - (Optional) A description of the resource. Changing this forces a new resource to be created.
- `ike_version` - (Optional) Either version 1 or 2. Default is 2. Changing this forces a new resource to be created.
- `local_traffic_selector` - (Optional) Specifies which CIDR ranges are announced to the VPN peer. Mandatory if the VPN gateway is attached to a custom subnetted network. Refer to Google documentation for more information.

- `remote_traffic_selector` - (Optional) Specifies which CIDR ranges the VPN tunnel can route to the remote side. Mandatory if the VPN gateway is attached to a custom subnetted network. Refer to Google documentation for more information.
- `router` - (Optional) Name of a Cloud Router in the same region to be used for dynamic routing. Refer to Google documentation for more information.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) The region this tunnel should sit in. If not specified, the project region will be used. Changing this forces a new resource to be created.

### » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `detailed_status` - Information about the status of the VPN tunnel.
- `self_link` - The URI of the created resource.

## » google_container_cluster

Creates a Google Kubernetes Engine (GKE) cluster. For more information see the official documentation and API.

**Note:** All arguments including the username and password will be stored in the raw state as plain-text. Read more about sensitive data in state.

### » Example usage

```
resource "google_container_cluster" "primary" {
  name               = "marcellus-wallace"
  zone               = "us-central1-a"
  initial_node_count = 3

  additional_zones = [
    "us-central1-b",
    "us-central1-c",
  ]
  master_auth {
    username = "mr.yoda"
    password = "adoy.rm"
  }

  node_config {
    oauth_scopes = [
      "https://www.googleapis.com/auth/compute",
      "https://www.googleapis.com/auth/devstorage.read_only",
      "https://www.googleapis.com/auth/logging.write",
      "https://www.googleapis.com/auth/monitoring",
    ]

    labels {
      foo = "bar"
    }

    tags = ["foo", "bar"]
  }
}

# The following outputs allow authentication and connectivity to the GKE Cluster.
output "client_certificate" {
  value = "${google_container_cluster.primary.master_auth.0.client_certificate}"
}

output "client_key" {
  value = "${google_container_cluster.primary.master_auth.0.client_key}"
}

output "cluster_ca_certificate" {
  value = "${google_container_cluster.primary.master_auth.0.cluster_ca_certificate}"
}
```

## » Argument Reference

- `name` - (Required) The name of the cluster, unique within the project and zone.
- `zone` - (Required) The zone that the master and the number of nodes specified in `initial_node_count` should be created in.

---

- `additional_zones` - (Optional) The list of additional Google Compute Engine locations in which the cluster's nodes should be located.

If additional zones are configured, the number of nodes specified in `initial_node_count` is created in all specified zones.

- `addons_config` - (Optional) The configuration for addons supported by GKE. Structure is documented below.
- `cluster_ipv4_cidr` - (Optional) The IP address range of the kubernetes pods in this cluster. Default is an automatically assigned CIDR.
- `description` - (Optional) Description of the cluster.
- `enable_kubernetes_alpha` - (Optional) Whether to enable Kubernetes Alpha features for this cluster. Note that when this option is enabled, the cluster cannot be upgraded and will be automatically deleted after 30 days.
- `enable_legacy_abac` - (Optional) Whether the ABAC authorizer is enabled for this cluster. When enabled, identities in the system, including service accounts, nodes, and controllers, will have statically granted permissions beyond those provided by the RBAC configuration or IAM. Defaults to `true`
- `initial_node_count` - (Optional) The number of nodes to create in this cluster (not including the Kubernetes master). Must be set if `node_pool` is not set.
- `ip_allocation_policy` - (Optional) Configuration for cluster IP allocation. As of now, only pre-allocated subnetworks (custom type with secondary ranges) are supported. This will activate IP aliases. See the official documentation Structure is documented below.
- `logging_service` - (Optional) The logging service that the cluster should write logs to. Available options include `logging.googleapis.com` and `none`. Defaults to `logging.googleapis.com`
- `maintenance_policy` - (Optional) The maintenance policy to use for the cluster. Structure is documented below.
- `master_auth` - (Optional) The authentication information for accessing the Kubernetes master. Structure is documented below.
- `master_authorized_networks_config` - (Optional) The desired configuration options for master authorized networks. Omit the nested `cidr_blocks` attribute to disallow external access (except the cluster node IPs, which GKE automatically whitelists).
- `min_master_version` - (Optional) The minimum version of the master. GKE will auto-update the master to new versions, so this does not guarantee the current master version--use the read-only `master_version` field to obtain that. If unset, the cluster's version will be set by GKE to the version of the most recent official release (which is not necessarily the latest version).

- monitoring_service - (Optional) The monitoring service that the cluster should write metrics to. Automatically send metrics from pods in the cluster to the Google Cloud Monitoring API. VM metrics will be collected by Google Compute Engine regardless of this setting Available options include monitoring.googleapis.com and none. Defaults to monitoring.googleapis.com

- network - (Optional) The name or self_link of the Google Compute Engine network to which the cluster is connected.

- network_policy - (Optional) Configuration options for the NetworkPolicy feature. Structure is documented below.

- node_config - (Optional) Parameters used in creating the cluster's nodes. Structure is documented below.

- node_pool - (Optional) List of node pools associated with this cluster. See google_container_node_pool for schema.

- node_version - (Optional) The Kubernetes version on the nodes. Must either be unset or set to the same value as min_master_version on create. Defaults to the default version set by GKE which is not necessarily the latest version.

- pod_security_policy_config - (Optional, Beta) Configuration for the PodSecurityPolicy feature. Structure is documented below.

- project - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

- subnetwork - (Optional) The name of the Google Compute Engine subnetwork in which the cluster's instances are launched.

The addons_config block supports:

- horizontal_pod_autoscaling - (Optional) The status of the Horizontal Pod Autoscaling addon, which increases or decreases the number of replica pods a replication controller has based on the resource usage of the existing pods. It ensures that a Heapster pod is running in the cluster, which is also used by the Cloud Monitoring service. It is enabled by default; set disabled = true to disable.
- http_load_balancing - (Optional) The status of the HTTP (L7) load balancing controller addon, which makes it easy to set up HTTP load balancers for services in a cluster. It is enabled by default; set disabled = true to disable.
- kubernetes_dashboard - (Optional) The status of the Kubernetes Dashboard add-on, which controls whether the Kubernetes Dashboard is enabled for this cluster. It is enabled by default; set disabled = true to disable.
- network_policy_config - (Optional) Whether we should enable the network policy addon for the master. This must be enabled in order to enable

network policy for the nodes. It can only be disabled if the nodes already do not have network policies enabled. Set `disabled = true` to disable.

This example `addons_config` disables two addons:

```
addons_config {
  http_load_balancing {
    disabled = true
  }
  horizontal_pod_autoscaling {
    disabled = true
  }
}
```

The `maintenance_policy` block supports:

- `daily_maintenance_window` - (Required) Time window specified for daily maintenance operations. Specify `start_time` in RFC3339 format ”HH:MM”, where HH : [00-23] and MM : [00-59] GMT. For example:

```
maintenance_policy {
  daily_maintenance_window {
    start_time = "03:00"
  }
}
```

The `ip_allocation_policy` block supports:

- `cluster_secondary_range_name` - (Optional) The name of the secondary range to be used as for the cluster CIDR block. The secondary range will be used for pod IP addresses. This must be an existing secondary range associated with the cluster subnetwork.
- `services_secondary_range_name` - (Optional) The name of the secondary range to be used as for the services CIDR block. The secondary range will be used for service ClusterIPs. This must be an existing secondary range associated with the cluster subnetwork.

The `master_auth` block supports:

- `password` - (Required) The password to use for HTTP basic authentication when accessing the Kubernetes master endpoint
- `username` - (Required) The username to use for HTTP basic authentication when accessing the Kubernetes master endpoint

If this block is provided and both `username` and `password` are empty, basic authentication will be disabled.

The `master_authorized_networks_config` block supports:

- `cidr_blocks` - (Optional) Defines up to 10 external networks that can access Kubernetes master through HTTPS.

The `master_authorized_networks_config.cidr_blocks` block supports:

- `cidr_block` - (Optional) External network that can access Kubernetes master through HTTPS. Must be specified in CIDR notation.
- `display_name` - (Optional) Field for users to identify CIDR blocks.

The `network_policy` block supports:

- `provider` - (Optional) The selected network policy provider. Defaults to PROVIDER_UNSPECIFIED.
- `enabled` - (Optional) Whether network policy is enabled on the cluster. Defaults to false.

The `node_config` block supports:

- `disk_size_gb` - (Optional) Size of the disk attached to each node, specified in GB. The smallest allowed disk size is 10GB. Defaults to 100GB.
- `guest_accelerator` - (Optional) List of the type and count of accelerator cards attached to the instance. Structure documented below.
- `image_type` - (Optional) The image type to use for this node.
- `labels` - (Optional) The Kubernetes labels (key/value pairs) to be applied to each node.
- `local_ssd_count` - (Optional) The amount of local SSD disks that will be attached to each cluster node. Defaults to 0.
- `machine_type` - (Optional) The name of a Google Compute Engine machine type. Defaults to `n1-standard-1`.
- `metadata` - (Optional) The metadata key/value pairs assigned to instances in the cluster.
- `min_cpu_platform` - (Optional) Minimum CPU platform to be used by this instance. The instance may be scheduled on the specified or newer CPU platform. Applicable values are the friendly names of CPU platforms, such as `Intel Haswell`. See the official documentation for more information.
- `oauth_scopes` - (Optional) The set of Google API scopes to be made available on all of the node VMs under the ”default” service account. These can be either FQDNs, or scope aliases. The following scopes are necessary to ensure the correct functioning of the cluster:
  - `compute-rw` (`https://www.googleapis.com/auth/compute`)
  - `storage-ro` (`https://www.googleapis.com/auth/devstorage.read_only`)
  - `logging-write` (`https://www.googleapis.com/auth/logging.write`), if `logging_service` points to Google
  - `monitoring` (`https://www.googleapis.com/auth/monitoring`), if `monitoring_service` points to Google

- `preemptible` - (Optional) A boolean that represents whether or not the underlying node VMs are preemptible. See the official documentation for more information. Defaults to false.
- `service_account` - (Optional) The service account to be used by the Node VMs. If not specified, the ”default” service account is used.
- `tags` - (Optional) The list of instance tags applied to all nodes. Tags are used to identify valid sources or targets for network firewalls.
- `workload_metadata_config` - (Optional) Metadata configuration to expose to workloads on the node pool. Structure is documented below.

The `guest_accelerator` block supports:

- `type` (Required) - The accelerator type resource to expose to this instance. E.g. `nvidia-tesla-k80`.
- `count` (Required) - The number of the guest accelerator cards exposed to this instance.

The `pod_security_policy_config` block supports:

- `enabled` (Required) - Enable the PodSecurityPolicy controller for this cluster. If enabled, pods must be valid under a PodSecurityPolicy to be created.

The `workload_metadata_config` block supports:

- `node_metadata` (Required) How to expose the node metadata to the workload running on the node. Accepted values are:
  - UNSPECIFIED: Not Set
  - SECURE: Prevent workloads not in hostNetwork from accessing certain VM metadata, specifically kube-env, which contains Kubelet credentials, and the instance identity token. See Metadata Concealment documentation.
  - EXPOSE: Expose all VM metadata to pods.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `endpoint` - The IP address of this cluster’s Kubernetes master.
- `instance_group_urls` - List of instance group URLs which have been assigned to the cluster.
- `maintenance_policy.0.daily_maintenance_window.0.duration` - Duration of the time window, automatically chosen to be smallest possible in the given scenario. Duration will be in RFC3339 format ”PTnHnMnS”.

- `master_auth.0.client_certificate` - Base64 encoded public certificate used by clients to authenticate to the cluster endpoint.
- `master_auth.0.client_key` - Base64 encoded private key used by clients to authenticate to the cluster endpoint.
- `master_auth.0.cluster_ca_certificate` - Base64 encoded public certificate that is the root of trust for the cluster.
- `master_version` - The current version of the master in the cluster. This may be different than the `min_master_version` set in the config if the master has been updated by GKE.

## » Timeouts

`google_container_cluster` provides the following Timeouts configuration options:

- `create` - (Default `30 minutes`) Used for clusters
- `update` - (Default `10 minutes`) Used for updates to clusters
- `delete` - (Default `10 minutes`) Used for destroying clusters.

## » Import

GKE clusters can be imported using the `zone`, and `name`, e.g.

```
$ terraform import google_container_cluster.mycluster us-east1-a/my-cluster
```

# » google_container_node_pool

Manages a Node Pool resource within GKE. For more information see the official documentation and API.

## » Example usage

### » Standard usage

```
resource "google_container_node_pool" "np" {
  name       = "my-node-pool"
  zone       = "us-central1-a"
  cluster    = "${google_container_cluster.primary.name}"
  node_count = 3
}
```

```
resource "google_container_cluster" "primary" {
  name               = "marcellus-wallace"
  zone               = "us-central1-a"
  initial_node_count = 3

  additional_zones = [
    "us-central1-b",
    "us-central1-c",
  ]

  master_auth {
    username = "mr.yoda"
    password = "adoy.rm"
  }

  node_config {
    oauth_scopes = [
      "https://www.googleapis.com/auth/compute",
      "https://www.googleapis.com/auth/devstorage.read_only",
      "https://www.googleapis.com/auth/logging.write",
      "https://www.googleapis.com/auth/monitoring",
    ]

    guest_accelerator {
      type  = "nvidia-tesla-k80"
      count = 1
    }
  }
}
```

### » Usage with an empty default pool.

```
resource "google_container_node_pool" "np" {
  name       = "my-node-pool"
  zone       = "us-central1-a"
  cluster    = "${google_container_cluster.primary.name}"
  node_count = 1

  node_config {
    preemptible  = true
    machine_type = "n1-standard-1"

    oauth_scopes = [
      "compute-rw",
      "storage-ro",
      "logging-write",
      "monitoring",
    ]
  }
}

resource "google_container_cluster" "primary" {
  name = "marcellus-wallace"
  zone = "us-central1-a"

  lifecycle {
    ignore_changes = ["node_pool"]
  }

  node_pool {
    name = "default-pool"
  }
}
```

## » Argument Reference

- `zone` - (Required) The zone in which the cluster resides.
- `cluster` - (Required) The cluster to create the node pool for. Cluster must be present in `zone` provided for this resource.

---

- `autoscaling` - (Optional) Configuration required by cluster autoscaler to adjust the size of the node pool to the current cluster usage. Structure is documented below.
- `initial_node_count` - (Optional) The initial node count for the pool. Changing this will force recreation of the resource.
- `management` - (Optional) Node management configuration, wherein auto-repair and auto-upgrade is configured. Structure is documented below.
- `name` - (Optional) The name of the node pool. If left blank, Terraform will auto-generate a unique name.
- `name_prefix` - (Optional) Creates a unique name for the node pool beginning with the specified prefix. Conflicts with `name`.
- `node_config` - (Optional) The node configuration of the pool. See google_container_cluster for schema.
- `node_count` - (Optional) The number of nodes per instance group. This field can be used to update the number of nodes per instance group but

should not be used alongside `autoscaling`.

- `project` - (Optional) The ID of the project in which to create the node pool. If blank, the provider-configured project will be used.

The `autoscaling` block supports:

- `min_node_count` - (Required) Minimum number of nodes in the NodePool. Must be >=1 and <= `max_node_count`.
- `max_node_count` - (Required) Maximum number of nodes in the NodePool. Must be >= min_node_count.

The `management` block supports:

- `auto_repair` - (Optional) Whether the nodes will be automatically repaired.
- `auto_upgrade` - (Optional) Whether the nodes will be automatically upgraded.

## » Import

Node pools can be imported using the `zone`, `cluster` and `name`, e.g.

```
$ terraform import google_container_node_pool.mainpool us-east1-a/my-cluster/main-pool
```

# » google_dataflow_job

Creates a job on Dataflow, which is an implementation of Apache Beam running on Google Compute Engine. For more information see the official documentation for Beam and Dataflow.

## » Example Usage

```
resource "google_dataflow_job" "big_data_job" {
    name = "dataflow-job"
    template_gcs_path = "gs://my-bucket/templates/template_file"
    temp_gcs_location = "gs://my-bucket/tmp_dir"
    parameters {
        foo = "bar"
        baz = "qux"
    }
}
```

## » Note on ”destroy” / ”apply”

There are many types of Dataflow jobs. Some Dataflow jobs run constantly, getting new data from (e.g.) a GCS bucket, and outputting data continuously. Some jobs process a set amount of data then terminate. All jobs can fail while running due to programming errors or other issues. In this way, Dataflow jobs are different from most other Terraform / Google resources.

The Dataflow resource is considered ’existing’ while it is in a nonterminal state. If it reaches a terminal state (e.g. ’FAILED’, ’COMPLETE’, ’CANCELLED’), it will be recreated on the next ’apply’. This is as expected for jobs which run continously, but may surprise users who use this resource for other kinds of Dataflow jobs.

A Dataflow job which is ’destroyed’ may be ”cancelled” or ”drained”. If ”cancelled”, the job terminates - any data written remains where it is, but no new data will be processed. If ”drained”, no new data will enter the pipeline, but any data currently in the pipeline will finish being processed. The default is ”cancelled”, but if a user sets `on_delete` to `"drain"` in the configuration, you may experience a long wait for your `terraform destroy` to complete.

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by Dataflow.
- `template_gcs_path` - (Required) The GCS path to the Dataflow job template.
- `temp_gcs_location` - (Required) A writeable location on GCS for the Dataflow job to dump its temporary data.

---

- `parameters` - (Optional) Key/Value pairs to be passed to the Dataflow job (as used in the template).
- `max_workers` - (Optional) The number of workers permitted to work on the job. More workers may improve processing speed at additional cost.
- `on_delete` - (Optional) One of ”drain” or ”cancel”. Specifies behavior of deletion during `terraform destroy`. See above note.
- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `zone` - (Optional) The zone in which the created job should run. If it is not provided, the provider zone is used.

## » Attributes Reference

- `state` - The current state of the resource, selected from the JobState enum

# » google_dataproc_cluster

Manages a Cloud Dataproc cluster resource within GCP. For more information see the official dataproc documentation.

**Warning:** Due to limitations of the API, all arguments except `labels`,`cluster_config.worker_config.num_i` and `cluster_config.preemptible_worker_config.num_instances` are non-updateable. Changing others will cause recreation of the whole cluster!

## » Example usage

```
resource "google_dataproc_cluster" "simplecluster" {
    name       = "simplecluster"
    region     = "us-central1"
}

resource "google_dataproc_cluster" "mycluster" {
    name       = "mycluster"
    region     = "us-central1"
    labels {
        foo = "bar"
    }

    cluster_config {
        staging_bucket        = "dataproc-staging-bucket"

        master_config {
            num_instances     = 1
            machine_type      = "n1-standard-1"
            disk_config {
                boot_disk_size_gb = 10
            }
        }

        worker_config {
            num_instances     = 2
            machine_type      = "n1-standard-1"
            disk_config {
                boot_disk_size_gb = 10
                num_local_ssds    = 1
            }
        }

        preemptible_worker_config {
            num_instances     = 0
        }

        # Override or set some custom properties
        software_config {
            image_version       = "preview"
            override_properties = {
                "dataproc:dataproc.allow.zero.workers" = "true"
            }
        }

        gce_cluster_config {
            #network = "${google_compute_network.dataproc_network.name}"
            tags    = ["foo", "bar"]
        }

        # You can define multiple initialization_action blocks
        initialization_action {
            script      = "gs://dataproc-initialization-actions/stackdriver/stackdriver.sh"
            timeout_sec = 500
        }

    }
}
```

## » Argument Reference

- `name` - (Required) The name of the cluster, unique within the project and zone.

---

- `project` - (Optional) The ID of the project in which the `cluster` will exist. If it is not provided, the provider project is used.
- `region` - (Optional) The region in which the cluster and associated nodes will be created in. Defaults to `global`.
- `labels` - (Optional, Computed) The list of labels (key/value pairs) to be applied to instances in the cluster. GCP generates some itself including `goog-dataproc-cluster-name` which is the name of the cluster.

- `cluster_config` - (Optional) Allows you to configure various aspects of the cluster. Structure defined below.

---

The **cluster_config** block supports:

```
cluster_config {
    gce_cluster_config        { ... }
    master_config             { ... }
    worker_config             { ... }
    preemptible_worker_config { ... }
    software_config           { ... }

    # You can define multiple initialization_action blocks
    initialization_action     { ... }
}
```

- `staging_bucket` - (Optional) The Cloud Storage staging bucket used to stage files, such as Hadoop jars, between client machines and the cluster. Note: If you don't explicitly specify a `staging_bucket` then GCP will auto create / assign one for you. However, you are not guaranteed an auto generated bucket which is solely dedicated to your cluster; it may be shared with other clusters in the same region/zone also choosing to use the auto generation option.
- `delete_autogen_bucket` (Optional, Deprecated) If this is set to true, upon destroying the cluster, if no explicit `staging_bucket` was specified (i.e. an auto generated bucket was relied upon) then this auto generated bucket will also be deleted as part of the cluster destroy. By default this is set to false. This value is deprecated: autogenerated buckets are shared by all clusters in the same region, so deleting the bucket could adversely harm other dataproc clusters.
- `gce_cluster_config` (Optional) Common config settings for resources of Google Compute Engine cluster instances, applicable to all instances in the cluster. Structure defined below.
- `master_config` (Optional) The Google Compute Engine config settings for the master instances in a cluster.. Structure defined below.
- `worker_config` (Optional) The Google Compute Engine config settings for the worker instances in a cluster.. Structure defined below.
- `preemptible_worker_config` (Optional) The Google Compute Engine config settings for the additional (aka preemptible) instancesin a cluster. Structure defined below.
- `software_config` (Optional) The config settings for software inside the cluster. Structure defined below.

- `initialization_action` (Optional) Commands to execute on each node after config is completed. You can specify multiple versions of these. Structure defined below.

---

The **cluster_config.gce_cluster_config** block supports:

```
cluster_config {
    gce_cluster_config {

        zone = "us-central1-a"

        # One of the below to hook into a custom network / subnetwork
        network    = "${google_compute_network.dataproc_network.name}"
        subnetwork = "${google_compute_network.dataproc_subnetwork.name}"

        tags    = ["foo", "bar"]
    }
}
```

- `zone` - (Optional, Computed) The GCP zone where your data is stored and used (i.e. where the master and the worker nodes will be created in). If `region` is set to 'global' (default) then `zone` is mandatory, otherwise GCP is able to make use of Auto Zone Placement to determine this automatically for you. Note: This setting additionally determines and restricts which computing resources are available for use with other configs such as `cluster_config.master_config.machine_type` and `cluster_config.worker_config.machine_type`.
- `network` - (Optional, Computed) The name or self_link of the Google Compute Engine network to the cluster will be part of. Conflicts with `subnetwork`. If neither is specified, this defaults to the "default" network.
- `subnetwork` - (Optional) The name or self_link of the Google Compute Engine subnetwork the cluster will be part of. Conflicts with `network`.
- `service_account` - (Optional) The service account to be used by the Node VMs. If not specified, the "default" service account is used.
- `service_account_scopes` - (Optional, Computed) The set of Google API scopes to be made available on all of the node VMs under the `service_account` specified. These can be either FQDNs, or scope aliases. The following scopes are necessary to ensure the correct functioning of the cluster:
  - `useraccounts-ro` (`https://www.googleapis.com/auth/cloud.useraccounts.readonly`)
  - `storage-rw` (`https://www.googleapis.com/auth/devstorage.read_write`)
  - `logging-write` (`https://www.googleapis.com/auth/logging.write`)

- `tags` - (Optional) The list of instance tags applied to instances in the cluster. Tags are used to identify valid sources or targets for network firewalls.
- `internal_ip_only` - (Optional) By default, clusters are not restricted to internal IP addresses, and will have ephemeral external IP addresses assigned to each instance. If set to true, all instances in the cluster will only have internal IP addresses. Note: Private Google Access (also known as `privateIpGoogleAccess`) must be enabled on the subnetwork that the cluster will be launched in.

---

The **cluster__config.master__config** block supports:

```
cluster_config {
    master_config {
        num_instances     = 1
        machine_type      = "n1-standard-1"
        disk_config {
            boot_disk_size_gb = 10
            num_local_ssds    = 1
        }
    }
}
```

- `num_instances`- (Optional, Computed) Specifies the number of master nodes to create. If not specified, GCP will default to a predetermined computed value (currently 1).
- `machine_type` - (Optional, Computed) The name of a Google Compute Engine machine type to create for the master. If not specified, GCP will default to a predetermined computed value (currently `n1-standard-4`).
- `disk_config.boot_disk_size_gb` - (Optional, Computed) Size of the primary disk attached to each node, specified in GB. The primary disk contains the boot volume and system libraries, and the smallest allowed disk size is 10GB. GCP will default to a predetermined computed value if not set (currently 500GB). Note: If SSDs are not attached, it also contains the HDFS data blocks and Hadoop working directories.
- `disk_config.num_local_ssds` - (Optional) The amount of local SSD disks that will be attached to each master cluster node. Defaults to 0.

---

The **cluster__config.worker__config** block supports:

```
cluster_config {
    worker_config {
        num_instances     = 3
        machine_type      = "n1-standard-1"
        disk_config {
            boot_disk_size_gb = 10
            num_local_ssds    = 1
        }
    }
}
```

- num_instances- (Optional, Computed) Specifies the number of worker nodes to create. If not specified, GCP will default to a predetermined computed value (currently 2). There is currently a beta feature which allows you to run a Single Node Cluster. In order to take advantage of this you need to set "dataproc:dataproc.allow.zero.workers" = "true" in cluster_config.software_config.properties
- machine_type - (Optional, Computed) The name of a Google Compute Engine machine type to create for the worker nodes. If not specified, GCP will default to a predetermined computed value (currently n1-standard-4).
- disk_config (Optional) Disk Config
  - boot_disk_size_gb - (Optional, Computed) Size of the primary disk attached to each worker node, specified in GB. The smallest allowed disk size is 10GB. GCP will default to a predetermined computed value if not set (currently 500GB). Note: If SSDs are not attached, it also contains the HDFS data blocks and Hadoop working directories.
  - num_local_ssds - (Optional) The amount of local SSD disks that will be attached to each worker cluster node. Defaults to 0.

---

The **cluster_config.preemptible_worker_config** block supports:

```
cluster_config {
    preemptible_worker_config {
        num_instances     = 1
        disk_config {
            boot_disk_size_gb = 10
        }
    }
}
```

Note: Unlike worker_config, you cannot set the machine_type value directly. This will be set for you based on whatever was set for the worker_config.machine_type value.

- num_instances- (Optional) Specifies the number of preemptible nodes to create. Defaults to 0.
- disk_config (Optional) Disk Config

  - boot_disk_size_gb - (Optional, Computed) Size of the primary disk attached to each preemptible worker node, specified in GB. The smallest allowed disk size is 10GB. GCP will default to a predetermined computed value if not set (currently 500GB). Note: If SSDs are not attached, it also contains the HDFS data blocks and Hadoop working directories.

---

The **cluster_config.software_config** block supports:

```
cluster_config {
    # Override or set some custom properties
    software_config {
        image_version       = "preview"
        override_properties = {
            "dataproc:dataproc.allow.zero.workers" = "true"
        }
    }
}
```

- image_version - (Optional, Computed) The Cloud Dataproc image version to use for the cluster - this controls the sets of software versions installed onto the nodes when you create clusters. If not specified, defaults to the latest version. For a list of valid versions see Cloud Dataproc versions
- override_properties - (Optional) A list of override and additional properties (key/value pairs) used to modify various aspects of the common configuration files used when creating a cluster. For a list of valid properties please see Cluster properties

---

The **initialization_action** block (Optional) can be specified multiple times and supports:

```
cluster_config {
    # You can define multiple initialization_action blocks
    initialization_action {
        script      = "gs://dataproc-initialization-actions/stackdriver/stackdriver.sh"
        timeout_sec = 500
    }
}
```

- script- (Required) The script to be executed during initialization of the cluster. The script must be a GCS file with a gs:// prefix.
- timeout_sec - (Optional, Computed) The maximum duration (in seconds) which script is allowed to take to execute its action. GCP will default to a predetermined computed value if not set (currently 300).

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `cluster_config.master_config.instance_names` - List of master instance names which have been assigned to the cluster.
- `cluster_config.worker_config.instance_names` - List of worker instance names which have been assigned to the cluster.
- `cluster_config.preemptible_worker_config.instance_names` - List of preemptible instance names which have been assigned to the cluster.
- `cluster_config.bucket` - The name of the cloud storage bucket ultimately used to house the staging data for the cluster. If `staging_bucket` is specified, it will contain this value, otherwise it will be the auto generated name.
- `cluster_config.software_config.properties` - A list of the properties used to set the daemon config files. This will include any values supplied by the user via `cluster_config.software_config.override_properties`

## » Timeouts

`google_dataproc_cluster` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for creating clusters.
- `update` - (Default `5 minutes`) Used for updating clusters
- `delete` - (Default `5 minutes`) Used for destroying clusters.

# » google_dataproc_job

Manages a job resource within a Dataproc cluster within GCE. For more information see the official dataproc documentation.

**Note:** This resource does not support 'update' and changing any attributes will cause the resource to be recreated.

## » Example usage

```
resource "google_dataproc_cluster" "mycluster" {
    name   = "dproc-cluster-unique-name"
    region = "us-central1"
}
```

```
# Submit an example spark job to a dataproc cluster
resource "google_dataproc_job" "spark" {
    region       = "${google_dataproc_cluster.mycluster.region}"
    force_delete = true
    placement {
        cluster_name = "${google_dataproc_cluster.mycluster.name}"
    }

    spark_config {
        main_class    = "org.apache.spark.examples.SparkPi"
        jar_file_uris = ["file:///usr/lib/spark/examples/jars/spark-examples.jar"]
        args          = ["1000"]

        properties    = {
            "spark.logConf" = "true"
        }

        logging_config {
            driver_log_levels {
                "root" = "INFO"
            }
        }
    }
}

# Submit an example pyspark job to a dataproc cluster
resource "google_dataproc_job" "pyspark" {
    region       = "${google_dataproc_cluster.mycluster.region}"
    force_delete = true
    placement {
        cluster_name = "${google_dataproc_cluster.mycluster.name}"
    }

    pyspark_config {
        main_python_file_uri = "gs://dataproc-examples-2f10d78d114f6aaec76462e3c310f31f/src/
        properties = {
            "spark.logConf" = "true"
        }
    }
}

# Check out current state of the jobs
output "spark_status" {
    value = "${google_dataproc_job.spark.status.0.state}"
}
```

```
output "pyspark_status" {
    value = "${google_dataproc_job.pyspark.status.0.state}"
}
```

## » Argument Reference

- `placement.cluster_name` - (Required) The name of the cluster where the job will be submitted.
- `xxx_config` - (Required) Exactly one of the specific job types to run on the cluster should be specified. If you want to submit multiple jobs, this will currently require the definition of multiple `google_dataproc_job` resources as shown in the example above, or by setting the `count` attribute. The following job configs are supported:
  - pyspark_config - Submits a PySpark job to the cluster
  - spark_config - Submits a Spark job to the cluster
  - hadoop_config - Submits a Hadoop job to the cluster
  - hive_config - Submits a Hive job to the cluster
  - hpig_config - Submits a Pig job to the cluster
  - sparksql_config - Submits a Spark SQL job to the cluster

---

- `project` - (Optional) The project in which the `cluster` can be found and jobs subsequently run against. If it is not provided, the provider project is used.
- `region` - (Optional) The Cloud Dataproc region. This essentially determines which clusters are available for this job to be submitted to. If not specified, defaults to `global`.
- `force_delete` - (Optional) By default, you can only delete inactive jobs within Dataproc. Setting this to true, and calling destroy, will ensure that the job is first cancelled before issuing the delete.
- `labels` - (Optional) The list of labels (key/value pairs) to add to the job.
- `scheduling.max_failures_per_hour` - (Optional) Maximum number of times per hour a driver may be restarted as a result of driver terminating with non-zero code before job is reported failed.

The `pyspark_config` block supports:

Submitting a pyspark job to the cluster. Below is an example configuration:

```
# Submit a pyspark job to the cluster
resource "google_dataproc_job" "pyspark" {
    ...
    pyspark_config {
        main_python_file_uri = "gs://dataproc-examples-2f10d78d114f6aaec76462e3c310f31f/src/
        properties = {
            "spark.logConf" = "true"
        }
    }
}
```

For configurations requiring Hadoop Compatible File System (HCFS) references, the options below are generally applicable:

```
  - GCS files with the `gs://` prefix
  - HDFS files on the cluster with the `hdfs://` prefix
  - Local files on the cluster with the `file://` prefix
```

- main_python_file_uri- (Required) The HCFS URI of the main Python file to use as the driver. Must be a .py file.
- args - (Optional) The arguments to pass to the driver.
- python_file_uris - (Optional) HCFS file URIs of Python files to pass to the PySpark framework. Supported file types: .py, .egg, and .zip.
- jar_file_uris - (Optional) HCFS URIs of jar files to add to the CLASSPATHs of the Python driver and tasks.
- file_uris - (Optional) HCFS URIs of files to be copied to the working directory of Python drivers and distributed tasks. Useful for naively parallel tasks.
- archive_uris - (Optional) HCFS URIs of archives to be extracted in the working directory of .jar, .tar, .tar.gz, .tgz, and .zip.
- properties - (Optional) A mapping of property names to values, used to configure PySpark. Properties that conflict with values set by the Cloud Dataproc API may be overwritten. Can include properties set in /etc/spark/conf/spark-defaults.conf and classes in user code.
- logging_config.driver_log_levels- (Optional) The per-package log levels for the driver. This may include 'root' package name to configure rootLogger. Examples: 'com.google = FATAL', 'root = INFO', 'org.apache = DEBUG'

The spark_config block supports:

```
# Submit a spark job to the cluster
resource "google_dataproc_job" "spark" {
    ...

    spark_config {
        main_class    = "org.apache.spark.examples.SparkPi"
        jar_file_uris = ["file:///usr/lib/spark/examples/jars/spark-examples.jar"]
        args          = ["1000"]

        properties    = {
            "spark.logConf" = "true"
        }

        logging_config {
            driver_log_levels {
                "root" = "INFO"
            }
        }
    }
}
```

- `main_class`- (Optional) The class containing the main method of the driver. Must be in a provided jar or jar that is already on the classpath. Conflicts with `main_jar_file_uri`
- `main_jar_file_uri` - (Optional) The HCFS URI of jar file containing the driver jar. Conflicts with `main_class`
- `args` - (Optional) The arguments to pass to the driver.
- `jar_file_uris` - (Optional) HCFS URIs of jar files to add to the CLASSPATHs of the Spark driver and tasks.
- `file_uris` - (Optional) HCFS URIs of files to be copied to the working directory of Spark drivers and distributed tasks. Useful for naively parallel tasks.
- `archive_uris` - (Optional) HCFS URIs of archives to be extracted in the working directory of .jar, .tar, .tar.gz, .tgz, and .zip.
- `properties` - (Optional) A mapping of property names to values, used to configure Spark. Properties that conflict with values set by the Cloud Dataproc API may be overwritten. Can include properties set in `/etc/spark/conf/spark-defaults.conf` and classes in user code.
- `logging_config.driver_log_levels`- (Optional) The per-package log levels for the driver. This may include 'root' package name to configure rootLogger. Examples: 'com.google = FATAL', 'root = INFO', 'org.apache = DEBUG'

The `hadoop_config` block supports:

```
# Submit a hadoop job to the cluster
resource "google_dataproc_job" "hadoop" {
    ...

    hadoop_config {
        main_jar_file_uri =  "file:///usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
        args              = [
            "wordcount",
            "file:///usr/lib/spark/NOTICE",
            "gs://${google_dataproc_cluster.basic.cluster_config.0.bucket}/hadoopjob_output"
        ]
    }
}
```

- `main_class`- (Optional) The name of the driver's main class. The jar file containing the class must be in the default CLASSPATH or specified in `jar_file_uris`. Conflicts with `main_jar_file_uri`
- `main_jar_file_uri` - (Optional) The HCFS URI of the jar file containing the main class. Examples: 'gs://foo-bucket/analytics-binaries/extract-useful-metrics-mr.jar' 'hdfs:/tmp/test-samples/custom-wordcount.jar' 'file:///home/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar'. Conflicts with `main_class`
- `args` - (Optional) The arguments to pass to the driver. Do not include arguments, such as -libjars or -Dfoo=bar, that can be set as job properties, since a collision may occur that causes an incorrect job submission.
- `jar_file_uris` - (Optional) HCFS URIs of jar files to add to the CLASSPATHs of the Spark driver and tasks.
- `file_uris` - (Optional) HCFS URIs of files to be copied to the working directory of Hadoop drivers and distributed tasks. Useful for naively parallel tasks.
- `archive_uris` - (Optional) HCFS URIs of archives to be extracted in the working directory of .jar, .tar, .tar.gz, .tgz, and .zip.
- `properties` - (Optional) A mapping of property names to values, used to configure Hadoop. Properties that conflict with values set by the Cloud Dataproc API may be overwritten. Can include properties set in `/etc/hadoop/conf/*-site` and classes in user code..
- `logging_config.driver_log_levels`- (Optional) The per-package log levels for the driver. This may include 'root' package name to configure rootLogger. Examples: 'com.google = FATAL', 'root = INFO', 'org.apache = DEBUG'

The `hive_config` block supports:

```
# Submit a hive job to the cluster
resource "google_dataproc_job" "hive" {
    ...

    hive_config {
        query_list = [
            "DROP TABLE IF EXISTS dprocjob_test",
            "CREATE EXTERNAL TABLE dprocjob_test(bar int) LOCATION 'gs://${google_dataproc_c
            "SELECT * FROM dprocjob_test WHERE bar > 2",
        ]
    }
}
```

- `query_list`- (Optional) The list of Hive queries or statements to execute as part of the job. Conflicts with `query_file_uri`
- `query_file_uri` - (Optional) HCFS URI of file containing Hive script to execute as the job. Conflicts with `query_list`
- `continue_on_failure -` (Optional) Whether to continue executing queries if a query fails. The default value is false. Setting to true can be useful when executing independent parallel queries. Defaults to false.
- `script_variables` - (Optional) Mapping of query variable names to values (equivalent to the Hive command: `SET name="value";`).
- `properties -` (Optional) A mapping of property names and values, used to configure Hive. Properties that conflict with values set by the Cloud Dataproc API may be overwritten. Can include properties set in `/etc/hadoop/conf/*-site.xml`, `/etc/hive/conf/hive-site.xml`, and classes in user code..
- `jar_file_uris` - (Optional) HCFS URIs of jar files to add to the CLASSPATH of the Hive server and Hadoop MapReduce (MR) tasks. Can contain Hive SerDes and UDFs.

The `pig_config` block supports:

```
# Submit a pig job to the cluster
resource "google_dataproc_job" "pig" {
    ...

    pig_config {
        query_list = [
            "LNS = LOAD 'file:///usr/lib/pig/LICENSE.txt ' AS (line)",
            "WORDS = FOREACH LNS GENERATE FLATTEN(TOKENIZE(line)) AS word",
            "GROUPS = GROUP WORDS BY word",
            "WORD_COUNTS = FOREACH GROUPS GENERATE group, COUNT(WORDS)",
            "DUMP WORD_COUNTS"
        ]
    }
}
```

- `query_list`- (Optional) The list of Hive queries or statements to execute as part of the job. Conflicts with `query_file_uri`

- query_file_uri - (Optional) HCFS URI of file containing Hive script to execute as the job. Conflicts with query_list
- continue_on_failure - (Optional) Whether to continue executing queries if a query fails. The default value is false. Setting to true can be useful when executing independent parallel queries. Defaults to false.
- script_variables - (Optional) Mapping of query variable names to values (equivalent to the Pig command: name=[value]).
- properties - (Optional) A mapping of property names to values, used to configure Pig. Properties that conflict with values set by the Cloud Dataproc API may be overwritten. Can include properties set in /etc/hadoop/conf/*-site.xml, /etc/pig/conf/pig.properties, and classes in user code.
- jar_file_uris - (Optional) HCFS URIs of jar files to add to the CLASSPATH of the Pig Client and Hadoop MapReduce (MR) tasks. Can contain Pig UDFs.
- logging_config.driver_log_levels- (Optional) The per-package log levels for the driver. This may include 'root' package name to configure rootLogger. Examples: 'com.google = FATAL', 'root = INFO', 'org.apache = DEBUG'

The sparksql_config block supports:

```
# Submit a spark SQL job to the cluster
resource "google_dataproc_job" "sparksql" {
    ...

    sparksql_config {
        query_list = [
            "DROP TABLE IF EXISTS dprocjob_test",
            "CREATE TABLE dprocjob_test(bar int)",
            "SELECT * FROM dprocjob_test WHERE bar > 2",
        ]
    }
}
```

- query_list- (Optional) The list of SQL queries or statements to execute as part of the job. Conflicts with query_file_uri
- query_file_uri - (Optional) The HCFS URI of the script that contains SQL queries. Conflicts with query_list
- script_variables - (Optional) Mapping of query variable names to values (equivalent to the Spark SQL command: SET name="value";).
- properties - (Optional) A mapping of property names to values, used to configure Spark SQL's SparkConf. Properties that conflict with values set

by the Cloud Dataproc API may be overwritten.

- `jar_file_uris` - (Optional) HCFS URIs of jar files to be added to the Spark CLASSPATH.
- `logging_config.driver_log_levels`- (Optional) The per-package log levels for the driver. This may include 'root' package name to configure rootLogger. Examples: 'com.google = FATAL', 'root = INFO', 'org.apache = DEBUG'

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `reference.0.cluster_uuid` - A cluster UUID generated by the Cloud Dataproc service when the job is submitted.
- `status.0.state` - A state message specifying the overall job state.
- `status.0.details` - Optional job state details, such as an error description if the state is ERROR.
- `status.0.state_start_time` - The time when this state was entered.
- `status.0.substate` - Additional state information, which includes status reported by the agent.
- `driver_output_resource_uri` - A URI pointing to the location of the stdout of the job's driver program.
- `driver_controls_files_uri` - If present, the location of miscellaneous control files which may be used as part of job setup and handling. If not present, control files may be placed in the same location as driver_output_uri.

## » Timeouts

`google_dataproc_cluster` provides the following Timeouts configuration options:

- `create` - (Default `10 minutes`) Used for submitting a job to a dataproc cluster.
- `delete` - (Default `10 minutes`) Used for deleting a job from a dataproc cluster.

# » google_dns_managed_zone

Manages a zone within Google Cloud DNS. For more information see the official documentation and API.

## » Example Usage

```
resource "google_dns_managed_zone" "prod" {
  name        = "prod-zone"
  dns_name    = "prod.mydomain.com."
  description = "Production DNS zone"
}
```

## » Argument Reference

The following arguments are supported:

- `dns_name` - (Required) The DNS name of this zone, e.g. ”terraform.io”.
- `name` - (Required) A unique name for the resource, required by GCE. Changing this forces a new resource to be created.

---

- `description` - (Optional) A textual description field. Defaults to ’Managed by Terraform’.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `name_servers` - The list of nameservers that will be authoritative for this domain. Use NS records to redirect from your DNS provider to these names, thus making Google Cloud DNS authoritative for this zone.

## » Import

DNS managed zones can be imported using the `name`, e.g.

```
$ terraform import google_dns_managed_zone.prod prod-zone
```

# » google_dns_record_set

Manages a set of DNS records within Google Cloud DNS. For more information see the official documentation and API.

**Note:** The Google Cloud DNS API requires NS records be present at all times. To accommodate this, when creating NS records, the default records Google automatically creates will be silently overwritten. Also, when destroying NS records, Terraform will not actually remove NS records, but will report that it did.

## » Example Usage

### » Binding a DNS name to the ephemeral IP of a new instance:

```
resource "google_dns_record_set" "frontend" {
  name = "frontend.${google_dns_managed_zone.prod.dns_name}"
  type = "A"
  ttl  = 300

  managed_zone = "${google_dns_managed_zone.prod.name}"

  rrdatas = ["${google_compute_instance.frontend.network_interface.0.access_config.0.assigne
}

resource "google_compute_instance" "frontend" {
  name         = "frontend"
  machine_type = "g1-small"
  zone         = "us-central1-b"

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-8"
    }
  }

  network_interface {
    network       = "default"
    access_config = {}
  }
}

resource "google_dns_managed_zone" "prod" {
  name     = "prod-zone"
  dns_name = "prod.mydomain.com."
}
```

### » Adding a SPF record

`\"` must be added around your `rrdatas` for a SPF record. Otherwise `rrdatas` string gets split on spaces.

```
resource "google_dns_record_set" "spf" {
  name = "frontend.${google_dns_managed_zone.prod.dns_name}"
  managed_zone = "${google_dns_managed_zone.prod.name}"
  type = "TXT"
  ttl  = 300

  rrdatas = ["\"v=spf1 ip4:111.111.111.111 include:backoff.email-example.com -all\""]
}

resource "google_dns_managed_zone" "prod" {
  name     = "prod-zone"
  dns_name = "prod.mydomain.com."
}
```

## » Argument Reference

The following arguments are supported:

- `managed_zone` - (Required) The name of the zone in which this record set will reside.
- `name` - (Required) The DNS name this record set will apply to.
- `rrdatas` - (Required) The string data for the records in this record set whose meaning depends on the DNS type. For TXT record, if the string data contains spaces, add surrounding `\"` if you don't want your string to get split on spaces.
- `ttl` - (Required) The time-to-live of this record set (seconds).
- `type` - (Required) The DNS record set type.

---

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

Only the arguments listed above are exposed as attributes.

## » Import

DNS record set can be imported using the `zone name`, `record name` and record `type`, e.g.

```
$ terraform import google_dns_record_set.frontend prod-zone/frontend.prod.mydomain.com./A
```

Note: The record name must include the trailing dot at the end.

# » google_endpoints_service

This resource creates and rolls out a Cloud Endpoints service using OpenAPI or gRPC. View the relevant docs for OpenAPI and gRPC.

## » Example Usage

```
resource "google_endpoints_service" "openapi_service" {
  service_name = "api-name.endpoints.project-id.cloud.goog"
  project = "project-id"
  openapi_config = "${file("openapi_spec.yml")}"
}

resource "google_endpoints_service" "grpc_service" {
  service_name = "api-name.endpoints.project-id.cloud.goog"
  project = "project-id"
  grpc_config = "${file("service_spec.yml")}"
  protoc_output = "${file("compiled_descriptor_file.pb")}"
}
```

The example in `examples/endpoints_on_compute_engine` shows the API from the quickstart running on a Compute Engine VM and reachable through Cloud Endpoints, which may also be useful.

## » Argument Reference

The following arguments are supported: * `service_name`: (Required) The name of the service. Usually of the form `$apiname.endpoints.$projectid.cloud.goog`. * `openapi_config`: (Optional) The full text of the OpenAPI YAML configuration as described here. Either this, or *both* of `grpc_config` and `protoc_output` must be specified. * `grpc_config`: (Optional) The full text of the Service Config YAML file (Example located here). If provided, must also provide `protoc_output`. `open_api` config must *not* be provided. * `protoc_output`: (Optional) The full contents of the Service Descriptor File generated by protoc.

This should be a compiled .pb file. * `project`: (Optional) The project ID that the service belongs to. If not provided, provider project is used.

## » Attributes Reference

In addition to the arguments, the following attributes are available: * `config_id`: The autogenerated ID for the configuration that is rolled out as part of the creation of this resource. Must be provided to compute engine instances as a tag. * `dns_address`: The address at which the service can be found - usually the same as the service name. * `apis`: A list of API objects; structure is documented below. * `endpoints`: A list of Endpoint objects; structure is documented below.

---

### » API Object Structure

- `name`: The FQDN of the API as described in the provided config.
- `syntax`: `SYNTAX_PROTO2` or `SYNTAX_PROTO3`.
- `version`: A version string for this api. If specified, will have the form major-version.minor-version, e.g. `1.10`.
- `methods`: A list of Method objects; structure is documented below.

### » Method Object Structure

- `name`: The simple name of this method as described in the provided config.
- `syntax`: `SYNTAX_PROTO2` or `SYNTAX_PROTO3`.
- `request_type`: The type URL for the request to this API.
- `response_type`: The type URL for the response from this API.

### » Endpoint Object Structure

- `name`: The simple name of the endpoint as described in the config.
- `address`: The FQDN of the endpoint as described in the config.

# » google_pubsub_topic

Creates a topic in Google's pubsub queueing system. For more information see the official documentation and API.

### » Example Usage

```
resource "google_pubsub_topic" "mytopic" {
  name = "default-topic"
}
```

### » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the pubsub topic. Changing this forces a new resource to be created.

---

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

### » Attributes Reference

Only the arguments listed above are exposed as attributes.

### » Import

Pubsub topics can be imported using the `name` or full topic id, e.g.

```
$ terraform import google_pubsub_topic.mytopic default-topic
```

```
$ terraform import google_pubsub_topic.mytopic projects/my-gcp-project/topics/default-topic
```

When importing using only the name, the provider project must be set.

## » IAM policy for Pubsub Topic

Three different resources help you manage your IAM policy for pubsub topic. Each of these resources serves a different use case:

- `google_pubsub_topic_iam_policy`: Authoritative. Sets the IAM policy for the topic and replaces any existing policy already attached.
- `google_pubsub_topic_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the topic are preserved.
- `google_pubsub_topic_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the topic are preserved.

**Note:** google_pubsub_topic_iam_policy **cannot** be used in conjunction with google_pubsub_topic_iam_binding and google_pubsub_topic_iam_member or they will fight over what your policy should be.

**Note:** google_pubsub_topic_iam_binding resources **can be** used in conjunction with google_pubsub_topic_iam_member resources **only if** they do not grant privilege to the same role.

## » google_pubsub_topic_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role    = "roles/editor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_pubsub_topic_iam_policy" "editor" {
  topic       = "your-topic-name"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_pubsub_topic_iam_binding

```
resource "google_pubsub_topic_iam_binding" "editor" {
  topic   = "your-topic-name"
  role    = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

## » google_pubsub_topic_iam_member

```
resource "google_pubsub_topic_iam_member" "editor" {
  topic  = "your-topic-name"
  role   = "roles/editor"
  member = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `topic` - (Required) The topic name or id to bind to attach IAM policy to.
- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_pubsub_topic_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `policy_data` - (Required only by `google_pubsub_topic_iam_policy`) The policy data generated by a `google_iam_policy` data source.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the topic's IAM policy.

## » Import

Pubsub topic IAM resources can be imported using the project, topic name, role and member.

```
$ terraform import google_pubsub_topic_iam_policy.editor projects/{your-project-id}/topics/
$ terraform import google_pubsub_topic_iam_binding.editor "projects/{your-project-id}/topics

$ terraform import google_pubsub_topic_iam_member.editor "projects/{your-project-id}/topics/
```

# » IAM policy for Pubsub Topic

Three different resources help you manage your IAM policy for pubsub topic. Each of these resources serves a different use case:

- `google_pubsub_topic_iam_policy`: Authoritative. Sets the IAM policy for the topic and replaces any existing policy already attached.
- `google_pubsub_topic_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the topic are preserved.
- `google_pubsub_topic_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the topic are preserved.

**Note:** `google_pubsub_topic_iam_policy` **cannot** be used in conjunction with `google_pubsub_topic_iam_binding` and `google_pubsub_topic_iam_member` or they will fight over what your policy should be.

**Note:** `google_pubsub_topic_iam_binding` resources **can be** used in conjunction with `google_pubsub_topic_iam_member` resources **only if** they do not grant privilege to the same role.

## » google_pubsub_topic_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role    = "roles/editor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_pubsub_topic_iam_policy" "editor" {
  topic       = "your-topic-name"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_pubsub_topic_iam_binding

```
resource "google_pubsub_topic_iam_binding" "editor" {
  topic   = "your-topic-name"
  role    = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

## » google_pubsub_topic_iam_member

```
resource "google_pubsub_topic_iam_member" "editor" {
  topic  = "your-topic-name"
  role   = "roles/editor"
  member = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `topic` - (Required) The topic name or id to bind to attach IAM policy to.
- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.

- `role` - (Required) The role that should be applied. Only one `google_pubsub_topic_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `policy_data` - (Required only by `google_pubsub_topic_iam_policy`) The policy data generated by a `google_iam_policy` data source.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the topic's IAM policy.

## » Import

Pubsub topic IAM resources can be imported using the project, topic name, role and member.

```
$ terraform import google_pubsub_topic_iam_policy.editor projects/{your-project-id}/topics/

$ terraform import google_pubsub_topic_iam_binding.editor "projects/{your-project-id}/topics

$ terraform import google_pubsub_topic_iam_member.editor "projects/{your-project-id}/topics/
```

# » IAM policy for Pubsub Topic

Three different resources help you manage your IAM policy for pubsub topic. Each of these resources serves a different use case:

- `google_pubsub_topic_iam_policy`: Authoritative. Sets the IAM policy for the topic and replaces any existing policy already attached.
- `google_pubsub_topic_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the topic are preserved.
- `google_pubsub_topic_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the topic are preserved.

**Note:** `google_pubsub_topic_iam_policy` **cannot** be used in conjunction with `google_pubsub_topic_iam_binding` and `google_pubsub_topic_iam_member` or they will fight over what your policy should be.

**Note:** `google_pubsub_topic_iam_binding` resources **can be** used in conjunction with `google_pubsub_topic_iam_member` resources **only if** they do not grant privilege to the same role.

## » google_pubsub_topic_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role    = "roles/editor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_pubsub_topic_iam_policy" "editor" {
  topic       = "your-topic-name"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_pubsub_topic_iam_binding

```
resource "google_pubsub_topic_iam_binding" "editor" {
  topic   = "your-topic-name"
  role    = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

## » google_pubsub_topic_iam_member

```
resource "google_pubsub_topic_iam_member" "editor" {
  topic  = "your-topic-name"
  role   = "roles/editor"
  member = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `topic` - (Required) The topic name or id to bind to attach IAM policy to.

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_pubsub_topic_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `policy_data` - (Required only by `google_pubsub_topic_iam_policy`) The policy data generated by a `google_iam_policy` data source.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the topic's IAM policy.

## » Import

Pubsub topic IAM resources can be imported using the project, topic name, role and member.

```
$ terraform import google_pubsub_topic_iam_policy.editor projects/{your-project-id}/topics/{

$ terraform import google_pubsub_topic_iam_binding.editor "projects/{your-project-id}/topics

$ terraform import google_pubsub_topic_iam_member.editor "projects/{your-project-id}/topics/
```

# » google_pubsub_subscription

Creates a subscription in Google's pubsub queueing system. For more information see the official documentation and API.

## » Example Usage

```
resource "google_pubsub_topic" "default-topic" {
  name = "default-topic"
}

resource "google_pubsub_subscription" "default" {
  name  = "default-subscription"
  topic = "${google_pubsub_topic.default-topic.name}"

  ack_deadline_seconds = 20

  push_config {
    push_endpoint = "https://example.com/push"

    attributes {
      x-goog-version = "v1"
    }
  }
}
```

If the subscription has a topic in a different project:

```
resource "google_pubsub_topic" "topic-different-project" {
  project = "another-project"
  name = "topic-different-project"
}

resource "google_pubsub_subscription" "default" {
  name  = "default-subscription"
  topic = "${google_pubsub_topic.topic-different-project.id}"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by pubsub. Changing this forces a new resource to be created.

- `topic` - (Required) The topic name or id to bind this subscription to, required by pubsub. Changing this forces a new resource to be created.

---

- `ack_deadline_seconds` - (Optional) The maximum number of seconds a subscriber has to acknowledge a received message, otherwise the message is redelivered. Changing this forces a new resource to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `push_config` - (Optional) Block configuration for push options. More configuration options are detailed below.

The optional `push_config` block supports:

- `push_endpoint` - (Required) The URL of the endpoint to which messages should be pushed. Changing this forces a new resource to be created.
- `attributes` - (Optional) Key-value pairs of API supported attributes used to control aspects of the message delivery. Currently, only `x-goog-version` is supported, which controls the format of the data delivery. For more information, read the API docs here. Changing this forces a new resource to be created.

## » Attributes Reference

- `path` - Path of the subscription in the format `projects/{project}/subscriptions/{sub}`

## » Import

Pubsub subscription can be imported using the `name`, e.g.

```
$ terraform import google_pubsub_subscription.default default-subscription
```

# » IAM policy for Pubsub Subscription

Three different resources help you manage your IAM policy for pubsub subscription. Each of these resources serves a different use case:

- `google_pubsub_subscription_iam_policy`: Authoritative. Sets the IAM policy for the subscription and replaces any existing policy already attached.
- `google_pubsub_subscription_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the subscription are preserved.

- `google_pubsub_subscription_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the subscription are preserved.

**Note:** `google_pubsub_subscription_iam_policy` **cannot** be used in conjunction with `google_pubsub_subscription_iam_binding` and `google_pubsub_subscription_iam_member` or they will fight over what your policy should be.

**Note:** `google_pubsub_subscription_iam_binding` resources **can be** used in conjunction with `google_pubsub_subscription_iam_member` resources **only if** they do not grant privilege to the same role.

## » google_pubsub_subscription_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role    = "roles/editor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_pubsub_subscription_iam_policy" "editor" {
  subscription = "your-subscription-name"
  policy_data  = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_pubsub_subscription_iam_binding

```
resource "google_pubsub_subscription_iam_binding" "editor" {
  subscription = "your-subscription-name"
  role         = "roles/editor"
  members      = [
    "user:jane@example.com",
  ]
}
```

## » google_pubsub_subscription_iam_member

```
resource "google_pubsub_subscription_iam_member" "editor" {
  subscription = "your-subscription-name"
  role         = "roles/editor"
  member       = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `subscription` - (Required) The subscription name or id to bind to attach IAM policy to.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_pubsub_subscription_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name`
- `policy_data` - (Required only by `google_pubsub_subscription_iam_policy`) The policy data generated by a `google_iam_policy` data source.

---

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the subscription's IAM policy.

### » Import

Pubsub subscription IAM resources can be imported using the project, subscription name, role and member.

```
$ terraform import google_pubsub_subscription_iam_policy.editor projects/{your-project-id}/s

$ terraform import google_pubsub_subscription_iam_binding.editor "projects/{your-project-id]

$ terraform import google_pubsub_subscription_iam_member.editor "projects/{your-project-id}/
```

## » IAM policy for Pubsub Subscription

Three different resources help you manage your IAM policy for pubsub subscription. Each of these resources serves a different use case:

- `google_pubsub_subscription_iam_policy`: Authoritative. Sets the IAM policy for the subscription and replaces any existing policy already attached.
- `google_pubsub_subscription_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the subscription are preserved.
- `google_pubsub_subscription_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the subscription are preserved.

**Note:** `google_pubsub_subscription_iam_policy` **cannot** be used in conjunction with `google_pubsub_subscription_iam_binding` and `google_pubsub_subscription_iam_member` or they will fight over what your policy should be.

**Note:** `google_pubsub_subscription_iam_binding` resources **can be** used in conjunction with `google_pubsub_subscription_iam_member` resources **only if** they do not grant privilege to the same role.

### » google_pubsub_subscription_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role    = "roles/editor"
    members = [
      "user:jane@example.com",
    ]
  }
}
```

```
resource "google_pubsub_subscription_iam_policy" "editor" {
  subscription = "your-subscription-name"
  policy_data  = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_pubsub_subscription_iam_binding

```
resource "google_pubsub_subscription_iam_binding" "editor" {
  subscription = "your-subscription-name"
  role         = "roles/editor"
  members      = [
    "user:jane@example.com",
  ]
}
```

## » google_pubsub_subscription_iam_member

```
resource "google_pubsub_subscription_iam_member" "editor" {
  subscription = "your-subscription-name"
  role         = "roles/editor"
  member       = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `subscription` - (Required) The subscription name or id to bind to attach IAM policy to.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.

  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.

- `role` - (Required) The role that should be applied. Only one `google_pubsub_subscription_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-nam`
- `policy_data` - (Required only by `google_pubsub_subscription_iam_policy`) The policy data generated by a `google_iam_policy` data source.

---

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the subscription's IAM policy.

## » Import

Pubsub subscription IAM resources can be imported using the project, subscription name, role and member.

```
$ terraform import google_pubsub_subscription_iam_policy.editor projects/{your-project-id}/s

$ terraform import google_pubsub_subscription_iam_binding.editor "projects/{your-project-id}

$ terraform import google_pubsub_subscription_iam_member.editor "projects/{your-project-id}/
```

# » IAM policy for Pubsub Subscription

Three different resources help you manage your IAM policy for pubsub subscription. Each of these resources serves a different use case:

- `google_pubsub_subscription_iam_policy`: Authoritative. Sets the IAM policy for the subscription and replaces any existing policy already attached.
- `google_pubsub_subscription_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the subscription are preserved.

- `google_pubsub_subscription_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the subscription are preserved.

**Note:** `google_pubsub_subscription_iam_policy` **cannot** be used in conjunction with `google_pubsub_subscription_iam_binding` and `google_pubsub_subscription_iam_member` or they will fight over what your policy should be.

**Note:** `google_pubsub_subscription_iam_binding` resources **can be** used in conjunction with `google_pubsub_subscription_iam_member` resources **only if** they do not grant privilege to the same role.

## » google_pubsub_subscription_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role    = "roles/editor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_pubsub_subscription_iam_policy" "editor" {
  subscription = "your-subscription-name"
  policy_data  = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_pubsub_subscription_iam_binding

```
resource "google_pubsub_subscription_iam_binding" "editor" {
  subscription = "your-subscription-name"
  role         = "roles/editor"
  members      = [
    "user:jane@example.com",
  ]
}
```

## » google_pubsub_subscription_iam_member

```
resource "google_pubsub_subscription_iam_member" "editor" {
  subscription = "your-subscription-name"
  role         = "roles/editor"
  member       = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `subscription` - (Required) The subscription name or id to bind to attach IAM policy to.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_pubsub_subscription_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-nam`
- `policy_data` - (Required only by `google_pubsub_subscription_iam_policy`) The policy data generated by a `google_iam_policy` data source.

---

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the subscription's IAM policy.

## » Import

Pubsub subscription IAM resources can be imported using the project, subscription name, role and member.

```
$ terraform import google_pubsub_subscription_iam_policy.editor projects/{your-project-id}/s

$ terraform import google_pubsub_subscription_iam_binding.editor "projects/{your-project-id]

$ terraform import google_pubsub_subscription_iam_member.editor "projects/{your-project-id}/
```

# » google_runtimeconfig_config

Manages a RuntimeConfig resource in Google Cloud. For more information, see the official documentation, or the JSON API.

## » Example Usage

Example creating a RuntimeConfig resource.

```
resource "google_runtimeconfig_config" "my-runtime-config" {
    name = "my-service-runtime-config"
    description = "Runtime configuration values for my service"
}
```

## » Argument Reference

The following arguments are supported:

- name - (Required) The name of the runtime config.

---

- project - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- description - (Optional) The description to associate with the runtime config.

# » google_runtimeconfig_variable

Manages a RuntimeConfig variable in Google Cloud. For more information, see the official documentation, or the JSON API.

## » Example Usage

Example creating a RuntimeConfig variable.

```
resource "google_runtimeconfig_config" "my-runtime-config" {
    name = "my-service-runtime-config"
    description = "Runtime configuration values for my service"
}

resource "google_runtimeconfig_variable" "environment" {
    parent = "${google_runtimeconfig_config.my-runtime-config.name}"
    name = "prod-variables/hostname"
    text = "example.com"
}
```

You can also encode binary content using the `value` argument instead. The value must be base64 encoded.

Example of using the `value` argument.

```
resource "google_runtimeconfig_config" "my-runtime-config" {
    name = "my-service-runtime-config"
    description = "Runtime configuration values for my service"
}

resource "google_runtimeconfig_variable" "my-secret" {
    parent = "${google_runtimeconfig_config.my-runtime-config.name}"
    name = "secret"
    value = "${base64encode(file("my-encrypted-secret.dat"))}"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the variable to manage. Note that variable names can be hierarchical using slashes (e.g. ”prod-variables/hostname”).
- `parent` - (Required) The name of the RuntimeConfig resource containing this variable.
- `text` or `value` - (Required) The content to associate with the variable. Exactly one of `text` or `variable` must be specified. If `text` is specified, it must be a valid UTF-8 string and less than 4096 bytes in length. If `value` is specified, it must be base64 encoded and less than 4096 bytes in length.

---

- project - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- update_time - (Computed) The timestamp in RFC3339 UTC ”Zulu” format, accurate to nanoseconds, representing when the variable was last updated. Example: ”2016-10-09T12:33:37.578138407Z”.

# » google_sourcerepo_repository

For more information, see the official documentation and API

## » Example Usage

This example is the common case of creating a repository within Google Cloud Source Repositories:

```
resource "google_sourcerepo_repository" "frontend" {
  name = "frontend"
}
```

## » Argument Reference

The following arguments are supported:

- name - (Required) The name of the repository that will be created.

---

- project - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

The following attributes are exported:

- size - The size of the repository.
- url - The url to clone the repository.

## » google_spanner_instance

Creates and manages a Google Spanner Instance. For more information, see the official documentation, or the JSON API.

### » Example Usage

Example creating a Spanner instance.

```
resource "google_spanner_instance" "main" {
  config       = "regional-europe-west1"
  display_name = "main-instance"
  name         = "main-instance"
  num_nodes    = 1
}
```

### » Argument Reference

The following arguments are supported:

- `config` - (Required) The name of the instance's configuration (similar but not quite the same as a region) which defines defines the geographic placement and replication of your databases in this instance. It determines where your data is stored. Values are typically of the form `regional-europe-west1` , `us-central` etc. In order to obtain a valid list please consult the Configuration section of the docs.
- `display_name` - (Required) The descriptive name for this instance as it appears in UIs. Can be updated, however should be kept globally unique to avoid confusion.

---

- `name` - (Optional, Computed) The unique name (ID) of the instance. If the name is left blank, Terraform will randomly generate one when the instance is first created.
- `num_nodes` - (Optional, Computed) The number of nodes allocated to this instance. Defaults to `1`. This can be updated after creation.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `labels` - (Optional) A mapping (key/value pairs) of labels to assign to the instance.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `state` - The current state of the instance.

## » Import

Instances can be imported using their `name` and optionally the `project` in which it is defined (Often used when the project is different to that defined in the provider), The format is thus either `{instanceId}` or `{projectId}/{instanceId}`. e.g.

```
$ terraform import google_spanner_instance.master instance123

$ terraform import google_spanner_instance.master project123/instance456
```

# » google_spanner_instance

Creates a Google Spanner Database within a Spanner Instance. For more information, see the official documentation, or the JSON API.

## » Example Usage

Example creating a Spanner database.

```
resource "google_spanner_instance" "main" {
  config       = "regional-europe-west1"
  display_name = "main-instance"
}

resource "google_spanner_database" "db" {
  instance  = "${google_spanner_instance.main.name}"
  name      = "main-instance"
  ddl       =  [
    "CREATE TABLE t1 (t1 INT64 NOT NULL,) PRIMARY KEY(t1)",
    "CREATE TABLE t2 (t2 INT64 NOT NULL,) PRIMARY KEY(t2)"
  ]
}
```

## » Argument Reference

The following arguments are supported:

- `instance` - (Required) The name of the instance that will serve the new database.
- `name` - (Required) The name of the database.

---

- `project` - (Optional) The ID of the project in which to look for the `instance` specified. If it is not provided, the provider project is used.
- `ddl` - (Optional) An optional list of DDL statements to run inside the newly created database. Statements can create tables, indexes, etc. These statements execute atomically with the creation of the database: if there is an error in any statement, the database is not created.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `state` - The current state of the database.

## » Import

Databases can be imported via their `instance` and `name` values, and optionally the `project` in which the instance is defined (Often used when the project is different to that defined in the provider). The format is thus either `{instanceName}/{dbName}` or `{projectId}/{instanceName}/{dbName}`. e.g.

```
$ terraform import google_spanner_database.db1 instance456/db789

$ terraform import google_spanner_database.db1 project123/instance456/db789
```

# » google_sql_database

Creates a new Google SQL Database on a Google SQL Database Instance. For more information, see the official documentation, or the JSON API. Postgres support for `google_sql_database` is in Beta.

## » Example Usage

Example creating a SQL Database.

```
resource "google_sql_database_instance" "master" {
  name = "master-instance"

  settings {
    tier = "D0"
  }
}

resource "google_sql_database" "users" {
  name      = "users-db"
  instance  = "${google_sql_database_instance.master.name}"
  charset   = "latin1"
  collation = "latin1_swedish_ci"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the database.
- `instance` - (Required) The name of containing instance.

---

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `charset` - (Optional) The charset value. See MySQL's Supported Character Sets and Collations and Postgres' Character Set Support for more details and supported values. Postgres databases are in Beta, and have limited `charset` support; they only support a value of `UTF8` at creation time.
- `collation` - (Optional) The collation value. See MySQL's Supported Character Sets and Collations and Postgres' Collation Support for more details and supported values. Postgres databases are in Beta, and have limited `collation` support; they only support a value of `en_US.UTF8` at creation time.

### » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.

### » Import

SQL databases can be imported using the `instance` and `name`, e.g.

```
$ terraform import google_sql_database.database master-instance:users-db
```

## » google_sql_database_instance

Creates a new Google SQL Database Instance. For more information, see the official documentation, or the JSON API. Postgres support for `google_sql_database_instance` is in Beta.

**NOTE on `google_sql_database_instance`:** - Second-generation instances include a default 'root'@'%' user with no password. This user will be deleted by Terraform on instance creation. You should use `google_sql_user` to define a custom user with a restricted host and strong password.

### » Example Usage

#### » SQL First Generation

```
resource "google_sql_database_instance" "master" {
  name = "master-instance"
  database_version = "MYSQL_5_6"
  # First-generation instance regions are not the conventional
  # Google Compute Engine regions. See argument reference below.
  region = "us-central"

  settings {
    tier = "D0"
  }
}
```

#### » SQL Second generation

```
resource "google_sql_database_instance" "master" {
  name = "master-instance"
  database_version = "POSTGRES_9_6"
  region = "us-central1"

  settings {
    # Second-generation instance tiers are based on the machine
    # type. See argument reference below.
    tier = "db-f1-micro"
  }
}
```

## » Argument Reference

The following arguments are supported:

- `region` - (Required) The region the instance will sit in. Note, first-generation Cloud SQL instance regions do not line up with the Google Compute Engine (GCE) regions, and Cloud SQL is not available in all regions - choose from one of the options listed here.
- `settings` - (Required) The settings to use for the database. The configuration is detailed below.

---

- `connection_name` - (Optional) The connection name of the instance to be used in connection strings.
- `database_version` - (Optional, Default: `MYSQL_5_6`) The MySQL version to use. Can be `MYSQL_5_6`, `MYSQL_5_7` or `POSTGRES_9_6` for second-generation instances, or `MYSQL_5_5` or `MYSQL_5_6` for first-generation instances. See Second Generation Capabilities for more information. `POSTGRES_9_6` support is in Beta.
- `name` - (Optional, Computed) The name of the instance. If the name is left blank, Terraform will randomly generate one when the instance is first created. This is done because after a name is used, it cannot be reused for up to one week.
- `master_instance_name` - (Optional) The name of the instance that will act as the master in the replication setup. Note, this requires the master to have `binary_log_enabled` set, as well as existing backups.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `replica_configuration` - (Optional) The configuration for replication. The configuration is detailed below.

The required `settings` block supports:

- tier - (Required) The machine tier (First Generation) or type (Second Generation) to use. See tiers for more details and supported versions. Postgres supports only shared-core machine types such as db-f1-micro, and custom machine types such as db-custom-2-13312. See the Custom Machine Type Documentation to learn about specifying custom machine types.
- activation_policy - (Optional) This specifies when the instance should be active. Can be either ALWAYS, NEVER or ON_DEMAND.
- authorized_gae_applications - (Optional) A list of Google App Engine (GAE) project names that are allowed to access this instance.
- availability_type - (Optional) This specifies whether a PostgreSQL instance should be set up for high availability (REGIONAL) or single zone (ZONAL).
- crash_safe_replication - (Optional) Specific to read instances, indicates when crash-safe replication flags are enabled.
- disk_autoresize - (Optional, Second Generation, Default: true) Configuration to increase storage size automatically.
- disk_size - (Optional, Second Generation, Default: 10) The size of data disk, in GB. Size of a running instance cannot be reduced but can be increased.
- disk_type - (Optional, Second Generation, Default: PD_SSD) The type of data disk: PD_SSD or PD_HDD.
- pricing_plan - (Optional, First Generation) Pricing plan for this instance, can be one of PER_USE or PACKAGE.
- replication_type - (Optional) Replication type for this instance, can be one of ASYNCHRONOUS or SYNCHRONOUS.

The optional settings.database_flags sublist supports:

- name - (Optional) Name of the flag.
- value - (Optional) Value of the flag.

The optional settings.backup_configuration subblock supports:

- binary_log_enabled - (Optional) True if binary logging is enabled. If logging is false, this must be as well.
- enabled - (Optional) True if backup configuration is enabled.
- start_time - (Optional) HH:MM format time indicating when backup configuration starts.

The optional settings.ip_configuration subblock supports:

- `ipv4_enabled` - (Optional) True if the instance should be assigned an IP address. The IPv4 address cannot be disabled for Second Generation instances.
- `require_ssl` - (Optional) True if mysqld should default to `REQUIRE X509` for users connecting over IP.

The optional `settings.ip_configuration.authorized_networks[]` sublist supports:

- `expiration_time` - (Optional) The RFC 3339 formatted date time string indicating when this whitelist expires.
- `name` - (Optional) A name for this whitelist entry.
- `value` - (Optional) A CIDR notation IPv4 or IPv6 address that is allowed to access this instance. Must be set even if other two attributes are not for the whitelist to become active.

The optional `settings.location_preference` subblock supports:

- `follow_gae_application` - (Optional) A GAE application whose zone to remain in. Must be in the same region as this instance.
- `zone` - (Optional) The preferred compute engine zone.

The optional `settings.maintenance_window` subblock for Second Generation instances declares a one-hour maintenance window when an Instance can automatically restart to apply updates. The maintenance window is specified in UTC time. It supports:

- `day` - (Optional) Day of week (`1-7`), starting on Monday
- `hour` - (Optional) Hour of day (`0-23`), ignored if `day` not set
- `update_track` - (Optional) Receive updates earlier (`canary`) or later (`stable`)

The optional `replica_configuration` block must have `master_instance_name` set to work, cannot be updated, and supports:

- `ca_certificate` - (Optional) PEM representation of the trusted CA's x509 certificate.
- `client_certificate` - (Optional) PEM representation of the slave's x509 certificate.
- `client_key` - (Optional) PEM representation of the slave's private key. The corresponding public key in encoded in the `client_certificate`.
- `connect_retry_interval` - (Optional, Default: 60) The number of seconds between connect retries.
- `dump_file_path` - (Optional) Path to a SQL file in GCS from which slave instances are created. Format is `gs://bucket/filename`.

- `failover_target` - (Optional) Specifies if the replica is the failover target. If the field is set to true the replica will be designated as a failover replica. If the master instance fails, the replica instance will be promoted as the new master instance.
- `master_heartbeat_period` - (Optional) Time in ms between replication heartbeats.
- `password` - (Optional) Password for the replication connection.
- `sslCipher` - (Optional) Permissible ciphers for use in SSL encryption.
- `username` - (Optional) Username for replication connection.
- `verify_server_certificate` - (Optional) True if the master's common name value is checked during the SSL handshake.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `first_ip_address` - The first IPv4 address of the addresses assigned. This is is to support accessing the first address in the list in a terraform output when the resource is configured with a `count`.
- `ip_address.0.ip_address` - The IPv4 address assigned.
- `ip_address.0.time_to_retire` - The time this IP address will be retired, in RFC 3339 format.
- `self_link` - The URI of the created resource.
- `settings.version` - Used to make sure changes to the `settings` block are atomic.

## » Import

Database instances can be imported using the `name`, e.g.

```
$ terraform import google_sql_database_instance.master master-instance
```

# » google_sql_user

Creates a new Google SQL User on a Google SQL User Instance. For more information, see the official documentation, or the JSON API.

**Note:** All arguments including the username and password will be stored in the raw state as plain-text. Read more about sensitive data in state. Passwords will not be retrieved when running ”terraform import”.

## » Example Usage

Example creating a SQL User.

```
resource "google_sql_database_instance" "master" {
  name = "master-instance"

  settings {
    tier = "D0"
  }
}

resource "google_sql_user" "users" {
  name     = "me"
  instance = "${google_sql_database_instance.master.name}"
  host     = "me.com"
  password = "changeme"
}
```

## » Argument Reference

The following arguments are supported:

- `instance` - (Required) The name of the Cloud SQL instance. Changing this forces a new resource to be created.
- `name` - (Required) The name of the user. Changing this forces a new resource to be created.
- `password` - (Optional) The password for the user. Can be updated.

---

- `host` - (Optional) The host the user can connect from. This is only supported for first generation SQL instances. Don’t set this field for second generation SQL instances. Can be an IP address. Changing this forces a new resource to be created.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

### » Attributes Reference

Only the arguments listed above are exposed as attributes.

### » Import

SQL users for 1st generation databases can be imported using the `instance`, `host` and `name`, e.g.

```
$ terraform import google_sql_user.users master-instance/my-domain.com/me
```

SQL users for 2nd generation databases can be imported using the `instance` and `name`, e.g.

```
$ terraform import google_sql_user.users master-instance/me
```

## » google_logging_billing_account_sink

Manages a billing account logging sink. For more information see the official documentation and Exporting Logs in the API.

Note that you must have the "Logs Configuration Writer" IAM role (`roles/logging.configWriter`) granted to the credentials used with terraform.

### » Example Usage

```
resource "google_logging_billing_account_sink" "my-sink" {
    name = "my-sink"
    billing_account = "ABCDEF-012345-GHIJKL"

    # Can export to pubsub, cloud storage, or bigtable
    destination = "storage.googleapis.com/${google_storage_bucket.log-bucket.name}"
}

resource "google_storage_bucket" "log-bucket" {
    name     = "billing-logging-bucket"
}

resource "google_project_iam_binding" "log-writer" {
    role = "roles/storage.objectCreator"

    members = [
        "${google_logging_billing_account_sink.my-sink.writer_identity}",
    ]
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the logging sink.
- `billing_account` - (Required) The billing account exported to the sink.
- `destination` - (Required) The destination of the sink (or, in other words, where logs are written to). Can be a Cloud Storage bucket, a PubSub topic, or a BigQuery dataset. Examples: `"storage.googleapis.com/[GCS_BUCKET]"` `"bigquery.googleapis.com/projects/[PROJECT_ID]/da` `"pubsub.googleapis.com/projects/[PROJECT_ID]/topics/[TOPIC_ID]"` The writer associated with the sink must have access to write to the above resource.
- `filter` - (Optional) The filter to apply when exporting logs. Only log entries that match the filter are exported. See Advanced Log Filters for information on how to write a filter.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `writer_identity` - The identity associated with this sink. This identity must be granted write access to the configured `destination`.

# » google_logging_organization_sink

Manages a organization-level logging sink. For more information see the official documentation and Exporting Logs in the API.

Note that you must have the ”Logs Configuration Writer” IAM role (`roles/logging.configWriter`) granted to the credentials used with terraform.

## » Example Usage

```
resource "google_logging_organization_sink" "my-sink" {
    name        = "my-sink"
    org_id      = "123456789"

    # Can export to pubsub, cloud storage, or bigtable
    destination = "storage.googleapis.com/${google_storage_bucket.log-bucket.name}"

    # Log all WARN or higher severity messages relating to instances
    filter      = "resource.type = gce_instance AND severity >= WARN"
}

resource "google_storage_bucket" "log-bucket" {
    name = "organization-logging-bucket"
}

resource "google_project_iam_binding" "log-writer" {
    role    = "roles/storage.objectCreator"

    members = [
        "${google_logging_organization_sink.my-sink.writer_identity}",
    ]
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the logging sink.
- `org_id` - (Required) The numeric ID of the organization to be exported to the sink.
- `destination` - (Required) The destination of the sink (or, in other words, where logs are written to). Can be a Cloud Storage bucket, a PubSub topic, or a BigQuery dataset. Examples: `"storage.googleapis.com/[GCS_BUCKET]"` `"bigquery.googleapis.com/projects/[PROJECT_ID]/da` `"pubsub.googleapis.com/projects/[PROJECT_ID]/topics/[TOPIC_ID]"` The writer associated with the sink must have access to write to the above resource.
- `filter` - (Optional) The filter to apply when exporting logs. Only log entries that match the filter are exported. See Advanced Log Filters for information on how to write a filter.
- `include_children` - (Optional) Whether or not to include children organizations in the sink export. If true, logs associated with child projects are also exported; otherwise only logs relating to the provided organization are included.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `writer_identity` - The identity associated with this sink. This identity must be granted write access to the configured `destination`.

# » google_logging_folder_sink

Manages a folder-level logging sink. For more information see the official documentation and Exporting Logs in the API.

Note that you must have the "Logs Configuration Writer" IAM role (`roles/logging.configWriter`) granted to the credentials used with terraform.

## » Example Usage

```
resource "google_logging_folder_sink" "my-sink" {
    name        = "my-sink"
    folder      = "${google_folder.my-folder.name}"

    # Can export to pubsub, cloud storage, or bigtable
    destination = "storage.googleapis.com/${google_storage_bucket.log-bucket.name}"

    # Log all WARN or higher severity messages relating to instances
    filter      = "resource.type = gce_instance AND severity >= WARN"
}

resource "google_storage_bucket" "log-bucket" {
    name = "folder-logging-bucket"
}

resource "google_project_iam_binding" "log-writer" {
    role    = "roles/storage.objectCreator"

    members = [
        "${google_logging_folder_sink.my-sink.writer_identity}",
    ]
}

resource "google_folder" "my-folder" {
    display_name = "My folder"
```

```
  parent       = "organizations/123456"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the logging sink.
- `folder` - (Required) The folder to be exported to the sink. Note that either [FOLDER_ID] or ”folders/[FOLDER_ID]” is accepted.
- `destination` - (Required) The destination of the sink (or, in other words, where logs are written to). Can be a Cloud Storage bucket, a PubSub topic, or a BigQuery dataset. Examples: `"storage.googleapis.com/[GCS_BUCKET]"` `"bigquery.googleapis.com/projects/[PROJECT_ID]/da` `"pubsub.googleapis.com/projects/[PROJECT_ID]/topics/[TOPIC_ID]"` The writer associated with the sink must have access to write to the above resource.
- `filter` - (Optional) The filter to apply when exporting logs. Only log entries that match the filter are exported. See Advanced Log Filters for information on how to write a filter.
- `include_children` - (Optional) Whether or not to include children folders in the sink export. If true, logs associated with child projects are also exported; otherwise only logs relating to the provided folder are included.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `writer_identity` - The identity associated with this sink. This identity must be granted write access to the configured `destination`.

# » google_logging_project_sink

Manages a project-level logging sink. For more information see the official documentation, Exporting Logs in the API and API.

Note that you must have the ”Logs Configuration Writer” IAM role (`roles/logging.configWriter`) granted to the credentials used with terraform.

## » Example Usage

```
resource "google_logging_project_sink" "my-sink" {
    name = "my-pubsub-instance-sink"

    # Can export to pubsub, cloud storage, or bigtable
    destination = "pubsub.googleapis.com/projects/my-project/topics/instance-activity"

    # Log all WARN or higher severity messages relating to instances
    filter = "resource.type = gce_instance AND severity >= WARN"

    # Use a unique writer (creates a unique service account used for writing)
    unique_writer_identity = true
}
```

A more complete example follows: this creates a compute instance, as well as a log sink that logs all activity to a cloud storage bucket. Because we are using `unique_writer_identity`, we must grant it access to the bucket. Note that this grant requires the ”Project IAM Admin” IAM role (`roles/resourcemanager.projectIamAdmin`) granted to the credentials used with terraform.

```
# Our logged compute instance
resource "google_compute_instance" "my-logged-instance" {
  name         = "my-instance"
  machine_type = "n1-standard-1"
  zone         = "us-central1-a"

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-8"
    }
  }

  network_interface {
    network = "default"

    access_config {}
  }
}

# A bucket to store logs in
resource "google_storage_bucket" "log-bucket" {
    name     = "my-unique-logging-bucket"
}
```

```
# Our sink; this logs all activity related to our "my-logged-instance" instance
resource "google_logging_project_sink" "instance-sink" {
    name = "my-instance-sink"
    destination = "storage.googleapis.com/${google_storage_bucket.log-bucket.name}"
    filter = "resource.type = gce_instance AND resource.labels.instance_id = \"${google_comp

    unique_writer_identity = true
}

# Because our sink uses a unique_writer, we must grant that writer access to the bucket.
resource "google_project_iam_binding" "log-writer" {
    role = "roles/storage.objectCreator"

    members = [
        "${google_logging_project_sink.instance-sink.writer_identity}",
    ]
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the logging sink.
- `destination` - (Required) The destination of the sink (or, in other words, where logs are written to). Can be a Cloud Storage bucket, a PubSub topic, or a BigQuery dataset. Examples: `"storage.googleapis.com/[GCS_BUCKET]"` `"bigquery.googleapis.com/projects/[PROJECT_ID]/da` `"pubsub.googleapis.com/projects/[PROJECT_ID]/topics/[TOPIC_ID]"` The writer associated with the sink must have access to write to the above resource.
- `filter` - (Optional) The filter to apply when exporting logs. Only log entries that match the filter are exported. See Advanced Log Filters for information on how to write a filter.
- `project` - (Optional) The ID of the project to create the sink in. If omitted, the project associated with the provider is used.
- `unique_writer_identity` - (Optional) Whether or not to create a unique identity associated with this sink. If `false` (the default), then the `writer_identity` used is `serviceAccount:cloud-logs@system.gserviceaccount.com`. If `true`, then a unique service account is created and used for this sink. If you wish to publish logs across projects, you must set `unique_writer_identity` to true.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `writer_identity` - The identity associated with this sink. This identity must be granted write access to the configured `destination`.

## » Import

Project-level logging sinks can be imported using their URI, e.g.

```
$ terraform import google_logging_project_sink.my_sink projects/my-project/sinks/my-sink
```

# » google_storage_bucket

Creates a new bucket in Google cloud storage service (GCS). Once a bucket has been created, its location can't be changed. ACLs can be applied using the `google_storage_bucket_acl` resource. For more information see the official documentation and API.

## » Example Usage

Example creating a private bucket in standard storage, in the EU region.

```
resource "google_storage_bucket" "image-store" {
  name     = "image-store-bucket"
  location = "EU"

  website {
    main_page_suffix = "index.html"
    not_found_page   = "404.html"
  }
}
```

## » Argument Reference

The following arguments are supported:

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

---

- `force_destroy` - (Optional, Default: false) When deleting a bucket, this boolean option will delete all contained objects. If you try to delete a bucket that contains objects, Terraform will fail that run.
- `location` - (Optional, Default: 'US') The GCS location
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `storage_class` - (Optional) The Storage Class of the new bucket. Supported values include: `MULTI_REGIONAL`, `REGIONAL`, `NEARLINE`, `COLDLINE`.
- `lifecycle_rule` - (Optional) The bucket's Lifecycle Rules configuration. Multiple blocks of this type are permitted. Structure is documented below.
- `versioning` - (Optional) The bucket's Versioning configuration.
- `website` - (Optional) Configuration if the bucket acts as a website. Structure is documented below.
- `cors` - (Optional) The bucket's Cross-Origin Resource Sharing (CORS) configuration. Multiple blocks of this type are permitted. Structure is documented below.
- `labels` - (Optional) A set of key/value label pairs to assign to the bucket.
- `logging` - (Optional) The bucket's Access & Storage Logs configuration.

The `lifecycle_rule` block supports:

- `action` - (Required) The Lifecycle Rule's action configuration. A single block of this type is supported. Structure is documented below.
- `condition` - (Required) The Lifecycle Rule's condition configuration. A single block of this type is supported. Structure is documented below.

The `action` block supports:

- `type` - The type of the action of this Lifecycle Rule. Supported values include: `Delete` and `SetStorageClass`.
- `storage_class` - (Required if action type is `SetStorageClass`) The target Storage Class of objects affected by this Lifecycle Rule. Supported values include: `MULTI_REGIONAL`, `REGIONAL`, `NEARLINE`, `COLDLINE`.

The `condition` block supports the following elements, and requires at least one to be defined:

- `age` - (Optional) Minimum age of an object in days to satisfy this condition.
- `created_before` - (Optional) Creation date of an object in RFC 3339 (e.g. `2017-06-13`) to satisfy this condition.

- `is_live` - (Optional) Relevant only for versioned objects. If `true`, this condition matches live objects, archived objects otherwise.
- `matches_storage_class` - (Optional) Storage Class of objects to satisfy this condition. Supported values include: `MULTI_REGIONAL`, `REGIONAL`, `NEARLINE`, `COLDLINE`, `STANDARD`, `DURABLE_REDUCED_AVAILABILITY`.
- `num_newer_versions` - (Optional) Relevant only for versioned objects. The number of newer versions of an object to satisfy this condition.

The `versioning` block supports:

- `enabled` - (Optional) While set to `true`, versioning is fully enabled for this bucket.

The `website` block supports:

- `main_page_suffix` - (Optional) Behaves as the bucket's directory index where missing objects are treated as potential directories.
- `not_found_page` - (Optional) The custom object to return when a requested resource is not found.

The `cors` block supports:

- `origin` - (Optional) The list of Origins eligible to receive CORS response headers. Note: "*" is permitted in the list of origins, and means "any Origin".
- `method` - (Optional) The list of HTTP methods on which to include CORS response headers, (GET, OPTIONS, POST, etc) Note: "*" is permitted in the list of methods, and means "any method".
- `response_header` - (Optional) The list of HTTP headers other than the simple response headers to give permission for the user-agent to share across domains.
- `max_age_seconds` - (Optional) The value, in seconds, to return in the Access-Control-Max-Age header used in preflight responses.

The `logging` block supports:

- `log_bucket` - (Required) The bucket that will receive log objects.
- `log_object_prefix` - (Optional, Computed) The object prefix for log objects. If it's not provided, by default GCS sets this to the log_bucket's name.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.
- `url` - The base URL of the bucket, in the format `gs://<bucket-name>`.

## » Import

Storage buckets can be imported using the `name`, e.g.

```
$ terraform import google_storage_bucket.image-store image-store-bucket
```

# » google_storage_bucket_acl

Creates a new bucket ACL in Google cloud storage service (GCS). For more information see the official documentation and API.

## » Example Usage

Example creating an ACL on a bucket with one owner, and one reader.

```
resource "google_storage_bucket" "image-store" {
  name     = "image-store-bucket"
  location = "EU"
}

resource "google_storage_bucket_acl" "image-store-acl" {
  bucket = "${google_storage_bucket.image-store.name}"

  role_entity = [
    "OWNER:user-my.email@gmail.com",
    "READER:group-mygroup",
  ]
}
```

## » Argument Reference

- `bucket` - (Required) The name of the bucket it applies to.

---

- `predefined_acl` - (Optional) The canned GCS ACL to apply. Must be set if `role_entity` is not.
- `role_entity` - (Optional) List of role/entity pairs in the form `ROLE:entity`. See GCS Bucket ACL documentation for more details. Must be set if `predefined_acl` is not.

- `default_acl` - (Optional) Configure this ACL to be the default ACL.

### » Attributes Reference

Only the arguments listed above are exposed as attributes.

## » IAM policy for Google storage bucket

Two different resources help you manage your IAM policy for storage bucket. Each of these resources serves a different use case:

- `google_storage_bucket_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the storage bucket are preserved.
- `google_storage_bucket_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the storage bucket are preserved.

**Note:** `google_storage_bucket_iam_binding` resources **can be** used in conjunction with `google_storage_bucket_iam_member` resources **only if** they do not grant privilege to the same role.

### » google_storage_bucket_iam_binding

```
resource "google_storage_bucket_iam_binding" "binding" {
  bucket = "your-bucket-name"
  role        = "roles/storage.objectViewer"

  members = [
    "user:jane@example.com",
  ]
}
```

### » google_storage_bucket_iam_member

```
resource "google_storage_bucket_iam_member" "member" {
  bucket = "your-bucket-name"
  role        = "roles/storage.objectViewer"
  member      = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket it applies to.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the storage bucket's IAM policy.

# » IAM policy for Google storage bucket

Two different resources help you manage your IAM policy for storage bucket. Each of these resources serves a different use case:

- `google_storage_bucket_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the storage bucket are preserved.
- `google_storage_bucket_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the storage bucket are preserved.

**Note:** `google_storage_bucket_iam_binding` resources **can be** used in conjunction with `google_storage_bucket_iam_member` resources **only if** they do not grant privilege to the same role.

## » google_storage_bucket_iam_binding

```
resource "google_storage_bucket_iam_binding" "binding" {
  bucket = "your-bucket-name"
  role        = "roles/storage.objectViewer"

  members = [
    "user:jane@example.com",
  ]
}
```

## » google_storage_bucket_iam_member

```
resource "google_storage_bucket_iam_member" "member" {
  bucket = "your-bucket-name"
  role        = "roles/storage.objectViewer"
  member      = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket it applies to.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.

  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.

- `role` - (Required) The role that should be applied. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

### » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the storage bucket's IAM policy.

## » google_storage_bucket_object

Creates a new object inside an existing bucket in Google cloud storage service (GCS). ACLs can be applied using the `google_storage_object_acl` resource. For more information see the official documentation and API.

### » Example Usage

Example creating a public object in an existing `image-store` bucket.

```
resource "google_storage_bucket_object" "picture" {
  name   = "butterfly01"
  source = "/images/nature/garden-tiger-moth.jpg"
  bucket = "image-store"
}
```

### » Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the containing bucket.
- `name` - (Required) The name of the object.

One of the following is required:

- `content` - (Optional) Data as `string` to be uploaded. Must be defined if `source` is not.
- `source` - (Optional) A path to the data you want to upload. Must be defined if `content` is not.

- cache_control - (Optional) Cache-Control directive to specify caching behavior of object data. If omitted and object is accessible to all anonymous users, the default will be public, max-age=3600
- content_disposition - (Optional) Content-Disposition of the object data.
- content_encoding - (Optional) Content-Encoding of the object data.
- content_language - (Optional) Content-Language of the object data.
- content_type - (Optional) Content-Type of the object data. Defaults to "application/octet-stream" or "text/plain; charset=utf-8".
- storage_class - (Optional) The StorageClass of the new bucket object. Supported values include: MULTI_REGIONAL, REGIONAL, NEARLINE, COLDLINE. If not provided, this defaults to the bucket's default storage class or to a standard class.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- crc32c - (Computed) Base 64 CRC32 hash of the uploaded data.
- md5hash - (Computed) Base 64 MD5 hash of the uploaded data.

# » google_storage_default_object_acl

Creates a new default object ACL in Google Cloud Storage service (GCS). For more information see the official documentation and API.

## » Example Usage

Example creating a default object ACL on a bucket with one owner, and one reader.

```
resource "google_storage_bucket" "image-store" {
  name     = "image-store-bucket"
  location = "EU"
}

resource "google_storage_default_object_acl" "image-store-default-acl" {
  bucket = "${google_storage_bucket.image-store.name}"
  role_entity = [
    "OWNER:user-my.email@gmail.com",
    "READER:group-mygroup",
  ]
}
```

### » Argument Reference

- `bucket` - (Required) The name of the bucket it applies to.
- `role_entity` - (Required) List of role/entity pairs in the form `ROLE:entity`. See GCS Object ACL documentation for more details.

### » Attributes Reference

Only the arguments listed above are exposed as attributes.

## » google_storage_notification

Creates a new notification configuration on a specified bucket, establishing a flow of event notifications from GCS to a Cloud Pub/Sub topic. For more information see the official documentation and API.

### » Example Usage

```
resource "google_storage_bucket" "bucket" {
    name = "default_bucket"
}

resource "google_pubsub_topic" "topic" {
    name = "default_topic"
}

// In order to enable notifications,
// a GCS service account unique to each project
// must have the IAM permission "projects.topics.publish" to a Cloud Pub/Sub topic from this
// The only reference to this requirement can be found here:
// https://cloud.google.com/storage/docs/gsutil/commands/notification
// The GCS service account has the format of <project-id>@gs-project-accounts.iam.gserviceac
// API for retrieving it https://cloud.google.com/storage/docs/json_api/v1/projects/serviceA

resource "google_pubsub_topic_iam_binding" "binding" {
    topic       = "${google_pubsub_topic.topic.name}"
    role        = "roles/pubsub.publisher"

    members     = ["serviceAccount:my-project-id@gs-project-accounts.iam.gserviceaccount.com
}

resource "google_storage_notification" "notification" {
    bucket            = "${google_storage_bucket.bucket.name}"
    payload_format    = "JSON_API_V1"
    topic             = "${google_pubsub_topic.topic.id}"
    event_types       = ["%s","%s"]
    custom_attributes {
        new-attribute = "new-attribute-value"
    }
    depends_on        = ["google_pubsub_topic_iam_binding.binding"]
}
```

## » Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket.
- `payload_format` - (Required) The desired content of the Payload. One of `"JSON_API_V1"` or `"NONE"`.
- `topic` - (Required) The Cloud PubSub topic to which this subscription publishes.

---

- `custom_attributes` - (Optional) A set of key/value attribute pairs to attach to each Cloud PubSub message published for this notification subscription
- `event_types` - (Optional) List of event type filters for this notification config. If not specified, Cloud Storage will send notifications for all event types. The valid types are: `"OBJECT_FINALIZE"`, `"OBJECT_METADATA_UPDATE"`, `"OBJECT_DELETE"`, `"OBJECT_ARCHIVE"`
- `object_name_prefix` - (Optional) Specifies a prefix path filter for this notification config. Cloud Storage will only send notifications for objects in this bucket whose names begin with the specified prefix.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `self_link` - The URI of the created resource.

## » Import

Storage notifications can be imported using the notification `id` in the format `<bucket_name>/notificationConfigs/<id>` e.g.

```
$ terraform import google_storage_notification.notification default_bucket/notificationConfi
```

# » google_storage_object_acl

Creates a new object ACL in Google cloud storage service (GCS). For more information see the official documentation and API.

## » Example Usage

Create an object ACL with one owner and one reader.

```
resource "google_storage_bucket" "image-store" {
  name     = "image-store-bucket"
  location = "EU"
}

resource "google_storage_bucket_object" "image" {
  name   = "image1"
  bucket = "${google_storage_bucket.image-store.name}"
  source = "image1.jpg"
}

resource "google_storage_object_acl" "image-store-acl" {
  bucket = "${google_storage_bucket.image-store.name}"
  object = "${google_storage_bucket_object.image.name}"

  role_entity = [
    "OWNER:user-my.email@gmail.com",
    "READER:group-mygroup",
  ]
}
```

## » Argument Reference

- `bucket` - (Required) The name of the bucket it applies to.
- `object` - (Required) The name of the object it applies to.

---

- `predefined_acl` - (Optional) The canned GCS ACL to apply. Must be set if `role_entity` is not.
- `role_entity` - (Optional) List of role/entity pairs in the form `ROLE:entity`. See GCS Object ACL documentation for more details. Must be set if `predefined_acl` is not.

## » Attributes Reference

Only the arguments listed above are exposed as attributes.

# » google_kms_key_ring

Allows creation of a Google Cloud Platform KMS KeyRing. For more information see the official documentation and API.

A KeyRing is a grouping of CryptoKeys for organizational purposes. A KeyRing belongs to a Google Cloud Platform Project and resides in a specific location.

Note: KeyRings cannot be deleted from Google Cloud Platform. Destroying a Terraform-managed KeyRing will remove it from state but **will not delete the resource on the server**.

## » Example Usage

```
resource "google_kms_key_ring" "my_key_ring" {
  name     = "my-key-ring"
  location = "us-central1"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The KeyRing's name. A KeyRing's name must be unique within a location and match the regular expression `[a-zA-Z0-9_-]{1,63}`

- `location` - (Required) The Google Cloud Platform location for the KeyRing. A full list of valid locations can be found by running `gcloud kms locations list`.

---

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `id` - The ID of the created KeyRing. Its format is `{projectId}/{location}/{keyRingName}`.

## » Import

KeyRings can be imported using the KeyRing autogenerated `id`, e.g.

```
$ terraform import google_kms_key_ring.my_key_ring my-gcp-project/us-central1/my-key-ring

$ terraform import google_kms_key_ring.my_key_ring us-central1/my-key-ring
```

# » IAM policy for Google Cloud KMS key ring

Three different resources help you manage your IAM policy for KMS key ring. Each of these resources serves a different use case:

- `google_kms_key_ring_iam_policy`: Authoritative. Sets the IAM policy for the key ring and replaces any existing policy already attached.
- `google_kms_key_ring_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the key ring are preserved.
- `google_kms_key_ring_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the key ring are preserved.

**Note:** `google_kms_key_ring_iam_policy` **cannot** be used in conjunction with `google_kms_key_ring_iam_binding` and `google_kms_key_ring_iam_member` or they will fight over what your policy should be.

**Note:** `google_kms_key_ring_iam_binding` resources **can be** used in conjunction with `google_kms_key_ring_iam_member` resources **only if** they do not grant privilege to the same role.

## » google_kms_key_ring_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"

    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_kms_key_ring_iam_policy" "key_ring" {
    key_ring_id = "your-key-ring-id"
    policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_kms_key_ring_iam_binding

```
resource "google_kms_key_ring_iam_binding" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"

  members = [
    "user:jane@example.com",
  ]
}
```

## » google_kms_key_ring_iam_member

```
resource "google_kms_key_ring_iam_member" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"
  member      = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `key_ring_id` - (Required) The key ring ID, in the form `{project_id}/{location_name}/{key_ring_nam` or `{location_name}/{key_ring_name}`. In the second form, the provider's project setting will be used as a fallback.

- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_kms_key_ring_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `policy_data` - (Required only by `google_kms_key_ring_iam_policy`) The policy data generated by a `google_iam_policy` data source.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the key ring's IAM policy.

## » Import

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account. This member resource can be imported using the `key_ring_id`, role, and account e.g.

```
$ terraform import google_kms_key_ring_iam_member.key_ring_iam "your-project-id/location-nam
```

IAM binging imports use space-delimited identifiers; the resource in question and the role. This binding resource can be imported using the `key_ring_id`, role, and account e.g.

```
$ terraform import google_kms_key_ring_iam_binding.key_ring_iam "your-project-id/location-na
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `key_ring_id`, role, and account e.g.

```
$ terraform import google_kms_key_ring_iam_policy.key_ring_iam your-project-id/location-name
```

## » IAM policy for Google Cloud KMS key ring

Three different resources help you manage your IAM policy for KMS key ring. Each of these resources serves a different use case:

- `google_kms_key_ring_iam_policy`: Authoritative. Sets the IAM policy for the key ring and replaces any existing policy already attached.
- `google_kms_key_ring_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the key ring are preserved.
- `google_kms_key_ring_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the key ring are preserved.

**Note:** `google_kms_key_ring_iam_policy` **cannot** be used in conjunction with `google_kms_key_ring_iam_binding` and `google_kms_key_ring_iam_member` or they will fight over what your policy should be.

**Note:** `google_kms_key_ring_iam_binding` resources **can be** used in conjunction with `google_kms_key_ring_iam_member` resources **only if** they do not grant privilege to the same role.

### » google_kms_key_ring_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"

    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_kms_key_ring_iam_policy" "key_ring" {
    key_ring_id = "your-key-ring-id"
    policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

### » google_kms_key_ring_iam_binding

```
resource "google_kms_key_ring_iam_binding" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"

  members = [
    "user:jane@example.com",
  ]
}
```

## » google_kms_key_ring_iam_member

```
resource "google_kms_key_ring_iam_member" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"
  member      = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `key_ring_id` - (Required) The key ring ID, in the form `{project_id}/{location_name}/{key_ring_nam` or `{location_name}/{key_ring_name}`. In the second form, the provider's project setting will be used as a fallback.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_kms_key_ring_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

- `policy_data` - (Required only by `google_kms_key_ring_iam_policy`) The policy data generated by a `google_iam_policy` data source.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the key ring's IAM policy.

## » Import

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account. This member resource can be imported using the `key_ring_id`, role, and account e.g.

```
$ terraform import google_kms_key_ring_iam_member.key_ring_iam "your-project-id/location-nam
```

IAM binging imports use space-delimited identifiers; the resource in question and the role. This binding resource can be imported using the `key_ring_id`, role, and account e.g.

```
$ terraform import google_kms_key_ring_iam_binding.key_ring_iam "your-project-id/location-na
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `key_ring_id`, role, and account e.g.

```
$ terraform import google_kms_key_ring_iam_policy.key_ring_iam your-project-id/location-name
```

# » IAM policy for Google Cloud KMS key ring

Three different resources help you manage your IAM policy for KMS key ring. Each of these resources serves a different use case:

- `google_kms_key_ring_iam_policy`: Authoritative. Sets the IAM policy for the key ring and replaces any existing policy already attached.
- `google_kms_key_ring_iam_binding`: Authoritative for a given role. Updates the IAM policy to grant a role to a list of members. Other roles within the IAM policy for the key ring are preserved.
- `google_kms_key_ring_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the key ring are preserved.

**Note:** `google_kms_key_ring_iam_policy` **cannot** be used in conjunction with `google_kms_key_ring_iam_binding` and `google_kms_key_ring_iam_member` or they will fight over what your policy should be.

**Note:** google_kms_key_ring_iam_binding resources **can be** used in conjunction with google_kms_key_ring_iam_member resources **only if** they do not grant privilege to the same role.

## » google_kms_key_ring_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"

    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_kms_key_ring_iam_policy" "key_ring" {
    key_ring_id = "your-key-ring-id"
    policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

## » google_kms_key_ring_iam_binding

```
resource "google_kms_key_ring_iam_binding" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"

  members = [
    "user:jane@example.com",
  ]
}
```

## » google_kms_key_ring_iam_member

```
resource "google_kms_key_ring_iam_member" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"
  member      = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `key_ring_id` - (Required) The key ring ID, in the form `{project_id}/{location_name}/{key_ring_nam` or `{location_name}/{key_ring_name}`. In the second form, the provider's project setting will be used as a fallback.
- `member/members` - (Required) Identities that will be granted the privilege in `role`. Each entry can have one of the following values:
  - **allUsers**: A special identifier that represents anyone who is on the internet; with or without a Google account.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.com or joe@example.com.
  - **serviceAccount:{emailid}**: An email address that represents a service account. For example, my-other-app@appspot.gserviceaccount.com.
  - **group:{emailid}**: An email address that represents a Google group. For example, admins@example.com.
  - **domain:{domain}**: A Google Apps domain name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required) The role that should be applied. Only one `google_kms_key_ring_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `policy_data` - (Required only by `google_kms_key_ring_iam_policy`) The policy data generated by a `google_iam_policy` data source.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the key ring's IAM policy.

## » Import

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account. This member resource can be imported using the `key_ring_id`, role, and account e.g.

```
$ terraform import google_kms_key_ring_iam_member.key_ring_iam "your-project-id/location-nam
```

IAM binging imports use space-delimited identifiers; the resource in question and the role. This binding resource can be imported using the `key_ring_id`, role, and account e.g.

```
$ terraform import google_kms_key_ring_iam_binding.key_ring_iam "your-project-id/location-na
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `key_ring_id`, role, and account e.g.

```
$ terraform import google_kms_key_ring_iam_policy.key_ring_iam your-project-id/location-name
```

# » google_kms_crypto_key

Allows creation of a Google Cloud Platform KMS CryptoKey. For more information see the official documentation and API.

A CryptoKey is an interface to key material which can be used to encrypt and decrypt data. A CryptoKey belongs to a Google Cloud KMS KeyRing.

Note: CryptoKeys cannot be deleted from Google Cloud Platform. Destroying a Terraform-managed CryptoKey will remove it from state and delete all CryptoKeyVersions, rendering the key unusable, but **will not delete the resource on the server**.

## » Example Usage

```
resource "google_kms_key_ring" "my_key_ring" {
  name     = "my-key-ring"
  project  = "my-project"
  location = "us-central1"
}

resource "google_kms_crypto_key" "my_crypto_key" {
  name            = "my-crypto-key"
  key_ring        = "${google_kms_key_ring.my_key_ring.id}"
  rotation_period = "100000s"
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The CryptoKey's name. A CryptoKey's name must be unique within a location and match the regular expression `[a-zA-Z0-9_-]{1,63}`

- `key_ring` - (Required) The id of the Google Cloud Platform KeyRing to which the key shall belong.

- `rotation_period` - (Optional) Every time this period passes, generate a new CryptoKeyVersion and set it as the primary. The first rotation will take place after the specified period. The rotation period has the format of a decimal number with up to 9 fractional digits, followed by the letter s (seconds). It must be greater than a day (ie, 83400).

### » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `id` - The ID of the created CryptoKey. Its format is `{projectId}/{location}/{keyRingName}/{cryptoK`

### » Import

CryptoKeys can be imported using the CryptoKey autogenerated `id`, e.g.

```
$ terraform import google_kms_crypto_key.my_crypto_key my-gcp-project/us-central1/my-key-rin

$ terraform import google_kms_crypto_key.my_crypto_key us-central1/my-key-ring/my-crypto-key
```

## » google_kms_crypto_key_iam_binding

Allows creation and management of a single binding within IAM policy for an existing Google Cloud KMS crypto key.

### » Example Usage

```
resource "google_kms_crypto_key_iam_binding" "crypto_key" {
  crypto_key_id = "your-crypto-key-id"
  role          = "roles/editor"

  members = [
    "user:jane@example.com",
  ]
}
```

### » Argument Reference

The following arguments are supported:

- `members` - (Required) A list of users that the role should apply to.
- `role` - (Required) The role that should be applied. Only one `google_kms_crypto_key_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `crypto_key_id` - (Required) The crypto key ID, in the form `{project_id}/{location_name}/{key_ring_name}/{crypto_key_name}` or `{location_name}/{key_ring_name}/{crypto_key_name}`. In the second form, the provider's project setting will be used as a fallback.

### » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the crypto key's IAM policy.

### » Import

IAM binding imports use space-delimited identifiers; first the resource in question and then the role. These bindings can be imported using the `crypto_key_id` and role, e.g.

```
$ terraform import google_kms_crypto_key_iam_binding.my_binding "your-project-id/location-na
```

## » google_kms_crypto_key_iam_member

Allows creation and management of a single member for a single binding within the IAM policy for an existing Google Cloud KMS crypto key.

**Note:** This resource *must not* be used in conjunction with `google_kms_crypto_key_iam_policy` or they will fight over what your policy should be. Similarly, roles controlled by `google_kms_crypto_key_iam_binding` should not be assigned to using `google_kms_crypto_key_iam_member`.

### » Example Usage

```
resource "google_kms_crypto_key_iam_member" "crypto_key" {
  crypto_key_id = "your-crypto-key-id"
  role          = "roles/editor"
  member        = "user:jane@example.com"
}
```

## » Argument Reference

The following arguments are supported:

- `member` - (Required) The user that the role should apply to.
- `role` - (Required) The role that should be applied. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `crypto_key_id` - (Required) The key ring ID, in the form `{project_id}/{location_name}/{key_ring_n` or `{location_name}/{key_ring_name}/{crypto_key_name}`. In the second form, the provider's project setting will be used as a fallback.

## » Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `etag` - (Computed) The etag of the project's IAM policy.

## » Import

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account. This member resource can be imported using the `crypto_key_id`, role, and account e.g.

```
$ terraform import google_kms_crypto_key_iam_member.member "your-project-id/location-name/ke
```

# » google\cloudiot__registry

Creates a device registry in Google's Cloud IoT Core platform. For more information see the official documentation and API.

## » Example Usage

```
resource "google_pubsub_topic" "default-devicestatus" {
  name = "default-devicestatus"
}
```

```
resource "google_pubsub_topic" "default-telemetry" {
  name = "default-telemetry"
}

resource "google_cloudiot_registry" "default-registry" {
  name = "default-registry"

  event_notification_config = {
    pubsub_topic_name = "${google_pubsub_topic.default-devicestatus.id}"
  }

  state_notification_config = {
    pubsub_topic_name = "${google_pubsub_topic.default-telemetry.id}"
  }

  http_config = {
    http_enabled_state = "HTTP_ENABLED"
  }

  mqtt_config = {
    mqtt_enabled_state = "MQTT_ENABLED"
  }

  credentials = [
    {
      public_key_certificate = {
        format      = "X509_CERTIFICATE_PEM"
        certificate = "${file("rsa_cert.pem")}"
      }
    },
  ]
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the resource, required by device registry. Changing this forces a new resource to be created.

---

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `region` - (Optional) The Region in which the created address should reside. If it is not provided, the provider region is used.

- event_notification_config - (Optional) A PubSub topics to publish device events. Structure is documented below.
- state_notification_config - (Optional) A PubSub topic to publish device state updates. Structure is documented below.
- mqtt_config - (Optional) Activate or deactivate MQTT. Structure is documented below.
- http_config - (Optional) Activate or deactivate HTTP. Structure is documented below.
- credentials - (Optional) List of public key certificates to authenticate devices. Structure is documented below.

The event_notification_config block supports:

- pubsub_topic_name - (Required) PubSub topic name to publish device events.

The state_notification_config block supports:

- pubsub_topic_name - (Required) PubSub topic name to publish device state updates.

The mqtt_config block supports:

- mqtt_enabled_state - (Required) The field allows MQTT_ENABLED or MQTT_DISABLED.

The http_config block supports:

- http_enabled_state - (Required) The field allows HTTP_ENABLED or HTTP_DISABLED.

The credentials block supports:

- public_key_certificate - (Required) The certificate format and data.

The public_key_certificate block supports:

- format - (Required) The field allows only X509_CERTIFICATE_PEM.
- certificate - (Required) The certificate data.

## » Attributes Reference

Only the arguments listed above are exposed as attributes.

## » Import

A device registry can be imported using the name, e.g.

```
$ terraform import google_cloudiot_registry.default-registry projects/{project}/locations/{r
```