» google_active_folder

Get an active folder within GCP by `display_name` and `parent`.

» Example Usage

```
data "google_active_folder" "department1" {
  display_name = "Department 1"
  parent       = "organizations/1234567"
}
```

» Argument Reference

The following arguments are supported:

- `display_name` - (Required) The folder's display name.
- `parent` - (Required) The resource name of the parent Folder or Organization.

» Attributes Reference

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

- `name` - The resource name of the Folder. This uniquely identifies the folder.

» 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
}
```

» Example Usage: Configure Kubernetes provider with OAuth2 access token

```
data "google_client_config" "default" {
}

data "google_container_cluster" "my_cluster" {
```

```
  name = "my-cluster"
  zone = "us-east1-a"
}

provider "kubernetes" {
  load_config_file = false

  host  = "https://${data.google_container_cluster.my_cluster.endpoint}"
  token = data.google_client_config.default.access_token
  cluster_ca_certificate = base64decode(
    data.google_container_cluster.my_cluster.master_auth[0].cluster_ca_certificate,
  )
}
```

» 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.
- `zone` - The zone to operate under.
- `access_token` - The OAuth2 access token used by the client to authenticate against the Google Cloud API.

» google_client_openid_userinfo

Get OpenID userinfo about the credentials used with the Google provider, specifically the email.

This datasource enables you to export the email of the account you've authenticated the provider with; this can be used alongside `data.google_client_config`'s `access_token` to perform OpenID Connect authentication with GKE and configure an RBAC role for the email used.

This resource will only work as expected if the provider is configured to use the `https://www.googleapis.com/auth/userinfo.email` scope! You will receive an error otherwise.

» Example Usage - exporting an email

```
data "google_client_openid_userinfo" "me" {
}

output "my-email" {
  value = data.google_client_openid_userinfo.me.email
}
```

» Example Usage - OpenID Connect w/ Kubernetes provider + RBAC IAM role

```
data "google_client_openid_userinfo" "provider_identity" {
}

data "google_client_config" "provider" {
}

data "google_container_cluster" "my_cluster" {
  name = "my-cluster"
  zone = "us-east1-a"
}

provider "kubernetes" {
  load_config_file = false

  host  = "https://${data.google_container_cluster.my_cluster.endpoint}"
  token = data.google_client_config.provider.access_token
  cluster_ca_certificate = base64decode(
    data.google_container_cluster.my_cluster.master_auth[0].cluster_ca_certificate,
  )
}

resource "kubernetes_cluster_role_binding" "user" {
  metadata {
    name = "provider-user-admin"
  }

  role_ref {
    api_group = "rbac.authorization.k8s.io"
    kind      = "ClusterRole"
    name      = "cluster-admin"
  }
```

```
  subject {
    kind = "User"
    name = data.google_client_openid_userinfo.provider_identity.email
  }
}
```

» Argument Reference

There are no arguments available for this data source.

» Attributes Reference

The following attributes are exported:

- `email` - The email of the account used by the provider to authenticate with GCP.

» 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).
- `runtime` - The runtime in which the function is running.
- `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.
- `event_trigger` - A source that fires events in response to a condition in another service. Structure is documented below.
- `https_trigger_url` - If function is triggered by HTTP, trigger URL is set here.
- `labels` - A map of labels applied to this function.
- `service_account_email` - The service account email to be assumed by the cloud function.

The `event_trigger` block contains:

- `event_type` - The type of event being observed. For example: `"providers/cloud.storage/eventTypes/object.change"` and `"providers/cloud.pubsub/eventTypes/topic.publish"`. See the documentation on calling Cloud Functions for a full reference.
- `resource` - The name of the resource whose events are being observed, for example, `"myBucket"`
- `failure_policy` - Policy for failed executions. Structure is documented below.

The `failure_policy` block supports:

- `retry` - Whether the function should be retried on failure.

» google_composer_image_versions

Provides access to available Cloud Composer versions in a region for a given project.

» Example Usage

```
data "google_composer_image_versions" "all" {
}

resource "google_composer_environment" "test" {
  name   = "test-env"
  region = "us-central1"
  config {
    software_config {
      image_version = data.google_composer_image_versions.all.image_versions[0].image_versio
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The ID of the project to list versions in. If it is not provided, the provider project is used.
- `region` - (Optional) The location to list versions in. If it is not provider, the provider region is used.

» Attributes Reference

The following attributes are exported:

- `image_versions` - A list of composer image versions available in the given project and location. Each `image_version` contains:
  - `image_version_id` - The string identifier of the image version, in the form: "composer-x.y.z-airflow-a.b(.c)"
  - `supported_python_versions` - Supported python versions for this image version

» 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_backend_service

Provide access to a Backend Service's attribute. For more information see the official documentation and the API.

» Example Usage

```
data "google_compute_backend_service" "baz" {
  name = "foobar"
}

resource "google_compute_backend_service" "default" {
  name          = "backend-service"
  health_checks = [tolist(data.google_compute_backend_service.baz.health_checks)[0]]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Backend Service.

- `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 attributes are exported:

- `connection_draining_timeout_sec` - Time for which instance will be drained (not accept new connections, but still work to finish started ones).
- `description` - Textual description for the Backend Service.
- `enable_cdn` - Whether or not Cloud CDN is enabled on the Backend Service.
- `fingerprint` - The fingerprint of the Backend Service.
- `port_name` - The name of a service that has been added to an instance group in this backend.
- `protocol` - The protocol for incoming requests.
- `self_link` - The URI of the Backend Service.

- `session_affinity` - The Backend Service session stickiness configuration.
- `timeout_sec` - The number of seconds to wait for a backend to respond to a request before considering the request failed.
- `backend` - The set of backends that serve this Backend Service.
- `health_checks` - The set of HTTP/HTTPS health checks used by the Backend Service.

» 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 = data.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
- `unique_id` - The unique id of the service account.
- `name` - The fully-qualified name of the service account.
- `display_name` - The display name for the service account.

» 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_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" {
  family  = "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_instance

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

» Example Usage

```
data "google_compute_instance" "appserver" {
  name = "primary-application-server"
  zone = "us-central1-a"
}
```

» Argument Reference

The following arguments are supported:

- `self_link` - (Optional) The self link of the instance. One of `name` or `self_link` must be provided.
- `name` - (Optional) The name of the instance. One of `name` or `self_link` must be provided.

- `project` - (Optional) The ID of the project in which the resource belongs. If `self_link` is provided, this value is ignored. If neither `self_link` nor `project` are provided, the provider project is used.
- `zone` - (Optional) The zone of the instance. If `self_link` is provided, this value is ignored. If neither `self_link` nor `zone` are provided, the provider zone is used.

» Attributes Reference

- `boot_disk` - The boot disk for the instance. Structure is documented below.
- `machine_type` - The machine type to create.
- `network_interface` - The networks attached to the instance. Structure is documented below.
- `attached_disk` - List of disks attached to the instance. Structure is documented below.
- `can_ip_forward` - Whether sending and receiving of packets with non-matching source or destination IPs is allowed.
- `description` - A brief description of the resource.
- `deletion_protection` - Whether deletion protection is enabled on this instance.
- `guest_accelerator` - List of the type and count of accelerator cards attached to the instance. Structure is documented below.

- `labels` - A set of key/value label pairs assigned to the instance.
- `metadata` - Metadata key/value pairs made available within the instance.
- `min_cpu_platform` - The minimum CPU platform specified for the VM instance.
- `scheduling` - The scheduling strategy being used by the instance.
- `scratch_disk` - The scratch disks attached to the instance. Structure is documented below.
- `service_account` - The service account to attach to the instance. Structure is documented below.
- `tags` - The list of tags attached to the instance.
- `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.
- `shielded_instance_config` - The shielded vm config being used by the instance. Structure is documented below.
- `enable_display` -- Whether the instance has virtual displays enabled.
- `network_interface.0.network_ip` - The internal ip address of the instance, either manually or dynamically assigned.
- `network_interface.0.access_config.0.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.

The `boot_disk` block supports:

- `auto_delete` - Whether the disk will be auto-deleted when the instance is deleted.
- `device_name` - Name with which attached disk will be accessible under `/dev/disk/by-id/`
- `initialize_params` - Parameters with which a disk was created alongside the instance. Structure is documented below.
- `source` - The name or self_link of an existing disk (such as those managed by `google_compute_disk`) that was attached to the instance.

The `initialize_params` block supports:

- `size` - The size of the image in gigabytes.
- `type` - The GCE disk type. One of `pd-standard` or `pd-ssd`.
- `image` - The image from which this disk was initialised.

The `scratch_disk` block supports:

- `interface` - The disk interface used for attaching this disk. One of `SCSI` or `NVME`.

The `attached_disk` block supports:

- `source` - The name or self_link of the disk attached to this instance.
- `device_name` - Name with which the attached disk is accessible under `/dev/disk/by-id/`
- `mode` - Read/write mode for the disk. One of `"READ_ONLY"` or `"READ_WRITE"`.

The `network_interface` block supports:

- `network` - The name or self_link of the network attached to this interface.
- `subnetwork` - The name or self_link of the subnetwork attached to this interface.
- `subnetwork_project` - The project in which the subnetwork belongs.
- `network_ip` - The private IP address assigned to the instance.
- `access_config` - Access configurations, i.e. IPs via which this instance can be accessed via the Internet. Structure documented below.
- `alias_ip_range` - An array of alias IP ranges for this network interface. Structure documented below.

The `access_config` block supports:

- `nat_ip` - The IP address that is be 1:1 mapped to the instance's network ip.

- `public_ptr_domain_name` - The DNS domain name for the public PTR record.
- `network_tier` - The networking tier used for configuring this instance. One of `PREMIUM` or `STANDARD`.

The `alias_ip_range` block supports:

- `ip_cidr_range` - The IP CIDR range represented by this alias IP range.
- `subnetwork_range_name` - The subnetwork secondary range name specifying the secondary range from which to allocate the IP CIDR range for this alias IP range.

The `service_account` block supports:

- `email` - The service account e-mail address.
- `scopes` - A list of service scopes.

The `scheduling` block supports:

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

The `guest_accelerator` block supports:

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

The `shielded_instance_config` block supports:

- `enable_secure_boot` -- Whether secure boot is enabled for the instance.
- `enable_vtpm` -- Whether the instance uses vTPM.
- `enable_integrity_monitoring` -- Whether integrity monitoring is enabled for the instance.

» google_compute_instance_group

Get a Compute Instance Group within GCE. For more information, see the official documentation and API

```
data "google_compute_instance_group" "all" {
    name = "instance-group-name"
```

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

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the instance group. Either `name` or `self_link` must be provided.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `self_link` - (Optional) The self link of the instance group. Either `name` or `self_link` must be provided.
- `zone` - (Optional) The zone of the instance group. If referencing the instance group by name and `zone` is not provided, the provider zone is used.

» Attributes Reference

The following arguments are exported:

- `description` - Textual description of the instance group.
- `instances` - List of instances in the group.
- `named_port` - List of named ports in the group.
- `network` - The URL of the network the instance group is in.
- `self_link` - The URI of the resource.
- `size` - The number of instances in the group.

» google_compute_lb_ip_ranges

Use this data source to access IP ranges in your firewall rules.

https://cloud.google.com/compute/docs/load-balancing/health-checks#health_check_source_ips_and_firewall_rules

» Example Usage

```
data "google_compute_lb_ip_ranges" "ranges" {
}
```

```
resource "google_compute_firewall" "lb" {
  name    = "lb-firewall"
  network = google_compute_network.main.name

  allow {
    protocol = "tcp"
    ports    = ["80"]
  }

  source_ranges = data.google_compute_lb_ip_ranges.ranges.network
  target_tags = [
    "InstanceBehindLoadBalancer",
  ]
}
```

» 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:

- `network` - The IP ranges used for health checks when **Network load balancing** is used
- `http_ssl_tcp_internal` - The IP ranges used for health checks when **HTTP(S), SSL proxy, TCP proxy, and Internal load balancing** is used

» 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_compute_network_endpoint_group

Use this data source to access a Network Endpoint Group's attributes.

The NEG may be found by providing either a `self_link`, or a `name` and a `zone`.

» Example Usage

```
data "google_compute_network_endpoint_group" "neg1" {
  name = "k8s1-abcdef01-myns-mysvc-8080-4b6bac43"
  zone = "us-central1-a"
}

data "google_compute_network_endpoint_group" "neg2" {
  self_link = "https://www.googleapis.com/compute/v1/projects/myproject/zones/us-central1-a/
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The ID of the project to list versions in. If it is not provided, the provider project is used.
- `name` - (Optional) The Network Endpoint Group name. Provide either this or a `self_link`.
- `zone` - (Optional) The Network Endpoint Group availability zone.
- `self_link` - (Optional) The Network Endpoint Group self_link.

» Attributes Reference

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

- `network` - The network to which all network endpoints in the NEG belong.
- `subnetwork` - subnetwork to which all network endpoints in the NEG belong.
- `description` - The NEG description.
- `network_endpoint_type` - Type of network endpoints in this network endpoint group.
- `default_port` - The NEG default port.
- `size` - Number of network endpoints in the network endpoint group.

» google_compute_node_types

Provides available node types for Compute Engine sole-tenant nodes in a zone for a given project. For more information, see the official documentation and API.

» Example Usage

```
data "google_compute_node_types" "central1b" {
  zone = "us-central1-b"
}

resource "google_compute_node_template" "tmpl" {
  name      = "terraform-test-tmpl"
  region    = "us-central1"
  node_type = data.google_compute_node_types.types.names[0]
}
```

» Argument Reference

The following arguments are supported:

- `zone` (Optional) - The zone to list node types for. Should be in zone of intended node groups and region of referencing node template. If `zone` is not specified, the provider-level zone must be set and is used instead.
- `project` (Optional) - ID of the project to list available node types for. Should match the project the nodes of this type will be deployed to. Defaults to the project that the provider is authenticated with.

» Attributes Reference

The following attributes are exported:

- `names` - A list of node types available in the given zone and project.

» google_compute_regions

Provides access to available Google Compute regions for a given project. See more about regions and regions in the upstream docs.

```
data "google_compute_regions" "available" {
}

resource "google_compute_subnetwork" "cluster" {
  count         = length(data.google_compute_regions.available.names)
  name          = "my-network"
  ip_cidr_range = "10.36.${count.index}.0/24"
  network       = "my-network"
  region        = data.google_compute_regions.available.names[count.index]
}
```

» Argument Reference

The following arguments are supported:

- `project` (Optional) - Project from which to list available regions. Defaults to project declared in the provider.
- `status` (Optional) - Allows to filter list of regions based on their current status. Status can be either `UP` or `DOWN`. Defaults to no filtering (all available regions - both `UP` and `DOWN`).

» Attributes Reference

The following attribute is exported:

- `names` - A list of regions available in the given project

» google_compute_region_instance_group

Get a Compute Region Instance Group within GCE. For more information, see the official documentation and API.

```
data "google_compute_region_instance_group" "group" {
  name = "instance-group-name"
}
```

The most common use of this datasource will be to fetch information about the instances inside regional managed instance groups, for instance:

```
resource "google_compute_region_instance_group_manager" "foo" {
  name               = "some_name"
    ...
  base_instance_name = "foo"
    ...
  instance_template  = google_compute_instance_template.foo.self_link
  target_pools = [google_compute_target_pool.foo.self_link]
    ...
}

data "google_compute_region_instance_group" "data_source" {
  self_link = google_compute_region_instance_group_manager.foo.instance_group
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the instance group. One of `name` or `self_link` must be provided.
- `self_link` - (Optional) The link to the instance group. One of `name` or `self_link` must be provided.

- `project` - (Optional) The ID of the project in which the resource belongs. If `self_link` is provided, this value is ignored. If neither `self_link` nor `project` are provided, the provider project is used.

- `region` - (Optional) The region in which the resource belongs. If `self_link` is provided, this value is ignored. If neither `self_link` nor `region` are provided, the provider region is used.

» Attributes Reference

The following arguments are exported:

- `size` - The number of instances in the group.
- `instances` - List of instances in the group, as a list of resources, each containing:
  - `instance` - URL to the instance.
  - `named_ports` - List of named ports in the group, as a list of resources, each containing:
    - `port` - Integer port number
    - `name` - String port name
  - `status` - String description of current state of the instance.

» google_compute_resource_policy

Provide access to a Resource Policy's attributes. For more information see the official documentation or the API.

Warning: This datasource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

```
provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}

data "google_compute_resource_policy" "daily" {
  provider = google-beta
  name     = "daily"
  region   = "us-central1"
}
```

» Argument Reference

The following arguments are supported:

- `name` (Required) - The name of the Resource Policy.

- `project` (Optional) - Project from which to list the Resource Policy. Defaults to project declared in the provider.
- `region` (Required) - Region where the Resource Policy resides.

» Attributes Reference

The following attributes are exported:

- `description` - Description of this Resource Policy.
- `self_link` - The URI of the resource.

» google_compute_router

Get a router within GCE from its name and VPC.

» Example Usage

```
data "google_compute_router" "my-router" {
  name   = "myrouter-us-east1"
  network = "my-network"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the router.
- `network` - (Required) The VPC network on which this router lives.
- `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 router has been created in. If unspecified, this defaults to the region configured in the provider.

» Attributes Reference

See google_compute_router resource for details of the available attributes.

» google_compute_ssl_certificate

Get info about a Google Compute SSL Certificate from its name.

» Example Usage

```
data "google_compute_ssl_certificate" "my_cert" {
  name = "my-cert"
}

output "certificate" {
  value = data.google_compute_ssl_certificate.my_cert.certificate
}

output "certificate_id" {
  value = data.google_compute_ssl_certificate.my_cert.certificate_id
}

output "self_link" {
  value = data.google_compute_ssl_certificate.my_cert.self_link
}
```

» Argument Reference

The following arguments are supported:

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

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

» Attributes Reference

See google_compute_ssl_certificate resource for details of the available attributes.

» google_compute_ssl_policy

Gets an SSL Policy within GCE from its name, for use with Target HTTPS and Target SSL Proxies. For more information see the official documentation.

» Example Usage

```
data "google_compute_ssl_policy" "my-ssl-policy" {
  name = "production-ssl-policy"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the SSL Policy.

- `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:

- `enabled_features` - The set of enabled encryption ciphers as a result of the policy config
- `description` - Description of this SSL Policy.
- `min_tls_version` - The minimum supported TLS version of this policy.
- `profile` - The Google-curated or custom profile used by this policy.
- `custom_features` - If the `profile` is `CUSTOM`, these are the custom encryption ciphers supported by the profile. If the `profile` is *not* `CUSTOM`, this attribute will be empty.
- `fingerprint` - Fingerprint of this resource.
- `self_link` - The URI of the created resource.

» google_compute_subnetwork

Get a subnetwork within GCE from its name and region.

» Example Usage

```
data "google_compute_subnetwork" "my-subnetwork" {
  name   = "default-us-east1"
```

```
  region = "us-east1"
}
```

» Argument Reference

The following arguments are supported:

- `self_link` - (Optional) The self link of the subnetwork. If `self_link` is specified, `name`, `project`, and `region` are ignored.
- `name` - (Optional) The name of the subnetwork. One of `name` or `self_link` must be specified.
- `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 has been created in. If unspecified, this defaults to the region configured in the provider.

» 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 subnetwork.
- `description` - Description of this subnetwork.
- `ip_cidr_range` - The IP address range that machines in this network are assigned to, represented as a CIDR block.
- `gateway_address` - The IP address of the gateway.
- `private_ip_google_access` - Whether the VMs in this subnet can access Google services without assigned external IP addresses.
- `secondary_ip_range` - 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` - The name associated with this subnetwork secondary range, used when adding an alias IP range to a VM instance.
- `ip_cidr_range` - The range of IP addresses belonging to this subnetwork secondary range.

» google_compute_vpn_gateway

Get a VPN gateway within GCE from its name.

» Example Usage

```
data "google_compute_vpn_gateway" "my-vpn-gateway" {
  name = "vpn-gateway-us-east1"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the VPN gateway.

- `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:

- `network` - The network of this VPN gateway.
- `description` - Description of this VPN gateway.
- `region` - Region of this VPN gateway.
- `self_link` - The URI of the resource.

» google_compute_zones

Provides access to available Google Compute zones in a region for a given project. See more about regions and zones in the upstream docs.

```
data "google_compute_zones" "available" {
}

resource "google_compute_instance_group_manager" "foo" {
  count = length(data.google_compute_zones.available.names)
```

```
  name               = "terraform-test-${count.index}"
  instance_template  = google_compute_instance_template.foobar.self_link
  base_instance_name = "foobar-${count.index}"
  zone               = data.google_compute_zones.available.names[count.index]
  target_size        = 1
}
```

» Argument Reference

The following arguments are supported:

- `project` (Optional) - Project from which to list available zones. Defaults to project declared in the provider.
- `region` (Optional) - Region from which to list available zones. Defaults to region declared in the provider.
- `status` (Optional) - Allows to filter list of zones based on their current status. Status can be either `UP` or `DOWN`. Defaults to no filtering (all available zones - both `UP` and `DOWN`).

» Attributes Reference

The following attribute is exported:

- `names` - A list of zones available in the given region

» google_container_cluster

Get info about a GKE cluster from its name and location.

» Example Usage

```
data "google_container_cluster" "my_cluster" {
  name     = "my-cluster"
  location = "us-east1-a"
}

output "cluster_username" {
  value = data.google_container_cluster.my_cluster.master_auth[0].username
}

output "cluster_password" {
```

```
  value = data.google_container_cluster.my_cluster.master_auth[0].password
}

output "endpoint" {
  value = data.google_container_cluster.my_cluster.endpoint
}

output "instance_group_urls" {
  value = data.google_container_cluster.my_cluster.instance_group_urls
}

output "node_config" {
  value = data.google_container_cluster.my_cluster.node_config
}

output "node_pools" {
  value = data.google_container_cluster.my_cluster.node_pool
}
```

» Argument Reference

The following arguments are supported:

- `name` (Required) - The name of the cluster.
- `location` (Optional) - The location (zone or region) this cluster has been created in. One of `location`, `region`, `zone`, or a provider-level `zone` must be specified.
- `zone` (Optional) - The zone this cluster has been created in. Deprecated in favour of `location`.
- `region` (Optional) - The region this cluster has been created in. Deprecated in favour of `location`.

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

» Attributes Reference

See google_container_cluster resource for details of the available attributes.

» google_container_engine_versions

Provides access to available Google Kubernetes Engine versions in a zone or region for a given project.

If you are using the `google_container_engine_versions` datasource with a regional cluster, ensure that you have provided a region as the `location` to the datasource. A region can have a different set of supported versions than its component zones, and not all zones in a region are guaranteed to support the same version.

» Example Usage

```
data "google_container_engine_versions" "central1b" {
  location       = "us-central1-b"
  version_prefix = "1.12."
}

resource "google_container_cluster" "foo" {
  name               = "terraform-test-cluster"
  location           = "us-central1-b"
  node_version       = data.google_container_engine_versions.central1b.latest_node_version
  initial_node_count = 1

  master_auth {
    username = "mr.yoda"
    password = "adoy.rm"
  }
}
```

» Argument Reference

The following arguments are supported:

- `location` (Optional) - The location (region or zone) to list versions for. Must exactly match the location the cluster will be deployed in, or listed versions may not be available. If `location`, `region`, and `zone` are not specified, the provider-level zone must be set and is used instead.
- `project` (Optional) - ID of the project to list available cluster versions for. Should match the project the cluster will be deployed to. Defaults to the project that the provider is authenticated with.
- `version_prefix` (Optional) - If provided, Terraform will only return versions that match the string prefix. For example, `1.11.` will match all

`1.11` series releases. Since this is just a string match, it's recommended that you append a `.` after minor versions to ensure that prefixes such as `1.1` don't match versions like `1.12.5-gke.10` accidentally. See the docs on versioning schema for full details on how version strings are formatted.

» Attributes Reference

The following attributes are exported:

- `valid_master_versions` - A list of versions available in the given zone for use with master instances.
- `valid_node_versions` - A list of versions available in the given zone for use with node instances.
- `latest_master_version` - The latest version available in the given zone for use with master instances.
- `latest_node_version` - The latest version available in the given zone for use with node instances.
- `default_cluster_version` - Version of Kubernetes the service deploys by default.

» google_container_registry_image

This data source fetches the project name, and provides the appropriate URLs to use for container registry for this project.

The URLs are computed entirely offline - as long as the project exists, they will be valid, but this data source does not contact Google Container Registry (GCR) at any point.

» Example Usage

```
data "google_container_registry_image" "debian" {
  name = "debian"
}

output "gcr_location" {
  value = data.google_container_registry_image.debian.image_url
}
```

» Argument Reference

- `name`: (Required) The image name.

- `project`: (Optional) The project ID that this image is attached to. If not provider, provider project will be used instead.
- `region`: (Optional) The GCR region to use. As of this writing, one of `asia`, `eu`, and `us`. See the documentation for additional information.
- `tag`: (Optional) The tag to fetch, if any.
- `digest`: (Optional) The image digest to fetch, if any.

» Attributes Reference

In addition to the arguments listed above, this data source exports: * `image_url`: The URL at which the image can be accessed.

» google_container_registry_repository

This data source fetches the project name, and provides the appropriate URLs to use for container registry for this project.

The URLs are computed entirely offline - as long as the project exists, they will be valid, but this data source does not contact Google Container Registry (GCR) at any point.

» Example Usage

```
data "google_container_registry_repository" "foo" {
}

output "gcr_location" {
  value = data.google_container_registry_repository.foo.repository_url
}
```

» Argument Reference

- `project`: (Optional) The project ID that this repository is attached to. If not provided, provider project will be used instead.
- `region`: (Optional) The GCR region to use. As of this writing, one of `asia`, `eu`, and `us`. See the documentation for additional information.

» Attributes Reference

In addition to the arguments listed above, this data source exports:

- `repository_url`: The URL at which the repository can be accessed.

» google_dns_managed_zone

Provides access to a zone's attributes within Google Cloud DNS. For more information see the official documentation and API.

```
data "google_dns_managed_zone" "env_dns_zone" {
  name = "qa-zone"
}

resource "google_dns_record_set" "dns" {
  name = "my-address.${data.google_dns_managed_zone.env_dns_zone.dns_name}"
  type = "TXT"
  ttl  = 300

  managed_zone = data.google_dns_managed_zone.env_dns_zone.name

  rrdatas = ["test"]
}
```

» Argument Reference

- `name` - (Required) A unique name for the resource.
- `project` - (Optional) The ID of the project for the Google Cloud DNS zone.

» Attributes Reference

The following attributes are exported:

- `dns_name` - The fully qualified DNS name of this zone, e.g. `terraform.io.`.
- `description` - A textual description field.
- `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.
- `visibility` - The zone's visibility: public zones are exposed to the Internet, while private zones are visible only to Virtual Private Cloud resources.

» google_iam_policy

Generates an IAM policy document that may be referenced by and applied to other Google Cloud Platform resources, such as the `google_project` resource.

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/compute.instanceAdmin"

    members = [
      "serviceAccount:your-custom-sa@your-project.iam.gserviceaccount.com",
    ]
  }

  binding {
    role = "roles/storage.objectViewer"

    members = [
      "user:alice@gmail.com",
    ]
  }

  audit_config {
    service = "cloudkms.googleapis.com"
    audit_log_configs {
      log_type = "DATA_READ",
      exempted_members = ["user:you@domain.com"]
    }

    audit_log_configs {
      log_type = "DATA_WRITE",
    }

    audit_log_configs {
      log_type = "ADMIN_READ",
    }
  }
}
```

This data source is used to define IAM policies to apply to other resources. Currently, defining a policy through a datasource and referencing that policy from another resource is the only way to apply an IAM policy to a resource.

Note: Several restrictions apply when setting IAM policies through this API. See the setIamPolicy docs for a list of these restrictions.

» Argument Reference

The following arguments are supported:

- `binding` (Required) - A nested configuration block (described below) defin-

ing a binding to be included in the policy document. Multiple `binding` arguments are supported.

Each document configuration must have one or more `binding` blocks, which each accept the following arguments:

- `role` (Required) - The role/permission that will be granted to the members. See the IAM Roles documentation for a complete list of roles. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.
- `members` (Required) - An array of identities that will be granted the privilege in the `role`. For more details on format and restrictions see https://cloud.google.com/billing/reference/rest/v1/Policy#Binding 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. It **can't** be used with the `google_project` resource.
  - **allAuthenticatedUsers**: A special identifier that represents anyone who is authenticated with a Google account or a service account. It **can't** be used with the `google_project` resource.
  - **user:{emailid}**: An email address that represents a specific Google account. For example, alice@gmail.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 G Suite domain (primary, instead of alias) name that represents all the users of that domain. For example, google.com or example.com.
- `audit_config` (Optional) - A nested configuration block that defines logging additional configuration for your project.
  - `service` (Required) Defines a service that will be enabled for audit logging. For example, `storage.googleapis.com`, `cloudsql.googleapis.com`. `allServices` is a special value that covers all services.
  - `audit_log_configs` (Required) A nested block that defines the operations you'd like to log.
  - `log_type` (Required) Defines the logging level. `DATA_READ`, `DATA_WRITE` and `ADMIN_READ` capture different types of events. See the audit configuration documentation for more details.
  - `exempted_members` (Optional) Specifies the identities that are exempt from these types of logging operations. Follows the same format of the `members` array for `binding`.

» Attributes Reference

The following attribute is exported:

- `policy_data` - The above bindings serialized in a format suitable for referencing from a resource that supports IAM.

» google_iam_role

Use this data source to get information about a Google IAM Role.

```
data "google_iam_role" "roleinfo" {
  name = "roles/compute.viewer"
}

output "the_role_permissions" {
  value = data.google_iam_role.roleinfo.included_permissions
}
```

» Argument Reference

The following arguments are supported:

- `name` (Required) - The name of the Role to lookup in the form `roles/{ROLE_NAME}`, `organizations/{ORGANIZATION_ID}/roles/{ROLE_NAME}` or `projects/{PROJECT_ID}/roles/{ROLE_NAME}`

» Attributes Reference

The following attributes are exported:

- `title` - is a friendly title for the role, such as "Role Viewer"
- `included_permissions` - specifies the list of one or more permissions to include in the custom role, such as - `iam.roles.get`
- `stage` - indicates the stage of a role in the launch lifecycle, such as `GA`, `BETA` or `ALPHA`.

» google_kms_crypto_key

Provides access to 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.

» Example Usage

```
data "google_kms_key_ring" "my_key_ring" {
  name     = "my-key-ring"
  location = "us-central1"
}

data "google_kms_crypto_key" "my_crypto_key" {
  name     = "my-crypto-key"
  key_ring = data.google_kms_key_ring.my_key_ring.self_link
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The CryptoKey's name. A CryptoKey's name belonging to the specified Google Cloud Platform KeyRing and match the regular expression `[a-zA-Z0-9_-]{1,63}`
- `key_ring` - (Required) The `self_link` of the Google Cloud Platform KeyRing to which the key belongs.

» Attributes Reference

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

- `rotation_period` - 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).
- `purpose` - Defines the cryptographic capabilities of the key.
- `self_link` - The self link of the created CryptoKey. Its format is `projects/{projectId}/locations/{location}/keyRings/{keyRingName}/cryptoKeys/{cryptoKeyN`

» google_kms_crypto_key_version

Provides access to a Google Cloud Platform KMS CryptoKeyVersion. For more information see the official documentation and API.

A CryptoKeyVersion represents an individual cryptographic key, and the associated key material.

» Example Usage

```
data "google_kms_key_ring" "my_key_ring" {
  name     = "my-key-ring"
  location = "us-central1"
}

data "google_kms_crypto_key" "my_crypto_key" {
  name     = "my-crypto-key"
  key_ring = data.google_kms_key_ring.my_key_ring.self_link
}

data "google_kms_crypto_key_version" "my_crypto_key_version" {
  crypto_key = data.google_kms_key.my_key.self_link
}
```

» Argument Reference

The following arguments are supported:

- `crypto_key` - (Required) The `self_link` of the Google Cloud Platform CryptoKey to which the key version belongs.
- `version` - (Optional) The version number for this CryptoKeyVersion. Defaults to `1`.

» Attributes Reference

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

- `state` - The current state of the CryptoKeyVersion. See the state reference for possible outputs.
- `protection_level` - The ProtectionLevel describing how crypto operations are performed with this CryptoKeyVersion. See the protection_level reference for possible outputs.

- `algorithm` - The CryptoKeyVersionAlgorithm that this CryptoKeyVersion supports. See the algorithm reference for possible outputs.
- `public_key` - If the enclosing CryptoKey has purpose `ASYMMETRIC_SIGN` or `ASYMMETRIC_DECRYPT`, this block contains details about the public key associated to this CryptoKeyVersion. Structure is documented below.

The `public_key` block, if present, contains:

- `pem` - The public key, encoded in PEM format. For more information, see the RFC 7468 sections for General Considerations and Textual Encoding of Subject Public Key Info.
- `algorithm` - The CryptoKeyVersionAlgorithm that this CryptoKeyVersion supports.

» google_kms_key_ring

Provides access to 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.

» Example Usage

```
data "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 name must exist within the provided 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:

- `self_link` - The self link of the created KeyRing. Its format is `projects/{projectId}/locations/{location}/keyRings/{keyRingName}`.

» google_kms_secret

This data source allows you to use data encrypted with Google Cloud KMS within your resource definitions.

For more information see the official documentation.

NOTE: Using this data provider will allow you to conceal secret data within your resource definitions, but it does not take care of protecting that data in the logging output, plan output, or state output. Please take care to secure your secret data outside of resource definitions.

» Example Usage

First, create a KMS KeyRing and CryptoKey using the resource definitions:

```
resource "google_kms_key_ring" "my_key_ring" {
  project  = "my-project"
  name     = "my-key-ring"
  location = "us-central1"
}

resource "google_kms_crypto_key" "my_crypto_key" {
  name     = "my-crypto-key"
  key_ring = google_kms_key_ring.my_key_ring.self_link
}
```

Next, use the Cloud SDK to encrypt some sensitive information:

```
$ echo -n my-secret-password | gcloud kms encrypt \
> --project my-project \
> --location us-central1 \
> --keyring my-key-ring \
> --key my-crypto-key \
> --plaintext-file - \
> --ciphertext-file - \
> | base64
CiQAqD+xX4SXOSziF4a8JYvq4spfAuWhhYSNul33H85HnVtNQW4SOgDu2UZ46dQCRFl5MF6ekabviN8xq+F+2035ZJ85
```

Finally, reference the encrypted ciphertext in your resource definitions:

```
data "google_kms_secret" "sql_user_password" {
  crypto_key = google_kms_crypto_key.my_crypto_key.self_link
  ciphertext = "CiQAqD+xX4SXOSziF4a8JYvq4spfAuWhhYSNul33H85HnVtNQW4SOgDu2UZ46dQCRFl5MF6ekabv
}

resource "random_id" "db_name_suffix" {
  byte_length = 4
}

resource "google_sql_database_instance" "master" {
  name = "master-instance-${random_id.db_name_suffix.hex}"

  settings {
    tier = "D0"
  }
}

resource "google_sql_user" "users" {
  name     = "me"
  instance = google_sql_database_instance.master.name
  host     = "me.com"
  password = data.google_kms_secret.sql_user_password.plaintext
}
```

This will result in a Cloud SQL user being created with password `my-secret-password`.

» Argument Reference

The following arguments are supported:

- `ciphertext` (Required) - The ciphertext to be decrypted, encoded in base64
- `crypto_key` (Required) - The id of the CryptoKey that will be used to decrypt the provided ciphertext. This is represented by the format `{projectId}/{location}/{keyRingName}/{cryptoKeyName}`.

» Attributes Reference

The following attribute is exported:

- `plaintext` - Contains the result of decrypting the provided ciphertext.

» google_kms_secret_ciphertext

Warning: This data source is deprecated. Use the `google_kms_secret_ciphertext` **resource** instead.

This data source allows you to encrypt data with Google Cloud KMS and use the ciphertext within your resource definitions.

For more information see the official documentation.

NOTE: Using this data source will allow you to conceal secret data within your resource definitions, but it does not take care of protecting that data in the logging output, plan output, or state output. Please take care to secure your secret data outside of resource definitions.

» Example Usage

First, create a KMS KeyRing and CryptoKey using the resource definitions:

```
resource "google_kms_key_ring" "my_key_ring" {
  project  = "my-project"
  name     = "my-key-ring"
  location = "us-central1"
}

resource "google_kms_crypto_key" "my_crypto_key" {
  name     = "my-crypto-key"
  key_ring = google_kms_key_ring.my_key_ring.self_link
}
```

Next, encrypt some sensitive information and use the encrypted data in your resource definitions:

```
data "google_kms_secret_ciphertext" "my_password" {
  crypto_key = google_kms_crypto_key.my_crypto_key.self_link
  plaintext  = "my-secret-password"
}

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

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-9"
    }
```

```
  }

  network_interface {
    network = "default"

    access_config {
    }
  }

  metadata = {
    password = data.google_kms_secret_ciphertext.my_password.ciphertext
  }
}
```

The resulting instance can then access the encrypted password from its metadata and decrypt it, e.g. using the Cloud SDK):

```
$ curl -H "Metadata-Flavor: Google" http://metadata.google.internal/computeMetadata/v1/insta
> | base64 -d | gcloud kms decrypt \
> --project my-project \
> --location us-central1 \
> --keyring my-key-ring \
> --key my-crypto-key \
> --plaintext-file - \
> --ciphertext-file - \
my-secret-password
```

» Argument Reference

The following arguments are supported:

- `plaintext` (Required) - The plaintext to be encrypted
- `crypto_key` (Required) - The id of the CryptoKey that will be used to encrypt the provided plaintext. This is represented by the format `{projectId}/{location}/{keyRingName}/{cryptoKeyName}`.

» Attributes Reference

The following attribute is exported:

- `ciphertext` - Contains the result of encrypting the provided plaintext, encoded in base64.

» User Project Overrides

This data source supports User Project Overrides.

» google_folder

Use this data source to get information about a Google Cloud Folder.

```
# Get folder by id
data "google_folder" "my_folder_1" {
  folder              = "folders/12345"
  lookup_organization = true
}

# Search by fields
data "google_folder" "my_folder_2" {
  folder = "folders/23456"
}

output "my_folder_1_organization" {
  value = data.google_folder.my_folder_1.organization
}

output "my_folder_2_parent" {
  value = data.google_folder.my_folder_2.parent
}
```

» Argument Reference

The following arguments are supported:

- `folder` (Required) - The name of the Folder in the form `{folder_id}` or `folders/{folder_id}`.
- `lookup_organization` (Optional) - `true` to find the organization that the folder belongs, `false` to avoid the lookup. It searches up the tree. (defaults to `false`)

» Attributes Reference

The following attributes are exported:

- `id` - The Folder ID.
- `name` - The resource name of the Folder in the form `folders/{folder_id}`.
- `parent` - The resource name of the parent Folder or Organization.

- `display_name` - The folder's display name.
- `create_time` - Timestamp when the Organization was created. A timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds. Example: "2014-10-02T15:01:23.045123456Z".
- `lifecycle_state` - The Folder's current lifecycle state.
- `organization` - If `lookup_organization` is enable, the resource name of the Organization that the folder belongs.

» google_folder_organization_policy

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

» Example Usage

```
data "google_folder_organization_policy" "policy" {
  folder     = "folders/folderid"
  constraint = "constraints/compute.trustedImageProjects"
}

output "version" {
  value = data.google_folder_organization_policy.policy.version
}
```

» 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) (Required) The name of the Constraint the Policy is configuring, for example, `serviceuser.services`. Check out the complete list of available constraints.

» Attributes Reference

See google_folder_organization_policy resource for details of the available attributes.

» google_netblock_ip_ranges

Use this data source to get the IP addresses from different special IP ranges on Google Cloud Platform.

» Example Usage - Cloud Ranges

```
data "google_netblock_ip_ranges" "netblock" {
}

output "cidr_blocks" {
  value = data.google_netblock_ip_ranges.netblock.cidr_blocks
}

output "cidr_blocks_ipv4" {
  value = data.google_netblock_ip_ranges.netblock.cidr_blocks_ipv4
}

output "cidr_blocks_ipv6" {
  value = data.google_netblock_ip_ranges.netblock.cidr_blocks_ipv6
}
```

» Example Usage - Allow Health Checks

```
data "google_netblock_ip_ranges" "legacy-hcs" {
  range_type = "legacy-health-checkers"
}

resource "google_compute_firewall" "allow-hcs" {
  name    = "allow-hcs"
  network = google_compute_network.default.name

  allow {
    protocol = "tcp"
    ports    = ["80"]
  }

  source_ranges = data.google_netblock_ip_ranges.legacy-hcs.cidr_blocks_ipv4
}

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

» Argument Reference

The following arguments are supported:

- `range_type` (Optional) - The type of range for which to provide results.

Defaults to `cloud-netblocks`. The following `range_type`s are supported:

- `cloud-netblocks` - Corresponds to the IP addresses used for resources on Google Cloud Platform. More details.
- `google-netblocks` - Corresponds to IP addresses used for Google services. More details.
- `restricted-googleapis` - Corresponds to the IP addresses used for Private Google Access only for services that support VPC Service Controls API access. More details.
- `private-googleapis` - Corresponds to the IP addresses used for Private Google Access for services that do not support VPC Service Controls. More details.
- `dns-forwarders` - Corresponds to the IP addresses used to originate Cloud DNS outbound forwarding. More details.
- `iap-forwarders` - Corresponds to the IP addresses used for Cloud IAP for TCP forwarding. More details.
- `health-checkers` - Corresponds to the IP addresses used for health checking in Cloud Load Balancing. More details.
- `legacy-health-checkers` - Corresponds to the IP addresses used for legacy style health checkers (used by Network Load Balancing). More details.

» Attributes Reference

- `cidr_blocks` - Retrieve list of all CIDR blocks.
- `cidr_blocks_ipv4` - Retrieve list of the IPv4 CIDR blocks
- `cidr_blocks_ipv6` - Retrieve list of the IPv6 CIDR blocks, if available.

» google_organization

Use this data source to get information about a Google Cloud Organization.

```
data "google_organization" "org" {
  domain = "example.com"
}
```

```
resource "google_folder" "sales" {
  display_name = "Sales"
  parent       = data.google_organization.org.name
}
```

» Argument Reference

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

- `organization` (Optional) - The name of the Organization in the form `{organization_id}` or `organizations/{organization_id}`.
- `domain` (Optional) - The domain name of the Organization.

NOTE: One of `organization` or `domain` must be specified.

» Attributes Reference

The following additional attributes are exported:

- `org_id` - The Organization ID.
- `name` - The resource name of the Organization in the form `organizations/{organization_id}`.
- `directory_customer_id` - The Google for Work customer ID of the Organization.
- `create_time` - Timestamp when the Organization was created. A timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds. Example: "2014-10-02T15:01:23.045123456Z".
- `lifecycle_state` - The Organization's current lifecycle state.

» google_project

Use this data source to get project details. For more information see API

» Example Usage

```
data "google_project" "project" {
}

output "project_number" {
  value = data.google_project.project.number
}
```

» Argument Reference

The following arguments are supported:

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

» Attributes Reference

The following attributes are exported:

See google_project resource for details of the available attributes.

» google_projects

Retrieve information about a set of projects based on a filter. See the REST API for more details.

» Example Usage - searching for projects about to be deleted in an org

```
data "google_projects" "my-org-projects" {
  filter = "parent.id:012345678910 lifecycleState:DELETE_REQUESTED"
}

data "google_project" "deletion-candidate" {
  project_id = data.google_projects.my-org-projects.projects[0].project_id
}
```

» Argument Reference

The following arguments are supported:

- `filter` - (Optional) A string filter as defined in the REST API.

» Attributes Reference

The following attributes are exported:

- `projects` - A list of projects matching the provided filter. Structure is defined below.

The `projects` block supports:

- `project_id` - The project id of the project.

» google_project_organization_policy

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

» Example Usage

```
data "google_project_organization_policy" "policy" {
  project    = "project-id"
  constraint = "constraints/serviceuser.services"
}

output "version" {
  value = data.google_project_organization_policy.policy.version
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Required) The project ID.
- `constraint` - (Required) (Required) The name of the Constraint the Policy is configuring, for example, `serviceuser.services`. Check out the complete list of available constraints.

» Attributes Reference

See google_project_organization_policy resource for details of the available attributes.

» google_service_account

Get the service account from a project. For more information see the official API documentation.

» Example Usage

```
data "google_service_account" "object_viewer" {
  account_id = "object-viewer"
}
```

» Example Usage, save key in Kubernetes secret

```
data "google_service_account" "myaccount" {
  account_id = "myaccount-id"
}

resource "google_service_account_key" "mykey" {
  service_account_id = data.google_service_account.myaccount.name
}

resource "kubernetes_secret" "google-application-credentials" {
  metadata {
    name = "google-application-credentials"
  }
  data = {
    credentials.json = base64decode(google_service_account_key.mykey.private_key)
  }
}
```

» Argument Reference

The following arguments are supported:

- `account_id` - (Required) The Service account id. (This is the part of the service account's email field that comes before the @ symbol.)
- `project` - (Optional) The ID of the project that the service account is present in. Defaults to the provider project configuration.

» 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.
- `unique_id` - The unique id of the service account.

- `name` - The fully-qualified name of the service account.
- `display_name` - The display name for the service account.

» google_service_account_access_token

This data source provides a google `oauth2` `access_token` for a different service account than the one initially running the script.

For more information see the official documentation as well as iamcredentials.generateAccessToken()

» Example Usage

To allow `service_A` to impersonate `service_B`, grant the Service Account Token Creator on B to A.

In the IAM policy below, `service_A` is given the Token Creator role impersonate `service_B`

```
resource "google_service_account_iam_binding" "token-creator-iam" {
    service_account_id = "projects/-/serviceAccounts/service_B@projectB.iam.gserviceaccount.
    role               = "roles/iam.serviceAccountTokenCreator"
    members = [
        "serviceAccount:service_A@projectA.iam.gserviceaccount.com",
    ]
}
```

Once the IAM permissions are set, you can apply the new token to a provider bootstrapped with it. Any resources that references the aliased provider will run as the new identity.

In the example below, `google_project` will run as `service_B`.

```
provider "google" {
}

data "google_client_config" "default" {
  provider = google
}

data "google_service_account_access_token" "default" {
  provider               = google
  target_service_account = "service_B@projectB.iam.gserviceaccount.com"
  scopes                 = ["userinfo-email", "cloud-platform"]
  lifetime               = "300s"
}
```

```
provider "google" {
  alias        = "impersonated"
  access_token = data.google_service_account_access_token.default.access_token
}

data "google_client_openid_userinfo" "me" {
  provider = google.impersonated
}

output "target-email" {
  value = data.google_client_openid_userinfo.me.email
}
```

> *Note*: the generated token is non-refreshable and can have a maximum `lifetime` of `3600` seconds.

» Argument Reference

The following arguments are supported:

- `target_service_account` (Required) - The service account *to* impersonate (e.g. `service_B@your-project-id.iam.gserviceaccount.com`)
- `scopes` (Required) - The scopes the new credential should have (e.g. `["storage-ro", "cloud-platform"]`)
- `delegates` (Optional) - Delegate chain of approvals needed to perform full impersonation. Specify the fully qualified service account name. (e.g. `["projects/-/serviceAccounts/delegate-svc-account@project-id.iam.gserviceaccount.com"]`)
- `lifetime` (Optional) Lifetime of the impersonated token (defaults to its max: `3600s`).

» Attributes Reference

The following attribute is exported:

- `access_token` - The `access_token` representing the new generated identity.

» google_service_account_key

Get service account public key. For more information, see the official documentation and API.

» Example Usage

```
resource "google_service_account" "myaccount" {
  account_id = "dev-foo-account"
}

resource "google_service_account_key" "mykey" {
  service_account_id = google_service_account.myaccount.name
}

data "google_service_account_key" "mykey" {
  name            = google_service_account_key.mykey.name
  public_key_type = "TYPE_X509_PEM_FILE"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the service account key. This must have format `projects/{PROJECT_ID}/serviceAccounts/{ACCOUNT}/keys/{KEYID}`, where `{ACCOUNT}` is the email address or unique id of the service account.
- `project` - (Optional) The ID of the project that the service account will be created in. Defaults to the provider project configuration.
- `public_key_type` (Optional) The output format of the public key requested. X509_PEM is the default output format.

» Attributes Reference

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

- `public_key` - The public key, base64 encoded

» google_storage_bucket_object

Gets an existing object inside an existing bucket in Google Cloud Storage service (GCS). See the official documentation and API.

» Example Usage

Example picture stored within a folder.

```
data "google_storage_bucket_object" "picture" {
  name   = "folder/butterfly01.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.

» Attributes Reference

The following attributes are exported:

- `cache_control` - (Computed) 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` - (Computed) Content-Disposition of the object data.
- `content_encoding` - (Computed) Content-Encoding of the object data.
- `content_language` - (Computed) Content-Language of the object data.
- `content_type` - (Computed) Content-Type of the object data. Defaults to "application/octet-stream" or "text/plain; charset=utf-8".
- `crc32c` - (Computed) Base 64 CRC32 hash of the uploaded data.
- `md5hash` - (Computed) Base 64 MD5 hash of the uploaded data.
- `self_link` - (Computed) A url reference to this object.
- `storage_class` - (Computed) 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.

» google_storage_object_signed_url

The Google Cloud storage signed URL data source generates a signed URL for a given storage object. Signed URLs provide a way to give time-limited read or write access to anyone in possession of the URL, regardless of whether they have a Google account.

For more info about signed URL's is available here.

» Example Usage

```
data "google_storage_object_signed_url" "artifact" {
  bucket = "install_binaries"
  path   = "path/to/install_file.bin"
}

resource "google_compute_instance" "vm" {
  name = "vm"

  provisioner "remote-exec" {
    inline = [
      "wget '${data.google_storage_object_signed_url.artifact.signed_url}' -O install_file.b
      "chmod +x install_file.bin",
      "./install_file.bin",
    ]
  }
}
```

» Full Example

```
data "google_storage_object_signed_url" "get_url" {
  bucket       = "fried_chicken"
  path         = "path/to/file"
  content_md5  = "pRviqwS4c4OTJRTe03FD1w=="
  content_type = "text/plain"
  duration     = "2d"
  credentials  = file("path/to/credentials.json")

  extension_headers = {
    x-goog-if-generation-match = 1
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket to read the object from
- `path` - (Required) The full path to the object inside the bucket

- `http_method` - (Optional) What HTTP Method will the signed URL allow (defaults to `GET`)
- `duration` - (Optional) For how long shall the signed URL be valid (defaults to 1 hour - i.e. `1h`). See here for info on valid duration formats.
- `credentials` - (Optional) What Google service account credentials json should be used to sign the URL. This data source checks the following locations for credentials, in order of preference: data source `credentials` attribute, provider `credentials` attribute and finally the GOOGLE_APPLICATION_CREDENTIALS environment variable.

  > **NOTE** the default google credentials configured by `gcloud` sdk or the service account associated with a compute instance cannot be used, because these do not include the private key required to sign the URL. A valid `json` service account credentials key file must be used, as generated via Google cloud console.

- `content_type` - (Optional) If you specify this in the datasource, the client must provide the `Content-Type` HTTP header with the same value in its request.
- `content_md5` - (Optional) The MD5 digest value in Base64. Typically retrieved from `google_storage_bucket_object.object.md5hash` attribute. If you provide this in the datasource, the client (e.g. browser, curl) must provide the `Content-MD5` HTTP header with this same value in its request.
- `extension_headers` - (Optional) As needed. The server checks to make sure that the client provides matching values in requests using the signed URL. Any header starting with `x-goog-` is accepted but see the Google Docs for list of headers that are supported by Google.

» Attributes Reference

The following attributes are exported:

- `signed_url` - The signed URL that can be used to access the storage object without authentication.

» google_storage_project_service_account

Get the email address of a project's unique Google Cloud Storage service account.

Each Google Cloud project has a unique service account for use with Google Cloud Storage. Only this special service account can be used to set up `google_storage_notification` resources.

For more information see the API reference.

» Example Usage

```
data "google_storage_project_service_account" "gcs_account" {
}

resource "google_pubsub_topic_iam_binding" "binding" {
  topic = google_pubsub_topic.topic.name
  role  = "roles/pubsub.publisher"

  members = ["serviceAccount:${data.google_storage_project_service_account.gcs_account.email
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The project the unique service account was created for. If it is not provided, the provider project is used.
- `user_project` - (Optional) The project the lookup originates from. This field is used if you are making the request from a different account than the one you are finding the service account for.

» Attributes Reference

The following attributes are exported:

- `email_address` - The email address of the service account. This value is often used to refer to the service account in order to grant IAM permissions.

» google_storage_transfer_project_service_account

Use this data source to retrieve Storage Transfer service account for this project

» Example Usage

```
data "google_storage_transfer_project_service_account" "default" {
}

output "default_account" {
  value = data.google_storage_transfer_project_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 Storage Transfer Jobs running in this project

» google_tpu_tensorflow_versions

Get TensorFlow versions available for a project. For more information see the official documentation and API.

» Example Usage

```
data "google_tpu_tensorflow_versions" "available" {
}
```

» Example Usage: Configure Basic TPU Node with available version

```
data "google_tpu_tensorflow_versions" "available" {
}

resource "google_tpu_node" "tpu" {
  name = "test-tpu"
```

```
  zone = "us-central1-b"

  accelerator_type   = "v3-8"
  tensorflow_version = data.google_tpu_tensorflow_versions.available.versions[0]
  cidr_block         = "10.2.0.0/29"
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The project to list versions for. If it is not provided, the provider project is used.
- `zone` - (Optional) The zone to list versions for. If it is not provided, the provider zone is used.

» Attributes Reference

The following attributes are exported:

- `versions` - The list of TensorFlow versions available for the given project and zone.

» google_access_context_manager_access_level

An AccessLevel is a label that can be applied to requests to GCP services, along with a list of requirements necessary for the label to be applied.

To get more information about AccessLevel, see:

- API documentation
- How-to Guides
  - Access Policy Quickstart

» Example Usage - Access Context Manager Access Level Basic

```
resource "google_access_context_manager_access_level" "access-level" {
  parent = "accessPolicies/${google_access_context_manager_access_policy.test-access.name}"
  name   = "accessPolicies/${google_access_context_manager_access_policy.test-access.name}/a
  title  = "chromeos_no_lock"
  basic {
    conditions {
```

```
      device_policy {
        require_screen_lock = false
        os_constraints {
          os_type = "DESKTOP_CHROME_OS"
        }
      }
    }
  }
}

resource "google_access_context_manager_access_policy" "access-policy" {
  parent = "organizations/123456789"
  title  = "my policy"
}
```

» Argument Reference

The following arguments are supported:

- `title` - (Required) Human readable title. Must be unique within the Policy.
- `parent` - (Required) The AccessPolicy this AccessLevel lives in. Format: accessPolicies/{policy_id}
- `name` - (Required) Resource name for the Access Level. The short_name component must begin with a letter and only include alphanumeric and '_'. Format: accessPolicies/{policy_id}/accessLevels/{short_name}

- `description` - (Optional) Description of the AccessLevel and its use. Does not affect behavior.
- `basic` - (Optional) A set of predefined conditions for the access level and a combining function. Structure is documented below.

The `basic` block supports:

- `combining_function` - (Optional) How the conditions list should be combined to determine if a request is granted this AccessLevel. If AND is used, each Condition in conditions must be satisfied for the AccessLevel to be applied. If OR is used, at least one Condition in conditions must be satisfied for the AccessLevel to be applied. Defaults to AND if unspecified.
- `conditions` - (Required) A set of requirements for the AccessLevel to be granted. Structure is documented below.

The `conditions` block supports:

- `ip_subnetworks` - (Optional) A list of CIDR block IP subnetwork specification. May be IPv4 or IPv6. Note that for a CIDR IP address block, the specified IP address portion must be properly truncated (i.e. all the host bits must be zero) or the input is considered malformed. For example, "192.0.2.0/24" is accepted but "192.0.2.1/24" is not. Similarly, for IPv6, "2001:db8::/32" is accepted whereas "2001:db8::1/32" is not. The originating IP of a request must be in one of the listed subnets in order for this Condition to be true. If empty, all IP addresses are allowed.
- `required_access_levels` - (Optional) A list of other access levels defined in the same Policy, referenced by resource name. Referencing an AccessLevel which does not exist is an error. All access levels listed must be granted for the Condition to be true. Format: accessPolicies/{policy_id}/accessLevels/{short_name}
- `members` - (Optional) An allowed list of members (users, service accounts). Using groups is not supported yet. The signed-in user originating the request must be a part of one of the provided members. If not specified, a request may come from any user (logged in/not logged in, not present in any groups, etc.). Formats: `user:{emailid}`, `serviceAccount:{emailid}`
- `negate` - (Optional) Whether to negate the Condition. If true, the Condition becomes a NAND over its non-empty fields, each field must be false for the Condition overall to be satisfied. Defaults to false.
- `device_policy` - (Optional) Device specific restrictions, all restrictions must hold for the Condition to be true. If not specified, all devices are allowed. Structure is documented below.

The `device_policy` block supports:

- `require_screen_lock` - (Optional) Whether or not screenlock is required for the DevicePolicy to be true. Defaults to false.
- `allowed_encryption_statuses` - (Optional) A list of allowed encryptions statuses. An empty list allows all statuses.
- `allowed_device_management_levels` - (Optional) A list of allowed device management levels. An empty list allows all management levels.
- `os_constraints` - (Optional) A list of allowed OS versions. An empty list allows all types and all versions. Structure is documented below.
- `require_admin_approval` - (Optional) Whether the device needs to be approved by the customer admin.
- `require_corp_owned` - (Optional) Whether the device needs to be corp owned.

The `os_constraints` block supports:

- `minimum_version` - (Optional) The minimum allowed OS version. If not set, any version of this OS satisfies the constraint. Format: "major.minor.patch" such as "10.5.301", "9.2.1".
- `os_type` - (Required) The operating system type of the device.

» 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

AccessLevel can be imported using any of these accepted formats:

```
$ terraform import google_access_context_manager_access_level.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» google_access_context_manager_access_policy

AccessPolicy is a container for AccessLevels (which define the necessary attributes to use GCP services) and ServicePerimeters (which define regions of services able to freely pass data within a perimeter). An access policy is globally visible within an organization, and the restrictions it specifies apply to all projects within an organization.

To get more information about AccessPolicy, see:

- API documentation
- How-to Guides
  - Access Policy Quickstart

» Example Usage - Access Context Manager Access Policy Basic

```
resource "google_access_context_manager_access_policy" "access-policy" {
  parent = "organizations/123456789"
  title  = "my policy"
```

```
}
```

» Argument Reference

The following arguments are supported:

- `parent` - (Required) The parent of this AccessPolicy in the Cloud Resource Hierarchy. Format: organizations/{organization_id}
- `title` - (Required) Human readable title. Does not affect behavior.

» Attributes Reference

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

- `name` - Resource name of the AccessPolicy. Format: {policy_id}
- `create_time` - Time the AccessPolicy was created in UTC.
- `update_time` - Time the AccessPolicy was updated in UTC.

» 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

AccessPolicy can be imported using any of these accepted formats:

```
$ terraform import google_access_context_manager_access_policy.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» google_access_context_manager_service_perimeter

ServicePerimeter describes a set of GCP resources which can freely import and export data amongst themselves, but not export outside of the ServicePerimeter. If a request with a source within this ServicePerimeter has a target outside of the ServicePerimeter, the request will be blocked. Otherwise the request is allowed. There are two types of Service Perimeter - Regular and Bridge. Regular Service Perimeters cannot overlap, a single GCP project can only belong to a single regular Service Perimeter. Service Perimeter Bridges can contain only GCP projects as members, a single GCP project may belong to multiple Service Perimeter Bridges.

To get more information about ServicePerimeter, see:

- API documentation
- How-to Guides
  - Service Perimeter Quickstart

» Example Usage - Access Context Manager Service Perimeter Basic

```
resource "google_access_context_manager_service_perimeter" "service-perimeter" {
  parent = "accessPolicies/${google_access_context_manager_access_policy.access-policy.name}
  name   = "accessPolicies/${google_access_context_manager_access_policy.access-policy.name}
  title  = "restrict_all"
  status {
    restricted_services = ["storage.googleapis.com"]
  }
}

resource "google_access_context_manager_access_level" "access-level" {
  parent = "accessPolicies/${google_access_context_manager_access_policy.access-policy.name}
  name   = "accessPolicies/${google_access_context_manager_access_policy.access-policy.name}
  title  = "chromeos_no_lock"
  basic {
    conditions {
      device_policy {
        require_screen_lock = false
        os_constraints {
          os_type = "DESKTOP_CHROME_OS"
        }
      }
    }
  }
}
```

```
resource "google_access_context_manager_access_policy" "access-policy" {
  parent = "organizations/123456789"
  title  = "my policy"
}
```

» Argument Reference

The following arguments are supported:

- `title` - (Required) Human readable title. Must be unique within the Policy.
- `parent` - (Required) The AccessPolicy this ServicePerimeter lives in. Format: accessPolicies/{policy_id}
- `name` - (Required) Resource name for the ServicePerimeter. The short_name component must begin with a letter and only include alphanumeric and '_'. Format: accessPolicies/{policy_id}/servicePerimeters/{short_name}

- `description` - (Optional) Description of the ServicePerimeter and its use. Does not affect behavior.
- `perimeter_type` - (Optional) Specifies the type of the Perimeter. There are two types: regular and bridge. Regular Service Perimeter contains resources, access levels, and restricted services. Every resource can be in at most ONE regular Service Perimeter. In addition to being in a regular service perimeter, a resource can also be in zero or more perimeter bridges. A perimeter bridge only contains resources. Cross project operations are permitted if all effected resources share some perimeter (whether bridge or regular). Perimeter Bridge does not contain access levels or services: those are governed entirely by the regular perimeter that resource is in. Perimeter Bridges are typically useful when building more complex topologies with many independent perimeters that need to share some data with a common perimeter, but should not be able to share data among themselves.
- `status` - (Optional) ServicePerimeter configuration. Specifies sets of resources, restricted services and access levels that determine perimeter content and boundaries. Structure is documented below.

The `status` block supports:

- `resources` - (Optional) A list of GCP resources that are inside of the service perimeter. Currently only projects are allowed. Format: projects/{project_number}

- `access_levels` - (Optional) A list of AccessLevel resource names that allow resources within the ServicePerimeter to be accessed from the internet. AccessLevels listed must be in the same policy as this ServicePerimeter. Referencing a nonexistent AccessLevel is a syntax error. If no AccessLevel names are listed, resources within the perimeter can only be accessed via GCP calls with request origins within the perimeter. For Service Perimeter Bridge, must be empty. Format: accessPolicies/{policy_id}/accessLevels/{access_level_name}
- `restricted_services` - (Optional) GCP services that are subject to the Service Perimeter restrictions. Must contain a list of services. For example, if `storage.googleapis.com` is specified, access to the storage buckets inside the perimeter must meet the perimeter's access restrictions.

» Attributes Reference

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

- `create_time` - Time the AccessPolicy was created in UTC.
- `update_time` - Time the AccessPolicy was updated in UTC.

» 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

ServicePerimeter can be imported using any of these accepted formats:

```
$ terraform import google_access_context_manager_service_perimeter.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» google_app_engine_application

Allows creation and management of an App Engine application.

App Engine applications cannot be deleted once they're created; you have to delete the entire project to delete the application. Terraform will report the application has been successfully deleted; this is a limitation of Terraform, and will go away in the future. Terraform is not able to delete App Engine applications.

» Example Usage

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

resource "google_app_engine_application" "app" {
  project     = google_project.my_project.project_id
  location_id = "us-central"
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Required) The project ID to create the application under. ~>**NOTE**: GCP only accepts project ID, not project number. If you are using number, you may get a "Permission denied" error.
- `location_id` - (Required) The location to serve the app from.
- `auth_domain` - (Optional) The domain to authenticate users with when using App Engine's User API.
- `serving_status` - (Optional) The serving status of the app.
- `feature_settings` - (Optional) A block of optional settings to configure specific App Engine features:
  - `split_health_checks` - (Required) Set to false to use the legacy health check instead of the readiness and liveness checks.

» Attributes Reference

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

- `name` - Unique name of the app, usually `apps/{PROJECT_ID}`
- `app_id` - Identifier of the app, usually `{PROJECT_ID}`

- `url_dispatch_rule` - A list of dispatch rule blocks. Each block has a `domain`, `path`, and `service` field.
- `code_bucket` - The GCS bucket code is being stored in for this app.
- `default_hostname` - The default hostname for this app.
- `default_bucket` - The GCS bucket content is being stored in for this app.
- `gcr_domain` - The GCR domain used for storing managed Docker images for this app.

» Import

Applications can be imported using the ID of the project the application belongs to, e.g.

```
$ terraform import google_app_engine_application.app your-project-id
```

» google_app_engine_domain_mapping

A domain serving an App Engine application.

To get more information about DomainMapping, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - App Engine Domain Mapping Basic

```
resource "google_app_engine_domain_mapping" "domain_mapping" {
  domain_name = "verified-domain.com"

  ssl_settings {
    ssl_management_type = "AUTOMATIC"
  }
}
```

» Argument Reference

The following arguments are supported:

- `domain_name` - (Required) Relative name of the domain serving the application. Example: example.com.

- `ssl_settings` - (Optional) SSL configuration for this domain. If unconfigured, this domain will not serve with SSL. Structure is documented below.
- `override_strategy` - (Optional) Whether the domain creation should override any existing mappings for this domain. By default, overrides are rejected.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `ssl_settings` block supports:

- `certificate_id` - (Optional) ID of the AuthorizedCertificate resource configuring SSL for the application. Clearing this field will remove SSL support. By default, a managed certificate is automatically created for every domain mapping. To omit SSL support or to configure SSL manually, specify `SslManagementType.MANUAL` on a `CREATE` or `UPDATE` request. You must be authorized to administer the `AuthorizedCertificate` resource to manually map it to a DomainMapping resource. Example: 12345.
- `ssl_management_type` - (Required) SSL management type for this domain. If `AUTOMATIC`, a managed certificate is automatically provisioned. If `MANUAL`, `certificateId` must be manually specified in order to configure SSL for this domain.
- `pending_managed_certificate_id` - ID of the managed `AuthorizedCertificate` resource currently being provisioned, if applicable. Until the new managed certificate has been successfully provisioned, the previous SSL state will be preserved. Once the provisioning process completes, the `certificateId` field will reflect the new managed certificate and this field will be left empty. To remove SSL support while there is still a pending managed certificate, clear the `certificateId` field with an update request.

» Attributes Reference

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

- `name` - Full path to the DomainMapping resource in the API. Example: apps/myapp/domainMapping/example.com.
- `resource_records` - The resource records required to configure this domain mapping. These records must be added to the domain's DNS configuration in order to serve the application via this domain mapping. Structure is documented below.

The `resource_records` block contains:

- `name` - (Optional) Relative name of the object affected by this record. Only applicable for CNAME records. Example: 'www'.
- `rrdata` - (Optional) Data for this record. Values vary by record type, as defined in RFC 1035 (section 5) and RFC 1034 (section 3.6.1).
- `type` - (Optional) Resource record type. Example: `AAAA`.

» 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

DomainMapping can be imported using any of these accepted formats:

```
$ terraform import google_app_engine_domain_mapping.default apps/{{project}}/domainMappings/
$ terraform import google_app_engine_domain_mapping.default {{project}}/{{domain_name}}
$ terraform import google_app_engine_domain_mapping.default {{domain_name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_app_engine_firewall_rule

A single firewall rule that is evaluated against incoming traffic and provides an action to take on matched requests.

To get more information about FirewallRule, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - App Engine Firewall Rule Basic

```
resource "google_project" "my_project" {
  name       = "tf-test-project"
  project_id = "test-project"
  org_id     = "123456789"
}

resource "google_app_engine_application" "app" {
  project     = google_project.my_project.project_id
  location_id = "us-central"
}

resource "google_app_engine_firewall_rule" "rule" {
  project      = google_app_engine_application.app.project
  priority     = 1000
  action       = "ALLOW"
  source_range = "*"
}
```

» Argument Reference

The following arguments are supported:

- `source_range` - (Required) IP address or range, defined using CIDR notation, of requests that this rule applies to.
- `action` - (Required) The action to take if this rule matches.

- `description` - (Optional) An optional string description of this rule.
- `priority` - (Optional) A positive integer that defines the order of rule evaluation. Rules with the lowest priority are evaluated first. A default rule at priority Int32.MaxValue matches all IPv4 and IPv6 traffic when

no previous rule matches. Only the action of this rule can be modified by the user.

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

» 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

FirewallRule can be imported using any of these accepted formats:

```
$ terraform import google_app_engine_firewall_rule.default apps/{{project}}/firewall/ingress
$ terraform import google_app_engine_firewall_rule.default {{project}}/{{priority}}
$ terraform import google_app_engine_firewall_rule.default {{priority}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_app_engine_standard_app_version

Standard App Version resource to create a new version of standard GAE Application. Currently supporting Zip and File Containers. Currently does not support async operation checking.

To get more information about StandardAppVersion, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - App Engine Standard App Version

```
resource "google_app_engine_standard_app_version" "myapp_v1" {
  version_id = "v1"
  service    = "myapp"
  runtime    = "nodejs10"

  entrypoint {
    shell = "node ./app.js"
  }

  deployment {
    zip {
      source_url = "https://storage.googleapis.com/${google_storage_bucket.bucket.name}/${go
    }
  }

  env_variables = {
    port = "8080"
  }

  delete_service_on_destroy = true
}

resource "google_app_engine_standard_app_version" "myapp_v2" {
  version_id = "v2"
  service    = "myapp"
  runtime    = "nodejs10"

  entrypoint {
    shell = "node ./app.js"
  }

  deployment {
    zip {
      source_url = "https://storage.googleapis.com/${google_storage_bucket.bucket.name}/${go
    }
  }
```

```
  env_variables = {
    port = "8080"
  }

  noop_on_destroy = true
}

resource "google_storage_bucket" "bucket" {
  name = "appengine-static-content"
}

resource "google_storage_bucket_object" "object" {
  name   = "hello-world.zip"
  bucket = google_storage_bucket.bucket.name
  source = "./test-fixtures/appengine/hello-world.zip"
}
```

» Argument Reference

The following arguments are supported:

- `runtime` - (Required) Desired runtime. Example python27.

- `version_id` - (Optional) Relative name of the version within the service. For example, `v1`. Version names can contain only lowercase letters, numbers, or hyphens. Reserved names,"default", "latest", and any name with the prefix "ah-".
- `threadsafe` - (Optional) Whether multiple requests can be dispatched to this version at once.
- `runtime_api_version` - (Optional) The version of the API in the given runtime environment. Please see the app.yaml reference for valid values at https://cloud.google.com/appengine/docs/standard//config/appref
- `handlers` - (Optional) An ordered list of URL-matching patterns that should be applied to incoming requests. The first matching URL handles the request and other request handlers are not attempted. Structure is documented below.
- `libraries` - (Optional) Configuration for third-party Python runtime libraries that are required by the application. Structure is documented below.
- `env_variables` - (Optional) Environment variables available to the application.

- `deployment` - (Optional) Code and application artifacts that make up this version. Structure is documented below.
- `entrypoint` - (Optional) The entrypoint for the application. Structure is documented below.
- `instance_class` - (Optional) Instance class that is used to run this version. Valid values are AutomaticScaling F1, F2, F4, F4_1G (Only AutomaticScaling is supported at the moment)
- `service` - (Optional) AppEngine service resource
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `noop_on_destroy` - (Optional) If set to `true`, the application version will not be deleted.
- `delete_service_on_destroy` - (Optional) If set to `true`, the service will be deleted if it is the last version.

The `handlers` block supports:

- `url_regex` - (Optional) URL prefix. Uses regular expression syntax, which means regexp special characters must be escaped, but should not contain groupings. All URLs that begin with this prefix are handled by this handler, using the portion of the URL after the prefix as part of the file path.
- `security_level` - (Optional) Security (HTTPS) enforcement for this URL.
- `login` - (Optional) Methods to restrict access to a URL based on login status.
- `auth_fail_action` - (Optional) Actions to take when the user is not logged in.
- `redirect_http_response_code` - (Optional) Redirect codes.
- `script` - (Optional) Executes a script to handle the requests that match this URL pattern. Only the auto value is supported for Node.js in the App Engine standard environment, for example "script:" "auto". Structure is documented below.
- `static_files` - (Optional) Files served directly to the user for a given URL, such as images, CSS stylesheets, or JavaScript source files. Static file handlers describe which files in the application directory are static files, and which URLs serve them. Structure is documented below.

The `script` block supports:

- `script_path` - (Required) Path to the script from the application root directory.

The `static_files` block supports:

- `path` - (Optional) Path to the static files matched by the URL pattern, from the application root directory. The path can refer to text matched in groupings in the URL pattern.
- `upload_path_regex` - (Optional) Regular expression that matches the file paths for all files that should be referenced by this handler.
- `http_headers` - (Optional) HTTP headers to use for all responses from these URLs. An object containing a list of "key:value" value pairs.".
- `mime_type` - (Optional) MIME type used to serve all files served by this handler. Defaults to file-specific MIME types, which are derived from each file's filename extension.
- `expiration` - (Optional) Time a static file served by this handler should be cached by web proxies and browsers. A duration in seconds with up to nine fractional digits, terminated by 's'. Example "3.5s".
- `require_matching_file` - (Optional) Whether this handler should match the request if the file referenced by the handler does not exist.
- `application_readable` - (Optional) Whether files should also be uploaded as code data. By default, files declared in static file handlers are uploaded as static data and are only served to end users; they cannot be read by the application. If enabled, uploads are charged against both your code and static data storage resource quotas.

The `libraries` block supports:

- `name` - (Optional) Name of the library. Example "django".
- `version` - (Optional) Version of the library to select, or "latest".

The `deployment` block supports:

- `zip` - (Optional) Zip File Structure is documented below.
- `files` - (Optional) Manifest of the files stored in Google Cloud Storage that are included as part of this version. All files must be readable using the credentials supplied with this call. Structure is documented below.

The `zip` block supports:

- `source_url` - (Required) Source URL
- `files_count` - (Optional) files count

The `files` block supports:

- `name` - (Required) The identifier for this object. Format specified above.
- `sha1_sum` - (Optional) SHA1 checksum of the file
- `source_url` - (Required) Source URL

The `entrypoint` block supports:

- `shell` - (Required) The format should be a shell command that can be fed to bash -c.

» Attributes Reference

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

- `name` - Full path to the Version resource in the API. Example, "v1".

» 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

StandardAppVersion can be imported using any of these accepted formats:

```
$ terraform import google_app_engine_standard_app_version.default apps/{{project}}/services/
$ terraform import google_app_engine_standard_app_version.default {{project}}/{{service}}/{
$ terraform import google_app_engine_standard_app_version.default {{service}}/{{version_id}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_app_engine_application_url_dispatch_rules

Rules to match an HTTP request and dispatch that request to a service.

To get more information about ApplicationUrlDispatchRules, see:

- API documentation

» Example Usage - App Engine Application Url Dispatch Rules Basic

```
resource "google_app_engine_application_url_dispatch_rules" "web_service" {
  dispatch_rules {
    domain  = "*"
    path    = "/*"
    service = "default"
  }

  dispatch_rules {
    domain  = "*"
    path    = "/admin/*"
    service = google_app_engine_standard_app_version.admin_v3.service
  }
}

resource "google_app_engine_standard_app_version" "admin_v3" {
  version_id = "v3"
  service    = "admin"
  runtime    = "nodejs10"

  entrypoint {
    shell = "node ./app.js"
  }

  deployment {
    zip {
      source_url = "https://storage.googleapis.com/${google_storage_bucket.bucket.name}/${go
    }
  }

  env_variables = {
    port = "8080"
  }

  noop_on_destroy = true
}
```

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

resource "google_storage_bucket_object" "object" {
  name   = "hello-world.zip"
  bucket = google_storage_bucket.bucket.name
  source = "./test-fixtures/appengine/hello-world.zip"
}
```

» Argument Reference

The following arguments are supported:

- `dispatch_rules` - (Required) Rules to match an HTTP request and dispatch that request to a service. Structure is documented below.

The `dispatch_rules` block supports:

- `domain` - (Optional) Domain name to match against. The wildcard "*" *is supported if specified before a period: "*.". Defaults to matching all domains: "*".
- `path` - (Required) Pathname within the host. Must start with a "/". A single "*" can be included at the end of the path. The sum of the lengths of the domain and path may not exceed 100 characters.
- `service` - (Required) Pathname within the host. Must start with a "/". A single "*" can be included at the end of the path. The sum of the lengths of the domain and path may not exceed 100 characters.

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

» 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

ApplicationUrlDispatchRules can be imported using any of these accepted formats:

```
$ terraform import google_app_engine_application_url_dispatch_rules.default {{project}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_bigquery_dataset

Datasets allow you to organize and control access to your tables.

» Example Usage - Bigquery Dataset Basic

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

  labels = {
    env = "default"
  }

  access {
    role          = "OWNER"
    user_by_email = "Joe@example.com"
  }
  access {
    role   = "READER"
```

```
    domain = "example.com"
  }
}
```

» Example Usage - Bigquery Dataset Cmek

```
resource "google_bigquery_dataset" "dataset" {
  dataset_id                  = "example_dataset"
  friendly_name               = "test"
  description                 = "This is a test description"
  location                    = "US"
  default_table_expiration_ms = 3600000

  default_encryption_configuration {
    kms_key_name = google_kms_crypto_key.crypto_key.self_link
  }
}

resource "google_kms_crypto_key" "crypto_key" {
  name     = "example-key"
  key_ring = google_kms_key_ring.key_ring.self_link
}

resource "google_kms_key_ring" "key_ring" {
  name     = "example-keyring"
  location = "us"
}
```

» Argument Reference

The following arguments are supported:

- `dataset_id` - (Required) A unique ID for this dataset, without the project name. The ID must contain only letters (a-z, A-Z), numbers (0-9), or underscores (_). The maximum length is 1,024 characters.

- `access` - (Optional) An array of objects that define dataset access for one or more entities. Structure is documented below.
- `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.

- `default_partition_expiration_ms` - (Optional) The default partition expiration for all partitioned tables in the dataset, in milliseconds.

Once this property is set, all newly-created partitioned tables in the dataset will have an `expirationMs` property in the `timePartitioning` settings set to this value, and changing the value will only affect new tables, not existing ones. The storage in a partition will have an expiration time of its partition time plus this value. Setting this property overrides the use of `defaultTableExpirationMs` for partitioned tables: only one of `defaultTableExpirationMs` and `defaultPartitionExpirationMs` will be used for any new partitioned table. If you provide an explicit `timePartitioning.expirationMs` when creating or updating a partitioned table, that value takes precedence over the default partition expiration time indicated by this property.

- `description` - (Optional) A user-friendly description of the dataset
- `friendly_name` - (Optional) A descriptive name for the dataset
- `labels` - (Optional) The labels associated with this dataset. You can use these to organize and group your datasets
- `location` - (Optional) The geographic location where the dataset should reside. See official docs.

There are two types of locations, regional or multi-regional. A regional location is a specific geographic place, such as Tokyo, and a multi-regional location is a large geographic area, such as the United States, that contains at least two geographic places.

Possible regional values include: `asia-east1`, `asia-northeast1`, `asia-southeast1`, `australia-southeast1`, `europe-north1`, `europe-west2` and `us-east4`.

Possible multi-regional values: `EU` and `US`.

The default value is multi-regional location `US`. Changing this forces a new resource to be created.

- `default_encryption_configuration` - (Optional) The default encryption key for all tables in the dataset. Once this property is set, all newly-created partitioned tables in the dataset will have encryption key set to this value, unless table creation request (or query) overrides the key. 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.
- `delete_contents_on_destroy` - (Optional) If set to `true`, delete all the tables in the dataset when destroying the resource; otherwise, destroying the resource will fail if tables are present.

The `access` block supports:

- `domain` - (Optional) A domain to grant access to. Any users signed in with the domain specified will be granted the specified access
- `group_by_email` - (Optional) An email address of a Google Group to grant access to.
- `role` - (Optional) Describes the rights granted to the user specified by the other member of the access object. Primitive, Predefined and custom roles are supported. Predefined roles that have equivalent primitive roles are swapped by the API to their Primitive counterparts, and will show a diff post-create. See official docs.
- `special_group` - (Optional) A special group to grant access to. Possible values include:
  - `projectOwners`: Owners of the enclosing project.
  - `projectReaders`: Readers of the enclosing project.
  - `projectWriters`: Writers of the enclosing project.
  - `allAuthenticatedUsers`: All authenticated BigQuery users.
- `user_by_email` - (Optional) An email address of a user to grant access to. For example: fred@example.com
- `view` - (Optional) A view from a different dataset to grant access to. Queries executed against that view will have read access to tables in this dataset. The role field is not required when this field is set. If that view is updated by any user, access to the view needs to be granted again via an update operation. Structure is documented below.

The `view` block supports:

- `dataset_id` - (Required) The ID of the dataset containing this table.
- `project_id` - (Required) The ID of the project containing this table.
- `table_id` - (Required) The ID of the table. The ID must contain only letters (a-z, A-Z), numbers (0-9), or underscores (_). The maximum length is 1,024 characters.

The `default_encryption_configuration` block supports:

- `kms_key_name` - (Required) Describes the Cloud KMS encryption key that will be used to protect destination BigQuery table. The BigQuery Service

Account associated with your project requires access to this encryption key.

» Attributes Reference

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

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

Dataset can be imported using any of these accepted formats:

```
$ terraform import google_bigquery_dataset.default projects/{{project}}/datasets/{{dataset_i
$ terraform import google_bigquery_dataset.default {{project}}/{{dataset_id}}
$ terraform import google_bigquery_dataset.default {{dataset_id}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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 = <<EOF
[
  {
    "name": "permalink",
    "type": "STRING",
    "mode": "NULLABLE",
    "description": "The Permalink"
  },
  {
    "name": "state",
    "type": "STRING",
    "mode": "NULLABLE",
    "description": "State where the head office is located"
```

```
  }
]
EOF

}

resource "google_bigquery_table" "sheet" {
  dataset_id = google_bigquery_dataset.default.dataset_id
  table_id   = "sheet"

  external_data_configuration {
    autodetect    = true
    source_format = "GOOGLE_SHEETS"

    google_sheets_options {
      skip_leading_rows = 1
    }

    source_uris = [
      "https://docs.google.com/spreadsheets/d/123456789012345",
    ]
  }
}
```

» 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.
- `external_data_configuration` - (Optional) Describes the data format, location, and other properties of a table stored outside of BigQuery. By defining these properties, the data source can then be queried as if it were a standard BigQuery table. Structure is documented below.

- `friendly_name` - (Optional) A descriptive name for the table.
- `encryption_configuration` - (Optional) Specifies how the table should be encrypted. If left blank, the table will be encrypted with a Google-managed key; that process is transparent to the user. Structure is documented below.
- `labels` - (Optional) A mapping of labels to assign to the resource.
- `schema` - (Optional) A JSON schema for the table. Schema is required for CSV and JSON formats and is disallowed for Google Cloud Bigtable, Cloud Datastore backups, and Avro formats when using external tables. For more information see the BigQuery API documentation. ~>**NOTE**: Because this field expects a JSON string, any changes to the string will create a diff, even if the JSON itself hasn't changed. If the API returns a different value for the same schema, e.g. it switched the order of values or replaced `STRUCT` field type with `RECORD` field type, we currently cannot suppress the recurring diff this causes. As a workaround, we recommend using the schema as returned by the API.
- `time_partitioning` - (Optional) If specified, configures time-based partitioning for this table. Structure is documented below.
- `clustering` - (Optional) Specifies column names to use for data clustering. Up to four top-level columns are allowed, and should be specified in descending priority order.
- `view` - (Optional) If specified, configures this table as a view. Structure is documented below.

The `external_data_configuration` block supports:

- `autodetect` - (Required) - Let BigQuery try to autodetect the schema and format of the table.
- `compression` (Optional) - The compression type of the data source. Valid values are "NONE" or "GZIP".
- `csv_options` (Optional) - Additional properties to set if `source_format` is set to "CSV". Structure is documented below.
- `google_sheets_options` (Optional) - Additional options if `source_format` is set to "GOOGLE_SHEETS". Structure is documented below.
- `ignore_unknown_values` (Optional) - Indicates if BigQuery should allow extra values that are not represented in the table schema. If true, the extra values are ignored. If false, records with extra columns are treated as bad records, and if there are too many bad records, an invalid error is returned in the job result. The default value is false.
- `max_bad_records` (Optional) - The maximum number of bad records that BigQuery can ignore when reading data.

- `source_format` (Required) - The data format. Supported values are: "CSV", "GOOGLE_SHEETS", "NEWLINE_DELIMITED_JSON", "AVRO", and "DATSTORE_BACKUP". To use "GOOGLE_SHEETS" the `scopes` must include "https://www.googleapis.com/auth/drive.readonly".
- `source_uris` - (Required) A list of the fully-qualified URIs that point to your data in Google Cloud.

The `csv_options` block supports:

- `quote` (Required) - The value that is used to quote data sections in a CSV file. If your data does not contain quoted sections, set the property value to an empty string. If your data contains quoted newline characters, you must also set the `allow_quoted_newlines` property to true. The API-side default is `"`, specified in Terraform escaped as `\"`. Due to limitations with Terraform default values, this value is required to be explicitly set.
- `allow_jagged_rows` (Optional) - Indicates if BigQuery should accept rows that are missing trailing optional columns.
- `allow_quoted_newlines` (Optional) - Indicates if BigQuery should allow quoted data sections that contain newline characters in a CSV file. The default value is false.
- `encoding` (Optional) - The character encoding of the data. The supported values are UTF-8 or ISO-8859-1.
- `field_delimiter` (Optional) - The separator for fields in a CSV file.
- `skip_leading_rows` (Optional) - The number of rows at the top of a CSV file that BigQuery will skip when reading the data.

The `google_sheets_options` block supports:

- `range` (Optional) - Range of a sheet to query from. Only used when non-empty. At least one of `range` or `skip_leading_rows` must be set. Typical format: "sheet_name!top_left_cell_id:bottom_right_cell_id" For example: "sheet1!A1:B20"
- `skip_leading_rows` (Optional) - The number of rows at the top of the sheet that BigQuery will skip when reading the data. At least one of `range` or `skip_leading_rows` must be set.

The `time_partitioning` block supports:

- `expiration_ms` - (Optional) Number of milliseconds for which to keep the storage for a partition.
- `field` - (Optional) The field used to determine how to create a time-based partition. If time-based partitioning is enabled without this value, the table is partitioned based on the load time.

- `type` - (Required) The only type supported is DAY, which will generate one partition per day based on data loading time.
- `require_partition_filter` - (Optional) If set to true, queries over this table require a partition filter that can be used for partition elimination to be specified.

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.

The `encryption_configuration` block supports the following arguments:

- `kms_key_name` - (Required) The self link or full name of a key which should be used to encrypt this table. Note that the default bigquery service account will need to have encrypt/decrypt permissions on this key - you may want to see the `google_bigquery_default_service_account` datasource and the `google_kms_crypto_key_iam_binding` resource.

» 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_bigquery_data_transfer_config

Represents a data transfer configuration. A transfer configuration contains all metadata needed to perform a data transfer.

To get more information about Config, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Scheduled Query

```
data "google_project" "project" {
}

resource "google_project_iam_member" "permissions" {
  role   = "roles/iam.serviceAccountShortTermTokenMinter"
  member = "serviceAccount:service-${data.google_project.project.number}@gcp-sa-bigquerydata
}

resource "google_bigquery_data_transfer_config" "query_config" {
  depends_on = [google_project_iam_member.permissions]

  display_name           = "my-query"
  location               = "asia-northeast1"
  data_source_id         = "scheduled_query"
  schedule               = "first sunday of quarter 00:00"
  destination_dataset_id = google_bigquery_dataset.my_dataset.dataset_id
  params = {
    destination_table_name_template = "my-table"
    write_disposition               = "WRITE_APPEND"
    query                           = "SELECT name FROM tabl WHERE x = 'y'"
  }
}

resource "google_bigquery_dataset" "my_dataset" {
  depends_on = [google_project_iam_member.permissions]
```

```
  dataset_id    = "my_dataset"
  friendly_name = "foo"
  description   = "bar"
  location      = "asia-northeast1"
}
```

» Argument Reference

The following arguments are supported:

- `display_name` - (Required) The user specified display name for the transfer config.
- `destination_dataset_id` - (Required) The BigQuery target dataset id.
- `data_source_id` - (Required) The data source id. Cannot be changed once the transfer config is created.
- `params` - (Required) These parameters are specific to each data source.

- `schedule` - (Optional) Data transfer schedule. If the data source does not support a custom schedule, this should be empty. If it is empty, the default value for the data source will be used. The specified times are in UTC. Examples of valid format: 1st,3rd monday of month 15:30, every wed,fri of jan, jun 13:15, and first sunday of quarter 00:00. See more explanation about the format here: https://cloud.google.com/appengine/docs/flexible/python/scheduling-jobs-with-cron-yaml#the_schedule_format NOTE: the granularity should be at least 8 hours, or less frequent.
- `data_refresh_window_days` - (Optional) The number of days to look back to automatically refresh the data. For example, if dataRefreshWindowDays = 10, then every day BigQuery reingests data for [today-10, today-1], rather than ingesting data for just [today-1]. Only valid if the data source supports the feature. Set the value to 0 to use the default value.
- `disabled` - (Optional) When set to true, no runs are scheduled for a given transfer.
- `location` - (Optional) The geographic location where the transfer config should reside. Examples: US, EU, asia-northeast1. The default value is US.
- `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` - The resource name of the transfer config. Transfer config names have the form projects/{projectId}/locations/{location}/transferConfigs/{configId}. Where configId is usually a uuid, but this is not required. The name is ignored when creating a transfer config.

» 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

Config can be imported using any of these accepted formats:

```
$ terraform import google_bigquery_data_transfer_config.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_bigtable_app_profile

App profile is a configuration object describing how Cloud Bigtable should treat traffic from a particular end user application.

» Example Usage - Bigtable App Profile Multicluster

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

resource "google_bigtable_app_profile" "ap" {
  instance       = google_bigtable_instance.instance.name
  app_profile_id = "tf-test-profile-"

  multi_cluster_routing_use_any = true
  ignore_warnings               = true
}
```

» Example Usage - Bigtable App Profile Singlecluster

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

resource "google_bigtable_app_profile" "ap" {
  instance       = google_bigtable_instance.instance.name
  app_profile_id = "tf-test-profile-"

  single_cluster_routing {
    cluster_id                 = "tf-test-instance-"
    allow_transactional_writes = true
```

```
  }

  ignore_warnings = true
}
```

» Argument Reference

The following arguments are supported:

- `app_profile_id` - (Required) The unique name of the app profile in the form `[_a-zA-Z0-9][-_.a-zA-Z0-9]*`.

- `description` - (Optional) Long form description of the use case for this app profile.
- `multi_cluster_routing_use_any` - (Optional) If true, read/write requests are routed to the nearest cluster in the instance, and will fail over to the nearest cluster that is available in the event of transient errors or delays. Clusters in a region are considered equidistant. Choosing this option sacrifices read-your-writes consistency to improve availability.
- `single_cluster_routing` - (Optional) Use a single-cluster routing policy. Structure is documented below.
- `instance` - (Optional) The name of the instance to create the app profile within.
- `ignore_warnings` - (Optional) If true, ignore safety checks when deleting/updating the app profile.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `single_cluster_routing` block supports:

- `cluster_id` - (Required) The cluster to which read/write requests should be routed.
- `allow_transactional_writes` - (Optional) If true, CheckAndMutateRow and ReadModifyWriteRow requests are allowed by this app profile. It is unsafe to send these requests to the same table/row/column in multiple clusters.

» Attributes Reference

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

- `name` - The unique name of the requested app profile. Values are of the form `projects/<project>/instances/<instance>/appProfiles/<appProfileId>`.

» 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

AppProfile can be imported using any of these accepted formats:

```
$ terraform import google_bigtable_app_profile.default projects/{{project}}/instances/{{inst
$ terraform import google_bigtable_app_profile.default {{project}}/{{instance}}/{{app_profil
$ terraform import google_bigtable_app_profile.default {{instance}}/{{app_profile_id}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_bigtable_gc_policy

Creates a Google Cloud Bigtable GC Policy inside a family. For more information see the official documentation and API.

» Example Usage

```
resource "google_bigtable_instance" "instance" {
  name = "tf-instance"
  cluster {
    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

  column_family {
    family = "name"
  }
}

resource "google_bigtable_gc_policy" "policy" {
  instance_name = google_bigtable_instance.instance.name
  table         = google_bigtable_table.table.name
  column_family = "name"

  max_age {
    days = 7
  }
}
```

Multiple conditions is also supported. `UNION` when any of its sub-policies apply (OR). `INTERSECTION` when all its sub-policies apply (AND)

```
resource "google_bigtable_gc_policy" "policy" {
  instance_name = google_bigtable_instance.instance.name
  table         = google_bigtable_table.table.name
  column_family = "name"

  mode = "UNION"

  max_age {
    days = 7
  }

  max_version {
    number = 10
  }
}
```

» Argument Reference

The following arguments are supported:

- `table` - (Required) The name of the table.
- `instance_name` - (Required) The name of the Bigtable instance.

- `column_family` - (Required) The name of the column family.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `mode` - (Optional) If multiple policies are set, you should choose between `UNION` OR `INTERSECTION`.
- `max_age` - (Optional) GC policy that applies to all cells older than the given age.
- `max_version` - (Optional) GC policy that applies to all versions of a cell except for the most recent.

`max_age` supports the following arguments:

- `days` - (Required) Number of days before applying GC policy.

`max_version` supports the following arguments:

- `number` - (Required) Number of version before applying the GC policy.

» Attributes Reference

Only the arguments listed above are exposed as attributes.

» google_bigtable_instance

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

» Example Usage - Production Instance

```
resource "google_bigtable_instance" "production-instance" {
  name = "tf-instance"

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

» Example Usage - Development Instance

```
resource "google_bigtable_instance" "development-instance" {
  name          = "tf-instance"
  instance_type = "DEVELOPMENT"

  cluster {
    cluster_id   = "tf-instance-cluster"
    zone         = "us-central1-b"
    storage_type = "HDD"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name (also called Instance Id in the Cloud Console) of the Cloud Bigtable instance.
- `cluster` - (Required) A block of cluster configuration options. This can be specified 1 or 2 times. See structure below.

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `instance_type` - (Optional) The instance type to create. One of `"DEVELOPMENT"` or `"PRODUCTION"`. Defaults to `"PRODUCTION"`.
- `display_name` - (Optional) The human-readable display name of the Bigtable instance. Defaults to the instance `name`.

The `cluster` block supports the following arguments:

- `cluster_id` - (Required) The ID of the Cloud Bigtable cluster.
- `zone` - (Required) The zone to create the Cloud Bigtable cluster in. Each cluster must have a different zone in the same region. Zones that support Bigtable instances are noted on the Cloud Bigtable locations page.
- `num_nodes` - (Optional) The number of nodes in your Cloud Bigtable cluster. Required, with a minimum of `3` for a `PRODUCTION` instance. Must be left unset for a `DEVELOPMENT` instance.
- `storage_type` - (Optional) The storage type to use. One of `"SSD"` or `"HDD"`. Defaults to `"SSD"`.

» Attributes Reference

Only the arguments listed above are exposed as attributes.

» Import

Bigtable Instances can be imported using any of these accepted formats:

```
$ terraform import google_bigtable_instance.default projects/{{project}}/instances/{{name}}
$ terraform import google_bigtable_instance.default {{project}}/{{name}}
$ terraform import google_bigtable_instance.default {{name}}
```

» IAM policy for Bigtable instance

Three different resources help you manage IAM policies on bigtable instances. Each of these resources serves a different use case:

- `google_bigtable_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.
- `google_bigtable_instance_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 instance are preserved.
- `google_bigtable_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_bigtable_instance_iam_policy` **cannot** be used in conjunction with `google_bigtable_instance_iam_binding` and `google_bigtable_instance_iam_member` or they will fight over what your policy should be. In addition, be careful not to accidentally unset ownership of the instance as `google_bigtable_instance_iam_policy` replaces the entire policy.

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

» google_bigtable_instance_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"
    members = [
      "user:jane@example.com",
    ]
```

```
  }
}

resource "google_bigtable_instance_iam_policy" "editor" {
  project     = "your-project"
  instance    = "your-bigtable-instance"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_bigtable_instance_iam_binding

```
resource "google_bigtable_instance_iam_binding" "editor" {
  instance = "your-bigtable-instance"
  role     = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_bigtable_instance_iam_member

```
resource "google_bigtable_instance_iam_member" "editor" {
  instance = "your-bigtable-instance"
  role     = "roles/editor"
  member   = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name or relative resource id of the instance to manage IAM policies for.

For `google_bigtable_instance_iam_member` or `google_bigtable_instance_iam_binding`:

- `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 G Suite domain (primary, instead of alias) 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_bigtable_instance_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-nam`

`google_bigtable_instance_iam_policy` only: * `policy_data` - (Required) The policy data generated by a `google_iam_policy` data source.

- `project` - (Optional) The project in which the instance belongs. If it is not provided, Terraform will use the provider default.

» Attributes Reference

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

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

» Import

Instance IAM resources can be imported using the project, instance name, role and/or member.

```
$ terraform import google_bigtable_instance_iam_policy.editor "projects/{project}/instances/

$ terraform import google_bigtable_instance_iam_binding.editor "projects/{project}/instances

$ terraform import google_bigtable_instance_iam_member.editor "projects/{project}/instances/
```

» IAM policy for Bigtable instance

Three different resources help you manage IAM policies on bigtable instances. Each of these resources serves a different use case:

- `google_bigtable_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.

- `google_bigtable_instance_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 instance are preserved.
- `google_bigtable_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_bigtable_instance_iam_policy` **cannot** be used in conjunction with `google_bigtable_instance_iam_binding` and `google_bigtable_instance_iam_member` or they will fight over what your policy should be. In addition, be careful not to accidentally unset ownership of the instance as `google_bigtable_instance_iam_policy` replaces the entire policy.

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

» google_bigtable_instance_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/editor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_bigtable_instance_iam_policy" "editor" {
  project     = "your-project"
  instance    = "your-bigtable-instance"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_bigtable_instance_iam_binding

```
resource "google_bigtable_instance_iam_binding" "editor" {
  instance = "your-bigtable-instance"
  role     = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_bigtable_instance_iam_member

```
resource "google_bigtable_instance_iam_member" "editor" {
  instance = "your-bigtable-instance"
  role     = "roles/editor"
  member   = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name or relative resource id of the instance to manage IAM policies for.

For `google_bigtable_instance_iam_member` or `google_bigtable_instance_iam_binding`:

- `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 G Suite domain (primary, instead of alias) 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_bigtable_instance_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-nam`

`google_bigtable_instance_iam_policy` only: * `policy_data` - (Required) The policy data generated by a `google_iam_policy` data source.

- `project` - (Optional) The project in which the instance belongs. If it is not provided, Terraform will use the provider default.

» Attributes Reference

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

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

» Import

Instance IAM resources can be imported using the project, instance name, role and/or member.

```
$ terraform import google_bigtable_instance_iam_policy.editor "projects/{project}/instances/

$ terraform import google_bigtable_instance_iam_binding.editor "projects/{project}/instances

$ terraform import google_bigtable_instance_iam_member.editor "projects/{project}/instances/
```

» IAM policy for Bigtable instance

Three different resources help you manage IAM policies on bigtable instances. Each of these resources serves a different use case:

- `google_bigtable_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.
- `google_bigtable_instance_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 instance are preserved.
- `google_bigtable_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_bigtable_instance_iam_policy` **cannot** be used in conjunction with `google_bigtable_instance_iam_binding` and `google_bigtable_instance_iam_member` or they will fight over what your policy should be. In addition, be careful not to accidentally unset ownership of the instance as `google_bigtable_instance_iam_policy` replaces the entire policy.

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

» google_bigtable_instance_iam_policy

```
data "google_iam_policy" "admin" {
```

```
  binding {
    role = "roles/editor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_bigtable_instance_iam_policy" "editor" {
  project     = "your-project"
  instance    = "your-bigtable-instance"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_bigtable_instance_iam_binding

```
resource "google_bigtable_instance_iam_binding" "editor" {
  instance = "your-bigtable-instance"
  role     = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_bigtable_instance_iam_member

```
resource "google_bigtable_instance_iam_member" "editor" {
  instance = "your-bigtable-instance"
  role     = "roles/editor"
  member   = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name or relative resource id of the instance to manage IAM policies for.

For `google_bigtable_instance_iam_member` or `google_bigtable_instance_iam_binding`:

- `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 G Suite domain (primary, instead of alias) 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_bigtable_instance_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-nam`

`google_bigtable_instance_iam_policy` only: * `policy_data` - (Required) The policy data generated by a `google_iam_policy` data source.

- `project` - (Optional) The project in which the instance belongs. If it is not provided, Terraform will use the provider default.

» Attributes Reference

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

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

» Import

Instance IAM resources can be imported using the project, instance name, role and/or member.

```
$ terraform import google_bigtable_instance_iam_policy.editor "projects/{project}/instances/

$ terraform import google_bigtable_instance_iam_binding.editor "projects/{project}/instances

$ terraform import google_bigtable_instance_iam_member.editor "projects/{project}/instances/
```

» google_bigtable_table

Creates a Google Cloud 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 {
    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.
- `column_family` - (Optional) A group of columns within a table which share a common configuration. This can be specified multiple times. 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.

`column_family` supports the following arguments:

- `family` - (Optional) The name of the column family.

» Attributes Reference

Only the arguments listed above are exposed as attributes.

» Import

Bigtable Tables can be imported using any of these accepted formats:

```
$ terraform import google_bigtable_table.default projects/{{project}}/instances/{{instance_n
$ terraform import google_bigtable_table.default {{project}}/{{instance_name}}/{{name}}
$ terraform import google_bigtable_table.default {{instance_name}}/{{name}}
```

The following fields can't be read and will show diffs if set in config when imported:

- `split_keys`

» google_binary_authorization_attestor

An attestor that attests to container image artifacts.

To get more information about Attestor, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Binary Authorization Attestor Basic

```
resource "google_binary_authorization_attestor" "attestor" {
  name = "test-attestor"
  attestation_authority_note {
    note_reference = google_container_analysis_note.note.name
    public_keys {
      ascii_armored_pgp_public_key = <<EOF
mQENBFtPOdoBCADF+joTiXWKVuP8kJt3fgpBSjT9h8ezMfKA4aXZctYLx5wslWQl
bB7Iu2ezkECNzoEeU7WxUe8a61pMCh9cisS9H5mB2K2uM4Jnf8tgFeXn3akJDVo0
oR1IC+Dp9mXbRSK3MAvKkOwWlG99sx3uEdvmeBRHBOO+grchLx24EThXFOyP9Fk6
V39j6xMjw4aggLD15B4V0v9JqBDdJiIYFzszZDL6pJwZrzcPOz8JO4rTZd+f64bD
Mpj52j/pQfA8lZHOaAgb1OrthLdMrBAjoDjArV4Ek7vSbrcgYWcI6BhsQrFoxKdX
```

```
83TZKai55ZCfCLIskwUIzA1NLVwyzCS+fSN/ABEBAAG0KCJUZXN0IEF0dGVzdG9y
IiA8ZGFuYWhvZmZtYW5AZ29vZ2xlLmNvbT6JAU4EEwEIADgWIQRfWkqHt6hpTA1L
uY060eeM4dc66AUCW0/R2gIbLwULCQgHAgYVCgkICwIEFgIDAQIeAQIXgAAKCRA6
0eeM4dc66HdpCAC4ot3b0OyxPb0Ip+WT2U0PbpTBPJklesuwpIrM4Lh0N+1nVRLC
51WSmVbM8BiAFhLbN9LpdHhds1kUrHF7+wWAjdR8sqAj9otc6HGRM/3qfa2qgh+U
WTEk/3us/rYSi7T7TkMuutRMIa1IkR13uKiW56csEMnbOQpn9rDqwIr5R8nlZP5h
MAU9vdm1DIv567meMqTaVZgR3w7bck2P49AO81O5ERFpVkErtu/98y+rUy9d789l
+OPuS1NGnxI1YKsNaWJF4uJVuvQuZ1twrhCbGNtVorO2U12+cEq+YtUxj7kmdOC1
qoIRW6y0+UlAc+MbqfL0ziHDOAmcqz1GnROg
=6Bvm
EOF

    }
  }
}

resource "google_container_analysis_note" "note" {
  name = "test-attestor-note"
  attestation_authority {
    hint {
      human_readable_name = "Attestor Note"
    }
  }
}
```

» Example Usage - Binary Authorization Attestor Kms

```
resource "google_binary_authorization_attestor" "attestor" {
  name = "test-attestor"
  attestation_authority_note {
    note_reference = google_container_analysis_note.note.name
    public_keys {
      id = data.google_kms_crypto_key_version.version.id
      pkix_public_key {
        public_key_pem      = data.google_kms_crypto_key_version.version.public_key[0].pem
        signature_algorithm = data.google_kms_crypto_key_version.version.public_key[0].algor
      }
    }
  }
}

data "google_kms_crypto_key_version" "version" {
  crypto_key = google_kms_crypto_key.crypto-key.self_link
}
```

```
resource "google_container_analysis_note" "note" {
  name = "test-attestor-note"
  attestation_authority {
    hint {
      human_readable_name = "Attestor Note"
    }
  }
}

resource "google_kms_crypto_key" "crypto-key" {
  name     = "test-attestor-key"
  key_ring = google_kms_key_ring.keyring.self_link
  purpose  = "ASYMMETRIC_SIGN"

  version_template {
    algorithm = "RSA_SIGN_PKCS1_4096_SHA512"
  }

  lifecycle {
    prevent_destroy = true
  }
}

resource "google_kms_key_ring" "keyring" {
  name     = "test-attestor-key-ring"
  location = "global"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource name.
- `attestation_authority_note` - (Required) A Container Analysis ATTESTATION_AUTHORITY Note, created by the user. Structure is documented below.

The `attestation_authority_note` block supports:

- `note_reference` - (Required) The resource name of a ATTESTATION_AUTHORITY Note, created by the user. If the Note is in a different project from the Attestor, it should be specified in the format `projects/*/notes/*` (or the legacy `providers/*/notes/*`). This field may not be updated. An attestation by this attestor is stored as a Container Analysis ATTESTATION_AUTHORITY Occurrence that names a container image and that links to this Note.

- `public_keys` - (Optional) Public keys that verify attestations signed by this attestor. This field may be updated. If this field is non-empty, one of the specified public keys must verify that an attestation was signed by this attestor for the image specified in the admission request. If this field is empty, this attestor always returns that no valid attestations exist. Structure is documented below.
- `delegation_service_account_email` - This field will contain the service account email address that this Attestor will use as the principal when querying Container Analysis. Attestor administrators must grant this service account the IAM role needed to read attestations from the noteReference in Container Analysis (containeranalysis.notes.occurrences.viewer). This email address is fixed for the lifetime of the Attestor, but callers should not make any other assumptions about the service account email; future versions may use an email based on a different naming pattern.

The `public_keys` block supports:

- `comment` - (Optional) A descriptive comment. This field may be updated.
- `id` - (Optional) The ID of this public key. Signatures verified by BinAuthz must include the ID of the public key that can be used to verify them, and that ID must match the contents of this field exactly. Additional restrictions on this field can be imposed based on which public key type is encapsulated. See the documentation on publicKey cases below for details.
- `ascii_armored_pgp_public_key` - (Optional) ASCII-armored representation of a PGP public key, as the entire output by the command `gpg --export --armor foo@example.com` (either LF or CRLF line endings). When using this field, id should be left blank. The BinAuthz API handlers will calculate the ID and fill it in automatically. BinAuthz computes this ID as the OpenPGP RFC4880 V4 fingerprint, represented as uppercase hex. If id is provided by the caller, it will be overwritten by the API-calculated ID.
- `pkix_public_key` - (Optional) A raw PKIX SubjectPublicKeyInfo format public key. NOTE: id may be explicitly provided by the caller when using this type of public key, but it MUST be a valid RFC3986 URI. If id is left blank, a default one will be computed based on the digest of the DER encoding of the public key. Structure is documented below.

The `pkix_public_key` block supports:

- `public_key_pem` - (Optional) A PEM-encoded public key, as described in `https://tools.ietf.org/html/rfc7468#section-13`
- `signature_algorithm` - (Optional) The signature algorithm used to verify a message against a signature using this key. These signature algorithm must match the structure and any object identifiers encoded in publicKeyPem (i.e. this algorithm must match that of the public key).

- `description` - (Optional) A descriptive comment. This field may be updated. The field may be displayed in chooser dialogs.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

» 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

Attestor can be imported using any of these accepted formats:

```
$ terraform import google_binary_authorization_attestor.default projects/{{project}}/attesto
$ terraform import google_binary_authorization_attestor.default {{project}}/{{name}}
$ terraform import google_binary_authorization_attestor.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for BinaryAuthorizationAttestor

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

- `google_binary_authorization_attestor_iam_policy`: Authoritative. Sets the IAM policy for the attestor and replaces any existing policy already attached.
- `google_binary_authorization_attestor_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 attestor are preserved.

- `google_binary_authorization_attestor_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the attestor are preserved.

Note: `google_binary_authorization_attestor_iam_policy` **cannot** be used in conjunction with `google_binary_authorization_attestor_iam_binding` and `google_binary_authorization_attestor_iam_member` or they will fight over what your policy should be.

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

» google_binary_authorization_attestor_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_binary_authorization_attestor_iam_policy" "editor" {
  project = "${google_binary_authorization_attestor.attestor.project}"
  attestor = "${google_binary_authorization_attestor.attestor.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_binary_authorization_attestor_iam_binding

```
resource "google_binary_authorization_attestor_iam_binding" "editor" {
  project = "${google_binary_authorization_attestor.attestor.project}"
  attestor = "${google_binary_authorization_attestor.attestor.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_binary_authorization_attestor_iam_member

```
resource "google_binary_authorization_attestor_iam_member" "editor" {
```

```
  project = "${google_binary_authorization_attestor.attestor.project}"
  attestor = "${google_binary_authorization_attestor.attestor.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `attestor` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_binary_authorization_attestor_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_binary_authorization_attestor_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/attestors/{{name}}
- {{project}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

BinaryAuthorization attestor IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_binary_authorization_attestor_iam_member.editor "projects/{{project}}/attestors/{{att roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_binary_authorization_attestor_iam_binding.editor "projects/{{project}}/attestors/{{attestor}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_binary_authorization_attestor_iam_policy.editor projects/{{project}}/attestors/{{attestor}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for BinaryAuthorizationAttestor

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

- `google_binary_authorization_attestor_iam_policy`: Authoritative. Sets the IAM policy for the attestor and replaces any existing policy already attached.
- `google_binary_authorization_attestor_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 attestor are preserved.
- `google_binary_authorization_attestor_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the attestor are preserved.

Note: `google_binary_authorization_attestor_iam_policy` **cannot** be used in conjunction with `google_binary_authorization_attestor_iam_binding` and `google_binary_authorization_attestor_iam_member` or they will fight over what your policy should be.

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

» google_binary_authorization_attestor_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_binary_authorization_attestor_iam_policy" "editor" {
  project = "${google_binary_authorization_attestor.attestor.project}"
  attestor = "${google_binary_authorization_attestor.attestor.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_binary_authorization_attestor_iam_binding

```
resource "google_binary_authorization_attestor_iam_binding" "editor" {
  project = "${google_binary_authorization_attestor.attestor.project}"
```

```
  attestor = "${google_binary_authorization_attestor.attestor.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_binary_authorization_attestor_iam_member

```
resource "google_binary_authorization_attestor_iam_member" "editor" {
  project = "${google_binary_authorization_attestor.attestor.project}"
  attestor = "${google_binary_authorization_attestor.attestor.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `attestor` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_binary_authorization_attestor_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_binary_authorization_attestor_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/attestors/{{name}}
- {{project}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

BinaryAuthorization attestor IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_binary_authorization_attestor_iam_member.editor "projects/{{project}}/attestors/{{att roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_binary_authorization_attestor_iam_binding.editor "projects/{{project}}/attestors/{{attestor}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_binary_authorization_attestor_iam_policy.editor projects/{{project}}/attestors/{{attestor}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for BinaryAuthorizationAttestor

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

- `google_binary_authorization_attestor_iam_policy`: Authoritative. Sets the IAM policy for the attestor and replaces any existing policy already attached.
- `google_binary_authorization_attestor_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 attestor are preserved.
- `google_binary_authorization_attestor_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the attestor are preserved.

Note: `google_binary_authorization_attestor_iam_policy` **cannot** be used in conjunction with `google_binary_authorization_attestor_iam_binding` and `google_binary_authorization_attestor_iam_member` or they will fight over what your policy should be.

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

» google_binary_authorization_attestor_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_binary_authorization_attestor_iam_policy" "editor" {
  project = "${google_binary_authorization_attestor.attestor.project}"
  attestor = "${google_binary_authorization_attestor.attestor.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_binary_authorization_attestor_iam_binding

```
resource "google_binary_authorization_attestor_iam_binding" "editor" {
  project = "${google_binary_authorization_attestor.attestor.project}"
  attestor = "${google_binary_authorization_attestor.attestor.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_binary_authorization_attestor_iam_member

```
resource "google_binary_authorization_attestor_iam_member" "editor" {
  project = "${google_binary_authorization_attestor.attestor.project}"
  attestor = "${google_binary_authorization_attestor.attestor.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `attestor` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_binary_authorization_attestor_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_binary_authorization_attestor_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/attestors/{{name}}
- {{project}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

BinaryAuthorization attestor IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_binary_authorization_attestor_iam_member.editor "projects/{{project}}/attestors/{{att roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_binary_authorization_attestor_iam_binding.editor "projects/{{project}}/attestors/{{attestor}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_binary_authorization_attestor_iam_policy.editor projects/{{project}}/attestors/{{attestor}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_binary_authorization_policy

A policy for container image binary authorization.

To get more information about Policy, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Binary Authorization Policy Basic

```
resource "google_binary_authorization_policy" "policy" {
  admission_whitelist_patterns {
    name_pattern = "gcr.io/google_containers/*"
  }

  default_admission_rule {
    evaluation_mode  = "ALWAYS_ALLOW"
    enforcement_mode = "ENFORCED_BLOCK_AND_AUDIT_LOG"
  }

  cluster_admission_rules {
    cluster                 = "us-central1-a.prod-cluster"
    evaluation_mode         = "REQUIRE_ATTESTATION"
    enforcement_mode        = "ENFORCED_BLOCK_AND_AUDIT_LOG"
    require_attestations_by = [google_binary_authorization_attestor.attestor.name]
  }
}

resource "google_container_analysis_note" "note" {
  name = "test-attestor-note"
  attestation_authority {
    hint {
      human_readable_name = "My attestor"
```

```
    }
  }
}

resource "google_binary_authorization_attestor" "attestor" {
  name = "test-attestor"
  attestation_authority_note {
    note_reference = google_container_analysis_note.note.name
  }
}
```

» Example Usage - Binary Authorization Policy Global Evaluation

```
resource "google_binary_authorization_policy" "policy" {
  default_admission_rule {
    evaluation_mode         = "REQUIRE_ATTESTATION"
    enforcement_mode        = "ENFORCED_BLOCK_AND_AUDIT_LOG"
    require_attestations_by = [google_binary_authorization_attestor.attestor.name]
  }

  global_policy_evaluation_mode = "ENABLE"
}

resource "google_container_analysis_note" "note" {
  name = "test-attestor-note"
  attestation_authority {
    hint {
      human_readable_name = "My attestor"
    }
  }
}

resource "google_binary_authorization_attestor" "attestor" {
  name = "test-attestor"
  attestation_authority_note {
    note_reference = google_container_analysis_note.note.name
  }
}
```

» Argument Reference

The following arguments are supported:

- `default_admission_rule` - (Required) Default admission rule for a cluster without a per-cluster admission rule. Structure is documented below.

The `default_admission_rule` block supports:

- `evaluation_mode` - (Required) How this admission rule will be evaluated.
- `require_attestations_by` - (Optional) The resource names of the attestors that must attest to a container image. If the attestor is in a different project from the policy, it should be specified in the format `projects/*/attestors/*`. Each attestor must exist before a policy can reference it. To add an attestor to a policy the principal issuing the policy change request must be able to read the attestor resource. Note: this field must be non-empty when the evaluation_mode field specifies REQUIRE_ATTESTATION, otherwise it must be empty.
- `enforcement_mode` - (Required) The action when a pod creation is denied by the admission rule.

- `description` - (Optional) A descriptive comment.
- `global_policy_evaluation_mode` - (Optional) Controls the evaluation of a Google-maintained global admission policy for common system-level images. Images not covered by the global policy will be subject to the project admission policy.
- `admission_whitelist_patterns` - (Optional) A whitelist of image patterns to exclude from admission rules. If an image's name matches a whitelist pattern, the image's admission requests will always be permitted regardless of your admission rules. Structure is documented below.
- `cluster_admission_rules` - (Optional) Per-cluster admission rules. An admission rule specifies either that all container images used in a pod creation request must be attested to by one or more attestors, that all pod creations will be allowed, or that all pod creations will be denied. There can be at most one admission rule per cluster spec.

Identifier format: `{{location}}.{{clusterId}}`. A location is either a compute zone (e.g. `us-central1-a`) or a region (e.g. `us-central1`). 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.

The `admission_whitelist_patterns` block supports:

- `name_pattern` - (Required) An image name pattern to whitelist, in the form `registry/path/to/image`. This supports a trailing * as a wildcard, but this is allowed only in text after the registry/ part.

The `cluster_admission_rules` block supports:

- `cluster` - (Required) The identifier for this object. Format specified above.
- `evaluation_mode` - (Required) How this admission rule will be evaluated.
- `require_attestations_by` - (Optional) The resource names of the attestors that must attest to a container image. If the attestor is in a different project from the policy, it should be specified in the format `projects/*/attestors/*`. Each attestor must exist before a policy can reference it. To add an attestor to a policy the principal issuing the policy change request must be able to read the attestor resource. Note: this field must be non-empty when the evaluation_mode field specifies REQUIRE_ATTESTATION, otherwise it must be empty.
- `enforcement_mode` - (Required) The action when a pod creation is denied by the admission rule.

» 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

Policy can be imported using any of these accepted formats:

```
$ terraform import google_binary_authorization_policy.default projects/{{project}}
$ terraform import google_binary_authorization_policy.default {{project}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_cloudbuild_trigger

Configuration for an automated build in response to source repository changes.

To get more information about Trigger, see:

- API documentation
- How-to Guides
  - Automating builds using build triggers

» Example Usage - Cloudbuild Trigger Filename

```
resource "google_cloudbuild_trigger" "filename-trigger" {
  trigger_template {
    branch_name = "master"
    repo_name   = "my-repo"
  }

  substitutions = {
    _FOO = "bar"
    _BAZ = "qux"
  }

  filename = "cloudbuild.yaml"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) Name of the trigger. Must be unique within the project.
- `description` - (Optional) Human-readable description of the trigger.
- `disabled` - (Optional) Whether the trigger is disabled or not. If true, the trigger will never result in a build.
- `substitutions` - (Optional) Substitutions data for Build resource.
- `filename` - (Optional) Path, from the source root, to a file whose contents is used for the template. Either a filename or build template must be provided.
- `ignored_files` - (Optional) ignoredFiles and includedFiles are file glob matches using http://godoc/pkg/path/filepath#Match extended with support for `**`. If ignoredFiles and changed files are both empty, then

they are not used to determine whether or not to trigger a build. If ignoredFiles is not empty, then we ignore any files that match any of the ignored_file globs. If the change has no files that are outside of the ignoredFiles globs, then we do not trigger a build.

- `included_files` - (Optional) ignoredFiles and includedFiles are file glob matches using http://godoc/pkg/path/filepath#Match extended with support for `**`. If any of the files altered in the commit pass the ignoredFiles filter and includedFiles is empty, then as far as this filter is concerned, we should trigger the build. If any of the files altered in the commit pass the ignoredFiles filter and includedFiles is not empty, then we make sure that at least one of those files matches a includedFiles glob. If not, then we do not trigger a build.

- `trigger_template` - (Optional) Template describing the types of source changes to trigger a build. Branch and tag names in trigger templates are interpreted as regular expressions. Any branch or tag change that matches that regular expression will trigger a build. One of `trigger_template` or `github` must be provided. Structure is documented below.

- `github` - (Optional, Beta) Describes the configuration of a trigger that creates a build whenever a GitHub event is received. One of `trigger_template` or `github` must be provided. Structure is documented below.

- `build` - (Optional) Contents of the build template. Either a filename or build template must be provided. 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.

The `trigger_template` block supports:

- `project_id` - (Optional) ID of the project that owns the Cloud Source Repository. If omitted, the project ID requesting the build is assumed.

- `repo_name` - (Optional) Name of the Cloud Source Repository. If omitted, the name "default" is assumed.

- `dir` - (Optional) Directory, relative to the source root, in which to run the build. This must be a relative path. If a step's dir is specified and is an absolute path, this value is ignored for that step's execution.

- `branch_name` - (Optional) Name of the branch to build. Exactly one a of branch name, tag, or commit SHA must be provided. This field is a regular expression.

- `tag_name` - (Optional) Name of the tag to build. Exactly one of a branch name, tag, or commit SHA must be provided. This field is a regular expression.

- `commit_sha` - (Optional) Explicit commit SHA to build. Exactly one of a branch name, tag, or commit SHA must be provided.

The `github` block supports:

- `owner` - (Optional) Owner of the repository. For example: The owner for https://github.com/googlecloudplatform/cloud-builders is "googlecloudplatform".
- `name` - (Optional) Name of the repository. For example: The name for https://github.com/googlecloudplatform/cloud-builders is "cloud-builders".
- `pull_request` - (Optional) filter to match changes in pull requests. Specify only one of pullRequest or push. Structure is documented below.
- `push` - (Optional) filter to match changes in refs, like branches or tags. Specify only one of pullRequest or push. Structure is documented below.

The `pull_request` block supports:

- `branch` - (Required) Regex of branches to match.
- `comment_control` - (Optional) Whether to block builds on a "/gcbrun" comment from a repository owner or collaborator.

The `push` block supports:

- `branch` - (Optional) Regex of branches to match. Specify only one of branch or tag.
- `tag` - (Optional) Regex of tags to match. Specify only one of branch or tag.

The `build` block supports:

- `tags` - (Optional) Tags for annotation of a Build. These are not docker tags.
- `images` - (Optional) A list of images to be pushed upon the successful completion of all build steps. The images are pushed using the builder service account's credentials. The digests of the pushed images will be stored in the Build resource's results field. If any of the images fail to be pushed, the build status is marked FAILURE.
- `timeout` - (Optional) Amount of time that this build should be allowed to run, to second granularity. If this amount of time elapses, work on the build will cease and the build status will be TIMEOUT. This timeout must be equal to or greater than the sum of the timeouts for build steps within the build. The expected format is the number of seconds followed by s. Default time is ten minutes (600s).
- `step` - (Required) The operations to be performed on the workspace. Structure is documented below.

The `step` block supports:

- `name` - (Required) The name of the container image that will run this particular build step. If the image is available in the host's Docker daemon's cache, it will be run directly. If not, the host will attempt to pull the image first, using the builder service account's credentials if necessary. The Docker daemon's cache will already have the latest versions of all of the officially supported build steps (https://github.com/GoogleCloudPlatform/cloud-builders). The Docker daemon will also have cached many of the layers for some popular images, like "ubuntu", "debian", but they will be refreshed at the time you attempt to use them. If you built an image in a previous build step, it will be stored in the host's Docker daemon's cache and is available to use as the name for a later build step.
- `args` - (Optional) A list of arguments that will be presented to the step when it is started. If the image used to run the step's container has an entrypoint, the args are used as arguments to that entrypoint. If the image does not define an entrypoint, the first element in args is used as the entrypoint, and the remainder will be used as arguments.
- `env` - (Optional) A list of environment variable definitions to be used when running a step. The elements are of the form "KEY=VALUE" for the environment variable "KEY" being given the value "VALUE".
- `id` - (Optional) Unique identifier for this build step, used in `wait_for` to reference this build step as a dependency.
- `entrypoint` - (Optional) Entrypoint to be used instead of the build step image's default entrypoint. If unset, the image's default entrypoint is used
- `dir` - (Optional) Working directory to use when running this step's container. If this value is a relative path, it is relative to the build's working directory. If this value is absolute, it may be outside the build's working directory, in which case the contents of the path may not be persisted across build step executions, unless a `volume` for that path is specified. If the build specifies a `RepoSource` with `dir` and a step with a `dir`, which specifies an absolute path, the `RepoSource dir` is ignored for the step's execution.
- `secret_env` - (Optional) A list of environment variables which are encrypted using a Cloud Key Management Service crypto key. These values must be specified in the build's `Secret`.
- `timeout` - (Optional) Time limit for executing this build step. If not defined, the step has no time limit and will be allowed to continue to run until either it completes or the build itself times out.
- `timing` - (Optional) Output only. Stores timing information for executing this build step.

- `volumes` - (Optional) List of volumes to mount into the build step. Each volume is created as an empty volume prior to execution of the build step. Upon completion of the build, volumes and their contents are discarded. Using a named volume in only one step is not valid as it is indicative of a build request with an incorrect configuration. Structure is documented below.
- `wait_for` - (Optional) The ID(s) of the step(s) that this build step depends on. This build step will not start until all the build steps in `wait_for` have completed successfully. If `wait_for` is empty, this build step will start when all previous build steps in the `Build.Steps` list have completed successfully.

The `volumes` block supports:

- `name` - (Required) Name of the volume to mount. Volume names must be unique per build step and must be valid names for Docker volumes. Each named volume must be used by at least two build steps.
- `path` - (Required) Path at which to mount the volume. Paths must be absolute and cannot conflict with other volume paths on the same build step or with certain reserved volume paths.

» Attributes Reference

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

- `trigger_id` - The unique identifier for the trigger.
- `create_time` - Time when the trigger was created.

» 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

Trigger can be imported using any of these accepted formats:

```
$ terraform import google_cloudbuild_trigger.default projects/{{project}}/triggers/{{trigger
$ terraform import google_cloudbuild_trigger.default {{project}}/{{trigger_id}}
$ terraform import google_cloudbuild_trigger.default {{trigger_id}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_composer_environment

An environment for running orchestration tasks.

Environments run Apache Airflow software on Google infrastructure.

To get more information about Environments, see:

- API documentation
- How-to Guides
  - Official Documentation
  - Configuring Shared VPC for Composer Environments
- Apache Airflow Documentation

Warning: We **STRONGLY** recommend you read the GCP guides as the Environment resource requires a long deployment process and involves several layers of GCP infrastructure, including a Kubernetes Engine cluster, Cloud Storage, and Compute networking resources. Due to limitations of the API, Terraform will not be able to automatically find or manage many of these underlying resources. In particular: * It can take up to one hour to create or update an environment resource. In addition, GCP may only detect some errors in configuration when they are used (e.g. ~40-50 minutes into the creation process), and is prone to limited error reporting. If you encounter confusing or uninformative errors, please verify your configuration is valid against GCP Cloud Composer before filing bugs against the Terraform provider. * **Environments create Google Cloud Storage buckets that do not get cleaned up automatically** on environment deletion. More about Composer's use of Cloud Storage.

» Example Usage

» Basic Usage

```
resource "google_composer_environment" "test" {
  name   = "my-composer-env"
  region = "us-central1"
```

```
}
```

» With GKE and Compute Resource Dependencies

NOTE To use service accounts, you need to give `role/composer.worker` to the service account on any resources that may be created for the environment (i.e. at a project level). This will probably require an explicit dependency on the IAM policy binding (see `google_project_iam_member` below).

```
resource "google_composer_environment" "test" {
  name   = "%s"
  region = "us-central1"
  config {
    node_count = 4

    node_config {
      zone         = "us-central1-a"
      machine_type = "n1-standard-1"

      network    = google_compute_network.test.self_link
      subnetwork = google_compute_subnetwork.test.self_link

      service_account = google_service_account.test.name
    }
  }

  depends_on = [google_project_iam_member.composer-worker]
}

resource "google_compute_network" "test" {
  name                    = "composer-test-network"
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "test" {
  name          = "composer-test-subnetwork"
  ip_cidr_range = "10.2.0.0/16"
  region        = "us-central1"
  network       = google_compute_network.test.self_link
}

resource "google_service_account" "test" {
  account_id   = "composer-env-account"
  display_name = "Test Service Account for Composer Environment"
}
```

```
resource "google_project_iam_member" "composer-worker" {
  role   = "roles/composer.worker"
  member = "serviceAccount:${google_service_account.test.email}"
}
```

» With Software (Airflow) Config

```
resource "google_composer_environment" "test" {
  name   = "%s"
  region = "us-central1"

  config {
    software_config {
      airflow_config_overrides = {
        core-load_example = "True"
      }

      pypi_packages = {
        numpy = ""
        scipy = "==1.1.0"
      }

      env_variables = {
        FOO = "bar"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the environment

- `config` - (Optional) Configuration parameters for this environment Structure is documented below.
- `labels` - (Optional) User-defined labels for this environment. The labels map can contain no more than 64 entries. Entries of the labels map are UTF8 strings that comply with the following restrictions: Label keys must be between 1 and 63 characters long and must conform to the following regular expression: `[a-z]([-a-z0-9]*[a-z0-9])?`. Label values must

be between 0 and 63 characters long and must conform to the regular expression `([a-z]([-a-z0-9]*[a-z0-9])?)?`. No more than 64 labels can be associated with a given environment. Both keys and values must be <= 128 bytes in size.

- `region` - (Optional) The location or Compute Engine region for the environment.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `config` block supports:

- `node_count` - (Optional) The number of nodes in the Kubernetes Engine cluster that will be used to run this environment.
- `node_config` - (Optional) The configuration used for the Kubernetes Engine cluster. Structure is documented below.
- `software_config` - (Optional) The configuration settings for software inside the environment. Structure is documented below.
- `private_environment_config` - (Optional) The configuration used for the Private IP Cloud Composer environment. Structure is documented below.

The `node_config` block supports:

- `zone` - (Required) The Compute Engine zone in which to deploy the VMs running the Apache Airflow software, specified as the zone name or relative resource name (e.g. "projects/{project}/zones/{zone}"). Must belong to the enclosing environment's project and region.
- `machine_type` - (Optional) The Compute Engine machine type used for cluster instances, specified as a name or relative resource name. For example: "projects/{project}/zones/{zone}/machineTypes/{machineType}". Must belong to the enclosing environment's project and region/zone.
- `network` - (Optional) The Compute Engine network to be used for machine communications, specified as a self-link, relative resource name (e.g. "projects/{project}/global/networks/{network}"), by name.

The network must belong to the environment's project. If unspecified, the "default" network ID in the environment's project is used. If a Custom Subnet Network is provided, subnetwork must also be provided.

- `subnetwork` - (Optional) The Compute Engine subnetwork to be used for machine communications, , specified as a self-link, relative resource name (e.g. "projects/{project}/regions/{region}/subnetworks/{subnetwork}"), or by name. If subnetwork is provided, network must also be provided and the subnetwork must belong to the enclosing environment's project and region.

- `disk_size_gb` - (Optional) The disk size in GB used for node VMs. Minimum size is 20GB. If unspecified, defaults to 100GB. Cannot be updated.
- `oauth_scopes` - (Optional) The set of Google API scopes to be made available on all node VMs. Cannot be updated. If empty, defaults to `["https://www.googleapis.com/auth/cloud-platform"]`
- `service_account` - (Optional) The Google Cloud Platform Service Account to be used by the node VMs. If a service account is not specified, the "default" Compute Engine service account is used. Cannot be updated. If given, note that the service account must have `roles/composer.worker` for any GCP resources created under the Cloud Composer Environment.
- `tags` - (Optional) The list of instance tags applied to all node VMs. Tags are used to identify valid sources or targets for network firewalls. Each tag within the list must comply with RFC1035. Cannot be updated.
- `ip_allocation_policy` - (Optional) Configuration for controlling how IPs are allocated in the GKE cluster. Structure is documented below. Cannot be updated.

The `software_config` block supports:

- `airflow_config_overrides` - (Optional) Apache Airflow configuration properties to override. Property keys contain the section and property names, separated by a hyphen, for example "core-dags_are_paused_at_creation".

Section names must not contain hyphens ("-"), opening square brackets ("["), or closing square brackets ("]"). The property name must not be empty and cannot contain "=" or ";". Section and property names cannot contain characters: "." Apache Airflow configuration property names must be written in snake_case. Property values can contain any character, and can be written in any lower/upper case format. Certain Apache Airflow configuration property values are blacklisted, and cannot be overridden.

- `pypi_packages` - (Optional) Custom Python Package Index (PyPI) packages to be installed in the environment. Keys refer to the lowercase package name (e.g. "numpy"). Values are the lowercase extras and version specifier (e.g. "==1.12.0", "[devel,gcp_api]", "[devel]>=1.8.2, <1.9.2"). To specify a package without pinning it to a version specifier, use the empty string as the value.
- `env_variables` - (Optional) Additional environment variables to provide to the Apache Airflow scheduler, worker, and webserver processes. Environment variable names must match the regular expression `[a-zA-Z_][a-zA-Z0-9_]*`. They cannot specify Apache Airflow software configuration overrides (they cannot match the regular expression `AIRFLOW__[A-Z0-9_]+__[A-Z0-9_]+`), and they cannot match any of the following reserved names: `AIRFLOW_HOME C_FORCE_ROOT`

CONTAINER_NAME DAGS_FOLDER GCP_PROJECT GCS_BUCKET GKE_CLUSTER_NAME SQL_DATABASE SQL_INSTANCE SQL_PASSWORD SQL_PROJECT SQL_REGION SQL_USER

- image_version (Optional) - The version of the software running in the environment. This encapsulates both the version of Cloud Composer functionality and the version of Apache Airflow. It must match the regular expression composer-[0-9]+\.[0-9]+(\.[0-9]+)?-airflow-[0-9]+\.[0-9]+(\.[0-9]+.*)?. The Cloud Composer portion of the version is a semantic version. The portion of the image version following 'airflow-' is an official Apache Airflow repository release name. See documentation for allowed release names.
- python_version (Optional) - The major version of Python used to run the Apache Airflow scheduler, worker, and webserver processes. Can be set to '2' or '3'. If not specified, the default is '2'. Cannot be updated.

The private_environment_config block supports:

- enable_private_endpoint - If true, access to the public endpoint of the GKE cluster is denied.
- master_ipv4_cidr_block - (Optional) The IP range in CIDR notation to use for the hosted master network. This range is used for assigning internal IP addresses to the cluster master or set of masters and to the internal load balancer virtual IP. This range must not overlap with any other ranges in use within the cluster's network. If left blank, the default value of '172.16.0.0/28' is used.

The ip_allocation_policy block supports:

- use_ip_aliases - (Required) Whether or not to enable Alias IPs in the GKE cluster. If true, a VPC-native cluster is created. Defaults to true if the ip_allocation_block is present in config.
- cluster_secondary_range_name - (Optional) The name of the cluster's secondary range used to allocate IP addresses to pods. Specify either cluster_secondary_range_name or cluster_ipv4_cidr_block but not both. This field is applicable only when use_ip_aliases is true.
- services_secondary_range_name - (Optional) The name of the services' secondary range used to allocate IP addresses to the cluster. Specify either services_secondary_range_name or services_ipv4_cidr_block but not both. This field is applicable only when use_ip_aliases is true.
- cluster_ipv4_cidr_block - (Optional) The IP address range used to allocate IP addresses to pods in the cluster. Set to blank to have GKE choose a range with the default size. Set to /netmask (e.g. /14) to have GKE choose a range with a specific netmask. Set to a CIDR notation (e.g. 10.96.0.0/14) from the RFC-1918 private networks (e.g. 10.0.0.0/8,

172.16.0.0/12, 192.168.0.0/16) to pick a specific range to use. Specify either `cluster_secondary_range_name` or `cluster_ipv4_cidr_block` but not both.

- `services_ipv4_cidr_block` - (Optional) The IP address range used to allocate IP addresses in this cluster. Set to blank to have GKE choose a range with the default size. Set to /netmask (e.g. /14) to have GKE choose a range with a specific netmask. Set to a CIDR notation (e.g. 10.96.0.0/14) from the RFC-1918 private networks (e.g. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) to pick a specific range to use. Specify either `services_secondary_range_name` or `services_ipv4_cidr_block` but not both.

» Attributes Reference

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

- `config.0.gke_cluster` - The Kubernetes Engine cluster used to run this environment.
- `config.0.dag_gcs_prefix` - The Cloud Storage prefix of the DAGs for this environment. Although Cloud Storage objects reside in a flat namespace, a hierarchical file tree can be simulated using '/'-delimited object name prefixes. DAG objects for this environment reside in a simulated directory with this prefix.
- `config.0.airflow_uri` - The URI of the Apache Airflow Web UI hosted within this environment.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 60 minutes.
- `update` - Default is 60 minutes.
- `delete` - Default is 6 minutes.

» Import

Environment can be imported using any of these accepted formats:

```
$ terraform import google_composer_environment.default projects/{{project}}/locations/{{regi
$ terraform import google_composer_environment.default {{project}}/{{region}}/{{name}}
$ terraform import google_composer_environment.default {{name}}
```

» google_cloudfunctions_function

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

Warning: As of November 1, 2019, newly created Functions are private-by-default and will require appropriate IAM permissions to be invoked. See below examples for how to set up the appropriate permissions, or view the Cloud Functions IAM resources for Cloud Functions.

» Example Usage - Public Function

```
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"
  runtime     = "nodejs10"

  available_memory_mb   = 128
  source_archive_bucket = google_storage_bucket.bucket.name
  source_archive_object = google_storage_bucket_object.archive.name
  trigger_http          = true
  entry_point           = "helloGET"
}

# IAM entry for all users to invoke the function
resource "google_cloudfunctions_function_iam_member" "invoker" {
  project        = google_cloudfunctions_function.function.project
  region         = google_cloudfunctions_function.function.region
  cloud_function = google_cloudfunctions_function.function.name

  role   = "roles/cloudfunctions.invoker"
  member = "allUsers"
}
```

» Example Usage - Single User

```
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"
  runtime     = "nodejs10"

  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"
  }

  environment_variables = {
    MY_ENV_VAR = "my-env-var-value"
  }
}

# IAM entry for a single user to invoke the function
resource "google_cloudfunctions_function_iam_member" "invoker" {
  project        = google_cloudfunctions_function.function.project
  region         = google_cloudfunctions_function.function.region
  cloud_function = google_cloudfunctions_function.function.name

  role   = "roles/cloudfunctions.invoker"
  member = "user:myFunctionInvoker@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A user-defined name of the function. Function names must be unique globally.
- `runtime` - (Required) The runtime in which the function is going to run. Eg. `"nodejs8"`, `"nodejs10"`, `"python37"`, `"go111"`.

- `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 the function that will be executed when the Google Cloud Function is triggered.
- `event_trigger` - (Optional) A source that fires events in response to a condition in another service. Structure is documented below. Cannot be used with `trigger_http`.
- `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`.
- `labels` - (Optional) A set of key/value label pairs to assign to the function.
- `service_account_email` - (Optional) If provided, the self-provided service account to run the function with.
- `environment_variables` - (Optional) A set of key/value environment variable pairs to assign to the function.
- `vpc_connector` - (Optional) The VPC Network Connector that this cloud function can connect to. It can be either the fully-qualified URI, or the short name of the network connector resource. The format of this field is `projects/*/locations/*/connectors/*`.
- `source_archive_bucket` - (Optional) The GCS bucket containing the zip archive which contains the function.
- `source_archive_object` - (Optional) The source archive object (file) in archive bucket.
- `source_repository` - (Optional) Represents parameters related to source repository where a function is hosted. Cannot be set alongside `source_archive_bucket` or `source_archive_object`. Structure is documented below.

- `max_instances` - (Optional) The limit on the maximum number of function instances that may coexist at a given time.

The `event_trigger` block supports:

- `event_type` - (Required) The type of event to observe. For example: `"google.storage.object.finalize"`. See the documentation on calling Cloud Functions for a full reference. Cloud Storage, Cloud Pub/Sub and Cloud Firestore triggers are supported at this time. Legacy triggers are supported, such as `"providers/cloud.storage/eventTypes/object.change"`, `"providers/cloud.pubsub/eventTypes/topic.publish"` and `"providers/cloud.firestore/eventTy`
- `resource` - (Required) Required. The name or partial URI of the resource from which to observe events. For example, `"myBucket"` or `"projects/my-project/topics/my-topic"`
- `failure_policy` - (Optional) Specifies policy for failed executions. Structure is documented below.

The `failure_policy` block supports:

- `retry` - (Required) Whether the function should be retried on failure. Defaults to `false`.

The `source_repository` block supports:

- `url` - (Required) The URL pointing to the hosted repository where the function is defined. There are supported Cloud Source Repository URLs in the following formats:
  - To refer to a specific commit: `https://source.developers.google.com/projects/*/repos/*/re`
  - To refer to a moveable alias (branch): `https://source.developers.google.com/projects/*/repo` To refer to HEAD, use the `master` moveable alias.
  - To refer to a specific fixed alias (tag): `https://source.developers.google.com/projects/*/repo`

» 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.
- `source_repository.0.deployed_url` - The URL pointing to the hosted repository where the function was defined at the time of deployment.
- `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.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 5 minutes.
- `update` - Default is 5 minutes.
- `delete` - Default is 5 minutes.

» Import

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

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

» IAM policy for CloudFunctionsCloudFunction

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

- `google_cloudfunctions_function_iam_policy`: Authoritative. Sets the IAM policy for the cloudfunction and replaces any existing policy already attached.
- `google_cloudfunctions_function_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 cloudfunction are preserved.
- `google_cloudfunctions_function_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the cloudfunction are preserved.

Note: `google_cloudfunctions_function_iam_policy` **cannot** be used in conjunction with `google_cloudfunctions_function_iam_binding` and `google_cloudfunctions_function_iam_member` or they will fight over what your policy should be.

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

» google_cloudfunctions_function_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
```

```
    ]
  }
}

resource "google_cloudfunctions_function_iam_policy" "editor" {
  project = "${google_cloudfunctions_function.function.project}"
  region = "${google_cloudfunctions_function.function.region}"
  cloud_function = "${google_cloudfunctions_function.function.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_cloudfunctions_function_iam_binding

```
resource "google_cloudfunctions_function_iam_binding" "editor" {
  project = "${google_cloudfunctions_function.function.project}"
  region = "${google_cloudfunctions_function.function.region}"
  cloud_function = "${google_cloudfunctions_function.function.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_cloudfunctions_function_iam_member

```
resource "google_cloudfunctions_function_iam_member" "editor" {
  project = "${google_cloudfunctions_function.function.project}"
  region = "${google_cloudfunctions_function.function.region}"
  cloud_function = "${google_cloudfunctions_function.function.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `cloud_function` - (Required) Used to find the parent resource to bind the IAM policy to
- `region` - (Optional) The location of this cloud function. Used to find the parent resource to bind the IAM policy to. If not specified, the value will be parsed from the identifier of the parent resource. If no region is

provided in the parent identifier and no region is specified, it is taken from the provider configuration.

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_cloudfunctions_function_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_cloudfunctions_function_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/locations/{{region}}/functions/{{cloud_function}}
- {{project}}/{{region}}/{{cloud_function}}

- {{region}}/{{cloud_function}}
- {{cloud_function}}

Any variables not passed in the import command will be taken from the provider configuration.

CloudFunctions cloudfunction IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_cloudfunctions_function_iam_member.editor "projects/{{project}}/locations/{{region}}/ roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_cloudfunctions_function_iam_binding.editor "projects/{{project}}/locations/{{region}}/functions/{{cloud_function}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_cloudfunctions_function_iam_policy.editor projects/{{project}}/locations/{{region}}/functions/{{cloud_function}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for CloudFunctionsCloudFunction

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

- `google_cloudfunctions_function_iam_policy`: Authoritative. Sets the IAM policy for the cloudfunction and replaces any existing policy already attached.
- `google_cloudfunctions_function_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 cloudfunction are preserved.
- `google_cloudfunctions_function_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the cloudfunction are preserved.

Note: `google_cloudfunctions_function_iam_policy` **cannot** be used in conjunction with `google_cloudfunctions_function_iam_binding` and `google_cloudfunctions_function_iam_member` or they will fight over what your policy should be.

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

» google_cloudfunctions_function_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_cloudfunctions_function_iam_policy" "editor" {
  project = "${google_cloudfunctions_function.function.project}"
  region = "${google_cloudfunctions_function.function.region}"
  cloud_function = "${google_cloudfunctions_function.function.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_cloudfunctions_function_iam_binding

```
resource "google_cloudfunctions_function_iam_binding" "editor" {
  project = "${google_cloudfunctions_function.function.project}"
  region = "${google_cloudfunctions_function.function.region}"
  cloud_function = "${google_cloudfunctions_function.function.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_cloudfunctions_function_iam_member

```
resource "google_cloudfunctions_function_iam_member" "editor" {
  project = "${google_cloudfunctions_function.function.project}"
```

```
  region = "${google_cloudfunctions_function.function.region}"
  cloud_function = "${google_cloudfunctions_function.function.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `cloud_function` - (Required) Used to find the parent resource to bind the IAM policy to
- `region` - (Optional) The location of this cloud function. Used to find the parent resource to bind the IAM policy to. If not specified, the value will be parsed from the identifier of the parent resource. If no region is provided in the parent identifier and no region is specified, it is taken from the provider configuration.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_cloudfunctions_function_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_cloudfunctions_function_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/locations/{{region}}/functions/{{cloud_function}}
- {{project}}/{{region}}/{{cloud_function}}
- {{region}}/{{cloud_function}}
- {{cloud_function}}

Any variables not passed in the import command will be taken from the provider configuration.

CloudFunctions cloudfunction IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_cloudfunctions_function_iam_member.editor "projects/{{project}}/locations/{{region}}/ roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_cloudfunctions_function_iam_binding.editor "projects/{{project}}/locations/{{region}}/functions/{{cloud_function}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_cloudfunctions_function_iam_policy.editor projects/{{project}}/locations/{{region}}/functions/{{cloud_function}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for CloudFunctionsCloudFunction

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

- `google_cloudfunctions_function_iam_policy`: Authoritative. Sets the IAM policy for the cloudfunction and replaces any existing policy already attached.
- `google_cloudfunctions_function_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 cloudfunction are preserved.
- `google_cloudfunctions_function_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the cloudfunction are preserved.

Note: `google_cloudfunctions_function_iam_policy` **cannot** be used in conjunction with `google_cloudfunctions_function_iam_binding` and `google_cloudfunctions_function_iam_member` or they will fight over what your policy should be.

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

» google_cloudfunctions_function_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_cloudfunctions_function_iam_policy" "editor" {
  project = "${google_cloudfunctions_function.function.project}"
  region = "${google_cloudfunctions_function.function.region}"
  cloud_function = "${google_cloudfunctions_function.function.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_cloudfunctions_function_iam_binding

```
resource "google_cloudfunctions_function_iam_binding" "editor" {
```

```
  project = "${google_cloudfunctions_function.function.project}"
  region = "${google_cloudfunctions_function.function.region}"
  cloud_function = "${google_cloudfunctions_function.function.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_cloudfunctions_function_iam_member

```
resource "google_cloudfunctions_function_iam_member" "editor" {
  project = "${google_cloudfunctions_function.function.project}"
  region = "${google_cloudfunctions_function.function.region}"
  cloud_function = "${google_cloudfunctions_function.function.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `cloud_function` - (Required) Used to find the parent resource to bind the IAM policy to
- `region` - (Optional) The location of this cloud function. Used to find the parent resource to bind the IAM policy to. If not specified, the value will be parsed from the identifier of the parent resource. If no region is provided in the parent identifier and no region is specified, it is taken from the provider configuration.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_cloudfunctions_function_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`

- `policy_data` - (Required only by `google_cloudfunctions_function_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/locations/{{region}}/functions/{{cloud_function}}
- {{project}}/{{region}}/{{cloud_function}}
- {{region}}/{{cloud_function}}
- {{cloud_function}}

Any variables not passed in the import command will be taken from the provider configuration.

CloudFunctions cloudfunction IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_cloudfunctions_function_iam_member.editor "projects/{{project}}/locations/{{region}}/ roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_cloudfunctions_function_iam_binding.editor "projects/{{project}}/locations/{{region}}/functions/{{cloud_function}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_cloudfunctions_function_iam_policy.editor projects/{{project}}/locations/{{region}}/functions/{{cloud_function}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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_configs {
    pubsub_topic_name = google_pubsub_topic.default-telemetry.id
  }

  state_notification_config = {
    pubsub_topic_name = google_pubsub_topic.default-devicestatus.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_configs` - (Optional) List of configurations for event notification, such as PubSub topics to publish device events to. 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_configs` block supports:

- `pubsub_topic_name` - (Required) PubSub topic name to publish device events.
- `subfolder_matches` - (Optional) If the subfolder name matches this string exactly, this configuration will be used. The string must not include the leading '/' character. If empty, all strings are matched. Empty value can only be used for the last `event_notification_configs` item.

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
```

» google_billing_account___iam_binding

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

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

Note: On create, this resource will overwrite members of any existing roles. Use `terraform import` and inspect the `terraform plan` output to ensure your existing members are preserved.

» Example Usage

```
resource "google_billing_account_iam_binding" "binding" {
  billing_account_id = "00AA00-000AAA-00AA0A"
  role               = "roles/billing.viewer"

  members = [
    "user:alice@gmail.com",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `billing_account_id` - (Required) The billing account id.
- `role` - (Required) The role that should be applied.
- `members` - (Required) A list of users that the role should apply to. For more details on format and restrictions see https://cloud.google.com/billing/reference/rest/v1/Policy#Binding

» Attributes Reference

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

- `etag` - (Computed) The etag of the billing account'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 `billing_account_id` and role, e.g.

```
$ terraform import google_billing_account_iam_binding.binding "your-billing-account-id roles
```

» google_billing_account_iam_member

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

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

» Example Usage

```
resource "google_billing_account_iam_member" "binding" {
  billing_account_id = "00AA00-000AAA-00AA0A"
  role               = "roles/billing.viewer"
  member             = "user:alice@gmail.com"
}
```

» Argument Reference

The following arguments are supported:

- `billing_account_id` - (Required) The billing account id.
- `role` - (Required) The role that should be applied.
- `member` - (Required) The user that the role should apply to. For more details on format and restrictions see https://cloud.google.com/billing/reference/rest/v1/Policy#Binding

» Attributes Reference

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

- `etag` - (Computed) The etag of the billing account'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 `billing_account_id`, role, and member identity, e.g.

```
$ terraform import google_billing_account_iam_member.binding "your-billing-account-id roles/
```

» google_billing_account_iam_policy

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

Warning: Billing accounts have a default user that can be **overwritten** by use of this resource. The safest alternative is to use multiple `google_billing_account_iam_binding` resources. 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_billing_account_iam_member` or `google_billing_account_iam_binding` or they will fight over what your policy should be.

» Example Usage

```
resource "google_billing_account_iam_policy" "policy" {
  billing_account_id = "00AA00-000AAA-00AA0A"
  policy_data        = data.google_iam_policy.admin.policy_data
}

data "google_iam_policy" "admin" {
  binding {
    role = "roles/billing.viewer"

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

» Argument Reference

The following arguments are supported:

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

» Import

```
$ terraform import google_billing_account_iam_policy.policy billing-account-id
```

» google_billing_budget

Budget configuration for a billing account.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about Budget, see:

- API documentation
- How-to Guides
  - Creating a budget

» Example Usage - Billing Budget Basic

```
data "google_billing_account" "account" {
  provider = google-beta
  billing_account = "000000-0000000-0000000-000000"
}

resource "google_billing_budget" "budget" {
  provider = google-beta
  billing_account = data.google_billing_account.account.id
  display_name = "Example Billing Budget"
  amount {
    specified_amount {
      currency_code = "USD"
      units = "100000"
    }
  }
  threshold_rules {
      threshold_percent =  0.5
  }
}
```

» Example Usage - Billing Budget Filter

```
data "google_billing_account" "account" {
  provider = google-beta
  billing_account = "000000-0000000-0000000-000000"
}

resource "google_billing_budget" "budget" {
  provider = google-beta
  billing_account = data.google_billing_account.account.id
  display_name = "Example Billing Budget"

  budget_filter {
    projects = ["projects/my-project-name"]
    credit_types_treatment = "EXCLUDE_ALL_CREDITS"
    services = ["services/24E6-581D-38E5"] # Bigquery
  }

  amount {
    specified_amount {
      currency_code = "USD"
      units = "100000"
    }
  }

  threshold_rules {
    threshold_percent = 0.5
  }
  threshold_rules {
    threshold_percent = 0.9
    spend_basis = "FORECASTED_SPEND"
  }
}
```

» Argument Reference

The following arguments are supported:

- `amount` - (Required) The budgeted amount for each usage period. Structure is documented below.
- `threshold_rules` - (Required) Rules that trigger alerts (notifications of thresholds being crossed) when spend exceeds the specified percentages of the budget. Structure is documented below.
- `billing_account` - (Required) ID of the billing account to set a budget

on.

The `budget_filter` block supports:

- `projects` - (Optional) A set of projects of the form projects/{project_id}, specifying that usage from only this set of projects should be included in the budget. If omitted, the report will include all usage for the billing account, regardless of which project the usage occurred on. Only zero or one project can be specified currently.
- `credit_types_treatment` - (Optional) Specifies how credits should be treated when determining spend for threshold calculations.
- `services` - (Optional) A set of services of the form services/{service_id}, specifying that usage from only this set of services should be included in the budget. If omitted, the report will include usage for all the services. The service names are available through the Catalog API: https://cloud.google.com/billing/v1/how-tos/catalog-api.

The `amount` block supports:

- `specified_amount` - (Required) A specified amount to use as the budget. currencyCode is optional. If specified, it must match the currency of the billing account. The currencyCode is provided on output. Structure is documented below.

The `specified_amount` block supports:

- `currency_code` - (Optional) The 3-letter currency code defined in ISO 4217.
- `units` - (Optional) The whole units of the amount. For example if currencyCode is "USD", then 1 unit is one US dollar.
- `nanos` - (Optional) Number of nano (10^{-9}) units of the amount. The value must be between -999,999,999 and +999,999,999 inclusive. If units is positive, nanos must be positive or zero. If units is zero, nanos can be positive, zero, or negative. If units is negative, nanos must be negative or zero. For example $-1.75 is represented as units=-1 and nanos=-750,000,000.

The `threshold_rules` block supports:

- `threshold_percent` - (Required) Send an alert when this threshold is exceeded. This is a 1.0-based percentage, so 0.5 = 50%. Must be >= 0.
- `spend_basis` - (Optional) The type of basis used to determine if spend has passed the threshold.

The `all_updates_rule` block supports:

- `pubsub_topic` - (Required) The name of the Cloud Pub/Sub topic where budget related messages will be published, in the form

projects/{project_id}/topics/{topic_id}. Updates are sent at regular intervals to the topic.

- `schema_version` - (Optional) The schema version of the notification. Only "1.0" is accepted. It represents the JSON schema as defined in https://cloud.google.com/billing/docs/how-to/budgets#notification_format.

- `display_name` - (Optional) User data for display name in UI. Must be <= 60 chars.
- `budget_filter` - (Optional) Filters that define which resources are used to compute the actual spend against the budget. Structure is documented below.
- `all_updates_rule` - (Optional) Defines notifications that are sent on every update to the billing account's spend, regardless of the thresholds defined using threshold rules. Structure is documented below.

» Attributes Reference

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

- `name` - Resource name of the budget. The resource name implies the scope of a budget. Values are of the form billingAccounts/{billingAccountId}/budgets/{budgetId}.

» 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

Budget can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_billing_budget.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» 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's id, 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.

Note: On create, this resource will overwrite members of any existing roles. Use `terraform import` and inspect the `terraform plan` output to ensure your existing members are preserved.

» 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:alice@gmail.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 identities that will be granted the privilege in the `role`. Each entry can have one of the following values:
  - **user:{emailid}**: An email address that is associated with a specific Google account. For example, alice@gmail.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 G Suite domain (primary, instead of alias) name that represents all the users of that domain. For example, google.com or example.com.
  - For more details on format and restrictions see https://cloud.google.com/billing/reference/rest/v1/Policy#Binding
- `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:alice@gmail.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`. For more details on format and restrictions see https://cloud.google.com/billing/reference/rest/v1/Policy#Binding 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 G Suite domain (primary, instead of alias) 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 member identity e.g.

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

» 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.

» Import

A policy can be imported using the `folder`, e.g.

```
$ terraform import google_folder_iam_policy.my-folder-policy {{folder_id}}
```

» 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 constraint:

```
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_value = "compute.googleapis.com"

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

To restore the default folder organization policy, use the following instead:

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

  restore_policy {
    default = true
  }
}
```

» 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.
- `restore_policy` - (Optional) A restore policy is a constraint to restore the default policy. Structure is documented below.

Note: If none of [`boolean_policy`, `list_policy`, `restore_policy`] are defined the policy for a given constraint will effectively be unset. This is represented in the UI as the constraint being 'Inherited'.

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_value` - (Optional) The Google Cloud Console will try to default to a configuration that matches the value specified in this field.
- `inherit_from_parent` - (Optional) If set to true, the values from the effective Policy of the parent resource are inherited, meaning the values set in this Policy are added to the values inherited up the hierarchy.

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.

The `restore_policy` block supports:

- `default` - (Required) May only be set to true. If set, then the default Policy is restored.

» 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

Folder organization policies can be imported using any of the follow formats:

```
$ terraform import google_folder_organization_policy.policy folders/folder-1234/constraints/
$ terraform import google_folder_organization_policy.policy folder-1234/serviceuser.services
```

» 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 constraint:

```
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_value = "compute.googleapis.com"

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

To restore the default organization policy, use the following instead:

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

  restore_policy {
    default = true
  }
}
```

» 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.
- `restore_policy` - (Optional) A restore policy is a constraint to restore the default policy. Structure is documented below.

Note: If none of [`boolean_policy`, `list_policy`, `restore_policy`] are defined the policy for a given constraint will effectively be unset. This is represented in the UI as the constraint being 'Inherited'.

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_value` - (Optional) The Google Cloud Console will try to default to a configuration that matches the value specified in this field.
- `inherit_from_parent` - (Optional) If set to true, the values from the effective Policy of the parent resource are inherited, meaning the values set in this Policy are added to the values inherited up the hierarchy.

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.

The `restore_policy` block supports:

- `default` - (Required) May only be set to true. If set, then the default Policy is restored.

» 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 `constraint`, e.g.

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

It is all right if the constraint contains a slash, as in the example above.

» google_organization_iam_audit_config

Allows management of audit logging config for a given service for a Google Cloud Platform Organization.

```
resource "google_organization_iam_audit_config" "config" {
  org_id = "your-organization-id"
  service = "allServices"
  audit_log_config {
    log_type = "DATA_READ"
    exempted_members = [
      "user:joebloggs@hashicorp.com",
    ]
  }
}
```

» Argument Reference

The following arguments are supported:

- `org_id` - (Required) The numeric ID of the organization in which you want to manage the audit logging config.
- `service` - (Required) Service which will be enabled for audit logging. The special value `allServices` covers all services. Note that if there are google_organization_iam_audit_config resources covering both `allServices` and a specific service then the union of the two AuditConfigs is used for that service: the `log_types` specified in each `audit_log_config` are enabled, and the `exempted_members` in each `audit_log_config` are exempted.
- `audit_log_config` - (Required) The configuration for logging of each type of permission. This can be specified multiple times. Structure is documented below.

The `audit_log_config` block supports:

- `log_type` - (Required) Permission type for which logging is to be configured. Must be one of `DATA_READ`, `DATA_WRITE`, or `ADMIN_READ`.
- `exempted_members` - (Optional) Identities that do not cause logging for this type of permission. 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 G Suite domain (primary, instead of alias) name that represents all the users of that domain. For example, google.com or example.com.

» Import

IAM audit config imports use the identifier of the resource in question and the service, e.g.

```
terraform import google_organization_iam_audit_config.config "your-organization-id foo.googl
```

» 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.

Note: On create, this resource will overwrite members of any existing roles. Use `terraform import` and inspect the `terraform plan` output to ensure your existing members are preserved.

» Example Usage

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

  members = [
    "user:alice@gmail.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. For more details on format and restrictions see https://cloud.google.com/billing/reference/rest/v1/Policy#Binding

» 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.

Warning: Note that custom roles in GCP have the concept of a soft-delete. There are two issues that may arise from this and how roles are propagated. 1) creating a role may involve undeleting and then updating a role with the same name, possibly causing confusing behavior between undelete and update. 2) A deleted role is permanently deleted after 7 days, but it can take up to 30 more days (i.e. between 7 and 37 days after deletion) before the role name is made available again. This means a deleted role that has been deleted for more than

7 days cannot be changed at all by Terraform, and new roles cannot share that name.

» 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.

» Attributes Reference

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

- `deleted` - (Optional) The current deleted state of the role.

» Import

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

```
$ terraform import google_organization_iam_custom_role.my-custom-role organizations/1234567
```

» 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:alice@gmail.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. For more details on format and restrictions see https://cloud.google.com/billing/reference/rest/v1/Policy#Binding

» 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 member identity, e.g.

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

» 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.
- `labels` - (Optional) A set of key/value label pairs to assign to the project.
- `auto_create_network` - (Optional) Create the 'default' network automatically. Default `true`. If set to `false`, the default network will be deleted. Note that, for quota purposes, you will still need to have 1 network slot available to create the project successfully, even if you set `auto_create_network` to `false`, since the network will exist momentarily.

» Attributes Reference

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

- `number` - The numeric identifier of the project.

» Import

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

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

» IAM policy for projects

Four different resources help you manage your IAM policy for a project. Each of these resources serves a different use case:

- `google_project_iam_policy`: Authoritative. Sets the IAM policy for the project and replaces any existing policy already attached.
- `google_project_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 project are preserved.
- `google_project_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the project are preserved.
- `google_project_iam_audit_config`: Authoritative for a given service. Updates the IAM policy to enable audit logging for the given service.

Note: `google_project_iam_policy` **cannot** be used in conjunction with `google_project_iam_binding`, `google_project_iam_member`, or `google_project_iam_audit_config` or they will fight over what your policy should be.

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

» google_project_iam_policy

Be careful! You can accidentally lock yourself out of your project using this resource. Deleting a `google_project_iam_policy` removes access from anyone without organization-level access to the project. Proceed with caution. It's not recommended to use `google_project_iam_policy` with your provider project

to avoid locking yourself out, and it should generally only be used with projects fully managed by Terraform.

```
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",
    ]
  }
}
```

With IAM Conditions (beta):

```
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",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}
```

» google_project_iam_binding

Note: If `role` is set to `roles/owner` and you don't specify a user or service account you have access to in `members`, you can lock yourself out of your project.

```
resource "google_project_iam_binding" "project" {
```

```
  project = "your-project-id"
  role    = "roles/editor"

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

With IAM Conditions (beta):

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

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_project_iam_member

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

With IAM Conditions (beta):

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_project_iam_audit_config

```
resource "google_project_iam_audit_config" "project" {
  project = "your-project-id"
  service = "allServices"
  audit_log_config {
    log_type = "DATA_READ"
    exempted_members = [
      "user:joebloggs@hashicorp.com",
    ]
  }
}
```

» Argument Reference

The following arguments are supported:

- `member/members` - (Required except for google_project_iam_audit_config) Identities that will be granted the privilege in `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 G Suite domain (primary, instead of alias) name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required except for google_project_iam_audit_config) 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}`.
- `policy_data` - (Required only by `google_project_iam_policy`) 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 all policies from the project, locking out users without organization-level access.

- `project` - (Optional) The project ID. If not specified for `google_project_iam_binding`, `google_project_iam_member`, or `google_project_iam_audit_config`, uses the ID of the project configured with the provider. Required for `google_project_iam_policy` - you must explicitly set the project, and it will not be inferred from the provider.
- `service` - (Required only by google_project_iam_audit_config) Service which will be enabled for audit logging. The special value `allServices` covers all services. Note that if there are google_project_iam_audit_config resources covering both `allServices` and a specific service then the union of the two AuditConfigs is used for that service: the `log_types` specified in each `audit_log_config` are enabled, and the `exempted_members` in each `audit_log_config` are exempted.
- `audit_log_config` - (Required only by google_project_iam_audit_config) The configuration for logging of each type of permission. This can be specified multiple times. Structure is documented below.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `audit_log_config` block supports:

- `log_type` - (Required) Permission type for which logging is to be configured. Must be one of `DATA_READ`, `DATA_WRITE`, or `ADMIN_READ`.
- `exempted_members` - (Optional) Identities that do not cause logging for this type of permission. The format is the same as that for `members`.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 member e.g.

```
$ terraform import google_project_iam_member.my_project "your-project-id roles/viewer user:f
```

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

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

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

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

IAM audit config imports use the identifier of the resource in question and the service, e.g.

```
terraform import google_project_iam_audit_config.my_project "your-project-id foo.googleapis.
```

» IAM policy for projects

Four different resources help you manage your IAM policy for a project. Each of these resources serves a different use case:

- `google_project_iam_policy`: Authoritative. Sets the IAM policy for the project and replaces any existing policy already attached.
- `google_project_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 project are preserved.
- `google_project_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the project are preserved.
- `google_project_iam_audit_config`: Authoritative for a given service. Updates the IAM policy to enable audit logging for the given service.

Note: `google_project_iam_policy` **cannot** be used in conjunction with `google_project_iam_binding`, `google_project_iam_member`, or `google_project_iam_audit_config` or they will fight over what your policy should be.

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

» google_project_iam_policy

Be careful! You can accidentally lock yourself out of your project using this resource. Deleting a `google_project_iam_policy` removes access from anyone without organization-level access to the project. Proceed with caution. It's not recommended to use `google_project_iam_policy` with your provider project to avoid locking yourself out, and it should generally only be used with projects fully managed by Terraform.

```
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",
    ]
  }
}
```

With IAM Conditions (beta):

```
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",
    ]
```

```
    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}
```

» google_project_iam_binding

Note: If `role` is set to `roles/owner` and you don't specify a user or service account you have access to in `members`, you can lock yourself out of your project.

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

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

With IAM Conditions (beta):

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

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_project_iam_member

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

```
  member  = "user:jane@example.com"
}
```

With IAM Conditions (beta):

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_project_iam_audit_config

```
resource "google_project_iam_audit_config" "project" {
  project = "your-project-id"
  service = "allServices"
  audit_log_config {
    log_type = "DATA_READ"
    exempted_members = [
      "user:joebloggs@hashicorp.com",
    ]
  }
}
```

» Argument Reference

The following arguments are supported:

- `member/members` - (Required except for google_project_iam_audit_config) Identities that will be granted the privilege in `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 G Suite domain (primary, instead of alias) name that represents all the users of that domain. For example, google.com or example.com.

- `role` - (Required except for google_project_iam_audit_config) 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}`.

- `policy_data` - (Required only by `google_project_iam_policy`) 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 all policies from the project, locking out users without organization-level access.

- `project` - (Optional) The project ID. If not specified for `google_project_iam_binding`, `google_project_iam_member`, or `google_project_iam_audit_config`, uses the ID of the project configured with the provider. Required for `google_project_iam_policy` - you must explicitly set the project, and it will not be inferred from the provider.

- `service` - (Required only by google_project_iam_audit_config) Service which will be enabled for audit logging. The special value `allServices` covers all services. Note that if there are google_project_iam_audit_config resources covering both `allServices` and a specific service then the union of the two AuditConfigs is used for that service: the `log_types` specified in each `audit_log_config` are enabled, and the `exempted_members` in each `audit_log_config` are exempted.

- `audit_log_config` - (Required only by google_project_iam_audit_config) The configuration for logging of each type of permission. This can be specified multiple times. Structure is documented below.

- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `audit_log_config` block supports:

- `log_type` - (Required) Permission type for which logging is to be configured. Must be one of `DATA_READ`, `DATA_WRITE`, or `ADMIN_READ`.

- `exempted_members` - (Optional) Identities that do not cause logging for this type of permission. The format is the same as that for `members`.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 member e.g.

```
$ terraform import google_project_iam_member.my_project "your-project-id roles/viewer user:f
```

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

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

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

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

IAM audit config imports use the identifier of the resource in question and the service, e.g.

```
terraform import google_project_iam_audit_config.my_project "your-project-id foo.googleapis.
```

» IAM policy for projects

Four different resources help you manage your IAM policy for a project. Each of these resources serves a different use case:

- `google_project_iam_policy`: Authoritative. Sets the IAM policy for the project and replaces any existing policy already attached.
- `google_project_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 project are preserved.
- `google_project_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the project are preserved.
- `google_project_iam_audit_config`: Authoritative for a given service. Updates the IAM policy to enable audit logging for the given service.

Note: `google_project_iam_policy` **cannot** be used in conjunction with `google_project_iam_binding`, `google_project_iam_member`, or `google_project_iam_audit_config` or they will fight over what your policy should be.

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

» google_project_iam_policy

Be careful! You can accidentally lock yourself out of your project using this resource. Deleting a `google_project_iam_policy` removes access from anyone without organization-level access to the project. Proceed with caution. It's not recommended to use `google_project_iam_policy` with your provider project to avoid locking yourself out, and it should generally only be used with projects fully managed by Terraform.

```
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",
    ]
```

```
  }
}
```

With IAM Conditions (beta):

```
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",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}
```

» google_project_iam_binding

Note: If `role` is set to `roles/owner` and you don't specify a user or service account you have access to in `members`, you can lock yourself out of your project.

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

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

With IAM Conditions (beta):

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

  members = [
```

```
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_project_iam_member

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

With IAM Conditions (beta):

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_project_iam_audit_config

```
resource "google_project_iam_audit_config" "project" {
  project = "your-project-id"
  service = "allServices"
  audit_log_config {
    log_type = "DATA_READ"
    exempted_members = [
      "user:joebloggs@hashicorp.com",
    ]
  }
}
```

» Argument Reference

The following arguments are supported:

- `member/members` - (Required except for google_project_iam_audit_config) Identities that will be granted the privilege in `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 G Suite domain (primary, instead of alias) name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required except for google_project_iam_audit_config) 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}`.
- `policy_data` - (Required only by `google_project_iam_policy`) 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 all policies from the project, locking out users without organization-level access.
- `project` - (Optional) The project ID. If not specified for `google_project_iam_binding`, `google_project_iam_member`, or `google_project_iam_audit_config`, uses the ID of the project configured with the provider. Required for `google_project_iam_policy` - you must explicitly set the project, and it will not be inferred from the provider.
- `service` - (Required only by google_project_iam_audit_config) Service which will be enabled for audit logging. The special value `allServices` covers all services. Note that if there are google_project_iam_audit_config resources covering both `allServices` and a specific service then the union of the two AuditConfigs is used for that service: the `log_types` specified in each `audit_log_config` are enabled, and the `exempted_members` in each `audit_log_config` are exempted.

- `audit_log_config` - (Required only by google_project_iam_audit_config) The configuration for logging of each type of permission. This can be specified multiple times. Structure is documented below.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `audit_log_config` block supports:

- `log_type` - (Required) Permission type for which logging is to be configured. Must be one of `DATA_READ`, `DATA_WRITE`, or `ADMIN_READ`.
- `exempted_members` - (Optional) Identities that do not cause logging for this type of permission. The format is the same as that for `members`.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 member e.g.

```
$ terraform import google_project_iam_member.my_project "your-project-id roles/viewer user:f
```

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

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

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

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

IAM audit config imports use the identifier of the resource in question and the service, e.g.

```
terraform import google_project_iam_audit_config.my_project "your-project-id foo.googleapis.
```

» IAM policy for projects

Four different resources help you manage your IAM policy for a project. Each of these resources serves a different use case:

- `google_project_iam_policy`: Authoritative. Sets the IAM policy for the project and replaces any existing policy already attached.
- `google_project_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 project are preserved.
- `google_project_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the project are preserved.
- `google_project_iam_audit_config`: Authoritative for a given service. Updates the IAM policy to enable audit logging for the given service.

Note: `google_project_iam_policy` **cannot** be used in conjunction with `google_project_iam_binding`, `google_project_iam_member`, or `google_project_iam_audit_config` or they will fight over what your policy should be.

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

» google_project_iam_policy

Be careful! You can accidentally lock yourself out of your project using this resource. Deleting a `google_project_iam_policy` removes access from anyone without organization-level access to the project. Proceed with caution. It's not recommended to use `google_project_iam_policy` with your provider project

to avoid locking yourself out, and it should generally only be used with projects fully managed by Terraform.

```
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",
    ]
  }
}
```

With IAM Conditions (beta):

```
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",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}
```

» google_project_iam_binding

Note: If `role` is set to `roles/owner` and you don't specify a user or service account you have access to in `members`, you can lock yourself out of your project.

```
resource "google_project_iam_binding" "project" {
```

```
  project = "your-project-id"
  role    = "roles/editor"

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

With IAM Conditions (beta):

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

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_project_iam_member

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

With IAM Conditions (beta):

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_project_iam_audit_config

```
resource "google_project_iam_audit_config" "project" {
  project = "your-project-id"
  service = "allServices"
  audit_log_config {
    log_type = "DATA_READ"
    exempted_members = [
      "user:joebloggs@hashicorp.com",
    ]
  }
}
```

» Argument Reference

The following arguments are supported:

- `member/members` - (Required except for google_project_iam_audit_config) Identities that will be granted the privilege in `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 G Suite domain (primary, instead of alias) name that represents all the users of that domain. For example, google.com or example.com.
- `role` - (Required except for google_project_iam_audit_config) 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}`.
- `policy_data` - (Required only by `google_project_iam_policy`) 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 all policies from the project, locking out users without organization-level access.

- `project` - (Optional) The project ID. If not specified for `google_project_iam_binding`, `google_project_iam_member`, or `google_project_iam_audit_config`, uses the ID of the project configured with the provider. Required for `google_project_iam_policy` - you must explicitly set the project, and it will not be inferred from the provider.
- `service` - (Required only by google_project_iam_audit_config) Service which will be enabled for audit logging. The special value `allServices` covers all services. Note that if there are google_project_iam_audit_config resources covering both `allServices` and a specific service then the union of the two AuditConfigs is used for that service: the `log_types` specified in each `audit_log_config` are enabled, and the `exempted_members` in each `audit_log_config` are exempted.
- `audit_log_config` - (Required only by google_project_iam_audit_config) The configuration for logging of each type of permission. This can be specified multiple times. Structure is documented below.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `audit_log_config` block supports:

- `log_type` - (Required) Permission type for which logging is to be configured. Must be one of `DATA_READ`, `DATA_WRITE`, or `ADMIN_READ`.
- `exempted_members` - (Optional) Identities that do not cause logging for this type of permission. The format is the same as that for `members`.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 member e.g.

```
$ terraform import google_project_iam_member.my_project "your-project-id roles/viewer user:f
```

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

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

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

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

IAM audit config imports use the identifier of the resource in question and the service, e.g.

```
terraform import google_project_iam_audit_config.my_project "your-project-id foo.googleapis.
```

» google_project_iam_custom_role

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

Warning: Note that custom roles in GCP have the concept of a soft-delete. There are two issues that may arise from this and how roles are propagated. 1) creating a role may involve undeleting and then updating a role with the same name, possibly causing confusing behavior between undelete and update. 2) A deleted role is permanently deleted after 7 days, but it can take up to 30 more days (i.e. between 7 and 37 days after deletion) before the role name is made available again. This means a deleted role that has been deleted for more than 7 days cannot be changed at all by Terraform, and new roles cannot share that name.

» 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.

» Attributes Reference

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

- `deleted` - (Optional) The current deleted state of the role.

» 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_organization_policy

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

» Example Usage

To set policy with a boolean constraint:

```
resource "google_project_organization_policy" "serial_port_policy" {
  project    = "your-project-id"
  constraint = "compute.disableSerialPortAccess"

  boolean_policy {
    enforced = true
  }
}
```

To set a policy with a list constraint:

```
resource "google_project_organization_policy" "services_policy" {
  project    = "your-project-id"
  constraint = "serviceuser.services"

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

Or to deny some services, use the following instead:

```
resource "google_project_organization_policy" "services_policy" {
  project    = "your-project-id"
  constraint = "serviceuser.services"

  list_policy {
    suggested_value = "compute.googleapis.com"

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

To restore the default project organization policy, use the following instead:

```
resource "google_project_organization_policy" "services_policy" {
  project    = "your-project-id"
  constraint = "serviceuser.services"

  restore_policy {
    default = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Required) The project id of the project 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.
- `restore_policy` - (Optional) A restore policy is a constraint to restore the default policy. Structure is documented below.

Note: If none of [`boolean_policy`, `list_policy`, `restore_policy`] are defined the policy for a given constraint will effectively be unset. This is represented in the UI as the constraint being 'Inherited'.

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_value` - (Optional) The Google Cloud Console will try to default to a configuration that matches the value specified in this field.

- `inherit_from_parent` - (Optional) If set to true, the values from the effective Policy of the parent resource are inherited, meaning the values set in this Policy are added to the values inherited up the hierarchy.

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.

The `restore_policy` block supports:

- `default` - (Required) May only be set to true. If set, then the default Policy is restored.

» 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

Project organization policies can be imported using any of the follow formats:

```
$ terraform import google_project_organization_policy.policy projects/test-project:constrain
$ terraform import google_project_organization_policy.policy test-project:constraints/servic
$ terraform import google_project_organization_policy.policy test-project:serviceuser.servic
```

» 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 services list`.

» Example Usage

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

  disable_dependent_services = true
}
```

» 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_dependent_services` - (Optional) If `true`, services that are enabled and which depend on this service should also be disabled when this service is destroyed. If `false` or unset, an error will be generated if any enabled services depend on this service when destroying it.
- `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.

» Import

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

```
$ terraform import google_project_service.my_project your-project-id/iam.googleapis.com
```

Note that unlike other resources that fail if they already exist, `terraform apply` can be successfully used to re-enable already enabled services. This means that when importing existing resources into Terraform, you can either import the `google_project_service` resources or treat them as new infrastructure and run `terraform apply` to re-enable them and add them to state.

» 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.

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.

» Example Usage

```
resource "google_project_usage_export_bucket" "usage_export" {
  project     = "development-project"
  bucket_name = "usage-tracking-bucket"
}
```

» Argument Reference

- `bucket_name`: (Required) The bucket to store reports in.

- `prefix`: (Optional) A prefix for the reports, for instance, the project name.
- `project`: (Optional) The project to set the export bucket on. If it is not provided, the provider project is used.

» Import

A project's Usage Export Bucket can be imported using this format:

```
$ terraform import google_project_usage_export_bucket.usage_export {{project}}
```

» google_resource_manager_lien

A Lien represents an encumbrance on the actions that can be performed on a resource.

» Example Usage - Resource Manager Lien

```
resource "google_resource_manager_lien" "lien" {
  parent       = "projects/${google_project.project.number}"
  restrictions = ["resourcemanager.projects.delete"]
  origin       = "machine-readable-explanation"
  reason       = "This project is an important environment"
```

```
}

resource "google_project" "project" {
  project_id = "staging-project"
  name       = "A very important project!"
}
```

» Argument Reference

The following arguments are supported:

- `reason` - (Required) Concise user-visible strings indicating why an action cannot be performed on a resource. Maximum length of 200 characters.
- `origin` - (Required) A stable, user-visible/meaningful string identifying the origin of the Lien, intended to be inspected programmatically. Maximum length of 200 characters.
- `parent` - (Required) A reference to the resource this Lien is attached to. The server will validate the parent against those for which Liens are supported. Since a variety of objects can have Liens against them, you must provide the type prefix (e.g. "projects/my-project-name").
- `restrictions` - (Required) The types of operations which should be blocked as a result of this Lien. Each value should correspond to an IAM permission. The server will validate the permissions against those for which Liens are supported. An empty list is meaningless and will be rejected. e.g. ['resourcemanager.projects.delete']

» Attributes Reference

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

- `name` - A system-generated unique identifier for this Lien.
- `create_time` - Time of creation

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

Lien can be imported using any of these accepted formats:

```
$ terraform import google_resource_manager_lien.default {{parent}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» google_service_account

Allows management of a Google Cloud Platform service account

Creation of service accounts is eventually consistent, and that can lead to errors when you try to apply ACLs to service accounts immediately after creation. If using these resources in the same config, you can add a `sleep` using `local-exec`.

» Example Usage

This snippet creates a service account in a project.

```
resource "google_service_account" "service_account" {
  account_id   = "service_account_id"
  display_name = "Service Account"
}
```

» Argument Reference

The following arguments are supported:

- `account_id` - (Required) The account id that is used to generate the service account email address and a stable unique id. It is unique within a project, must be 6-30 characters long, and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])` to comply with RFC1035. 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.
- `description` - (Optional) A text description of the service account.
- `project` - (Optional) The ID of the project that the service account will be created in. Defaults to the provider project configuration.

» 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 **to configure permissions for who can edit the service account**. To configure permissions for a service account to act as an identity that can manage other GCP resources, use the google_project_iam set of resources.

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/iam.serviceAccountUser"

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

resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that only Jane can interact with"
}

resource "google_service_account_iam_policy" "admin-account-iam" {
  service_account_id = google_service_account.sa.name
  policy_data        = data.google_iam_policy.admin.policy_data
}
```

» google_service_account_iam_binding

```
resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that only Jane can use"
}

resource "google_service_account_iam_binding" "admin-account-iam" {
  service_account_id = google_service_account.sa.name
  role               = "roles/iam.serviceAccountUser"

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

With IAM Conditions (beta):

```
resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that only Jane can use"
}

resource "google_service_account_iam_binding" "admin-account-iam" {
```

```
  service_account_id = "${google_service_account.sa.name}"
  role               = "roles/iam.serviceAccountUser"

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_service_account_iam_member

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

resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that Jane can use"
}

resource "google_service_account_iam_member" "admin-account-iam" {
  service_account_id = google_service_account.sa.name
  role               = "roles/iam.serviceAccountUser"
  member             = "user:jane@example.com"
}

# Allow SA service account use the default GCE account
resource "google_service_account_iam_member" "gce-default-account-iam" {
  service_account_id = data.google_compute_default_service_account.default.name
  role               = "roles/iam.serviceAccountUser"
  member             = "serviceAccount:${google_service_account.sa.email}"
}
```

With IAM Conditions (beta):

```
resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that Jane can use"
}

resource "google_service_account_iam_member" "admin-account-iam" {
  service_account_id = "${google_service_account.sa.name}"
```

```
  role               = "roles/iam.serviceAccountUser"
  member             = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `service_account_id` - (Required) The fully-qualified name of the service account 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 G Suite domain (primary, instead of alias) 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.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.

- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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, member identity, and condition (beta).

```
$ 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
```

With conditions: “‘ $ terraform import -provider=google-beta google_service_account_iam_binding.admin-account-iam ”projects/{your-project-id}/serviceAccounts/{your-service-account-email} iam.serviceAccountUser expires_after_2019_12_31”

$ terraform import -provider=google-beta google_service_account_iam_member.admin-account-iam ”projects/{your-project-id}/serviceAccounts/{your-service-account-email} iam.serviceAccountUser user:foo@example.com expires_after_2019_12_31”
“‘

» 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 **to configure permissions for who can edit the service account**.

To configure permissions for a service account to act as an identity that can manage other GCP resources, use the google_project_iam set of resources.

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/iam.serviceAccountUser"

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

resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that only Jane can interact with"
}

resource "google_service_account_iam_policy" "admin-account-iam" {
  service_account_id = google_service_account.sa.name
  policy_data        = data.google_iam_policy.admin.policy_data
}
```

» google_service_account_iam_binding

```
resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that only Jane can use"
}

resource "google_service_account_iam_binding" "admin-account-iam" {
  service_account_id = google_service_account.sa.name
  role               = "roles/iam.serviceAccountUser"

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

With IAM Conditions (beta):

```
resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that only Jane can use"
}

resource "google_service_account_iam_binding" "admin-account-iam" {
  service_account_id = "${google_service_account.sa.name}"
  role               = "roles/iam.serviceAccountUser"

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_service_account_iam_member

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

resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that Jane can use"
```

```
}

resource "google_service_account_iam_member" "admin-account-iam" {
  service_account_id = google_service_account.sa.name
  role               = "roles/iam.serviceAccountUser"
  member             = "user:jane@example.com"
}

# Allow SA service account use the default GCE account
resource "google_service_account_iam_member" "gce-default-account-iam" {
  service_account_id = data.google_compute_default_service_account.default.name
  role               = "roles/iam.serviceAccountUser"
  member             = "serviceAccount:${google_service_account.sa.email}"
}
```

With IAM Conditions (beta):

```
resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that Jane can use"
}

resource "google_service_account_iam_member" "admin-account-iam" {
  service_account_id = "${google_service_account.sa.name}"
  role               = "roles/iam.serviceAccountUser"
  member             = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `service_account_id` - (Required) The fully-qualified name of the service account 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 G Suite domain (primary, instead of alias) 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.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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, member identity, and condition (beta).

```
$ 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
```

With conditions: “‘ $ terraform import -provider=google-beta google_service_account_iam_binding.admin-account-iam ”projects/{your-project-id}/serviceAccounts/{your-service-account-email} iam.serviceAccountUser expires_after_2019_12_31”

$ terraform import -provider=google-beta google_service_account_iam_member.admin-account-iam ”projects/{your-project-id}/serviceAccounts/{your-service-account-email} iam.serviceAccountUser user:foo@example.com expires_after_2019_12_31”
“‘

» 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 **to configure permissions for who can edit the service account**. To configure permissions for a service account to act as an identity that can manage other GCP resources, use the google_project_iam set of resources.

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/iam.serviceAccountUser"

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

resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that only Jane can interact with"
}

resource "google_service_account_iam_policy" "admin-account-iam" {
  service_account_id = google_service_account.sa.name
  policy_data        = data.google_iam_policy.admin.policy_data
}
```

» google_service_account_iam_binding

```
resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that only Jane can use"
}

resource "google_service_account_iam_binding" "admin-account-iam" {
  service_account_id = google_service_account.sa.name
  role               = "roles/iam.serviceAccountUser"

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

With IAM Conditions (beta):

```
resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that only Jane can use"
```

```
}

resource "google_service_account_iam_binding" "admin-account-iam" {
  service_account_id = "${google_service_account.sa.name}"
  role               = "roles/iam.serviceAccountUser"

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_service_account_iam_member

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

resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that Jane can use"
}

resource "google_service_account_iam_member" "admin-account-iam" {
  service_account_id = google_service_account.sa.name
  role               = "roles/iam.serviceAccountUser"
  member             = "user:jane@example.com"
}

# Allow SA service account use the default GCE account
resource "google_service_account_iam_member" "gce-default-account-iam" {
  service_account_id = data.google_compute_default_service_account.default.name
  role               = "roles/iam.serviceAccountUser"
  member             = "serviceAccount:${google_service_account.sa.email}"
}
```

With IAM Conditions (beta):

```
resource "google_service_account" "sa" {
  account_id   = "my-service-account"
  display_name = "A service account that Jane can use"
}
```

```
resource "google_service_account_iam_member" "admin-account-iam" {
  service_account_id = "${google_service_account.sa.name}"
  role               = "roles/iam.serviceAccountUser"
  member             = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `service_account_id` - (Required) The fully-qualified name of the service account 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 G Suite domain (primary, instead of alias) 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.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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, member identity, and condition (beta).

```
$ 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
```

With conditions: “‘ $ terraform import -provider=google-beta google_service_account_iam_binding.admin-account-iam ”projects/{your-project-id}/serviceAccounts/{your-service-account-email} iam.serviceAccountUser expires_after_2019_12_31”

$ terraform import -provider=google-beta google_service_account_iam_member.admin-account-iam ”projects/{your-project-id}/serviceAccounts/{your-service-account-email} iam.serviceAccountUser user:foo@example.com expires_after_2019_12_31”
“‘

» 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" "myaccount" {
  account_id   = "myaccount"
  display_name = "My Service Account"
}

resource "google_service_account_key" "mykey" {
  service_account_id = google_service_account.myaccount.name
  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.name
}

resource "kubernetes_secret" "google-application-credentials" {
  metadata {
    name = "google-application-credentials"
  }
  data = {
    credentials.json = base64decode(google_service_account_key.mykey.private_key)
  }
}
```

» Argument Reference

The following arguments are supported:

- `service_account_id` - (Required) The Service account id of the Key Pair. This can be a string in the format `{ACCOUNT}` or `projects/{PROJECT_ID}/serviceAccounts/{ACCOUNT}`, where `{ACCOUNT}` is the email address or unique id of the service account. If the `{ACCOUNT}` syntax is used, the project will be inferred from the account.
- `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. TYPE_GOOGLE_CREDENTIALS_FILE is the default output format.

» 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.
- `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_cloud_run_domain_mapping

Resource to hold the state and status of a user's domain mapping.

To get more information about DomainMapping, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Cloud Run Domain Mapping Basic

```
resource "google_cloud_run_service" "default" {
  name     = "tftest-cloudrun"
  location = "us-central1"

  metadata {
    namespace = "my-project-name"
  }

  template {
    spec {
      containers {
        image = "gcr.io/cloudrun/hello"
      }
    }
  }
}

resource "google_cloud_run_domain_mapping" "default" {
  location = "us-central1"
  name     = "verified-domain.com"

  metadata {
    namespace = "my-project-name"
  }

  spec {
    route_name = google_cloud_run_service.default.name
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name should be a verified domain

- `spec` - (Required) The spec for this DomainMapping. Structure is documented below.
- `metadata` - (Required) Metadata associated with this DomainMapping. Structure is documented below.
- `location` - (Required) The location of the cloud run instance. eg uscentral1

The `spec` block supports:

- `force_override` - (Optional) If set, the mapping will override any mapping set before this spec was set. It is recommended that the user leaves this empty to receive an error warning about a potential conflict and only set it once the respective UI has given such a warning.
- `route_name` - (Required) The name of the Cloud Run Service that this DomainMapping applies to. The route must exist.
- `certificate_mode` - (Optional) The mode of the certificate.

The `metadata` block supports:

- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and routes. More info: http://kubernetes.io/docs/user-guide/labels
- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this object that can be used by clients to determine when objects have changed. May be used for optimistic concurrency, change detection, and the watch operation on a resource or set of resources. They may only be valid for a particular resource or set of resources. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#concurrency-control-and-consistency
- `self_link` - SelfLink is a URL representing this object.
- `uid` - UID is a unique id generated by the server on successful creation of a resource and is not allowed to change on PUT operations. More info: http://kubernetes.io/docs/user-guide/identifiers#uids
- `namespace` - (Required) In Cloud Run the namespace must be equal to either the project ID or project number.
- `annotations` - (Optional) Annotations is a key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. More info: http://kubernetes.io/docs/user-guide/annotations

---

- `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:

- `status` - The current status of the DomainMapping. Structure is documented below.

The `status` block contains:

- `conditions` - Array of observed DomainMappingConditions, indicating the current state of the DomainMapping. Structure is documented below.
- `observed_generation` - ObservedGeneration is the 'Generation' of the DomainMapping that was last processed by the controller.
- `resource_records` - (Optional) The resource records required to configure this domain mapping. These records must be added to the domain's DNS configuration in order to serve the application via this domain mapping. Structure is documented below.
- `mapped_route_name` - The name of the route that the mapping currently points to.

The `conditions` block contains:

- `message` - Human readable message indicating details about the current status.
- `status` - Status of the condition, one of True, False, Unknown.
- `reason` - One-word CamelCase reason for the condition's current status.
- `type` - Type of domain mapping condition.

The `resource_records` block supports:

- `type` - (Optional) Resource record type. Example: `AAAA`.
- `rrdata` - Data for this record. Values vary by record type, as defined in RFC 1035 (section 5) and RFC 1034 (section 3.6.1).
- `name` - Relative name of the object affected by this record. Only applicable for `CNAME` records. Example: 'www'.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 6 minutes.
- `delete` - Default is 4 minutes.

» Import

DomainMapping can be imported using any of these accepted formats:

```
$ terraform import google_cloud_run_domain_mapping.default locations/{{location}}/namespaces
$ terraform import google_cloud_run_domain_mapping.default {{location}}/{{project}}/{{name}}
$ terraform import google_cloud_run_domain_mapping.default {{location}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_cloud_run_service

Service acts as a top-level container that manages a set of Routes and Configurations which implement a network service. Service exists to provide a singular abstraction which can be access controlled, reasoned about, and which encapsulates software lifecycle decisions such as rollout policy and team resource ownership. Service acts only as an orchestrator of the underlying Routes and Configurations (much as a kubernetes Deployment orchestrates ReplicaSets).

The Service's controller will track the statuses of its owned Configuration and Route, reflecting their statuses and conditions as its own.

See also: https://github.com/knative/serving/blob/master/docs/spec/overview.md#service

To get more information about Service, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Cloud Run Service Basic

```
resource "google_cloud_run_service" "default" {
  name     = "tftest-cloudrun"
  location = "us-central1"

  template {
    spec {
      containers {
        image = "gcr.io/cloudrun/hello"
      }
    }
  }

  traffic {
    percent         = 100
    latest_revision = true
  }
}
```

» Example Usage - Cloud Run Service Sql

```
resource "google_cloud_run_service" "default" {
  name     = "tftest-cloudrun"
  location = "us-central1"

  template {
    spec {
      containers {
        image = "gcr.io/cloudrun/hello"
      }
    }

    metadata {
      annotations = {
        "autoscaling.knative.dev/maxScale"      = "1000"
        "run.googleapis.com/cloudsql-instances" = "my-project-name:us-central1:${google_sql_
        "run.googleapis.com/client-name"        = "cloud-console"
      }
```

```
    }
  }
}

resource "google_sql_database_instance" "instance" {
  name   = "cloudrun-sql"
  region = "us-east1"
  settings {
    tier = "D0"
  }
}
```

» Example Usage - Cloud Run Service Noauth

```
resource "google_cloud_run_service" "default" {
  name     = "tftest-cloudrun"
  location = "us-central1"

  template {
    spec {
      containers {
        image = "gcr.io/cloudrun/hello"
      }
    }
  }
}

data "google_iam_policy" "noauth" {
  binding {
    role = "roles/run.invoker"
    members = [
      "allUsers",
    ]
  }
}

resource "google_cloud_run_service_iam_policy" "noauth" {
  location    = google_cloud_run_service.default.location
  project     = google_cloud_run_service.default.project
  service     = google_cloud_run_service.default.name
```

```
  policy_data = data.google_iam_policy.noauth.policy_data
}
```

» Example Usage - Cloud Run Service Multiple Environment Variables

```
resource "google_cloud_run_service" "default" {
  name     = "tftest-cloudrun"
  location = "us-central1"

  template {
    spec {
      containers {
        image = "gcr.io/cloudrun/hello"
        env {
          name = "SOURCE"
          value = "remote"
        }
        env {
          name = "TARGET"
          value = "home"
        }
      }
    }
  }

  traffic {
    percent         = 100
    latest_revision = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name must be unique within a namespace, within a Cloud Run region. Is required when creating resources. Name is primarily

intended for creation idempotence and configuration definition. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names

- `location` - (Required) The location of the cloud run instance. eg us-central1

The `traffic` block supports:

- `revision_name` - (Optional) RevisionName of a specific revision to which to send this portion of traffic.
- `percent` - (Required) Percent specifies percent of the traffic to this Revision or Configuration.
- `latest_revision` - (Optional) LatestRevision may be optionally provided to indicate that the latest ready Revision of the Configuration should be used for this traffic target. When provided LatestRevision must be true if RevisionName is empty; it must be false when RevisionName is non-empty.

The `template` block supports:

- `metadata` - (Optional) Optional metadata for this Revision, including labels and annotations. Name will be generated by the Configuration. To set minimum instances for this revision, use the "autoscaling.knative.dev/minScale" annotation key. To set maximum instances for this revision, use the "autoscaling.knative.dev/maxScale" annotation key. To set Cloud SQL connections for the revision, use the "run.googleapis.com/cloudsql-instances" annotation key. Structure is documented below.
- `spec` - (Required) RevisionSpec holds the desired state of the Revision (from the client). Structure is documented below.

The `metadata` block supports:

- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and routes. More info: http://kubernetes.io/docs/user-guide/labels
- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this object that can be used by clients to determine when objects have changed. May be used for optimistic concurrency, change detection, and the watch operation on a resource or set of resources. They may only be valid for a particular resource or set of resources. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#concurrency-control-and-consistency

- `self_link` - SelfLink is a URL representing this object.
- `uid` - UID is a unique id generated by the server on successful creation of a resource and is not allowed to change on PUT operations. More info: http://kubernetes.io/docs/user-guide/identifiers#uids
- `namespace` - (Optional) In Cloud Run the namespace must be equal to either the project ID or project number. It will default to the resource's project.
- `annotations` - (Optional) Annotations is a key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. More info: http://kubernetes.io/docs/user-guide/annotations
- `name` - (Optional) Name must be unique within a namespace, within a Cloud Run region. Is required when creating resources. Name is primarily intended for creation idempotence and configuration definition. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names

The `spec` block supports:

- `containers` - (Required) Container defines the unit of execution for this Revision. In the context of a Revision, we disallow a number of the fields of this Container, including: name, ports, and volumeMounts. The runtime contract is documented here: https://github.com/knative/serving/blob/master/docs/runtime-contract.md Structure is documented below.
- `container_concurrency` - (Optional) ContainerConcurrency specifies the maximum allowed in-flight (concurrent) requests per container of the Revision. Values are:
  - `0` thread-safe, the system should manage the max concurrency. This is the default value.
  - `1` not-thread-safe. Single concurrency
  - `2-N` thread-safe, max concurrency of N
- `service_account_name` - (Optional) Email address of the IAM service account associated with the revision of the service. The service account represents the identity of the running revision, and determines what permissions the revision has. If not provided, the revision will use the project's default service account.
- `serving_state` - ServingState holds a value describing the state the resources are in for this Revision. It is expected that the system will manipulate this based on routability and load.

The `containers` block supports:

- `working_dir` - (Optional, Deprecated) Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image.

- `args` - (Optional) Arguments to the entrypoint. The docker image's CMD is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
- `env_from` - (Optional, Deprecated) List of sources to populate environment variables in the container. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Structure is documented below.
- `image` - (Required) Docker image name. This is most often a reference to a container located in the container registry, such as gcr.io/cloudrun/hello More info: https://kubernetes.io/docs/concepts/containers/images
- `command` - (Optional) Entrypoint array. Not executed within a shell. The docker image's ENTRYPOINT is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
- `env` - (Optional) List of environment variables to set in the container. Structure is documented below.
- `resources` - (Optional) Compute Resources required by this container. Used to set values such as max memory More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources Structure is documented below.

The `env_from` block supports:

- `prefix` - (Optional) An optional identifier to prepend to each key in the ConfigMap.
- `config_map_ref` - (Optional) The ConfigMap to select from. Structure is documented below.
- `secret_ref` - (Optional) The Secret to select from. Structure is documented below.

The `config_map_ref` block supports:

- `optional` - (Optional) Specify whether the ConfigMap must be defined
- `local_object_reference` - (Optional) The ConfigMap to select from. Structure is documented below.

The `local_object_reference` block supports:

- `name` - (Required) Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

The `secret_ref` block supports:

- `local_object_reference` - (Optional) The Secret to select from. Structure is documented below.
- `optional` - (Optional) Specify whether the Secret must be defined

The `local_object_reference` block supports:

- `name` - (Required) Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

The `env` block supports:

- `name` - (Optional) Name of the environment variable.
- `value` - (Optional) Variable references $(VAR_NAME) are expanded using the previous defined environment variables in the container and any route environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".

The `resources` block supports:

- `limits` - (Optional) Limits describes the maximum amount of compute resources allowed. The values of the map is string form of the 'quantity' k8s type: https://github.com/kubernetes/kubernetes/blob/master/staging/src/k8s.io/apimachinery/pkg/api/resource/quantity.go
- `requests` - (Optional) Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. The values of the map is string form of the 'quantity' k8s type: https://github.com/kubernetes/kubernetes/blob/master/staging/src/k8s.io/apimachinery/pkg/api/resource/quantity.go

---

- `traffic` - (Optional) Traffic specifies how to distribute traffic over a collection of Knative Revisions and Configurations Structure is documented below.

- `template` - (Optional) template holds the latest specification for the Revision to be stamped out. The template references the container image, and may also include labels and annotations that should be attached to the Revision. To correlate a Revision, and/or to force a Revision to be created when the spec doesn't otherwise change, a nonce label may be provided in the template metadata. For more details, see: https://github.com/knative/serving/blob/master/docs/client-conventions.md#associate-modifications-with-revisions Cloud Run does not currently support referencing a build that is responsible for materializing the container image from source. Structure is documented below.

- `metadata` - (Optional) Metadata associated with this Service, including name, namespace, labels, and annotations. 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.

The `metadata` block supports:

- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and routes. More info: http://kubernetes.io/docs/user-guide/labels

- `generation` - A sequence number representing a specific generation of the desired state.

- `resource_version` - An opaque value that represents the internal version of this object that can be used by clients to determine when objects have changed. May be used for optimistic concurrency, change detection, and the watch operation on a resource or set of resources. They may only be valid for a particular resource or set of resources. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#concurrency-control-and-consistency

- `self_link` - SelfLink is a URL representing this object.

- `uid` - UID is a unique id generated by the server on successful creation of a resource and is not allowed to change on PUT operations. More info: http://kubernetes.io/docs/user-guide/identifiers#uids

- `namespace` - (Optional) In Cloud Run the namespace must be equal to either the project ID or project number.

- `annotations` - (Optional) Annotations is a key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. More info: http://kubernetes.io/docs/user-guide/annotations

» Attributes Reference

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

- `status` - The current status of the Service. Structure is documented below.

The `status` block contains:

- `conditions` - Array of observed Service Conditions, indicating the current ready state of the service. Structure is documented below.
- `url` - From RouteStatus. URL holds the url that will distribute traffic over the provided traffic targets. It generally has the form https://{route-hash}-{project-hash}-{cluster-level-suffix}.a.run.app
- `observed_generation` - ObservedGeneration is the 'Generation' of the Route that was last processed by the controller. Clients polling for completed reconciliation should poll until observedGeneration = metadata.generation and the Ready condition's status is True or False.
- `latest_created_revision_name` - From ConfigurationStatus. LatestCreatedRevisionName is the last revision that was created from this Service's Configuration. It might not be ready yet, for that use LatestReadyRevisionName.
- `latest_ready_revision_name` - From ConfigurationStatus. LatestReadyRevisionName holds the name of the latest Revision stamped out from this Service's Configuration that has had its "Ready" condition become "True".

The `conditions` block contains:

- `message` - Human readable message indicating details about the current status.
- `status` - Status of the condition, one of True, False, Unknown.
- `reason` - One-word CamelCase reason for the condition's current status.
- `type` - Type of domain mapping condition.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 6 minutes.
- `update` - Default is 6 minutes.
- `delete` - Default is 4 minutes.

» Import

Service can be imported using any of these accepted formats:

```
$ terraform import google_cloud_run_service.default locations/{{location}}/namespaces/{{pro
$ terraform import google_cloud_run_service.default {{location}}/{{project}}/{{name}}
$ terraform import google_cloud_run_service.default {{location}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for CloudRunService

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

- `google_cloud_run_service_iam_policy`: Authoritative. Sets the IAM policy for the service and replaces any existing policy already attached.
- `google_cloud_run_service_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 are preserved.
- `google_cloud_run_service_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the service are preserved.

Note: `google_cloud_run_service_iam_policy` **cannot** be used in conjunction with `google_cloud_run_service_iam_binding` and `google_cloud_run_service_iam_member` or they will fight over what your policy should be.

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

» google_cloud_run_service_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
```

```
    ]
  }
}

resource "google_cloud_run_service_iam_policy" "editor" {
  location = "${google_cloud_run_service.default.location}"
  project = "${google_cloud_run_service.default.project}"
  service = "${google_cloud_run_service.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_cloud_run_service_iam_binding

```
resource "google_cloud_run_service_iam_binding" "editor" {
  location = "${google_cloud_run_service.default.location}"
  project = "${google_cloud_run_service.default.project}"
  service = "${google_cloud_run_service.default.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_cloud_run_service_iam_member

```
resource "google_cloud_run_service_iam_member" "editor" {
  location = "${google_cloud_run_service.default.location}"
  project = "${google_cloud_run_service.default.project}"
  service = "${google_cloud_run_service.default.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `service` - (Required) Used to find the parent resource to bind the IAM policy to
- `location` - (Required) The location of the cloud run instance. eg us-central1 Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the

parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_cloud_run_service_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_cloud_run_service_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/locations/{{location}}/services/{{service}}
- {{project}}/{{location}}/{{service}}
- {{location}}/{{service}}
- {{service}}

Any variables not passed in the import command will be taken from the provider configuration.

CloudRun service IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_cloud_run_service_iam_member.editor "locations/{{location}}/namespaces/{{project}}/se roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_cloud_run_service_iam_binding.editor "projects/{{project}}/locations/{{location}}/services/{{service}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_cloud_run_service_iam_policy.editor projects/{{project}}/locations/{{location}}/services/{{service}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for CloudRunService

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

- `google_cloud_run_service_iam_policy`: Authoritative. Sets the IAM policy for the service and replaces any existing policy already attached.
- `google_cloud_run_service_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 are preserved.
- `google_cloud_run_service_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the service are preserved.

Note: `google_cloud_run_service_iam_policy` **cannot** be used in conjunction with `google_cloud_run_service_iam_binding` and `google_cloud_run_service_iam_member` or they will fight over what your policy should be.

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

» google_cloud_run_service_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_cloud_run_service_iam_policy" "editor" {
  location = "${google_cloud_run_service.default.location}"
  project = "${google_cloud_run_service.default.project}"
  service = "${google_cloud_run_service.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_cloud_run_service_iam_binding

```
resource "google_cloud_run_service_iam_binding" "editor" {
  location = "${google_cloud_run_service.default.location}"
  project = "${google_cloud_run_service.default.project}"
  service = "${google_cloud_run_service.default.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_cloud_run_service_iam_member

```
resource "google_cloud_run_service_iam_member" "editor" {
  location = "${google_cloud_run_service.default.location}"
  project = "${google_cloud_run_service.default.project}"
  service = "${google_cloud_run_service.default.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `service` - (Required) Used to find the parent resource to bind the IAM policy to
- `location` - (Required) The location of the cloud run instance. eg us-central1 Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_cloud_run_service_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_cloud_run_service_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/locations/{{location}}/services/{{service}}
- {{project}}/{{location}}/{{service}}
- {{location}}/{{service}}
- {{service}}

Any variables not passed in the import command will be taken from the provider configuration.

CloudRun service IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_cloud_run_service_iam_member.editor "locations/{{location}}/namespaces/{{project}}/se roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_cloud_run_service_iam_binding.editor "projects/{{project}}/locations/{{location}}/services/{{service}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_cloud_run_service_iam_policy.editor projects/{{project}}/locations/{{location}}/services/{{service}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for CloudRunService

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

- `google_cloud_run_service_iam_policy`: Authoritative. Sets the IAM policy for the service and replaces any existing policy already attached.
- `google_cloud_run_service_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 are preserved.
- `google_cloud_run_service_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the service are preserved.

Note: `google_cloud_run_service_iam_policy` **cannot** be used in conjunction with `google_cloud_run_service_iam_binding` and `google_cloud_run_service_iam_member` or they will fight over what your policy should be.

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

» google_cloud_run_service_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_cloud_run_service_iam_policy" "editor" {
  location = "${google_cloud_run_service.default.location}"
  project = "${google_cloud_run_service.default.project}"
  service = "${google_cloud_run_service.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_cloud_run_service_iam_binding

```
resource "google_cloud_run_service_iam_binding" "editor" {
  location = "${google_cloud_run_service.default.location}"
  project = "${google_cloud_run_service.default.project}"
  service = "${google_cloud_run_service.default.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_cloud_run_service_iam_member

```
resource "google_cloud_run_service_iam_member" "editor" {
  location = "${google_cloud_run_service.default.location}"
  project = "${google_cloud_run_service.default.project}"
```

```
  service = "${google_cloud_run_service.default.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `service` - (Required) Used to find the parent resource to bind the IAM policy to
- `location` - (Required) The location of the cloud run instance. eg us-central1 Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_cloud_run_service_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_cloud_run_service_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/locations/{{location}}/services/{{service}}
- {{project}}/{{location}}/{{service}}
- {{location}}/{{service}}
- {{service}}

Any variables not passed in the import command will be taken from the provider configuration.

CloudRun service IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_cloud_run_service_iam_member.editor "locations/{{location}}/namespaces/{{project}}/se roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_cloud_run_service_iam_binding.editor "projects/{{project}}/locations/{{location}}/services/{{service}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_cloud_run_service_iam_policy.editor projects/{{project}}/locations/{{location}}/services/{{service}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_cloud_scheduler_job

A scheduled job that can publish a pubsub message or a http request every X interval of time, using crontab format string.

To use Cloud Scheduler your project must contain an App Engine app that is located in one of the supported regions. If your project does not have an App Engine app, you must create one.

To get more information about Job, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Scheduler Job Pubsub

```
resource "google_pubsub_topic" "topic" {
  name = "job-topic"
}

resource "google_cloud_scheduler_job" "job" {
  name        = "test-job"
  description = "test job"
  schedule    = "*/2 * * * *"

  pubsub_target {
    # topic.id is the topic's full resource name.
    topic_name = google_pubsub_topic.topic.id
    data       = base64encode("test")
  }
}
```

» Example Usage - Scheduler Job Http

```
resource "google_cloud_scheduler_job" "job" {
  name        = "test-job"
  description = "test http job"
  schedule    = "*/8 * * * *"
  time_zone   = "America/New_York"

  http_target {
    http_method = "POST"
    uri         = "https://example.com/ping"
  }
}
```

» Example Usage - Scheduler Job App Engine

```
resource "google_cloud_scheduler_job" "job" {
  name        = "test-job"
  schedule    = "*/4 * * * *"
  description = "test app engine job"
  time_zone   = "Europe/London"

  app_engine_http_target {
    http_method = "POST"

    app_engine_routing {
      service  = "web"
      version  = "prod"
      instance = "my-instance-001"
    }

    relative_uri = "/ping"
  }
}
```

» Example Usage - Scheduler Job Oauth

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

resource "google_cloud_scheduler_job" "job" {
  name        = "test-job"
  description = "test http job"
  schedule    = "*/8 * * * *"
  time_zone   = "America/New_York"

  http_target {
    http_method = "GET"
    uri         = "https://cloudscheduler.googleapis.com/v1/projects/my-project-name/locatio

    oauth_token {
      service_account_email = data.google_compute_default_service_account.default.email
    }
  }
}
```

» Example Usage - Scheduler Job Oidc

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

resource "google_cloud_scheduler_job" "job" {
  name        = "test-job"
  description = "test http job"
  schedule    = "*/8 * * * *"
  time_zone   = "America/New_York"

  http_target {
    http_method = "GET"
    uri         = "https://example.com/ping"

    oidc_token {
      service_account_email = data.google_compute_default_service_account.default.email
    }
```

```
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the job.
- `region` - (Required) Region where the scheduler job resides

- `description` - (Optional) A human-readable description for the job. This string must not contain more than 500 characters.
- `schedule` - (Optional) Describes the schedule on which the job will be executed.
- `time_zone` - (Optional) Specifies the time zone to be used in interpreting schedule. The value of this field must be a time zone name from the tz database.
- `retry_config` - (Optional) By default, if a job does not complete successfully, meaning that an acknowledgement is not received from the handler, then it will be retried with exponential backoff according to the settings Structure is documented below.
- `pubsub_target` - (Optional) Pub/Sub target If the job providers a Pub/Sub target the cron will publish a message to the provided topic Structure is documented below.
- `app_engine_http_target` - (Optional) App Engine HTTP target. If the job providers a App Engine HTTP target the cron will send a request to the service instance Structure is documented below.
- `http_target` - (Optional) HTTP target. If the job providers a http_target the cron will send a request to the targeted url 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.

The `retry_config` block supports:

- `retry_count` - (Optional) The number of attempts that the system will make to run a job using the exponential backoff procedure described by maxDoublings. Values greater than 5 and negative values are not allowed.
- `max_retry_duration` - (Optional) The time limit for retrying a failed job, measured from time when an execution was first attempted. If specified

with retryCount, the job will be retried until both limits are reached. A duration in seconds with up to nine fractional digits, terminated by 's'.

- `min_backoff_duration` - (Optional) The minimum amount of time to wait before retrying a job after it fails. A duration in seconds with up to nine fractional digits, terminated by 's'.
- `max_backoff_duration` - (Optional) The maximum amount of time to wait before retrying a job after it fails. A duration in seconds with up to nine fractional digits, terminated by 's'.
- `max_doublings` - (Optional) The time between retries will double maxDoublings times. A job's retry interval starts at minBackoffDuration, then doubles maxDoublings times, then increases linearly, and finally retries retries at intervals of maxBackoffDuration up to retryCount times.

The `pubsub_target` block supports:

- `topic_name` - (Required) The full resource name for the Cloud Pub/Sub topic to which messages will be published when a job is delivered. ~>**NOTE**: The topic name must be in the same format as required by PubSub's PublishRequest.name, e.g. `projects/my-project/topics/my-topic`.
- `data` - (Optional) The message payload for PubsubMessage. Pubsub message must contain either non-empty data, or at least one attribute.
- `attributes` - (Optional) Attributes for PubsubMessage. Pubsub message must contain either non-empty data, or at least one attribute.

The `app_engine_http_target` block supports:

- `http_method` - (Optional) Which HTTP method to use for the request.
- `app_engine_routing` - (Optional) App Engine Routing setting for the job. Structure is documented below.
- `relative_uri` - (Required) The relative URI. The relative URL must begin with "/" and must be a valid HTTP relative URL. It can contain a path, query string arguments, and # fragments. If the relative URL is empty, then the root path "/" will be used. No spaces are allowed, and the maximum length allowed is 2083 characters
- `body` - (Optional) HTTP request body. A request body is allowed only if the HTTP method is POST or PUT. It will result in invalid argument error to set a body on a job with an incompatible HttpMethod.
- `headers` - (Optional) HTTP request headers. This map contains the header field names and values. Headers can be set when the job is created.

The `app_engine_routing` block supports:

- `service` - (Optional) App service. By default, the job is sent to the service which is the default service when the job is attempted.

- `version` - (Optional) App version. By default, the job is sent to the version which is the default version when the job is attempted.
- `instance` - (Optional) App instance. By default, the job is sent to an instance which is available when the job is attempted.

The `http_target` block supports:

- `uri` - (Required) The full URI path that the request will be sent to.
- `http_method` - (Optional) Which HTTP method to use for the request.
- `body` - (Optional) HTTP request body. A request body is allowed only if the HTTP method is POST, PUT, or PATCH. It is an error to set body on a job with an incompatible HttpMethod.
- `headers` - (Optional) This map contains the header field names and values. Repeated headers are not supported, but a header value can contain commas.
- `oauth_token` - (Optional) Contains information needed for generating an OAuth token. This type of authorization should be used when sending requests to a GCP endpoint. Structure is documented below.
- `oidc_token` - (Optional) Contains information needed for generating an OpenID Connect token. This type of authorization should be used when sending requests to third party endpoints or Cloud Run. Structure is documented below.

The `oauth_token` block supports:

- `service_account_email` - (Required) Service account email to be used for generating OAuth token. The service account must be within the same project as the job.
- `scope` - (Optional) OAuth scope to be used for generating OAuth access token. If not specified, "https://www.googleapis.com/auth/cloud-platform" will be used.

The `oidc_token` block supports:

- `service_account_email` - (Required) Service account email to be used for generating OAuth token. The service account must be within the same project as the job.
- `audience` - (Optional) Audience to be used when generating OIDC token. If not specified, the URI specified in target will be used.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

Job can be imported using any of these accepted formats:

```
$ terraform import google_cloud_scheduler_job.default projects/{{project}}/locations/{{regio
$ terraform import google_cloud_scheduler_job.default {{project}}/{{region}}/{{name}}
$ terraform import google_cloud_scheduler_job.default {{region}}/{{name}}
$ terraform import google_cloud_scheduler_job.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_security_scanner_scan_config

A ScanConfig resource contains the configurations to launch a scan.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about ScanConfig, see:

- API documentation
- How-to Guides
  - Using Cloud Security Scanner

» Example Usage - Scan Config Basic

```
resource "google_compute_address" "scanner_static_ip" {
  provider = google-beta
  name     = "scan-basic-static-ip"
```

```
}

resource "google_security_scanner_scan_config" "scan-config" {
  provider         = google-beta
  display_name     = "terraform-scan-config"
  starting_urls    = ["http://${google_compute_address.scanner_static_ip.address}"]
  target_platforms = ["COMPUTE"]
}

provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}
```

» Argument Reference

The following arguments are supported:

- `display_name` - (Required) The user provider display name of the ScanConfig.
- `starting_urls` - (Required) The starting URLs from which the scanner finds site pages.

- `max_qps` - (Optional) The maximum QPS during scanning. A valid value ranges from 5 to 20 inclusively. Defaults to 15.
- `authentication` - (Optional) The authentication configuration. If specified, service will use the authentication configuration during scanning. Structure is documented below.
- `user_agent` - (Optional) Type of the user agents used for scanning
- `blacklist_patterns` - (Optional) The blacklist URL patterns as described in https://cloud.google.com/security-scanner/docs/excluded-urls
- `schedule` - (Optional) The schedule of the ScanConfig Structure is documented below.
- `target_platforms` - (Optional) Set of Cloud Platforms targeted by the scan. If empty, APP_ENGINE will be used as a default.
- `export_to_security_command_center` - (Optional) Controls export of scan configurations and results to Cloud Security Command Center.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `authentication` block supports:

- `google_account` - (Optional) Describes authentication configuration that uses a Google account. Structure is documented below.
- `custom_account` - (Optional) Describes authentication configuration that uses a custom account. Structure is documented below.

The `google_account` block supports:

- `username` - (Required) The user name of the Google account.
- `password` - (Required) The password of the Google account. The credential is stored encrypted in GCP.

The `custom_account` block supports:

- `username` - (Required) The user name of the custom account.
- `password` - (Required) The password of the custom account. The credential is stored encrypted in GCP.
- `login_url` - (Required) The login form URL of the website.

The `schedule` block supports:

- `schedule_time` - (Optional) A timestamp indicates when the next run will be scheduled. The value is refreshed by the server after each run. If unspecified, it will default to current server time, which means the scan will be scheduled to start immediately.
- `interval_duration_days` - (Required) The duration of time between executions in days

» Attributes Reference

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

- `name` - A server defined name for this index. Format: `projects/{{project}}/scanConfigs/{{server_ge`

» 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

ScanConfig can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_security_scanner_scan_config.default project
$ terraform import -provider=google-beta google_security_scanner_scan_config.default {{proje
$ terraform import -provider=google-beta google_security_scanner_scan_config.default {{name}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_tpu_node

A Cloud TPU instance.

To get more information about Node, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - TPU Node Basic

```
data "google_tpu_tensorflow_versions" "available" {
}

resource "google_tpu_node" "tpu" {
  name = "test-tpu"
  zone = "us-central1-b"

  accelerator_type   = "v3-8"
  tensorflow_version = data.google_tpu_tensorflow_versions.available.versions[0]
  cidr_block         = "10.2.0.0/29"
}
```

» Example Usage - TPU Node Full

```
data "google_tpu_tensorflow_versions" "available" {
}


resource "google_tpu_node" "tpu" {
  name = "test-tpu"
  zone = "us-central1-b"

  accelerator_type = "v3-8"

  cidr_block         = "10.3.0.0/29"
  tensorflow_version = data.google_tpu_tensorflow_versions.available.versions[0]

  description = "Terraform Google Provider test TPU"
  network = "default"

  labels = {
    foo = "bar"
  }

  scheduling_config {
    preemptible = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The immutable name of the TPU.
- `accelerator_type` - (Required) The type of hardware accelerators associated with this node.
- `tensorflow_version` - (Required) The version of Tensorflow running in the Node.

- `cidr_block` - (Required) The CIDR block that the TPU node will use when selecting an IP address. This CIDR block must be a /29 block; the Compute Engine networks API forbids a smaller block, and using a larger block would be wasteful (a node can only consume one IP address). Errors will occur if the CIDR block has already been used for a currently existing TPU node, the CIDR block conflicts with any subnetworks in the user's provided network, or the provided network is peered with another network that is using that CIDR block.
- `zone` - (Required) The GCP location for the TPU.

- `description` - (Optional) The user-supplied description of the TPU. Maximum of 512 characters.
- `network` - (Optional) The name of a network to peer the TPU node to. It must be a preexisting Compute Engine network inside of the project on which this API has been activated. If none is provided, "default" will be used.
- `scheduling_config` - (Optional) Sets the scheduling options for this TPU instance. Structure is documented below.
- `labels` - (Optional) Resource labels to represent user provided metadata.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `scheduling_config` block supports:

- `preemptible` - (Required) Defines whether the TPU instance is preemptible.

» Attributes Reference

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

- `service_account` - The service account used to run the tensor flow services within the node. To share resources, including Google Cloud Storage data, with the Tensorflow job running in the Node, this account must have permissions to that data.
- `network_endpoints` - The network endpoints where TPU workers can be accessed and sent work. It is recommended that Tensorflow clients of the node first reach out to the first (index 0) entry. Structure is documented below.

The `network_endpoints` block contains:

- `ip_address` - The IP address of this network endpoint.
- `port` - The port of this network endpoint.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 15 minutes.
- `update` - Default is 15 minutes.
- `delete` - Default is 15 minutes.

» Import

Node can be imported using any of these accepted formats:

```
$ terraform import google_tpu_node.default projects/{{project}}/locations/{{zone}}/nodes/{{n
$ terraform import google_tpu_node.default {{project}}/{{zone}}/{{name}}
$ terraform import google_tpu_node.default {{zone}}/{{name}}
$ terraform import google_tpu_node.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_address

Represents an Address resource.

Each virtual machine instance has an ephemeral internal IP address and, optionally, an external IP address. To communicate between instances on the same network, you can use an instance's internal IP address. To communicate with the Internet and instances outside of the same network, you must specify the instance's external IP address.

Internal IP addresses are ephemeral and only belong to an instance for the lifetime of the instance; if the instance is deleted and recreated, the instance is assigned a new internal IP address, either by Compute Engine or by you. External IP addresses can be either ephemeral or static.

To get more information about Address, see:

- API documentation
- How-to Guides
  - Reserving a Static External IP Address
  - Reserving a Static Internal IP Address

» Example Usage - Address Basic

```
resource "google_compute_address" "ip_address" {
  name = "my-address"
}
```

» Example Usage - Address With Subnetwork

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

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

resource "google_compute_address" "internal_with_subnet_and_address" {
  name         = "my-internal-address"
  subnetwork   = google_compute_subnetwork.default.self_link
  address_type = "INTERNAL"
  address      = "10.0.42.42"
  region       = "us-central1"
}
```

» Example Usage - Address With Gce Endpoint

```
resource "google_compute_address" "internal_with_gce_endpoint" {
  name         = "my-internal-address-"
  address_type = "INTERNAL"
  purpose      = "GCE_ENDPOINT"
}
```

» Example Usage - Instance With Ip

```
resource "google_compute_address" "static" {
  name = "ipv4-address"
}

data "google_compute_image" "debian_image" {
  family  = "debian-9"
  project = "debian-cloud"
}

resource "google_compute_instance" "instance_with_ip" {
  name         = "vm-instance"
  machine_type = "f1-micro"
  zone         = "us-central1-a"

  boot_disk {
    initialize_params {
      image = data.google_compute_image.debian_image.self_link
    }
  }

  network_interface {
    network = "default"
    access_config {
```

```
      nat_ip = google_compute_address.static.address
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `address` - (Optional) The static external IP address represented by this resource. Only IPv4 is supported. An address may only be specified for INTERNAL address types. The IP address must be inside the specified subnetwork, if any.
- `address_type` - (Optional) The type of address to reserve, either INTERNAL or EXTERNAL. If unspecified, defaults to EXTERNAL.
- `description` - (Optional) An optional description of this resource.
- `purpose` - (Optional) The purpose of this resource, which can be one of the following values:
  - GCE_ENDPOINT for addresses that are used by VM instances, alias IP ranges, internal load balancers, and similar resources. This should only be set when using an Internal address.
- `network_tier` - (Optional) The networking tier used for configuring this address. This field can take the following values: PREMIUM or STANDARD. If this field is not specified, it is assumed to be PREMIUM.
- `subnetwork` - (Optional) The URL of the subnetwork in which to reserve the address. If an IP address is specified, it must be within the subnetwork's IP range. This field can only be used with INTERNAL type with GCE_ENDPOINT/DNS_RESOLVER purposes.
- `labels` - (Optional, Beta) Labels to apply to this address. A list of key->value pairs.
- `region` - (Optional) The Region in which the created address should reside. If it is not provided, the provider region is used.

- `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.
- `users` - The URLs of the resources that are using this address.
- `label_fingerprint` - The fingerprint used for optimistic locking of this resource. Used internally during updates.
- `self_link` - The URI of the created resource.
- `address` - The IP 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

Address can be imported using any of these accepted formats:

```
$ terraform import google_compute_address.default projects/{{project}}/regions/{{region}}/ad
$ terraform import google_compute_address.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_address.default {{region}}/{{name}}
$ terraform import google_compute_address.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_attached_disk

Persistent disks can be attached to a compute instance using the `attached_disk` section within the compute instance configuration. However there may be situations where managing the attached disks via the compute instance config isn't preferable or possible, such as attaching dynamic numbers of disks using the `count` variable.

To get more information about attaching disks, see:

- API documentation
- Resource: google_compute_disk
- How-to Guides
  - Adding a persistent disk

Note: When using `google_compute_attached_disk` you **must** use `lifecycle.ignore_changes = ["attached_disk"]` on the `google_compute_instance` resource that has the disks attached. Otherwise the two resources will fight for control of the attached disk block.

» Example Usage

```
resource "google_compute_attached_disk" "default" {
  disk     = google_compute_disk.default.self_link
  instance = google_compute_instance.default.self_link
}

resource "google_compute_instance" "default" {
  name         = "attached-disk-instance"
  machine_type = "n1-standard-1"
  zone         = "us-west1-a"

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-9"
    }
  }

  network_interface {
    network = "default"
  }

  lifecycle {
    ignore_changes = [attached_disk]
  }
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) `name` or `self_link` of the compute instance that the disk will be attached to. If the `self_link` is provided then `zone` and `project` are extracted from the self link. If only the name is used then `zone` and `project` must be defined as properties on the resource or provider.
- `disk` - (Required) `name` or `self_link` of the disk that will be attached.

- `project` - (Optional) The project that the referenced compute instance is a part of. If `instance` is referenced by its `self_link` the project defined in the link will take precedence.
- `zone` - (Optional) The zone that the referenced compute instance is located within. If `instance` is referenced by its `self_link` the zone defined in the link will take precedence.
- `device_name` - (Optional) Specifies a unique device name of your choice that is reflected into the /dev/disk/by-id/google-* tree of a Linux operating system running within the instance. This name can be used to reference the device for mounting, resizing, and so on, from within the instance.

  If not specified, the server chooses a default device name to apply to this disk, in the form persistent-disks-x, where x is a number assigned by Google Compute Engine.
- `mode` - (Optional) The mode in which to attach this disk, either READ_WRITE or READ_ONLY. If not specified, the default is to attach the disk in READ_WRITE mode.

  Possible values: "READ_ONLY" "READ_WRITE"

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 5 minutes.
- `delete` - Default is 5 minutes.

» Import

Attached Disk can be imported the following ways:

```
$ terraform import google_compute_attached_disk.default projects/{{project}}/zones/{{zone}}/
$ terraform import google_compute_attached_disk.default {{project}}/{{zone}}/{{instance.name
```

» google_compute_autoscaler

Represents an Autoscaler resource.

Autoscalers allow you to automatically scale virtual machine instances in managed instance groups according to an autoscaling policy that you define.

To get more information about Autoscaler, see:

- API documentation
- How-to Guides
  - Autoscaling Groups of Instances

» Example Usage - Autoscaler Single Instance

```
resource "google_compute_autoscaler" "default" {
  provider = google-beta

  name   = "my-autoscaler"
  zone   = "us-central1-f"
  target = google_compute_instance_group_manager.default.self_link

  autoscaling_policy {
    max_replicas    = 5
    min_replicas    = 1
    cooldown_period = 60

    metric {
      name                       = "pubsub.googleapis.com/subscription/num_undelivered_messa
      filter                     = "resource.type = pubsub_subscription AND resource.label.s
      single_instance_assignment = 65535
    }
  }
}

resource "google_compute_instance_template" "default" {
```

```
  provider = google-beta

  name           = "my-instance-template"
  machine_type   = "n1-standard-1"
  can_ip_forward = false

  tags = ["foo", "bar"]

  disk {
    source_image = data.google_compute_image.debian_9.self_link
  }

  network_interface {
    network = "default"
  }

  metadata = {
    foo = "bar"
  }

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

resource "google_compute_target_pool" "default" {
  provider = google-beta

  name = "my-target-pool"
}

resource "google_compute_instance_group_manager" "default" {
  provider = google-beta

  name = "my-igm"
  zone = "us-central1-f"

  version {
    instance_template = google_compute_instance_template.default.self_link
    name              = "primary"
  }

  target_pools       = [google_compute_target_pool.default.self_link]
  base_instance_name = "autoscaler-sample"
}
```

```
data "google_compute_image" "debian_9" {
  provider = google-beta

  family  = "debian-9"
  project = "debian-cloud"
}

provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}
```

» Example Usage - Autoscaler Basic

```
resource "google_compute_autoscaler" "foobar" {
  name   = "my-autoscaler"
  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
    }
  }
}

resource "google_compute_instance_template" "foobar" {
  name           = "my-instance-template"
  machine_type   = "n1-standard-1"
  can_ip_forward = false

  tags = ["foo", "bar"]

  disk {
    source_image = data.google_compute_image.debian_9.self_link
  }
```

```
  network_interface {
    network = "default"
  }

  metadata = {
    foo = "bar"
  }

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

resource "google_compute_target_pool" "foobar" {
  name = "my-target-pool"
}

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

  version {
    instance_template  = google_compute_instance_template.foobar.self_link
    name               = "primary"
  }

  target_pools       = [google_compute_target_pool.foobar.self_link]
  base_instance_name = "foobar"
}

data "google_compute_image" "debian_9" {
  family  = "debian-9"
  project = "debian-cloud"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. The name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `autoscaling_policy` - (Required) The configuration parameters for the autoscaling algorithm. You can define one or more of the policies for an autoscaler: cpuUtilization, customMetricUtilizations, and loadBalancingUtilization. If none of these are specified, the default will be to autoscale based on cpuUtilization to 0.6 or 60%. Structure is documented below.
- `target` - (Required) URL of the managed instance group that this autoscaler will scale.

The `autoscaling_policy` block supports:

- `min_replicas` - (Required) The minimum number of replicas that the autoscaler can scale down to. This cannot be less than 0. If not provided, autoscaler will choose a default value depending on maximum number of instances allowed.
- `max_replicas` - (Required) The maximum number of instances that the autoscaler can scale up to. This is required when creating or updating an autoscaler. The maximum number of replicas should not be lower than minimal number of replicas.
- `cooldown_period` - (Optional) The number of seconds that the autoscaler should wait before it starts collecting information from a new instance. This prevents the autoscaler from collecting information when the instance is initializing, during which the collected usage would not be reliable. The default time autoscaler waits is 60 seconds. Virtual machine initialization times might vary because of numerous factors. We recommend that you test how long an instance may take to initialize. To do this, create an instance and time the startup process.
- `cpu_utilization` - (Optional) Defines the CPU utilization policy that allows the autoscaler to scale based on the average CPU utilization of a managed instance group. Structure is documented below.
- `metric` - (Optional) Defines the CPU utilization policy that allows the autoscaler to scale based on the average CPU utilization of a managed instance group. Structure is documented below.
- `load_balancing_utilization` - (Optional) Configuration parameters of autoscaling based on a load balancer. Structure is documented below.

The `cpu_utilization` block supports:

- `target` - (Required) The target CPU utilization that the autoscaler should maintain. Must be a float value in the range (0, 1]. If not specified, the default is 0.6. If the CPU level is below the target utilization, the autoscaler scales down the number of instances until it reaches the minimum number of instances you specified or until the average CPU of your instances reaches the target utilization. If the average CPU is above the target utilization, the autoscaler scales up until it reaches the maximum number of

instances you specified or until the average utilization reaches the target utilization.

The `metric` block supports:

- `name` - (Required) The identifier (type) of the Stackdriver Monitoring metric. The metric cannot have negative values. The metric must have a value type of INT64 or DOUBLE.
- `single_instance_assignment` - (Optional, Beta) If scaling is based on a per-group metric value that represents the total amount of work to be done or resource usage, set this value to an amount assigned for a single instance of the scaled group. The autoscaler will keep the number of instances proportional to the value of this metric, the metric itself should not change value due to group resizing. For example, a good metric to use with the target is `pubsub.googleapis.com/subscription/num_undelivered_messages` or a custom metric exporting the total number of requests coming to your instances. A bad example would be a metric exporting an average or median latency, since this value can't include a chunk assignable to a single instance, it could be better used with utilization_target instead.
- `target` - (Optional) The target value of the metric that autoscaler should maintain. This must be a positive value. A utilization metric scales number of virtual machines handling requests to increase or decrease proportionally to the metric. For example, a good metric to use as a utilization-Target is www.googleapis.com/compute/instance/network/received_bytes_count. The autoscaler will work to keep this value constant for each of the instances.
- `type` - (Optional) Defines how target utilization value is expressed for a Stackdriver Monitoring metric. Either GAUGE, DELTA_PER_SECOND, or DELTA_PER_MINUTE.
- `filter` - (Optional, Beta) A filter string to be used as the filter string for a Stackdriver Monitoring TimeSeries.list API call. This filter is used to select a specific TimeSeries for the purpose of autoscaling and to determine whether the metric is exporting per-instance or per-group data. You can only use the AND operator for joining selectors. You can only use direct equality comparison operator (=) without any functions for each selector. You can specify the metric in both the filter string and in the metric field. However, if specified in both places, the metric must be identical. The monitored resource type determines what kind of values are expected for the metric. If it is a gce_instance, the autoscaler expects the metric to include a separate TimeSeries for each instance in a group. In such a case, you cannot filter on resource labels. If the resource type is any other value, the autoscaler expects this metric to contain values that apply to the entire autoscaled instance group and resource label filtering can be performed to point autoscaler at the correct TimeSeries to scale upon. This is called

a per-group metric for the purpose of autoscaling. If not specified, the type defaults to gce_instance. You should provide a filter that is selective enough to pick just one TimeSeries for the autoscaled group or for each of the instances (if you are using gce_instance resource type). If multiple TimeSeries are returned upon the query execution, the autoscaler will sum their respective values to obtain its scaling value.

The `load_balancing_utilization` block supports:

- `target` - (Required) Fraction of backend capacity utilization (set in HTTP(s) load balancing configuration) that autoscaler should maintain. Must be a positive float value. If not defined, the default is 0.8.

- `description` - (Optional) An optional description of this resource.
- `zone` - (Optional) URL of the zone where the instance group resides.
- `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

Autoscaler can be imported using any of these accepted formats:

```
$ terraform import google_compute_autoscaler.default projects/{{project}}/zones/{{zone}}/aut
$ terraform import google_compute_autoscaler.default {{project}}/{{zone}}/{{name}}
$ terraform import google_compute_autoscaler.default {{zone}}/{{name}}
$ terraform import google_compute_autoscaler.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_backend_bucket

Backend buckets allow you to use Google Cloud Storage buckets with HTTP(S) load balancing.

An HTTP(S) load balancer can direct traffic to specified URLs to a backend bucket rather than a backend service. It can send requests for static content to a Cloud Storage bucket and requests for dynamic content to a virtual machine instance.

To get more information about BackendBucket, see:

- API documentation
- How-to Guides
  - Using a Cloud Storage bucket as a load balancer backend

» Example Usage - Backend Bucket Basic

```
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:

- `bucket_name` - (Required) Cloud Storage bucket name.
- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `cdn_policy` - (Optional) Cloud CDN configuration for this Backend Bucket. Structure is documented below.
- `description` - (Optional) An optional textual description of the resource; provided by the client when the resource is created.
- `enable_cdn` - (Optional) If true, enable Cloud CDN for this BackendBucket.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `cdn_policy` block supports:

- `signed_url_cache_max_age_sec` - (Required) Maximum number of seconds the response to a signed URL request will be considered fresh. After this time period, the response will be revalidated before being served. When serving responses to signed URL requests, Cloud CDN will internally behave as though all responses from this backend had a "Cache-Control: public, max-age=[TTL]" header, regardless of any existing Cache-Control header. The actual headers served in responses will not be altered.

» 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

BackendBucket can be imported using any of these accepted formats:

```
$ terraform import google_compute_backend_bucket.default projects/{{project}}/global/backend
$ terraform import google_compute_backend_bucket.default {{project}}/{{name}}
$ terraform import google_compute_backend_bucket.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_backend_bucket_signed_url_key

A key for signing Cloud CDN signed URLs for BackendBuckets.

To get more information about BackendBucketSignedUrlKey, see:

- API documentation
- How-to Guides
  - Using Signed URLs

Warning: All arguments including the key's value will be stored in the raw state as plain-text. Read more about sensitive data in state. Because the API does not return the sensitive key value, we cannot confirm or reverse changes to a key outside of Terraform.

» Example Usage - Backend Bucket Signed Url Key

```
resource "google_compute_backend_bucket_signed_url_key" "backend_key" {
  name           = "test-key"
  key_value      = "pPsVemX8GM46QVeezid6Rw=="
```

```
  backend_bucket = google_compute_backend_bucket.test_backend.name
}

resource "google_compute_backend_bucket" "test_backend" {
  name        = "test-signed-backend-bucket"
  description = "Contains beautiful images"
  bucket_name = google_storage_bucket.bucket.name
  enable_cdn  = true
}

resource "google_storage_bucket" "bucket" {
  name     = "test-storage-bucket"
  location = "EU"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the signed URL key.
- `key_value` - (Required) 128-bit key value used for signing the URL. The key value must be a valid RFC 4648 Section 5 base64url encoded string.
- `backend_bucket` - (Required) The backend bucket this signed URL key belongs.

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

» Timeouts

This resource provides the following Timeouts configuration options:

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

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_backend_service

A Backend Service defines a group of virtual machines that will serve traffic for load balancing. This resource is a global backend service, appropriate for external load balancing or self-managed internal load balancing. For managed internal load balancing, use a regional backend service instead.

Currently self-managed internal load balancing is only available in beta.

To get more information about BackendService, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Backend Service Basic

```
resource "google_compute_backend_service" "default" {
  name          = "backend-service"
  health_checks = [google_compute_http_health_check.default.self_link]
}

resource "google_compute_http_health_check" "default" {
  name               = "health-check"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

» Example Usage - Backend Service Traffic Director Round Robin

```
resource "google_compute_backend_service" "default" {
  provider = google-beta
```

```
  name                  = "backend-service"
  health_checks         = [google_compute_health_check.health_check.self_link]
  load_balancing_scheme = "INTERNAL_SELF_MANAGED"
  locality_lb_policy    = "ROUND_ROBIN"
}

resource "google_compute_health_check" "health_check" {
  provider = google-beta

  name = "health-check"
  http_health_check {
    port = 80
  }
}
```

» Example Usage - Backend Service Traffic Director Ring Hash

```
resource "google_compute_backend_service" "default" {
  provider = google-beta

  name                  = "backend-service"
  health_checks         = [google_compute_health_check.health_check.self_link]
  load_balancing_scheme = "INTERNAL_SELF_MANAGED"
  locality_lb_policy    = "RING_HASH"
  session_affinity      = "HTTP_COOKIE"
  circuit_breakers {
    max_connections = 10
  }
  consistent_hash {
    http_cookie {
      ttl {
        seconds = 11
        nanos   = 1111
      }
      name = "mycookie"
    }
  }
  outlier_detection {
```

```
    consecutive_errors = 2
  }
}

resource "google_compute_health_check" "health_check" {
  provider = google-beta

  name = "health-check"
  http_health_check {
    port = 80
  }
}
```

» Argument Reference

The following arguments are supported:

- `health_checks` - (Required) The set of URLs to the HttpHealthCheck or HttpsHealthCheck resource for health checking this BackendService. Currently at most one health check can be specified, and a health check is required. For internal load balancing, a URL to a HealthCheck resource must be specified instead.
- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `affinity_cookie_ttl_sec` - (Optional) Lifetime of cookies in seconds if session_affinity is GENERATED_COOKIE. If set to 0, the cookie is non-persistent and lasts only until the end of the browser session (or equivalent). The maximum allowed value for TTL is one day. When the load balancing scheme is INTERNAL, this field is not used.
- `backend` - (Optional) The set of backends that serve this BackendService. Structure is documented below.
- `circuit_breakers` - (Optional, Beta) Settings controlling the volume of connections to a backend service. This field is applicable only when the load_balancing_scheme is set to INTERNAL_SELF_MANAGED. Structure is documented below.

- `consistent_hash` - (Optional, Beta) Consistent Hash-based load balancing can be used to provide soft session affinity based on HTTP headers, cookies or other properties. This load balancing policy is applicable only for HTTP connections. The affinity to a particular destination host will be lost when one or more hosts are added/removed from the destination service. This field specifies parameters that control consistent hashing. This field only applies if the load_balancing_scheme is set to INTERNAL_SELF_MANAGED. This field is only applicable when locality_lb_policy is set to MAGLEV or RING_HASH. Structure is documented below.
- `cdn_policy` - (Optional) Cloud CDN configuration for this BackendService. Structure is documented below.
- `connection_draining_timeout_sec` - (Optional) Time for which instance will be drained (not accept new connections, but still work to finish started).
- `custom_request_headers` - (Optional, Beta) Headers that the HTTP/S load balancer should add to proxied requests.
- `description` - (Optional) An optional description of this resource.
- `enable_cdn` - (Optional) If true, enable Cloud CDN for this BackendService.
- `iap` - (Optional) Settings for enabling Cloud Identity Aware Proxy Structure is documented below.
- `load_balancing_scheme` - (Optional) Indicates whether the backend service will be used with internal or external load balancing. A backend service created for one type of load balancing cannot be used with the other. Must be `EXTERNAL` or `INTERNAL_SELF_MANAGED` for a global backend service. Defaults to `EXTERNAL`.
- `locality_lb_policy` - (Optional, Beta) The load balancing algorithm used within the scope of the locality. The possible values are - ROUND_ROBIN - This is a simple policy in which each healthy backend is selected in round robin order. LEAST_REQUEST - An O(1) algorithm which selects two random healthy hosts and picks the host which has fewer active requests. RING_HASH - The ring/modulo hash load balancer implements consistent hashing to backends. The algorithm has the property that the addition/removal of a host from a set of N hosts only affects 1/N of the requests. RANDOM - The load balancer selects a random healthy host. ORIGINAL_DESTINATION - Backend host is selected based on the client connection metadata, i.e., connections are opened to the same address as the destination address of the incoming connection before the connection was redirected to the load balancer. MAGLEV - used as a drop in replacement for the ring hash load balancer. Maglev is not as stable as ring hash but has faster

table lookup build times and host selection times. For more information about Maglev, refer to https://ai.google/research/pubs/pub44824 This field is applicable only when the load_balancing_scheme is set to INTERNAL_SELF_MANAGED.

- `outlier_detection` - (Optional, Beta) Settings controlling eviction of unhealthy hosts from the load balancing pool. This field is applicable only when the load_balancing_scheme is set to INTERNAL_SELF_MANAGED. Structure is documented below.
- `port_name` - (Optional) Name of backend port. The same name should appear in the instance groups referenced by this service. Required when the load balancing scheme is EXTERNAL.
- `protocol` - (Optional) The protocol this BackendService uses to communicate with backends. Possible values are HTTP, HTTPS, HTTP2, TCP, and SSL. The default is HTTP. **NOTE**: HTTP2 is only valid for beta HTTP/2 load balancer types and may result in errors if used with the GA API.
- `security_policy` - (Optional) The security policy associated with this backend service.
- `session_affinity` - (Optional) Type of session affinity to use. The default is NONE. Session affinity is not applicable if the protocol is UDP.
- `timeout_sec` - (Optional) How many seconds to wait for the backend before considering it a failed request. Default is 30 seconds. Valid range is [1, 86400].
- `log_config` - (Optional, Beta) This field denotes the logging options for the load balancer traffic served by this backend service. If logging is enabled, logs will be exported to Stackdriver. 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.

The `backend` block supports:

- `balancing_mode` - (Optional) Specifies the balancing mode for this backend. For global HTTP(S) or TCP/SSL load balancing, the default is UTILIZATION. Valid values are UTILIZATION, RATE (for HTTP(S)) and CONNECTION (for TCP/SSL).
- `capacity_scaler` - (Optional) A multiplier applied to the group's maximum servicing capacity (based on UTILIZATION, RATE or CONNECTION). Default value is 1, which means the group will serve up to 100% of its configured capacity (depending on balancingMode). A setting of 0 means the group is completely drained, offering 0% of its available Capacity. Valid range is [0.0,1.0].

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `group` - (Required) The fully-qualified URL of an Instance Group or Network Endpoint Group resource. In case of instance group this defines the list of instances that serve traffic. Member virtual machine instances from each instance group must live in the same zone as the instance group itself. No two backends in a backend service are allowed to use same Instance Group resource. For Network Endpoint Groups this defines list of endpoints. All endpoints of Network Endpoint Group must be hosted on instances located in the same zone as the Network Endpoint Group. Backend services cannot mix Instance Group and Network Endpoint Group backends. Note that you must specify an Instance Group or Network Endpoint Group resource using the fully-qualified URL, rather than a partial URL.
- `max_connections` - (Optional) The max number of simultaneous connections for the group. Can be used with either CONNECTION or UTILIZATION balancing modes. For CONNECTION mode, either maxConnections or one of maxConnectionsPerInstance or maxConnectionsPerEndpoint, as appropriate for group type, must be set.
- `max_connections_per_instance` - (Optional) The max number of simultaneous connections that a single backend instance can handle. This is used to calculate the capacity of the group. Can be used in either CONNECTION or UTILIZATION balancing modes. For CONNECTION mode, either maxConnections or maxConnectionsPerInstance must be set.
- `max_connections_per_endpoint` - (Optional) The max number of simultaneous connections that a single backend network endpoint can handle. This is used to calculate the capacity of the group. Can be used in either CONNECTION or UTILIZATION balancing modes. For CONNECTION mode, either maxConnections or maxConnectionsPerEndpoint must be set.
- `max_rate` - (Optional) The max requests per second (RPS) of the group. Can be used with either RATE or UTILIZATION balancing modes, but required if RATE mode. For RATE mode, either maxRate or one of maxRatePerInstance or maxRatePerEndpoint, as appropriate for group type, must be set.
- `max_rate_per_instance` - (Optional) The max requests per second (RPS) that a single backend instance can handle. This is used to calculate the capacity of the group. Can be used in either balancing mode. For RATE mode, either maxRate or maxRatePerInstance must be set.
- `max_rate_per_endpoint` - (Optional) The max requests per second (RPS) that a single backend network endpoint can handle. This is used to calculate the capacity of the group. Can be used in either balancing mode.

For RATE mode, either maxRate or maxRatePerEndpoint must be set.

- `max_utilization` - (Optional) Used when balancingMode is UTILIZATION. This ratio defines the CPU utilization target for the group. The default is 0.8. Valid range is [0.0, 1.0].

The `circuit_breakers` block supports:

- `connect_timeout` - (Optional) The timeout for new network connections to hosts. Structure is documented below.
- `max_requests_per_connection` - (Optional) Maximum requests for a single backend connection. This parameter is respected by both the HTTP/1.1 and HTTP/2 implementations. If not specified, there is no limit. Setting this parameter to 1 will effectively disable keep alive.
- `max_connections` - (Optional) The maximum number of connections to the backend cluster. Defaults to 1024.
- `max_pending_requests` - (Optional) The maximum number of pending requests to the backend cluster. Defaults to 1024.
- `max_requests` - (Optional) The maximum number of parallel requests to the backend cluster. Defaults to 1024.
- `max_retries` - (Optional) The maximum number of parallel retries to the backend cluster. Defaults to 3.

The `connect_timeout` block supports:

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.
- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 seconds field and a positive nanos field. Must be from 0 to 999,999,999 inclusive.

The `consistent_hash` block supports:

- `http_cookie` - (Optional) Hash is based on HTTP Cookie. This field describes a HTTP cookie that will be used as the hash key for the consistent hash load balancer. If the cookie is not present, it will be generated. This field is applicable if the sessionAffinity is set to HTTP_COOKIE. Structure is documented below.
- `http_header_name` - (Optional) The hash based on the value of the specified header field. This field is applicable if the sessionAffinity is set to HEADER_FIELD.
- `minimum_ring_size` - (Optional) The minimum number of virtual nodes to use for the hash ring. Larger ring sizes result in more granular load distributions. If the number of hosts in the load balancing pool is larger

than the ring size, each host will be assigned a single virtual node. Defaults to 1024.

The `http_cookie` block supports:

- `ttl` - (Optional) Lifetime of the cookie. Structure is documented below.
- `name` - (Optional) Name of the cookie.
- `path` - (Optional) Path to set for the cookie.

The `ttl` block supports:

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.
- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 seconds field and a positive nanos field. Must be from 0 to 999,999,999 inclusive.

The `cdn_policy` block supports:

- `cache_key_policy` - (Optional) The CacheKeyPolicy for this CdnPolicy. Structure is documented below.
- `signed_url_cache_max_age_sec` - (Optional) Maximum number of seconds the response to a signed URL request will be considered fresh, defaults to 1hr (3600s). After this time period, the response will be revalidated before being served. When serving responses to signed URL requests, Cloud CDN will internally behave as though all responses from this backend had a "Cache-Control: public, max-age=[TTL]" header, regardless of any existing Cache-Control header. The actual headers served in responses will not be altered.

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) OAuth2 Client ID for IAP
- `oauth2_client_secret` - (Required) OAuth2 Client Secret for IAP
- `oauth2_client_secret_sha256` - OAuth2 Client Secret SHA-256 for IAP

The `outlier_detection` block supports:

- `base_ejection_time` - (Optional) The base time that a host is ejected for. The real time is equal to the base time multiplied by the number of times the host has been ejected. Defaults to 30000ms or 30s. Structure is documented below.
- `consecutive_errors` - (Optional) Number of errors before a host is ejected from the connection pool. When the backend host is accessed over HTTP, a 5xx return code qualifies as an error. Defaults to 5.
- `consecutive_gateway_failure` - (Optional) The number of consecutive gateway failures (502, 503, 504 status or connection errors that are mapped to one of those status codes) before a consecutive gateway failure ejection occurs. Defaults to 5.
- `enforcing_consecutive_errors` - (Optional) The percentage chance that a host will be actually ejected when an outlier status is detected through consecutive 5xx. This setting can be used to disable ejection or to ramp it up slowly. Defaults to 100.
- `enforcing_consecutive_gateway_failure` - (Optional) The percentage chance that a host will be actually ejected when an outlier status is detected through consecutive gateway failures. This setting can be used to disable ejection or to ramp it up slowly. Defaults to 0.
- `enforcing_success_rate` - (Optional) The percentage chance that a host will be actually ejected when an outlier status is detected through success rate statistics. This setting can be used to disable ejection or to ramp it up slowly. Defaults to 100.
- `interval` - (Optional) Time interval between ejection sweep analysis. This can result in both new ejections as well as hosts being returned to service. Defaults to 10 seconds. Structure is documented below.
- `max_ejection_percent` - (Optional) Maximum percentage of hosts in the load balancing pool for the backend service that can be ejected. Defaults to 10%.

- `success_rate_minimum_hosts` - (Optional) The number of hosts in a cluster that must have enough request volume to detect success rate outliers. If the number of hosts is less than this setting, outlier detection via success rate statistics is not performed for any host in the cluster. Defaults to 5.
- `success_rate_request_volume` - (Optional) The minimum number of total requests that must be collected in one interval (as defined by the interval duration above) to include this host in success rate based outlier detection. If the volume is lower than this setting, outlier detection via success rate statistics is not performed for that host. Defaults to 100.
- `success_rate_stdev_factor` - (Optional) This factor is used to determine the ejection threshold for success rate outlier ejection. The ejection threshold is the difference between the mean success rate, and the product of this factor and the standard deviation of the mean success rate: mean - (stdev * success_rate_stdev_factor). This factor is divided by a thousand to get a double. That is, if the desired factor is 1.9, the runtime value should be 1900. Defaults to 1900.

The `base_ejection_time` block supports:

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.
- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.

The `interval` block supports:

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.
- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.

The `log_config` block supports:

- `enable` - (Optional) Whether to enable logging for the load balancer traffic served by this backend service.
- `sample_rate` - (Optional) This field can only be specified if logging is enabled for this backend service. The value of the field must be in [0, 1]. This configures the sampling rate of requests to the load balancer where 1.0 means all logged requests are reported and 0.0 means no logged requests are reported. The default value is 1.0.

» Attributes Reference

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

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `fingerprint` - Fingerprint of this resource. A hash of the contents stored in this object. This field is used in optimistic locking.
- `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

BackendService can be imported using any of these accepted formats:

```
$ terraform import google_compute_backend_service.default projects/{{project}}/global/backer
$ terraform import google_compute_backend_service.default {{project}}/{{name}}
$ terraform import google_compute_backend_service.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_backend_service_signed_url_key

A key for signing Cloud CDN signed URLs for Backend Services.

To get more information about BackendServiceSignedUrlKey, see:

- API documentation
- How-to Guides
  - Using Signed URLs

Warning: All arguments including the key's value will be stored in the raw state as plain-text. Read more about sensitive data in state. Because the API does not return the sensitive key value, we cannot confirm or reverse changes to a key outside of Terraform.

» Example Usage - Backend Service Signed Url Key

```
resource "google_compute_backend_service_signed_url_key" "backend_key" {
  name            = "test-key"
  key_value       = "pPsVemX8GM46QVeezid6Rw=="
  backend_service = google_compute_backend_service.example_backend.name
}

resource "google_compute_backend_service" "example_backend" {
  name        = "my-backend-service"
  description = "Our company website"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10
  enable_cdn  = true

  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"

  version {
    instance_template  = google_compute_instance_template.webserver.self_link
    name               = "primary"
  }

  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-9"
    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) Name of the signed URL key.
- `key_value` - (Required) 128-bit key value used for signing the URL. The key value must be a valid RFC 4648 Section 5 base64url encoded string.
- `backend_service` - (Required) The backend service this signed URL key belongs.

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

» Timeouts

This resource provides the following Timeouts configuration options:

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

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_disk

Persistent disks are durable storage devices that function similarly to the physical disks in a desktop or a server. Compute Engine manages the hardware behind these devices to ensure data redundancy and optimize performance for you. Persistent disks are available as either standard hard disk drives (HDD) or solid-state drives (SSD).

Persistent disks are located independently from your virtual machine instances, so you can detach or move persistent disks to keep your data even after you delete your instances. Persistent disk performance scales automatically with size, so you can resize your existing persistent disks or add more persistent disks to an instance to meet your performance and storage space requirements.

Add a persistent disk to your instance when you need reliable and affordable storage with consistent performance characteristics.

To get more information about Disk, see:

- API documentation
- How-to Guides
  - Adding a persistent disk

Warning: 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 - Disk Basic

```
resource "google_compute_disk" "default" {
  name  = "test-disk"
  type  = "pd-ssd"
  zone  = "us-central1-a"
  image = "debian-8-jessie-v20170523"
  labels = {
    environment = "dev"
  }
  physical_block_size_bytes = 4096
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `labels` - (Optional) Labels to apply to this disk. A list of key->value pairs.
- `size` - (Optional) Size of the persistent disk, specified in GB. You can specify this field when creating a persistent disk using the `image` or `snapshot` parameter, or specify it alone to create an empty persistent disk. If you specify this field along with `image` or `snapshot`, the value must not be less than the size of the image or the size of the snapshot.
- `physical_block_size_bytes` - (Optional) Physical block size of the persistent disk, in bytes. If not present in a request, a default value is used. Currently supported sizes are 4096 and 16384, other sizes may be added in the future. If an unsupported value is requested, the error message will list the supported values for the caller's project.
- `type` - (Optional) URL of the disk type resource describing which disk type to use to create the disk. Provide this when creating the 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.
- `resource_policies` - (Optional, Beta) Resource policies applied to this disk for automatic snapshot creations.
- `zone` - (Optional) A reference to the zone where the disk resides.

- `source_image_encryption_key` - (Optional) The customer-supplied encryption key of the source image. Required if the source image is protected by a customer-supplied encryption key. Structure is documented below.
- `disk_encryption_key` - (Optional) Encrypts the disk using a customer-supplied encryption key. After you encrypt a disk with a customer-supplied key, you must provide the same key if you use the disk later (e.g. to create a disk snapshot or an image, or to attach the disk to a virtual machine). Customer-supplied encryption keys do not protect access to metadata of the disk. If you do not provide an encryption key when creating the disk, then the disk will be encrypted using an automatically generated key and you do not need to provide a key to use the disk later. Structure is documented below.
- `snapshot` - (Optional) The source snapshot used to create this disk. You can provide this as a partial or full URL to the resource. If the snapshot is in another project than this disk, you must supply a full URL. For example, the following are valid values:
  - `https://www.googleapis.com/compute/v1/projects/project/global/snapshots/snapshot`
  - `projects/project/global/snapshots/snapshot`
  - `global/snapshots/snapshot`
  - `snapshot`
- `source_snapshot_encryption_key` - (Optional) The customer-supplied encryption key of the source snapshot. Required if the source snapshot is protected by a customer-supplied encryption key. 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.

The `source_image_encryption_key` block supports:

- `raw_key` - (Optional) Specifies a 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to either encrypt or decrypt this resource.
- `sha256` - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.
- `kms_key_self_link` - (Optional) The self link of the encryption key used to encrypt the disk. Also called KmsKeyName in the cloud console. Your project's Compute Engine System service account (`service-{{PROJECT_NUMBER}}@compute-system.iam.gserviceaccount.com`) must have `roles/cloudkms.cryptoKeyEncrypterDecrypter` to use this feature. See https://cloud.google.com/compute/docs/disks/customer-managed-encryption#encrypt_a_new_persistent_disk_with_your_own_keys

The `disk_encryption_key` block supports:

- `raw_key` - (Optional) Specifies a 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to either encrypt or decrypt this resource.
- `sha256` - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.
- `kms_key_self_link` - (Optional) The self link of the encryption key used to encrypt the disk. Also called KmsKeyName in the cloud console. Your project's Compute Engine System service account (`service-{{PROJECT_NUMBER}}@compute-system.iam.gserviceaccount.com`) must have `roles/cloudkms.cryptoKeyEncrypterDecrypter` to use this feature. See https://cloud.google.com/compute/docs/disks/customer-managed-encryption#encrypt_a_new_persistent_disk_with_your_own_keys

The `source_snapshot_encryption_key` block supports:

- `raw_key` - (Optional) Specifies a 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to either encrypt or decrypt this resource.
- `kms_key_self_link` - (Optional) The self link of the encryption key used to encrypt the disk. Also called KmsKeyName in the cloud console. Your project's Compute Engine System service account (`service-{{PROJECT_NUMBER}}@compute-system.iam.gserviceaccount.com`) must have `roles/cloudkms.cryptoKeyEncrypterDecrypter` to use this feature. See https://cloud.google.com/compute/docs/disks/customer-managed-encryption#encrypt_a_new_persistent_disk_with_your_own_keys
- `sha256` - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.

» Attributes Reference

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

- `label_fingerprint` - The fingerprint used for optimistic locking of this resource. Used internally during updates.
- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `last_attach_timestamp` - Last attach timestamp in RFC3339 text format.
- `last_detach_timestamp` - Last detach timestamp in RFC3339 text format.
- `users` - Links to the users of the disk (attached instances) in form: project/zones/zone/instances/instance

- `source_image_id` - The ID value of the image used to create this disk. This value identifies the exact image that was used to create this persistent disk. For example, if you created the persistent disk from an image that was later deleted and recreated under the same name, the source image ID would identify the exact version of the image that was used.
- `source_snapshot_id` - The unique ID of the snapshot used to create this disk. This value identifies the exact snapshot that was used to create this persistent disk. For example, if you created the persistent disk from a snapshot that was later deleted and recreated under the same name, the source snapshot ID would identify the exact version of the snapshot that was used.
- `self_link` - The URI of the created resource.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 5 minutes.
- `update` - Default is 4 minutes.
- `delete` - Default is 4 minutes.

» Import

Disk can be imported using any of these accepted formats:

```
$ terraform import google_compute_disk.default projects/{{project}}/zones/{{zone}}/disks/{{n
$ terraform import google_compute_disk.default {{project}}/{{zone}}/{{name}}
$ terraform import google_compute_disk.default {{zone}}/{{name}}
$ terraform import google_compute_disk.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_disk_resource_policy_attachment

Disk resource policies define a schedule for taking snapshots and a retention period for these snapshots.

» Example Usage - Disk Resource Policy Attachment Basic

```
resource "google_compute_disk_resource_policy_attachment" "attachment" {
  name = google_compute_resource_policy.policy.name
  disk = google_compute_disk.ssd.name
  zone = "us-central1-a"
}

resource "google_compute_disk" "ssd" {
  name  = "my-disk"
  image = data.google_compute_image.my_image.self_link
  size  = 50
  type  = "pd-ssd"
  zone  = "us-central1-a"
}

resource "google_compute_resource_policy" "policy" {
  name = "my-resource-policy"
  region = "us-central1"
  snapshot_schedule_policy {
    schedule {
      daily_schedule {
        days_in_cycle = 1
        start_time = "04:00"
      }
    }
  }
}

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

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource policy to be attached to the disk for scheduling snapshot creation. Do not specify the self link.
- `disk` - (Required) The name of the disk in which the resource policies are attached to.

- `zone` - (Optional) A reference to the zone where the disk resides.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

DiskResourcePolicyAttachment can be imported using any of these accepted formats:

```
$ terraform import google_compute_disk_resource_policy_attachment.default projects/{{project
$ terraform import google_compute_disk_resource_policy_attachment.default {{project}}/{{zone
$ terraform import google_compute_disk_resource_policy_attachment.default {{zone}}/{{disk}}/
$ terraform import google_compute_disk_resource_policy_attachment.default {{disk}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_external_vpn_gateway

Represents a VPN gateway managed outside of GCP.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about ExternalVpnGateway, see:

- API documentation

» Example Usage - External Vpn Gateway

```
resource "google_compute_ha_vpn_gateway" "ha_gateway" {
  provider = google-beta
  region   = "us-central1"
  name     = "ha-vpn"
  network  = google_compute_network.network.self_link
}

resource "google_compute_external_vpn_gateway" "external_gateway" {
  provider        = google-beta
  name            = "external-gateway"
  redundancy_type = "SINGLE_IP_INTERNALLY_REDUNDANT"
  description     = "An externally managed VPN gateway"
  interface {
    id         = 0
    ip_address = "8.8.8.8"
  }
}

resource "google_compute_network" "network" {
  provider                = google-beta
  name                    = "network"
  routing_mode            = "GLOBAL"
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "network_subnet1" {
  provider      = google-beta
  name          = "ha-vpn-subnet-1"
  ip_cidr_range = "10.0.1.0/24"
  region        = "us-central1"
  network       = google_compute_network.network.self_link
}

resource "google_compute_subnetwork" "network_subnet2" {
```

```
  provider      = google-beta
  name          = "ha-vpn-subnet-2"
  ip_cidr_range = "10.0.2.0/24"
  region        = "us-west1"
  network       = google_compute_network.network.self_link
}

resource "google_compute_router" "router1" {
  provider = google-beta
  name     = "ha-vpn-router1"
  network  = google_compute_network.network.name
  bgp {
    asn = 64514
  }
}

resource "google_compute_vpn_tunnel" "tunnel1" {
  provider                        = google-beta
  name                            = "ha-vpn-tunnel1"
  region                          = "us-central1"
  vpn_gateway                     = google_compute_ha_vpn_gateway.ha_gateway.self_link
  peer_external_gateway           = google_compute_external_vpn_gateway.external_gateway.sel
  peer_external_gateway_interface = 0
  shared_secret                   = "a secret message"
  router                          = google_compute_router.router1.self_link
  vpn_gateway_interface           = 0
}

resource "google_compute_vpn_tunnel" "tunnel2" {
  provider                        = google-beta
  name                            = "ha-vpn-tunnel2"
  region                          = "us-central1"
  vpn_gateway                     = google_compute_ha_vpn_gateway.ha_gateway.self_link
  peer_external_gateway           = google_compute_external_vpn_gateway.external_gateway.sel
  peer_external_gateway_interface = 0
  shared_secret                   = "a secret message"
  router                          = " ${google_compute_router.router1.self_link}"
  vpn_gateway_interface           = 1
}

resource "google_compute_router_interface" "router1_interface1" {
  provider   = google-beta
  name       = "router1-interface1"
  router     = google_compute_router.router1.name
  region     = "us-central1"
  ip_range   = "169.254.0.1/30"
```

```
  vpn_tunnel = google_compute_vpn_tunnel.tunnel1.name
}

resource "google_compute_router_peer" "router1_peer1" {
  provider                  = google-beta
  name                      = "router1-peer1"
  router                    = google_compute_router.router1.name
  region                    = "us-central1"
  peer_ip_address           = "169.254.0.2"
  peer_asn                  = 64515
  advertised_route_priority = 100
  interface                 = google_compute_router_interface.router1_interface1.name
}

resource "google_compute_router_interface" "router1_interface2" {
  provider   = google-beta
  name       = "router1-interface2"
  router     = google_compute_router.router1.name
  region     = "us-central1"
  ip_range   = "169.254.1.1/30"
  vpn_tunnel = google_compute_vpn_tunnel.tunnel2.name
}

resource "google_compute_router_peer" "router1_peer2" {
  provider                  = google-beta
  name                      = "router1-peer2"
  router                    = google_compute_router.router1.name
  region                    = "us-central1"
  peer_ip_address           = "169.254.1.2"
  peer_asn                  = 64515
  advertised_route_priority = 100
  interface                 = google_compute_router_interface.router1_interface2.name
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `description` - (Optional) An optional description of this resource.
- `redundancy_type` - (Optional) Indicates the redundancy type of this external VPN gateway
- `interface` - (Optional) A list of interfaces on this external VPN gateway. 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.

The `interface` block supports:

- `id` - (Optional) The numberic ID for this interface. Allowed values are based on the redundancy type of this external VPN gateway
  - `0 - SINGLE_IP_INTERNALLY_REDUNDANT`
  - `0, 1 - TWO_IPS_REDUNDANCY`
  - `0, 1, 2, 3 - FOUR_IPS_REDUNDANCY`
- `ip_address` - (Optional) IP address of the interface in the external VPN gateway. Only IPv4 is supported. This IP address can be either from your on-premise gateway or another Cloud provider's VPN gateway, it cannot be an IP address from Google Compute Engine.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

ExternalVpnGateway can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_compute_external_vpn_gateway.default project
$ terraform import -provider=google-beta google_compute_external_vpn_gateway.default {{proje
$ terraform import -provider=google-beta google_compute_external_vpn_gateway.default {{name}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_firewall

Each network has its own firewall controlling access to and from the instances.

All traffic to instances, even from other instances, is blocked by the firewall unless firewall rules are created to allow it.

The default network has automatically created firewall rules that are shown in default firewall rules. No manually created network has automatically created firewall rules except for a default "allow" rule for outgoing traffic and a default "deny" for incoming traffic. For all networks except the default network, you must create any firewall rules you need.

To get more information about Firewall, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Firewall Basic

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

  allow {
    protocol = "icmp"
  }

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

  source_tags = ["web"]
}

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

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `network` - (Required) The name or self_link of the network to attach this firewall to.

- `allow` - (Optional) The list of ALLOW rules specified by this firewall. Each rule specifies a protocol and port-range tuple that describes a permitted connection. Structure is documented below.
- `deny` - (Optional) The list of DENY rules specified by this firewall. Each rule specifies a protocol and port-range tuple that describes a denied connection. Structure is documented below.
- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `destination_ranges` - (Optional) If destination ranges are specified, the firewall will apply only to traffic that has destination IP address in these ranges. These ranges must be expressed in CIDR format. Only IPv4 is supported.
- `direction` - (Optional) Direction of traffic to which this firewall applies; default is INGRESS. Note: For INGRESS traffic, it is NOT supported to specify destinationRanges; For EGRESS traffic, it is NOT supported to specify sourceRanges OR sourceTags.
- `disabled` - (Optional) Denotes whether the firewall rule is disabled, i.e not applied to the network it is associated with. When set to true, the firewall rule is not enforced and the network behaves as if it did not exist. If this is unspecified, the firewall rule will be enabled.
- `enable_logging` - (Optional) This field denotes whether to enable logging for a particular firewall rule. If logging is enabled, logs will be exported to Stackdriver.
- `priority` - (Optional) Priority for this rule. This is an integer between 0 and 65535, both inclusive. When not specified, the value assumed is 1000. Relative priorities determine precedence of conflicting rules. Lower value of priority implies higher precedence (eg, a rule with priority 0 has higher

precedence than a rule with priority 1). DENY rules take precedence over ALLOW rules having equal priority.

- `source_ranges` - (Optional) If source ranges are specified, the firewall will apply only to traffic that has source IP address in these ranges. These ranges must be expressed in CIDR format. One or both of sourceRanges and sourceTags may be set. If both properties are set, the firewall will apply to traffic that has source IP address within sourceRanges OR the source IP that belongs to a tag listed in the sourceTags property. The connection does not need to match both properties for the firewall to apply. Only IPv4 is supported.
- `source_service_accounts` - (Optional) If source service accounts are specified, 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. sourceRanges can be set at the same time as sourceServiceAccounts. If both are set, the firewall will apply to traffic that has source IP address within sourceRanges OR the source IP belongs to an instance with service account listed in sourceServiceAccount. The connection does not need to match both properties for the firewall to apply. sourceServiceAccounts cannot be used at the same time as sourceTags or targetTags.
- `source_tags` - (Optional) If source tags are specified, the firewall will apply only to traffic with source IP that belongs to a tag listed in source tags. Source tags cannot be used to control traffic to an instance's external IP address. Because tags are associated with an instance, not an IP address. One or both of sourceRanges and sourceTags may be set. If both properties are set, the firewall will apply to traffic that has source IP address within sourceRanges OR the source IP that belongs to a tag listed in the sourceTags property. The connection does not need to match both properties for the firewall to apply.
- `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 allowed[]. targetServiceAccounts cannot be used at the same time as targetTags or sourceTags. If neither targetServiceAccounts nor targetTags are specified, the firewall rule applies to all instances on the specified network.
- `target_tags` - (Optional) A list of instance tags indicating sets of instances located in the network that may make network connections as specified in allowed[]. If no targetTags are specified, the firewall rule applies to all instances on the specified network.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `allow` block supports:

- `protocol` - (Required) The IP protocol to which this rule applies. The protocol type is required when creating a firewall rule. This value can either be one of the following well known protocol strings (tcp, udp, icmp, esp, ah, sctp), or the IP protocol number.
- `ports` - (Optional) An optional list of ports to which this rule applies. This field is only applicable for UDP or TCP protocol. Each entry must be either an integer or a range. If not specified, this rule applies to connections through any port. Example inputs include: ["22"], ["80","443"], and ["12345-12349"].

The `deny` block supports:

- `protocol` - (Required) The IP protocol to which this rule applies. The protocol type is required when creating a firewall rule. This value can either be one of the following well known protocol strings (tcp, udp, icmp, esp, ah, sctp), or the IP protocol number.
- `ports` - (Optional) An optional list of ports to which this rule applies. This field is only applicable for UDP or TCP protocol. Each entry must be either an integer or a range. If not specified, this rule applies to connections through any port. Example inputs include: ["22"], ["80","443"], and ["12345-12349"].

» 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

Firewall can be imported using any of these accepted formats:

```
$ terraform import google_compute_firewall.default projects/{{project}}/global/firewalls/{{
$ terraform import google_compute_firewall.default {{project}}/{{name}}
$ terraform import google_compute_firewall.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_forwarding_rule

A ForwardingRule resource. A ForwardingRule resource specifies which pool of target virtual machines to forward a packet to if it matches the given [IPAddress, IPProtocol, portRange] tuple.

To get more information about ForwardingRule, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Forwarding Rule Global Internallb

```
// Forwarding rule for Internal Load Balancing
resource "google_compute_forwarding_rule" "default" {
  provider = "google-beta"
  name                  = "website-forwarding-rule"
  region                = "us-central1"
  load_balancing_scheme = "INTERNAL"
  backend_service       = "${google_compute_region_backend_service.backend.self_link}"
  all_ports             = true
  allow_global_access   = true
  network               = "${google_compute_network.default.name}"
  subnetwork            = "${google_compute_subnetwork.default.name}"
}
resource "google_compute_region_backend_service" "backend" {
```

```
  provider = "google-beta"
  name                  = "website-backend"
  region                = "us-central1"
  health_checks         = ["${google_compute_health_check.hc.self_link}"]
}
resource "google_compute_health_check" "hc" {
  provider = "google-beta"
  name               = "check-website-backend"
  check_interval_sec = 1
  timeout_sec        = 1
  tcp_health_check {
    port = "80"
  }
}
resource "google_compute_network" "default" {
  provider = "google-beta"
  name = "website-net"
  auto_create_subnetworks = false
}
resource "google_compute_subnetwork" "default" {
  provider = "google-beta"
  name          = "website-net"
  ip_cidr_range = "10.0.0.0/16"
  region        = "us-central1"
  network       = "${google_compute_network.default.self_link}"
}
```

» Example Usage - Forwarding Rule Basic

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

resource "google_compute_target_pool" "default" {
  name = "website-target-pool"
}
```

» Example Usage - Forwarding Rule Internallb

```
// Forwarding rule for Internal Load Balancing
resource "google_compute_forwarding_rule" "default" {
  name   = "website-forwarding-rule"
  region = "us-central1"

  load_balancing_scheme = "INTERNAL"
  backend_service       = google_compute_region_backend_service.backend.self_link
  all_ports             = true
  network               = google_compute_network.default.name
  subnetwork            = google_compute_subnetwork.default.name
}

resource "google_compute_region_backend_service" "backend" {
  name          = "website-backend"
  region        = "us-central1"
  health_checks = [google_compute_health_check.hc.self_link]
}

resource "google_compute_health_check" "hc" {
  name               = "check-website-backend"
  check_interval_sec = 1
  timeout_sec        = 1

  tcp_health_check {
    port = "80"
  }
}

resource "google_compute_network" "default" {
  name                    = "website-net"
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "default" {
  name          = "website-net"
  ip_cidr_range = "10.0.0.0/16"
  region        = "us-central1"
```

```
  network       = google_compute_network.default.self_link
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource; provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `ip_address` - (Optional) The IP address that this forwarding rule is serving on behalf of. Addresses are restricted based on the forwarding rule's load balancing scheme (EXTERNAL or INTERNAL) and scope (global or regional). When the load balancing scheme is EXTERNAL, for global forwarding rules, the address must be a global IP, and for regional forwarding rules, the address must live in the same region as the forwarding rule. If this field is empty, an ephemeral IPv4 address from the same scope (global or regional) will be assigned. A regional forwarding rule supports IPv4 only. A global forwarding rule supports either IPv4 or IPv6. When the load balancing scheme is INTERNAL, this can only be an RFC 1918 IP address belonging to the network/subnet configured for the forwarding rule. By default, if this field is empty, an ephemeral internal IP address will be automatically allocated from the IP range of the subnet or network configured for this forwarding rule. An address must be specified by a literal IP address. ~> **NOTE**: While the API allows you to specify various resource paths for an address resource instead, Terraform requires this to specifically be an IP address to avoid needing to fetching the IP address from resource paths on refresh or unnecessary diffs.
- `ip_protocol` - (Optional) The IP protocol to which this rule applies. Valid options are TCP, UDP, ESP, AH, SCTP or ICMP. When the load balancing scheme is INTERNAL, only TCP and UDP are valid.
- `backend_service` - (Optional) A BackendService to receive the matched traffic. This is used only for INTERNAL load balancing.
- `load_balancing_scheme` - (Optional) This signifies what the ForwardingRule will be used for and can be EXTERNAL, INTERNAL, or INTER-

NAL_MANAGED. EXTERNAL is used for Classic Cloud VPN gateways, protocol forwarding to VMs from an external IP address, and HTTP(S), SSL Proxy, TCP Proxy, and Network TCP/UDP load balancers. INTERNAL is used for protocol forwarding to VMs from an internal IP address, and internal TCP/UDP load balancers. INTERNAL_MANAGED is used for internal HTTP(S) load balancers.

- `network` - (Optional) For internal load balancing, this field identifies the network that the load balanced IP should belong to for this Forwarding Rule. If this field is not specified, the default network will be used. This field is only used for INTERNAL load balancing.

- `port_range` - (Optional) This field is used along with the target field for TargetHttpProxy, TargetHttpsProxy, TargetSslProxy, TargetTcpProxy, TargetVpnGateway, TargetPool, TargetInstance. Applicable only when IPProtocol is TCP, UDP, or SCTP, only packets addressed to ports in the specified range will be forwarded to target. Forwarding rules with the same [IPAddress, IPProtocol] pair must have disjoint port ranges. Some types of forwarding target have constraints on the acceptable ports:

  - TargetHttpProxy: 80, 8080
  - TargetHttpsProxy: 443
  - TargetTcpProxy: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
  - TargetSslProxy: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
  - TargetVpnGateway: 500, 4500

- `ports` - (Optional) This field is used along with the backend_service field for internal load balancing. When the load balancing scheme is INTERNAL, a single port or a comma separated list of ports can be configured. Only packets addressed to these ports will be forwarded to the backends configured with this forwarding rule. You may specify a maximum of up to 5 ports.

- `subnetwork` - (Optional) The subnetwork that the load balanced IP should belong to for this Forwarding Rule. This field is only used for INTERNAL load balancing. If the network specified is in auto subnet mode, this field is optional. However, if the network is in custom subnet mode, a subnetwork must be specified.

- `target` - (Optional) This field is only used for EXTERNAL load balancing. A reference to a TargetPool resource to receive the matched traffic. This target must live in the same region as the forwarding rule. The forwarded traffic must be of a type appropriate to the target object.

- `allow_global_access` - (Optional, Beta) If true, clients can access ILB from all regions. Otherwise only allows from the local region the ILB is located at.

- `labels` - (Optional, Beta) Labels to apply to this forwarding rule. A list of key->value pairs.
- `all_ports` - (Optional) For internal TCP/UDP load balancing (i.e. load balancing scheme is INTERNAL and protocol is TCP/UDP), set this to true to allow packets addressed to any ports to be forwarded to the backends configured with this forwarding rule. Used with backend service. Cannot be set if port or portRange are set.
- `network_tier` - (Optional) The networking tier used for configuring this address. This field can take the following values: PREMIUM or STANDARD. If this field is not specified, it is assumed to be PREMIUM.
- `service_label` - (Optional) An optional prefix to the service name for this Forwarding Rule. If specified, will be the first label of the fully qualified service name. The label must be 1-63 characters long, and comply with RFC1035. Specifically, the label must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash. This field is only used for INTERNAL load balancing.
- `region` - (Optional) A reference to the region where the regional forwarding rule resides. This field is not applicable to global forwarding rules.
- `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.
- `label_fingerprint` - The fingerprint used for optimistic locking of this resource. Used internally during updates.
- `service_name` - The internal fully qualified service name for this Forwarding Rule. This field is only used for INTERNAL load balancing.
- `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

ForwardingRule can be imported using any of these accepted formats:

```
$ terraform import google_compute_forwarding_rule.default projects/{{project}}/regions/{{reg
$ terraform import google_compute_forwarding_rule.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_forwarding_rule.default {{region}}/{{name}}
$ terraform import google_compute_forwarding_rule.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_global_address

Represents a Global Address resource. Global addresses are used for HTTP(S) load balancing.

To get more information about GlobalAddress, see:

- API documentation
- How-to Guides
  - Reserving a Static External IP Address

» Example Usage - Global Address Basic

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

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `address` - (Optional) The IP address or beginning of the address range represented by this resource. This can be supplied as an input to reserve a specific address or omitted to allow GCP to choose a valid one for you.
- `description` - (Optional) An optional description of this resource.
- `labels` - (Optional, Beta) Labels to apply to this address. A list of key->value pairs.
- `ip_version` - (Optional) The IP Version that will be used by this address. Valid options are `IPV4` or `IPV6`. The default value is `IPV4`.
- `prefix_length` - (Optional) The prefix length of the IP range. If not present, it means the address field is a single IP address. This field is not applicable to addresses with addressType=EXTERNAL.
- `address_type` - (Optional) The type of the address to reserve, default is EXTERNAL.
  - EXTERNAL indicates public/external single IP address.
  - INTERNAL indicates internal IP ranges belonging to some network.
- `purpose` - (Optional) The purpose of the resource. For global internal addresses it can be
  - VPC_PEERING - for peer networks This should only be set when using an Internal address.
- `network` - (Optional) The URL of the network in which to reserve the IP range. The IP range must be in RFC1918 space. The network cannot be deleted if there are any reserved IP ranges referring to it. This should only be set when using an Internal address.
- `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.
- `label_fingerprint` - The fingerprint used for optimistic locking of this resource. Used internally during updates.
- `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

GlobalAddress can be imported using any of these accepted formats:

```
$ terraform import google_compute_global_address.default projects/{{project}}/global/address
$ terraform import google_compute_global_address.default {{project}}/{{name}}
$ terraform import google_compute_global_address.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_global_forwarding_rule

Represents a GlobalForwardingRule resource. Global forwarding rules are used to forward traffic to the correct load balancer for HTTP load balancing. Global forwarding rules can only be used for HTTP load balancing.

For more information, see https://cloud.google.com/compute/docs/load-balancing/http/

» Example Usage - Global Forwarding Rule Http

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

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

resource "google_compute_url_map" "default" {
  name            = "url-map-target-proxy"
  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        = "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               = "check-backend"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

» Example Usage - Global Forwarding Rule Internal

```
resource "google_compute_global_forwarding_rule" "default" {
  provider              = google-beta
  name                  = "global-rule"
  target                = google_compute_target_http_proxy.default.self_link
  port_range            = "80"
  load_balancing_scheme = "INTERNAL_SELF_MANAGED"
  ip_address            = "0.0.0.0"
  metadata_filters {
    filter_match_criteria = "MATCH_ANY"
    filter_labels {
      name  = "PLANET"
      value = "MARS"
    }
  }
}

resource "google_compute_target_http_proxy" "default" {
  provider    = google-beta
  name        = "target-proxy"
  description = "a description"
  url_map     = google_compute_url_map.default.self_link
}

resource "google_compute_url_map" "default" {
  provider        = google-beta
  name            = "url-map-target-proxy"
```

```
  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" {
  provider              = google-beta
  name                  = "backend"
  port_name             = "http"
  protocol              = "HTTP"
  timeout_sec           = 10
  load_balancing_scheme = "INTERNAL_SELF_MANAGED"

  backend {
    group                 = google_compute_instance_group_manager.igm.instance_group
    balancing_mode        = "RATE"
    capacity_scaler       = 0.4
    max_rate_per_instance = 50
  }

  health_checks = [google_compute_health_check.default.self_link]
}

data "google_compute_image" "debian_image" {
  provider = google-beta
  family   = "debian-9"
  project  = "debian-cloud"
}

resource "google_compute_instance_group_manager" "igm" {
  provider = google-beta
  name     = "igm-internal"
```

```
  version {
    instance_template = google_compute_instance_template.instance_template.self_link
    name              = "primary"
  }
  base_instance_name = "internal-glb"
  zone               = "us-central1-f"
  target_size        = 1
}

resource "google_compute_instance_template" "instance_template" {
  provider     = google-beta
  name         = "template-backend"
  machine_type = "n1-standard-1"

  network_interface {
    network = "default"
  }

  disk {
    source_image = data.google_compute_image.debian_image.self_link
    auto_delete  = true
    boot         = true
  }
}

resource "google_compute_health_check" "default" {
  provider           = google-beta
  name               = "check-backend"
  check_interval_sec = 1
  timeout_sec        = 1

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

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource; provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following char-

acters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `target` - (Required) The URL of the target resource to receive the matched traffic. The forwarded traffic must be of a type appropriate to the target object. For INTERNAL_SELF_MANAGED load balancing, only HTTP and HTTPS targets are valid.

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `ip_address` - (Optional) The IP address that this forwarding rule is serving on behalf of. Addresses are restricted based on the forwarding rule's load balancing scheme (EXTERNAL or INTERNAL) and scope (global or regional). When the load balancing scheme is EXTERNAL, for global forwarding rules, the address must be a global IP, and for regional forwarding rules, the address must live in the same region as the forwarding rule. If this field is empty, an ephemeral IPv4 address from the same scope (global or regional) will be assigned. A regional forwarding rule supports IPv4 only. A global forwarding rule supports either IPv4 or IPv6. When the load balancing scheme is INTERNAL, this can only be an RFC 1918 IP address belonging to the network/subnet configured for the forwarding rule. By default, if this field is empty, an ephemeral internal IP address will be automatically allocated from the IP range of the subnet or network configured for this forwarding rule. An address must be specified by a literal IP address. ~> **NOTE**: While the API allows you to specify various resource paths for an address resource instead, Terraform requires this to specifically be an IP address to avoid needing to fetching the IP address from resource paths on refresh or unnecessary diffs.
- `ip_protocol` - (Optional) The IP protocol to which this rule applies. Valid options are TCP, UDP, ESP, AH, SCTP or ICMP. When the load balancing scheme is INTERNAL_SELF_MANAGED, only TCP is valid.
- `ip_version` - (Optional) The IP Version that will be used by this global forwarding rule. Valid options are IPV4 or IPV6.
- `labels` - (Optional, Beta) Labels to apply to this forwarding rule. A list of key->value pairs.
- `load_balancing_scheme` - (Optional) This signifies what the GlobalForwardingRule will be used for. The value of INTERNAL_SELF_MANAGED means that this will be used for Internal Global HTTP(S) LB. The value of EXTERNAL means that this will be used for External Global Load Balancing (HTTP(S) LB, External TCP/UDP LB, SSL Proxy) NOTE: Currently global forwarding rules cannot be used for INTERNAL load balancing.

- `metadata_filters` - (Optional) Opaque filter criteria used by Loadbalancer to restrict routing configuration to a limited set xDS compliant clients. In their xDS requests to Loadbalancer, xDS clients present node metadata. If a match takes place, the relevant routing configuration is made available to those proxies. For each metadataFilter in this list, if its filterMatchCriteria is set to MATCH_ANY, at least one of the filterLabels must match the corresponding label provided in the metadata. If its filterMatchCriteria is set to MATCH_ALL, then all of its filterLabels must match with corresponding labels in the provided metadata. metadataFilters specified here can be overridden by those specified in the UrlMap that this ForwardingRule references. metadataFilters only applies to Loadbalancers that have their loadBalancingScheme set to INTERNAL_SELF_MANAGED. Structure is documented below.

- `network` - (Optional, Beta) This field is not used for external load balancing. For INTERNAL_SELF_MANAGED load balancing, this field identifies the network that the load balanced IP should belong to for this global forwarding rule. If this field is not specified, the default network will be used.

- `port_range` - (Optional) This field is used along with the target field for TargetHttpProxy, TargetHttpsProxy, TargetSslProxy, TargetTcpProxy, TargetVpnGateway, TargetPool, TargetInstance. Applicable only when IPProtocol is TCP, UDP, or SCTP, only packets addressed to ports in the specified range will be forwarded to target. Forwarding rules with the same [IPAddress, IPProtocol] pair must have disjoint port ranges. Some types of forwarding target have constraints on the acceptable ports:

  - TargetHttpProxy: 80, 8080
  - TargetHttpsProxy: 443
  - TargetTcpProxy: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
  - TargetSslProxy: 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
  - TargetVpnGateway: 500, 4500

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

The `metadata_filters` block supports:

- `filter_match_criteria` - (Required) Specifies how individual filterLabel matches within the list of filterLabels contribute towards the overall metadataFilter match. MATCH_ANY - At least one of the filterLabels must have a matching label in the provided metadata. MATCH_ALL - All filterLabels must have matching labels in the provided metadata.

- `filter_labels` - (Required) The list of label value pairs that must match labels in the provided metadata based on filterMatchCriteria This list

must not be empty and can have at the most 64 entries. Structure is documented below.

The `filter_labels` block supports:

- `name` - (Required) Name of the metadata label. The length must be between 1 and 1024 characters, inclusive.
- `value` - (Required) The value that the label must match. The value has a maximum length of 1024 characters.

» Attributes Reference

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

- `label_fingerprint` - The fingerprint used for optimistic locking of this resource. Used internally during updates.
- `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

GlobalForwardingRule can be imported using any of these accepted formats:

```
$ terraform import google_compute_global_forwarding_rule.default projects/{{project}}/global
$ terraform import google_compute_global_forwarding_rule.default {{project}}/{{name}}
$ terraform import google_compute_global_forwarding_rule.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_ha_vpn_gateway

Represents a VPN gateway running in GCP. This virtual device is managed by Google, but used only by you. This type of VPN Gateway allows for the creation of VPN solutions with higher availability than classic Target VPN Gateways.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about HaVpnGateway, see:

- API documentation
- How-to Guides
  - Choosing a VPN
  - Cloud VPN Overview

» Example Usage - Ha Vpn Gateway Basic

```
resource "google_compute_ha_vpn_gateway" "ha_gateway1" {
  provider = google-beta
  region   = "us-central1"
  name     = "ha-vpn-1"
  network  = google_compute_network.network1.self_link
}

resource "google_compute_network" "network1" {
  provider                = google-beta
  name                    = "network1"
  auto_create_subnetworks = false
}
```

» Example Usage - Ha Vpn Gateway Gcp To Gcp

```
resource "google_compute_ha_vpn_gateway" "ha_gateway1" {
```

```
  provider = google-beta
  region   = "us-central1"
  name     = "ha-vpn-1"
  network  = google_compute_network.network1.self_link
}

resource "google_compute_ha_vpn_gateway" "ha_gateway2" {
  provider = google-beta
  region   = "us-central1"
  name     = "ha-vpn-2"
  network  = google_compute_network.network2.self_link
}

resource "google_compute_network" "network1" {
  provider                = google-beta
  name                    = "network1"
  routing_mode            = "GLOBAL"
  auto_create_subnetworks = false
}

resource "google_compute_network" "network2" {
  provider                = google-beta
  name                    = "network2"
  routing_mode            = "GLOBAL"
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "network1_subnet1" {
  provider      = google-beta
  name          = "ha-vpn-subnet-1"
  ip_cidr_range = "10.0.1.0/24"
  region        = "us-central1"
  network       = google_compute_network.network1.self_link
}

resource "google_compute_subnetwork" "network1_subnet2" {
  provider      = google-beta
  name          = "ha-vpn-subnet-2"
  ip_cidr_range = "10.0.2.0/24"
  region        = "us-west1"
  network       = google_compute_network.network1.self_link
}

resource "google_compute_subnetwork" "network2_subnet1" {
  provider      = google-beta
  name          = "ha-vpn-subnet-3"
```

```
  ip_cidr_range = "192.168.1.0/24"
  region        = "us-central1"
  network       = google_compute_network.network2.self_link
}

resource "google_compute_subnetwork" "network2_subnet2" {
  provider      = google-beta
  name          = "ha-vpn-subnet-4"
  ip_cidr_range = "192.168.2.0/24"
  region        = "us-east1"
  network       = google_compute_network.network2.self_link
}

resource "google_compute_router" "router1" {
  provider = google-beta
  name     = "ha-vpn-router1"
  network  = google_compute_network.network1.name
  bgp {
    asn = 64514
  }
}

resource "google_compute_router" "router2" {
  provider = google-beta
  name     = "ha-vpn-router2"
  network  = google_compute_network.network2.name
  bgp {
    asn = 64515
  }
}

resource "google_compute_vpn_tunnel" "tunnel1" {
  provider              = google-beta
  name                  = "ha-vpn-tunnel1"
  region                = "us-central1"
  vpn_gateway           = google_compute_ha_vpn_gateway.ha_gateway1.self_link
  peer_gcp_gateway      = google_compute_ha_vpn_gateway.ha_gateway2.self_link
  shared_secret         = "a secret message"
  router                = google_compute_router.router1.self_link
  vpn_gateway_interface = 0
}

resource "google_compute_vpn_tunnel" "tunnel2" {
  provider              = google-beta
  name                  = "ha-vpn-tunnel2"
  region                = "us-central1"
```

```
  vpn_gateway           = google_compute_ha_vpn_gateway.ha_gateway1.self_link
  peer_gcp_gateway      = google_compute_ha_vpn_gateway.ha_gateway2.self_link
  shared_secret         = "a secret message"
  router                = google_compute_router.router1.self_link
  vpn_gateway_interface = 1
}

resource "google_compute_vpn_tunnel" "tunnel3" {
  provider              = google-beta
  name                  = "ha-vpn-tunnel3"
  region                = "us-central1"
  vpn_gateway           = google_compute_ha_vpn_gateway.ha_gateway2.self_link
  peer_gcp_gateway      = google_compute_ha_vpn_gateway.ha_gateway1.self_link
  shared_secret         = "a secret message"
  router                = google_compute_router.router2.self_link
  vpn_gateway_interface = 0
}

resource "google_compute_vpn_tunnel" "tunnel4" {
  provider              = google-beta
  name                  = "ha-vpn-tunnel4"
  region                = "us-central1"
  vpn_gateway           = google_compute_ha_vpn_gateway.ha_gateway2.self_link
  peer_gcp_gateway      = google_compute_ha_vpn_gateway.ha_gateway1.self_link
  shared_secret         = "a secret message"
  router                = google_compute_router.router2.self_link
  vpn_gateway_interface = 1
}

resource "google_compute_router_interface" "router1_interface1" {
  provider   = google-beta
  name       = "router1-interface1"
  router     = google_compute_router.router1.name
  region     = "us-central1"
  ip_range   = "169.254.0.1/30"
  vpn_tunnel = google_compute_vpn_tunnel.tunnel1.name
}

resource "google_compute_router_peer" "router1_peer1" {
  provider                  = google-beta
  name                      = "router1-peer1"
  router                    = google_compute_router.router1.name
  region                    = "us-central1"
  peer_ip_address           = "169.254.0.2"
  peer_asn                  = 64515
  advertised_route_priority = 100
```

```
  interface                 = google_compute_router_interface.router1_interface1.name
}

resource "google_compute_router_interface" "router1_interface2" {
  provider   = google-beta
  name       = "router1-interface2"
  router     = google_compute_router.router1.name
  region     = "us-central1"
  ip_range   = "169.254.1.1/30"
  vpn_tunnel = google_compute_vpn_tunnel.tunnel2.name
}

resource "google_compute_router_peer" "router1_peer2" {
  provider                  = google-beta
  name                      = "router1-peer2"
  router                    = google_compute_router.router1.name
  region                    = "us-central1"
  peer_ip_address           = "169.254.1.2"
  peer_asn                  = 64515
  advertised_route_priority = 100
  interface                 = google_compute_router_interface.router1_interface2.name
}

resource "google_compute_router_interface" "router2_interface1" {
  provider   = google-beta
  name       = "router2-interface1"
  router     = google_compute_router.router2.name
  region     = "us-central1"
  ip_range   = "169.254.0.1/30"
  vpn_tunnel = google_compute_vpn_tunnel.tunnel3.name
}

resource "google_compute_router_peer" "router2_peer1" {
  provider                  = google-beta
  name                      = "router2-peer1"
  router                    = google_compute_router.router2.name
  region                    = "us-central1"
  peer_ip_address           = "169.254.0.2"
  peer_asn                  = 64514
  advertised_route_priority = 100
  interface                 = google_compute_router_interface.router2_interface1.name
}

resource "google_compute_router_interface" "router2_interface2" {
  provider   = google-beta
  name       = "router2-interface2"
```

```
  router     = google_compute_router.router2.name
  region     = "us-central1"
  ip_range   = "169.254.1.1/30"
  vpn_tunnel = google_compute_vpn_tunnel.tunnel4.name
}

resource "google_compute_router_peer" "router2_peer2" {
  provider                  = google-beta
  name                      = "router2-peer2"
  router                    = google_compute_router.router2.name
  region                    = "us-central1"
  peer_ip_address           = "169.254.1.2"
  peer_asn                  = 64514
  advertised_route_priority = 100
  interface                 = google_compute_router_interface.router2_interface2.name
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `network` - (Required) The network this VPN gateway is accepting traffic for.

- `description` - (Optional) An optional description of this resource.
- `region` - (Optional) The region this gateway should sit in.
- `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:

- `vpn_interfaces` - A list of interfaces on this VPN gateway. Structure is documented below.
- `self_link` - The URI of the created resource.

The `vpn_interfaces` block contains:

- `id` - (Optional) The numeric ID of this VPN gateway interface.
- `ip_address` - (Optional) The external IP address for this VPN gateway interface.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

HaVpnGateway can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_compute_ha_vpn_gateway.default projects/{{pr
$ terraform import -provider=google-beta google_compute_ha_vpn_gateway.default {{project}}/{
$ terraform import -provider=google-beta google_compute_ha_vpn_gateway.default {{region}}/{{
$ terraform import -provider=google-beta google_compute_ha_vpn_gateway.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_health_check

Health Checks determine whether instances are responsive and able to do work. They are an important part of a comprehensive load balancing configuration, as they enable monitoring instances behind load balancers.

Health Checks poll instances at a specified interval. Instances that do not respond successfully to some number of probes in a row are marked as unhealthy. No new connections are sent to unhealthy instances, though existing connections will continue. The health check will continue to poll unhealthy instances. If an

instance later responds successfully to some number of consecutive probes, it is marked healthy again and can receive new connections.

To get more information about HealthCheck, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Health Check Tcp

```
resource "google_compute_health_check" "tcp-health-check" {
  name = "tcp-health-check"

  timeout_sec        = 1
  check_interval_sec = 1

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

» Example Usage - Health Check Tcp Full

```
resource "google_compute_health_check" "tcp-health-check" {
  name        = "tcp-health-check"
  description = "Health check via tcp"

  timeout_sec         = 1
  check_interval_sec  = 1
  healthy_threshold   = 4
  unhealthy_threshold = 5

  tcp_health_check {
    port_name          = "health-check-port"
```

```
    port_specification = "USE_NAMED_PORT"
    request            = "ARE YOU HEALTHY?"
    proxy_header       = "NONE"
    response           = "I AM HEALTHY"
  }
}
```

» Example Usage - Health Check Ssl

```
resource "google_compute_health_check" "ssl-health-check" {
  name = "ssl-health-check"

  timeout_sec        = 1
  check_interval_sec = 1

  ssl_health_check {
    port = "443"
  }
}
```

» Example Usage - Health Check Ssl Full

```
resource "google_compute_health_check" "ssl-health-check" {
  name        = "ssl-health-check"
  description = "Health check via ssl"

  timeout_sec         = 1
  check_interval_sec  = 1
  healthy_threshold   = 4
  unhealthy_threshold = 5

  ssl_health_check {
    port_name          = "health-check-port"
```

```
    port_specification = "USE_NAMED_PORT"
    request            = "ARE YOU HEALTHY?"
    proxy_header       = "NONE"
    response           = "I AM HEALTHY"
  }
}
```

» Example Usage - Health Check Http

```
resource "google_compute_health_check" "http-health-check" {
  name = "http-health-check"

  timeout_sec        = 1
  check_interval_sec = 1

  http_health_check {
    port = 80
  }
}
```

» Example Usage - Health Check Http Full

```
resource "google_compute_health_check" "http-health-check" {
  name        = "http-health-check"
  description = "Health check via http"

  timeout_sec         = 1
  check_interval_sec  = 1
  healthy_threshold   = 4
  unhealthy_threshold = 5

  http_health_check {
    port_name          = "health-check-port"
```

```
    port_specification = "USE_NAMED_PORT"
    host               = "1.2.3.4"
    request_path       = "/mypath"
    proxy_header       = "NONE"
    response           = "I AM HEALTHY"
  }
}
```

» Example Usage - Health Check Https

```
resource "google_compute_health_check" "https-health-check" {
  name = "https-health-check"

  timeout_sec        = 1
  check_interval_sec = 1

  https_health_check {
    port = "443"
  }
}
```

» Example Usage - Health Check Https Full

```
resource "google_compute_health_check" "https-health-check" {
  name        = "https-health-check"
  description = "Health check via https"

  timeout_sec         = 1
  check_interval_sec  = 1
  healthy_threshold   = 4
  unhealthy_threshold = 5

  https_health_check {
```

```
    port_name          = "health-check-port"
    port_specification = "USE_NAMED_PORT"
    host               = "1.2.3.4"
    request_path       = "/mypath"
    proxy_header       = "NONE"
    response           = "I AM HEALTHY"
  }
}
```

» Example Usage - Health Check Http2

```
resource "google_compute_health_check" "http2-health-check" {
  name = "http2-health-check"

  timeout_sec        = 1
  check_interval_sec = 1

  http2_health_check {
    port = "443"
  }
}
```

» Example Usage - Health Check Http2 Full

```
resource "google_compute_health_check" "http2-health-check" {
  name        = "http2-health-check"
  description = "Health check via http2"

  timeout_sec         = 1
  check_interval_sec  = 1
  healthy_threshold   = 4
  unhealthy_threshold = 5
```

```
  http2_health_check {
    port_name          = "health-check-port"
    port_specification = "USE_NAMED_PORT"
    host               = "1.2.3.4"
    request_path       = "/mypath"
    proxy_header       = "NONE"
    response           = "I AM HEALTHY"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `check_interval_sec` - (Optional) How often (in seconds) to send a health check. The default value is 5 seconds.
- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `healthy_threshold` - (Optional) A so-far unhealthy instance will be marked healthy after this many consecutive successes. The default value is 2.
- `timeout_sec` - (Optional) How long (in seconds) to wait before claiming failure. The default value is 5 seconds. It is invalid for timeoutSec to have greater value than checkIntervalSec.
- `unhealthy_threshold` - (Optional) A so-far healthy instance will be marked unhealthy after this many consecutive failures. The default value is 2.
- `http_health_check` - (Optional) A nested object resource Structure is documented below.
- `https_health_check` - (Optional) A nested object resource Structure is documented below.
- `tcp_health_check` - (Optional) A nested object resource Structure is documented below.

- `ssl_health_check` - (Optional) A nested object resource Structure is documented below.
- `http2_health_check` - (Optional) A nested object resource 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.

The `http_health_check` block supports:

- `host` - (Optional) The value of the host header in the HTTP health check request. If left empty (default value), the public IP on behalf of which this health check is performed will be used.
- `request_path` - (Optional) The request path of the HTTP health check request. The default value is /.
- `response` - (Optional) The bytes to match against the beginning of the response data. If left empty (the default value), any response will indicate health. The response data can only be ASCII.
- `port` - (Optional) The TCP port number for the HTTP health check request. The default value is 80.
- `port_name` - (Optional) Port name as defined in InstanceGroup#NamedPort#name. If both port and port_name are defined, port takes precedence.
- `proxy_header` - (Optional) Specifies the type of proxy header to append before sending data to the backend, either NONE or PROXY_V1. The default is NONE.
- `port_specification` - (Optional) Specifies how port is selected for health checking, can be one of the following values:
  - `USE_FIXED_PORT`: The port number in `port` is used for health checking.
  - `USE_NAMED_PORT`: The `portName` is used for health checking.
  - `USE_SERVING_PORT`: For NetworkEndpointGroup, the port specified for each network endpoint is used for health checking. For other backends, the port or named port specified in the Backend Service is used for health checking. If not specified, HTTP health check follows behavior specified in `port` and `portName` fields.

The `https_health_check` block supports:

- `host` - (Optional) The value of the host header in the HTTPS health check request. If left empty (default value), the public IP on behalf of which this health check is performed will be used.
- `request_path` - (Optional) The request path of the HTTPS health check request. The default value is /.

- `response` - (Optional) The bytes to match against the beginning of the response data. If left empty (the default value), any response will indicate health. The response data can only be ASCII.
- `port` - (Optional) The TCP port number for the HTTPS health check request. The default value is 443.
- `port_name` - (Optional) Port name as defined in InstanceGroup#NamedPort#name. If both port and port_name are defined, port takes precedence.
- `proxy_header` - (Optional) Specifies the type of proxy header to append before sending data to the backend, either NONE or PROXY_V1. The default is NONE.
- `port_specification` - (Optional) Specifies how port is selected for health checking, can be one of the following values:
  - `USE_FIXED_PORT`: The port number in `port` is used for health checking.
  - `USE_NAMED_PORT`: The `portName` is used for health checking.
  - `USE_SERVING_PORT`: For NetworkEndpointGroup, the port specified for each network endpoint is used for health checking. For other backends, the port or named port specified in the Backend Service is used for health checking. If not specified, HTTPS health check follows behavior specified in `port` and `portName` fields.

The `tcp_health_check` block supports:

- `request` - (Optional) The application data to send once the TCP connection has been established (default value is empty). If both request and response are empty, the connection establishment alone will indicate health. The request data can only be ASCII.
- `response` - (Optional) The bytes to match against the beginning of the response data. If left empty (the default value), any response will indicate health. The response data can only be ASCII.
- `port` - (Optional) The TCP port number for the TCP health check request. The default value is 443.
- `port_name` - (Optional) Port name as defined in InstanceGroup#NamedPort#name. If both port and port_name are defined, port takes precedence.
- `proxy_header` - (Optional) Specifies the type of proxy header to append before sending data to the backend, either NONE or PROXY_V1. The default is NONE.
- `port_specification` - (Optional) Specifies how port is selected for health checking, can be one of the following values:
  - `USE_FIXED_PORT`: The port number in `port` is used for health checking.

  - `USE_NAMED_PORT`: The `portName` is used for health checking.
  - `USE_SERVING_PORT`: For NetworkEndpointGroup, the port specified for each network endpoint is used for health checking. For other backends, the port or named port specified in the Backend Service is used for health checking. If not specified, TCP health check follows behavior specified in `port` and `portName` fields.

The `ssl_health_check` block supports:

- `request` - (Optional) The application data to send once the SSL connection has been established (default value is empty). If both request and response are empty, the connection establishment alone will indicate health. The request data can only be ASCII.
- `response` - (Optional) The bytes to match against the beginning of the response data. If left empty (the default value), any response will indicate health. The response data can only be ASCII.
- `port` - (Optional) The TCP port number for the SSL health check request. The default value is 443.
- `port_name` - (Optional) Port name as defined in InstanceGroup#NamedPort#name. If both port and port_name are defined, port takes precedence.
- `proxy_header` - (Optional) Specifies the type of proxy header to append before sending data to the backend, either NONE or PROXY_V1. The default is NONE.
- `port_specification` - (Optional) Specifies how port is selected for health checking, can be one of the following values:
  - `USE_FIXED_PORT`: The port number in `port` is used for health checking.
  - `USE_NAMED_PORT`: The `portName` is used for health checking.
  - `USE_SERVING_PORT`: For NetworkEndpointGroup, the port specified for each network endpoint is used for health checking. For other backends, the port or named port specified in the Backend Service is used for health checking. If not specified, SSL health check follows behavior specified in `port` and `portName` fields.

The `http2_health_check` block supports:

- `host` - (Optional) The value of the host header in the HTTP2 health check request. If left empty (default value), the public IP on behalf of which this health check is performed will be used.
- `request_path` - (Optional) The request path of the HTTP2 health check request. The default value is /.
- `response` - (Optional) The bytes to match against the beginning of the response data. If left empty (the default value), any response will indicate health. The response data can only be ASCII.

- `port` - (Optional) The TCP port number for the HTTP2 health check request. The default value is 443.
- `port_name` - (Optional) Port name as defined in InstanceGroup#NamedPort#name. If both port and port_name are defined, port takes precedence.
- `proxy_header` - (Optional) Specifies the type of proxy header to append before sending data to the backend, either NONE or PROXY_V1. The default is NONE.
- `port_specification` - (Optional) Specifies how port is selected for health checking, can be one of the following values:
  - `USE_FIXED_PORT`: The port number in `port` is used for health checking.
  - `USE_NAMED_PORT`: The `portName` is used for health checking.
  - `USE_SERVING_PORT`: For NetworkEndpointGroup, the port specified for each network endpoint is used for health checking. For other backends, the port or named port specified in the Backend Service is used for health checking. If not specified, HTTP2 health check follows behavior specified in `port` and `portName` fields.

» Attributes Reference

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

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `type` - The type of the health check. One of HTTP, HTTPS, TCP, or SSL.
- `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

HealthCheck can be imported using any of these accepted formats:

```
$ terraform import google_compute_health_check.default projects/{{project}}/global/healthChe
$ terraform import google_compute_health_check.default {{project}}/{{name}}
$ terraform import google_compute_health_check.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_http_health_check

An HttpHealthCheck resource. This resource defines a template for how individual VMs should be checked for health, via HTTP.

Note: google_compute_http_health_check is a legacy health check. The newer google_compute_health_check should be preferred for all uses except Network Load Balancers which still require the legacy version.

To get more information about HttpHealthCheck, see:

- API documentation
- How-to Guides
  - Adding Health Checks

» Example Usage - Http Health Check Basic

```
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) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `check_interval_sec` - (Optional) How often (in seconds) to send a health check. The default value is 5 seconds.
- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `healthy_threshold` - (Optional) A so-far unhealthy instance will be marked healthy after this many consecutive successes. The default value is 2.
- `host` - (Optional) The value of the host header in the HTTP health check request. If left empty (default value), the public IP on behalf of which this health check is performed will be used.
- `port` - (Optional) The TCP port number for the HTTP health check request. The default value is 80.
- `request_path` - (Optional) The request path of the HTTP health check request. The default value is /.
- `timeout_sec` - (Optional) How long (in seconds) to wait before claiming failure. The default value is 5 seconds. It is invalid for timeoutSec to have greater value than checkIntervalSec.
- `unhealthy_threshold` - (Optional) A so-far healthy instance will be marked unhealthy after this many consecutive failures. The default value is 2.
- `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

HttpHealthCheck can be imported using any of these accepted formats:

```
$ terraform import google_compute_http_health_check.default projects/{{project}}/global/http
$ terraform import google_compute_http_health_check.default {{project}}/{{name}}
$ terraform import google_compute_http_health_check.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_https_health_check

An HttpsHealthCheck resource. This resource defines a template for how individual VMs should be checked for health, via HTTPS.

Note: google_compute_https_health_check is a legacy health check. The newer google_compute_health_check should be preferred for all uses except Network Load Balancers which still require the legacy version.

To get more information about HttpsHealthCheck, see:

- API documentation
- How-to Guides
  - Adding Health Checks

» Example Usage - Https Health Check Basic

```
resource "google_compute_https_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) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `check_interval_sec` - (Optional) How often (in seconds) to send a health check. The default value is 5 seconds.
- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `healthy_threshold` - (Optional) A so-far unhealthy instance will be marked healthy after this many consecutive successes. The default value is 2.
- `host` - (Optional) The value of the host header in the HTTPS health check request. If left empty (default value), the public IP on behalf of which this health check is performed will be used.
- `port` - (Optional) The TCP port number for the HTTPS health check request. The default value is 80.

- `request_path` - (Optional) The request path of the HTTPS health check request. The default value is /.
- `timeout_sec` - (Optional) How long (in seconds) to wait before claiming failure. The default value is 5 seconds. It is invalid for timeoutSec to have greater value than checkIntervalSec.
- `unhealthy_threshold` - (Optional) A so-far healthy instance will be marked unhealthy after this many consecutive failures. The default value is 2.
- `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

HttpsHealthCheck can be imported using any of these accepted formats:

```
$ terraform import google_compute_https_health_check.default projects/{{project}}/global/htt
$ terraform import google_compute_https_health_check.default {{project}}/{{name}}
$ terraform import google_compute_https_health_check.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_image

Represents an Image resource.

Google Compute Engine uses operating system images to create the root persistent disks for your instances. You specify an image when you create an instance. Images contain a boot loader, an operating system, and a root file system. Linux operating system images are also capable of running containers on Compute Engine.

Images can be either public or custom.

Public images are provided and maintained by Google, open-source communities, and third-party vendors. By default, all projects have access to these images and can use them to create instances. Custom images are available only to your project. You can create a custom image from root persistent disks and other images. Then, use the custom image to create an instance.

To get more information about Image, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Image Basic

```
resource "google_compute_image" "example" {
  name = "example-image"

  raw_disk {
    source = "https://storage.googleapis.com/bosh-cpi-artifacts/bosh-stemcell-3262.4-google-
  }
}
```

» Example Usage - Image Guest Os

```
resource "google_compute_image" "example" {
  name = "example-image"

  raw_disk {
    source = "https://storage.googleapis.com/bosh-cpi-artifacts/bosh-stemcell-3262.4-google-
  }

  guest_os_features {
    type = "SECURE_BOOT"
  }

  guest_os_features {
    type = "MULTI_IP_SUBNET"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource; provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `disk_size_gb` - (Optional) Size of the image when restored onto a persistent disk (in GB).
- `family` - (Optional) The name of the image family to which this image belongs. You can create disks by specifying an image family instead of a specific image name. The image family always returns its latest image that is not deprecated. The name of the image family must comply with RFC1035.
- `guest_os_features` - (Optional) A list of features to enable on the guest operating system. Applicable only for bootable images. Structure is documented below.

- `labels` - (Optional) Labels to apply to this Image.
- `licenses` - (Optional) Any applicable license URI.
- `raw_disk` - (Optional) The parameters of the raw disk image. Structure is documented below.
- `source_disk` - (Optional) The source disk to create this image based on. You must provide either this property or the rawDisk.source property but not both to create an image.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `guest_os_features` block supports:

- `type` - (Required) The type of supported feature. Read Enabling guest operating system features to see a list of available options.

The `raw_disk` block supports:

- `container_type` - (Optional) The format used to encode and transmit the block device, which should be TAR. This is just a container and transmission format and not a runtime format. Provided by the client when the disk image is created.
- `sha1` - (Optional) An optional SHA1 checksum of the disk image before unpackaging. This is provided by the client when the disk image is created.
- `source` - (Required) The full Google Cloud Storage URL where disk storage is stored You must provide either this property or the sourceDisk property but not both.

» Attributes Reference

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

- `archive_size_bytes` - Size of the image tar.gz archive stored in Google Cloud Storage (in bytes).
- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `label_fingerprint` - The fingerprint used for optimistic locking of this resource. Used internally during updates.
- `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

Image can be imported using any of these accepted formats:

```
$ terraform import google_compute_image.default projects/{{project}}/global/images/{{name}}
$ terraform import google_compute_image.default {{project}}/{{name}}
$ terraform import google_compute_image.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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-9"
    }
  }

  // Local SSD disk
  scratch_disk {
```

```
    interface = "SCSI"
  }

  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.

  Note: If you want to update this value (resize the VM) after initial creation, you must set `allow_stopping_for_update` to `true`.

  Custom machine types can be formatted as `custom-NUMBER_OF_CPUS-AMOUNT_OF_MEMORY_MB`, e.g. `custom-6-20480` for 6 vCPU and 20GB of RAM.

  There is a limit of 6.5 GB per CPU unless you add extended memory. You must do this explicitly by adding the suffix `-ext`, e.g. `custom-2-15360-ext` for 2 vCPU and 15 GB of memory.
- `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) Additional disks to attach to the instance. Can be repeated multiple times for multiple disks. 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.
- `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 before removing the resource (e.g., via `terraform destroy`), or the instance cannot be deleted and the Terraform run will not complete successfully.
- `hostname` - (Optional) A custom hostname for the instance. Must be a fully qualified DNS name and RFC-1035-valid. Valid format is a series of labels 1-63 characters long matching the regular expression `[a-z]([-a-z0-9]*[a-z0-9])`, concatenated with periods. The entire hostname must not exceed 253 characters. Changing this forces a new resource to be created.
- `guest_accelerator` - (Optional) List of the type and count of accelerator cards attached to the instance. Structure documented below. **Note:** GPU accelerators can only be used with `on_host_maintenance` option set to TERMINATE.
- `labels` - (Optional) A map of key/value label pairs to assign to the instance.
- `metadata` - (Optional) Metadata key/value pairs to make available from within the instance. Ssh keys attached in the Cloud Console will be removed. Add them to your config in order to keep them attached to your instance.

On import, `metadata_startup_script` will be set while `metadata.startup-script` will not be. You'll need to match `metadata_startup_script` to your `startup-script` value.

- `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.
- `shielded_instance_config` - (Optional) Enable Shielded VM on this instance. Shielded VM provides verifiable integrity to prevent against malware and rootkits. Defaults to disabled. Structure is documented below. **Note**: `shielded_instance_config` can only be used with boot images with shielded vm support. See the complete list here.
- `enable_display` - (Optional) Enable Virtual Displays on this instance. **Note**: `allow_stopping_for_update` must be set to true in order to update this field.

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. On the instance, this device will be `/dev/disk/by-id/google-{{device_name}}`.
- `mode` - (Optional) The mode in which to attach this disk, either `READ_WRITE` or `READ_ONLY`. If not specified, the default is to attach the disk in `READ_WRITE` mode.
- `disk_encryption_key_raw` - (Optional) A 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to encrypt this disk. Only one of `kms_key_self_link` and `disk_encryption_key_raw` may be set.
- `kms_key_self_link` - (Optional) The self_link of the encryption key that is stored in Google Cloud KMS to encrypt this disk. Only one of `kms_key_self_link` and `disk_encryption_key_raw` may be set.

- `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`) or disk image. To create an instance from a snapshot, first create a `google_compute_disk` from a snapshot and reference it here.

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

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/google-*`
- `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. Only one of `kms_key_self_link` and `disk_encryption_key_raw` may be set.
- `kms_key_self_link` - (Optional) The self_link of the encryption key that is stored in Google Cloud KMS to encrypt this disk. Only one of `kms_key_self_link` and `disk_encryption_key_raw` may be set.

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.
- `network_ip` - (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. If omitted, ssh provisioners will not work unless 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.
- `public_ptr_domain_name` - (Optional) The DNS domain name for the public PTR record. To set this field on an instance, you must be verified as the owner of the domain. See the docs for how to become verified as a domain owner.
- `network_tier` - (Optional) The networking tier used for configuring this instance. This field can take the following values: PREMIUM or STANDARD. If this field is not specified, it is assumed to be PREMIUM.

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) Specifies if the instance is preemptible. If this field is set to true, then `automatic_restart` must be set to false. Defaults to false.
- `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). Defaults to true.
- `node_affinities` - (Optional) Specifies node affinities or anti-affinities to determine which sole-tenant nodes your instances and managed instance groups will use as host systems. Read more on sole-tenant node creation here. Structure 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 `node_affinities` block supports:

- `key` (Required) - The key for the node affinity label.
- `operator` (Required) - The operator. Can be `IN` for node-affinities or `NOT` for anti-affinities.
- `value` (Required) - The values for the node affinity label.

The `shielded_instance_config` block supports:

- `enable_secure_boot` (Optional) -- Verify the digital signature of all boot components, and halt the boot process if signature verification fails. Defaults to false.
- `enable_vtpm` (Optional) -- Use a virtualized trusted platform module, which is a specialized computer chip you can use to encrypt objects like keys and certificates. Defaults to true.
- `enable_integrity_monitoring` (Optional) -- Compare the most recent boot measurements to the integrity policy baseline and return a pair of pass/fail results depending on whether they match or not. Defaults to true.

» 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.network_ip` - The internal ip address of the instance, either manually or dynamically assigned.
- `network_interface.0.access_config.0.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.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 20 minutes.
- `update` - Default is 20 minutes.
- `delete` - Default is 20 minutes.

» Import

Note: The fields `boot_disk.0.disk_encryption_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
```

» IAM policy for ComputeInstance

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

- `google_compute_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.
- `google_compute_instance_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 instance are preserved.
- `google_compute_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_compute_instance_iam_policy` **cannot** be used in conjunction with `google_compute_instance_iam_binding` and `google_compute_instance_iam_member` or they will fight over what your policy should be.

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

» google_compute_instance_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
```

```
    role = "roles/compute.osLogin"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_compute_instance_iam_policy" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/compute.osLogin"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_compute_instance_iam_policy" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_compute_instance_iam_binding

```
resource "google_compute_instance_iam_binding" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  members = [
```

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

With IAM Conditions (beta):

```
resource "google_compute_instance_iam_binding" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_compute_instance_iam_member

```
resource "google_compute_instance_iam_member" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_compute_instance_iam_member" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `instance_name` - (Required) Used to find the parent resource to bind the IAM policy to
- `zone` - (Optional) A reference to the zone where the machine resides. Used to find the parent resource to bind the IAM policy to. If not specified, the value will be parsed from the identifier of the parent resource. If no zone is provided in the parent identifier and no zone is specified, it is taken from the provider configuration.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_compute_instance_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_compute_instance_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/zones/{{zone}}/instances/{{name}}
- {{project}}/{{zone}}/{{name}}
- {{zone}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Compute instance IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_compute_instance_iam_member.editor "projects/{{project}}/zones/{{zone}}/instances/{{i roles/compute.osLogin jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_compute_instance_iam_binding.editor "projects/{{project}}/zones/{{zone}}/instances/{{instance}} roles/compute.osLogin"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_compute_instance_iam_policy.editor projects/{{project}}/zones/{{zone}}/instances/{{instance}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for ComputeInstance

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

- `google_compute_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.
- `google_compute_instance_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 instance are preserved.
- `google_compute_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_compute_instance_iam_policy` **cannot** be used in conjunction with `google_compute_instance_iam_binding` and `google_compute_instance_iam_member` or they will fight over what your policy should be.

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

» google_compute_instance_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/compute.osLogin"
    members = [
      "user:jane@example.com",
    ]
  }
}
```

```
resource "google_compute_instance_iam_policy" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/compute.osLogin"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_compute_instance_iam_policy" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_compute_instance_iam_binding

```
resource "google_compute_instance_iam_binding" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_compute_instance_iam_binding" "editor" {
```

```
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_compute_instance_iam_member

```
resource "google_compute_instance_iam_member" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_compute_instance_iam_member" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `instance_name` - (Required) Used to find the parent resource to bind the IAM policy to
- `zone` - (Optional) A reference to the zone where the machine resides. Used to find the parent resource to bind the IAM policy to. If not specified, the value will be parsed from the identifier of the parent resource. If no zone is provided in the parent identifier and no zone is specified, it is taken from the provider configuration.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_compute_instance_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_compute_instance_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.

- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/zones/{{zone}}/instances/{{name}}
- {{project}}/{{zone}}/{{name}}
- {{zone}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Compute instance IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_compute_instance_iam_member.editor "projects/{{project}}/zones/{{zone}}/instances/{{i roles/compute.osLogin jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_compute_instance_iam_binding.editor "projects/{{project}}/zones/{{zone}}/instances/{{instance}} roles/compute.osLogin"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_compute_instance_iam_policy.editor projects/{{project}}/zones/{{zone}}/instances/{{instance}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for ComputeInstance

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

- `google_compute_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.
- `google_compute_instance_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 instance are preserved.
- `google_compute_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_compute_instance_iam_policy` **cannot** be used in conjunction with `google_compute_instance_iam_binding` and `google_compute_instance_iam_member` or they will fight over what your policy should be.

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

» google_compute_instance_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/compute.osLogin"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_compute_instance_iam_policy" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
```

```
  instance_name = "${google_compute_instance.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/compute.osLogin"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_compute_instance_iam_policy" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_compute_instance_iam_binding

```
resource "google_compute_instance_iam_binding" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_compute_instance_iam_binding" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
```

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_compute_instance_iam_member

```
resource "google_compute_instance_iam_member" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_compute_instance_iam_member" "editor" {
  project = "${google_compute_instance.default.project}"
  zone = "${google_compute_instance.default.zone}"
  instance_name = "${google_compute_instance.default.name}"
  role = "roles/compute.osLogin"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `instance_name` - (Required) Used to find the parent resource to bind the IAM policy to
- `zone` - (Optional) A reference to the zone where the machine resides. Used to find the parent resource to bind the IAM policy to. If not specified, the

value will be parsed from the identifier of the parent resource. If no zone is provided in the parent identifier and no zone is specified, it is taken from the provider configuration.

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_compute_instance_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_compute_instance_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/zones/{{zone}}/instances/{{name}}
- {{project}}/{{zone}}/{{name}}
- {{zone}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Compute instance IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_compute_instance_iam_member.editor "projects/{{project}}/zones/{{zone}}/instances/{{i roles/compute.osLogin jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_compute_instance_iam_binding.editor "projects/{{project}}/zones/{{zone}}/instances/{{instance}} roles/compute.osLogin"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_compute_instance_iam_policy.editor projects/{{project}}/zones/{{zone}}/instances/{{instance}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_instance_from_template

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

This resource is specifically to create a compute instance from a given `source_instance_template`. To create an instance without a template, use the `google_compute_instance` resource.

» Example Usage

```
resource "google_compute_instance_template" "tpl" {
  name         = "template"
  machine_type = "n1-standard-1"

  disk {
    source_image = "debian-cloud/debian-9"
    auto_delete  = true
    disk_size_gb = 100
    boot         = true
  }

  network_interface {
    network = "default"
  }

  metadata = {
    foo = "bar"
  }

  can_ip_forward = true
}

resource "google_compute_instance_from_template" "tpl" {
  name = "instance-from-template"
  zone = "us-central1-a"

  source_instance_template = google_compute_instance_template.tpl.self_link

  // Override fields from instance template
```

```
  can_ip_forward = false
  labels = {
    my_key = "my_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.
- `source_instance_template` - (Required) Name or self link of an instance template to create the instance based on.

- `zone` - (Optional) The zone that the machine should be created in. If not set, the provider zone is used.

In addition to these, all arguments from `google_compute_instance` are supported as a way to override the properties in the template. All exported attributes from `google_compute_instance` are likewise exported here.

To support removal of Optional/Computed fields in Terraform 0.12 the following fields are marked Attributes as Blocks: * `attached_disk` * `guest_accelerator` * `service_account` * `scratch_disk` * `network_interface.alias_ip_range` * `network_interface.access_config`

» Timeouts

This resource provides the following Timeouts configuration options:

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

» google_compute_instance_group

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

Recreating an instance group that's in use by another resource will give a `resourceInUseByAnotherResource` error. You can avoid this error with a Terraform `lifecycle` block as outlined in the example below.

» 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
}
```

» Example Usage - 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"
}
```

» Example Usage - Recreating an instance group in use

Recreating an instance group that's in use by another resource will give a `resourceInUseByAnotherResource` error. Use `lifecycle.create_before_destroy` as shown in this example to avoid this type of error.

```
resource "google_compute_instance_group" "staging_group" {
  name      = "staging-instance-group"
  zone      = "us-central1-c"
  instances = [google_compute_instance.staging_vm.self_link]
  named_port {
    name = "http"
```

```
    port = "8080"
  }

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

  lifecycle {
    create_before_destroy = true
  }
}

data "google_compute_image" "debian_image" {
  family  = "debian-9"
  project = "debian-cloud"
}

resource "google_compute_instance" "staging_vm" {
  name         = "staging-vm"
  machine_type = "n1-standard-1"
  zone         = "us-central1-c"
  boot_disk {
    initialize_params {
      image = data.google_compute_image.debian_image.self_link
    }
  }

  network_interface {
    network = "default"
  }
}

resource "google_compute_backend_service" "staging_service" {
  name      = "staging-service"
  port_name = "https"
  protocol  = "HTTPS"

  backend {
    group = google_compute_instance_group.staging_group.self_link
  }

  health_checks = [
    google_compute_https_health_check.staging_health.self_link,
  ]
}
```

```
resource "google_compute_https_health_check" "staging_health" {
  name         = "staging-health"
  request_path = "/health_check"
}
```

» 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` with an optional `project`, e.g.

```
$ terraform import google_compute_instance_group.webservers us-central1-a/terraform-webserve
$ terraform import google_compute_instance_group.webservers big-project/us-central1-a/terraf
$ terraform import google_compute_instance_group.webservers projects/big-project/zones/us-ce
```

» 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 with top level instance template (`google` provider)

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

  version {
    instance_template  = google_compute_instance_template.appserver.self_link
  }

  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
  }
}
```

» Example Usage with multiple versions (google-beta provider)

```
resource "google_compute_instance_group_manager" "appserver" {
  provider = google-beta
  name     = "appserver-igm"

  base_instance_name = "app"
  zone               = "us-central1-a"

  target_size = 5

  version {
    name              = "appserver"
    instance_template = google_compute_instance_template.appserver.self_link
  }

  version {
    name              = "appserver-canary"
    instance_template = google_compute_instance_template.appserver-canary.self_link
    target_size {
      fixed = 1
```

```
    }
  }
}
```

» 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.
- `version` - (Required) Application versions managed by this instance group. Each version deals with a specific instance template, allowing canary release scenarios. Structure is documented below.
- `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.
- `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.
- `wait_for_instances` - (Optional) Whether to wait for all instances to be created/updated before returning. Note that if this is set to true and the operation does not succeed, Terraform will continue trying until it times out.

- `auto_healing_policies` - (Optional) 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.
- `update_policy` - (Optional) The update policy for this managed instance group. Structure is documented below. For more information, see the official documentation and API

The `update_policy` block supports:

```
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. You can specify either `RESTART` to restart existing instances or `REPLACE` to delete and create new instances from the target template. If you specify a `RESTART`, the Updater will attempt to perform that action only. However, if the Updater determines that the minimal action you specify is not enough to perform the update, it might perform a more disruptive action.
- `type` - (Required) - The type of update process. You can specify either `PROACTIVE` so that the instance group manager proactively executes actions in order to bring instances to their target versions or `OPPORTUNISTIC` so that no action is proactively executed but the update will be performed as part of other actions (for example, resizes or recreateInstances calls).
- `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.

The `version` block supports:

```
version {
  name              = "appserver-canary"
  instance_template = google_compute_instance_template.appserver-canary.self_link

  target_size {
    fixed = 1
  }
}

version {
  name              = "appserver-canary"
  instance_template = google_compute_instance_template.appserver-canary.self_link

  target_size {
    percent = 20
  }
}
```

- `name` - (Required) - Version name.
- `instance_template` - (Required) - The full URL to an instance template from which all new instances of this version will be created.
- `target_size` - (Optional) - The number of instances calculated as a fixed number or a percentage depending on the settings. Structure is documented below.

Exactly one `version` you specify must not have a `target_size` specified. During a rolling update, the instance group manager will fulfill the `target_size`

constraints of every other `version`, and any remaining instances will be provisioned with the version where `target_size` is unset.

The `target_size` block supports:

- `fixed` - (Optional), The number of instances which are managed for this version. Conflicts with `percent`.
- `percent` - (Optional), The number of instances (calculated as percentage) which are managed for this version. Conflicts with `fixed`. Note that when using `percent`, rounding will be in favor of explicitly set `target_size` values; a managed instance group with 2 instances and 2 `version`s, one of which has a `target_size.percent` of `60` will create 2 instances of that `version`.

» 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.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 5 minutes.
- `update` - Default is 5 minutes.
- `delete` - Default is 15 minutes.

» Import

Instance group managers can be imported using any of these accepted formats:

```
$ terraform import google_compute_instance_group_manager.appserver projects/{{project}}/zone
$ terraform import google_compute_instance_group_manager.appserver {{project}}/{{zone}}/{{na
$ terraform import google_compute_instance_group_manager.appserver {{project}}/{{name}}
$ terraform import google_compute_instance_group_manager.appserver {{name}}
```

» 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-9"
    auto_delete  = true
    boot         = true
  }

  // Use an existing disk resource
  disk {
    // Instance Templates reference disks by name, not self link
    source      = google_compute_disk.foobar.name
    auto_delete = false
    boot        = false
  }

  network_interface {
    network = "default"
  }
```

```
  metadata = {
    foo = "bar"
  }

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

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

resource "google_compute_disk" "foobar" {
  name  = "existing-disk"
  image = data.google_compute_image.my_image.self_link
  size  = 10
  type  = "pd-ssd"
  zone  = "us-central1-a"
}
```

» 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.

» Deploying the Latest Image

A common way to use instance templates and managed instance groups is to deploy the latest image in a family, usually the latest build of your application. There are two ways to do this in Terraform, and they have their pros and cons. The difference ends up being in how "latest" is interpreted. You can either deploy the latest image available when Terraform runs, or you can have each instance check what the latest image is when it's being created, either as part of a scaling event or being rebuilt by the instance group manager.

If you're not sure, we recommend deploying the latest image available when Terraform runs, because this means all the instances in your group will be based on the same image, always, and means that no upgrades or changes to your instances happen outside of a `terraform apply`. You can achieve this by using the `google_compute_image` data source, which will retrieve the latest image on every `terraform apply`, and will update the template to use that specific image:

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

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

```
  region       = "us-central1"

  // boot disk
  disk {
    source_image = google_compute_image.my_image.self_link
  }
}
```

To have instances update to the latest on every scaling event or instance re-creation, use the family as the image for the disk, and it will use GCP's default behavior, setting the image for the template to the family:

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

  // boot disk
  disk {
    source_image = "debian-cloud/debian-9"
  }
}
```

» 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.

  To create a machine with a custom type (such as extended memory), format the value like `custom-VCPUS-MEM_IN_MB` like `custom-6-20480` for 6 vCPU and 20GB of RAM.

- `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.
- `shielded_instance_config` - (Optional) Enable Shielded VM on this instance. Shielded VM provides verifiable integrity to prevent against malware and rootkits. Defaults to disabled. Structure is documented below. **Note**: `shielded_instance_config` can only be used with boot images with shielded vm support. See the complete list here.

- `enable_display` - (Optional) Enable Virtual Displays on this instance. **Note**: `allow_stopping_for_update` must be set to true in order to update this field.

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 (**not self_link**) 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. For SCRATCH disks, the size must be exactly 375GB.
- `type` - (Optional) The type of GCE disk, can be either `"SCRATCH"` or `"PERSISTENT"`.
- `disk_encryption_key` - (Optional) Encrypts or decrypts a disk using a customer-supplied encryption key.

  If you are creating a new disk, this field encrypts the new disk using an encryption key that you provide. If you are attaching an existing disk that is already encrypted, this field decrypts the disk using the customer-supplied encryption key.

If you encrypt a disk using a customer-supplied key, you must provide the same key again when you attempt to use this resource at a later time. For example, you must provide the key when you create a snapshot or an image from the disk or when you attach the disk to a virtual machine instance.

If you do not provide an encryption key, then the disk will be encrypted using an automatically generated key and you do not need to provide a key to use the disk later.

Instance templates do not store customer-supplied encryption keys, so you cannot use your own keys to encrypt disks in a managed instance group.

The `disk_encryption_key` block supports:

- `kms_key_self_link` - (Required) The self link of the encryption key that is stored in Google Cloud KMS

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.
- `network_ip` - (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.

- `network_tier` - (Optional) The networking tier used for configuring this instance template. This field can take the following values: PREMIUM or STANDARD. If this field is not specified, it is assumed to be PREMIUM.

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 service accounts documentation explains that access scopes are the legacy method of specifying permissions for your instance. If you are following best practices and using IAM roles to grant permissions to service accounts, then you can define this field as an empty list.

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.
- `node_affinities` - (Optional) Specifies node affinities or anti-affinities to determine which sole-tenant nodes your instances and managed instance groups will use as host systems. Read more on sole-tenant node creation here. Structure 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 `node_affinities` block supports:

- `key` (Required) - The key for the node affinity label.
- `operator` (Required) - The operator. Can be `IN` for node-affinities or `NOT` for anti-affinities.
- `value` (Required) - The values for the node affinity label.

The `shielded_instance_config` block supports:

- `enable_secure_boot` (Optional) -- Verify the digital signature of all boot components, and halt the boot process if signature verification fails. Defaults to false.
- `enable_vtpm` (Optional) -- Use a virtualized trusted platform module, which is a specialized computer chip you can use to encrypt objects like keys and certificates. Defaults to true.
- `enable_integrity_monitoring` (Optional) -- Compare the most recent boot measurements to the integrity policy baseline and return a pair of pass/fail results depending on whether they match or not. Defaults to true.

» 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 any of these accepted formats:

```
$ terraform import google_compute_instance_template.default projects/{{project}}/global/inst
$ terraform import google_compute_instance_template.default {{project}}/{{name}}
$ terraform import google_compute_instance_template.default {{name}}
```

» google_compute_interconnect_attachment

Represents an InterconnectAttachment (VLAN attachment) resource. For more information, see Creating VLAN Attachments.

» Example Usage - Interconnect Attachment Basic

```
resource "google_compute_interconnect_attachment" "on_prem" {
  name         = "on-prem-attachment"
  interconnect = "my-interconnect-id"
  router       = google_compute_router.foobar.self_link
}

resource "google_compute_router" "foobar" {
  name    = "router"
  network = google_compute_network.foobar.name
}
```

» Argument Reference

The following arguments are supported:

- `router` - (Required) URL of the cloud router to be used for dynamic routing. This router must be in the same region as this InterconnectAttachment. The InterconnectAttachment will automatically connect the Interconnect to the network & region within which the Cloud Router is configured.
- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `admin_enabled` - (Optional) Whether the VLAN attachment is enabled or disabled. When using PARTNER type this will Pre-Activate the interconnect attachment
- `interconnect` - (Optional) URL of the underlying Interconnect object that this attachment's traffic will traverse through. Required if type is DEDICATED, must not be set if type is PARTNER.
- `description` - (Optional) An optional description of this resource.

- `bandwidth` - (Optional) Provisioned bandwidth capacity for the interconnect attachment. For attachments of type DEDICATED, the user can set the bandwidth. For attachments of type PARTNER, the Google Partner that is operating the interconnect must set the bandwidth. Output only for PARTNER type, mutable for PARTNER_PROVIDER and DEDICATED, Defaults to BPS_10G
- `edge_availability_domain` - (Optional) Desired availability domain for the attachment. Only available for type PARTNER, at creation time. For improved reliability, customers should configure a pair of attachments with one per availability domain. The selected availability domain will be provided to the Partner via the pairing key so that the provisioned circuit will lie in the specified domain. If not specified, the value will default to AVAILABILITY_DOMAIN_ANY.
- `type` - (Optional) The type of InterconnectAttachment you wish to create. Defaults to DEDICATED.
- `candidate_subnets` - (Optional) Up to 16 candidate prefixes that can be used to restrict the allocation of cloudRouterIpAddress and customerRouterIpAddress for this attachment. All prefixes must be within link-local address space (169.254.0.0/16) and must be /29 or shorter (/28, /27, etc). Google will attempt to select an unused /29 from the supplied candidate prefix(es). The request will fail if all possible /29s are in use on Google's edge. If not supplied, Google will randomly select an unused /29 from all of link-local space.
- `vlan_tag8021q` - (Optional) The IEEE 802.1Q VLAN tag for this attachment, in the range 2-4094. When using PARTNER type this will be managed upstream.
- `region` - (Optional) Region where the regional interconnect attachment resides.
- `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:

- `cloud_router_ip_address` - IPv4 address + prefix length to be configured on Cloud Router Interface for this interconnect attachment.
- `customer_router_ip_address` - IPv4 address + prefix length to be configured on the customer router subinterface for this interconnect attachment.

- `pairing_key` - [Output only for type PARTNER. Not present for DEDICATED]. The opaque identifier of an PARTNER attachment used to initiate provisioning with a selected partner. Of the form "XXXXX/region/domain"
- `partner_asn` - [Output only for type PARTNER. Not present for DEDICATED]. Optional BGP ASN for the router that should be supplied by a layer 3 Partner if they configured BGP on behalf of the customer.
- `private_interconnect_info` - Information specific to an InterconnectAttachment. This property is populated if the interconnect that this is attached to is of type DEDICATED. Structure is documented below.
- `state` - [Output Only] The current state of this attachment's functionality.
- `google_reference_id` - Google reference ID, to be used when raising support tickets with Google or otherwise to debug backend connectivity issues.
- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `self_link` - The URI of the created resource.

The `private_interconnect_info` block contains:

- `tag8021q` - 802.1q encapsulation tag to be used for traffic between Google and the customer, going to and from this network and region.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

InterconnectAttachment can be imported using any of these accepted formats:

```
$ terraform import google_compute_interconnect_attachment.default projects/{{project}}/regio
$ terraform import google_compute_interconnect_attachment.default {{project}}/{{region}}/{{n
$ terraform import google_compute_interconnect_attachment.default {{region}}/{{name}}
$ terraform import google_compute_interconnect_attachment.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_network

Manages a VPC network or legacy network resource on GCP.

To get more information about Network, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Network Basic

```
resource "google_compute_network" "vpc_network" {
  name = "vpc-network"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `description` - (Optional) An optional description of this resource. The resource must be recreated to modify this field.
- `auto_create_subnetworks` - (Optional) When set to `true`, the network is created in "auto subnet mode" and it will create a subnet for each region automatically across the `10.128.0.0/9` address range. When set to `false`,

the network is created in "custom subnet mode" so the user can explicitly connect subnetwork resources.

- `routing_mode` - (Optional) The network-wide routing mode to use. If set to `REGIONAL`, this network's cloud routers will only advertise routes with subnetworks of this network in the same region as the router. If set to `GLOBAL`, this network's cloud routers will advertise routes with all subnetworks of this network, across regions.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `delete_default_routes_on_create` - (Optional) If set to `true`, default routes (`0.0.0.0/0`) will be deleted immediately after network creation. Defaults to `false`.

» Attributes Reference

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

- `gateway_ipv4` - The gateway address for default routing out of the network. This value is selected by GCP.
- `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

Network can be imported using any of these accepted formats:

```
$ terraform import google_compute_network.default projects/{{project}}/global/networks/{{nam
$ terraform import google_compute_network.default {{project}}/{{name}}
$ terraform import google_compute_network.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_network_endpoint

A Network endpoint represents a IP address and port combination that is part of a specific network endpoint group (NEG). NEGs are zonals collection of these endpoints for GCP resources within a single subnet. **NOTE**: Network endpoints cannot be created outside of a network endpoint group.

To get more information about NetworkEndpoint, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Network Endpoint

```
resource "google_compute_network_endpoint" "default-endpoint" {
  network_endpoint_group = google_compute_network_endpoint_group.neg.name

  instance   = google_compute_instance.endpoint-instance.name
  port       = google_compute_network_endpoint_group.neg.default_port
  ip_address = google_compute_instance.endpoint-instance.network_interface[0].network_ip
}

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

resource "google_compute_instance" "endpoint-instance" {
  name         = "endpoint-instance"
  machine_type = "n1-standard-1"

  boot_disk {
    initialize_params {
      image = data.google_compute_image.my_image.self_link
    }
  }

  network_interface {
    subnetwork = google_compute_subnetwork.default.self_link
```

```
    access_config {
    }
  }
}

resource "google_compute_network_endpoint_group" "group" {
  name         = "my-lb-neg"
  network      = google_compute_network.default.self_link
  subnetwork   = google_compute_subnetwork.default.self_link
  default_port = "90"
  zone         = "us-central1-a"
}

resource "google_compute_network" "default" {
  name                    = "neg-network"
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "default" {
  name          = "neg-subnetwork"
  ip_cidr_range = "10.0.0.1/16"
  region        = "us-central1"
  network       = google_compute_network.default.self_link
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name for a specific VM instance that the IP address belongs to. This is required for network endpoints of type GCE_VM_IP_PORT. The instance must be in the same zone of network endpoint group.
- `port` - (Required) Port number of network endpoint.
- `ip_address` - (Required) IPv4 address of network endpoint. The IP address must belong to a VM in GCE (either the primary IP or as part of an aliased IP range).
- `network_endpoint_group` - (Required) The network endpoint group this endpoint is part of.

- `zone` - (Optional) Zone where the containing network endpoint group is located.

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

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

NetworkEndpoint can be imported using any of these accepted formats:

```
$ terraform import google_compute_network_endpoint.default projects/{{project}}/zones/{{zone
$ terraform import google_compute_network_endpoint.default {{project}}/{{zone}}/{{network_en
$ terraform import google_compute_network_endpoint.default {{zone}}/{{network_endpoint_group
$ terraform import google_compute_network_endpoint.default {{network_endpoint_group}}/{{inst
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_network_endpoint_group

Network endpoint groups (NEGs) are zonal resources that represent collections of IP address and port combinations for GCP resources within a single subnet. Each IP address and port combination is called a network endpoint.

Network endpoint groups can be used as backends in backend services for HTTP(S), TCP proxy, and SSL proxy load balancers. You cannot use NEGs as a backend with internal load balancers. Because NEG backends allow you to specify IP addresses and ports, you can distribute traffic in a granular fashion among applications or containers running within VM instances.

To get more information about NetworkEndpointGroup, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Network Endpoint Group

```
resource "google_compute_network_endpoint_group" "neg" {
  name         = "my-lb-neg"
  network      = google_compute_network.default.self_link
  subnetwork   = google_compute_subnetwork.default.self_link
  default_port = "90"
  zone         = "us-central1-a"
}

resource "google_compute_network" "default" {
  name                    = "neg-network"
  auto_create_subnetworks = false
}

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

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource; provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `network` - (Required) The network to which all network endpoints in the NEG belong. Uses "default" project network if unspecified.

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `network_endpoint_type` - (Optional) Type of network endpoints in this network endpoint group. Currently the only supported value is GCE_VM_IP_PORT.
- `subnetwork` - (Optional) Optional subnetwork to which all network endpoints in the NEG belong.
- `default_port` - (Optional) The default port used if the port number is not specified in the network endpoint.
- `zone` - (Optional) Zone where the network endpoint group is located.
- `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:

- `size` - Number of network endpoints in the network endpoint group.
- `self_link` - The URI of the created resource.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

NetworkEndpointGroup can be imported using any of these accepted formats:

```
$ terraform import google_compute_network_endpoint_group.default projects/{{project}}/zones/
$ terraform import google_compute_network_endpoint_group.default {{project}}/{{zone}}/{{name
$ terraform import google_compute_network_endpoint_group.default {{zone}}/{{name}}
$ terraform import google_compute_network_endpoint_group.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_network_peering

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

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

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) The primary network of the peering.
- `peer_network` - (Required) The peer network in the peering. The peer network may belong to a different project.
- `export_custom_routes` - (Optional, Beta) Whether to export the custom routes to the peer network. Defaults to `false`.
- `import_custom_routes` - (Optional, Beta) Whether to export the custom routes from the peer network. Defaults to `false`.

» Attributes Reference

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

- `state` - State for the peering, either `ACTIVE` or `INACTIVE`. The peering is `ACTIVE` when there's a matching configuration in the peer network.
- `state_details` - Details about the current state of the peering.

» Import

VPC network peerings can be imported using the name and project of the primary network the peering exists in and the name of the network peering

```
$ terraform import google_compute_network_peering.peering_network project-name/network-name/
```

» google_compute_node_group

Represents a NodeGroup resource to manage a group of sole-tenant nodes.

To get more information about NodeGroup, see:

- API documentation
- How-to Guides
  - Sole-Tenant Nodes

Warning: Due to limitations of the API, Terraform cannot update the number of nodes in a node group and changes to node group size either through Terraform config or through external changes will cause Terraform to delete and recreate the node group.

» Example Usage - Node Group Basic

```
data "google_compute_node_types" "central1a" {
  zone = "us-central1-a"
}

resource "google_compute_node_template" "soletenant-tmpl" {
  name      = "soletenant-tmpl"
  region    = "us-central1"
  node_type = data.google_compute_node_types.central1a.names[0]
}

resource "google_compute_node_group" "nodes" {
  name        = "soletenant-group"
  zone        = "us-central1-a"
  description = "example google_compute_node_group for Terraform Google Provider"

  size          = 1
  node_template = google_compute_node_template.soletenant-tmpl.self_link
}
```

» Argument Reference

The following arguments are supported:

- `node_template` - (Required) The URL of the node template to which this node group belongs.
- `size` - (Required) The total number of nodes in the node group.

- `description` - (Optional) An optional textual description of the resource.
- `name` - (Optional) Name of the resource.
- `zone` - (Optional) Zone where this node group is located
- `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

NodeGroup can be imported using any of these accepted formats:

```
$ terraform import google_compute_node_group.default projects/{{project}}/zones/{{zone}}/noc
$ terraform import google_compute_node_group.default {{project}}/{{zone}}/{{name}}
$ terraform import google_compute_node_group.default {{zone}}/{{name}}
$ terraform import google_compute_node_group.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_node_template

Represents a NodeTemplate resource. Node templates specify properties for creating sole-tenant nodes, such as node type, vCPU and memory requirements, node affinity labels, and region.

To get more information about NodeTemplate, see:

- API documentation
- How-to Guides
  - Sole-Tenant Nodes

» Example Usage - Node Template Basic

```
data "google_compute_node_types" "central1a" {
  zone = "us-central1-a"
}

resource "google_compute_node_template" "template" {
  name      = "soletenant-tmpl"
  region    = "us-central1"
  node_type = data.google_compute_node_types.central1a.names[0]
}
```

» Example Usage - Node Template Server Binding

```
provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}

data "google_compute_node_types" "central1a" {
  provider = google-beta
  zone     = "us-central1-a"
}

resource "google_compute_node_template" "template" {
  provider = google-beta

  name      = "soletenant-with-licenses"
  region    = "us-central1"
  node_type = data.google_compute_node_types.central1a.names[0]

  node_affinity_labels = {
    foo = "baz"
  }

  server_binding {
    type = "RESTART_NODE_ON_MINIMAL_SERVERS"
  }
}
```

» Argument Reference

The following arguments are supported:

- `description` - (Optional) An optional textual description of the resource.
- `name` - (Optional) Name of the resource.
- `node_affinity_labels` - (Optional) Labels to use for node affinity, which will be used in instance scheduling.
- `node_type` - (Optional) Node type to use for nodes group that are created from this template. Only one of nodeTypeFlexibility and nodeType can be specified.
- `node_type_flexibility` - (Optional) Flexible properties for the desired node type. Node groups that use this node template will create nodes of a type that matches these properties. Only one of nodeTypeFlexibility and nodeType can be specified. Structure is documented below.
- `server_binding` - (Optional, Beta) The server binding policy for nodes using this template. Determines where the nodes should restart following a maintenance event. Structure is documented below.
- `region` - (Optional) Region where nodes using the node template will be created. If it is not provided, the provider region is used.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `node_type_flexibility` block supports:

- `cpus` - (Optional) Number of virtual CPUs to use.
- `memory` - (Optional) Physical memory available to the node, defined in MB.
- `local_ssd` - Use local SSD

The `server_binding` block supports:

- `type` - (Required) Type of server binding policy. If `RESTART_NODE_ON_ANY_SERVER`, nodes using this template will restart on any physical server following a maintenance event. If `RESTART_NODE_ON_MINIMAL_SERVER`, nodes using this template will restart on the same physical server following a maintenance event, instead of being live migrated to or restarted on a new physical server. This option may be useful if you are using software licenses tied to the underlying server characteristics such as physical sockets or cores, to avoid the need for additional licenses when maintenance occurs. However, VMs on such nodes will experience outages while maintenance is applied.

» 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.
- `delete` - Default is 4 minutes.

» Import

NodeTemplate can be imported using any of these accepted formats:

```
$ terraform import google_compute_node_template.default projects/{{project}}/regions/{{regio
$ terraform import google_compute_node_template.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_node_template.default {{region}}/{{name}}
$ terraform import google_compute_node_template.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_project_default_network_tier

Configures the Google Compute Engine Default Network Tier for a project.

For more information, see, the Project API documentation.

» Example Usage

```
resource "google_compute_project_default_network_tier" "default" {
  network_tier = "PREMIUM"
}
```

» Argument Reference

The following arguments are supported:

- `network_tier` - (Required) The default network tier to be configured for the project. This field can take the following values: `PREMIUM` or `STANDARD`.

- `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

This resource can be imported using the project ID:

```
terraform import google_compute_project_default_network_tier.default
project-id
```

» google_compute_project_metadata

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

Note: This resource manages all project-level metadata including project-level ssh keys. Keys unset in config but set on the server will be removed. 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.

- `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

This resource can be imported using the project ID:

```
terraform import google_compute_project_metadata.foo my-project-id
```

» 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
```

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 5 minutes.
- `update` - Default is 5 minutes.
- `delete` - Default is 5 minutes.

» google_compute_region_autoscaler

Represents an Autoscaler resource.

Autoscalers allow you to automatically scale virtual machine instances in managed instance groups according to an autoscaling policy that you define.

To get more information about RegionAutoscaler, see:

- API documentation
- How-to Guides
  - Autoscaling Groups of Instances

» Example Usage - Region Autoscaler Beta

```
resource "google_compute_region_autoscaler" "foobar" {
  provider = google-beta
```

```
  name   = "my-region-autoscaler"
  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
    }
  }
}

resource "google_compute_instance_template" "foobar" {
  provider = google-beta

  name           = "my-instance-template"
  machine_type   = "n1-standard-1"
  can_ip_forward = false

  tags = ["foo", "bar"]

  disk {
    source_image = data.google_compute_image.debian_9.self_link
  }

  network_interface {
    network = "default"
  }

  metadata = {
    foo = "bar"
  }

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

resource "google_compute_target_pool" "foobar" {
  provider = google-beta

  name = "my-target-pool"
```

```
}

resource "google_compute_region_instance_group_manager" "foobar" {
  provider = google-beta

  name   = "my-region-igm"
  region = "us-central1"

  version {
    instance_template = google_compute_instance_template.foobar.self_link
    name              = "primary"
  }

  target_pools       = [google_compute_target_pool.foobar.self_link]
  base_instance_name = "foobar"
}

data "google_compute_image" "debian_9" {
  provider = google-beta

  family  = "debian-9"
  project = "debian-cloud"
}

provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}
```

» Example Usage - Region Autoscaler Basic

```
resource "google_compute_region_autoscaler" "foobar" {
  name   = "my-region-autoscaler"
  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
    }
  }
}

resource "google_compute_instance_template" "foobar" {
  name           = "my-instance-template"
  machine_type   = "n1-standard-1"
  can_ip_forward = false

  tags = ["foo", "bar"]

  disk {
    source_image = data.google_compute_image.debian_9.self_link
  }

  network_interface {
    network = "default"
  }

  metadata = {
    foo = "bar"
  }

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

resource "google_compute_target_pool" "foobar" {
  name = "my-target-pool"
}

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

  version {
    instance_template  = google_compute_instance_template.foobar.self_link
    name               = "primary"
  }

  target_pools       = [google_compute_target_pool.foobar.self_link]
  base_instance_name = "foobar"
```

```
}

data "google_compute_image" "debian_9" {
  family  = "debian-9"
  project = "debian-cloud"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. The name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `autoscaling_policy` - (Required) The configuration parameters for the autoscaling algorithm. You can define one or more of the policies for an autoscaler: cpuUtilization, customMetricUtilizations, and loadBalancingUtilization. If none of these are specified, the default will be to autoscale based on cpuUtilization to 0.6 or 60%. Structure is documented below.
- `target` - (Required) URL of the managed instance group that this autoscaler will scale.

The `autoscaling_policy` block supports:

- `min_replicas` - (Required) The minimum number of replicas that the autoscaler can scale down to. This cannot be less than 0. If not provided, autoscaler will choose a default value depending on maximum number of instances allowed.
- `max_replicas` - (Required) The maximum number of instances that the autoscaler can scale up to. This is required when creating or updating an autoscaler. The maximum number of replicas should not be lower than minimal number of replicas.
- `cooldown_period` - (Optional) The number of seconds that the autoscaler should wait before it starts collecting information from a new instance. This prevents the autoscaler from collecting information when the instance is initializing, during which the collected usage would not be reliable. The default time autoscaler waits is 60 seconds. Virtual machine initialization times might vary because of numerous factors. We recommend that you test how long an instance may take to initialize. To do this, create an instance and time the startup process.
- `cpu_utilization` - (Optional) Defines the CPU utilization policy that allows the autoscaler to scale based on the average CPU utilization of a

managed instance group. Structure is documented below.

- `metric` - (Optional) Defines the CPU utilization policy that allows the autoscaler to scale based on the average CPU utilization of a managed instance group. Structure is documented below.
- `load_balancing_utilization` - (Optional) Configuration parameters of autoscaling based on a load balancer. Structure is documented below.

The `cpu_utilization` block supports:

- `target` - (Required) The target CPU utilization that the autoscaler should maintain. Must be a float value in the range (0, 1]. If not specified, the default is 0.6. If the CPU level is below the target utilization, the autoscaler scales down the number of instances until it reaches the minimum number of instances you specified or until the average CPU of your instances reaches the target utilization. If the average CPU is above the target utilization, the autoscaler scales up until it reaches the maximum number of instances you specified or until the average utilization reaches the target utilization.

The `metric` block supports:

- `name` - (Required) The identifier (type) of the Stackdriver Monitoring metric. The metric cannot have negative values. The metric must have a value type of INT64 or DOUBLE.
- `single_instance_assignment` - (Optional, Beta) If scaling is based on a per-group metric value that represents the total amount of work to be done or resource usage, set this value to an amount assigned for a single instance of the scaled group. The autoscaler will keep the number of instances proportional to the value of this metric, the metric itself should not change value due to group resizing. For example, a good metric to use with the target is `pubsub.googleapis.com/subscription/num_undelivered_messages` or a custom metric exporting the total number of requests coming to your instances. A bad example would be a metric exporting an average or median latency, since this value can't include a chunk assignable to a single instance, it could be better used with utilization_target instead.
- `target` - (Optional) The target value of the metric that autoscaler should maintain. This must be a positive value. A utilization metric scales number of virtual machines handling requests to increase or decrease proportionally to the metric. For example, a good metric to use as a utilization-Target is www.googleapis.com/compute/instance/network/received_bytes_count. The autoscaler will work to keep this value constant for each of the instances.
- `type` - (Optional) Defines how target utilization value is expressed for a Stackdriver Monitoring metric. Either GAUGE, DELTA_PER_SECOND,

or DELTA_PER_MINUTE.

- `filter` - (Optional, Beta) A filter string to be used as the filter string for a Stackdriver Monitoring TimeSeries.list API call. This filter is used to select a specific TimeSeries for the purpose of autoscaling and to determine whether the metric is exporting per-instance or per-group data. You can only use the AND operator for joining selectors. You can only use direct equality comparison operator (=) without any functions for each selector. You can specify the metric in both the filter string and in the metric field. However, if specified in both places, the metric must be identical. The monitored resource type determines what kind of values are expected for the metric. If it is a gce_instance, the autoscaler expects the metric to include a separate TimeSeries for each instance in a group. In such a case, you cannot filter on resource labels. If the resource type is any other value, the autoscaler expects this metric to contain values that apply to the entire autoscaled instance group and resource label filtering can be performed to point autoscaler at the correct TimeSeries to scale upon. This is called a per-group metric for the purpose of autoscaling. If not specified, the type defaults to gce_instance. You should provide a filter that is selective enough to pick just one TimeSeries for the autoscaled group or for each of the instances (if you are using gce_instance resource type). If multiple TimeSeries are returned upon the query execution, the autoscaler will sum their respective values to obtain its scaling value.

The `load_balancing_utilization` block supports:

- `target` - (Required) Fraction of backend capacity utilization (set in HTTP(s) load balancing configuration) that autoscaler should maintain. Must be a positive float value. If not defined, the default is 0.8.

- `description` - (Optional) An optional description of this resource.
- `region` - (Optional) URL of the region where the instance group resides.
- `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

RegionAutoscaler can be imported using any of these accepted formats:

```
$ terraform import google_compute_region_autoscaler.default projects/{{project}}/regions/{{
$ terraform import google_compute_region_autoscaler.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_region_autoscaler.default {{region}}/{{name}}
$ terraform import google_compute_region_autoscaler.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_region_backend_service

A Region Backend Service defines a regionally-scoped group of virtual machines that will serve traffic for load balancing.

To get more information about RegionBackendService, see:

- API documentation
- How-to Guides
  - Internal TCP/UDP Load Balancing

» Example Usage - Region Backend Service Basic

```
resource "google_compute_region_backend_service" "default" {
  name                            = "region-backend-service"
  region                          = "us-central1"
  health_checks                   = [google_compute_health_check.default.self_link]
  connection_draining_timeout_sec = 10
  session_affinity                = "CLIENT_IP"
}

resource "google_compute_health_check" "default" {
  name               = "health-check"
  check_interval_sec = 1
  timeout_sec        = 1

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

» Example Usage - Region Backend Service Ilb Round Robin

```
resource "google_compute_region_backend_service" "default" {
  provider = "google-beta"

  region = "us-central1"
  name = "region-backend-service"
  health_checks = ["${google_compute_health_check.health_check.self_link}"]
  protocol = "HTTP"
  load_balancing_scheme = "INTERNAL_MANAGED"
  locality_lb_policy = "ROUND_ROBIN"
}

resource "google_compute_health_check" "health_check" {
  provider = "google-beta"

  name               = "health-check"
  http_health_check {
```

```
    port = 80
  }
}
```

» Example Usage - Region Backend Service Ilb Ring Hash

```
resource "google_compute_region_backend_service" "default" {
  provider = "google-beta"

  region = "us-central1"
  name = "region-backend-service"
  health_checks = ["${google_compute_health_check.health_check.self_link}"]
  load_balancing_scheme = "INTERNAL_MANAGED"
  locality_lb_policy = "RING_HASH"
  session_affinity = "HTTP_COOKIE"
  protocol = "HTTP"
  circuit_breakers {
    max_connections = 10
  }
  consistent_hash {
    http_cookie {
      ttl {
        seconds = 11
        nanos = 1111
      }
      name = "mycookie"
    }
  }
  outlier_detection {
    consecutive_errors = 2
  }
}

resource "google_compute_health_check" "health_check" {
  provider = "google-beta"

  name               = "health-check"
  http_health_check {
    port = 80
  }
```

```
}
```

» Argument Reference

The following arguments are supported:

- `health_checks` - (Required) The set of URLs to HealthCheck resources for health checking this RegionBackendService. Currently at most one health check can be specified, and a health check is required.
- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `affinity_cookie_ttl_sec` - (Optional, Beta) Lifetime of cookies in seconds if session_affinity is GENERATED_COOKIE. If set to 0, the cookie is non-persistent and lasts only until the end of the browser session (or equivalent). The maximum allowed value for TTL is one day. When the load balancing scheme is INTERNAL, this field is not used.
- `backend` - (Optional) The set of backends that serve this RegionBackendService. Structure is documented below.
- `circuit_breakers` - (Optional, Beta) Settings controlling the volume of connections to a backend service. This field is applicable only when the `load_balancing_scheme` is set to INTERNAL_MANAGED and the `protocol` is set to HTTP, HTTPS, or HTTP2. Structure is documented below.
- `consistent_hash` - (Optional, Beta) Consistent Hash-based load balancing can be used to provide soft session affinity based on HTTP headers, cookies or other properties. This load balancing policy is applicable only for HTTP connections. The affinity to a particular destination host will be lost when one or more hosts are added/removed from the destination service. This field specifies parameters that control consistent hashing. This field only applies when all of the following are true -
  - `load_balancing_scheme` is set to INTERNAL_MANAGED
  - `protocol` is set to HTTP, HTTPS, or HTTP2
  - `locality_lb_policy` is set to MAGLEV or RING_HASH Structure is documented below.

- `connection_draining_timeout_sec` - (Optional) Time for which instance will be drained (not accept new connections, but still work to finish started).
- `description` - (Optional) An optional description of this resource.
- `failover_policy` - (Optional, Beta) Policy for failovers. Structure is documented below.
- `load_balancing_scheme` - (Optional) Indicates what kind of load balancing this regional backend service will be used for. A backend service created for one type of load balancing cannot be used with the other(s). Must be `INTERNAL` or `INTERNAL_MANAGED`. Defaults to `INTERNAL`.
- `locality_lb_policy` - (Optional, Beta) The load balancing algorithm used within the scope of the locality. The possible values are - ROUND_ROBIN - This is a simple policy in which each healthy backend is selected in round robin order. LEAST_REQUEST - An O(1) algorithm which selects two random healthy hosts and picks the host which has fewer active requests. RING_HASH - The ring/modulo hash load balancer implements consistent hashing to backends. The algorithm has the property that the addition/removal of a host from a set of N hosts only affects 1/N of the requests. RANDOM - The load balancer selects a random healthy host. ORIGINAL_DESTINATION - Backend host is selected based on the client connection metadata, i.e., connections are opened to the same address as the destination address of the incoming connection before the connection was redirected to the load balancer. MAGLEV - used as a drop in replacement for the ring hash load balancer. Maglev is not as stable as ring hash but has faster table lookup build times and host selection times. For more information about Maglev, refer to https://ai.google/research/pubs/pub44824 This field is applicable only when the `load_balancing_scheme` is set to INTERNAL_MANAGED and the `protocol` is set to HTTP, HTTPS, or HTTP2.
- `outlier_detection` - (Optional, Beta) Settings controlling eviction of unhealthy hosts from the load balancing pool. This field is applicable only when the `load_balancing_scheme` is set to INTERNAL_MANAGED and the `protocol` is set to HTTP, HTTPS, or HTTP2. Structure is documented below.
- `protocol` - (Optional) The protocol this RegionBackendService uses to communicate with backends. Possible values are HTTP, HTTPS, HTTP2, SSL, TCP, and UDP. The default is HTTP. **NOTE**: HTTP2 is only valid for beta HTTP/2 load balancer types and may result in errors if used with the GA API.
- `session_affinity` - (Optional) Type of session affinity to use. The default is NONE. Session affinity is not applicable if the protocol is UDP.
- `timeout_sec` - (Optional) How many seconds to wait for the backend

before considering it a failed request. Default is 30 seconds. Valid range is [1, 86400].

- `log_config` - (Optional, Beta) This field denotes the logging options for the load balancer traffic served by this backend service. If logging is enabled, logs will be exported to Stackdriver. Structure is documented below.
- `network` - (Optional, Beta) The URL of the network to which this backend service belongs. This field can only be specified when the load balancing scheme is set to INTERNAL.
- `region` - (Optional) The Region in which the created backend service should reside. If it is not provided, the provider region is used.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `backend` block supports:

- `balancing_mode` - (Optional) Specifies the balancing mode for this backend. Defaults to CONNECTION.
- `capacity_scaler` - (Optional) A multiplier applied to the group's maximum servicing capacity (based on UTILIZATION, RATE or CONNECTION). A setting of 0 means the group is completely drained, offering 0% of its available Capacity. Valid range is [0.0,1.0].
- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `failover` - (Optional, Beta) This field designates whether this is a failover backend. More than one failover backend can be configured for a given RegionBackendService.
- `group` - (Required) The fully-qualified URL of an Instance Group or Network Endpoint Group resource. In case of instance group this defines the list of instances that serve traffic. Member virtual machine instances from each instance group must live in the same zone as the instance group itself. No two backends in a backend service are allowed to use same Instance Group resource. For Network Endpoint Groups this defines list of endpoints. All endpoints of Network Endpoint Group must be hosted on instances located in the same zone as the Network Endpoint Group. Backend services cannot mix Instance Group and Network Endpoint Group backends. When the `load_balancing_scheme` is INTERNAL, only instance groups are supported. Note that you must specify an Instance Group or Network Endpoint Group resource using the fully-qualified URL, rather than a partial URL.
- `max_connections` - (Optional) The max number of simultaneous connections for the group. Can be used with either CONNECTION or UTI-

LIZATION balancing modes. For CONNECTION mode, either maxConnections or one of maxConnectionsPerInstance or maxConnectionsPerEndpoint, as appropriate for group type, must be set.

- `max_connections_per_instance` - (Optional) The max number of simultaneous connections that a single backend instance can handle. This is used to calculate the capacity of the group. Can be used in either CONNECTION or UTILIZATION balancing modes. For CONNECTION mode, either maxConnections or maxConnectionsPerInstance must be set.
- `max_connections_per_endpoint` - (Optional) The max number of simultaneous connections that a single backend network endpoint can handle. This is used to calculate the capacity of the group. Can be used in either CONNECTION or UTILIZATION balancing modes. For CONNECTION mode, either maxConnections or maxConnectionsPerEndpoint must be set.
- `max_rate` - (Optional) The max requests per second (RPS) of the group. Can be used with either RATE or UTILIZATION balancing modes, but required if RATE mode. Either maxRate or one of maxRatePerInstance or maxRatePerEndpoint, as appropriate for group type, must be set.
- `max_rate_per_instance` - (Optional) The max requests per second (RPS) that a single backend instance can handle. This is used to calculate the capacity of the group. Can be used in either balancing mode. For RATE mode, either maxRate or maxRatePerInstance must be set.
- `max_rate_per_endpoint` - (Optional) The max requests per second (RPS) that a single backend network endpoint can handle. This is used to calculate the capacity of the group. Can be used in either balancing mode. For RATE mode, either maxRate or maxRatePerEndpoint must be set.
- `max_utilization` - (Optional) Used when balancingMode is UTILIZATION. This ratio defines the CPU utilization target for the group. Valid range is [0.0, 1.0].

The `circuit_breakers` block supports:

- `connect_timeout` - (Optional) The timeout for new network connections to hosts. Structure is documented below.
- `max_requests_per_connection` - (Optional) Maximum requests for a single backend connection. This parameter is respected by both the HTTP/1.1 and HTTP/2 implementations. If not specified, there is no limit. Setting this parameter to 1 will effectively disable keep alive.
- `max_connections` - (Optional) The maximum number of connections to the backend cluster. Defaults to 1024.
- `max_pending_requests` - (Optional) The maximum number of pending requests to the backend cluster. Defaults to 1024.

- `max_requests` - (Optional) The maximum number of parallel requests to the backend cluster. Defaults to 1024.
- `max_retries` - (Optional) The maximum number of parallel retries to the backend cluster. Defaults to 3.

The `connect_timeout` block supports:

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.
- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 seconds field and a positive nanos field. Must be from 0 to 999,999,999 inclusive.

The `consistent_hash` block supports:

- `http_cookie` - (Optional) Hash is based on HTTP Cookie. This field describes a HTTP cookie that will be used as the hash key for the consistent hash load balancer. If the cookie is not present, it will be generated. This field is applicable if the sessionAffinity is set to HTTP_COOKIE. Structure is documented below.
- `http_header_name` - (Optional) The hash based on the value of the specified header field. This field is applicable if the sessionAffinity is set to HEADER_FIELD.
- `minimum_ring_size` - (Optional) The minimum number of virtual nodes to use for the hash ring. Larger ring sizes result in more granular load distributions. If the number of hosts in the load balancing pool is larger than the ring size, each host will be assigned a single virtual node. Defaults to 1024.

The `http_cookie` block supports:

- `ttl` - (Optional) Lifetime of the cookie. Structure is documented below.
- `name` - (Optional) Name of the cookie.
- `path` - (Optional) Path to set for the cookie.

The `ttl` block supports:

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.
- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 seconds field and a positive nanos field. Must be from 0 to 999,999,999 inclusive.

The `failover_policy` block supports:

- `disable_connection_drain_on_failover` - (Optional) On failover or failback, this field indicates whether connection drain will be honored.

Setting this to true has the following effect: connections to the old active pool are not drained. Connections to the new active pool use the timeout of 10 min (currently fixed). Setting to false has the following effect: both old and new connections will have a drain timeout of 10 min. This can be set to true only if the protocol is TCP. The default is false.

- `drop_traffic_if_unhealthy` - (Optional) This option is used only when no healthy VMs are detected in the primary and backup instance groups. When set to true, traffic is dropped. When set to false, new connections are sent across all VMs in the primary group. The default is false.
- `failover_ratio` - (Optional) The value of the field must be in [0, 1]. If the ratio of the healthy VMs in the primary backend is at or below this number, traffic arriving at the load-balanced IP will be directed to the failover backend. In case where 'failoverRatio' is not set or all the VMs in the backup backend are unhealthy, the traffic will be directed back to the primary backend in the "force" mode, where traffic will be spread to the healthy VMs with the best effort, or to all VMs when no VM is healthy. This field is only used with l4 load balancing.

The `outlier_detection` block supports:

- `base_ejection_time` - (Optional) The base time that a host is ejected for. The real time is equal to the base time multiplied by the number of times the host has been ejected. Defaults to 30000ms or 30s. Structure is documented below.
- `consecutive_errors` - (Optional) Number of errors before a host is ejected from the connection pool. When the backend host is accessed over HTTP, a 5xx return code qualifies as an error. Defaults to 5.
- `consecutive_gateway_failure` - (Optional) The number of consecutive gateway failures (502, 503, 504 status or connection errors that are mapped to one of those status codes) before a consecutive gateway failure ejection occurs. Defaults to 5.
- `enforcing_consecutive_errors` - (Optional) The percentage chance that a host will be actually ejected when an outlier status is detected through consecutive 5xx. This setting can be used to disable ejection or to ramp it up slowly. Defaults to 100.
- `enforcing_consecutive_gateway_failure` - (Optional) The percentage chance that a host will be actually ejected when an outlier status is detected through consecutive gateway failures. This setting can be used to disable ejection or to ramp it up slowly. Defaults to 0.
- `enforcing_success_rate` - (Optional) The percentage chance that a host will be actually ejected when an outlier status is detected through success rate statistics. This setting can be used to disable ejection or to ramp it up slowly. Defaults to 100.

- `interval` - (Optional) Time interval between ejection sweep analysis. This can result in both new ejections as well as hosts being returned to service. Defaults to 10 seconds. Structure is documented below.
- `max_ejection_percent` - (Optional) Maximum percentage of hosts in the load balancing pool for the backend service that can be ejected. Defaults to 10%.
- `success_rate_minimum_hosts` - (Optional) The number of hosts in a cluster that must have enough request volume to detect success rate outliers. If the number of hosts is less than this setting, outlier detection via success rate statistics is not performed for any host in the cluster. Defaults to 5.
- `success_rate_request_volume` - (Optional) The minimum number of total requests that must be collected in one interval (as defined by the interval duration above) to include this host in success rate based outlier detection. If the volume is lower than this setting, outlier detection via success rate statistics is not performed for that host. Defaults to 100.
- `success_rate_stdev_factor` - (Optional) This factor is used to determine the ejection threshold for success rate outlier ejection. The ejection threshold is the difference between the mean success rate, and the product of this factor and the standard deviation of the mean success rate: mean - (stdev * success_rate_stdev_factor). This factor is divided by a thousand to get a double. That is, if the desired factor is 1.9, the runtime value should be 1900. Defaults to 1900.

The `base_ejection_time` block supports:

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.
- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.

The `interval` block supports:

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.
- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.

The `log_config` block supports:

- `enable` - (Optional) Whether to enable logging for the load balancer traffic served by this backend service.

- `sample_rate` - (Optional) This field can only be specified if logging is enabled for this backend service. The value of the field must be in [0, 1]. This configures the sampling rate of requests to the load balancer where 1.0 means all logged requests are reported and 0.0 means no logged requests are reported. The default value is 1.0.

» Attributes Reference

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

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `fingerprint` - Fingerprint of this resource. A hash of the contents stored in this object. This field is used in optimistic locking.
- `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

RegionBackendService can be imported using any of these accepted formats:

```
$ terraform import google_compute_region_backend_service.default projects/{{project}}/regio
$ terraform import google_compute_region_backend_service.default {{project}}/{{region}}/{{na
$ terraform import google_compute_region_backend_service.default {{region}}/{{name}}
$ terraform import google_compute_region_backend_service.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_region_disk

Persistent disks are durable storage devices that function similarly to the physical disks in a desktop or a server. Compute Engine manages the hardware behind these devices to ensure data redundancy and optimize performance for you. Persistent disks are available as either standard hard disk drives (HDD) or solid-state drives (SSD).

Persistent disks are located independently from your virtual machine instances, so you can detach or move persistent disks to keep your data even after you delete your instances. Persistent disk performance scales automatically with size, so you can resize your existing persistent disks or add more persistent disks to an instance to meet your performance and storage space requirements.

Add a persistent disk to your instance when you need reliable and affordable storage with consistent performance characteristics.

To get more information about RegionDisk, see:

- API documentation
- How-to Guides
  - Adding or Resizing Regional Persistent Disks

Warning: 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 - Region Disk Basic

```
resource "google_compute_region_disk" "regiondisk" {
  name                      = "my-region-disk"
  snapshot                  = google_compute_snapshot.snapdisk.self_link
  type                      = "pd-ssd"
  region                    = "us-central1"
  physical_block_size_bytes = 4096

  replica_zones = ["us-central1-a", "us-central1-f"]
}

resource "google_compute_disk" "disk" {
  name  = "my-disk"
  image = "debian-cloud/debian-9"
  size  = 50
```

```
  type  = "pd-ssd"
  zone  = "us-central1-a"
}

resource "google_compute_snapshot" "snapdisk" {
  name        = "my-snapshot"
  source_disk = google_compute_disk.disk.name
  zone        = "us-central1-a"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `replica_zones` - (Required) URLs of the zones where the disk should be replicated to.

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `labels` - (Optional) Labels to apply to this disk. A list of key->value pairs.
- `size` - (Optional) Size of the persistent disk, specified in GB. You can specify this field when creating a persistent disk using the sourceImage or sourceSnapshot parameter, or specify it alone to create an empty persistent disk. If you specify this field along with sourceImage or sourceSnapshot, the value of sizeGb must not be less than the size of the sourceImage or the size of the snapshot.
- `physical_block_size_bytes` - (Optional) Physical block size of the persistent disk, in bytes. If not present in a request, a default value is used. Currently supported sizes are 4096 and 16384, other sizes may be added in the future. If an unsupported value is requested, the error message will list the supported values for the caller's project.
- `type` - (Optional) URL of the disk type resource describing which disk type to use to create the disk. Provide this when creating the disk.

- `region` - (Optional) A reference to the region where the disk resides.
- `disk_encryption_key` - (Optional) Encrypts the disk using a customer-supplied encryption key. After you encrypt a disk with a customer-supplied key, you must provide the same key if you use the disk later (e.g. to create a disk snapshot or an image, or to attach the disk to a virtual machine). Customer-supplied encryption keys do not protect access to metadata of the disk. If you do not provide an encryption key when creating the disk, then the disk will be encrypted using an automatically generated key and you do not need to provide a key to use the disk later. Structure is documented below.
- `snapshot` - (Optional) The source snapshot used to create this disk. You can provide this as a partial or full URL to the resource. For example, the following are valid values:
  - `https://www.googleapis.com/compute/v1/projects/project/global/snapshots/snapshot`
  - `projects/project/global/snapshots/snapshot`
  - `global/snapshots/snapshot`
  - `snapshot`
- `source_snapshot_encryption_key` - (Optional) The customer-supplied encryption key of the source snapshot. Required if the source snapshot is protected by a customer-supplied encryption key. 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.

The `disk_encryption_key` block supports:

- `raw_key` - (Optional) Specifies a 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to either encrypt or decrypt this resource.
- `sha256` - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.
- `kms_key_name` - (Optional, Beta) The name of the encryption key that is stored in Google Cloud KMS.

The `source_snapshot_encryption_key` block supports:

- `raw_key` - (Optional) Specifies a 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to either encrypt or decrypt this resource.
- `kms_key_name` - (Optional, Beta) The name of the encryption key that is stored in Google Cloud KMS.
- `sha256` - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.

» Attributes Reference

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

- `label_fingerprint` - The fingerprint used for optimistic locking of this resource. Used internally during updates.
- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `last_attach_timestamp` - Last attach timestamp in RFC3339 text format.
- `last_detach_timestamp` - Last detach timestamp in RFC3339 text format.
- `users` - Links to the users of the disk (attached instances) in form: project/zones/zone/instances/instance
- `source_snapshot_id` - The unique ID of the snapshot used to create this disk. This value identifies the exact snapshot that was used to create this persistent disk. For example, if you created the persistent disk from a snapshot that was later deleted and recreated under the same name, the source snapshot ID would identify the exact version of the snapshot that was used.
- `self_link` - The URI of the created resource.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 5 minutes.
- `update` - Default is 4 minutes.
- `delete` - Default is 4 minutes.

» Import

RegionDisk can be imported using any of these accepted formats:

```
$ terraform import google_compute_region_disk.default projects/{{project}}/regions/{{region}
$ terraform import google_compute_region_disk.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_region_disk.default {{region}}/{{name}}
$ terraform import google_compute_region_disk.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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 with top level instance template (google provider)

```
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"
  region                     = "us-central1"
  distribution_policy_zones  = ["us-central1-a", "us-central1-f"]

  version {
    instance_template = google_compute_instance_template.appserver.self_link
  }

  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
  }
}
```

» Example Usage with multiple versions

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

  base_instance_name = "app"
  region             = "us-central1"

  target_size = 5

  version {
    instance_template = google_compute_instance_template.appserver.self_link
  }

  version {
    instance_template = google_compute_instance_template.appserver-canary.self_link
    target_size {
      fixed = 1
    }
  }
}
```

» 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.

- `version` - (Required) Application versions managed by this instance group. Each version deals with a specific instance template, allowing canary release scenarios. Structure is documented below.
- `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.
- `wait_for_instances` - (Optional) Whether to wait for all instances to be created/updated before returning. Note that if this is set to true and the operation does not succeed, Terraform will continue trying until it times out.

- `auto_healing_policies` - (Optional) 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.
- `update_policy` - (Optional) The update policy for this managed instance group. Structure is documented below. For more information, see the official documentation and API
- `distribution_policy_zones` - (Optional) The distribution policy for this managed instance group. You can specify one or more values. For more information, see the official documentation.

The `update_policy` block supports:

```
update_policy {
  type                         = "PROACTIVE"
  instance_redistribution_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. You can specify either `RESTART` to restart existing instances or `REPLACE` to delete and create new instances from the target template. If you specify a `RESTART`, the Updater will attempt to perform that action only. However, if the Updater determines that the minimal action you specify is not enough to perform the update, it might perform a more disruptive action.
- `type` - (Required) - The type of update process. You can specify either `PROACTIVE` so that the instance group manager proactively executes actions in order to bring instances to their target versions or `OPPORTUNISTIC` so that no action is proactively executed but the update will be performed as part of other actions (for example, resizes or recreateInstances calls).
- `instance_redistribution_type` - (Optional) - The instance redistribution policy for regional managed instance groups. Valid values are: `"PROACTIVE"`, `"NONE"`. If `PROACTIVE` (default), the group attempts to maintain an even distribution of VM instances across zones in the region. If `NONE`, proactive redistribution is disabled.
- `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`. It has to be either 0 or at least equal to the number of zones. If fixed values are used, at least one of `max_unavailable_fixed` or `max_surge_fixed` must be greater than 0.
- `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`. Percent value is only allowed for regional managed instance groups with size at least 10.
- `max_unavailable_fixed` - (Optional), The maximum number of instances that can be unavailable during the update process. Conflicts with `max_unavailable_percent`. It has to be either 0 or at least equal to the number of zones. If fixed values are used, at least one of `max_unavailable_fixed` or `max_surge_fixed` must be greater than 0.
- `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`. Percent value is

only allowed for regional managed instance groups with size at least 10.

- `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.

The `version` block supports:

```
version {
  name              = "appserver-canary"
  instance_template = google_compute_instance_template.appserver-canary.self_link

  target_size {
    fixed = 1
  }
}

version {
  name              = "appserver-canary"
  instance_template = google_compute_instance_template.appserver-canary.self_link

  target_size {
    percent = 20
  }
}
```

- `name` - (Required) - Version name.
- `instance_template` - (Required) - The full URL to an instance template from which all new instances of this version will be created.

- `target_size` - (Optional) - The number of instances calculated as a fixed number or a percentage depending on the settings. Structure is documented below.

Exactly one `version` you specify must not have a `target_size` specified. During a rolling update, the instance group manager will fulfill the `target_size` constraints of every other `version`, and any remaining instances will be provisioned with the version where `target_size` is unset.

The `target_size` block supports:

- `fixed` - (Optional), The number of instances which are managed for this version. Conflicts with `percent`.
- `percent` - (Optional), The number of instances (calculated as percentage) which are managed for this version. Conflicts with `fixed`. Note that when using `percent`, rounding will be in favor of explicitly set `target_size` values; a managed instance group with 2 instances and 2 `version`s, one of which has a `target_size.percent` of `60` will create 2 instances of that `version`.

» 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.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 5 minutes.
- `update` - Default is 5 minutes.
- `delete` - Default is 15 minutes.

» 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_region_ssl_certificate

A RegionSslCertificate resource, used for HTTPS load balancing. This resource provides a mechanism to upload an SSL key and certificate to the load balancer to serve secure connections from the user.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about RegionSslCertificate, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Region Ssl Certificate Basic

```
resource "google_compute_region_ssl_certificate" "default" {
  provider    = google-beta
  region      = "us-central1"
  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
  }
}
```

» Example Usage - Region Ssl Certificate Random Provider

```
# You may also want to control name generation explicitly:
resource "google_compute_region_ssl_certificate" "default" {
```

```
  provider = google-beta
  region   = "us-central1"

  # The name will contain 8 random hex digits,
  # e.g. "my-certificate-48ab27cd2a"
  name        = random_id.certificate.hex
  private_key = file("path/to/private.key")
  certificate = file("path/to/certificate.crt")

  lifecycle {
    create_before_destroy = true
  }
}

resource "random_id" "certificate" {
  byte_length = 4
  prefix      = "my-certificate-"

  # For security, do not expose raw certificate values in the output
  keepers = {
    private_key = filebase64sha256("path/to/private.key")
    certificate = filebase64sha256("path/to/certificate.crt")
  }
}
```

» Example Usage - Region Ssl Certificate Target Https Proxies

```
// Using with Region 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, specify a partial
// name with name_prefix, or use random_id resource. Example:

resource "google_compute_region_ssl_certificate" "default" {
```

```
  provider    = google-beta
  region      = "us-central1"
  name_prefix = "my-certificate-"
  private_key = file("path/to/private.key")
  certificate = file("path/to/certificate.crt")

  lifecycle {
    create_before_destroy = true
  }
}

resource "google_compute_region_target_https_proxy" "default" {
  provider         = google-beta
  region           = "us-central1"
  name             = "test-proxy"
  url_map          = google_compute_region_url_map.default.self_link
  ssl_certificates = [google_compute_region_ssl_certificate.default.self_link]
}

resource "google_compute_region_url_map" "default" {
  provider    = google-beta
  region      = "us-central1"
  name        = "url-map"
  description = "a description"

  default_service = google_compute_region_backend_service.default.self_link

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

  path_matcher {
    name            = "allpaths"
    default_service = google_compute_region_backend_service.default.self_link

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

resource "google_compute_region_backend_service" "default" {
  provider    = google-beta
  region      = "us-central1"
```

```
  name        = "backend-service"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = [google_compute_region_health_check.default.self_link]
}

resource "google_compute_region_health_check" "default" {
  provider = google-beta
  region   = "us-central1"
  name     = "http-health-check"
  http_health_check {
    port = 80
  }
}
```

» Argument Reference

The following arguments are supported:

- `certificate` - (Required) The certificate in PEM format. The certificate chain must be no greater than 5 certs long. The chain must include at least one intermediate cert.
- `private_key` - (Required) The write-only private key in PEM format.

- `description` - (Optional) An optional description of this resource.
- `name` - (Optional) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

These are in the same namespace as the managed SSL certificates.

- `region` - (Optional) The Region in which the created regional ssl certificate should reside. If it is not provided, the provider region is used.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.

» Attributes Reference

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

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

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

RegionSslCertificate can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_compute_region_ssl_certificate.default proje
$ terraform import -provider=google-beta google_compute_region_ssl_certificate.default {{pro
$ terraform import -provider=google-beta google_compute_region_ssl_certificate.default {{reg
$ terraform import -provider=google-beta google_compute_region_ssl_certificate.default {{nam
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_region_target_http_proxy

Represents a RegionTargetHttpProxy resource, which is used by one or more forwarding rules to route incoming HTTP requests to a URL map.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about RegionTargetHttpProxy, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Region Target Http Proxy Basic

```
resource "google_compute_region_target_http_proxy" "default" {
  provider = google-beta

  region  = "us-central1"
  name    = "test-proxy"
  url_map = google_compute_region_url_map.default.self_link
}

resource "google_compute_region_url_map" "default" {
  provider = google-beta

  region          = "us-central1"
  name            = "url-map"
  default_service = google_compute_region_backend_service.default.self_link

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

  path_matcher {
    name            = "allpaths"
    default_service = google_compute_region_backend_service.default.self_link

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

resource "google_compute_region_backend_service" "default" {
  provider = google-beta
```

```
  region      = "us-central1"
  name        = "backend-service"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = [google_compute_region_health_check.default.self_link]
}

resource "google_compute_region_health_check" "default" {
  provider = google-beta

  region = "us-central1"
  name   = "http-health-check"
  http_health_check {
    port = 80
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `url_map` - (Required) A reference to the RegionUrlMap resource that defines the mapping from URL to the BackendService.

- `description` - (Optional) An optional description of this resource.
- `region` - (Optional) The Region in which the created target https proxy should reside. If it is not provided, the provider region is used.
- `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.
- `proxy_id` - The unique identifier for the resource.
- `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

RegionTargetHttpProxy can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_compute_region_target_http_proxy.default pro
$ terraform import -provider=google-beta google_compute_region_target_http_proxy.default {{p
$ terraform import -provider=google-beta google_compute_region_target_http_proxy.default {{r
$ terraform import -provider=google-beta google_compute_region_target_http_proxy.default {{r
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_region_target_https_proxy

Represents a RegionTargetHttpsProxy resource, which is used by one or more forwarding rules to route incoming HTTPS requests to a URL map.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about RegionTargetHttpsProxy, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Region Target Https Proxy Basic

```
resource "google_compute_region_target_https_proxy" "default" {
  provider = google-beta

  region           = "us-central1"
  name             = "test-proxy"
  url_map          = google_compute_region_url_map.default.self_link
  ssl_certificates = [google_compute_region_ssl_certificate.default.self_link]
}

resource "google_compute_region_ssl_certificate" "default" {
  provider = google-beta

  region      = "us-central1"
  name        = "my-certificate"
  private_key = file("path/to/private.key")
  certificate = file("path/to/certificate.crt")
}

resource "google_compute_region_url_map" "default" {
  provider = google-beta

  region      = "us-central1"
  name        = "url-map"
  description = "a description"

  default_service = google_compute_region_backend_service.default.self_link

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

  path_matcher {
    name            = "allpaths"
    default_service = google_compute_region_backend_service.default.self_link

    path_rule {
```

```
      paths   = ["/*"]
      service = google_compute_region_backend_service.default.self_link
    }
  }
}

resource "google_compute_region_backend_service" "default" {
  provider = google-beta

  region      = "us-central1"
  name        = "backend-service"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = [google_compute_region_health_check.default.self_link]
}

resource "google_compute_region_health_check" "default" {
  provider = google-beta

  region = "us-central1"
  name   = "http-health-check"
  http_health_check {
    port = 80
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `ssl_certificates` - (Required) A list of RegionSslCertificate resources that are used to authenticate connections between users and the load balancer. Currently, exactly one SSL certificate must be specified.
- `url_map` - (Required) A reference to the RegionUrlMap resource that defines the mapping from URL to the RegionBackendService.

- `description` - (Optional) An optional description of this resource.
- `region` - (Optional) The Region in which the created target https proxy should reside. If it is not provided, the provider region is used.
- `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.
- `proxy_id` - The unique identifier for the resource.
- `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

RegionTargetHttpsProxy can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_compute_region_target_https_proxy.default pr
$ terraform import -provider=google-beta google_compute_region_target_https_proxy.default {+
$ terraform import -provider=google-beta google_compute_region_target_https_proxy.default {+
$ terraform import -provider=google-beta google_compute_region_target_https_proxy.default {+
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_reservation

Represents a reservation resource. A reservation ensures that capacity is held in a specific zone even if the reserved VMs are not running.

Reservations apply only to Compute Engine, Cloud Dataproc, and Google Kubernetes Engine VM usage.Reservations do not apply to `f1-micro` or `g1-small` machine types, preemptible VMs, sole tenant nodes, or other services not listed above like Cloud SQL and Dataflow.

To get more information about Reservation, see:

- API documentation
- How-to Guides
  - Reserving zonal resources

» Example Usage - Reservation Basic

```
resource "google_compute_reservation" "gce_reservation" {
  name = "gce-reservation"
  zone = "us-central1-a"

  specific_reservation {
    count = 1
    instance_properties {
      min_cpu_platform = "Intel Cascade Lake"
      machine_type     = "n2-standard-2"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which

means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `specific_reservation` - (Required) Reservation for instances with specific machine shapes. Structure is documented below.
- `zone` - (Required) The zone where the reservation is made.

The `specific_reservation` block supports:

- `count` - (Required) The number of resources that are allocated.
- `in_use_count` - How many instances are in use.
- `instance_properties` - (Required) The instance properties for the reservation. Structure is documented below.

The `instance_properties` block supports:

- `machine_type` - (Required) The name of the machine type to reserve.
- `min_cpu_platform` - (Optional) The minimum CPU platform for the reservation. For example, `"Intel Skylake"`. See the CPU platform availability reference](https://cloud.google.com/compute/docs/instances/specify-min-cpu-platform#availablezones) for information on available CPU platforms.
- `guest_accelerators` - (Optional) Guest accelerator type and count. Structure is documented below.
- `local_ssds` - (Optional) The amount of local ssd to reserve with each instance. This reserves disks of type `local-ssd`. Structure is documented below.

The `guest_accelerators` block supports:

- `accelerator_type` - (Required) The full or partial URL of the accelerator type to attach to this instance. For example: `projects/my-project/zones/us-central1-c/accelerato`
  If you are creating an instance template, specify only the accelerator name.
- `accelerator_count` - (Required) The number of the guest accelerator cards exposed to this instance.

The `local_ssds` block supports:

- `interface` - (Optional) The disk interface to use for attaching this disk, one of `SCSI` or `NVME`. The default is `SCSI`.
- `disk_size_gb` - (Required) The size of the disk in base-2 GB.

- `description` - (Optional) An optional description of this resource.

- `specific_reservation_required` - (Optional) When set to true, only VMs that target this reservation by name can consume this reservation. Otherwise, it can be consumed by VMs with affinity for any reservation. Defaults to false.
- `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.
- `commitment` - Full or partial URL to a parent commitment. This field displays for reservations that are tied to a commitment.
- `status` - The status of the reservation.
- `self_link` - The URI of the created resource.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

Reservation can be imported using any of these accepted formats:

```
$ terraform import google_compute_reservation.default projects/{{project}}/zones/{{zone}}/re
$ terraform import google_compute_reservation.default {{project}}/{{zone}}/{{name}}
$ terraform import google_compute_reservation.default {{zone}}/{{name}}
$ terraform import google_compute_reservation.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_resource_policy

A policy that can be attached to a resource to specify or schedule actions on that resource.

» Example Usage - Resource Policy Basic

```
resource "google_compute_resource_policy" "foo" {
  name   = "policy"
  region = "us-central1"
  snapshot_schedule_policy {
    schedule {
      daily_schedule {
        days_in_cycle = 1
        start_time    = "04:00"
      }
    }
  }
}
```

» Example Usage - Resource Policy Full

```
resource "google_compute_resource_policy" "bar" {
  name   = "policy"
  region = "us-central1"
  snapshot_schedule_policy {
    schedule {
      hourly_schedule {
        hours_in_cycle = 20
        start_time     = "23:00"
      }
    }
    retention_policy {
```

```
      max_retention_days    = 10
      on_source_disk_delete = "KEEP_AUTO_SNAPSHOTS"
    }
    snapshot_properties {
      labels = {
        my_label = "value"
      }
      storage_locations = ["us"]
      guest_flush       = true
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the resource, provided by the client when initially creating the resource. The resource name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `snapshot_schedule_policy` - (Optional) Policy for creating snapshots of persistent disks. Structure is documented below.
- `region` - (Optional) Region where resource policy resides.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `snapshot_schedule_policy` block supports:

- `schedule` - (Required) Contains one of an `hourlySchedule`, `dailySchedule`, or `weeklySchedule`. Structure is documented below.
- `retention_policy` - (Optional) Retention policy applied to snapshots created by this resource policy. Structure is documented below.
- `snapshot_properties` - (Optional) Properties with which the snapshots are created, such as labels. Structure is documented below.

The `schedule` block supports:

- `hourly_schedule` - (Optional) The policy will execute every nth hour starting at the specified time. Structure is documented below.

- `daily_schedule` - (Optional) The policy will execute every nth day at the specified time. Structure is documented below.
- `weekly_schedule` - (Optional) Allows specifying a snapshot time for each day of the week. Structure is documented below.

The `hourly_schedule` block supports:

- `hours_in_cycle` - (Required) The number of hours between snapshots.
- `start_time` - (Required) Time within the window to start the operations. It must be in format "HH:MM", where HH : [00-23] and MM : [00-00] GMT.

The `daily_schedule` block supports:

- `days_in_cycle` - (Required) The number of days between snapshots.
- `start_time` - (Required) This must be in UTC format that resolves to one of 00:00, 04:00, 08:00, 12:00, 16:00, or 20:00. For example, both 13:00-5 and 08:00 are valid.

The `weekly_schedule` block supports:

- `day_of_weeks` - (Required) May contain up to seven (one for each day of the week) snapshot times. Structure is documented below.

The `day_of_weeks` block supports:

- `start_time` - (Required) Time within the window to start the operations. It must be in format "HH:MM", where HH : [00-23] and MM : [00-00] GMT.
- `day` - (Required) The day of the week to create the snapshot. e.g. MONDAY

The `retention_policy` block supports:

- `max_retention_days` - (Required) Maximum age of the snapshot that is allowed to be kept.
- `on_source_disk_delete` - (Optional) Specifies the behavior to apply to scheduled snapshots when the source disk is deleted. Valid options are KEEP_AUTO_SNAPSHOTS and APPLY_RETENTION_POLICY

The `snapshot_properties` block supports:

- `labels` - (Optional) A set of key-value pairs.
- `storage_locations` - (Optional) GCS bucket location in which to store the snapshot (regional or multi-regional).
- `guest_flush` - (Optional) Whether to perform a 'guest aware' snapshot.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

ResourcePolicy can be imported using any of these accepted formats:

```
$ terraform import google_compute_resource_policy.default projects/{{project}}/regions/{{reg
$ terraform import google_compute_resource_policy.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_resource_policy.default {{region}}/{{name}}
$ terraform import google_compute_resource_policy.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_route

Represents a Route resource.

A route is a rule that specifies how certain packets should be handled by the virtual network. Routes are associated with virtual machines by tag, and the set of routes for a particular virtual machine is called its routing table. For each packet leaving a virtual machine, the system searches that virtual machine's routing table for a single best matching route.

Routes match packets by destination IP address, preferring smaller or more specific ranges over larger ones. If there is a tie, the system selects the route with the smallest priority value. If there is still a tie, it uses the layer three and four packet headers to select just one of the remaining matching routes. The packet is then forwarded as specified by the next_hop field of the winning route -- either to another virtual machine destination, a virtual machine gateway or a Compute Engine-operated gateway. Packets that do not match any route in the sending virtual machine's routing table will be dropped.

A Route resource must have exactly one specification of either nextHopGateway, nextHopInstance, nextHopIp, nextHopVpnTunnel, or nextHopIlb.

To get more information about Route, see:

- API documentation
- How-to Guides
  - Using Routes

» Example Usage - Route Basic

```
resource "google_compute_route" "default" {
  name        = "network-route"
  dest_range  = "15.0.0.0/24"
  network     = google_compute_network.default.name
  next_hop_ip = "10.132.1.5"
  priority    = 100
}

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

» Example Usage - Route Ilb

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

resource "google_compute_subnetwork" "default" {
  name          = "compute-subnet"
  ip_cidr_range = "10.0.1.0/24"
  region        = "us-central1"
  network       = google_compute_network.default.self_link
}
```

```
resource "google_compute_health_check" "hc" {
  name               = "proxy-health-check"
  check_interval_sec = 1
  timeout_sec        = 1

  tcp_health_check {
    port = "80"
  }
}

resource "google_compute_region_backend_service" "backend" {
  name          = "compute-backend"
  region        = "us-central1"
  health_checks = [google_compute_health_check.hc.self_link]
}

resource "google_compute_forwarding_rule" "default" {
  name     = "compute-forwarding-rule"
  region   = "us-central1"

  load_balancing_scheme = "INTERNAL"
  backend_service       = google_compute_region_backend_service.backend.self_link
  all_ports             = true
  network               = google_compute_network.default.name
  subnetwork            = google_compute_subnetwork.default.name
}

resource "google_compute_route" "route-ilb" {
  name         = "route-ilb"
  dest_range   = "0.0.0.0/0"
  network      = google_compute_network.default.name
  next_hop_ilb = google_compute_forwarding_rule.default.self_link
  priority     = 2000
}
```

» Argument Reference

The following arguments are supported:

- `dest_range` - (Required) The destination range of outgoing packets that this route applies to. Only IPv4 is supported.
- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which

means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `network` - (Required) The network that this route applies to.

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `priority` - (Optional) The priority of this route. Priority is used to break ties in cases where there is more than one matching route of equal prefix length. In the case of two routes with equal prefix length, the one with the lowest-numbered priority value wins. Default value is 1000. Valid range is 0 through 65535.
- `tags` - (Optional) A list of instance tags to which this route applies.
- `next_hop_gateway` - (Optional) URL to a gateway that should handle matching packets. Currently, you can only specify the internet gateway, using a full or partial valid URL:
  - `https://www.googleapis.com/compute/v1/projects/project/global/gateways/default-inte`
  - `projects/project/global/gateways/default-internet-gateway`
  - `global/gateways/default-internet-gateway`
  - The string `default-internet-gateway`.
- `next_hop_instance` - (Optional) URL to an instance that should handle matching packets. You can specify this as a full or partial URL. For example:
  - `https://www.googleapis.com/compute/v1/projects/project/zones/zone/instances/instanc`
  - `projects/project/zones/zone/instances/instance`
  - `zones/zone/instances/instance`
  - Just the instance name, with the zone in `next_hop_instance_zone`.
- `next_hop_ip` - (Optional) Network IP address of an instance that should handle matching packets.
- `next_hop_vpn_tunnel` - (Optional) URL to a VpnTunnel that should handle matching packets.
- `next_hop_ilb` - (Optional) The URL to a forwarding rule of type loadBalancingScheme=INTERNAL that should handle matching packets. You can only specify the forwarding rule as a partial or full URL. For example, the following are all valid URLs: https://www.googleapis.com/compute/v1/projects/project/regions/region/forwardingRules/forwardingRule regions/region/forwardingRules/forwardingRule Note that this can only be used when the destinationRange is a public (non-RFC 1918) IP CIDR range.

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `next_hop_instance_zone` - (Optional when `next_hop_instance` is specified) The zone of the instance specified in `next_hop_instance`. Omit if `next_hop_instance` is specified as a URL.

» Attributes Reference

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

- `next_hop_network` - URL to a Network that should handle matching packets.
- `self_link` - The URI of the created resource.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

Route can be imported using any of these accepted formats:

```
$ terraform import google_compute_route.default projects/{{project}}/global/routes/{{name}}
$ terraform import google_compute_route.default {{project}}/{{name}}
$ terraform import google_compute_route.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_router

Represents a Router resource.

To get more information about Router, see:

- API documentation
- How-to Guides
  - Google Cloud Router

» Example Usage - Router Basic

```
resource "google_compute_router" "foobar" {
  name    = "my-router"
  network = google_compute_network.foobar.name
  bgp {
    asn               = 64514
    advertise_mode    = "CUSTOM"
    advertised_groups = ["ALL_SUBNETS"]
    advertised_ip_ranges {
      range = "1.2.3.4"
    }
    advertised_ip_ranges {
      range = "6.7.0.0/16"
    }
  }
}

resource "google_compute_network" "foobar" {
  name                    = "my-network"
  auto_create_subnetworks = false
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `network` - (Required) A reference to the network to which this router belongs.

- `description` - (Optional) An optional description of this resource.
- `bgp` - (Optional) BGP information specific to this router. Structure is documented below.
- `region` - (Optional) Region where the router resides.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `bgp` block supports:

- `asn` - (Required) Local BGP Autonomous System Number (ASN). Must be an RFC6996 private ASN, either 16-bit or 32-bit. The value will be fixed for this router resource. All VPN tunnels that link to this router will have the same local ASN.
- `advertise_mode` - (Optional) User-specified flag to indicate which mode to use for advertisement. Valid values of this enum field are: DEFAULT, CUSTOM
- `advertised_groups` - (Optional) User-specified list of prefix groups to advertise in custom mode. This field can only be populated if advertiseMode is CUSTOM and is advertised to all peers of the router. These groups will be advertised in addition to any specified prefixes. Leave this field blank to advertise no custom groups. This enum field has the one valid value: ALL_SUBNETS
- `advertised_ip_ranges` - (Optional) User-specified list of individual IP ranges to advertise in custom mode. This field can only be populated if advertiseMode is CUSTOM and is advertised to all peers of the router. These IP ranges will be advertised in addition to any specified groups. Leave this field blank to advertise no custom IP ranges. Structure is documented below.

The `advertised_ip_ranges` block supports:

- `range` - (Required) The IP range to advertise. The value must be a CIDR-formatted string.
- `description` - (Optional) User-specified description for the IP range.

» 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

Router can be imported using any of these accepted formats:

```
$ terraform import google_compute_router.default projects/{{project}}/regions/{{region}}/rou
$ terraform import google_compute_router.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_router.default {{region}}/{{name}}
$ terraform import google_compute_router.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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.

In addition to the above required fields, a router interface must have specified either `ip_range` or exactly one of `vpn_tunnel` or `interconnect_attachment`, or both.

- `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.
- `vpn_tunnel` - (Optional) The name or resource link to the VPN tunnel this interface will be linked to. Changing this forces a new interface to be created. Only one of `vpn_tunnel` and `interconnect_attachment` can be specified.
- `interconnect_attachment` - (Optional) The name or resource link to the VLAN interconnect for this interface. Changing this forces a new interface to be created. Only one of `vpn_tunnel` and `interconnect_attachment` can be specified.
- `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_nat

A NAT service created in a router.

To get more information about RouterNat, see:

- API documentation
- How-to Guides
  - Google Cloud Router

» Example Usage - Router Nat Basic

```
resource "google_compute_network" "net" {
  name = "my-network"
}

resource "google_compute_subnetwork" "subnet" {
  name          = "my-subnetwork"
  network       = google_compute_network.net.self_link
  ip_cidr_range = "10.0.0.0/16"
  region        = "us-central1"
}

resource "google_compute_router" "router" {
  name    = "my-router"
  region  = google_compute_subnetwork.subnet.region
  network = google_compute_network.net.self_link

  bgp {
    asn = 64514
  }
}

resource "google_compute_router_nat" "nat" {
  name                               = "my-router-nat"
  router                             = google_compute_router.router.name
  region                             = google_compute_router.router.region
  nat_ip_allocate_option             = "AUTO_ONLY"
  source_subnetwork_ip_ranges_to_nat = "ALL_SUBNETWORKS_ALL_IP_RANGES"

  log_config {
    enable = true
    filter = "ERRORS_ONLY"
  }
}
```

» Example Usage - Router Nat Manual Ips

```
resource "google_compute_network" "net" {
  name = "my-network"
}

resource "google_compute_subnetwork" "subnet" {
  name          = "my-subnetwork"
  network       = google_compute_network.net.self_link
  ip_cidr_range = "10.0.0.0/16"
  region        = "us-central1"
}

resource "google_compute_router" "router" {
  name    = "my-router"
  region  = google_compute_subnetwork.subnet.region
  network = google_compute_network.net.self_link
}

resource "google_compute_address" "address" {
  count  = 2
  name   = "nat-manual-ip-${count.index}"
  region = google_compute_subnetwork.subnet.region
}

resource "google_compute_router_nat" "nat_manual" {
  name   = "my-router-nat"
  router = google_compute_router.router.name
  region = google_compute_router.router.region

  nat_ip_allocate_option = "MANUAL_ONLY"
  nat_ips                = google_compute_address.address.*.self_link

  source_subnetwork_ip_ranges_to_nat = "LIST_OF_SUBNETWORKS"
  subnetwork {
    name                    = google_compute_subnetwork.default.self_link
    source_ip_ranges_to_nat = ["ALL_IP_RANGES"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the NAT service. The name must be 1-63

characters long and comply with RFC1035.

- `nat_ip_allocate_option` - (Required) How external IPs should be allocated for this NAT. Valid values are `AUTO_ONLY` for only allowing NAT IPs allocated by Google Cloud Platform, or `MANUAL_ONLY` for only user-allocated NAT IP addresses.
- `source_subnetwork_ip_ranges_to_nat` - (Required) How NAT should be configured per Subnetwork. If `ALL_SUBNETWORKS_ALL_IP_RANGES`, all of the IP ranges in every Subnetwork are allowed to Nat. If `ALL_SUBNETWORKS_ALL_PRIMARY_IP_RANGES`, all of the primary IP ranges in every Subnetwork are allowed to Nat. `LIST_OF_SUBNETWORKS`: A list of Subnetworks are allowed to Nat (specified in the field subnetwork below). Note that if this field contains ALL_SUBNETWORKS_ALL_IP_RANGES or ALL_SUBNETWORKS_ALL_PRIMARY_IP_RANGES, then there should not be any other RouterNat section in any Router for this network in this region.
- `router` - (Required) The name of the Cloud Router in which this NAT will be configured.

- `nat_ips` - (Optional) Self-links of NAT IPs. Only valid if natIpAllocateOption is set to MANUAL_ONLY.
- `drain_nat_ips` - (Optional, Beta) A list of URLs of the IP resources to be drained. These IPs must be valid static external IPs that have been assigned to the NAT.
- `subnetwork` - (Optional) One or more subnetwork NAT configurations. Only used if `source_subnetwork_ip_ranges_to_nat` is set to `LIST_OF_SUBNETWORKS` Structure is documented below.
- `min_ports_per_vm` - (Optional) Minimum number of ports allocated to a VM from this NAT.
- `udp_idle_timeout_sec` - (Optional) Timeout (in seconds) for UDP connections. Defaults to 30s if not set.
- `icmp_idle_timeout_sec` - (Optional) Timeout (in seconds) for ICMP connections. Defaults to 30s if not set.
- `tcp_established_idle_timeout_sec` - (Optional) Timeout (in seconds) for TCP established connections. Defaults to 1200s if not set.
- `tcp_transitory_idle_timeout_sec` - (Optional) Timeout (in seconds) for TCP transitory connections. Defaults to 30s if not set.
- `log_config` - (Optional) Configuration for logging on NAT Structure is documented below.
- `region` - (Optional) Region where the router and NAT reside.

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

The `subnetwork` block supports:

- `name` - (Required) Self-link of subnetwork to NAT
- `source_ip_ranges_to_nat` - (Required) List of options for which source IPs in the subnetwork should have NAT enabled. Supported values include: `ALL_IP_RANGES`, `LIST_OF_SECONDARY_IP_RANGES`, `PRIMARY_IP_RANGE`.
- `secondary_ip_range_names` - (Optional) List of the secondary ranges of the subnetwork that are allowed to use NAT. This can be populated only if `LIST_OF_SECONDARY_IP_RANGES` is one of the values in sourceIpRangesToNat

The `log_config` block supports:

- `enable` - (Required) Indicates whether or not to export logs.
- `filter` - (Required) Specifies the desired filtering of logs on this NAT. Valid values are: `"ERRORS_ONLY"`, `"TRANSLATIONS_ONLY"`, `"ALL"`

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 10 minutes.
- `update` - Default is 10 minutes.
- `delete` - Default is 10 minutes.

» Import

RouterNat can be imported using any of these accepted formats:

```
$ terraform import google_compute_router_nat.default projects/{{project}}/regions/{{region}}
$ terraform import google_compute_router_nat.default {{project}}/{{region}}/{{router}}/{{nam
$ terraform import google_compute_router_nat.default {{region}}/{{router}}/{{name}}
$ terraform import google_compute_router_nat.default {{router}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_router_peer

BGP information that must be configured into the routing stack to establish BGP peering. This information must specify the peer ASN and either the interface name, IP address, or peer IP address. Please refer to RFC4273.

To get more information about RouterBgpPeer, see:

- API documentation
- How-to Guides
  - Google Cloud Router

» Example Usage - Router Peer Basic

```
resource "google_compute_router_peer" "peer" {
  name                      = "my-router-peer"
  router                    = "my-router"
  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) Name of this BGP peer. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `interface` - (Required) Name of the interface the BGP peer is associated with.
- `peer_ip_address` - (Required) IP address of the BGP interface outside Google Cloud Platform. Only IPv4 is supported.
- `peer_asn` - (Required) Peer BGP Autonomous System Number (ASN). Each BGP interface may use a different value.
- `router` - (Required) The name of the Cloud Router in which this BgpPeer will be configured.

- `advertised_route_priority` - (Optional) The priority of routes advertised to this BGP peer. Where there is more than one matching route of maximum length, the routes with the lowest priority value win.
- `advertise_mode` - (Optional) User-specified flag to indicate which mode to use for advertisement. Valid values of this enum field are: `DEFAULT`, `CUSTOM`
- `advertised_groups` - (Optional) User-specified list of prefix groups to advertise in custom mode, which can take one of the following options:
  - `ALL_SUBNETS`: Advertises all available subnets, including peer VPC subnets.
  - `ALL_VPC_SUBNETS`: Advertises the router's own VPC subnets.
  - `ALL_PEER_VPC_SUBNETS`: Advertises peer subnets of the router's VPC network.

Note that this field can only be populated if advertiseMode is `CUSTOM` and overrides the list defined for the router (in the "bgp" message). These groups are advertised in addition to any specified prefixes. Leave this field blank to advertise no custom groups.

- `advertised_ip_ranges` - (Optional) User-specified list of individual IP ranges to advertise in custom mode. This field can only be populated if advertiseMode is `CUSTOM` and is advertised to all peers of the router. These IP ranges will be advertised in addition to any specified groups. Leave this field blank to advertise no custom IP ranges. Structure is documented below.
- `region` - (Optional) Region where the router and BgpPeer reside. If it is not provided, the provider region is used.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `advertised_ip_ranges` block supports:

- `range` - (Required) The IP range to advertise. The value must be a CIDR-formatted string.
- `description` - (Optional) User-specified description for the IP range.

» 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. Only IPv4 is supported.

- `management_type` - The resource that configures and manages this BGP peer.
  - `MANAGED_BY_USER` is the default value and can be managed by you or other users
  - `MANAGED_BY_ATTACHMENT` is a BGP peer that is configured and managed by Cloud Interconnect, specifically by an InterconnectAttachment of type PARTNER. Google automatically creates, updates, and deletes this type of BGP peer when the PARTNER InterconnectAttachment is created, updated, or deleted.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 10 minutes.
- `update` - Default is 10 minutes.
- `delete` - Default is 10 minutes.

» Import

RouterBgpPeer can be imported using any of these accepted formats:

```
$ terraform import google_compute_router_peer.default projects/{{project}}/regions/{{region}
$ terraform import google_compute_router_peer.default {{project}}/{{region}}/{{router}}/{{na
$ terraform import google_compute_router_peer.default {{region}}/{{router}}/{{name}}
$ terraform import google_compute_router_peer.default {{router}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_security_policy

A Security Policy defines an IP blacklist or whitelist that protects load balanced Google Cloud services by denying or permitting traffic from specified IP ranges. For more information see the official documentation and the API.

Security Policy is used by `google_compute_backend_service`.

» Example Usage

```
resource "google_compute_security_policy" "policy" {
  name = "my-policy"

  rule {
    action   = "deny(403)"
    priority = "1000"
    match {
      versioned_expr = "SRC_IPS_V1"
      config {
        src_ip_ranges = ["9.9.9.0/24"]
      }
    }
    description = "Deny access to IPs in 9.9.9.0/24"
  }

  rule {
    action   = "allow"
    priority = "2147483647"
    match {
      versioned_expr = "SRC_IPS_V1"
      config {
        src_ip_ranges = ["*"]
      }
    }
    description = "default rule"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the security policy.

- `description` - (Optional) An optional description of this security policy. Max size is 2048.
- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `rule` - (Optional) The set of rules that belong to this policy. There must always be a default rule (rule with priority 2147483647 and match "*"). If

no rules are provided when creating a security policy, a default rule with action "allow" will be added. Structure is documented below.

The `rule` block supports:

- `action` - (Required) Action to take when `match` matches the request. Valid values:
  - "allow" : allow access to target
  - "deny(status)" : deny access to target, returns the HTTP response code specified (valid values are 403, 404 and 502)
- `priority` - (Required) An unique positive integer indicating the priority of evaluation for a rule. Rules are evaluated from highest priority (lowest numerically) to lowest priority (highest numerically) in order.
- `match` - (Required) A match condition that incoming traffic is evaluated against. If it evaluates to true, the corresponding `action` is enforced. Structure is documented below.
- `description` - (Optional) An optional description of this rule. Max size is 64.
- `preview` - (Optional) When set to true, the `action` specified above is not enforced. Stackdriver logs for requests that trigger a preview action are annotated as such.

The `match` block supports:

- `config` - (Optional) The configuration options available when specifying `versioned_expr`. This field must be specified if `versioned_expr` is specified and cannot be specified if `versioned_expr` is not specified. Structure is documented below.
- `versioned_expr` - (Optional) Predefined rule expression. If this field is specified, `config` must also be specified. Available options:
  - SRC_IPS_V1: Must specify the corresponding `src_ip_ranges` field in `config`.
- `expr` - (Optional) User defined CEVAL expression. A CEVAL expression is used to specify match criteria such as origin.ip, source.region_code and contents in the request header. Structure is documented below.

The `config` block supports:

- `src_ip_ranges` - (Required) Set of IP addresses or ranges (IPV4 or IPV6) in CIDR notation to match against inbound traffic. There is a limit of 5 IP ranges per rule. A value of '*' matches all IPs (can be used to override the default behavior).

The `expr` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax. The application context of the containing message determines which well-known feature set of CEL is supported.

» Attributes Reference

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

- `fingerprint` - Fingerprint of this resource.
- `self_link` - The URI of the created resource.

» Import

Security policies can be imported using any of the following formats

```
$ terraform import google_compute_security_policy.policy projects/{{project}}/global/securit
$ terraform import google_compute_security_policy.policy {{project}}/{{name}}
$ terraform import google_compute_security_policy.policy {{name}}
```

» 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

Represents a Persistent Disk Snapshot resource.

Use snapshots to back up data from your persistent disks. Snapshots are different from public images and custom images, which are used primarily to create instances or configure instance templates. Snapshots are useful for periodic backup of the data on your persistent disks. You can create snapshots from persistent disks even while they are attached to running instances.

Snapshots are incremental, so you can create regular snapshots on a persistent disk faster and at a much lower cost than if you regularly created a full image of the disk.

To get more information about Snapshot, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Snapshot Basic

```
resource "google_compute_snapshot" "snapshot" {
  name        = "my-snapshot"
  source_disk = google_compute_disk.persistent.name
  zone        = "us-central1-a"
  labels = {
    my_label = "value"
  }
}
```

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

resource "google_compute_disk" "persistent" {
  name  = "debian-disk"
  image = data.google_compute_image.debian.self_link
  size  = 10
  type  = "pd-ssd"
  zone  = "us-central1-a"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource; provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `source_disk` - (Required) A reference to the disk used to create this snapshot.

- `description` - (Optional) An optional description of this resource.
- `labels` - (Optional) Labels to apply to this Snapshot.
- `zone` - (Optional) A reference to the zone where the disk is hosted.
- `snapshot_encryption_key` - (Optional) The customer-supplied encryption key of the snapshot. Required if the source snapshot is protected by a customer-supplied encryption key. Structure is documented below.
- `source_disk_encryption_key` - (Optional) The customer-supplied encryption key of the source snapshot. Required if the source snapshot is protected by a customer-supplied encryption key. 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.

The `snapshot_encryption_key` block supports:

- `raw_key` - (Required) Specifies a 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to either encrypt or decrypt this resource.
- `sha256` - The RFC 4648 base64 encoded SHA-256 hash of the customer-supplied encryption key that protects this resource.

The `source_disk_encryption_key` block supports:

- `raw_key` - (Optional) Specifies a 256-bit customer-supplied encryption key, encoded in RFC 4648 base64 to either encrypt or decrypt this resource.

» Attributes Reference

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

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `snapshot_id` - The unique identifier for the resource.
- `disk_size_gb` - Size of the snapshot, specified in GB.
- `storage_bytes` - A size of the the storage used by the snapshot. As snapshots share storage, this number is expected to change with snapshot creation/deletion.
- `licenses` - A list of public visible licenses that apply to this snapshot. This can be because the original image had licenses attached (such as a Windows image). snapshotEncryptionKey nested object Encrypts the snapshot using a customer-supplied encryption key.
- `label_fingerprint` - The fingerprint used for optimistic locking of this resource. Used internally during updates.
- `self_link` - The URI of the created resource.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 5 minutes.
- `update` - Default is 5 minutes.
- `delete` - Default is 5 minutes.

» Import

Snapshot can be imported using any of these accepted formats:

```
$ terraform import google_compute_snapshot.default projects/{{project}}/global/snapshots/{{
$ terraform import google_compute_snapshot.default {{project}}/{{name}}
$ terraform import google_compute_snapshot.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_managed_ssl_certificate

An SslCertificate resource, used for HTTPS load balancing. This resource represents a certificate for which the certificate secrets are created and managed by Google.

For a resource where you provide the key, see the SSL Certificate resource.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about ManagedSslCertificate, see:

- API documentation
- How-to Guides
  - Official Documentation

Warning: This resource should be used with extreme caution! Provisioning an SSL certificate is complex. Ensure that you understand the lifecycle of a certificate before attempting complex tasks like cert rotation automatically. This resource will "return" as soon as the certificate object is created, but post-creation the certificate object will go through a "provisioning" process. The provisioning process can complete only when the domain name for which the certificate is created points to a target pool which, itself, points at the certificate. Depending on your DNS provider, this may take some time, and migrating from self-managed certificates to Google-managed certificates may entail some downtime while the certificate provisions.

In conclusion: Be extremely cautious.

» Example Usage - Managed Ssl Certificate Basic

```
resource "google_compute_managed_ssl_certificate" "default" {
  provider = google-beta

  name = "test-cert"

  managed {
    domains = ["sslcert.tf-test.club."]
  }
}

resource "google_compute_target_https_proxy" "default" {
  provider = google-beta

  name             = "test-proxy"
  url_map          = google_compute_url_map.default.self_link
  ssl_certificates = [google_compute_managed_ssl_certificate.default.self_link]
}

resource "google_compute_url_map" "default" {
  provider = google-beta

  name        = "url-map"
  description = "a description"

  default_service = google_compute_backend_service.default.self_link

  host_rule {
    hosts        = ["sslcert.tf-test.club"]
    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" {
  provider = google-beta

  name        = "backend-service"
  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" {
  provider = google-beta

  name               = "http-health-check"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}

resource "google_dns_managed_zone" "zone" {
  provider = google-beta

  name     = "dnszone"
  dns_name = "sslcert.tf-test.club."
}

resource "google_compute_global_forwarding_rule" "default" {
  provider = google-beta

  name       = "forwarding-rule"
  target     = google_compute_target_https_proxy.default.self_link
  port_range = 443
}

resource "google_dns_record_set" "set" {
  provider = google-beta

  name         = "sslcert.tf-test.club."
  type         = "A"
  ttl          = 3600
```

```
  managed_zone = google_dns_managed_zone.zone.name
  rrdatas      = [google_compute_global_forwarding_rule.default.ip_address]
}

provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}
```

» Argument Reference

The following arguments are supported:

- `description` - (Optional) An optional description of this resource.
- `name` - (Optional) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

These are in the same namespace as the managed SSL certificates.

- `managed` - (Optional) Properties relevant to a managed certificate. These will be used if the certificate is managed (as indicated by a value of `MANAGED` in `type`). Structure is documented below.
- `type` - (Optional) Enum field whose value is always `MANAGED` - used to signal to the API which type this is.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `managed` block supports:

- `domains` - (Required) Domains for which a managed SSL certificate will be valid. Currently, there can be up to 100 domains in this list.

» Attributes Reference

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

- `creation_timestamp` - Creation timestamp in RFC3339 text format.

- `certificate_id` - The unique identifier for the resource.
- `subject_alternative_names` - Domains associated with the certificate via Subject Alternative Name.
- `expire_time` - Expire time of the certificate.
- `self_link` - The URI of the created resource.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 6 minutes.
- `delete` - Default is 30 minutes.

» Import

ManagedSslCertificate can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_compute_managed_ssl_certificate.default pro
$ terraform import -provider=google-beta google_compute_managed_ssl_certificate.default {{p
$ terraform import -provider=google-beta google_compute_managed_ssl_certificate.default {{n
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_ssl_certificate

An SslCertificate resource, used for HTTPS load balancing. This resource provides a mechanism to upload an SSL key and certificate to the load balancer to serve secure connections from the user.

To get more information about SslCertificate, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Ssl Certificate Basic

```
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
  }
}
```

» Example Usage - Ssl Certificate Random Provider

```
# You may also want to control name generation explicitly:
resource "google_compute_ssl_certificate" "default" {
  # The name will contain 8 random hex digits,
  # e.g. "my-certificate-48ab27cd2a"
  name        = random_id.certificate.hex
  private_key = file("path/to/private.key")
  certificate = file("path/to/certificate.crt")

  lifecycle {
    create_before_destroy = true
  }
}

resource "random_id" "certificate" {
  byte_length = 4
  prefix      = "my-certificate-"

  # For security, do not expose raw certificate values in the output
```

```
  keepers = {
    private_key = filebase64sha256("path/to/private.key")
    certificate = filebase64sha256("path/to/certificate.crt")
  }
}
```

» Example Usage - Ssl Certificate Target Https Proxies

```
// 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, specify a partial
// name with name_prefix, or use random_id resource. Example:

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

  lifecycle {
    create_before_destroy = true
  }
}

resource "google_compute_target_https_proxy" "default" {
  name             = "test-proxy"
  url_map          = google_compute_url_map.default.self_link
  ssl_certificates = [google_compute_ssl_certificate.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        = "backend-service"
  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               = "http-health-check"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

» Argument Reference

The following arguments are supported:

- `certificate` - (Required) The certificate in PEM format. The certificate chain must be no greater than 5 certs long. The chain must include at least one intermediate cert.
- `private_key` - (Required) The write-only private key in PEM format.

- `description` - (Optional) An optional description of this resource.
- `name` - (Optional) Name of the resource. Provided by the client when the

resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

These are in the same namespace as the managed SSL certificates.

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `name_prefix` - (Optional) Creates a unique name beginning with the specified prefix. Conflicts with `name`.

» Attributes Reference

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

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

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

SslCertificate can be imported using any of these accepted formats:

```
$ terraform import google_compute_ssl_certificate.default projects/{{project}}/global/sslCer
$ terraform import google_compute_ssl_certificate.default {{project}}/{{name}}
$ terraform import google_compute_ssl_certificate.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_ssl_policy

Represents a SSL policy. SSL policies give you the ability to control the features of SSL that your SSL proxy or HTTPS load balancer negotiates.

To get more information about SslPolicy, see:

- API documentation
- How-to Guides
  - Using SSL Policies

» Example Usage - Ssl Policy Basic

```
resource "google_compute_ssl_policy" "prod-ssl-policy" {
  name    = "production-ssl-policy"
  profile = "MODERN"
}

resource "google_compute_ssl_policy" "nonprod-ssl-policy" {
  name            = "nonprod-ssl-policy"
  profile         = "MODERN"
  min_tls_version = "TLS_1_2"
}

resource "google_compute_ssl_policy" "custom-ssl-policy" {
  name            = "custom-ssl-policy"
  min_tls_version = "TLS_1_2"
  profile         = "CUSTOM"
  custom_features = ["TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384", "TLS_ECDHE_RSA_WITH_AES_256_
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `description` - (Optional) An optional description of this resource.
- `profile` - (Optional) Profile specifies the set of SSL features that can be used by the load balancer when negotiating SSL with clients. This can be one of `COMPATIBLE`, `MODERN`, `RESTRICTED`, or `CUSTOM`. If using `CUSTOM`, the set of SSL features to enable must be specified in the `customFeatures` field. See the official documentation for information on what cipher suites each profile provides. If `CUSTOM` is used, the `custom_features` attribute **must be set**. Default is `COMPATIBLE`.
- `min_tls_version` - (Optional) The minimum version of SSL protocol that can be used by the clients to establish a connection with the load balancer. This can be one of `TLS_1_0`, `TLS_1_1`, `TLS_1_2`. Default is `TLS_1_0`.
- `custom_features` - (Optional) Profile specifies the set of SSL features that can be used by the load balancer when negotiating SSL with clients. This can be one of `COMPATIBLE`, `MODERN`, `RESTRICTED`, or `CUSTOM`. If using `CUSTOM`, the set of SSL features to enable must be specified in the `customFeatures` field. See the official documentation for which ciphers are available to use. **Note**: this argument *must* be present when using the `CUSTOM` profile. This argument *must not* be present when using any other profile.
- `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.
- `enabled_features` - The list of features enabled in the SSL policy.
- `fingerprint` - Fingerprint of this resource. A hash of the contents stored in this object. This field is used in optimistic locking.
- `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

SslPolicy can be imported using any of these accepted formats:

```
$ terraform import google_compute_ssl_policy.default projects/{{project}}/global/sslPolicies
$ terraform import google_compute_ssl_policy.default {{project}}/{{name}}
$ terraform import google_compute_ssl_policy.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_subnetwork

A VPC network is a virtual version of the traditional physical networks that exist within and between physical data centers. A VPC network provides connectivity for your Compute Engine virtual machine (VM) instances, Container Engine containers, App Engine Flex services, and other network-related resources.

Each GCP project contains one or more VPC networks. Each VPC network is a global entity spanning all GCP regions. This global VPC network allows VM instances and other resources to communicate with each other via internal, private IP addresses.

Each VPC network is subdivided into subnets, and each subnet is contained within a single region. You can have more than one subnet in a region for a given VPC network. Each subnet has a contiguous private RFC1918 IP space. You create instances, containers, and the like in these subnets. When you create an instance, you must create it in a subnet, and the instance draws its internal IP address from that subnet.

Virtual machine (VM) instances in a VPC network can communicate with instances in all other subnets of the same VPC network, regardless of region, using

their RFC1918 private IP addresses. You can isolate portions of the network, even entire subnets, using firewall rules.

To get more information about Subnetwork, see:

- API documentation
- How-to Guides
  - Private Google Access
  - Cloud Networking

» Example Usage - Subnetwork Basic

```
resource "google_compute_subnetwork" "network-with-private-secondary-ip-ranges" {
  name          = "test-subnetwork"
  ip_cidr_range = "10.2.0.0/16"
  region        = "us-central1"
  network       = google_compute_network.custom-test.self_link
  secondary_ip_range {
    range_name    = "tf-test-secondary-range-update1"
    ip_cidr_range = "192.168.10.0/24"
  }
}

resource "google_compute_network" "custom-test" {
  name                    = "test-network"
  auto_create_subnetworks = false
}
```

» Example Usage - Subnetwork Logging Config

```
resource "google_compute_subnetwork" "subnet-with-logging" {
  name          = "log-test-subnetwork"
  ip_cidr_range = "10.2.0.0/16"
  region        = "us-central1"
```

```
  network       = google_compute_network.custom-test.self_link

  log_config {
    aggregation_interval = "INTERVAL_10_MIN"
    flow_sampling        = 0.5
    metadata             = "INCLUDE_ALL_METADATA"
  }
}

resource "google_compute_network" "custom-test" {
  name                    = "log-test-network"
  auto_create_subnetworks = false
}
```

» Example Usage - Subnetwork Internal L7lb

```
resource "google_compute_subnetwork" "network-for-l7lb" {
  provider = google-beta

  name          = "l7lb-test-subnetwork"
  ip_cidr_range = "10.0.0.0/22"
  region        = "us-central1"
  purpose       = "INTERNAL_HTTPS_LOAD_BALANCER"
  role          = "ACTIVE"
  network       = google_compute_network.custom-test.self_link
}

resource "google_compute_network" "custom-test" {
  provider = google-beta

  name                    = "l7lb-test-network"
  auto_create_subnetworks = false
}
```

» Argument Reference

The following arguments are supported:

- `ip_cidr_range` - (Required) The range of internal addresses that are owned by this subnetwork. Provide this property when you create the subnetwork. For example, 10.0.0.0/8 or 192.168.0.0/16. Ranges must be unique and non-overlapping within a network. Only IPv4 is supported.
- `name` - (Required) The name of the resource, provided by the client when initially creating the resource. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `network` - (Required) The network this subnet belongs to. Only networks that are in the distributed mode can have subnetworks.

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource. This field can be set only at resource creation time.
- `purpose` - (Optional, Beta) The purpose of the resource. This field can be either PRIVATE or INTERNAL_HTTPS_LOAD_BALANCER. A subnetwork with purpose set to INTERNAL_HTTPS_LOAD_BALANCER is a user-created subnetwork that is reserved for Internal HTTP(S) Load Balancing. If unspecified, the purpose defaults to PRIVATE. If set to INTERNAL_HTTPS_LOAD_BALANCER you must also set the role.
- `role` - (Optional, Beta) The role of subnetwork. Currently, this field is only used when purpose = INTERNAL_HTTPS_LOAD_BALANCER. The value can be set to ACTIVE or BACKUP. An ACTIVE subnetwork is one that is currently being used for Internal HTTP(S) Load Balancing. A BACKUP subnetwork is one that is ready to be promoted to ACTIVE or is currently draining.
- `secondary_ip_range` - (Optional) An array of configurations for secondary IP ranges for VM instances contained in this subnetwork. The primary IP of such VM must belong to the primary ipCidrRange of the subnetwork. The alias IPs may belong to either primary or secondary ranges. This field uses attr-as-block mode to avoid breaking users during the 0.12 upgrade. See the Attr-as-Block page for more details. Structure is documented below.
- `private_ip_google_access` - (Optional) When enabled, VMs in this subnetwork without external IP addresses can access Google APIs and services by using Private Google Access.
- `region` - (Optional) URL of the GCP region for this subnetwork.

- `log_config` - (Optional) Denotes the logging options for the subnetwork flow logs. If logging is enabled logs will be exported to Stackdriver. This field cannot be set if the `purpose` of this subnetwork is `INTERNAL_HTTPS_LOAD_BALANCER` 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.

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. The name must be 1-63 characters long, and comply with RFC1035. The name must be unique within the subnetwork.
- `ip_cidr_range` - (Required) The range of IP addresses belonging to this subnetwork secondary range. Provide this property when you create the subnetwork. Ranges must be unique and non-overlapping with all primary and secondary IP ranges within a network. Only IPv4 is supported.

The `log_config` block supports:

- `aggregation_interval` - (Optional) Can only be specified if VPC flow logging for this subnetwork is enabled. Toggles the aggregation interval for collecting flow logs. Increasing the interval time will reduce the amount of generated flow logs for long lasting connections. Default is an interval of 5 seconds per connection. Possible values are INTERVAL_5_SEC, INTERVAL_30_SEC, INTERVAL_1_MIN, INTERVAL_5_MIN, INTERVAL_10_MIN, INTERVAL_15_MIN
- `flow_sampling` - (Optional) Can only be specified if VPC flow logging for this subnetwork is enabled. The value of the field must be in [0, 1]. Set the sampling rate of VPC flow logs within the subnetwork where 1.0 means all collected logs are reported and 0.0 means no logs are reported. Default is 0.5 which means half of all collected logs are reported.
- `metadata` - (Optional) Can only be specified if VPC flow logging for this subnetwork is enabled. Configures whether metadata fields should be added to the reported VPC flow logs. Default is `INCLUDE_ALL_METADATA`.

» Attributes Reference

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

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `gateway_address` - The gateway address for default routes to reach destination addresses outside this subnetwork.

- `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 any of these accepted formats:

```
$ terraform import google_compute_subnetwork.default projects/{{project}}/regions/{{region}}
$ terraform import google_compute_subnetwork.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_subnetwork.default {{region}}/{{name}}
$ terraform import google_compute_subnetwork.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for ComputeSubnetwork

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

- `google_compute_subnetwork_iam_policy`: Authoritative. Sets the IAM policy for the subnetwork and replaces any existing policy already attached.
- `google_compute_subnetwork_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 subnetwork are preserved.
- `google_compute_subnetwork_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the subnetwork are preserved.

Note: `google_compute_subnetwork_iam_policy` **cannot** be used in conjunction with `google_compute_subnetwork_iam_binding` and `google_compute_subnetwork_iam_member` or they will fight over what your policy should be.

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

» google_compute_subnetwork_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/compute.networkUser"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_compute_subnetwork_iam_policy" "editor" {
  project = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.project}"
  region = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.region}"
  subnetwork = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_compute_subnetwork_iam_binding

```
resource "google_compute_subnetwork_iam_binding" "editor" {
  project = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.project}"
  region = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.region}"
  subnetwork = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.name}"
  role = "roles/compute.networkUser"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_compute_subnetwork_iam_member

```
resource "google_compute_subnetwork_iam_member" "editor" {
  project = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.project}"
  region = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.region}"
```

```
  subnetwork = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.name}"
  role = "roles/compute.networkUser"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `subnetwork` - (Required) Used to find the parent resource to bind the IAM policy to
- `region` - (Optional) URL of the GCP region for this subnetwork. Used to find the parent resource to bind the IAM policy to. If not specified, the value will be parsed from the identifier of the parent resource. If no region is provided in the parent identifier and no region is specified, it is taken from the provider configuration.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_compute_subnetwork_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_compute_subnetwork_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/regions/{{region}}/subnetworks/{{name}}
- {{project}}/{{region}}/{{name}}
- {{region}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Compute subnetwork IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_compute_subnetwork_iam_member.editor "projects/{{project}}/regions/{{region}}/subnetw roles/compute.networkUser jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_compute_subnetwork_iam_binding.editor "projects/{{project}}/regions/{{region}}/subnetworks/{{subnetwork}} roles/compute.networkUser"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_compute_subnetwork_iam_policy.editor projects/{{project}}/regions/{{region}}/subnetworks/{{subnetwork}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for ComputeSubnetwork

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

- `google_compute_subnetwork_iam_policy`: Authoritative. Sets the IAM policy for the subnetwork and replaces any existing policy already attached.
- `google_compute_subnetwork_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 subnetwork are preserved.
- `google_compute_subnetwork_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the subnetwork are preserved.

Note: `google_compute_subnetwork_iam_policy` **cannot** be used in conjunction with `google_compute_subnetwork_iam_binding` and `google_compute_subnetwork_iam_member` or they will fight over what your policy should be.

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

» google_compute_subnetwork_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/compute.networkUser"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_compute_subnetwork_iam_policy" "editor" {
  project = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.project}"
  region = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.region}"
  subnetwork = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_compute_subnetwork_iam_binding

```
resource "google_compute_subnetwork_iam_binding" "editor" {
  project = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.project}"
```

```
  region = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.region}"
  subnetwork = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.name}"
  role = "roles/compute.networkUser"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_compute_subnetwork_iam_member

```
resource "google_compute_subnetwork_iam_member" "editor" {
  project = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.project}"
  region = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.region}"
  subnetwork = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.name}"
  role = "roles/compute.networkUser"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `subnetwork` - (Required) Used to find the parent resource to bind the IAM policy to
- `region` - (Optional) URL of the GCP region for this subnetwork. Used to find the parent resource to bind the IAM policy to. If not specified, the value will be parsed from the identifier of the parent resource. If no region is provided in the parent identifier and no region is specified, it is taken from the provider configuration.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_compute_subnetwork_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_compute_subnetwork_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/regions/{{region}}/subnetworks/{{name}}
- {{project}}/{{region}}/{{name}}
- {{region}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Compute subnetwork IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_compute_subnetwork_iam_member.editor "projects/{{project}}/regions/{{region}}/subnetw roles/compute.networkUser jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_compute_subnetwork_iam_binding.editor "projects/{{project}}/regions/{{region}}/subnetworks/{{subnetwork}} roles/compute.networkUser"`

IAM policy imports use the identifier of the resource in question, e.g.
`$ terraform import google_compute_subnetwork_iam_policy.editor projects/{{project}}/regions/{{region}}/subnetworks/{{subnetwork}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for ComputeSubnetwork

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

- `google_compute_subnetwork_iam_policy`: Authoritative. Sets the IAM policy for the subnetwork and replaces any existing policy already attached.
- `google_compute_subnetwork_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 subnetwork are preserved.
- `google_compute_subnetwork_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the subnetwork are preserved.

Note: `google_compute_subnetwork_iam_policy` **cannot** be used in conjunction with `google_compute_subnetwork_iam_binding` and `google_compute_subnetwork_iam_member` or they will fight over what your policy should be.

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

» google_compute_subnetwork_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/compute.networkUser"
    members = [
      "user:jane@example.com",
    ]
  }
```

```
}

resource "google_compute_subnetwork_iam_policy" "editor" {
  project = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.project}"
  region = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.region}"
  subnetwork = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_compute_subnetwork_iam_binding

```
resource "google_compute_subnetwork_iam_binding" "editor" {
  project = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.project}"
  region = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.region}"
  subnetwork = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.name}"
  role = "roles/compute.networkUser"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_compute_subnetwork_iam_member

```
resource "google_compute_subnetwork_iam_member" "editor" {
  project = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.project}"
  region = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.region}"
  subnetwork = "${google_compute_subnetwork.network-with-private-secondary-ip-ranges.name}"
  role = "roles/compute.networkUser"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `subnetwork` - (Required) Used to find the parent resource to bind the IAM policy to
- `region` - (Optional) URL of the GCP region for this subnetwork. Used to find the parent resource to bind the IAM policy to. If not specified, the value will be parsed from the identifier of the parent resource. If no region is provided in the parent identifier and no region is specified, it is taken from the provider configuration.

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_compute_subnetwork_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_compute_subnetwork_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/regions/{{region}}/subnetworks/{{name}}
- {{project}}/{{region}}/{{name}}
- {{region}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Compute subnetwork IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_compute_subnetwork_iam_member.editor "projects/{{project}}/regions/{{region}}/subnetw roles/compute.networkUser jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_compute_subnetwork_iam_binding.editor "projects/{{project}}/regions/{{region}}/subnetworks/{{subnetwork}} roles/compute.networkUser"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_compute_subnetwork_iam_policy.editor projects/{{project}}/regions/{{region}}/subnetworks/{{subnetwork}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_target_http_proxy

Represents a TargetHttpProxy resource, which is used by one or more global forwarding rule to route incoming HTTP requests to a URL map.

To get more information about TargetHttpProxy, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Target Http Proxy Basic

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

resource "google_compute_url_map" "default" {
  name            = "url-map"
  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        = "backend-service"
  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               = "http-health-check"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `url_map` - (Required) A reference to the UrlMap resource that defines the mapping from URL to the BackendService.

- `description` - (Optional) An optional 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:

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `proxy_id` - The unique identifier for the resource.
- `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

TargetHttpProxy can be imported using any of these accepted formats:

```
$ terraform import google_compute_target_http_proxy.default projects/{{project}}/global/targ
$ terraform import google_compute_target_http_proxy.default {{project}}/{{name}}
$ terraform import google_compute_target_http_proxy.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_target_https_proxy

Represents a TargetHttpsProxy resource, which is used by one or more global forwarding rule to route incoming HTTPS requests to a URL map.

To get more information about TargetHttpsProxy, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Target Https Proxy Basic

```
resource "google_compute_target_https_proxy" "default" {
  name             = "test-proxy"
  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"
  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        = "backend-service"
  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               = "http-health-check"
  request_path       = "/"
  check_interval_sec = 1
  timeout_sec        = 1
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `ssl_certificates` - (Required) A list of SslCertificate resources that are

used to authenticate connections between users and the load balancer. At least one SSL certificate must be specified.

- `url_map` - (Required) A reference to the UrlMap resource that defines the mapping from URL to the BackendService.

- `description` - (Optional) An optional description of this resource.
- `quic_override` - (Optional) Specifies the QUIC override policy for this resource. This determines whether the load balancer will attempt to negotiate QUIC with clients or not. Can specify one of NONE, ENABLE, or DISABLE. If NONE is specified, uses the QUIC policy with no user overrides, which is equivalent to DISABLE. Not specifying this field is equivalent to specifying NONE.
- `ssl_policy` - (Optional) A reference to the SslPolicy resource that will be associated with the TargetHttpsProxy resource. If not set, the TargetHttpsProxy resource will not have any SSL policy configured.
- `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.
- `proxy_id` - The unique identifier for the resource.
- `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

TargetHttpsProxy can be imported using any of these accepted formats:

```
$ terraform import google_compute_target_https_proxy.default projects/{{project}}/global/ta
$ terraform import google_compute_target_https_proxy.default {{project}}/{{name}}
$ terraform import google_compute_target_https_proxy.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_target_instance

Represents a TargetInstance resource which defines an endpoint instance that terminates traffic of certain protocols. In particular, they are used in Protocol Forwarding, where forwarding rules can send packets to a non-NAT'ed target instance. Each target instance contains a single virtual machine instance that receives and handles traffic from the corresponding forwarding rules.

To get more information about TargetInstance, see:

- API documentation
- How-to Guides
  - Using Protocol Forwarding

» Example Usage - Target Instance Basic

```
resource "google_compute_target_instance" "default" {
  name     = "target"
  instance = google_compute_instance.target-vm.self_link
}

data "google_compute_image" "vmimage" {
  family  = "debian-9"
  project = "debian-cloud"
}
```

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

  boot_disk {
    initialize_params {
      image = data.google_compute_image.vmimage.self_link
    }
  }

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

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `instance` - (Required) The Compute instance VM handling traffic for this target instance. Accepts the instance self-link, relative path (e.g. `projects/project/zones/zone/instances/instance`) or name. If name is given, the zone will default to the given zone or the provider-default zone and the project will default to the provider-level project.

- `description` - (Optional) An optional description of this resource.
- `nat_policy` - (Optional) NAT option controlling how IPs are NAT'ed to the instance. Currently only NO_NAT (default value) is supported.
- `zone` - (Optional) URL of the zone where the target instance resides.
- `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.
- `delete` - Default is 4 minutes.

» Import

TargetInstance can be imported using any of these accepted formats:

```
$ terraform import google_compute_target_instance.default projects/{{project}}/zones/{{zone}
$ terraform import google_compute_target_instance.default {{project}}/{{zone}}/{{name}}
$ terraform import google_compute_target_instance.default {{zone}}/{{name}}
$ terraform import google_compute_target_instance.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_target_ssl_proxy

Represents a TargetSslProxy resource, which is used by one or more global forwarding rule to route incoming SSL requests to a backend service.

To get more information about TargetSslProxy, see:

- API documentation
- How-to Guides
  - Setting Up SSL proxy for Google Cloud Load Balancing

» Example Usage - Target Ssl Proxy Basic

```
resource "google_compute_target_ssl_proxy" "default" {
  name             = "test-proxy"
  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/private.key")
  certificate = file("path/to/certificate.crt")
}

resource "google_compute_backend_service" "default" {
  name          = "backend-service"
  protocol      = "SSL"
  health_checks = [google_compute_health_check.default.self_link]
}

resource "google_compute_health_check" "default" {
  name               = "health-check"
  check_interval_sec = 1
  timeout_sec        = 1
  tcp_health_check {
    port = "443"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following char-

acters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `backend_service` - (Required) A reference to the BackendService resource.
- `ssl_certificates` - (Required) A list of SslCertificate resources that are used to authenticate connections between users and the load balancer. Currently, exactly one SSL certificate must be specified.

- `description` - (Optional) An optional description of this resource.
- `proxy_header` - (Optional) Specifies the type of proxy header to append before sending data to the backend, either NONE or PROXY_V1. The default is NONE.
- `ssl_policy` - (Optional) A reference to the SslPolicy resource that will be associated with the TargetSslProxy resource. If not set, the TargetSslProxy resource will not have any SSL policy configured.
- `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.
- `proxy_id` - The unique identifier for the resource.
- `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

TargetSslProxy can be imported using any of these accepted formats:

```
$ terraform import google_compute_target_ssl_proxy.default projects/{{project}}/global/targe
$ terraform import google_compute_target_ssl_proxy.default {{project}}/{{name}}
$ terraform import google_compute_target_ssl_proxy.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_target_tcp_proxy

Represents a TargetTcpProxy resource, which is used by one or more global forwarding rule to route incoming TCP requests to a Backend service.

To get more information about TargetTcpProxy, see:

- API documentation
- How-to Guides
  - Setting Up TCP proxy for Google Cloud Load Balancing

» Example Usage - Target Tcp Proxy Basic

```
resource "google_compute_target_tcp_proxy" "default" {
  name            = "test-proxy"
  backend_service = google_compute_backend_service.default.self_link
}

resource "google_compute_backend_service" "default" {
  name        = "backend-service"
  protocol    = "TCP"
  timeout_sec = 10

  health_checks = [google_compute_health_check.default.self_link]
}
```

```
resource "google_compute_health_check" "default" {
  name               = "health-check"
  timeout_sec        = 1
  check_interval_sec = 1

  tcp_health_check {
    port = "443"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `backend_service` - (Required) A reference to the BackendService resource.

- `description` - (Optional) An optional description of this resource.
- `proxy_header` - (Optional) Specifies the type of proxy header to append before sending data to the backend, either NONE or PROXY_V1. The default is NONE.
- `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.
- `proxy_id` - The unique identifier for the resource.
- `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

TargetTcpProxy can be imported using any of these accepted formats:

```
$ terraform import google_compute_target_tcp_proxy.default projects/{{project}}/global/targe
$ terraform import google_compute_target_tcp_proxy.default {{project}}/{{name}}
$ terraform import google_compute_target_tcp_proxy.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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 any of the following formats:

```
$ terraform import google_compute_target_pool.default projects/{{project}}/regions/{{region}
$ terraform import google_compute_target_pool.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_target_pool.default {{region}}/{{name}}
$ terraform import google_compute_target_pool.default {{name}}
```

» google_compute_url_map

UrlMaps are used to route requests to a backend service based on rules that you define for the host and path of an incoming URL.

To get more information about UrlMap, see:

- API documentation

» Example Usage - Url Map Basic

```
resource "google_compute_url_map" "urlmap" {
  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"
  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"
  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               = "health-check"
```

```
  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"
}
```

» Example Usage - Url Map Traffic Director Route

```
resource "google_compute_url_map" "urlmap" {
  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

    route_rules {
      priority = 1
      header_action {
        request_headers_to_remove = ["RemoveMe2"]
        request_headers_to_add {
          header_name = "AddSomethingElse"
          header_value = "MyOtherValue"
          replace = true
```

```
        }
        response_headers_to_remove = ["RemoveMe3"]
        response_headers_to_add {
          header_name = "AddMe"
          header_value = "MyValue"
          replace = false
        }
      }
      match_rules {
        full_path_match = "a full path"
        header_matches {
          header_name = "someheader"
          exact_match = "match this exactly"
          invert_match = true
        }
        ignore_case = true
        metadata_filters {
          filter_match_criteria = "MATCH_ANY"
          filter_labels {
            name = "PLANET"
            value = "MARS"
          }
        }
        query_parameter_matches {
          name = "a query parameter"
          present_match = true
        }
      }
      url_redirect {
        host_redirect = "A host"
        https_redirect = false
        path_redirect = "some/path"
        redirect_response_code = "TEMPORARY_REDIRECT"
        strip_query = true
      }
    }
  }

  test {
    service = google_compute_backend_service.home.self_link
    host    = "hi.com"
    path    = "/home"
  }
}

resource "google_compute_backend_service" "home" {
```

```
  name        = "home"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = [google_compute_health_check.default.self_link]
  load_balancing_scheme = "INTERNAL_SELF_MANAGED"
}

resource "google_compute_health_check" "default" {
  name               = "health-check"
  http_health_check {
    port = 80
  }
}
```

» Example Usage - Url Map Traffic Director Route Partial

```
resource "google_compute_url_map" "urlmap" {
  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

    route_rules {
      priority = 1
      match_rules {
        prefix_match = "/someprefix"
        header_matches {
          header_name = "someheader"
          exact_match = "match this exactly"
          invert_match = true
```

```
        }
      }
      url_redirect {
        path_redirect = "some/path"
        redirect_response_code = "TEMPORARY_REDIRECT"
      }
    }
  }

  test {
    service = google_compute_backend_service.home.self_link
    host    = "hi.com"
    path    = "/home"
  }
}

resource "google_compute_backend_service" "home" {
  name        = "home"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = [google_compute_health_check.default.self_link]
  load_balancing_scheme = "INTERNAL_SELF_MANAGED"
}

resource "google_compute_health_check" "default" {
  name               = "health-check"
  http_health_check {
    port = 80
  }
}
```

» Example Usage - Url Map Traffic Director Path

```
resource "google_compute_url_map" "urlmap" {
  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"]
    route_action {
      cors_policy {
        allow_credentials = true
        allow_headers = ["Allowed content"]
        allow_methods = ["GET"]
        allow_origin_regexes = ["abc.*"]
        allow_origins = ["Allowed origin"]
        expose_headers = ["Exposed header"]
        max_age = 30
        disabled = false
      }
      fault_injection_policy {
        abort {
          http_status = 234
          percentage = 5.6
        }
        delay {
          fixed_delay {
            seconds = 0
            nanos = 50000
          }
          percentage = 7.8
        }
      }
      request_mirror_policy {
        backend_service = google_compute_backend_service.home.self_link
      }
      retry_policy {
        num_retries = 4
        per_try_timeout {
          seconds = 30
        }
        retry_conditions = ["5xx", "deadline-exceeded"]
      }
      timeout {
```

```
          seconds = 20
          nanos = 750000000
        }
        url_rewrite {
          host_rewrite = "A replacement header"
          path_prefix_rewrite = "A replacement path"
        }
        weighted_backend_services {
          backend_service = google_compute_backend_service.home.self_link
          weight = 400
          header_action {
            request_headers_to_remove = ["RemoveMe"]
            request_headers_to_add {
              header_name = "AddMe"
              header_value = "MyValue"
              replace = true
            }
            response_headers_to_remove = ["RemoveMe"]
            response_headers_to_add {
              header_name = "AddMe"
              header_value = "MyValue"
              replace = false
            }
          }
        }
      }
    }
  }

  test {
    service = google_compute_backend_service.home.self_link
    host    = "hi.com"
    path    = "/home"
  }
}

resource "google_compute_backend_service" "home" {
  name        = "home"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = [google_compute_health_check.default.self_link]
  load_balancing_scheme = "INTERNAL_SELF_MANAGED"
}
```

```
resource "google_compute_health_check" "default" {
  name               = "health-check"
  http_health_check {
    port = 80
  }
}
```

» Example Usage - Url Map Traffic Director Path Partial

```
resource "google_compute_url_map" "urlmap" {
  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"]
      route_action {
        cors_policy {
          allow_credentials = true
          allow_headers = ["Allowed content"]
          allow_methods = ["GET"]
          allow_origin_regexes = ["abc.*"]
          allow_origins = ["Allowed origin"]
          expose_headers = ["Exposed header"]
          max_age = 30
          disabled = false
        }
        weighted_backend_services {
          backend_service = google_compute_backend_service.home.self_link
          weight = 400
          header_action {
```

```
            request_headers_to_remove = ["RemoveMe"]
            request_headers_to_add {
              header_name = "AddMe"
              header_value = "MyValue"
              replace = true
            }
            response_headers_to_remove = ["RemoveMe"]
            response_headers_to_add {
              header_name = "AddMe"
              header_value = "MyValue"
              replace = false
            }
          }
        }
      }
    }
  }

  test {
    service = google_compute_backend_service.home.self_link
    host    = "hi.com"
    path    = "/home"
  }
}

resource "google_compute_backend_service" "home" {
  name        = "home"
  port_name   = "http"
  protocol    = "HTTP"
  timeout_sec = 10

  health_checks = [google_compute_health_check.default.self_link]
  load_balancing_scheme = "INTERNAL_SELF_MANAGED"
}

resource "google_compute_health_check" "default" {
  name               = "health-check"
  http_health_check {
    port = 80
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.

- `default_service` - (Optional) The backend service or backend bucket to use when none of the given rules match.
- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `header_action` - (Optional) Specifies changes to request and response headers that need to take effect for the selected backendService. The headerAction specified here take effect after headerAction specified under pathMatcher. Structure is documented below.
- `host_rule` - (Optional) The list of HostRules to use against the URL. Structure is documented below.
- `path_matcher` - (Optional) The list of named PathMatchers to use against the URL. Structure is documented below.
- `test` - (Optional) The list of expected URL mapping tests. Request to update this UrlMap will succeed only if all of the test cases pass. You can specify a maximum of 100 tests per UrlMap. 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.

The `header_action` block supports:

- `request_headers_to_add` - (Optional) Headers to add to a matching request prior to forwarding the request to the backendService. Structure is documented below.
- `request_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the request prior to forwarding the request to the backendService.
- `response_headers_to_add` - (Optional) Headers to add the response prior to sending the response back to the client. Structure is documented below.
- `response_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the response prior to sending the response back to the client.

The `request_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.
- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `response_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.
- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `host_rule` block supports:

- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `hosts` - (Required) The list of host patterns to match. They must be valid hostnames, except * will match any string of ([a-z0-9-.]*). In that case, * must be the first character and must be followed in the pattern by either - or ..
- `path_matcher` - (Required) The name of the PathMatcher to use to match the path portion of the URL if the hostRule matches the URL's host portion.

The `path_matcher` block supports:

- `default_service` - (Optional) The backend service or backend bucket to use when none of the given paths match.
- `description` - (Optional) An optional description of this resource. Provide this property when you create the resource.
- `header_action` - (Optional) Specifies changes to request and response headers that need to take effect for the selected backendService. HeaderAction specified here are applied after the matching HttpRouteRule HeaderAction and before the HeaderAction in the UrlMap Structure is documented below.
- `name` - (Required) The name to which this PathMatcher is referred by the HostRule.
- `path_rule` - (Optional) The list of path rules. Use this list instead of routeRules when routing based on simple path matching is all that's required. The order by which path rules are specified does not matter. Matches are always done on the longest-path-first basis. For example: a pathRule with a path /a/b/c/* will match before /a/b/* irrespective of

the order in which those paths appear in this list. Within a given pathMatcher, only one of pathRules or routeRules must be set. Structure is documented below.

- `route_rules` - (Optional) The list of ordered HTTP route rules. Use this list instead of pathRules when advanced route matching and routing actions are desired. The order of specifying routeRules matters: the first rule that matches will cause its specified routing action to take effect. Within a given pathMatcher, only one of pathRules or routeRules must be set. routeRules are not supported in UrlMaps intended for External load balancers. Structure is documented below.

The `header_action` block supports:

- `request_headers_to_add` - (Optional) Headers to add to a matching request prior to forwarding the request to the backendService. Structure is documented below.
- `request_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the request prior to forwarding the request to the backendService.
- `response_headers_to_add` - (Optional) Headers to add the response prior to sending the response back to the client. Structure is documented below.
- `response_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the response prior to sending the response back to the client.

The `request_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.
- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `response_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.
- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `path_rule` block supports:

- `service` - (Optional) The backend service or backend bucket to use if any of the given paths match.

- `paths` - (Required) The list of path patterns to match. Each must start with / and the only place a
  - is allowed is at the end following a /. The string fed to the path matcher does not include any text after the first ? or #, and those chars are not allowed here.
- `route_action` - (Optional) In response to a matching path, the load balancer performs advanced routing actions like URL rewrites, header transformations, etc. prior to forwarding the request to the selected backend. If routeAction specifies any weightedBackendServices, service must not be set. Conversely if service is set, routeAction cannot contain any weightedBackendServices. Only one of routeAction or urlRedirect must be set. Structure is documented below.
- `url_redirect` - (Optional) When a path pattern is matched, the request is redirected to a URL specified by urlRedirect. If urlRedirect is specified, service or routeAction must not be set. Structure is documented below.

The `route_action` block supports:

- `cors_policy` - (Optional) The specification for allowing client side cross-origin requests. Please see W3C Recommendation for Cross Origin Resource Sharing Structure is documented below.
- `fault_injection_policy` - (Optional) The specification for fault injection introduced into traffic to test the resiliency of clients to backend service failure. As part of fault injection, when clients send requests to a backend service, delays can be introduced by Loadbalancer on a percentage of requests before sending those request to the backend service. Similarly requests from clients can be aborted by the Loadbalancer for a percentage of requests. timeout and retry_policy will be ignored by clients that are configured with a fault_injection_policy. Structure is documented below.
- `request_mirror_policy` - (Optional) Specifies the policy on how requests intended for the route's backends are shadowed to a separate mirrored backend service. Loadbalancer does not wait for responses from the shadow service. Prior to sending traffic to the shadow service, the host / authority header is suffixed with -shadow. Structure is documented below.
- `retry_policy` - (Optional) Specifies the retry policy associated with this route. Structure is documented below.
- `timeout` - (Optional) Specifies the timeout for the selected route. Timeout is computed from the time the request is has been fully processed (i.e. end-of-stream) up until the response has been completely processed. Timeout includes all retries. If not specified, the default value is 15 seconds. Structure is documented below.

- `url_rewrite` - (Optional) The spec to modify the URL of the request, prior to forwarding the request to the matched service Structure is documented below.
- `weighted_backend_services` - (Optional) A list of weighted backend services to send traffic to when a route match occurs. The weights determine the fraction of traffic that flows to their corresponding backend service. If all traffic needs to go to a single backend service, there must be one weightedBackendService with weight set to a non 0 number. Once a backendService is identified and before forwarding the request to the backend service, advanced routing actions like Url rewrites and header transformations are applied depending on additional settings specified in this HttpRouteAction. Structure is documented below.

The `cors_policy` block supports:

- `allow_credentials` - (Optional) In response to a preflight request, setting this to true indicates that the actual request can include user credentials. This translates to the Access- Control-Allow-Credentials header. Defaults to false.
- `allow_headers` - (Optional) Specifies the content for the Access-Control-Allow-Headers header.
- `allow_methods` - (Optional) Specifies the content for the Access-Control-Allow-Methods header.
- `allow_origin_regexes` - (Optional) Specifies the regualar expression patterns that match allowed origins. For regular expression grammar please see en.cppreference.com/w/cpp/regex/ecmascript An origin is allowed if it matches either allow_origins or allow_origin_regex.
- `allow_origins` - (Optional) Specifies the list of origins that will be allowed to do CORS requests. An origin is allowed if it matches either allow_origins or allow_origin_regex.
- `disabled` - (Required) If true, specifies the CORS policy is disabled.
- `expose_headers` - (Optional) Specifies the content for the Access-Control-Expose-Headers header.
- `max_age` - (Optional) Specifies how long the results of a preflight request can be cached. This translates to the content for the Access-Control-Max-Age header.

The `fault_injection_policy` block supports:

- `abort` - (Optional) The specification for how client requests are aborted as part of fault injection. Structure is documented below.
- `delay` - (Optional) The specification for how client requests are delayed as part of fault injection, before being sent to a backend service. Structure

is documented below.

The `abort` block supports:

- `http_status` - (Required) The HTTP status code used to abort the request. The value must be between 200 and 599 inclusive.
- `percentage` - (Required) The percentage of traffic (connections/operations/requests) which will be aborted as part of fault injection. The value must be between 0.0 and 100.0 inclusive.

The `delay` block supports:

- `fixed_delay` - (Required) Specifies the value of the fixed delay interval. Structure is documented below.
- `percentage` - (Required) The percentage of traffic (connections/operations/requests) on which delay will be introduced as part of fault injection. The value must be between 0.0 and 100.0 inclusive.

The `fixed_delay` block supports:

- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.
- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.

The `request_mirror_policy` block supports:

- `backend_service` - (Required) The BackendService resource being mirrored to.

The `retry_policy` block supports:

- `num_retries` - (Optional) Specifies the allowed number retries. This number must be > 0.
- `per_try_timeout` - (Optional) Specifies a non-zero timeout per retry attempt. Structure is documented below.
- `retry_conditions` - (Optional) Specifies one or more conditions when this retry rule applies. Valid values are:
  - 5xx: Loadbalancer will attempt a retry if the backend service responds with any 5xx response code, or if the backend service does not respond at all, example: disconnects, reset, read timeout, connection failure, and refused streams.
  - gateway-error: Similar to 5xx, but only applies to response codes 502, 503 or 504.
  - connect-failure: Loadbalancer will retry on failures connecting to backend services, for example due to connection timeouts.

  - retriable-4xx: Loadbalancer will retry for retriable 4xx response codes. Currently the only retriable error supported is 409.
  - refused-stream: Loadbalancer will retry if the backend service resets the stream with a REFUSED_STREAM error code. This reset type indicates that it is safe to retry.
  - cancelled: Loadbalancer will retry if the gRPC status code in the response header is set to cancelled
  - deadline-exceeded: Loadbalancer will retry if the gRPC status code in the response header is set to deadline-exceeded
  - resource-exhausted: Loadbalancer will retry if the gRPC status code in the response header is set to resource-exhausted
  - unavailable: Loadbalancer will retry if the gRPC status code in the response header is set to unavailable

The `per_try_timeout` block supports:

- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.

The `timeout` block supports:

- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.

- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.

The `url_rewrite` block supports:

- `host_rewrite` - (Optional) Prior to forwarding the request to the selected service, the request's host header is replaced with contents of hostRewrite. The value must be between 1 and 255 characters.

- `path_prefix_rewrite` - (Optional) Prior to forwarding the request to the selected backend service, the matching portion of the request's path is replaced by pathPrefixRewrite. The value must be between 1 and 1024 characters.

The `weighted_backend_services` block supports:

- `backend_service` - (Required) The default BackendService resource. Before forwarding the request to backendService, the loadbalancer applies any relevant headerActions specified as part of this backendServiceWeight.

- `header_action` - (Optional) Specifies changes to request and response headers that need to take effect for the selected backendService. headerAction specified here take effect before headerAction in the enclosing HttpRouteRule, PathMatcher and UrlMap. Structure is documented below.
- `weight` - (Required) Specifies the fraction of traffic sent to backendService, computed as weight / (sum of all weightedBackendService weights in routeAction) . The selection of a backend service is determined only for new traffic. Once a user's request has been directed to a backendService, subsequent requests will be sent to the same backendService as determined by the BackendService's session affinity policy. The value must be between 0 and 1000

The `header_action` block supports:

- `request_headers_to_add` - (Optional) Headers to add to a matching request prior to forwarding the request to the backendService. Structure is documented below.
- `request_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the request prior to forwarding the request to the backendService.
- `response_headers_to_add` - (Optional) Headers to add the response prior to sending the response back to the client. Structure is documented below.
- `response_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the response prior to sending the response back to the client.

The `request_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.
- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `response_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.
- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `url_redirect` block supports:

- `host_redirect` - (Optional) The host that will be used in the redirect response instead of the one that was supplied in the request. The value must be between 1 and 255 characters.
- `https_redirect` - (Optional) If set to true, the URL scheme in the redirected request is set to https. If set to false, the URL scheme of the redirected request will remain the same as that of the request. This must only be set for UrlMaps used in TargetHttpProxys. Setting this true for TargetHttpsProxy is not permitted. Defaults to false.
- `path_redirect` - (Optional) The path that will be used in the redirect response instead of the one that was supplied in the request. Only one of pathRedirect or prefixRedirect must be specified. The value must be between 1 and 1024 characters.
- `prefix_redirect` - (Optional) The prefix that replaces the prefixMatch specified in the HttpRouteRuleMatch, retaining the remaining portion of the URL before redirecting the request.
- `redirect_response_code` - (Optional) The HTTP Status code to use for this RedirectAction. Supported values are:
  - MOVED_PERMANENTLY_DEFAULT, which is the default value and corresponds to 301.
  - FOUND, which corresponds to 302.
  - SEE_OTHER which corresponds to 303.
  - TEMPORARY_REDIRECT, which corresponds to 307. In this case, the request method will be retained.
  - PERMANENT_REDIRECT, which corresponds to 308. In this case, the request method will be retained.
- `strip_query` - (Required) If set to true, any accompanying query portion of the original URL is removed prior to redirecting the request. If set to false, the query portion of the original URL is retained.

The `route_rules` block supports:

- `priority` - (Required) For routeRules within a given pathMatcher, priority determines the order in which load balancer will interpret routeRules. RouteRules are evaluated in order of priority, from the lowest to highest number. The priority of a rule decreases as its number increases (1, 2, 3, N+1). The first rule that matches the request is applied. You cannot configure two or more routeRules with the same priority. Priority for each rule must be set to a number between 0 and 2147483647 inclusive. Priority numbers can have gaps, which enable you to add or remove rules in the future without affecting the rest of the rules. For example, 1, 2, 3, 4, 5, 9, 12, 16 is a valid series of priority numbers to which you could add rules numbered from 6 to 8, 10 to 11, and 13 to 15 in the future without any impact on existing rules.

- `service` - (Optional) The backend service resource to which traffic is directed if this rule is matched. If routeAction is additionally specified, advanced routing actions like URL Rewrites, etc. take effect prior to sending the request to the backend. However, if service is specified, routeAction cannot contain any weightedBackendService s. Conversely, if routeAction specifies any weightedBackendServices, service must not be specified. Only one of urlRedirect, service or routeAction.weightedBackendService must be set.
- `header_action` - (Optional) Specifies changes to request and response headers that need to take effect for the selected backendService. The headerAction specified here are applied before the matching pathMatchers[].headerAction and after pathMatchers[].routeRules[].r outeAction.weightedBackendService.backendServiceWeightAction[].headerAction Structure is documented below.
- `match_rules` - (Optional) The rules for determining a match. Structure is documented below.
- `route_action` - (Optional) In response to a matching matchRule, the load balancer performs advanced routing actions like URL rewrites, header transformations, etc. prior to forwarding the request to the selected backend. If routeAction specifies any weightedBackendServices, service must not be set. Conversely if service is set, routeAction cannot contain any weightedBackendServices. Only one of routeAction or urlRedirect must be set. Structure is documented below.
- `url_redirect` - (Optional) When this rule is matched, the request is redirected to a URL specified by urlRedirect. If urlRedirect is specified, service or routeAction must not be set. Structure is documented below.

The `header_action` block supports:

- `request_headers_to_add` - (Optional) Headers to add to a matching request prior to forwarding the request to the backendService. Structure is documented below.
- `request_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the request prior to forwarding the request to the backendService.
- `response_headers_to_add` - (Optional) Headers to add the response prior to sending the response back to the client. Structure is documented below.
- `response_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the response prior to sending the response back to the client.

The `request_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.

- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `response_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.
- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `match_rules` block supports:

- `full_path_match` - (Optional) For satifying the matchRule condition, the path of the request must exactly match the value specified in fullPathMatch after removing any query parameters and anchor that may be part of the original URL. FullPathMatch must be between 1 and 1024 characters. Only one of prefixMatch, fullPathMatch or regexMatch must be specified.
- `header_matches` - (Optional) Specifies a list of header match criteria, all of which must match corresponding headers in the request. Structure is documented below.
- `ignore_case` - (Optional) Specifies that prefixMatch and fullPathMatch matches are case sensitive. Defaults to false.
- `metadata_filters` - (Optional) Opaque filter criteria used by Loadbalancer to restrict routing configuration to a limited set xDS compliant clients. In their xDS requests to Loadbalancer, xDS clients present node metadata. If a match takes place, the relevant routing configuration is made available to those proxies. For each metadataFilter in this list, if its filterMatchCriteria is set to MATCH_ANY, at least one of the filterLabels must match the corresponding label provided in the metadata. If its filterMatchCriteria is set to MATCH_ALL, then all of its filterLabels must match with corresponding labels in the provided metadata. metadataFilters specified here can be overrides those specified in ForwardingRule that refers to this UrlMap. metadataFilters only applies to Loadbalancers that have their loadBalancingScheme set to INTERNAL_SELF_MANAGED. Structure is documented below.
- `prefix_match` - (Optional) For satifying the matchRule condition, the request's path must begin with the specified prefixMatch. prefixMatch must begin with a /. The value must be between 1 and 1024 characters. Only one of prefixMatch, fullPathMatch or regexMatch must be specified.

- `query_parameter_matches` - (Optional) Specifies a list of query parameter match criteria, all of which must match corresponding query parameters in the request. Structure is documented below.
- `regex_match` - (Optional) For satifying the matchRule condition, the path of the request must satisfy the regular expression specified in regexMatch after removing any query parameters and anchor supplied with the original URL. For regular expression grammar please see en.cppreference.com/w/cpp/regex/ecmascript Only one of prefixMatch, fullPathMatch or regexMatch must be specified.

The `header_matches` block supports:

- `exact_match` - (Optional) The value should exactly match contents of exactMatch. Only one of exactMatch, prefixMatch, suffixMatch, regexMatch, presentMatch or rangeMatch must be set.
- `header_name` - (Required) The name of the HTTP header to match. For matching against the HTTP request's authority, use a headerMatch with the header name ":authority". For matching a request's method, use the headerName ":method".
- `invert_match` - (Optional) If set to false, the headerMatch is considered a match if the match criteria above are met. If set to true, the headerMatch is considered a match if the match criteria above are NOT met. Defaults to false.
- `prefix_match` - (Optional) The value of the header must start with the contents of prefixMatch. Only one of exactMatch, prefixMatch, suffixMatch, regexMatch, presentMatch or rangeMatch must be set.
- `present_match` - (Optional) A header with the contents of headerName must exist. The match takes place whether or not the request's header has a value or not. Only one of exactMatch, prefixMatch, suffixMatch, regexMatch, presentMatch or rangeMatch must be set.
- `range_match` - (Optional) The header value must be an integer and its value must be in the range specified in rangeMatch. If the header does not contain an integer, number or is empty, the match fails. For example for a range [-5, 0] - -3 will match. - 0 will not match. - 0.25 will not match. - -3someString will not match. Only one of exactMatch, prefixMatch, suffixMatch, regexMatch, presentMatch or rangeMatch must be set. Structure is documented below.
- `regex_match` - (Optional) The value of the header must match the regualar expression specified in regexMatch. For regular expression grammar, please see: en.cppreference.com/w/cpp/regex/ecmascript For matching against a port specified in the HTTP request, use a headerMatch with headerName set to PORT and a regular expression that satisfies the RFC2616 Host header's port specifier. Only one of

exactMatch, prefixMatch, suffixMatch, regexMatch, presentMatch or rangeMatch must be set.

- `suffix_match` - (Optional) The value of the header must end with the contents of suffixMatch. Only one of exactMatch, prefixMatch, suffixMatch, regexMatch, presentMatch or rangeMatch must be set.

The `range_match` block supports:

- `range_end` - (Required) The end of the range (exclusive).
- `range_start` - (Required) The start of the range (inclusive).

The `metadata_filters` block supports:

- `filter_labels` - (Required) The list of label value pairs that must match labels in the provided metadata based on filterMatchCriteria This list must not be empty and can have at the most 64 entries. Structure is documented below.
- `filter_match_criteria` - (Required) Specifies how individual filterLabel matches within the list of filterLabels contribute towards the overall metadataFilter match. Supported values are:
  - MATCH_ANY: At least one of the filterLabels must have a matching label in the provided metadata.
  - MATCH_ALL: All filterLabels must have matching labels in the provided metadata.

The `filter_labels` block supports:

- `name` - (Required) Name of metadata label. The name can have a maximum length of 1024 characters and must be at least 1 character long.
- `value` - (Required) The value of the label must match the specified value. value can have a maximum length of 1024 characters.

The `query_parameter_matches` block supports:

- `exact_match` - (Optional) The queryParameterMatch matches if the value of the parameter exactly matches the contents of exactMatch. Only one of presentMatch, exactMatch and regexMatch must be set.
- `name` - (Required) The name of the query parameter to match. The query parameter must exist in the request, in the absence of which the request match fails.
- `present_match` - (Optional) Specifies that the queryParameterMatch matches if the request contains the query parameter, irrespective of whether the parameter has a value or not. Only one of presentMatch, exactMatch and regexMatch must be set.
- `regex_match` - (Optional) The queryParameterMatch matches if the value of the parameter matches the regular expression specified

by regexMatch. For the regular expression grammar, please see en.cppreference.com/w/cpp/regex/ecmascript Only one of presentMatch, exactMatch and regexMatch must be set.

The `route_action` block supports:

- `cors_policy` - (Optional) The specification for allowing client side cross-origin requests. Please see W3C Recommendation for Cross Origin Resource Sharing Structure is documented below.
- `fault_injection_policy` - (Optional) The specification for fault injection introduced into traffic to test the resiliency of clients to backend service failure. As part of fault injection, when clients send requests to a backend service, delays can be introduced by Loadbalancer on a percentage of requests before sending those request to the backend service. Similarly requests from clients can be aborted by the Loadbalancer for a percentage of requests. timeout and retry_policy will be ignored by clients that are configured with a fault_injection_policy. Structure is documented below.
- `request_mirror_policy` - (Optional) Specifies the policy on how requests intended for the route's backends are shadowed to a separate mirrored backend service. Loadbalancer does not wait for responses from the shadow service. Prior to sending traffic to the shadow service, the host / authority header is suffixed with -shadow. Structure is documented below.
- `retry_policy` - (Optional) Specifies the retry policy associated with this route. Structure is documented below.
- `timeout` - (Optional) Specifies the timeout for the selected route. Timeout is computed from the time the request is has been fully processed (i.e. end-of-stream) up until the response has been completely processed. Timeout includes all retries. If not specified, the default value is 15 seconds. Structure is documented below.
- `url_rewrite` - (Optional) The spec to modify the URL of the request, prior to forwarding the request to the matched service Structure is documented below.
- `weighted_backend_services` - (Optional) A list of weighted backend services to send traffic to when a route match occurs. The weights determine the fraction of traffic that flows to their corresponding backend service. If all traffic needs to go to a single backend service, there must be one weightedBackendService with weight set to a non 0 number. Once a backendService is identified and before forwarding the request to the backend service, advanced routing actions like Url rewrites and header transformations are applied depending on additional settings specified in this HttpRouteAction. Structure is documented below.

The `cors_policy` block supports:

- `allow_credentials` - (Optional) In response to a preflight request, setting this to true indicates that the actual request can include user credentials. This translates to the Access- Control-Allow-Credentials header. Defaults to false.
- `allow_headers` - (Optional) Specifies the content for the Access-Control-Allow-Headers header.
- `allow_methods` - (Optional) Specifies the content for the Access-Control-Allow-Methods header.
- `allow_origin_regexes` - (Optional) Specifies the regualar expression patterns that match allowed origins. For regular expression grammar please see en.cppreference.com/w/cpp/regex/ecmascript An origin is allowed if it matches either allow_origins or allow_origin_regex.
- `allow_origins` - (Optional) Specifies the list of origins that will be allowed to do CORS requests. An origin is allowed if it matches either allow_origins or allow_origin_regex.
- `disabled` - (Optional) If true, specifies the CORS policy is disabled. which indicates that the CORS policy is in effect. Defaults to false.
- `expose_headers` - (Optional) Specifies the content for the Access-Control-Expose-Headers header.
- `max_age` - (Optional) Specifies how long the results of a preflight request can be cached. This translates to the content for the Access-Control-Max-Age header.

The `fault_injection_policy` block supports:

- `abort` - (Optional) The specification for how client requests are aborted as part of fault injection. Structure is documented below.
- `delay` - (Optional) The specification for how client requests are delayed as part of fault injection, before being sent to a backend service. Structure is documented below.

The `abort` block supports:

- `http_status` - (Optional) The HTTP status code used to abort the request. The value must be between 200 and 599 inclusive.
- `percentage` - (Optional) The percentage of traffic (connections/operations/requests) which will be aborted as part of fault injection. The value must be between 0.0 and 100.0 inclusive.

The `delay` block supports:

- `fixed_delay` - (Optional) Specifies the value of the fixed delay interval. Structure is documented below.

- `percentage` - (Optional) The percentage of traffic (connections/operations/requests) on which delay will be introduced as part of fault injection. The value must be between 0.0 and 100.0 inclusive.

The `fixed_delay` block supports:

- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.
- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.

The `request_mirror_policy` block supports:

- `backend_service` - (Required) The BackendService resource being mirrored to.

The `retry_policy` block supports:

- `num_retries` - (Required) Specifies the allowed number retries. This number must be > 0.
- `per_try_timeout` - (Optional) Specifies a non-zero timeout per retry attempt. If not specified, will use the timeout set in HttpRouteAction. If timeout in HttpRouteAction is not set, will use the largest timeout among all backend services associated with the route. Structure is documented below.
- `retry_conditions` - (Optional) Specfies one or more conditions when this retry rule applies. Valid values are:
  - 5xx: Loadbalancer will attempt a retry if the backend service responds with any 5xx response code, or if the backend service does not respond at all, example: disconnects, reset, read timeout, connection failure, and refused streams.
  - gateway-error: Similar to 5xx, but only applies to response codes 502, 503 or 504.
  - connect-failure: Loadbalancer will retry on failures connecting to backend services, for example due to connection timeouts.
  - retriable-4xx: Loadbalancer will retry for retriable 4xx response codes. Currently the only retriable error supported is 409.
  - refused-stream: Loadbalancer will retry if the backend service resets the stream with a REFUSED_STREAM error code. This reset type indicates that it is safe to retry.
  - cancelled: Loadbalancer will retry if the gRPC status code in the response header is set to cancelled
  - deadline-exceeded: Loadbalancer will retry if the gRPC status code in the response header is set to deadline-exceeded

  - resource-exhausted: Loadbalancer will retry if the gRPC status code in the response header is set to resource-exhausted
  - unavailable: Loadbalancer will retry if the gRPC status code in the response header is set to unavailable

The `per_try_timeout` block supports:

- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.
- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.

The `timeout` block supports:

- `nanos` - (Optional) Span of time that's a fraction of a second at nanosecond resolution. Durations less than one second are represented with a 0 `seconds` field and a positive `nanos` field. Must be from 0 to 999,999,999 inclusive.
- `seconds` - (Required) Span of time at a resolution of a second. Must be from 0 to 315,576,000,000 inclusive.

The `url_rewrite` block supports:

- `host_rewrite` - (Optional) Prior to forwarding the request to the selected service, the request's host header is replaced with contents of hostRewrite. The value must be between 1 and 255 characters.
- `path_prefix_rewrite` - (Optional) Prior to forwarding the request to the selected backend service, the matching portion of the request's path is replaced by pathPrefixRewrite. The value must be between 1 and 1024 characters.

The `weighted_backend_services` block supports:

- `backend_service` - (Required) The default BackendService resource. Before forwarding the request to backendService, the loadbalancer applies any relevant headerActions specified as part of this backendServiceWeight.
- `header_action` - (Optional) Specifies changes to request and response headers that need to take effect for the selected backendService. headerAction specified here take effect before headerAction in the enclosing HttpRouteRule, PathMatcher and UrlMap. Structure is documented below.
- `weight` - (Required) Specifies the fraction of traffic sent to backendService, computed as weight / (sum of all weightedBackendService weights in routeAction) . The selection of a backend service is determined only for new traffic. Once a user's request has been directed to a backendService,

subsequent requests will be sent to the same backendService as determined by the BackendService's session affinity policy. The value must be between 0 and 1000

The `header_action` block supports:

- `request_headers_to_add` - (Optional) Headers to add to a matching request prior to forwarding the request to the backendService. Structure is documented below.
- `request_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the request prior to forwarding the request to the backendService.
- `response_headers_to_add` - (Optional) Headers to add the response prior to sending the response back to the client. Structure is documented below.
- `response_headers_to_remove` - (Optional) A list of header names for headers that need to be removed from the response prior to sending the response back to the client.

The `request_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.
- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `response_headers_to_add` block supports:

- `header_name` - (Required) The name of the header.
- `header_value` - (Required) The value of the header to add.
- `replace` - (Required) If false, headerValue is appended to any values that already exist for the header. If true, headerValue is set for the header, discarding any values that were set for that header.

The `url_redirect` block supports:

- `host_redirect` - (Optional) The host that will be used in the redirect response instead of the one that was supplied in the request. The value must be between 1 and 255 characters.
- `https_redirect` - (Optional) If set to true, the URL scheme in the redirected request is set to https. If set to false, the URL scheme of the redirected request will remain the same as that of the request. This must only be set for UrlMaps used in TargetHttpProxys. Setting this true for TargetHttpsProxy is not permitted. Defaults to false.

- `path_redirect` - (Optional) The path that will be used in the redirect response instead of the one that was supplied in the request. Only one of pathRedirect or prefixRedirect must be specified. The value must be between 1 and 1024 characters.
- `prefix_redirect` - (Optional) The prefix that replaces the prefixMatch specified in the HttpRouteRuleMatch, retaining the remaining portion of the URL before redirecting the request.
- `redirect_response_code` - (Optional) The HTTP Status code to use for this RedirectAction. Supported values are: - MOVED_PERMANENTLY_DEFAULT, which is the default value and corresponds to 301. - FOUND, which corresponds to 302. - SEE_OTHER which corresponds to 303. - TEMPORARY_REDIRECT, which corresponds to 307. In this case, the request method will be retained. - PERMANENT_REDIRECT, which corresponds to 308. In this case, the request method will be retained.
- `strip_query` - (Optional) If set to true, any accompanying query portion of the original URL is removed prior to redirecting the request. If set to false, the query portion of the original URL is retained. Defaults to false.

The `test` block supports:

- `description` - (Optional) Description of this test case.
- `host` - (Required) Host portion of the URL.
- `path` - (Required) Path portion of the URL.
- `service` - (Required) The backend service or backend bucket link that should be matched by this test.

» Attributes Reference

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

- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `map_id` - The unique identifier for the resource.
- `fingerprint` - Fingerprint of this resource. A hash of the contents stored in this object. This field is used in optimistic locking.
- `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

UrlMap can be imported using any of these accepted formats:

```
$ terraform import google_compute_url_map.default projects/{{project}}/global/urlMaps/{{name
$ terraform import google_compute_url_map.default {{project}}/{{name}}
$ terraform import google_compute_url_map.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_vpn_gateway

Represents a VPN gateway running in GCP. This virtual device is managed by Google, but used only by you.

To get more information about VpnGateway, see:

- API documentation

» Example Usage - Target Vpn Gateway Basic

```
resource "google_compute_vpn_gateway" "target_gateway" {
  name    = "vpn1"
  network = google_compute_network.network1.self_link
}

resource "google_compute_network" "network1" {
  name = "network1"
```

```
}

resource "google_compute_address" "vpn_static_ip" {
  name = "vpn-static-ip"
}

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"
  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"
  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

  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) Name of the resource. Provided by the client when the resource is created. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `network` - (Required) The network this VPN gateway is accepting traffic for.

- `description` - (Optional) An optional description of this resource.
- `region` - (Optional) The region this gateway should sit in.
- `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.
- `gateway_id` - The unique identifier for the resource.
- `self_link` - The URI of the created resource.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

VpnGateway can be imported using any of these accepted formats:

```
$ terraform import google_compute_vpn_gateway.default projects/{{project}}/regions/{{region}
$ terraform import google_compute_vpn_gateway.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_vpn_gateway.default {{region}}/{{name}}
$ terraform import google_compute_vpn_gateway.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_compute_vpn_tunnel

VPN tunnel resource.

To get more information about VpnTunnel, see:

- API documentation
- How-to Guides
  - Cloud VPN Overview
  - Networks and Tunnel Routing

Warning: 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 - Vpn Tunnel Basic

```
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
```

```
  depends_on = [
    google_compute_forwarding_rule.fr_esp,
    google_compute_forwarding_rule.fr_udp500,
    google_compute_forwarding_rule.fr_udp4500,
  ]
}

resource "google_compute_vpn_gateway" "target_gateway" {
  name    = "vpn1"
  network = google_compute_network.network1.self_link
}

resource "google_compute_network" "network1" {
  name = "network1"
}

resource "google_compute_address" "vpn_static_ip" {
  name = "vpn-static-ip"
}

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"
  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"
  ip_address  = google_compute_address.vpn_static_ip.address
  target      = google_compute_vpn_gateway.target_gateway.self_link
}

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
}
```

» Example Usage - Vpn Tunnel Beta

```
resource "google_compute_vpn_tunnel" "tunnel1" {
  provider      = google-beta
  name          = "tunnel1"
  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,
  ]

  labels = {
    foo = "bar"
  }
}

resource "google_compute_vpn_gateway" "target_gateway" {
  provider = google-beta
  name     = "vpn1"
  network  = google_compute_network.network1.self_link
}

resource "google_compute_network" "network1" {
  provider = google-beta
  name     = "network1"
}

resource "google_compute_address" "vpn_static_ip" {
```

```
  provider = google-beta
  name     = "vpn-static-ip"
}

resource "google_compute_forwarding_rule" "fr_esp" {
  provider    = google-beta
  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" {
  provider    = google-beta
  name        = "fr-udp500"
  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" {
  provider    = google-beta
  name        = "fr-udp4500"
  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_route" "route1" {
  provider   = google-beta
  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
}

provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the resource. The name must be 1-63 characters long, and comply with RFC1035. Specifically, the name must be 1-63 characters long and match the regular expression `[a-z]([-a-z0-9]*[a-z0-9])?` which means the first character must be a lowercase letter, and all following characters must be a dash, lowercase letter, or digit, except the last character, which cannot be a dash.
- `shared_secret` - (Required) Shared secret used to set the secure session between the Cloud VPN gateway and the peer VPN gateway.

- `description` - (Optional) An optional description of this resource.
- `target_vpn_gateway` - (Optional) URL of the Target VPN gateway with which this VPN tunnel is associated.
- `vpn_gateway` - (Optional, Beta) URL of the VPN gateway with which this VPN tunnel is associated. This must be used if a High Availability VPN gateway resource is created. This field must reference a `google_compute_ha_vpn_gateway` resource.
- `vpn_gateway_interface` - (Optional, Beta) The interface ID of the VPN gateway with which this VPN tunnel is associated.
- `peer_external_gateway` - (Optional, Beta) URL of the peer side external VPN gateway to which this VPN tunnel is connected.
- `peer_external_gateway_interface` - (Optional, Beta) The interface ID of the external VPN gateway to which this VPN tunnel is connected.
- `peer_gcp_gateway` - (Optional, Beta) URL of the peer side HA GCP VPN gateway to which this VPN tunnel is connected. If provided, the VPN tunnel will automatically use the same vpn_gateway_interface ID in the peer GCP VPN gateway. This field must reference a `google_compute_ha_vpn_gateway` resource.
- `router` - (Optional) URL of router resource to be used for dynamic routing.
- `peer_ip` - (Optional) IP address of the peer VPN gateway. Only IPv4 is supported.
- `ike_version` - (Optional) IKE protocol version to use when establishing the VPN tunnel with peer VPN gateway. Acceptable IKE versions are 1 or 2. Default version is 2.

- `local_traffic_selector` - (Optional) Local traffic selector to use when establishing the VPN tunnel with peer VPN gateway. The value should be a CIDR formatted string, for example `192.168.0.0/16`. The ranges should be disjoint. Only IPv4 is supported.
- `remote_traffic_selector` - (Optional) Remote traffic selector to use when establishing the VPN tunnel with peer VPN gateway. The value should be a CIDR formatted string, for example `192.168.0.0/16`. The ranges should be disjoint. Only IPv4 is supported.
- `labels` - (Optional, Beta) Labels to apply to this VpnTunnel.
- `region` - (Optional) The region where the tunnel is located. If unset, is set to the region of `target_vpn_gateway`.
- `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:

- `tunnel_id` - The unique identifier for the resource. This identifier is defined by the server.
- `creation_timestamp` - Creation timestamp in RFC3339 text format.
- `shared_secret_hash` - Hash of the shared secret.
- `label_fingerprint` - The fingerprint used for optimistic locking of this resource. Used internally during updates.
- `detailed_status` - Detailed status message for the VPN tunnel.
- `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

VpnTunnel can be imported using any of these accepted formats:

```
$ terraform import google_compute_vpn_tunnel.default projects/{{project}}/regions/{{region}}
$ terraform import google_compute_vpn_tunnel.default {{project}}/{{region}}/{{name}}
$ terraform import google_compute_vpn_tunnel.default {{region}}/{{name}}
$ terraform import google_compute_vpn_tunnel.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_container_analysis_note

Provides a detailed description of a Note.

To get more information about Note, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Container Analysis Note Basic

```
resource "google_container_analysis_note" "note" {
  name = "test-attestor-note"
  attestation_authority {
    hint {
      human_readable_name = "Attestor Note"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the note.
- `attestation_authority` - (Required) Note kind that represents a logical attestation "role" or "authority". For example, an organization might have one AttestationAuthority for "QA" and one for "build". This Note is intended to act strictly as a grouping mechanism for the attached Occurrences (Attestations). This grouping mechanism also provides a security boundary, since IAM ACLs gate the ability for a principle to attach an Occurrence to a given Note. It also provides a single point of lookup to find all attached Attestation Occurrences, even if they don't all live in the same project. Structure is documented below.

The `attestation_authority` block supports:

- `hint` - (Required) This submessage provides human-readable hints about the purpose of the AttestationAuthority. Because the name of a Note acts as its resource reference, it is important to disambiguate the canonical name of the Note (which might be a UUID for security purposes) from "readable" names more suitable for debug output. Note that these hints should NOT be used to look up AttestationAuthorities in security sensitive contexts, such as when looking up Attestations to verify. Structure is documented below.

The `hint` block supports:

- `human_readable_name` - (Required) The human readable name of this Attestation Authority, for example "qa".

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

» 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

Note can be imported using any of these accepted formats:

```
$ terraform import google_container_analysis_note.default projects/{{project}}/notes/{{name}
$ terraform import google_container_analysis_note.default {{project}}/{{name}}
$ terraform import google_container_analysis_note.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_container_cluster

Manages a Google Kubernetes Engine (GKE) cluster. For more information see the official documentation and the API reference.

Note: All arguments and attributes, including basic auth username and passwords as well as certificate outputs will be stored in the raw state as plaintext. Read more about sensitive data in state.

» Example Usage - with a separately managed node pool (recommended)

```
resource "google_container_cluster" "primary" {
  name     = "my-gke-cluster"
  location = "us-central1"

  # We can't create a cluster with no node pool defined, but we want to only use
  # separately managed node pools. So we create the smallest possible default
  # node pool and immediately delete it.
  remove_default_node_pool = true
  initial_node_count       = 1

  master_auth {
    username = ""
    password = ""

    client_certificate_config {
      issue_client_certificate = false
    }
  }
}

resource "google_container_node_pool" "primary_preemptible_nodes" {
  name       = "my-node-pool"
  location   = "us-central1"
```

```
  cluster    = google_container_cluster.primary.name
  node_count = 1

  node_config {
    preemptible  = true
    machine_type = "n1-standard-1"

    metadata = {
      disable-legacy-endpoints = "true"
    }

    oauth_scopes = [
      "https://www.googleapis.com/auth/logging.write",
      "https://www.googleapis.com/auth/monitoring",
    ]
  }
}
```

» Example Usage - with the default node pool

```
resource "google_container_cluster" "primary" {
  name               = "marcellus-wallace"
  location           = "us-central1-a"
  initial_node_count = 3

  master_auth {
    username = ""
    password = ""

    client_certificate_config {
      issue_client_certificate = false
    }
  }

  node_config {
    oauth_scopes = [
      "https://www.googleapis.com/auth/logging.write",
      "https://www.googleapis.com/auth/monitoring",
    ]

    metadata = {
      disable-legacy-endpoints = "true"
    }

    labels = {
```

```
      foo = "bar"
    }

    tags = ["foo", "bar"]
  }

  timeouts {
    create = "30m"
    update = "40m"
  }
}
```

» Argument Reference

- `name` - (Required) The name of the cluster, unique within the project and location.

- `location` - (Optional) The location (region or zone) in which the cluster master will be created, as well as the default node location. If you specify a zone (such as `us-central1-a`), the cluster will be a zonal cluster with a single cluster master. If you specify a region (such as `us-west1`), the cluster will be a regional cluster with multiple masters spread across zones in the region, and with default node locations in those zones as well
- `node_locations` - (Optional) The list of zones in which the cluster's nodes are located. Nodes must be in the region of their regional cluster or in the same region as their cluster's zone for zonal clusters. If this is specified for a zonal cluster, omit the cluster's zone.

A "multi-zonal" cluster is a zonal cluster with at least one additional zone defined; in a multi-zonal cluster, the cluster master is only present in a single zone while nodes are present in each of the primary zone and the node locations. In contrast, in a regional cluster, cluster master nodes are present in multiple zones in the region. For that reason, regional clusters should be preferred.

- `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 in CIDR notation (e.g. `10.96.0.0/14`). Leave blank to have one automatically chosen or specify a `/14` block in `10.0.0.0/8`. This field will only work for routes-based clusters, where `ip_allocation_policy` is not defined.
- `cluster_autoscaling` - (Optional, Beta) Per-cluster configuration of Node Auto-Provisioning with Cluster Autoscaler to automatically

adjust the size of the cluster and create/delete node pools based on the current needs of the cluster's workload. See the guide to using Node Auto-Provisioning for more details. Structure is documented below.

- `database_encryption` - (Optional, Beta). Structure is documented below.
- `description` - (Optional) Description of the cluster.
- `default_max_pods_per_node` - (Optional) The default maximum number of pods per node in this cluster. This doesn't work on "routes-based" clusters, clusters that don't have IP Aliasing enabled. See the official documentation for more information.
- `enable_binary_authorization` - (Optional, Beta) Enable Binary Authorization for this cluster. If enabled, all container images will be validated by Google Binary Authorization.
- `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_tpu` - (Optional, Beta) Whether to enable Cloud TPU resources in this cluster. See the official documentation.
- `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 `false`
- `enable_shielded_nodes` - (Optional, Beta) Enable Shielded Nodes features on all nodes in this cluster. Defaults to `false`.
- `initial_node_count` - (Optional) The number of nodes to create in this cluster's default node pool. In regional or multi-zonal clusters, this is the number of nodes per zone. Must be set if `node_pool` is not set. If you're using `google_container_node_pool` objects with no default node pool, you'll need to set this to a value of at least `1`, alongside setting `remove_default_node_pool` to `true`.
- `ip_allocation_policy` - (Optional) Configuration of cluster IP allocation for VPC-native clusters. Adding this block enables IP aliasing, making the cluster VPC-native instead of routes-based. Structure is documented below.
- `logging_service` - (Optional) The logging service that the cluster should write logs to. Available options include `logging.googleapis.com`, `logging.googleapis.com/kubernetes`, and `none`. Defaults to `logging.googleapis.com/kubernetes`

- `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. Some values in this block are only returned by the API if your service account has permission to get credentials for your GKE cluster. If you see an unexpected diff removing a username/password or unsetting your client cert, ensure you have the `container.clusters.getCredentials` permission. 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). Most users will find the `google_container_engine_versions` data source useful - it indicates which versions are available, and can be use to approximate fuzzy versions in a Terraform-compatible way. If you intend to specify versions manually, the docs describe the various acceptable formats for this field.

If you are using the `google_container_engine_versions` datasource with a regional cluster, ensure that you have provided a `location` to the datasource. A region can have a different set of supported versions than its corresponding zones, and not all zones in a region are guaranteed to support the same 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`, `monitoring.googleapis.com/kubernetes`, and `none`. Defaults to `monitoring.googleapis.com/kubernetes`
- `network` - (Optional) The name or self_link of the Google Compute Engine network to which the cluster is connected. For Shared VPC, set this to the self link of the shared network.
- `network_policy` - (Optional) Configuration options for the NetworkPolicy feature. Structure is documented below.
- `node_config` - (Optional) Parameters used in creating the default node pool. Generally, this field should not be used at the same time as a `google_container_node_pool` or a `node_pool` block; this configuration

manages the default node pool, which isn't recommended to be used with Terraform. Structure is documented below.

- `node_pool` - (Optional) List of node pools associated with this cluster. See google_container_node_pool for schema. **Warning:** node pools defined inside a cluster can't be changed (or added/removed) after cluster creation without deleting and recreating the entire cluster. Unless you absolutely need the ability to say "these are the *only* node pools associated with this cluster", use the google_container_node_pool resource instead of this property.
- `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. This only affects nodes in the default node pool. While a fuzzy version can be specified, it's recommended that you specify explicit versions as Terraform will see spurious diffs when fuzzy versions are used. See the `google_container_engine_versions` data source's `version_prefix` field to approximate fuzzy versions in a Terraform-compatible way. To update nodes in other node pools, use the `version` attribute on the node pool.
- `pod_security_policy_config` - (Optional, Beta) Configuration for the PodSecurityPolicy feature. Structure is documented below.
- `authenticator_groups_config` - (Optional) Configuration for the Google Groups for GKE feature. Structure is documented below.
- `private_cluster_config` - (Optional) Configuration for private clusters, clusters with private nodes. 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.
- `release_channel` - (Optional, Beta) Configuration options for the Release channel feature, which provide more control over automatic upgrades of your GKE clusters. Structure is documented below.
- `remove_default_node_pool` - (Optional) If `true`, deletes the default node pool upon cluster creation. If you're using `google_container_node_pool` resources with no default node pool, this should be set to `true`, alongside setting `initial_node_count` to at least `1`.
- `resource_labels` - (Optional) The GCE resource labels (a map of key/value pairs) to be applied to the cluster.
- `resource_usage_export_config` - (Optional, Beta) Configuration for the ResourceUsageExportConfig feature. Structure is documented below.
- `subnetwork` - (Optional) The name or self_link of the Google Compute Engine subnetwork in which the cluster's instances are launched.

- `vertical_pod_autoscaling` - (Optional, Beta) Vertical Pod Autoscaling automatically adjusts the resources of pods controlled by it. Structure is documented below.
- `workload_identity_config` - (Optional, Beta) Workload Identity allows Kubernetes service accounts to act as a user-managed Google IAM Service Account. Structure is documented below.
- `enable_intranode_visibility` - (Optional, Beta) Whether Intra-node visibility is enabled for this cluster. This makes same node pod to pod traffic visible for VPC network.

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.
- `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. To enable this, you must also define a `network_policy` block, otherwise nothing will happen. It can only be disabled if the nodes already do not have network policies enabled. Defaults to disabled; set `disabled = false` to enable.
- `istio_config` - (Optional, Beta). Structure is documented below.
- `cloudrun_config` - (Optional, Beta). The status of the CloudRun addon. It requires `istio_config` enabled. It is disabled by default. Set `disabled = false` to enable. This addon can only be enabled at cluster creation time.

This example `addons_config` disables two addons:

```
addons_config {
  http_load_balancing {
    disabled = true
  }

  horizontal_pod_autoscaling {
    disabled = true
  }
}
```

The `database_encryption` block supports:

- `state` - (Required) `ENCRYPTED` or `DECRYPTED`
- `key_name` - (Required) the key to use to encrypt/decrypt secrets. See the DatabaseEncryption definition for more information.

The `istio_config` block supports:

- `disabled` - (Optional) The status of the Istio addon, which makes it easy to set up Istio for services in a cluster. It is disabled by default. Set `disabled = false` to enable.
- `auth` - (Optional) The authentication type between services in Istio. Available options include `AUTH_MUTUAL_TLS`.

The `cluster_autoscaling` block supports:

- `enabled` - (Required) Whether node auto-provisioning is enabled. Resource limits for `cpu` and `memory` must be defined to enable node auto-provisioning.
- `resource_limits` - (Optional) Global constraints for machine resources in the cluster. Configuring the `cpu` and `memory` types is required if node auto-provisioning is enabled. These limits will apply to node pool autoscaling in addition to node auto-provisioning. Structure is documented below.
- `auto_provisioning_defaults` - (Optional) Contains defaults for a node pool created by NAP. Structure is documented below.

The `resource_limits` block supports:

- `resource_type` - (Required) The type of the resource. For example, `cpu` and `memory`. See the guide to using Node Auto-Provisioning for a list of types.
- `minimum` - (Optional) Minimum amount of the resource in the cluster.
- `maximum` - (Optional) Maximum amount of the resource in the cluster.

The `auto_provisioning_defaults` block supports:

- `oauth_scopes` - (Optional) Scopes that are used by NAP when creating node pools. If `oauth_scopes` are specified, `service_account` must be empty.

`monitoring.write` is always enabled regardless of user input. `monitoring` and `logging.write` may also be enabled depending on the values for `monitoring_service` and `logging_service`.

- `service_account` - (Optional) The Google Cloud Platform Service Account to be used by the node VMs. If `service_account` is specified, `oauth_scopes` must be empty.

The `authenticator_groups_config` block supports:

- `security_group` - (Required) The name of the RBAC security group for use with Google security groups in Kubernetes RBAC. Group name must be in format `gke-security-groups@yourdomain.com`.

The `maintenance_policy` block supports:

- `daily_maintenance_window` - (Required in GA, Optional in Beta) 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"
  }
}
```

- `recurring_window` - (Optional, Beta) Time window for recurring maintenance operations.

Specify `start_time` and `end_time` in RFC3339 date format. The start time's date is the initial date that the window starts, and the end time is used for calculating duration. Specify `recurrence` in RFC5545 RRULE format, to specify when this recurs.

Examples: `maintenance_policy { recurring_window { start_time = "2019-08-01T02:00:00Z" end_time = "2019-08-01T06:00:00Z" recurrence = "FREQ=DAILY" } }`

```
maintenance_policy {
  recurring_window {
    start_time = "2019-01-01T09:00:00-04:00"
    end_time = "2019-01-01T17:00:00-04:00"
    recurrence = "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR"
  }
}
```

In beta, one or the other of `recurring_window` and `daily_maintenance_window` is required if a `maintenance_policy` block is supplied.

The `ip_allocation_policy` block supports:

- `cluster_secondary_range_name` - (Optional) The name of the existing secondary range in the cluster's subnetwork to use for pod IP addresses. Alternatively, `cluster_ipv4_cidr_block` can be used to automatically create a GKE-managed one.

- `services_secondary_range_name` - (Optional) The name of the existing secondary range in the cluster's subnetwork to use for service `ClusterIP`s. Alternatively, `services_ipv4_cidr_block` can be used to automatically create a GKE-managed one.

- `cluster_ipv4_cidr_block` - (Optional) The IP address range for the cluster pod IPs. Set to blank to have a range chosen with the default size. Set to /netmask (e.g. /14) to have a range chosen with a specific netmask. Set to a CIDR notation (e.g. 10.96.0.0/14) from the RFC-1918 private networks (e.g. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) to pick a specific range to use.
- `services_ipv4_cidr_block` - (Optional) The IP address range of the services IPs in this cluster. Set to blank to have a range chosen with the default size. Set to /netmask (e.g. /14) to have a range chosen with a specific netmask. Set to a CIDR notation (e.g. 10.96.0.0/14) from the RFC-1918 private networks (e.g. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) to pick a specific range to use.

The `master_auth` block supports:

- `password` - (Optional) The password to use for HTTP basic authentication when accessing the Kubernetes master endpoint.
- `username` - (Optional) The username to use for HTTP basic authentication when accessing the Kubernetes master endpoint. If not present basic auth will be disabled.
- `client_certificate_config` - (Optional) Whether client certificate authorization is enabled for this cluster. For example:

```
master_auth {
  client_certificate_config {
    issue_client_certificate = false
  }
}
```

If this block is provided and both `username` and `password` are empty, basic authentication will be disabled. This block also contains several computed attributes, documented below. If this block is not provided, GKE will generate a password for you with the username `admin`.

The `master_authorized_networks_config` block supports:

- `cidr_blocks` - (Optional) External networks that can access the Kubernetes cluster 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` - (Required) Whether network policy is enabled on the cluster.

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.
- `disk_type` - (Optional) Type of the disk attached to each node (e.g. 'pd-standard' or 'pd-ssd'). If unspecified, the default disk type is 'pd-standard'
- `guest_accelerator` - (Optional) List of the type and count of accelerator cards attached to the instance. Structure documented below. To support removal of guest_accelerators in Terraform 0.12 this field is an Attribute as Block
- `image_type` - (Optional) The image type to use for this node. Note that changing the image type will delete and recreate all nodes in the node pool.
- `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`. To create a custom machine type, value should be set as specified here.
- `metadata` - (Optional) The metadata key/value pairs assigned to instances in the cluster. From GKE `1.12` onwards, `disable-legacy-endpoints` is set to `true` by the API; if `metadata` is set but that default value is not included, Terraform will attempt to unset the value. To avoid this, set the value in your config.
- `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:
  - `storage-ro` (`https://www.googleapis.com/auth/devstorage.read_only`), if the cluster must read private images from GCR. Note this will grant read access to ALL GCS content unless you also specify a custom role. See https://cloud.google.com/kubernetes-engine/docs/how-to/access-scopes

  - `logging-write` (`https://www.googleapis.com/auth/logging.write`), if `logging_service` is not `none`.
  - `monitoring` (`https://www.googleapis.com/auth/monitoring`), if `monitoring_service` is not `none`.

- `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.

- `sandbox_config` - (Optional, Beta) GKE Sandbox configuration. When enabling this feature you must specify `image_type = "COS_CONTAINERD"` and `node_version = "1.12.7-gke.17"` or later to use it. Structure is documented below.

- `service_account` - (Optional) The service account to be used by the Node VMs. If not specified, the "default" service account is used. In order to use the configured `oauth_scopes` for logging and monitoring, the service account being used needs the roles/logging.logWriter and roles/monitoring.metricWriter roles.

  Projects that enable the Cloud Compute Engine API with Terraform may need these roles added manually to the service account. Projects that enable the API in the Cloud Console should have them added automatically.

- `shielded_instance_config` - (Optional) Shielded Instance options. Structure is documented below.

- `tags` - (Optional) The list of instance tags applied to all nodes. Tags are used to identify valid sources or targets for network firewalls.

- `taint` - (Optional) A list of Kubernetes taints to apply to nodes. GKE's API can only set this field on cluster creation. However, GKE will add taints to your nodes if you enable certain features such as GPUs. If this field is set, any diffs on this field will cause Terraform to recreate the underlying resource. Taint values can be updated safely in Kubernetes (eg. through `kubectl`), and it's recommended that you do not use this field to manage taints. If you do, `lifecycle.ignore_changes` is recommended. Structure is documented below.

- `workload_metadata_config` - (Optional, Beta) 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 `workload_identity_config` block supports:

- `identity_namespace` (Required) - Currently, the only supported identity namespace is the project's default. `hcl workload_identity_config { identity_namespace = "${data.google_project.project.project_id}.svc.id.goog" }`

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 `private_cluster_config` block supports:

- `enable_private_nodes` (Optional) - Enables the private cluster feature, creating a private endpoint on the cluster. In a private cluster, nodes only have RFC 1918 private addresses and communicate with the master's private endpoint via private networking.
- `enable_private_endpoint` (Optional) - When `true`, the cluster's private endpoint is used as the cluster endpoint and access through the public endpoint is disabled. When `false`, either endpoint can be used. This field only applies to private clusters, when `enable_private_nodes` is `true`.
- `master_ipv4_cidr_block` (Optional) - The IP range in CIDR notation to use for the hosted master network. This range will be used for assigning private IP addresses to the cluster master(s) and the ILB VIP. This range must not overlap with any other ranges in use within the cluster's network, and it must be a /28 subnet. See Private Cluster Limitations for more details. This field only applies to private clusters, when `enable_private_nodes` is `true`.

In addition, the `private_cluster_config` allows access to the following read-only fields:

- `peering_name` - The name of the peering between this cluster and the Google owned VPC.
- `private_endpoint` - The internal IP address of this cluster's master endpoint.
- `public_endpoint` - The external IP address of this cluster's master endpoint.

The Google provider is unable to validate certain configurations of `private_cluster_config` when `enable_private_nodes` is `false`. It's recommended that you omit the block entirely if the field is not set to `true`.

The `sandbox_type` block supports:

- `sandbox_type` (Required) Which sandbox to use for pods in the node pool. Accepted values are:
  - `"gvisor"`: Pods run within a gVisor sandbox.

The `release_channel` block supports:

- `channel` - (Required) The selected release channel. Accepted values are:
  - UNSPECIFIED: Not set.
  - RAPID: Weekly upgrade cadence; Early testers and developers who requires new features.
  - REGULAR: Multiple per month upgrade cadence; Production users who need features not yet offered in the Stable channel.
  - STABLE: Every few months upgrade cadence; Production users who need stability above all else, and for whom frequent upgrades are too risky.

The `resource_usage_export_config` block supports:

- `enable_network_egress_metering` (Optional) - Whether to enable network egress metering for this cluster. If enabled, a daemonset will be created in the cluster to meter network egress traffic.
- `bigquery_destination` (Required) - Parameters for using BigQuery as the destination of resource usage export.
- `bigquery_destination.dataset_id` (Required) - The ID of a BigQuery Dataset. For Example:

```
resource_usage_export_config {
  enable_network_egress_metering = false

  bigquery_destination {
    dataset_id = "cluster_resource_usage"
  }
}
```

The `shielded_instance_config` block supports:

- `enable_secure_boot` (Optional) - Defines if the instance has Secure Boot enabled.

Secure Boot helps ensure that the system only runs authentic software by verifying the digital signature of all boot components, and halting the boot process if signature verification fails. Defaults to `false`.

- `enable_integrity_monitoring` (Optional) - Defines if the instance has integrity monitoring enabled.

Enables monitoring and attestation of the boot integrity of the instance. The attestation is performed against the integrity policy baseline. This baseline is initially derived from the implicitly trusted boot image when the instance is created. Defaults to `true`.

The `taint` block supports:

- `key` (Required) Key for taint.

- `value` (Required) Value for taint.
- `effect` (Required) Effect for taint. Accepted values are `NO_SCHEDULE`, `PREFER_NO_SCHEDULE`, and `NO_EXECUTE`.

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.
  - GKE_METADATA_SERVER: Enables workload identity on the node.

The `vertical_pod_autoscaling` block supports:

- `enabled` (Required) - Enables vertical pod autoscaling

» 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.
- `tpu_ipv4_cidr_block` - (Beta) The IP address range of the Cloud TPUs in this cluster, in CIDR notation (e.g. `1.2.3.4/29`).

- `services_ipv4_cidr` - The IP address range of the Kubernetes services in this cluster, in CIDR notation (e.g. `1.2.3.4/29`). Service addresses are typically put in the last `/16` from the container CIDR.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 40 minutes.
- `update` - Default is 60 minutes.
- `delete` - Default is 40 minutes.

» Import

GKE clusters can be imported using the `project` , `location`, and `name`. If the project is omitted, the default provider value will be used. Examples:

```
$ terraform import google_container_cluster.mycluster projects/my-gcp-project/locations/us-e

$ terraform import google_container_cluster.mycluster my-gcp-project/us-east1-a/my-cluster

$ terraform import google_container_cluster.mycluster us-east1-a/my-cluster
```

Note: This resource has several fields that control Terraform-specific behavior and aren't present in the API. If they are set in config and you import a cluster, Terraform may need to perform an update immediately after import. Most of these updates should be no-ops but some may modify your cluster if the imported state differs.

For example, the following fields will show diffs if set in config:

- `min_master_version`
- `remove_default_node_pool`

» google_container_node_pool

Manages a node pool in a Google Kubernetes Engine (GKE) cluster separately from the cluster control plane. For more information see the official documentation and the API reference.

» Example Usage - using a separately managed node pool (recommended)

```
resource "google_container_cluster" "primary" {
```

```
  name     = "my-gke-cluster"
  location = "us-central1"

  # We can't create a cluster with no node pool defined, but we want to only use
  # separately managed node pools. So we create the smallest possible default
  # node pool and immediately delete it.
  remove_default_node_pool = true
  initial_node_count       = 1
}

resource "google_container_node_pool" "primary_preemptible_nodes" {
  name       = "my-node-pool"
  location   = "us-central1"
  cluster    = google_container_cluster.primary.name
  node_count = 1

  node_config {
    preemptible  = true
    machine_type = "n1-standard-1"

    oauth_scopes = [
      "https://www.googleapis.com/auth/logging.write",
      "https://www.googleapis.com/auth/monitoring",
    ]
  }
}
```

» Example Usage - 2 node pools, 1 separately managed + the default node pool

```
resource "google_container_node_pool" "np" {
  name       = "my-node-pool"
  location   = "us-central1-a"
  cluster    = google_container_cluster.primary.name
  node_count = 3

  timeouts {
    create = "30m"
    update = "20m"
  }
}

resource "google_container_cluster" "primary" {
  name               = "marcellus-wallace"
  location           = "us-central1-a"
```

```
  initial_node_count = 3

  node_locations = [
    "us-central1-c",
  ]

  master_auth {
    username = ""
    password = ""

    client_certificate_config {
      issue_client_certificate = false
    }
  }

  node_config {
    oauth_scopes = [
      "https://www.googleapis.com/auth/logging.write",
      "https://www.googleapis.com/auth/monitoring",
    ]

    metadata = {
      disable-legacy-endpoints = "true"
    }

    guest_accelerator {
      type  = "nvidia-tesla-k80"
      count = 1
    }
  }
}
```

» Argument Reference

- `cluster` - (Required) The cluster to create the node pool for. Cluster must be present in `location` provided for zonal clusters.

- `location` - (Optional) The location (region or zone) of the cluster.

- `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 number of nodes for the pool. In regional or multi-zonal clusters, this is the number of nodes per zone. 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.
- `max_pods_per_node` - (Optional, Beta) The maximum number of pods per node in this node pool. Note that this does not work on node pools which are "route-based" - that is, node pools belonging to clusters that do not have IP Aliasing enabled. See the official documentation for more information.
- `node_locations` - (Optional, Beta) The list of zones in which the node pool's nodes should be located. Nodes must be in the region of their regional cluster or in the same region as their cluster's zone for zonal clusters. If unspecified, the cluster-level `node_locations` will be used.

Note: `node_locations` will not revert to the cluster's default set of zones upon being unset. You must manually reconcile the list of zones with your cluster.

- `name` - (Optional) The name of the node pool. If left blank, Terraform will auto-generate a unique 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.
- `upgrade_settings` (Optional, Beta) Specify node upgrade settings to change how many nodes GKE attempts to upgrade at once. The number of nodes upgraded simultaneously is the sum of `max_surge` and `max_unavailable`. The maximum number of nodes upgraded simultaneously is limited to 20.
- `version` - (Optional) The Kubernetes version for the nodes in this pool. Note that if this field and `auto_upgrade` are both specified, they will fight each other for what the node version should be, so setting both is highly discouraged. While a fuzzy version can be specified, it's recommended that you specify explicit versions as Terraform will see spurious diffs when fuzzy versions are used. See the `google_container_engine_versions` data source's `version_prefix` field to approximate fuzzy versions in a Terraform-compatible way.

The `autoscaling` block supports:

- `min_node_count` - (Required) Minimum number of nodes in the NodePool. Must be >=0 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.

The `upgrade_settings` block supports:

- `max_surge` - (Required) The number of additional nodes that can be added to the node pool during an upgrade. Increasing `max_surge` raises the number of nodes that can be upgraded simultaneously. Can be set to 0 or greater.
- `max_unavailable` - (Required) The number of nodes that can be simultaneously unavailable during an upgrade. Increasing `max_unavailable` raises the number of nodes that can be upgraded in parallel. Can be set to 0 or greater.

`max_surge` and `max_unavailable` must not be negative and at least one of them must be greater than zero.

» Timeouts

`google_container_node_pool` provides the following Timeouts configuration options:

- `create` - (Default `30 minutes`) Used for adding node pools
- `update` - (Default `30 minutes`) Used for updates to node pools
- `delete` - (Default `30 minutes`) Used for removing node pools.

» Import

Node pools can be imported using the `project`, `zone`, `cluster` and `name`. If the project is omitted, the default provider value will be used. Examples:

```
$ terraform import google_container_node_pool.mainpool my-gcp-project/us-east1-a/my-cluster/

$ terraform import google_container_node_pool.mainpool us-east1-a/my-cluster/main-pool
```

» google_data_fusion_instance

Represents a Data Fusion instance.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about Instance, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Data Fusion Instance Basic

```
resource "google_data_fusion_instance" "basic_instance" {
  provider = "google-beta"
  name = "my-instance"
  region = "us-central1"
  type = "BASIC"
}
```

» Example Usage - Data Fusion Instance Full

```
resource "google_data_fusion_instance" "extended_instance" {
  provider = "google-beta"
  name = "my-instance"
  description = "My Data Fusion instance"
  region = "us-central1"
  type = "BASIC"
  enable_stackdriver_logging = true
  enable_stackdriver_monitoring = true
  labels = {
    example_key = "example_value"
```

```
  }
  private_instance = true
  network_config {
    network = "default"
    ip_allocation = "10.89.48.0/22"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The ID of the instance or a fully qualified identifier for the instance.
- `type` - (Required) Represents the type of Data Fusion instance. Each type is configured with the default settings for processing and memory.
  - BASIC: Basic Data Fusion instance. In Basic type, the user will be able to create data pipelines using point and click UI. However, there are certain limitations, such as fewer number of concurrent pipelines, no support for streaming pipelines, etc.
  - ENTERPRISE: Enterprise Data Fusion instance. In Enterprise type, the user will have more features available, such as support for streaming pipelines, higher number of concurrent pipelines, etc.

- `description` - (Optional) An optional description of the instance.
- `enable_stackdriver_logging` - (Optional) Option to enable Stackdriver Logging.
- `enable_stackdriver_monitoring` - (Optional) Option to enable Stackdriver Monitoring.
- `labels` - (Optional) The resource labels for instance to use to annotate any related underlying resources, such as Compute Engine VMs.
- `options` - (Optional) Map of additional options used to configure the behavior of Data Fusion instance.
- `private_instance` - (Optional) Specifies whether the Data Fusion instance should be private. If set to true, all Data Fusion nodes will have private IP addresses and will not be able to access the public internet.
- `network_config` - (Optional) Network configuration options. These are required when a private Data Fusion instance is to be created. Structure is documented below.
- `region` - (Optional) The region of the Data Fusion instance.

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

The `network_config` block supports:

- `ip_allocation` - (Required) The IP range in CIDR notation to use for the managed Data Fusion instance nodes. This range must not overlap with any other ranges used in the Data Fusion instance network.
- `network` - (Required) Name of the network in the project with which the tenant project will be peered for executing pipelines. In case of shared VPC where the network resides in another host project the network should specified in the form of projects/{host-project-id}/global/networks/{network}

» Attributes Reference

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

- `create_time` - The time the instance was created in RFC3339 UTC "Zulu" format, accurate to nanoseconds.
- `update_time` - The time the instance was last updated in RFC3339 UTC "Zulu" format, accurate to nanoseconds.
- `state` - The current state of this Data Fusion instance.
  - CREATING: Instance is being created
  - RUNNING: Instance is running and ready for requests
  - FAILED: Instance creation failed
  - DELETING: Instance is being deleted
  - UPGRADING: Instance is being upgraded
  - RESTARTING: Instance is being restarted
- `state_message` - Additional information about the current state of this Data Fusion instance if available.
- `service_endpoint` - Endpoint on which the Data Fusion UI and REST APIs are accessible.
- `version` - Current version of the Data Fusion.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 60 minutes.
- `update` - Default is 25 minutes.

- `delete` - Default is 50 minutes.

» Import

Instance can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_data_fusion_instance.default projects/{{pro
$ terraform import -provider=google-beta google_data_fusion_instance.default {{project}}/{{
$ terraform import -provider=google-beta google_data_fusion_instance.default {{region}}/{{na
$ terraform import -provider=google-beta google_data_fusion_instance.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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 continuously, 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).
- `labels` - (Optional) User labels to be specified for the job. Keys and values should follow the restrictions specified in the labeling restrictions page. **NOTE**: Google-provided Dataflow templates often provide default labels that begin with `goog-dataflow-provided`. Unless explicitly set in config, these labels will be ignored to prevent diffs on re-apply.
- `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.
- `service_account_email` - (Optional) The Service Account email used to create the job.
- `network` - (Optional) The network to which VMs will be assigned. If it is not provided, "default" will be used.

- `subnetwork` - (Optional) The subnetwork to which VMs will be assigned. Should be of the form "regions/REGION/subnetworks/SUBNETWORK".
- `machine_type` - (Optional) The machine type to use for the job.
- `ip_configuration` - (Optional) The configuration for VM IPs. Options are `"WORKER_IP_PUBLIC"` or `"WORKER_IP_PRIVATE"`.

» Attributes Reference

- `state` - The current state of the resource, selected from the JobState enum

» google_dataproc_autoscaling_policy

Describes an autoscaling policy for Dataproc cluster autoscaler.

» Example Usage - Dataproc Autoscaling Policy

```
resource "google_dataproc_cluster" "basic" {
  name     = "tf-dataproc-test-"
  region   = "us-central1"

  cluster_config {
    autoscaling_config {
      policy_uri = google_dataproc_autoscaling_policy.asp.name
    }
  }
}

resource "google_dataproc_autoscaling_policy" "asp" {
  policy_id = "tf-dataproc-test-"
  location  = "us-central1"

  worker_config {
    max_instances = 3
  }

  basic_algorithm {
    yarn_config {
```

```
      graceful_decommission_timeout = "30s"

      scale_up_factor   = 0.5
      scale_down_factor = 0.5
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `policy_id` - (Required) The policy id. The id must contain only letters (a-z, A-Z), numbers (0-9), underscores (_), and hyphens (-). Cannot begin or end with underscore or hyphen. Must consist of between 3 and 50 characters.

- `worker_config` - (Optional) Describes how the autoscaler will operate for primary workers. Structure is documented below.
- `secondary_worker_config` - (Optional) Describes how the autoscaler will operate for secondary workers. Structure is documented below.
- `basic_algorithm` - (Optional) Basic algorithm for autoscaling. Structure is documented below.
- `location` - (Optional) The location where the autoscaling poicy should reside. The default value is `global`.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `worker_config` block supports:

- `min_instances` - (Optional) Minimum number of instances for this group. Bounds: [2, maxInstances]. Defaults to 2.
- `max_instances` - (Required) Maximum number of instances for this group.
- `weight` - (Optional) Weight for the instance group, which is used to determine the fraction of total workers in the cluster from this instance group. For example, if primary workers have weight 2, and secondary workers have weight 1, the cluster will have approximately 2 primary workers for each secondary worker. The cluster may not reach the specified balance if constrained by min/max bounds or other autoscaling settings. For example, if maxInstances for secondary workers is 0, then only primary workers will be added. The cluster can also be out of balance when created. If weight is not set on any instance group, the cluster will default to equal

weight for all groups: the cluster will attempt to maintain an equal number of workers in each group within the configured size bounds for each group. If weight is set for one group only, the cluster will default to zero weight on the unset group. For example if weight is set only on primary workers, the cluster will use primary workers only and no secondary workers.

The `secondary_worker_config` block supports:

- `min_instances` - (Optional) Minimum number of instances for this group. Bounds: [0, maxInstances]. Defaults to 0.
- `max_instances` - (Optional) Maximum number of instances for this group. Note that by default, clusters will not use secondary workers. Required for secondary workers if the minimum secondary instances is set. Bounds: [minInstances,). Defaults to 0.
- `weight` - (Optional) Weight for the instance group, which is used to determine the fraction of total workers in the cluster from this instance group. For example, if primary workers have weight 2, and secondary workers have weight 1, the cluster will have approximately 2 primary workers for each secondary worker. The cluster may not reach the specified balance if constrained by min/max bounds or other autoscaling settings. For example, if maxInstances for secondary workers is 0, then only primary workers will be added. The cluster can also be out of balance when created. If weight is not set on any instance group, the cluster will default to equal weight for all groups: the cluster will attempt to maintain an equal number of workers in each group within the configured size bounds for each group. If weight is set for one group only, the cluster will default to zero weight on the unset group. For example if weight is set only on primary workers, the cluster will use primary workers only and no secondary workers.

The `basic_algorithm` block supports:

- `cooldown_period` - (Optional) Duration between scaling events. A scaling period starts after the update operation from the previous event has completed. Bounds: [2m, 1d]. Default: 2m.
- `yarn_config` - (Required) YARN autoscaling configuration. Structure is documented below.

The `yarn_config` block supports:

- `graceful_decommission_timeout` - (Required) Timeout for YARN graceful decommissioning of Node Managers. Specifies the duration to wait for jobs to complete before forcefully removing workers (and potentially interrupting jobs). Only applicable to downscaling operations. Bounds: [0s, 1d].
- `scale_up_factor` - (Required) Fraction of average pending memory in the last cooldown period for which to add workers. A scale-up factor of 1.0 will result in scaling up so that there is no pending memory remaining

after the update (more aggressive scaling). A scale-up factor closer to 0 will result in a smaller magnitude of scaling up (less aggressive scaling). Bounds: [0.0, 1.0].

- `scale_down_factor` - (Required) Fraction of average pending memory in the last cooldown period for which to remove workers. A scale-down factor of 1 will result in scaling down so that there is no available memory remaining after the update (more aggressive scaling). A scale-down factor of 0 disables removing workers, which can be beneficial for autoscaling a single job. Bounds: [0.0, 1.0].
- `scale_up_min_worker_fraction` - (Optional) Minimum scale-up threshold as a fraction of total cluster size before scaling occurs. For example, in a 20-worker cluster, a threshold of 0.1 means the autoscaler must recommend at least a 2-worker scale-up for the cluster to scale. A threshold of 0 means the autoscaler will scale up on any recommended change. Bounds: [0.0, 1.0]. Default: 0.0.
- `scale_down_min_worker_fraction` - (Optional) Minimum scale-down threshold as a fraction of total cluster size before scaling occurs. For example, in a 20-worker cluster, a threshold of 0.1 means the autoscaler must recommend at least a 2 worker scale-down for the cluster to scale. A threshold of 0 means the autoscaler will scale down on any recommended change. Bounds: [0.0, 1.0]. Default: 0.0.

» Attributes Reference

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

- `name` - The ”resource name” of the autoscaling policy.

» 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

AutoscalingPolicy can be imported using any of these accepted formats:

```
$ terraform import google_dataproc_autoscaling_policy.default projects/{{project}}/locations
$ terraform import google_dataproc_autoscaling_policy.default {{project}}/{{location}}/{{pol
```

```
$ terraform import google_dataproc_autoscaling_policy.default {{location}}/{{policy_id}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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-updatable. Changing others will cause recreation of the whole cluster!

» Example Usage - Basic

```
resource "google_dataproc_cluster" "simplecluster" {
  name   = "simplecluster"
  region = "us-central1"
}
```

» Example Usage - Advanced

```
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_type    = "pd-ssd"
      boot_disk_size_gb = 15
    }
  }

  worker_config {
    num_instances    = 2
    machine_type     = "n1-standard-1"
    min_cpu_platform = "Intel Skylake"
    disk_config {
      boot_disk_size_gb = 15
      num_local_ssds    = 1
    }
  }

  preemptible_worker_config {
    num_instances = 0
  }

  # Override or set some custom properties
  software_config {
    image_version = "1.3.7-deb9"
    override_properties = {
      "dataproc:dataproc.allow.zero.workers" = "true"
    }
  }

  gce_cluster_config {
    tags = ["foo", "bar"]
    service_account_scopes = [
      "https://www.googleapis.com/auth/monitoring",
      "useraccounts-ro",
      "storage-rw",
      "logging-write",
    ]
  }

  # You can define multiple initialization_action blocks
  initialization_action {
    script      = "gs://dataproc-initialization-actions/stackdriver/stackdriver.sh"
    timeout_sec = 500
  }
 }
}
```

» Example Usage - Using a GPU accelerator

```
resource "google_dataproc_cluster" "accelerated_cluster" {
  name   = "my-cluster-with-gpu"
  region = "us-central1"

  cluster_config {
    gce_cluster_config {
      zone = "us-central1-a"
    }

    master_config {
      accelerators {
        accelerator_type  = "nvidia-tesla-k80"
        accelerator_count = "1"
      }
    }
  }
}
```

» 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     { ... }
    encryption_config         { ... }
}
```

- `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.
- `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) instances in a cluster. Structure defined below.
- `software_config` (Optional) The config settings for software inside the cluster. Structure defined below.
- `security_config` (Optional) Security related configuration. Structure defined below.
- `autoscaling_config` (Optional) The autoscaling policy config associated with 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.
- `encryption_config` (Optional) The Customer managed encryption keys settings for the cluster. 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 must be set if any other scopes are set. They're necessary to ensure the correct functioning ofthe cluster, and are set automatically by the API:
  - `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.

- `metadata` - (Optional) A map of the Compute Engine metadata entries to add to all instances (see Project and instance metadata).

The `cluster_config.master_config` block supports:

```
cluster_config {
  master_config {
    num_instances    = 1
    machine_type     = "n1-standard-1"
    min_cpu_platform = "Intel Skylake"

    disk_config {
      boot_disk_type    = "pd-ssd"
      boot_disk_size_gb = 15
      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`).
- `min_cpu_platform` - (Optional, Computed) The name of a minimum generation of CPU family for the master. If not specified, GCP will default to a predetermined computed value for each zone. See the guide for details about which CPU families are available (and defaulted) for each zone.
- `image_uri` (Optional) The URI for the image to use for this worker. See the guide for more information.
- `disk_config` (Optional) Disk Config
  - `boot_disk_type` - (Optional) The disk type of the primary disk attached to each node. One of `"pd-ssd"` or `"pd-standard"`. Defaults to `"pd-standard"`.
  - `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.
  - `num_local_ssds` - (Optional) The amount of local SSD disks that will be attached to each master cluster node. Defaults to 0.

- `accelerators` (Optional) The Compute Engine accelerator (GPU) configuration for these instances. Can be specified multiple times.
  - `accelerator_type` - (Required) The short name of the accelerator type to expose to this instance. For example, `nvidia-tesla-k80`.
  - `accelerator_count` - (Required) The number of the accelerator cards of this type exposed to this instance. Often restricted to one of `1`, `2`, `4`, or `8`.

The Cloud Dataproc API can return unintuitive error messages when using accelerators; even when you have defined an accelerator, Auto Zone Placement does not exclusively select zones that have that accelerator available. If you get a 400 error that the accelerator can't be found, this is a likely cause. Make sure you check accelerator availability by zone if you are trying to use accelerators in a given zone.

The `cluster_config.worker_config` block supports:

```
cluster_config {
  worker_config {
    num_instances    = 3
    machine_type     = "n1-standard-1"
    min_cpu_platform = "Intel Skylake"

    disk_config {
      boot_disk_type    = "pd-standard"
      boot_disk_size_gb = 15
      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`).
- `min_cpu_platform` - (Optional, Computed) The name of a minimum generation of CPU family for the master. If not specified, GCP will default to a predetermined computed value for each zone. See the guide for details about which CPU families are available (and defaulted) for each zone.

- `disk_config` (Optional) Disk Config
  - `boot_disk_type` - (Optional) The disk type of the primary disk attached to each node. One of `"pd-ssd"` or `"pd-standard"`. Defaults to `"pd-standard"`.
  - `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.
- `image_uri` (Optional) The URI for the image to use for this worker. See the guide for more information.
- `accelerators` (Optional) The Compute Engine accelerator configuration for these instances. Can be specified multiple times.
  - `accelerator_type` - (Required) The short name of the accelerator type to expose to this instance. For example, `nvidia-tesla-k80`.
  - `accelerator_count` - (Required) The number of the accelerator cards of this type exposed to this instance. Often restricted to one of `1`, `2`, `4`, or `8`.

The Cloud Dataproc API can return unintuitive error messages when using accelerators; even when you have defined an accelerator, Auto Zone Placement does not exclusively select zones that have that accelerator available. If you get a 400 error that the accelerator can't be found, this is a likely cause. Make sure you check accelerator availability by zone if you are trying to use accelerators in a given zone.

The `cluster_config.preemptible_worker_config` block supports:

```
cluster_config {
  preemptible_worker_config {
    num_instances = 1

    disk_config {
      boot_disk_type    = "pd-standard"
      boot_disk_size_gb = 15
      num_local_ssds    = 1
    }
  }
}
```

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_type` - (Optional) The disk type of the primary disk attached to each preemptible worker node. One of `"pd-ssd"` or `"pd-standard"`. Defaults to `"pd-standard"`.
  - `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.
  - `num_local_ssds` - (Optional) The amount of local SSD disks that will be attached to each preemptible worker node. Defaults to 0.

The `cluster_config.software_config` block supports:

```
cluster_config {
  # Override or set some custom properties
  software_config {
    image_version = "1.3.7-deb9"

    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 `cluster_config.security_config` block supports:

```
cluster_config {
  # Override or set some custom properties
```

```
  security_config {
    kerberos_config {
      kms_key_uri = "projects/projectId/locations/locationId/keyRings/keyRingId/cryptoKeys/k
      root_principal_password_uri = "bucketId/o/objectId"
    }
  }
}
```

- `kerberos_config` (Required) Kerberos Configuration
  - `cross_realm_trust_admin_server` - (Optional) The admin server (IP or hostname) for the remote trusted realm in a cross realm trust relationship.
  - `cross_realm_trust_kdc` - (Optional) The KDC (IP or hostname) for the remote trusted realm in a cross realm trust relationship.
  - `cross_realm_trust_realm` - (Optional) The remote realm the Dataproc on-cluster KDC will trust, should the user enable cross realm trust.
  - `cross_realm_trust_shared_password_uri` - (Optional) The Cloud Storage URI of a KMS encrypted file containing the shared password between the on-cluster Kerberos realm and the remote trusted realm, in a cross realm trust relationship.
  - `enable_kerberos` - (Optional) Flag to indicate whether to Kerberize the cluster.
  - `kdc_db_key_uri` - (Optional) The Cloud Storage URI of a KMS encrypted file containing the master key of the KDC database.
  - `key_password_uri` - (Optional) The Cloud Storage URI of a KMS encrypted file containing the password to the user provided key. For the self-signed certificate, this password is generated by Dataproc.
  - `keystore_uri` - (Optional) The Cloud Storage URI of the keystore file used for SSL encryption. If not provided, Dataproc will provide a self-signed certificate.
  - `keystore_password_uri` - (Optional) The Cloud Storage URI of a KMS encrypted file containing the password to the user provided keystore. For the self-signed certificated, the password is generated by Dataproc.
  - `kms_key_uri` - (Required) The URI of the KMS key used to encrypt various sensitive files.
  - `realm` - (Optional) The name of the on-cluster Kerberos realm. If not specified, the uppercased domain of hostnames will be the realm.
  - `root_principal_password_uri` - (Required) The Cloud Storage URI of a KMS encrypted file containing the root principal password.
  - `tgt_lifetime_hours` - (Optional) The lifetime of the ticket granting ticket, in hours.
  - `truststore_password_uri` - (Optional) The Cloud Storage URI of a KMS encrypted file containing the password to the user provided truststore. For the self-signed certificate, this password is generated

by Dataproc.
  - `truststore_uri` - (Optional) The Cloud Storage URI of the truststore file used for SSL encryption. If not provided, Dataproc will provide a self-signed certificate.

The `cluster_config.autoscaling_config` block supports:

```
cluster_config {
  # Override or set some custom properties
  autoscaling_config {
    policy_uri = "projects/projectId/locations/region/autoscalingPolicies/policyId"
  }
}
```

- `policy_uri` - (Required) The autoscaling policy used by the cluster.

Only resource names including projectid and location (region) are valid. Examples:

`https://www.googleapis.com/compute/v1/projects/[projectId]/locations/[dataproc_region]/autos`
`projects/[projectId]/locations/[dataproc_region]/autoscalingPolicies/[policy_id]`
Note that the policy must be in the same project and Cloud Dataproc region.

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).

The `encryption_config` block supports:

```
cluster_config {
  encryption_config {
    kms_key_name = "projects/projectId/locations/region/keyRings/keyRingName/cryptoKeys/keyN
  }
```

}
```

- `kms_key_name` - (Required) The Cloud KMS key name to use for PD disk encryption for all instances in the cluster.

### » Attributes Reference

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

- `cluster_config.0.master_config.0.instance_names` - List of master instance names which have been assigned to the cluster.
- `cluster_config.0.worker_config.0.instance_names` - List of worker instance names which have been assigned to the cluster.
- `cluster_config.0.preemptible_worker_config.0.instance_names` - List of preemptible instance names which have been assigned to the cluster.
- `cluster_config.0.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.0.software_config.0.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

This resource provides the following Timeouts configuration options:

- `create` - Default is 20 minutes.
- `update` - Default is 20 minutes.
- `delete` - Default is 20 minutes.

## » IAM policy for Dataproc cluster

Three different resources help you manage IAM policies on dataproc clusters. Each of these resources serves a different use case:

- `google_dataproc_cluster_iam_policy`: Authoritative. Sets the IAM policy for the cluster and replaces any existing policy already attached.
```

- `google_dataproc_cluster_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 cluster are preserved.
- `google_dataproc_cluster_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the cluster are preserved.

Note: `google_dataproc_cluster_iam_policy` **cannot** be used in conjunction with `google_dataproc_cluster_iam_binding` and `google_dataproc_cluster_iam_member` or they will fight over what your policy should be. In addition, be careful not to accidentally unset ownership of the cluster as `google_dataproc_cluster_iam_policy` replaces the entire policy.

Note: `google_dataproc_cluster_iam_binding` resources **can be** used in conjunction with `google_dataproc_cluster_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_dataproc_cluster_iam_policy" "editor" {
  project     = "your-project"
  region      = "your-region"
  cluster     = "your-dataproc-cluster"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_pubsub_subscription_iam_binding

```
resource "google_dataproc_cluster_iam_binding" "editor" {
  cluster = "your-dataproc-cluster"
  role    = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_pubsub_subscription_iam_member

```
resource "google_dataproc_cluster_iam_member" "editor" {
  cluster = "your-dataproc-cluster"
  role    = "roles/editor"
  member  = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `cluster` - (Required) The name or relative resource id of the cluster to manage IAM policies for.

For `google_dataproc_cluster_iam_member` or `google_dataproc_cluster_iam_binding`:

- `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 G Suite domain (primary, instead of alias) 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_dataproc_cluster_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

`google_dataproc_cluster_iam_policy` only: * `policy_data` - (Required) The policy data generated by a `google_iam_policy` data source.

- `project` - (Optional) The project in which the cluster belongs. If it is not provided, Terraform will use the provider default.
- `region` - (Optional) The region in which the cluster belongs. If it is not provided, Terraform will use the provider default.

» Attributes Reference

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

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

» Import

Cluster IAM resources can be imported using the project, region, cluster name, role and/or member.

```
$ terraform import google_dataproc_cluster_iam_policy.editor "projects/{project}/regions/{re

$ terraform import google_dataproc_cluster_iam_binding.editor "projects/{project}/regions/{r

$ terraform import google_dataproc_cluster_iam_member.editor "projects/{project}/regions/{re
```

» IAM policy for Dataproc cluster

Three different resources help you manage IAM policies on dataproc clusters. Each of these resources serves a different use case:

- `google_dataproc_cluster_iam_policy`: Authoritative. Sets the IAM policy for the cluster and replaces any existing policy already attached.
- `google_dataproc_cluster_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 cluster are preserved.
- `google_dataproc_cluster_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the cluster are preserved.

Note: `google_dataproc_cluster_iam_policy` **cannot** be used in conjunction with `google_dataproc_cluster_iam_binding` and `google_dataproc_cluster_iam_member` or they will fight over what your policy should be. In addition, be careful not to accidentally unset ownership of the cluster as `google_dataproc_cluster_iam_policy` replaces the entire policy.

Note: `google_dataproc_cluster_iam_binding` resources **can be** used in conjunction with `google_dataproc_cluster_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_dataproc_cluster_iam_policy" "editor" {
  project     = "your-project"
  region      = "your-region"
  cluster     = "your-dataproc-cluster"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_pubsub_subscription_iam_binding

```
resource "google_dataproc_cluster_iam_binding" "editor" {
  cluster = "your-dataproc-cluster"
  role    = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_pubsub_subscription_iam_member

```
resource "google_dataproc_cluster_iam_member" "editor" {
  cluster = "your-dataproc-cluster"
  role    = "roles/editor"
  member  = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `cluster` - (Required) The name or relative resource id of the cluster to manage IAM policies for.

For `google_dataproc_cluster_iam_member` or `google_dataproc_cluster_iam_binding`:

- `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 G Suite domain (primary, instead of alias) 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_dataproc_cluster_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

`google_dataproc_cluster_iam_policy` only: * `policy_data` - (Required) The policy data generated by a `google_iam_policy` data source.

- `project` - (Optional) The project in which the cluster belongs. If it is not provided, Terraform will use the provider default.
- `region` - (Optional) The region in which the cluster belongs. If it is not provided, Terraform will use the provider default.

» Attributes Reference

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

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

» Import

Cluster IAM resources can be imported using the project, region, cluster name, role and/or member.

```
$ terraform import google_dataproc_cluster_iam_policy.editor "projects/{project}/regions/{re

$ terraform import google_dataproc_cluster_iam_binding.editor "projects/{project}/regions/{r

$ terraform import google_dataproc_cluster_iam_member.editor "projects/{project}/regions/{re
```

» IAM policy for Dataproc cluster

Three different resources help you manage IAM policies on dataproc clusters. Each of these resources serves a different use case:

- `google_dataproc_cluster_iam_policy`: Authoritative. Sets the IAM policy for the cluster and replaces any existing policy already attached.
- `google_dataproc_cluster_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 cluster are preserved.
- `google_dataproc_cluster_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the cluster are preserved.

Note: `google_dataproc_cluster_iam_policy` **cannot** be used in conjunction with `google_dataproc_cluster_iam_binding` and `google_dataproc_cluster_iam_member` or they will fight over what your policy should be. In addition, be careful not to accidentally unset ownership of the cluster as `google_dataproc_cluster_iam_policy` replaces the entire policy.

Note: `google_dataproc_cluster_iam_binding` resources **can be** used in conjunction with `google_dataproc_cluster_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_dataproc_cluster_iam_policy" "editor" {
  project     = "your-project"
  region      = "your-region"
  cluster     = "your-dataproc-cluster"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_pubsub_subscription_iam_binding

```
resource "google_dataproc_cluster_iam_binding" "editor" {
```

```
  cluster = "your-dataproc-cluster"
  role    = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_pubsub_subscription_iam_member

```
resource "google_dataproc_cluster_iam_member" "editor" {
  cluster = "your-dataproc-cluster"
  role    = "roles/editor"
  member  = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `cluster` - (Required) The name or relative resource id of the cluster to manage IAM policies for.

For `google_dataproc_cluster_iam_member` or `google_dataproc_cluster_iam_binding`:

- `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 G Suite domain (primary, instead of alias) 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_dataproc_cluster_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

`google_dataproc_cluster_iam_policy` only: * `policy_data` - (Required) The policy data generated by a `google_iam_policy` data source.

- `project` - (Optional) The project in which the cluster belongs. If it is not provided, Terraform will use the provider default.
- `region` - (Optional) The region in which the cluster belongs. If it is not provided, Terraform will use the provider default.

» Attributes Reference

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

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

» Import

Cluster IAM resources can be imported using the project, region, cluster name, role and/or member.

```
$ terraform import google_dataproc_cluster_iam_policy.editor "projects/{project}/regions/{re

$ terraform import google_dataproc_cluster_iam_binding.editor "projects/{project}/regions/{r

$ terraform import google_dataproc_cluster_iam_member.editor "projects/{project}/regions/{re
```

» 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/pysp
    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` - (Required) 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/pysp
    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`- (Required) 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`- (Required) 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`- (Required) 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_cluster
      "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`- (Required) 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`- (Required) 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.

» IAM policy for Dataproc job

Three different resources help you manage IAM policies on dataproc jobs. Each of these resources serves a different use case:

- `google_dataproc_job_iam_policy`: Authoritative. Sets the IAM policy for the job and replaces any existing policy already attached.

- `google_dataproc_job_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 job are preserved.
- `google_dataproc_job_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the job are preserved.

Note: `google_dataproc_job_iam_policy` **cannot** be used in conjunction with `google_dataproc_job_iam_binding` and `google_dataproc_job_iam_member` or they will fight over what your policy should be. In addition, be careful not to accidentally unset ownership of the job as `google_dataproc_job_iam_policy` replaces the entire policy.

Note: `google_dataproc_job_iam_binding` resources **can be** used in conjunction with `google_dataproc_job_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_dataproc_job_iam_policy" "editor" {
  project     = "your-project"
  region      = "your-region"
  job_id      = "your-dataproc-job"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_pubsub_subscription_iam_binding

```
resource "google_dataproc_job_iam_binding" "editor" {
  job_id = "your-dataproc-job"
  role   = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_pubsub_subscription_iam_member

```
resource "google_dataproc_job_iam_member" "editor" {
  job_id = "your-dataproc-job"
  role   = "roles/editor"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `job` - (Required) The name or relative resource id of the job to manage IAM policies for.

For `google_dataproc_job_iam_member` or `google_dataproc_job_iam_binding`:

- `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 G Suite domain (primary, instead of alias) 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_dataproc_job_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

`google_dataproc_job_iam_policy` only: * `policy_data` - (Required) The policy data generated by a `google_iam_policy` data source.

- `project` - (Optional) The project in which the job belongs. If it is not provided, Terraform will use the provider default.
- `region` - (Optional) The region in which the job belongs. If it is not provided, Terraform will use the provider default.

» Attributes Reference

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

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

» Import

Job IAM resources can be imported using the project, region, job id, role and/or member.

```
$ terraform import google_dataproc_job_iam_policy.editor "projects/{project}/regions/{regio

$ terraform import google_dataproc_job_iam_binding.editor "projects/{project}/regions/{regi

$ terraform import google_dataproc_job_iam_member.editor "projects/{project}/regions/{regio
```

» IAM policy for Dataproc job

Three different resources help you manage IAM policies on dataproc jobs. Each of these resources serves a different use case:

- `google_dataproc_job_iam_policy`: Authoritative. Sets the IAM policy for the job and replaces any existing policy already attached.
- `google_dataproc_job_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 job are preserved.
- `google_dataproc_job_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the job are preserved.

Note: `google_dataproc_job_iam_policy` **cannot** be used in conjunction with `google_dataproc_job_iam_binding` and `google_dataproc_job_iam_member` or they will fight over what your policy should be. In addition, be careful not to accidentally unset ownership of the job as `google_dataproc_job_iam_policy` replaces the entire policy.

Note: `google_dataproc_job_iam_binding` resources **can be** used in conjunction with `google_dataproc_job_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_dataproc_job_iam_policy" "editor" {
  project     = "your-project"
  region      = "your-region"
  job_id      = "your-dataproc-job"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_pubsub_subscription_iam_binding

```
resource "google_dataproc_job_iam_binding" "editor" {
  job_id = "your-dataproc-job"
  role   = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_pubsub_subscription_iam_member

```
resource "google_dataproc_job_iam_member" "editor" {
  job_id = "your-dataproc-job"
  role   = "roles/editor"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `job` - (Required) The name or relative resource id of the job to manage IAM policies for.

For `google_dataproc_job_iam_member` or `google_dataproc_job_iam_binding`:

- `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 G Suite domain (primary, instead of alias) 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_dataproc_job_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

`google_dataproc_job_iam_policy` only: * `policy_data` - (Required) The policy data generated by a `google_iam_policy` data source.

- `project` - (Optional) The project in which the job belongs. If it is not provided, Terraform will use the provider default.
- `region` - (Optional) The region in which the job belongs. If it is not provided, Terraform will use the provider default.

» Attributes Reference

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

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

» Import

Job IAM resources can be imported using the project, region, job id, role and/or member.

```
$ terraform import google_dataproc_job_iam_policy.editor "projects/{project}/regions/{regio

$ terraform import google_dataproc_job_iam_binding.editor "projects/{project}/regions/{regi

$ terraform import google_dataproc_job_iam_member.editor "projects/{project}/regions/{regio
```

» IAM policy for Dataproc job

Three different resources help you manage IAM policies on dataproc jobs. Each of these resources serves a different use case:

- `google_dataproc_job_iam_policy`: Authoritative. Sets the IAM policy for the job and replaces any existing policy already attached.
- `google_dataproc_job_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 job are preserved.
- `google_dataproc_job_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the job are preserved.

Note: `google_dataproc_job_iam_policy` **cannot** be used in conjunction with `google_dataproc_job_iam_binding` and `google_dataproc_job_iam_member` or they will fight over what your policy should be. In addition, be careful not to accidentally unset ownership of the job as `google_dataproc_job_iam_policy` replaces the entire policy.

Note: `google_dataproc_job_iam_binding` resources **can be** used in conjunction with `google_dataproc_job_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_dataproc_job_iam_policy" "editor" {
  project     = "your-project"
  region      = "your-region"
  job_id      = "your-dataproc-job"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_pubsub_subscription_iam_binding

```
resource "google_dataproc_job_iam_binding" "editor" {
```

```
  job_id = "your-dataproc-job"
  role   = "roles/editor"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_pubsub_subscription_iam_member

```
resource "google_dataproc_job_iam_member" "editor" {
  job_id = "your-dataproc-job"
  role   = "roles/editor"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `job` - (Required) The name or relative resource id of the job to manage IAM policies for.

For `google_dataproc_job_iam_member` or `google_dataproc_job_iam_binding`:

- `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 G Suite domain (primary, instead of alias) 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_dataproc_job_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role-name}`.

`google_dataproc_job_iam_policy` only: * `policy_data` - (Required) The policy data generated by a `google_iam_policy` data source.

- `project` - (Optional) The project in which the job belongs. If it is not provided, Terraform will use the provider default.
- `region` - (Optional) The region in which the job belongs. If it is not provided, Terraform will use the provider default.

» Attributes Reference

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

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

» Import

Job IAM resources can be imported using the project, region, job id, role and/or member.

```
$ terraform import google_dataproc_job_iam_policy.editor "projects/{project}/regions/{regio

$ terraform import google_dataproc_job_iam_binding.editor "projects/{project}/regions/{regi

$ terraform import google_dataproc_job_iam_member.editor "projects/{project}/regions/{regio
```

» google_dns_managed_zone

A zone is a subtree of the DNS namespace under one administrative responsibility. A ManagedZone is a resource that represents a DNS zone hosted by the Cloud DNS service.

To get more information about ManagedZone, see:

- API documentation
- How-to Guides
  - Managing Zones

» Example Usage - Dns Managed Zone Basic

```
resource "google_dns_managed_zone" "example-zone" {
  name        = "example-zone"
  dns_name    = "example-${random_id.rnd.hex}.com."
  description = "Example DNS zone"
  labels = {
    foo = "bar"
  }
}

resource "random_id" "rnd" {
  byte_length = 4
}
```

» Example Usage - Dns Managed Zone Private

```
resource "google_dns_managed_zone" "private-zone" {
  name        = "private-zone"
  dns_name    = "private.example.com."
  description = "Example private DNS zone"
  labels = {
    foo = "bar"
  }

  visibility = "private"

  private_visibility_config {
    networks {
      network_url = google_compute_network.network-1.self_link
    }
    networks {
      network_url = google_compute_network.network-2.self_link
    }
  }
}

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

```
  auto_create_subnetworks = false
}

resource "google_compute_network" "network-2" {
  name                    = "network-2"
  auto_create_subnetworks = false
}
```

» Example Usage - Dns Managed Zone Private Forwarding

```
resource "google_dns_managed_zone" "private-zone" {
  provider    = google-beta
  name        = "private-zone"
  dns_name    = "private.example.com."
  description = "Example private DNS zone"
  labels = {
    foo = "bar"
  }

  visibility = "private"

  private_visibility_config {
    networks {
      network_url = google_compute_network.network-1.self_link
    }
    networks {
      network_url = google_compute_network.network-2.self_link
    }
  }

  forwarding_config {
    target_name_servers {
      ipv4_address = "172.16.1.10"
    }
    target_name_servers {
      ipv4_address = "172.16.1.20"
    }
  }
}

resource "google_compute_network" "network-1" {
  name                    = "network-1"
  auto_create_subnetworks = false
}
```

```
resource "google_compute_network" "network-2" {
  name                    = "network-2"
  auto_create_subnetworks = false
}
```

» Example Usage - Dns Managed Zone Private Peering

```
resource "google_dns_managed_zone" "peering-zone" {
  provider = google-beta

  name        = "peering-zone"
  dns_name    = "peering.example.com."
  description = "Example private DNS peering zone"

  visibility = "private"

  private_visibility_config {
    networks {
      network_url = google_compute_network.network-source.self_link
    }
  }

  peering_config {
    target_network {
      network_url = google_compute_network.network-target.self_link
    }
  }
}

resource "google_compute_network" "network-source" {
  provider = google-beta

  name                    = "network-source"
  auto_create_subnetworks = false
}

resource "google_compute_network" "network-target" {
  provider = google-beta

  name                    = "network-target"
```

```
  auto_create_subnetworks = false
}

provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}
```

» Argument Reference

The following arguments are supported:

- `dns_name` - (Required) The DNS name of this managed zone, for instance "example.com.".
- `name` - (Required) User assigned name for this resource. Must be unique within the project.

- `description` - (Optional) A textual description field. Defaults to 'Managed by Terraform'.
- `dnssec_config` - (Optional) DNSSEC configuration Structure is documented below.
- `labels` - (Optional) A set of key/value label pairs to assign to this ManagedZone.
- `visibility` - (Optional) The zone's visibility: public zones are exposed to the Internet, while private zones are visible only to Virtual Private Cloud resources. Must be one of: `public`, `private`.
- `private_visibility_config` - (Optional) For privately visible zones, the set of Virtual Private Cloud resources that the zone is visible from. Structure is documented below.
- `forwarding_config` - (Optional, Beta) The presence for this field indicates that outbound forwarding is enabled for this zone. The value of this field contains the set of destinations to forward to. Structure is documented below.
- `peering_config` - (Optional, Beta) The presence of this field indicates that DNS Peering is enabled for this zone. The value of this field contains the network to peer with. 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.

The `dnssec_config` block supports:

- `kind` - (Optional) Identifies what kind of resource this is
- `non_existence` - (Optional) Specifies the mechanism used to provide authenticated denial-of-existence responses.
- `state` - (Optional) Specifies whether DNSSEC is enabled, and what mode it is in
- `default_key_specs` - (Optional) Specifies parameters that will be used for generating initial DnsKeys for this ManagedZone. If you provide a spec for keySigning or zoneSigning, you must also provide one for the other. Structure is documented below.

The `default_key_specs` block supports:

- `algorithm` - (Optional) String mnemonic specifying the DNSSEC algorithm of this key
- `key_length` - (Optional) Length of the keys in bits
- `key_type` - (Optional) Specifies whether this is a key signing key (KSK) or a zone signing key (ZSK). Key signing keys have the Secure Entry Point flag set and, when active, will only be used to sign resource record sets of type DNSKEY. Zone signing keys do not have the Secure Entry Point flag set and will be used to sign all other types of resource record sets.
- `kind` - (Optional) Identifies what kind of resource this is

The `private_visibility_config` block supports:

- `networks` - (Required) The list of VPC networks that can see this zone. Until the provider updates to use the Terraform 0.12 SDK in a future release, you may experience issues with this resource while updating. If you've defined a `networks` block and add another `networks` block while keeping the old block, Terraform will see an incorrect diff and apply an incorrect update to the resource. If you encounter this issue, remove all `networks` blocks in an update and then apply another update adding all of them back simultaneously. Structure is documented below.

The `networks` block supports:

- `network_url` - (Required) The fully qualified URL of the VPC network to bind to. This should be formatted like `https://www.googleapis.com/compute/v1/projects/{project`

The `forwarding_config` block supports:

- `target_name_servers` - (Required) List of target name servers to forward to. Cloud DNS will select the best available name server if more than one target is given. Structure is documented below.

The `target_name_servers` block supports:

- `ipv4_address` - (Required) IPv4 address of a target name server.

The `peering_config` block supports:

- `target_network` - (Required) The network with which to peer. Structure is documented below.

The `target_network` block supports:

- `network_url` - (Required) The fully qualified URL of the VPC network to forward queries to. This should be formatted like `https://www.googleapis.com/compute/v1/projects/{project}/global/networks/{network}`

» Attributes Reference

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

- `name_servers` - Delegate your managed_zone to these virtual name servers; defined by the server

» 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

ManagedZone can be imported using any of these accepted formats:

```
$ terraform import google_dns_managed_zone.default projects/{{project}}/managedZones/{{name}
$ terraform import google_dns_managed_zone.default {{project}}/{{name}}
$ terraform import google_dns_managed_zone.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_dns_policy

A policy is a collection of DNS rules applied to one or more Virtual Private Cloud resources.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about Policy, see:

- API documentation
- How-to Guides
  - Using DNS server policies

» Example Usage - Dns Policy Basic

```
resource "google_dns_policy" "example-policy" {
  provider = google-beta

  name                      = "example-policy"
  enable_inbound_forwarding = true

  enable_logging = true

  alternative_name_server_config {
    target_name_servers {
      ipv4_address = "172.16.1.10"
    }
    target_name_servers {
      ipv4_address = "172.16.1.20"
    }
  }

  networks {
    network_url = google_compute_network.network-1.self_link
  }
  networks {
    network_url = google_compute_network.network-2.self_link
  }
}
```

```
resource "google_compute_network" "network-1" {
  provider = google-beta

  name                    = "network-1"
  auto_create_subnetworks = false
}

resource "google_compute_network" "network-2" {
  provider = google-beta

  name                    = "network-2"
  auto_create_subnetworks = false
}

provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) User assigned name for this policy.

- `alternative_name_server_config` - (Optional) Sets an alternative name server for the associated networks. When specified, all DNS queries are forwarded to a name server that you choose. Names such as .internal are not available when an alternative name server is specified. Structure is documented below.
- `description` - (Optional) A textual description field. Defaults to 'Managed by Terraform'.
- `enable_inbound_forwarding` - (Optional) Allows networks bound to this policy to receive DNS queries sent by VMs or applications over VPN connections. When enabled, a virtual IP address will be allocated from each of the sub-networks that are bound to this policy.
- `enable_logging` - (Optional) Controls whether logging is enabled for the networks bound to this policy. Defaults to no logging if not set.
- `networks` - (Optional) List of network names specifying networks to which this policy is applied. 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.

The `alternative_name_server_config` block supports:

- `target_name_servers` - (Required) Sets an alternative name server for the associated networks. When specified, all DNS queries are forwarded to a name server that you choose. Names such as .internal are not available when an alternative name server is specified. Structure is documented below.

The `target_name_servers` block supports:

- `ipv4_address` - (Required) IPv4 address to forward to.

The `networks` block supports:

- `network_url` - (Required) The fully qualified URL of the VPC network to bind to. This should be formatted like `https://www.googleapis.com/compute/v1/projects/{project`

» 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

Policy can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_dns_policy.default projects/{{project}}/poli
$ terraform import -provider=google-beta google_dns_policy.default {{project}}/{{name}}
$ terraform import -provider=google-beta google_dns_policy.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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 provider treats this resource as an authoritative record set. This means existing records (including the default records) for the given type will be overwritten when you create this resource in Terraform. In addition, the Google Cloud DNS API requires NS records to be present at all times, so Terraform will not actually remove NS records during destroy 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].nat_ip]
}

resource "google_compute_instance" "frontend" {
  name         = "frontend"
  machine_type = "g1-small"
  zone         = "us-central1-b"

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-9"
    }
  }

  network_interface {
    network = "default"
    access_config {
    }
  }
}

resource "google_dns_managed_zone" "prod" {
  name     = "prod-zone"
```

```
  dns_name = "prod.mydomain.com."
}
```

» Adding an A record

```
resource "google_dns_record_set" "a" {
  name         = "backend.${google_dns_managed_zone.prod.dns_name}"
  managed_zone = google_dns_managed_zone.prod.name
  type         = "A"
  ttl          = 300

  rrdatas = ["8.8.8.8"]
}

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

» Adding an MX record

```
resource "google_dns_record_set" "mx" {
  name         = google_dns_managed_zone.prod.dns_name
  managed_zone = google_dns_managed_zone.prod.name
  type         = "MX"
  ttl          = 3600

  rrdatas = [
    "1 aspmx.l.google.com.",
    "5 alt1.aspmx.l.google.com.",
    "5 alt2.aspmx.l.google.com.",
    "10 alt3.aspmx.l.google.com.",
    "10 alt4.aspmx.l.google.com.",
  ]
}

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

» Adding an SPF record

Quotes ("") 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."
}
```

» Adding a CNAME record

The list of `rrdatas` should only contain a single string corresponding to the Canonical Name intended.

```
resource "google_dns_record_set" "cname" {
  name         = "frontend.${google_dns_managed_zone.prod.dns_name}"
  managed_zone = google_dns_managed_zone.prod.name
  type         = "CNAME"
  ttl          = 300
  rrdatas      = ["frontend.mydomain.com."]
}

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. To specify a single record value longer than 255 characters such as a TXT record for DKIM, add `\"\"` inside the Terraform configuration string (e.g. `"first255characters\"\"morecharacters"`).
- `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 sets can be imported using either of these accepted formats:

```
$ terraform import google_dns_record_set.frontend {{project}}/{{zone}}/{{name}}/{{type}}
$ terraform import google_dns_record_set.frontend {{zone}}/{{name}}/{{type}}
```

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_base64 = base64encode(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_base64` 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_base64`. `open_api` config must *not* be provided.
- `protoc_output_base64`: (Optional) The full contents of the Service Descriptor File generated by protoc. This should be a compiled .pb file, base64-encoded.
- `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_filestore_instance

A Google Cloud Filestore instance.

To get more information about Instance, see:

- API documentation
- How-to Guides
  - Official Documentation
  - Use with Kubernetes
  - Copying Data In/Out

» Example Usage - Filestore Instance Basic

```
resource "google_filestore_instance" "instance" {
  name = "test-instance"
  zone = "us-central1-b"
```

```
  tier = "PREMIUM"

  file_shares {
    capacity_gb = 2660
    name        = "share1"
  }

  networks {
    network = "default"
    modes   = ["MODE_IPV4"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource name of the instance.
- `tier` - (Required) The service tier of the instance.
- `file_shares` - (Required) File system shares on the instance. For this version, only a single file share is supported. Structure is documented below.
- `networks` - (Required) VPC networks to which the instance is connected. For this version, only a single network is supported. Structure is documented below.
- `zone` - (Required) The name of the Filestore zone of the instance.

The `file_shares` block supports:

- `name` - (Required) The name of the fileshare (16 characters or less)
- `capacity_gb` - (Required) File share capacity in GiB. This must be at least 1024 GiB for the standard tier, or 2560 GiB for the premium tier.

The `networks` block supports:

- `network` - (Required) The name of the GCE VPC network to which the instance is connected.
- `modes` - (Required) IP versions for which the instance has IP addresses assigned.
- `reserved_ip_range` - (Optional) A /29 CIDR block that identifies the range of IP addresses reserved for this instance.
- `ip_addresses` - A list of IPv4 or IPv6 addresses.

- `description` - (Optional) A description of the instance.
- `labels` - (Optional) Resource labels to represent user-provided metadata.
- `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:

- `create_time` - Creation timestamp in RFC3339 text format.
- `etag` - Server-specified ETag for the instance resource to prevent simultaneous updates from overwriting each other.

» 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 can be imported using any of these accepted formats:

```
$ terraform import google_filestore_instance.default projects/{{project}}/locations/{{zone}}
$ terraform import google_filestore_instance.default {{project}}/{{zone}}/{{name}}
$ terraform import google_filestore_instance.default {{zone}}/{{name}}
$ terraform import google_filestore_instance.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_firestore_index

Cloud Firestore indexes enable simple and complex queries against documents in a database. This resource manages composite indexes and not single field indexes.

To get more information about Index, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Firestore Index Basic

```
resource "google_firestore_index" "my-index" {
  project = "my-project-name"

  collection = "chatrooms"

  fields {
    field_path = "name"
    order      = "ASCENDING"
  }

  fields {
    field_path = "description"
    order      = "DESCENDING"
  }

  fields {
    field_path = "__name__"
    order      = "DESCENDING"
  }
}
```

» Argument Reference

The following arguments are supported:

- `collection` - (Required) The collection being indexed.

- `fields` - (Required) The fields supported by this index. The last field entry is always for the field path `__name__`. If, on creation, `__name__` was not specified as the last field, it will be added automatically with the same direction as that of the last field defined. If the final field in a composite index is not directional, the `__name__` will be ordered `"ASCENDING"` (unless explicitly specified otherwise). Structure is documented below.

The `fields` block supports:

- `field_path` - (Optional) Name of the field.
- `order` - (Optional) Indicates that this field supports ordering by the specified order or comparing using =, <, <=, >, >=. Only one of `order` and `arrayConfig` can be specified.
- `array_config` - (Optional) Indicates that this field supports operations on arrayValues. Only one of `order` and `arrayConfig` can be specified.

- `database` - (Optional) The Firestore database id. Defaults to `"(default)"`.
- `query_scope` - (Optional) The scope at which a query is run. One of `"COLLECTION"` or `"COLLECTION_GROUP"`. Defaults to `"COLLECTION"`.
- `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` - A server defined name for this index. Format: `projects/{{project}}/databases/{{database}}/`

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 10 minutes.
- `delete` - Default is 10 minutes.

» Import

Index can be imported using any of these accepted formats:

```
$ terraform import google_firestore_index.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_healthcare_dataset

A Healthcare `Dataset` is a toplevel logical grouping of `dicomStores`, `fhirStores` and `hl7V2Stores`.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about Dataset, see:

- API documentation
- How-to Guides
  - Creating a dataset

» Example Usage - Healthcare Dataset Basic

```
resource "google_healthcare_dataset" "default" {
  name      = "example-dataset"
  location  = "us-central1"
  time_zone = "UTC"
  provider  = google-beta
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource name for the Dataset.
- `location` - (Required) The location for the Dataset.

- `time_zone` - (Optional) The default timezone used by this dataset. Must be a either a valid IANA time zone name such as "America/New_York" or

empty, which defaults to UTC. This is used for parsing times in resources (e.g., HL7 messages) where no explicit timezone is specified.

- `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:

- `self_link` - The fully qualified name of this dataset

» 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

Dataset can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_healthcare_dataset.default projects/{{projec
$ terraform import -provider=google-beta google_healthcare_dataset.default {{project}}/{{loc
$ terraform import -provider=google-beta google_healthcare_dataset.default {{location}}/{{na
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for Google Cloud Healthcare dataset

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

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

- `google_healthcare_dataset_iam_policy`: Authoritative. Sets the IAM policy for the dataset and replaces any existing policy already attached.
- `google_healthcare_dataset_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 dataset are preserved.
- `google_healthcare_dataset_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the dataset are preserved.

Note: `google_healthcare_dataset_iam_policy` **cannot** be used in conjunction with `google_healthcare_dataset_iam_binding` and `google_healthcare_dataset_iam_member` or they will fight over what your policy should be.

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

» google_healthcare_dataset_iam_policy

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

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

resource "google_healthcare_dataset_iam_policy" "dataset" {
  dataset_id  = "your-dataset-id"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_dataset_iam_binding

```
resource "google_healthcare_dataset_iam_binding" "dataset" {
  dataset_id = "your-dataset-id"
  role       = "roles/editor"

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

```
  ]
}
```

» google_healthcare_dataset_iam_member

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

» Argument Reference

The following arguments are supported:

- `dataset_id` - (Required) The dataset ID, in the form `{project_id}/{location_name}/{dataset_name}` or `{location_name}/{dataset_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 G Suite domain (primary, instead of alias) 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_healthcare_dataset_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_healthcare_dataset_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 dataset'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 `dataset_id`, role, and account e.g.

```
$ terraform import google_healthcare_dataset_iam_member.dataset_iam "your-project-id/locatio
```

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

```
$ terraform import google_healthcare_dataset_iam_binding.dataset_iam "your-project-id/locati
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `dataset_id`, role, and account e.g.

```
$ terraform import google_healthcare_dataset_iam_policy.dataset_iam your-project-id/location
```

» IAM policy for Google Cloud Healthcare dataset

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

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

- `google_healthcare_dataset_iam_policy`: Authoritative. Sets the IAM policy for the dataset and replaces any existing policy already attached.
- `google_healthcare_dataset_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 dataset are preserved.
- `google_healthcare_dataset_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the dataset are preserved.

Note: `google_healthcare_dataset_iam_policy` **cannot** be used in conjunction with `google_healthcare_dataset_iam_binding` and `google_healthcare_dataset_iam_member` or they will fight over what your policy should be.

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

» google_healthcare_dataset_iam_policy

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

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

resource "google_healthcare_dataset_iam_policy" "dataset" {
  dataset_id  = "your-dataset-id"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_dataset_iam_binding

```
resource "google_healthcare_dataset_iam_binding" "dataset" {
  dataset_id = "your-dataset-id"
  role       = "roles/editor"

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

» google_healthcare_dataset_iam_member

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

» Argument Reference

The following arguments are supported:

- `dataset_id` - (Required) The dataset ID, in the form `{project_id}/{location_name}/{dataset_name}` or `{location_name}/{dataset_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 G Suite domain (primary, instead of alias) 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_healthcare_dataset_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_healthcare_dataset_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 dataset'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 `dataset_id`, role, and account e.g.

```
$ terraform import google_healthcare_dataset_iam_member.dataset_iam "your-project-id/locatio
```

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

```
$ terraform import google_healthcare_dataset_iam_binding.dataset_iam "your-project-id/locati
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `dataset_id`, role, and account e.g.

```
$ terraform import google_healthcare_dataset_iam_policy.dataset_iam your-project-id/location
```

» IAM policy for Google Cloud Healthcare dataset

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

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

- `google_healthcare_dataset_iam_policy`: Authoritative. Sets the IAM policy for the dataset and replaces any existing policy already attached.
- `google_healthcare_dataset_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 dataset are preserved.
- `google_healthcare_dataset_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the dataset are preserved.

Note: `google_healthcare_dataset_iam_policy` **cannot** be used in conjunction with `google_healthcare_dataset_iam_binding` and `google_healthcare_dataset_iam_member` or they will fight over what your policy should be.

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

» google_healthcare_dataset_iam_policy

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

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

```
    ]
  }
}

resource "google_healthcare_dataset_iam_policy" "dataset" {
  dataset_id  = "your-dataset-id"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_dataset_iam_binding

```
resource "google_healthcare_dataset_iam_binding" "dataset" {
  dataset_id = "your-dataset-id"
  role       = "roles/editor"

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

» google_healthcare_dataset_iam_member

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

» Argument Reference

The following arguments are supported:

- `dataset_id` - (Required) The dataset ID, in the form `{project_id}/{location_name}/{dataset_name}` or `{location_name}/{dataset_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 G Suite domain (primary, instead of alias) 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_healthcare_dataset_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_healthcare_dataset_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 dataset'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 `dataset_id`, role, and account e.g.

```
$ terraform import google_healthcare_dataset_iam_member.dataset_iam "your-project-id/locatio
```

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

```
$ terraform import google_healthcare_dataset_iam_binding.dataset_iam "your-project-id/locati
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `dataset_id`, role, and account e.g.

```
$ terraform import google_healthcare_dataset_iam_policy.dataset_iam your-project-id/location
```

» google_healthcare_fhir_store

A FhirStore is a datastore inside a Healthcare dataset that conforms to the FHIR (https://www.hl7.org/fhir/STU3/) standard for Healthcare information

exchange

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about FhirStore, see:

- API documentation
- How-to Guides
  - Creating a FHIR store

» Example Usage - Healthcare Fhir Store Basic

```
resource "google_healthcare_fhir_store" "default" {
  name    = "example-fhir-store"
  dataset = google_healthcare_dataset.dataset.id

  enable_update_create          = false
  disable_referential_integrity = false
  disable_resource_versioning   = false
  enable_history_import         = false

  notification_config {
    pubsub_topic = google_pubsub_topic.topic.id
  }

  labels = {
    label1 = "labelvalue1"
  }
  provider = google-beta
}

resource "google_pubsub_topic" "topic" {
  name     = "fhir-notifications"
  provider = google-beta
}

resource "google_healthcare_dataset" "dataset" {
  name     = "example-dataset"
  location = "us-central1"
  provider = google-beta
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource name for the FhirStore. ** Changing this property may recreate the FHIR store (removing all data) **
- `dataset` - (Required) Identifies the dataset addressed by this request. Must be in the format 'projects/{project}/locations/{location}/datasets/{dataset}'

- `enable_update_create` - (Optional) Whether this FHIR store has the updateCreate capability. This determines if the client can use an Update operation to create a new resource with a client-specified ID. If false, all IDs are server-assigned through the Create operation and attempts to Update a non-existent resource will return errors. Please treat the audit logs with appropriate levels of care if client-specified resource IDs contain sensitive data such as patient identifiers, those IDs will be part of the FHIR resource path recorded in Cloud audit logs and Cloud Pub/Sub notifications.
- `disable_referential_integrity` - (Optional) Whether to disable referential integrity in this FHIR store. This field is immutable after FHIR store creation. The default value is false, meaning that the API will enforce referential integrity and fail the requests that will result in inconsistent state in the FHIR store. When this field is set to true, the API will skip referential integrity check. Consequently, operations that rely on references, such as Patient.get$everything, will not return all the results if broken references exist. ** Changing this property may recreate the FHIR store (removing all data) **
- `disable_resource_versioning` - (Optional) Whether to disable resource versioning for this FHIR store. This field can not be changed after the creation of FHIR store. If set to false, which is the default behavior, all write operations will cause historical versions to be recorded automatically. The historical versions can be fetched through the history APIs, but cannot be updated. If set to true, no historical versions will be kept. The server will send back errors for attempts to read the historical versions. ** Changing this property may recreate the FHIR store (removing all data) **
- `enable_history_import` - (Optional) Whether to allow the bulk import API to accept history bundles and directly insert historical resource versions into the FHIR store. Importing resource histories creates resource interactions that appear to have occurred in the past, which clients may not want to allow. If set to false, history bundles within an import will fail with an error. ** Changing this property may recreate the FHIR store (removing all data) ** ** This property can be changed manually in

the Google Cloud Healthcare admin console without recreating the FHIR store **

- `labels` - (Optional) User-supplied key-value pairs used to organize FHIR stores. Label keys must be between 1 and 63 characters long, have a UTF-8 encoding of maximum 128 bytes, and must conform to the following PCRE regular expression: [\p{Ll}\p{Lo}][\p{Ll}\p{Lo}\p{N}-]{0,62} *Label values are optional, must be between 1 and 63 characters long, have a UTF-8 encoding of maximum 128 bytes, and must conform to the following PCRE regular expression: [\p{Ll}\p{Lo}\p{N}-]{0,63}* No more than 64 labels can be associated with a given store. An object containing a list of "key": value pairs. Example: { "name": "wrench", "mass": "1.3kg", "count": "3" }.
- `notification_config` - (Optional) A nested object resource Structure is documented below.

The `notification_config` block supports:

- `pubsub_topic` - (Required) The Cloud Pub/Sub topic that notifications of changes are published on. Supplied by the client. PubsubMessage.Data will contain the resource name. PubsubMessage.MessageId is the ID of this message. It is guaranteed to be unique within the topic. PubsubMessage.PublishTime is the time at which the message was published. Notifications are only sent if the topic is non-empty. Topic names must be scoped to a project. cloud-healthcare@system.gserviceaccount.com must have publisher permissions on the given Cloud Pub/Sub topic. Not having adequate permissions will cause the calls that send notifications to fail.

» Attributes Reference

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

- `self_link` - The fully qualified name of this dataset

» 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

FhirStore can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_healthcare_fhir_store.default {{dataset}}/fh
$ terraform import -provider=google-beta google_healthcare_fhir_store.default {{dataset}}/{
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» IAM policy for Google Cloud Healthcare FHIR store

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for Healthcare FHIR store. Each of these resources serves a different use case:

- `google_healthcare_fhir_store_iam_policy`: Authoritative. Sets the IAM policy for the FHIR store and replaces any existing policy already attached.
- `google_healthcare_fhir_store_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 FHIR store are preserved.
- `google_healthcare_fhir_store_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the FHIR store are preserved.

Note: `google_healthcare_fhir_store_iam_policy` **cannot** be used in conjunction with `google_healthcare_fhir_store_iam_binding` and `google_healthcare_fhir_store_iam_member` or they will fight over what your policy should be.

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

» google_healthcare_fhir_store_iam_policy

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

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

resource "google_healthcare_fhir_store_iam_policy" "fhir_store" {
  fhir_store_id = "your-fhir-store-id"
  policy_data   = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_fhir_store_iam_binding

```
resource "google_healthcare_fhir_store_iam_binding" "fhir_store" {
  fhir_store_id = "your-fhir-store-id"
  role          = "roles/editor"

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

» google_healthcare_fhir_store_iam_member

```
resource "google_healthcare_fhir_store_iam_member" "fhir_store" {
  fhir_store_id = "your-fhir-store-id"
  role          = "roles/editor"
  member        = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `fhir_store_id` - (Required) The FHIR store ID, in the form `{project_id}/{location_name}/{dataset_name}/{fhir_store_name}` or `{location_name}/{dataset_name}/{fhir_store_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 G Suite domain (primary, instead of alias) 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_healthcare_fhir_store_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`

- `policy_data` - (Required only by `google_healthcare_fhir_store_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 FHIR store'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 `fhir_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_fhir_store_iam_member.fhir_store_iam "your-project-id/l
```

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

```
$ terraform import google_healthcare_fhir_store_iam_binding.fhir_store_iam "your-project-id/
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `fhir_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_fhir_store_iam_policy.fhir_store_iam your-project-id/lo
```

» IAM policy for Google Cloud Healthcare FHIR store

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for Healthcare FHIR store. Each of these resources serves a different use case:

- `google_healthcare_fhir_store_iam_policy`: Authoritative. Sets the IAM policy for the FHIR store and replaces any existing policy already attached.
- `google_healthcare_fhir_store_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 FHIR store are preserved.
- `google_healthcare_fhir_store_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the FHIR store are preserved.

Note: `google_healthcare_fhir_store_iam_policy` **cannot** be used in conjunction with `google_healthcare_fhir_store_iam_binding` and `google_healthcare_fhir_store_iam_member` or they will fight over what your policy should be.

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

» google_healthcare_fhir_store_iam_policy

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

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

resource "google_healthcare_fhir_store_iam_policy" "fhir_store" {
  fhir_store_id = "your-fhir-store-id"
  policy_data   = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_fhir_store_iam_binding

```
resource "google_healthcare_fhir_store_iam_binding" "fhir_store" {
  fhir_store_id = "your-fhir-store-id"
  role          = "roles/editor"

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

» google_healthcare_fhir_store_iam_member

```
resource "google_healthcare_fhir_store_iam_member" "fhir_store" {
  fhir_store_id = "your-fhir-store-id"
  role          = "roles/editor"
  member        = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `fhir_store_id` - (Required) The FHIR store ID, in the form `{project_id}/{location_name}/{dataset_name}/{fhir_store_name}` or `{location_name}/{dataset_name}/{fhir_store_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 G Suite domain (primary, instead of alias) 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_healthcare_fhir_store_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_healthcare_fhir_store_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 FHIR store'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 `fhir_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_fhir_store_iam_member.fhir_store_iam "your-project-id/l
```

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

```
$ terraform import google_healthcare_fhir_store_iam_binding.fhir_store_iam "your-project-id/
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `fhir_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_fhir_store_iam_policy.fhir_store_iam your-project-id/lo
```

» IAM policy for Google Cloud Healthcare FHIR store

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for Healthcare FHIR store. Each of these resources serves a different use case:

- `google_healthcare_fhir_store_iam_policy`: Authoritative. Sets the IAM policy for the FHIR store and replaces any existing policy already attached.

- `google_healthcare_fhir_store_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 FHIR store are preserved.
- `google_healthcare_fhir_store_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the FHIR store are preserved.

Note: `google_healthcare_fhir_store_iam_policy` **cannot** be used in conjunction with `google_healthcare_fhir_store_iam_binding` and `google_healthcare_fhir_store_iam_member` or they will fight over what your policy should be.

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

» google_healthcare_fhir_store_iam_policy

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

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

resource "google_healthcare_fhir_store_iam_policy" "fhir_store" {
  fhir_store_id = "your-fhir-store-id"
  policy_data   = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_fhir_store_iam_binding

```
resource "google_healthcare_fhir_store_iam_binding" "fhir_store" {
  fhir_store_id = "your-fhir-store-id"
  role          = "roles/editor"

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

» google_healthcare_fhir_store_iam_member

```
resource "google_healthcare_fhir_store_iam_member" "fhir_store" {
  fhir_store_id = "your-fhir-store-id"
  role          = "roles/editor"
  member        = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `fhir_store_id` - (Required) The FHIR store ID, in the form `{project_id}/{location_name}/{dataset_name}/{fhir_store_name}` or `{location_name}/{dataset_name}/{fhir_store_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 G Suite domain (primary, instead of alias) 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_healthcare_fhir_store_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_healthcare_fhir_store_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 FHIR store'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 `fhir_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_fhir_store_iam_member.fhir_store_iam "your-project-id/l
```

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

```
$ terraform import google_healthcare_fhir_store_iam_binding.fhir_store_iam "your-project-id/
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `fhir_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_fhir_store_iam_policy.fhir_store_iam your-project-id/lo
```

» google_healthcare_dicom_store

A DicomStore is a datastore inside a Healthcare dataset that conforms to the DICOM (https://www.dicomstandard.org/about/) standard for Healthcare information exchange

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about DicomStore, see:

- API documentation
- How-to Guides
  - Creating a DICOM store

» Example Usage - Healthcare Dicom Store Basic

```
resource "google_healthcare_dicom_store" "default" {
  name    = "example-dicom-store"
  dataset = google_healthcare_dataset.dataset.id

  notification_config {
    pubsub_topic = google_pubsub_topic.topic.id
  }
```

```
  labels = {
    label1 = "labelvalue1"
  }
  provider = google-beta
}

resource "google_pubsub_topic" "topic" {
  name     = "dicom-notifications"
  provider = google-beta
}

resource "google_healthcare_dataset" "dataset" {
  name     = "example-dataset"
  location = "us-central1"
  provider = google-beta
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource name for the DicomStore. ** Changing this property may recreate the Dicom store (removing all data) **
- `dataset` - (Required) Identifies the dataset addressed by this request. Must be in the format 'projects/{project}/locations/{location}/datasets/{dataset}'

- `labels` - (Optional) User-supplied key-value pairs used to organize DICOM stores. Label keys must be between 1 and 63 characters long, have a UTF-8 encoding of maximum 128 bytes, and must conform to the following PCRE regular expression: [\p{Ll}\p{Lo}][\p{Ll}\p{Lo}\p{N}-*]{0,62} Label values are optional, must be between 1 and 63 characters long, have a UTF-8 encoding of maximum 128 bytes, and must conform to the following PCRE regular expression: [\p{Ll}\p{Lo}\p{N}-]{0,63}* No more than 64 labels can be associated with a given store. An object containing a list of "key": value pairs. Example: { "name": "wrench", "mass": "1.3kg", "count": "3" }.
- `notification_config` - (Optional) A nested object resource Structure is documented below.

The `notification_config` block supports:

- `pubsub_topic` - (Required) The Cloud Pub/Sub topic that notifications of changes are published on. Supplied by the client. PubsubMessage.Data

will contain the resource name. PubsubMessage.MessageId is the ID of this message. It is guaranteed to be unique within the topic. PubsubMessage.PublishTime is the time at which the message was published. Notifications are only sent if the topic is non-empty. Topic names must be scoped to a project. cloud-healthcare@system.gserviceaccount.com must have publisher permissions on the given Cloud Pub/Sub topic. Not having adequate permissions will cause the calls that send notifications to fail.

» Attributes Reference

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

- `self_link` - The fully qualified name of this dataset

» 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

DicomStore can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_healthcare_dicom_store.default {{dataset}}/d
$ terraform import -provider=google-beta google_healthcare_dicom_store.default {{dataset}}/
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» IAM policy for Google Cloud Healthcare DICOM store

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for Healthcare DICOM store. Each of these resources serves a different use case:

- `google_healthcare_dicom_store_iam_policy`: Authoritative. Sets the IAM policy for the DICOM store and replaces any existing policy already attached.
- `google_healthcare_dicom_store_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 DICOM store are preserved.
- `google_healthcare_dicom_store_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the DICOM store are preserved.

Note: `google_healthcare_dicom_store_iam_policy` **cannot** be used in conjunction with `google_healthcare_dicom_store_iam_binding` and `google_healthcare_dicom_store_iam_member` or they will fight over what your policy should be.

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

» google_healthcare_dicom_store_iam_policy

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

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

resource "google_healthcare_dicom_store_iam_policy" "dicom_store" {
  dicom_store_id = "your-dicom-store-id"
  policy_data    = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_dicom_store_iam_binding

```
resource "google_healthcare_dicom_store_iam_binding" "dicom_store" {
  dicom_store_id = "your-dicom-store-id"
  role           = "roles/editor"

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

```
}
```

» google_healthcare_dicom_store_iam_member

```
resource "google_healthcare_dicom_store_iam_member" "dicom_store" {
  dicom_store_id = "your-dicom-store-id"
  role           = "roles/editor"
  member         = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `dicom_store_id` - (Required) The DICOM store ID, in the form `{project_id}/{location_name}/{dataset_name}/{dicom_store_name}` or `{location_name}/{dataset_name}/{dicom_store_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 G Suite domain (primary, instead of alias) 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_healthcare_dicom_store_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_healthcare_dicom_store_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 DICOM store'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 `dicom_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_dicom_store_iam_member.dicom_store_iam "your-project-id
```

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

```
$ terraform import google_healthcare_dicom_store_iam_binding.dicom_store_iam "your-project-i
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `dicom_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_dicom_store_iam_policy.dicom_store_iam your-project-id/
```

» IAM policy for Google Cloud Healthcare DICOM store

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for Healthcare DICOM store. Each of these resources serves a different use case:

- `google_healthcare_dicom_store_iam_policy`: Authoritative. Sets the IAM policy for the DICOM store and replaces any existing policy already attached.
- `google_healthcare_dicom_store_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 DICOM store are preserved.
- `google_healthcare_dicom_store_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the DICOM store are preserved.

Note: `google_healthcare_dicom_store_iam_policy` **cannot** be used in conjunction with `google_healthcare_dicom_store_iam_binding` and `google_healthcare_dicom_store_iam_member` or they will fight over what your policy should be.

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

» google_healthcare_dicom_store_iam_policy

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

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

resource "google_healthcare_dicom_store_iam_policy" "dicom_store" {
  dicom_store_id = "your-dicom-store-id"
  policy_data    = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_dicom_store_iam_binding

```
resource "google_healthcare_dicom_store_iam_binding" "dicom_store" {
  dicom_store_id = "your-dicom-store-id"
  role           = "roles/editor"

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

» google_healthcare_dicom_store_iam_member

```
resource "google_healthcare_dicom_store_iam_member" "dicom_store" {
  dicom_store_id = "your-dicom-store-id"
  role           = "roles/editor"
  member         = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `dicom_store_id` - (Required) The DICOM store ID, in the form `{project_id}/{location_name}/{dataset_name}/{dicom_store_name}` or `{location_name}/{dataset_name}/{dicom_store_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 G Suite domain (primary, instead of alias) 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_healthcare_dicom_store_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_healthcare_dicom_store_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 DICOM store'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 `dicom_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_dicom_store_iam_member.dicom_store_iam "your-project-id
```

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

```
$ terraform import google_healthcare_dicom_store_iam_binding.dicom_store_iam "your-project-i
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `dicom_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_dicom_store_iam_policy.dicom_store_iam your-project-id/
```

» IAM policy for Google Cloud Healthcare DICOM store

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for Healthcare DICOM store. Each of these resources serves a different use case:

- `google_healthcare_dicom_store_iam_policy`: Authoritative. Sets the IAM policy for the DICOM store and replaces any existing policy already attached.
- `google_healthcare_dicom_store_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 DICOM store are preserved.
- `google_healthcare_dicom_store_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the DICOM store are preserved.

Note: `google_healthcare_dicom_store_iam_policy` **cannot** be used in conjunction with `google_healthcare_dicom_store_iam_binding` and `google_healthcare_dicom_store_iam_member` or they will fight over what your policy should be.

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

» google_healthcare_dicom_store_iam_policy

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

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

resource "google_healthcare_dicom_store_iam_policy" "dicom_store" {
  dicom_store_id = "your-dicom-store-id"
  policy_data    = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_dicom_store_iam_binding

```
resource "google_healthcare_dicom_store_iam_binding" "dicom_store" {
  dicom_store_id = "your-dicom-store-id"
  role           = "roles/editor"

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

» google_healthcare_dicom_store_iam_member

```
resource "google_healthcare_dicom_store_iam_member" "dicom_store" {
  dicom_store_id = "your-dicom-store-id"
  role           = "roles/editor"
  member         = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `dicom_store_id` - (Required) The DICOM store ID, in the form `{project_id}/{location_name}/{dataset_name}/{dicom_store_name}` or `{location_name}/{dataset_name}/{dicom_store_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 G Suite domain (primary, instead of alias) 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_healthcare_dicom_store_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`

- `policy_data` - (Required only by `google_healthcare_dicom_store_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 DICOM store'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 `dicom_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_dicom_store_iam_member.dicom_store_iam "your-project-id
```

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

```
$ terraform import google_healthcare_dicom_store_iam_binding.dicom_store_iam "your-project-i
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `dicom_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_dicom_store_iam_policy.dicom_store_iam your-project-id/
```

» google_healthcare_hl7_v2_store

A Hl7V2Store is a datastore inside a Healthcare dataset that conforms to the FHIR (https://www.hl7.org/hl7V2/STU3/) standard for Healthcare information exchange

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about Hl7V2Store, see:

- API documentation
- How-to Guides
  - Creating a HL7v2 Store

» Example Usage - Healthcare Hl7 V2 Store Basic

```
resource "google_healthcare_hl7_v2_store" "default" {
  name    = "example-hl7-v2-store"
  dataset = google_healthcare_dataset.dataset.id

  parser_config {
    allow_null_header  = false
    segment_terminator = "Jw=="
  }

  notification_config {
    pubsub_topic = google_pubsub_topic.topic.id
  }

  labels = {
    label1 = "labelvalue1"
  }
  provider = google-beta
}

resource "google_pubsub_topic" "topic" {
  name     = "hl7-v2-notifications"
  provider = google-beta
}

resource "google_healthcare_dataset" "dataset" {
  name     = "example-dataset"
  location = "us-central1"
  provider = google-beta
```

```
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource name for the Hl7V2Store. ** Changing this property may recreate the Hl7v2 store (removing all data) **
- `dataset` - (Required) Identifies the dataset addressed by this request. Must be in the format 'projects/{project}/locations/{location}/datasets/{dataset}'

- `parser_config` - (Optional) A nested object resource Structure is documented below.
- `labels` - (Optional) User-supplied key-value pairs used to organize HL7v2 stores. Label keys must be between 1 and 63 characters long, have a UTF-8 encoding of maximum 128 bytes, and must conform to the following PCRE regular expression: [\p{Ll}\p{Lo}][\p{Ll}\p{Lo}\p{N}-*]{0,62} Label values are optional, must be between 1 and 63 characters long, have a UTF-8 encoding of maximum 128 bytes, and must conform to the following PCRE regular expression: [\p{Ll}\p{Lo}\p{N}-]{0,63}* No more than 64 labels can be associated with a given store. An object containing a list of "key": value pairs. Example: { "name": "wrench", "mass": "1.3kg", "count": "3" }.
- `notification_config` - (Optional) A nested object resource Structure is documented below.

The `parser_config` block supports:

- `allow_null_header` - (Optional) Determines whether messages with no header are allowed.
- `segment_terminator` - (Optional) Byte(s) to be used as the segment terminator. If this is unset, '\r' will be used as segment terminator. A base64-encoded string.

The `notification_config` block supports:

- `pubsub_topic` - (Required) The Cloud Pub/Sub topic that notifications of changes are published on. Supplied by the client. PubsubMessage.Data will contain the resource name. PubsubMessage.MessageId is the ID of this message. It is guaranteed to be unique within the topic. PubsubMessage.PublishTime is the time at which the message was published. Notifications are only sent if the topic is non-empty. Topic names must be scoped to a project. cloud-healthcare@system.gserviceaccount.com must

have publisher permissions on the given Cloud Pub/Sub topic. Not having adequate permissions will cause the calls that send notifications to fail.

» Attributes Reference

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

- `self_link` - The fully qualified name of this dataset

» 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

Hl7V2Store can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_healthcare_hl7_v2_store.default {{dataset}}/
$ terraform import -provider=google-beta google_healthcare_hl7_v2_store.default {{dataset}}/
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» IAM policy for Google Cloud Healthcare HL7v2 store

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for Healthcare HL7v2 store. Each of these resources serves a different use case:

- `google_healthcare_hl7_v2_store_iam_policy`: Authoritative. Sets the IAM policy for the HL7v2 store and replaces any existing policy already attached.

- `google_healthcare_hl7_v2_store_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 HL7v2 store are preserved.
- `google_healthcare_hl7_v2_store_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the HL7v2 store are preserved.

Note: `google_healthcare_hl7_v2_store_iam_policy` **cannot** be used in conjunction with `google_healthcare_hl7_v2_store_iam_binding` and `google_healthcare_hl7_v2_store_iam_member` or they will fight over what your policy should be.

Note: `google_healthcare_hl7_v2_store_iam_binding` resources **can be** used in conjunction with `google_healthcare_hl7_v2_store_iam_member` resources **only if** they do not grant privilege to the same role.

» google_healthcare_hl7_v2_store_iam_policy

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

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

resource "google_healthcare_hl7_v2_store_iam_policy" "hl7_v2_store" {
  hl7_v2_store_id = "your-hl7-v2-store-id"
  policy_data     = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_hl7_v2_store_iam_binding

```
resource "google_healthcare_hl7_v2_store_iam_binding" "hl7_v2_store" {
  hl7_v2_store_id = "your-hl7-v2-store-id"
  role            = "roles/editor"

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

» google_healthcare_hl7_v2_store_iam_member

```
resource "google_healthcare_hl7_v2_store_iam_member" "hl7_v2_store" {
  hl7_v2_store_id = "your-hl7-v2-store-id"
  role            = "roles/editor"
  member          = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `hl7_v2_store_id` - (Required) The HL7v2 store ID, in the form `{project_id}/{location_name}/{dataset_name}/{hl7_v2_store_name}` or `{location_name}/{dataset_name}/{hl7_v2_store_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 G Suite domain (primary, instead of alias) 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_healthcare_hl7_v2_store_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_healthcare_hl7_v2_store_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 HL7v2 store'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 `hl7_v2_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_hl7_v2_store_iam_member.hl7_v2_store_iam "your-project-
```

IAM binding imports use space-delimited identifiers; the resource in question and the role. This binding resource can be imported using the `hl7_v2_store_id` and role, e.g.

```
$ terraform import google_healthcare_hl7_v2_store_iam_binding.hl7_v2_store_iam "your-project
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `hl7_v2_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_hl7_v2_store_iam_policy.hl7_v2_store_iam your-project-i
```

» IAM policy for Google Cloud Healthcare HL7v2 store

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for Healthcare HL7v2 store. Each of these resources serves a different use case:

- `google_healthcare_hl7_v2_store_iam_policy`: Authoritative. Sets the IAM policy for the HL7v2 store and replaces any existing policy already attached.
- `google_healthcare_hl7_v2_store_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 HL7v2 store are preserved.
- `google_healthcare_hl7_v2_store_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the HL7v2 store are preserved.

Note: `google_healthcare_hl7_v2_store_iam_policy` **cannot** be used in conjunction with `google_healthcare_hl7_v2_store_iam_binding` and `google_healthcare_hl7_v2_store_iam_member` or they will fight over what your policy should be.

Note: `google_healthcare_hl7_v2_store_iam_binding` resources **can be** used in conjunction with `google_healthcare_hl7_v2_store_iam_member` resources **only if** they do not grant privilege to the same role.

» google_healthcare_hl7_v2_store_iam_policy

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

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

resource "google_healthcare_hl7_v2_store_iam_policy" "hl7_v2_store" {
  hl7_v2_store_id = "your-hl7-v2-store-id"
  policy_data     = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_hl7_v2_store_iam_binding

```
resource "google_healthcare_hl7_v2_store_iam_binding" "hl7_v2_store" {
  hl7_v2_store_id = "your-hl7-v2-store-id"
  role            = "roles/editor"

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

» google_healthcare_hl7_v2_store_iam_member

```
resource "google_healthcare_hl7_v2_store_iam_member" "hl7_v2_store" {
  hl7_v2_store_id = "your-hl7-v2-store-id"
  role            = "roles/editor"
  member          = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `hl7_v2_store_id` - (Required) The HL7v2 store ID, in the form `{project_id}/{location_name}/{dataset_name}/{hl7_v2_store_name}` or `{location_name}/{dataset_name}/{hl7_v2_store_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 G Suite domain (primary, instead of alias) 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_healthcare_hl7_v2_store_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_healthcare_hl7_v2_store_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 HL7v2 store'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 `hl7_v2_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_hl7_v2_store_iam_member.hl7_v2_store_iam "your-project-
```

IAM binding imports use space-delimited identifiers; the resource in question and the role. This binding resource can be imported using the `hl7_v2_store_id` and role, e.g.

```
$ terraform import google_healthcare_hl7_v2_store_iam_binding.hl7_v2_store_iam "your-project
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `hl7_v2_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_hl7_v2_store_iam_policy.hl7_v2_store_iam your-project-i
```

» IAM policy for Google Cloud Healthcare HL7v2 store

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for Healthcare HL7v2 store. Each of these resources serves a different use case:

- `google_healthcare_hl7_v2_store_iam_policy`: Authoritative. Sets the IAM policy for the HL7v2 store and replaces any existing policy already attached.
- `google_healthcare_hl7_v2_store_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 HL7v2 store are preserved.
- `google_healthcare_hl7_v2_store_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the HL7v2 store are preserved.

Note: `google_healthcare_hl7_v2_store_iam_policy` **cannot** be used in conjunction with `google_healthcare_hl7_v2_store_iam_binding` and `google_healthcare_hl7_v2_store_iam_member` or they will fight over what your policy should be.

Note: `google_healthcare_hl7_v2_store_iam_binding` resources **can be** used in conjunction with `google_healthcare_hl7_v2_store_iam_member` resources **only if** they do not grant privilege to the same role.

» google_healthcare_hl7_v2_store_iam_policy

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

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

resource "google_healthcare_hl7_v2_store_iam_policy" "hl7_v2_store" {
  hl7_v2_store_id = "your-hl7-v2-store-id"
  policy_data     = data.google_iam_policy.admin.policy_data
}
```

» google_healthcare_hl7_v2_store_iam_binding

```
resource "google_healthcare_hl7_v2_store_iam_binding" "hl7_v2_store" {
  hl7_v2_store_id = "your-hl7-v2-store-id"
  role            = "roles/editor"

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

» google_healthcare_hl7_v2_store_iam_member

```
resource "google_healthcare_hl7_v2_store_iam_member" "hl7_v2_store" {
  hl7_v2_store_id = "your-hl7-v2-store-id"
  role            = "roles/editor"
  member          = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `hl7_v2_store_id` - (Required) The HL7v2 store ID, in the form `{project_id}/{location_name}/{dataset_name}/{hl7_v2_store_name}` or `{location_name}/{dataset_name}/{hl7_v2_store_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 G Suite domain (primary, instead of alias) 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_healthcare_hl7_v2_store_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`

- `policy_data` - (Required only by `google_healthcare_hl7_v2_store_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 HL7v2 store'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 `hl7_v2_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_hl7_v2_store_iam_member.hl7_v2_store_iam "your-project-
```

IAM binding imports use space-delimited identifiers; the resource in question and the role. This binding resource can be imported using the `hl7_v2_store_id` and role, e.g.

```
$ terraform import google_healthcare_hl7_v2_store_iam_binding.hl7_v2_store_iam "your-project
```

IAM policy imports use the identifier of the resource in question. This policy resource can be imported using the `hl7_v2_store_id`, role, and account e.g.

```
$ terraform import google_healthcare_hl7_v2_store_iam_policy.hl7_v2_store_iam your-project-i
```

» google_kms_crypto_key

A `CryptoKey` represents a logical key that can be used for cryptographic operations.

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.* When Terraform destroys these keys, any data previously encrypted with these keys will be irrecoverable. For this reason, it is strongly recommended that you add lifecycle hooks to the resource to prevent accidental destruction.

To get more information about CryptoKey, see:

- API documentation
- How-to Guides
  - Creating a key

» Example Usage - Kms Crypto Key Basic

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}

resource "google_kms_crypto_key" "example-key" {
  name            = "crypto-key-example"
  key_ring        = google_kms_key_ring.keyring.self_link
  rotation_period = "100000s"

  lifecycle {
    prevent_destroy = true
  }
}
```

» Example Usage - Kms Crypto Key Asymmetric Sign

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}

resource "google_kms_crypto_key" "example-asymmetric-sign-key" {
  name     = "crypto-key-example"
  key_ring = google_kms_key_ring.keyring.self_link
```

```
  purpose  = "ASYMMETRIC_SIGN"

  version_template {
    algorithm = "EC_SIGN_P384_SHA384"
  }

  lifecycle {
    prevent_destroy = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource name for the CryptoKey.
- `key_ring` - (Required) The KeyRing that this key belongs to. Format: `'projects/{{project}}/locations/{{location}}/keyRings/{{keyRing}}'`.

- `labels` - (Optional) Labels with user-defined metadata to apply to this resource.
- `purpose` - (Optional) The immutable purpose of this CryptoKey. See the purpose reference for possible inputs.
- `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, 86400).
- `version_template` - (Optional) A template describing settings for new crypto key versions. Structure is documented below.

The `version_template` block supports:

- `algorithm` - (Required) The algorithm to use when creating a version based on this template. See the algorithm reference for possible inputs.
- `protection_level` - (Optional) The protection level to use when creating a version based on this template.

» Attributes Reference

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

- `self_link`: The self link of the created CryptoKey. Its format is `{{key_ring}}/cryptoKeys/{{name}}`.

» 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

CryptoKey can be imported using any of these accepted formats:

```
$ terraform import google_kms_crypto_key.default {{key_ring}}/cryptoKeys/{{name}}
$ terraform import google_kms_crypto_key.default {{key_ring}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for Google Cloud KMS crypto key

Three different resources help you manage your IAM policy for KMS crypto key. Each of these resources serves a different use case:

- `google_kms_crypto_key_iam_policy`: Authoritative. Sets the IAM policy for the crypto key and replaces any existing policy already attached.
- `google_kms_crypto_key_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 crypto key are preserved.
- `google_kms_crypto_key_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the crypto key are preserved.

Note: `google_kms_crypto_key_iam_policy` **cannot** be used in conjunction with `google_kms_crypto_key_iam_binding` and `google_kms_crypto_key_iam_member` or they will fight over what your policy should be.

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

» google_kms_crypto_key_iam_policy

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}
resource "google_kms_crypto_key" "key" {
  name            = "crypto-key-example"
  key_ring        = google_kms_key_ring.keyring.id
  rotation_period = "100000s"
  lifecycle {
    prevent_destroy = true
  }
}

data "google_iam_policy" "admin" {
  binding {
    role = "roles/cloudkms.cryptoKeyEncrypter"

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

resource "google_kms_crypto_key_iam_policy" "crypto_key" {
  crypto_key_id = google_kms_crypto_key.key.id
  policy_data = data.google_iam_policy.admin.policy_data
}
```

With IAM Conditions (beta): “‘hcl data ”google_iam_policy” ”admin” { binding { role = ”roles/cloudkms.cryptoKeyEncrypter”

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

condition {
  title       = "expires_after_2019_12_31"
  description = "Expiring at midnight of 2019-12-31"
  expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
```

```
}
```

} } “‘

» google_kms_crypto_key_iam_binding

```
resource "google_kms_crypto_key_iam_binding" "crypto_key" {
  crypto_key_id = google_kms_crypto_key.key.id
  role          = "roles/cloudkms.cryptoKeyEncrypter"

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

With IAM Conditions (beta): “‘hcl resource ”google_kms_crypto_key_iam_binding” ”crypto_key” { crypto_key_id = google_kms_crypto_key.key.id role = ”roles/cloudkms.cryptoKeyEncrypter”

members = [”user:jane@example.com”,]

condition { title = ”expires_after_2019_12_31” description = ”Expiring at midnight of 2019-12-31” expression = ”request.time < timestamp(\”2020-01-01T00:00:00Z\”)” } } “‘

» google_kms_crypto_key_iam_member

```
resource "google_kms_crypto_key_iam_member" "crypto_key" {
  crypto_key_id = google_kms_crypto_key.key.id
  role          = "roles/cloudkms.cryptoKeyEncrypter"
  member        = "user:jane@example.com"
}
```

With IAM Conditions (beta): “‘hcl resource ”google_kms_crypto_key_iam_member” ”crypto_key” { crypto_key_id = google_kms_crypto_key.key.id role = ”roles/cloudkms.cryptoKeyEncrypter” member = ”user:jane@example.com”

condition { title = ”expires_after_2019_12_31” description = ”Expiring at midnight of 2019-12-31” expression = ”request.time < timestamp(\”2020-01-01T00:00:00Z\”)” } } “‘

» Argument Reference

The following arguments are supported:

- `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.
- `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, jane@example.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 G Suite domain (primary, instead of alias) 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}`.
- `policy_data` - (Required only by `google_kms_crypto_key_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 member identity e.g.

```
$ terraform import google_kms_crypto_key_iam_member.crypto_key "your-project-id/location-nam
```

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.crypto_key "your-project-id/location-na
```

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

```
$ terraform import google_kms_crypto_key_iam_policy.crypto_key your-project-id/location-name
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» IAM policy for Google Cloud KMS crypto key

Three different resources help you manage your IAM policy for KMS crypto key. Each of these resources serves a different use case:

- `google_kms_crypto_key_iam_policy`: Authoritative. Sets the IAM policy for the crypto key and replaces any existing policy already attached.
- `google_kms_crypto_key_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 crypto key are preserved.
- `google_kms_crypto_key_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the crypto key are preserved.

Note: `google_kms_crypto_key_iam_policy` **cannot** be used in conjunction with `google_kms_crypto_key_iam_binding` and `google_kms_crypto_key_iam_member` or they will fight over what your policy should be.

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

» google_kms_crypto_key_iam_policy

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}
resource "google_kms_crypto_key" "key" {
  name            = "crypto-key-example"
  key_ring        = google_kms_key_ring.keyring.id
  rotation_period = "100000s"
  lifecycle {
    prevent_destroy = true
  }
}

data "google_iam_policy" "admin" {
  binding {
    role = "roles/cloudkms.cryptoKeyEncrypter"

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

resource "google_kms_crypto_key_iam_policy" "crypto_key" {
  crypto_key_id = google_kms_crypto_key.key.id
  policy_data = data.google_iam_policy.admin.policy_data
}
```

With IAM Conditions (beta): “‘hcl data ”google_iam_policy” ”admin” { binding { role = ”roles/cloudkms.cryptoKeyEncrypter”

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

condition {
  title       = "expires_after_2019_12_31"
  description = "Expiring at midnight of 2019-12-31"
  expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
```

```
}
```

} } “‘

» google_kms_crypto_key_iam_binding

```
resource "google_kms_crypto_key_iam_binding" "crypto_key" {
  crypto_key_id = google_kms_crypto_key.key.id
  role          = "roles/cloudkms.cryptoKeyEncrypter"

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

With IAM Conditions (beta): “‘hcl resource ”google_kms_crypto_key_iam_binding” ”crypto_key” { crypto_key_id = google_kms_crypto_key.key.id role = ”roles/cloudkms.cryptoKeyEncrypter”

members = [”user:jane@example.com”,]

condition { title = ”expires_after_2019_12_31” description = ”Expiring at midnight of 2019-12-31” expression = ”request.time < timestamp(\”2020-01-01T00:00:00Z\”)” } } “‘

» google_kms_crypto_key_iam_member

```
resource "google_kms_crypto_key_iam_member" "crypto_key" {
  crypto_key_id = google_kms_crypto_key.key.id
  role          = "roles/cloudkms.cryptoKeyEncrypter"
  member        = "user:jane@example.com"
}
```

With IAM Conditions (beta): “‘hcl resource ”google_kms_crypto_key_iam_member” ”crypto_key” { crypto_key_id = google_kms_crypto_key.key.id role = ”roles/cloudkms.cryptoKeyEncrypter” member = ”user:jane@example.com”

condition { title = ”expires_after_2019_12_31” description = ”Expiring at midnight of 2019-12-31” expression = ”request.time < timestamp(\”2020-01-01T00:00:00Z\”)” } } “‘

» Argument Reference

The following arguments are supported:

- `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.
- `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, jane@example.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 G Suite domain (primary, instead of alias) 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}`.
- `policy_data` - (Required only by `google_kms_crypto_key_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 member identity e.g.

```
$ terraform import google_kms_crypto_key_iam_member.crypto_key "your-project-id/location-nam
```

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.crypto_key "your-project-id/location-na
```

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

```
$ terraform import google_kms_crypto_key_iam_policy.crypto_key your-project-id/location-name
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» IAM policy for Google Cloud KMS crypto key

Three different resources help you manage your IAM policy for KMS crypto key. Each of these resources serves a different use case:

- `google_kms_crypto_key_iam_policy`: Authoritative. Sets the IAM policy for the crypto key and replaces any existing policy already attached.
- `google_kms_crypto_key_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 crypto key are preserved.
- `google_kms_crypto_key_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the crypto key are preserved.

Note: `google_kms_crypto_key_iam_policy` **cannot** be used in conjunction with `google_kms_crypto_key_iam_binding` and `google_kms_crypto_key_iam_member` or they will fight over what your policy should be.

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

» google_kms_crypto_key_iam_policy

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}
resource "google_kms_crypto_key" "key" {
  name            = "crypto-key-example"
  key_ring        = google_kms_key_ring.keyring.id
  rotation_period = "100000s"
  lifecycle {
    prevent_destroy = true
  }
}

data "google_iam_policy" "admin" {
  binding {
    role = "roles/cloudkms.cryptoKeyEncrypter"

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

resource "google_kms_crypto_key_iam_policy" "crypto_key" {
  crypto_key_id = google_kms_crypto_key.key.id
  policy_data = data.google_iam_policy.admin.policy_data
}
```

With IAM Conditions (beta): “‘hcl data ”google_iam_policy” ”admin” { binding { role = ”roles/cloudkms.cryptoKeyEncrypter”

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

condition {
  title       = "expires_after_2019_12_31"
  description = "Expiring at midnight of 2019-12-31"
  expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
```

```
}
```

} } “‘

» google_kms_crypto_key_iam_binding

```
resource "google_kms_crypto_key_iam_binding" "crypto_key" {
  crypto_key_id = google_kms_crypto_key.key.id
  role          = "roles/cloudkms.cryptoKeyEncrypter"

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

With IAM Conditions (beta): “‘hcl resource ”google_kms_crypto_key_iam_binding” ”crypto_key” { crypto_key_id = google_kms_crypto_key.key.id role = ”roles/cloudkms.cryptoKeyEncrypter”

members = [”user:jane@example.com”,]

condition { title = ”expires_after_2019_12_31” description = ”Expiring at midnight of 2019-12-31” expression = ”request.time < timestamp(\”2020-01-01T00:00:00Z\”)” } } “‘

» google_kms_crypto_key_iam_member

```
resource "google_kms_crypto_key_iam_member" "crypto_key" {
  crypto_key_id = google_kms_crypto_key.key.id
  role          = "roles/cloudkms.cryptoKeyEncrypter"
  member        = "user:jane@example.com"
}
```

With IAM Conditions (beta): “‘hcl resource ”google_kms_crypto_key_iam_member” ”crypto_key” { crypto_key_id = google_kms_crypto_key.key.id role = ”roles/cloudkms.cryptoKeyEncrypter” member = ”user:jane@example.com”

condition { title = ”expires_after_2019_12_31” description = ”Expiring at midnight of 2019-12-31” expression = ”request.time < timestamp(\”2020-01-01T00:00:00Z\”)” } } “‘

» Argument Reference

The following arguments are supported:

- `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.
- `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, jane@example.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 G Suite domain (primary, instead of alias) 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}`.
- `policy_data` - (Required only by `google_kms_crypto_key_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 member identity e.g.

```
$ terraform import google_kms_crypto_key_iam_member.crypto_key "your-project-id/location-nam
```

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.crypto_key "your-project-id/location-na
```

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

```
$ terraform import google_kms_crypto_key_iam_policy.crypto_key your-project-id/location-name
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» google_kms_key_ring

A `KeyRing` is a toplevel logical grouping of `CryptoKeys`.

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.*

To get more information about KeyRing, see:

- API documentation
- How-to Guides
  - Creating a key ring

» Example Usage - Kms Key Ring Basic

```
resource "google_kms_key_ring" "example-keyring" {
```

```
  name     = "keyring-example"
  location = "global"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The resource name for the KeyRing.
- `location` - (Required) The location for the KeyRing. A full list of valid locations can be found by running `gcloud kms locations list`.

- `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:

- `self_link`: The self link of the created KeyRing in the format `projects/{project}/locations/{location}/keyRings/{name}`

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

KeyRing can be imported using any of these accepted formats:

```
$ terraform import google_kms_key_ring.default projects/{{project}}/locations/{{location}}/k
$ terraform import google_kms_key_ring.default {{project}}/{{location}}/{{name}}
$ terraform import google_kms_key_ring.default {{location}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}

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 = google_kms_key_ring.keyring.id
```

```
  policy_data = data.google_iam_policy.admin.policy_data
}
```

With IAM Conditions (beta):

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}

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

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

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_kms_key_ring_iam_policy" "key_ring" {
  key_ring_id = google_kms_key_ring.keyring.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",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_kms_key_ring_iam_binding" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"
```

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» 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"
}
```

With IAM Conditions (beta):

```
resource "google_kms_key_ring_iam_member" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"
  member      = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» 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 G Suite domain (primary, instead of alias) 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.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 binding imports use space-delimited identifiers; the resource in question and the role. This binding resource can be imported using the `key_ring_id` and role, 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`, e.g.

```
$ terraform import google_kms_key_ring_iam_policy.key_ring_iam your-project-id/location-name
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» 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

```
resource "google_kms_key_ring" "keyring" {
```

```
  name     = "keyring-example"
  location = "global"
}

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 = google_kms_key_ring.keyring.id
  policy_data = data.google_iam_policy.admin.policy_data
}
```

With IAM Conditions (beta):

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}

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

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

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_kms_key_ring_iam_policy" "key_ring" {
  key_ring_id = google_kms_key_ring.keyring.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",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_kms_key_ring_iam_binding" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» 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"
}
```

With IAM Conditions (beta):

```
resource "google_kms_key_ring_iam_member" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"
  member      = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
```

```
  }
}
```

» 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 G Suite domain (primary, instead of alias) 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.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.

- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 binding imports use space-delimited identifiers; the resource in question and the role. This binding resource can be imported using the `key_ring_id` and role, 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`, e.g.

```
$ terraform import google_kms_key_ring_iam_policy.key_ring_iam your-project-id/location-name
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» 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

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}

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 = google_kms_key_ring.keyring.id
  policy_data = data.google_iam_policy.admin.policy_data
}
```

With IAM Conditions (beta):

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}

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

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

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_kms_key_ring_iam_policy" "key_ring" {
  key_ring_id = google_kms_key_ring.keyring.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",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_kms_key_ring_iam_binding" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"

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

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» 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"
}
```

With IAM Conditions (beta):

```
resource "google_kms_key_ring_iam_member" "key_ring" {
  key_ring_id = "your-key-ring-id"
  role        = "roles/editor"
  member      = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» 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 G Suite domain (primary, instead of alias) 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.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» 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 binding imports use space-delimited identifiers; the resource in question and the role. This binding resource can be imported using the `key_ring_id` and role, 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`, e.g.

```
$ terraform import google_kms_key_ring_iam_policy.key_ring_iam your-project-id/location-name
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» google_kms_secret_ciphertext

Encrypts secret data with Google Cloud KMS and provides access to the ciphertext.

NOTE: Using this resource will allow you to conceal secret data within your resource definitions, but it does not take care of protecting that data in the logging output, plan output, or state output. Please take care to secure your secret data outside of resource definitions.

To get more information about SecretCiphertext, see:

- API documentation
- How-to Guides
  - Encrypting and decrypting data with a symmetric key

» Example Usage - Kms Secret Ciphertext Basic

```
resource "google_kms_key_ring" "keyring" {
  name     = "keyring-example"
  location = "global"
}

resource "google_kms_crypto_key" "cryptokey" {
  name            = "crypto-key-example"
  key_ring        = google_kms_key_ring.keyring.id
  rotation_period = "100000s"

  lifecycle {
    prevent_destroy = true
  }
}

resource "google_kms_secret_ciphertext" "my_password" {
  crypto_key = google_kms_crypto_key.cryptokey.id
  plaintext  = "my-secret-password"
}
```

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

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-9"
    }
  }

  network_interface {
    network = "default"

    access_config {
    }
  }

  metadata = {
    password = google_kms_secret_ciphertext.my_password.ciphertext
  }
}
```

» Argument Reference

The following arguments are supported:

- `plaintext` - (Required) The plaintext to be encrypted.
- `crypto_key` - (Required) The full name of the CryptoKey that will be used to encrypt the provided plaintext. Format: `'projects/{{project}}/locations/{{location}}/keyRin`

» Attributes Reference

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

- `ciphertext` - Contains the result of encrypting the provided plaintext, encoded in base64.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IAP Tunnel Instance

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for IAP Tunnel Instance. Each of these resources serves a different use case:

- `google_iap_tunnel_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.
- `google_iap_tunnel_instance_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 instance are preserved.
- `google_iap_tunnel_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_iap_tunnel_instance_iam_policy` **cannot** be used in conjunction with `google_iap_tunnel_instance_iam_binding` and `google_iap_tunnel_instance_iam_member` or they will fight over what your policy should be.

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

» google_iap_tunnel_instance_iam_policy

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

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

resource "google_iap_tunnel_instance_iam_policy" "instance" {
    instance  = "your-instance-name"
    policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_tunnel_instance_iam_binding

```
resource "google_iap_tunnel_instance_iam_binding" "instance" {
  instance = "your-instance-name"
  role       = "roles/compute.networkUser"

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

» google_iap_tunnel_instance_iam_member

```
resource "google_iap_tunnel_instance_iam_member" "instance" {
  instance = "your-instance-name"
  role       = "roles/compute.networkUser"
  member     = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name of the instance.
- `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 G Suite domain (primary, instead of alias) 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_iap_tunnel_instance_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_iap_tunnel_instance_iam_policy`) The policy data generated by a `google_iam_policy` data source.

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

- `zone` - (Optional) The zone of the instance. If unspecified, this defaults to the zone configured in the provider.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- full self link or relative link (projects/{{project}}/zones/{{zone}}/instances/{{name}})
- {{project}}/{{zone}}/{{name}}
- {{zone}}/{{name}} (project is taken from provider project)
- {{name}} (project and zone are taken from provider project)

IAM member imports use space-delimited identifiers; the resource in question, the role, and the member identity, e.g.

```
$ terraform import google_iap_tunnel_instance_iam_member.instance "project-name/zone-name/i
```

IAM binding imports use space-delimited identifiers; the resource in question and the role, e.g.

```
$ terraform import google_iap_tunnel_instance_iam_binding.instance "project-name/zone-name/i
```

IAM policy imports use the identifier of the resource in question, e.g.

```
$ terraform import google_iap_tunnel_instance_iam_policy.instance project-name/zone-name/ins
```

» IAM policy for IAP Tunnel Instance

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

Three different resources help you manage your IAM policy for IAP Tunnel Instance. Each of these resources serves a different use case:

- `google_iap_tunnel_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.
- `google_iap_tunnel_instance_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 instance are preserved.
- `google_iap_tunnel_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_iap_tunnel_instance_iam_policy` **cannot** be used in conjunction with `google_iap_tunnel_instance_iam_binding` and `google_iap_tunnel_instance_iam_member` or they will fight over what your policy should be.

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

» google_iap_tunnel_instance_iam_policy

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

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

resource "google_iap_tunnel_instance_iam_policy" "instance" {
    instance  = "your-instance-name"
```

```
    policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_tunnel_instance_iam_binding

```
resource "google_iap_tunnel_instance_iam_binding" "instance" {
  instance = "your-instance-name"
  role       = "roles/compute.networkUser"

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

» google_iap_tunnel_instance_iam_member

```
resource "google_iap_tunnel_instance_iam_member" "instance" {
  instance = "your-instance-name"
  role       = "roles/compute.networkUser"
  member     = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name of the instance.
- `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 G Suite domain (primary, instead of alias) 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_iap_tunnel_instance_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_iap_tunnel_instance_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `zone` - (Optional) The zone of the instance. If unspecified, this defaults to the zone configured in the provider.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- full self link or relative link (projects/{{project}}/zones/{{zone}}/instances/{{name}})
- {{project}}/{{zone}}/{{name}}
- {{zone}}/{{name}} (project is taken from provider project)
- {{name}} (project and zone are taken from provider project)

IAM member imports use space-delimited identifiers; the resource in question, the role, and the member identity, e.g.

```
$ terraform import google_iap_tunnel_instance_iam_member.instance "project-name/zone-name/in
```

IAM binding imports use space-delimited identifiers; the resource in question and the role, e.g.

```
$ terraform import google_iap_tunnel_instance_iam_binding.instance "project-name/zone-name/i
```

IAM policy imports use the identifier of the resource in question, e.g.

```
$ terraform import google_iap_tunnel_instance_iam_policy.instance project-name/zone-name/ins
```

» IAM policy for IAP Tunnel Instance

Warning: These resources are in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta

resources.

Three different resources help you manage your IAM policy for IAP Tunnel Instance. Each of these resources serves a different use case:

- `google_iap_tunnel_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.
- `google_iap_tunnel_instance_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 instance are preserved.
- `google_iap_tunnel_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_iap_tunnel_instance_iam_policy` **cannot** be used in conjunction with `google_iap_tunnel_instance_iam_binding` and `google_iap_tunnel_instance_iam_member` or they will fight over what your policy should be.

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

» google_iap_tunnel_instance_iam_policy

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

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

resource "google_iap_tunnel_instance_iam_policy" "instance" {
    instance  = "your-instance-name"
    policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_tunnel_instance_iam_binding

```
resource "google_iap_tunnel_instance_iam_binding" "instance" {
  instance = "your-instance-name"
  role       = "roles/compute.networkUser"
```

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

» google_iap_tunnel_instance_iam_member

```
resource "google_iap_tunnel_instance_iam_member" "instance" {
  instance = "your-instance-name"
  role       = "roles/compute.networkUser"
  member     = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name of the instance.
- `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 G Suite domain (primary, instead of alias) 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_iap_tunnel_instance_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_iap_tunnel_instance_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

- `zone` - (Optional) The zone of the instance. If unspecified, this defaults to the zone configured in the provider.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- full self link or relative link (projects/{{project}}/zones/{{zone}}/instances/{{name}})
- {{project}}/{{zone}}/{{name}}
- {{zone}}/{{name}} (project is taken from provider project)
- {{name}} (project and zone are taken from provider project)

IAM member imports use space-delimited identifiers; the resource in question, the role, and the member identity, e.g.

```
$ terraform import google_iap_tunnel_instance_iam_member.instance "project-name/zone-name/in
```

IAM binding imports use space-delimited identifiers; the resource in question and the role, e.g.

```
$ terraform import google_iap_tunnel_instance_iam_binding.instance "project-name/zone-name/i
```

IAM policy imports use the identifier of the resource in question, e.g.

```
$ terraform import google_iap_tunnel_instance_iam_policy.instance project-name/zone-name/ins
```

» IAM policy for IapAppEngineService

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

- `google_iap_app_engine_service_iam_policy`: Authoritative. Sets the IAM policy for the appengineservice and replaces any existing policy already attached.
- `google_iap_app_engine_service_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 appengineservice are preserved.

- `google_iap_app_engine_service_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the appengineservice are preserved.

Note: `google_iap_app_engine_service_iam_policy` **cannot** be used in conjunction with `google_iap_app_engine_service_iam_binding` and `google_iap_app_engine_service_iam_member` or they will fight over what your policy should be.

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

» google_iap_app_engine_service_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_app_engine_service_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}
```

```
resource "google_iap_app_engine_service_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_app_engine_service_iam_binding

```
resource "google_iap_app_engine_service_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_service_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_app_engine_service_iam_member

```
resource "google_iap_app_engine_service_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
```

```
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_service_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) Id of the App Engine application. Used to find the parent resource to bind the IAM policy to
- `service` - (Required) Service id of the App Engine application Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_app_engine_service_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_app_engine_service_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}
- {{project}}/{{appId}}/{{service}}
- {{appId}}/{{service}}
- {{service}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap appengineservice IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_app_engine_service_iam_member.editor "projects/{{project}}/iap_web/appengine-{{ap roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_app_engine_service_iam_binding.editor "projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_app_engine_service_iam_policy.editor projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapAppEngineService

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

- `google_iap_app_engine_service_iam_policy`: Authoritative. Sets the IAM policy for the appengineservice and replaces any existing policy already attached.
- `google_iap_app_engine_service_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 appengineservice are preserved.
- `google_iap_app_engine_service_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the appengineservice are preserved.

Note: `google_iap_app_engine_service_iam_policy` **cannot** be used in conjunction with `google_iap_app_engine_service_iam_binding` and `google_iap_app_engine_service_iam_member` or they will fight over what your policy should be.

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

» google_iap_app_engine_service_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_app_engine_service_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_app_engine_service_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_app_engine_service_iam_binding

```
resource "google_iap_app_engine_service_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_service_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_app_engine_service_iam_member

```
resource "google_iap_app_engine_service_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_service_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
```

```
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) Id of the App Engine application. Used to find the parent resource to bind the IAM policy to
- `service` - (Required) Service id of the App Engine application Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_app_engine_service_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_app_engine_service_iam_policy`) The policy data generated by a `google_iam_policy` data source.

- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}
- {{project}}/{{appId}}/{{service}}
- {{appId}}/{{service}}
- {{service}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap appengineservice IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_app_engine_service_iam_member.editor "projects/{{project}}/iap_web/appengine-{{ap roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_app_engine_service_iam_binding.editor "projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_app_engine_service_iam_policy.editor projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapAppEngineService

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

- `google_iap_app_engine_service_iam_policy`: Authoritative. Sets the IAM policy for the appengineservice and replaces any existing policy already attached.
- `google_iap_app_engine_service_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 appengineservice are preserved.
- `google_iap_app_engine_service_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the appengineservice are preserved.

Note: `google_iap_app_engine_service_iam_policy` **cannot** be used in conjunction with `google_iap_app_engine_service_iam_binding` and `google_iap_app_engine_service_iam_member` or they will fight over what your policy should be.

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

» google_iap_app_engine_service_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
```

```
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_app_engine_service_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_app_engine_service_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_app_engine_service_iam_binding

```
resource "google_iap_app_engine_service_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
```

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

With IAM Conditions (beta):

```
resource "google_iap_app_engine_service_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_app_engine_service_iam_member

```
resource "google_iap_app_engine_service_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_service_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) Id of the App Engine application. Used to find the parent resource to bind the IAM policy to
- `service` - (Required) Service id of the App Engine application Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_app_engine_service_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_app_engine_service_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.

- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}
- {{project}}/{{appId}}/{{service}}
- {{appId}}/{{service}}
- {{service}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap appengineservice IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_app_engine_service_iam_member.editor "projects/{{project}}/iap_web/appengine-{{ap roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_app_engine_service_iam_binding.editor "projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_app_engine_service_iam_policy.editor projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapAppEngineVersion

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

- `google_iap_app_engine_version_iam_policy`: Authoritative. Sets the IAM policy for the appengineversion and replaces any existing policy already attached.
- `google_iap_app_engine_version_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 appengineversion are preserved.
- `google_iap_app_engine_version_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the appengineversion are preserved.

Note: `google_iap_app_engine_version_iam_policy` **cannot** be used in conjunction with `google_iap_app_engine_version_iam_binding` and `google_iap_app_engine_version_iam_member` or they will fight over what your policy should be.

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

» google_iap_app_engine_version_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_app_engine_version_iam_policy" "editor" {
```

```
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_app_engine_version_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_app_engine_version_iam_binding

```
resource "google_iap_app_engine_version_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_version_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_app_engine_version_iam_member

```
resource "google_iap_app_engine_version_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_version_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) Id of the App Engine application. Used to find the parent resource to bind the IAM policy to
- `service` - (Required) Service id of the App Engine application Used to find the parent resource to bind the IAM policy to
- `version_id` - (Required) Version id of the App Engine application Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_app_engine_version_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_app_engine_version_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}/versions/{{versionId}}
- {{project}}/{{appId}}/{{service}}/{{versionId}}
- {{appId}}/{{service}}/{{versionId}}
- {{version}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap appengineversion IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_app_engine_version_iam_member.editor "projects/{{project}}/iap_web/appengine-{{ap roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_app_engine_version_iam_binding.editor "projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}/versions/{{versionId} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_app_engine_version_iam_policy.editor projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}/versions/{{versionId}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapAppEngineVersion

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

- `google_iap_app_engine_version_iam_policy`: Authoritative. Sets the IAM policy for the appengineversion and replaces any existing policy already attached.
- `google_iap_app_engine_version_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 appengineversion are preserved.
- `google_iap_app_engine_version_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the appengineversion are preserved.

Note: `google_iap_app_engine_version_iam_policy` **cannot** be used in conjunction with `google_iap_app_engine_version_iam_binding` and `google_iap_app_engine_version_iam_member` or they will fight over what your policy should be.

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

» google_iap_app_engine_version_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
```

```
  }
}

resource "google_iap_app_engine_version_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_app_engine_version_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_app_engine_version_iam_binding

```
resource "google_iap_app_engine_version_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
```

```
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_version_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_app_engine_version_iam_member

```
resource "google_iap_app_engine_version_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_version_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
```

```
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) Id of the App Engine application. Used to find the parent resource to bind the IAM policy to
- `service` - (Required) Service id of the App Engine application Used to find the parent resource to bind the IAM policy to
- `version_id` - (Required) Version id of the App Engine application Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_app_engine_version_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_app_engine_version_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}/versions/{{versionId}}
- {{project}}/{{appId}}/{{service}}/{{versionId}}
- {{appId}}/{{service}}/{{versionId}}
- {{version}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap appengineversion IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_app_engine_version_iam_member.editor "projects/{{project}}/iap_web/appengine-{{ap roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_app_engine_version_iam_binding.editor "projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}/versions/{{versionId} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_app_engine_version_iam_policy.editor projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}/versions/{{versionId}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapAppEngineVersion

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

- `google_iap_app_engine_version_iam_policy`: Authoritative. Sets the IAM policy for the appengineversion and replaces any existing policy already attached.
- `google_iap_app_engine_version_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 appengineversion are preserved.
- `google_iap_app_engine_version_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the appengineversion are preserved.

Note: `google_iap_app_engine_version_iam_policy` **cannot** be used in conjunction with `google_iap_app_engine_version_iam_binding` and `google_iap_app_engine_version_iam_member` or they will fight over what your policy should be.

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

» google_iap_app_engine_version_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
```

```
  }
}

resource "google_iap_app_engine_version_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_app_engine_version_iam_policy" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_app_engine_version_iam_binding

```
resource "google_iap_app_engine_version_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
```

```
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_version_iam_binding" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_app_engine_version_iam_member

```
resource "google_iap_app_engine_version_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_app_engine_version_iam_member" "editor" {
  project = "${google_app_engine_standard_app_version.version.project}"
  app_id = "${google_app_engine_standard_app_version.version.project}"
  service = "${google_app_engine_standard_app_version.version.service}"
  version_id = "${google_app_engine_standard_app_version.version.version_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
```

```
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) Id of the App Engine application. Used to find the parent resource to bind the IAM policy to
- `service` - (Required) Service id of the App Engine application Used to find the parent resource to bind the IAM policy to
- `version_id` - (Required) Version id of the App Engine application Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_app_engine_version_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_app_engine_version_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}/versions/{{versionId}}
- {{project}}/{{appId}}/{{service}}/{{versionId}}
- {{appId}}/{{service}}/{{versionId}}
- {{version}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap appengineversion IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_app_engine_version_iam_member.editor "projects/{{project}}/iap_web/appengine-{{ap roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_app_engine_version_iam_binding.editor "projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}/versions/{{versionId} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_app_engine_version_iam_policy.editor projects/{{project}}/iap_web/appengine-{{appId}}/services/{{service}}/versions/{{versionId}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWebBackendService

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

- `google_iap_web_backend_service_iam_policy`: Authoritative. Sets the IAM policy for the webbackendservice and replaces any existing policy already attached.
- `google_iap_web_backend_service_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 webbackendservice are preserved.
- `google_iap_web_backend_service_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the webbackendservice are preserved.

Note: `google_iap_web_backend_service_iam_policy` **cannot** be used in conjunction with `google_iap_web_backend_service_iam_binding` and `google_iap_web_backend_service_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_backend_service_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
```

```
    ]
  }
}

resource "google_iap_web_backend_service_iam_policy" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_backend_service_iam_policy" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_backend_service_iam_binding

```
resource "google_iap_web_backend_service_iam_binding" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_backend_service_iam_binding" "editor" {
```

```
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_backend_service_iam_member

```
resource "google_iap_web_backend_service_iam_member" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_backend_service_iam_member" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `web_backend_service` - (Required) Used to find the parent resource to bind the IAM policy to

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_backend_service_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_web_backend_service_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/compute/services/{{name}}
- {{project}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap webbackendservice IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_backend_service_iam_member.editor "projects/{{project}}/iap_web/compute/servi roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_backend_service_iam_binding.editor "projects/{{project}}/iap_web/compute/services/{{web_backend_service}} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_backend_service_iam_policy.editor projects/{{project}}/iap_web/compute/services/{{web_backend_service}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWebBackendService

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

- `google_iap_web_backend_service_iam_policy`: Authoritative. Sets the IAM policy for the webbackendservice and replaces any existing policy already attached.
- `google_iap_web_backend_service_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 webbackendservice are preserved.
- `google_iap_web_backend_service_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the webbackendservice are preserved.

Note: `google_iap_web_backend_service_iam_policy` **cannot** be used in conjunction with `google_iap_web_backend_service_iam_binding` and `google_iap_web_backend_service_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_backend_service_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_web_backend_service_iam_policy" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
```

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

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_backend_service_iam_policy" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_backend_service_iam_binding

```
resource "google_iap_web_backend_service_iam_binding" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_backend_service_iam_binding" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_backend_service_iam_member

```
resource "google_iap_web_backend_service_iam_member" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_backend_service_iam_member" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `web_backend_service` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_backend_service_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_web_backend_service_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/compute/services/{{name}}
- {{project}}/{{name}}

- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap webbackendservice IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_backend_service_iam_member.editor "projects/{{project}}/iap_web/compute/servi roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_backend_service_iam_binding.editor "projects/{{project}}/iap_web/compute/services/{{web_backend_service}} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_backend_service_iam_policy.editor projects/{{project}}/iap_web/compute/services/{{web_backend_service}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWebBackendService

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

- `google_iap_web_backend_service_iam_policy`: Authoritative. Sets the IAM policy for the webbackendservice and replaces any existing policy already attached.
- `google_iap_web_backend_service_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 webbackendservice are preserved.
- `google_iap_web_backend_service_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the webbackendservice are preserved.

Note: `google_iap_web_backend_service_iam_policy` **cannot** be used in conjunction with `google_iap_web_backend_service_iam_binding` and `google_iap_web_backend_service_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_backend_service_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_web_backend_service_iam_policy" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_backend_service_iam_policy" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
```

```
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_backend_service_iam_binding

```
resource "google_iap_web_backend_service_iam_binding" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_backend_service_iam_binding" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_backend_service_iam_member

```
resource "google_iap_web_backend_service_iam_member" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_backend_service_iam_member" "editor" {
  project = "${google_compute_backend_service.default.project}"
  web_backend_service = "${google_compute_backend_service.default.name}"
  role = "roles/iap.httpsResourceAccessor"
```

```
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `web_backend_service` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_backend_service_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_web_backend_service_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/compute/services/{{name}}
- {{project}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap webbackendservice IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_backend_service_iam_member.editor "projects/{{project}}/iap_web/compute/servi roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_backend_service_iam_binding.editor`

```
"projects/{{project}}/iap_web/compute/services/{{web_backend_service}}
roles/iap.httpsResourceAccessor"
```

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_backend_service_iam_policy.editor projects/{{project}}/iap_web/compute/services/{{web_backend_service}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWeb

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

- `google_iap_web_iam_policy`: Authoritative. Sets the IAM policy for the web and replaces any existing policy already attached.
- `google_iap_web_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 web are preserved.
- `google_iap_web_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the web are preserved.

Note: `google_iap_web_iam_policy` **cannot** be used in conjunction with `google_iap_web_iam_binding` and `google_iap_web_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
```

```
  }
}

resource "google_iap_web_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_iam_binding

```
resource "google_iap_web_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
```

```
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_iam_member

```
resource "google_iap_web_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_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_iap_web_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web
- {{project}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap web IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_iam_member.editor "projects/{{project}}/iap_web roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_iam_binding.editor "projects/{{project}}/iap_web roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_iam_policy.editor projects/{{project}}/iap_web`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWeb

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

- `google_iap_web_iam_policy`: Authoritative. Sets the IAM policy for the web and replaces any existing policy already attached.
- `google_iap_web_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 web are preserved.
- `google_iap_web_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the web are preserved.

Note: `google_iap_web_iam_policy` **cannot** be used in conjunction with `google_iap_web_iam_binding` and `google_iap_web_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_web_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_iam_binding

```
resource "google_iap_web_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_iam_member

```
resource "google_iap_web_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_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_iap_web_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web
- {{project}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap web IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_iam_member.editor "projects/{{project}}/iap_web roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_iam_binding.editor "projects/{{project}}/iap_web roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_iam_policy.editor projects/{{project}}/iap_web`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWeb

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

- `google_iap_web_iam_policy`: Authoritative. Sets the IAM policy for the web and replaces any existing policy already attached.
- `google_iap_web_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 web are preserved.
- `google_iap_web_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the web are preserved.

Note: `google_iap_web_iam_policy` **cannot** be used in conjunction with `google_iap_web_iam_binding` and `google_iap_web_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_web_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
```

```
    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_iam_binding

```
resource "google_iap_web_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_iam_member

```
resource "google_iap_web_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
```

```
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_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_iap_web_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web
- {{project}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap web IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_iam_member.editor "projects/{{project}}/iap_web roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_iam_binding.editor "projects/{{project}}/iap_web roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_iam_policy.editor projects/{{project}}/iap_web`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWebTypeAppEngine

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

- `google_iap_web_type_app_engine_iam_policy`: Authoritative. Sets the IAM policy for the webtypeappengine and replaces any existing policy already attached.
- `google_iap_web_type_app_engine_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 webtypeappengine are preserved.
- `google_iap_web_type_app_engine_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the webtypeappengine are preserved.

Note: `google_iap_web_type_app_engine_iam_policy` **cannot** be used in conjunction with `google_iap_web_type_app_engine_iam_binding` and `google_iap_web_type_app_engine_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_type_app_engine_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
```

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

resource "google_iap_web_type_app_engine_iam_policy" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_type_app_engine_iam_policy" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_type_app_engine_iam_binding

```
resource "google_iap_web_type_app_engine_iam_binding" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_app_engine_iam_binding" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_type_app_engine_iam_member

```
resource "google_iap_web_type_app_engine_iam_member" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_app_engine_iam_member" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) Id of the App Engine application. Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_type_app_engine_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_web_type_app_engine_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition

is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/appengine-{{appId}}
- {{project}}/{{appId}}
- {{appId}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap webtypeappengine IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_type_app_engine_iam_member.editor "projects/{{project}}/iap_web/appengine-{{a roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_type_app_engine_iam_binding.editor "projects/{{project}}/iap_web/appengine-{{appId}} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_type_app_engine_iam_policy.editor projects/{{project}}/iap_web/appengine-{{appId}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWebTypeAppEngine

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

- `google_iap_web_type_app_engine_iam_policy`: Authoritative. Sets the IAM policy for the webtypeappengine and replaces any existing policy already attached.
- `google_iap_web_type_app_engine_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 webtypeappengine are preserved.
- `google_iap_web_type_app_engine_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the webtypeappengine are preserved.

Note: `google_iap_web_type_app_engine_iam_policy` **cannot** be used in conjunction with `google_iap_web_type_app_engine_iam_binding` and `google_iap_web_type_app_engine_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_type_app_engine_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_web_type_app_engine_iam_policy" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
```

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

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_type_app_engine_iam_policy" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_type_app_engine_iam_binding

```
resource "google_iap_web_type_app_engine_iam_binding" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_app_engine_iam_binding" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_type_app_engine_iam_member

```
resource "google_iap_web_type_app_engine_iam_member" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_app_engine_iam_member" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) Id of the App Engine application. Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_type_app_engine_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_web_type_app_engine_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/appengine-{{appId}}
- {{project}}/{{appId}}

- {{appId}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap webtypeappengine IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_type_app_engine_iam_member.editor "projects/{{project}}/iap_web/appengine-{{a roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_type_app_engine_iam_binding.editor "projects/{{project}}/iap_web/appengine-{{appId}} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_type_app_engine_iam_policy.editor projects/{{project}}/iap_web/appengine-{{appId}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWebTypeAppEngine

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

- `google_iap_web_type_app_engine_iam_policy`: Authoritative. Sets the IAM policy for the webtypeappengine and replaces any existing policy already attached.
- `google_iap_web_type_app_engine_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 webtypeappengine are preserved.
- `google_iap_web_type_app_engine_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the webtypeappengine are preserved.

Note: `google_iap_web_type_app_engine_iam_policy` **cannot** be used in conjunction with `google_iap_web_type_app_engine_iam_binding` and

google_iap_web_type_app_engine_iam_member or they will fight over what your policy should be.

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

» google_iap_web_type_app_engine_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_web_type_app_engine_iam_policy" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_type_app_engine_iam_policy" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_type_app_engine_iam_binding

```
resource "google_iap_web_type_app_engine_iam_binding" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_app_engine_iam_binding" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_type_app_engine_iam_member

```
resource "google_iap_web_type_app_engine_iam_member" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_app_engine_iam_member" "editor" {
  project = "${google_app_engine_application.app.project}"
  app_id = "${google_app_engine_application.app.app_id}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
```

```
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) Id of the App Engine application. Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_type_app_engine_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_iap_web_type_app_engine_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/appengine-{{appId}}
- {{project}}/{{appId}}
- {{appId}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap webtypeappengine IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_type_app_engine_iam_member.editor "projects/{{project}}/iap_web/appengine-{{a roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_type_app_engine_iam_binding.editor "projects/{{project}}/iap_web/appengine-{{appId}} roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_type_app_engine_iam_policy.editor projects/{{project}}/iap_web/appengine-{{appId}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWebTypeCompute

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

- `google_iap_web_type_compute_iam_policy`: Authoritative. Sets the IAM policy for the webtypecompute and replaces any existing policy already attached.
- `google_iap_web_type_compute_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 webtypecompute are preserved.
- `google_iap_web_type_compute_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the webtypecompute are preserved.

Note: `google_iap_web_type_compute_iam_policy` **cannot** be used in conjunction with `google_iap_web_type_compute_iam_binding` and `google_iap_web_type_compute_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_type_compute_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_web_type_compute_iam_policy" "editor" {
```

```
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_type_compute_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_type_compute_iam_binding

```
resource "google_iap_web_type_compute_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_compute_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
```

```
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_type_compute_iam_member

```
resource "google_iap_web_type_compute_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_compute_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_type_compute_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_iap_web_type_compute_iam_policy`) The policy data generated by a `google_iam_policy` data source.

- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.

- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.

- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/compute
- {{project}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap webtypecompute IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_type_compute_iam_member.editor "projects/{{project}}/iap_web/compute roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_type_compute_iam_binding.editor "projects/{{project}}/iap_web/compute roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_type_compute_iam_policy.editor projects/{{project}}/iap_web/compute`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWebTypeCompute

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

- `google_iap_web_type_compute_iam_policy`: Authoritative. Sets the IAM policy for the webtypecompute and replaces any existing policy already attached.
- `google_iap_web_type_compute_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 webtypecompute are preserved.
- `google_iap_web_type_compute_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the webtypecompute are preserved.

Note: `google_iap_web_type_compute_iam_policy` **cannot** be used in conjunction with `google_iap_web_type_compute_iam_binding` and

google_iap_web_type_compute_iam_member or they will fight over what your policy should be.

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

» google_iap_web_type_compute_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_web_type_compute_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_type_compute_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_type_compute_iam_binding

```
resource "google_iap_web_type_compute_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_compute_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_type_compute_iam_member

```
resource "google_iap_web_type_compute_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_compute_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_type_compute_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_iap_web_type_compute_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/compute
- {{project}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap webtypecompute IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_type_compute_iam_member.editor "projects/{{project}}/iap_web/compute roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_type_compute_iam_binding.editor "projects/{{project}}/iap_web/compute roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_type_compute_iam_policy.editor projects/{{project}}/iap_web/compute`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for IapWebTypeCompute

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

- `google_iap_web_type_compute_iam_policy`: Authoritative. Sets the IAM policy for the webtypecompute and replaces any existing policy already attached.
- `google_iap_web_type_compute_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 webtypecompute are preserved.
- `google_iap_web_type_compute_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the webtypecompute are preserved.

Note: `google_iap_web_type_compute_iam_policy` **cannot** be used in conjunction with `google_iap_web_type_compute_iam_binding` and `google_iap_web_type_compute_iam_member` or they will fight over what your policy should be.

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

» google_iap_web_type_compute_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_iap_web_type_compute_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/iap.httpsResourceAccessor"
    members = [
      "user:jane@example.com",
```

```
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_iap_web_type_compute_iam_policy" "editor" {
  project = "${google_project_service.project_service.project}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_iap_web_type_compute_iam_binding

```
resource "google_iap_web_type_compute_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_compute_iam_binding" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_iap_web_type_compute_iam_member

```
resource "google_iap_web_type_compute_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
```

```
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_iap_web_type_compute_iam_member" "editor" {
  project = "${google_project_service.project_service.project}"
  role = "roles/iap.httpsResourceAccessor"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_iap_web_type_compute_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_iap_web_type_compute_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/iap_web/compute
- {{project}}

Any variables not passed in the import command will be taken from the provider configuration.

Iap webtypecompute IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_iap_web_type_compute_iam_member.editor "projects/{{project}}/iap_web/compute roles/iap.httpsResourceAccessor jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_iap_web_type_compute_iam_binding.editor "projects/{{project}}/iap_web/compute roles/iap.httpsResourceAccessor"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_iap_web_type_compute_iam_policy.editor projects/{{project}}/iap_web/compute`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_identity_platform_default_supported_idp_config

Configurations options for authenticating with a the standard set of Identity Toolkit-trusted IDPs.

You must enable the ((Google Identity Platform)[https://console.cloud.google.com/marketplace/details/google-cloud-platform/customer-identity%5D) in the marketplace prior to using this resource.

» Example Usage - Identity Platform Default Supported Idp Config Basic

```
resource "google_identity_platform_default_supported_idp_config" "idp_config" {
  enabled       = true
  client_id     = "playgames.google.com"
  client_secret = "secret"
}
```

» Argument Reference

The following arguments are supported:

- `client_id` - (Required) OAuth client ID
- `client_secret` - (Required) OAuth client secret

- `enabled` - (Optional) If this IDP allows the user to sign in

- `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` - The name of the DefaultSupportedIdpConfig 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

DefaultSupportedIdpConfig can be imported using any of these accepted formats:

```
$ terraform import google_identity_platform_default_supported_idp_config.default projects/{
$ terraform import google_identity_platform_default_supported_idp_config.default {{project}}
$ terraform import google_identity_platform_default_supported_idp_config.default {{client_id
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_identity_platform_inbound_saml_config

Inbound SAML configuration for a Identity Toolkit project.

You must enable the ((Google Identity Platform)[https://console.cloud.google.com/marketplace/details/google-cloud-platform/customer-identity%5D) in the marketplace prior to using this resource.

» Example Usage - Identity Platform Inbound Saml Config Basic

```
resource "google_identity_platform_inbound_saml_config" "saml_config" {
  name         = "saml.tf-config"
  display_name = "Display Name"
  idp_config {
    idp_entity_id = "tf-idp"
    sign_request  = true
    sso_url       = "example.com"
    idp_certificates {
      x509_certificate = file("test-fixtures/rsa_cert.pem")
    }
  }

  sp_config {
    sp_entity_id = "tf-sp"
    callback_uri = "https://example.com"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the InboundSamlConfig resource. Must start with 'saml.' and can only have alphanumeric characters, hyphens, underscores or periods. The part after 'saml.' must also start with a lowercase letter, end with an alphanumeric character, and have at least 2 characters.
- `display_name` - (Required) Human friendly display name.
- `idp_config` - (Required) SAML IdP configuration when the project acts as the relying party Structure is documented below.
- `sp_config` - (Required) SAML SP (Service Provider) configuration when the project acts as the relying party to receive and accept an authentication assertion issued by a SAML identity provider. Structure is documented below.

The `idp_config` block supports:

- `idp_entity_id` - (Required) Unique identifier for all SAML entities
- `sso_url` - (Required) URL to send Authentication request to.
- `sign_request` - (Optional) Indicates if outbounding SAMLRequest should be signed.
- `idp_certificates` - (Required) The IdP's certificate data to verify the signature in the SAMLResponse issued by the IDP. Structure is documented below.

The `idp_certificates` block supports:

- `x509_certificate` - (Optional) The IdP's x509 certificate.

The `sp_config` block supports:

- `sp_entity_id` - (Optional) Unique identifier for all SAML entities.
- `callback_uri` - (Optional) Callback URI where responses from IDP are handled. Must start with `https://`.
- `sp_certificates` - The IDP's certificate data to verify the signature in the SAMLResponse issued by the IDP. Structure is documented below.

The `sp_certificates` block contains:

- `x509_certificate` - The x509 certificate

- `enabled` - (Optional) If this config allows users to sign in with the provider.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

» 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

InboundSamlConfig can be imported using any of these accepted formats:

```
$ terraform import google_identity_platform_inbound_saml_config.default projects/{{project}}
$ terraform import google_identity_platform_inbound_saml_config.default {{project}}/{{name}}
$ terraform import google_identity_platform_inbound_saml_config.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_identity_platform_oauth_idp_config

OIDC IdP configuration for a Identity Toolkit project.

You must enable the ((Google Identity Platform)[https://console.cloud.google.com/marketplace/details/google-cloud-platform/customer-identity%5D) in the marketplace prior to using this resource.

» Example Usage - Identity Platform Oauth Idp Config Basic

```
resource "google_identity_platform_oauth_idp_config" "oauth_idp_config" {
  name          = "oidc.oauth-idp-config"
  display_name  = "Display Name"
  client_id     = "client-id"
  issuer        = "issuer"
  enabled       = true
  client_secret = "secret"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the OauthIdpConfig. Must start with `oidc.`.

- `issuer` - (Required) For OIDC Idps, the issuer identifier.
- `client_id` - (Required) The client id of an OAuth client.

- `display_name` - (Optional) Human friendly display name.
- `enabled` - (Optional) If this config allows users to sign in with the provider.
- `client_secret` - (Optional) The client secret of the OAuth client, to enable OIDC code flow.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

» 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

OauthIdpConfig can be imported using any of these accepted formats:

```
$ terraform import google_identity_platform_oauth_idp_config.default projects/{{project}}/oa
$ terraform import google_identity_platform_oauth_idp_config.default {{project}}/{{name}}
$ terraform import google_identity_platform_oauth_idp_config.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_identity_platform_tenant

Tenant configuration in a multi-tenant project.

You must enable the ((Google Identity Platform)[https://console.cloud.google.com/marketplace/details/google-cloud-platform/customer-identity%5D) in the marketplace prior to using this resource.

You must ((enable multi-tenancy)[https://cloud.google.com/identity-platform/docs/multi-tenancy-quickstart%5D) via the Cloud Console prior to creating tenants.

» Example Usage - Identity Platform Tenant Basic

```
resource "google_identity_platform_tenant" "tenant" {
  display_name          = "tenant"
  allow_password_signup = true
}
```

» Argument Reference

The following arguments are supported:

- `display_name` - (Required) Human friendly display name of the tenant.

- `allow_password_signup` - (Optional) Whether to allow email/password user authentication.
- `enable_email_link_signin` - (Optional) Whether to enable email link user authentication.
- `disable_auth` - (Optional) Whether authentication is disabled for the tenant. If true, the users under the disabled tenant are not allowed to sign-in. Admins of the disabled tenant are not able to manage its users.
- `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` - The name of the tenant that is generated by the server

» 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

Tenant can be imported using any of these accepted formats:

```
$ terraform import google_identity_platform_tenant.default projects/{{project}}/tenants/{{na
$ terraform import google_identity_platform_tenant.default {{project}}/{{name}}
$ terraform import google_identity_platform_tenant.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_identity_platform_tenant_default_supported_idp_confi

Configurations options for the tenant for authenticating with a the standard set of Identity Toolkit-trusted IDPs.

You must enable the ((Google Identity Platform)[https://console.cloud.google.com/marketplace/details/google-cloud-platform/customer-identity%5D) in the marketplace prior to using this resource.

» Example Usage - Identity Platform Tenant Default Supported Idp Config Basic

```
resource "google_identity_platform_tenant" "tenant" {
  display_name  = "tenant"
```

```
}

resource "google_identity_platform_tenant_default_supported_idp_config" "idp_config" {
  enabled       = true
  tenant        = google_identity_platform_tenant.tenant.name
  client_id     = "playgames.google.com"
  client_secret = "secret"
}
```

» Argument Reference

The following arguments are supported:

- `tenant` - (Required) The name of the tenant where this DefaultSupportedIdpConfig resource exists
- `client_id` - (Required) OAuth client ID
- `client_secret` - (Required) OAuth client secret

- `enabled` - (Optional) If this IDP allows the user to sign in
- `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` - The name of the default supported IDP config 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

TenantDefaultSupportedIdpConfig can be imported using any of these accepted formats:

```
$ terraform import google_identity_platform_tenant_default_supported_idp_config.default proj
$ terraform import google_identity_platform_tenant_default_supported_idp_config.default {{pr
$ terraform import google_identity_platform_tenant_default_supported_idp_config.default {{te
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_identity_platform_tenant_inbound_saml_config

Inbound SAML configuration for a Identity Toolkit tenant.

You must enable the ((Google Identity Platform)[https://console.cloud.google.com/marketplace/details/google-cloud-platform/customer-identity%5D) in the marketplace prior to using this resource.

» Example Usage - Identity Platform Tenant Inbound Saml Config Basic

```
resource "google_identity_platform_tenant" "tenant" {
  display_name  = "tenant"
}

resource "google_identity_platform_tenant_inbound_saml_config" "tenant_saml_config" {
  name         = "saml.tf-config"
  display_name = "Display Name"
  tenant       = google_identity_platform_tenant.tenant.name
  idp_config {
    idp_entity_id = "tf-idp"
    sign_request  = true
    sso_url       = "example.com"
    idp_certificates {
      x509_certificate = file("test-fixtures/rsa_cert.pem")
    }
```

```
  }

  sp_config {
    sp_entity_id = "tf-sp"
    callback_uri = "https://example.com"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the InboundSamlConfig resource. Must start with 'saml.' and can only have alphanumeric characters, hyphens, underscores or periods. The part after 'saml.' must also start with a lowercase letter, end with an alphanumeric character, and have at least 2 characters.
- `tenant` - (Required) The name of the tenant where this inbound SAML config resource exists
- `display_name` - (Required) Human friendly display name.
- `idp_config` - (Required) SAML IdP configuration when the project acts as the relying party Structure is documented below.
- `sp_config` - (Required) SAML SP (Service Provider) configuration when the project acts as the relying party to receive and accept an authentication assertion issued by a SAML identity provider. Structure is documented below.

The `idp_config` block supports:

- `idp_entity_id` - (Required) Unique identifier for all SAML entities
- `sso_url` - (Required) URL to send Authentication request to.
- `sign_request` - (Optional) Indicates if outbounding SAMLRequest should be signed.
- `idp_certificates` - (Required) The IDP's certificate data to verify the signature in the SAMLResponse issued by the IDP. Structure is documented below.

The `idp_certificates` block supports:

- `x509_certificate` - (Optional) The x509 certificate

The `sp_config` block supports:

- `sp_entity_id` - (Required) Unique identifier for all SAML entities.

- `callback_uri` - (Required) Callback URI where responses from IDP are handled. Must start with `https://`.
- `sp_certificates` - The IDP's certificate data to verify the signature in the SAMLResponse issued by the IDP. Structure is documented below.

The `sp_certificates` block contains:

- `x509_certificate` - The x509 certificate

- `enabled` - (Optional) If this config allows users to sign in with the provider.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

» 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

TenantInboundSamlConfig can be imported using any of these accepted formats:

```
$ terraform import google_identity_platform_tenant_inbound_saml_config.default projects/{{pr
$ terraform import google_identity_platform_tenant_inbound_saml_config.default {{project}}/{
$ terraform import google_identity_platform_tenant_inbound_saml_config.default {{tenant}}/{{
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_identity_platform_tenant_oauth_idp_config

OIDC IdP configuration for a Identity Toolkit project within a tenant.

You must enable the ((Google Identity Platform)[https://console.cloud.google.com/marketplace/details/google-cloud-platform/customer-identity%5D) in the marketplace prior to using this resource.

» Example Usage - Identity Platform Tenant Oauth Idp Config Basic

```
resource "google_identity_platform_tenant" "tenant" {
  display_name  = "tenant"
}

resource "google_identity_platform_tenant_oauth_idp_config" "tenant_oauth_idp_config" {
  name          = "oidc.oauth-idp-config"
  tenant        = google_identity_platform_tenant.tenant.name
  display_name  = "Display Name"
  client_id     = "client-id"
  issuer        = "issuer"
  enabled       = true
  client_secret = "secret"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the OauthIdpConfig. Must start with `oidc.`.
- `tenant` - (Required) The name of the tenant where this OIDC IDP configuration resource exists
- `display_name` - (Required) Human friendly display name.
- `issuer` - (Required) For OIDC Idps, the issuer identifier.
- `client_id` - (Required) The client id of an OAuth client.

- `enabled` - (Optional) If this config allows users to sign in with the provider.

- `client_secret` - (Optional) The client secret of the OAuth client, to enable OIDC code flow.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

» 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

TenantOauthIdpConfig can be imported using any of these accepted formats:

```
$ terraform import google_identity_platform_tenant_oauth_idp_config.default projects/{{proje
$ terraform import google_identity_platform_tenant_oauth_idp_config.default {{project}}/{{te
$ terraform import google_identity_platform_tenant_oauth_idp_config.default {{tenant}}/{{nam
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_ml_engine_model

Represents a machine learning solution.

A model can have multiple versions, each of which is a deployed, trained model ready to receive prediction requests. The model itself is just a container.

» Example Usage - Ml Model Basic

```
resource "google_ml_engine_model" "default" {
  name        = "default"
  description = "My model"
  regions     = ["us-central1"]
}
```

» Example Usage - Ml Model Full

```
resource "google_ml_engine_model" "default" {
  name        = "default"
  description = "My model"
  regions     = ["us-central1"]
  labels = {
    my_model = "foo"
  }
  online_prediction_logging         = true
  online_prediction_console_logging = true
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name specified for the model.

- `description` - (Optional) The description specified for the model when it was created.
- `default_version` - (Optional) The default version of the model. This version will be used to handle prediction requests that do not specify a version. Structure is documented below.
- `regions` - (Optional) The list of regions where the model is going to be deployed. Currently only one region per model is supported
- `online_prediction_logging` - (Optional) If true, online prediction access logs are sent to StackDriver Logging.

- `online_prediction_console_logging` - (Optional) If true, online prediction nodes send stderr and stdout streams to Stackdriver Logging
- `labels` - (Optional) One or more labels that you can add, to organize your models.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

The `default_version` block supports:

- `name` - (Required) The name specified for the version when it was created.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

Model can be imported using any of these accepted formats:

```
$ terraform import google_ml_engine_model.default projects/{{project}}/models/{{name}}
$ terraform import google_ml_engine_model.default {{project}}/{{name}}
$ terraform import google_ml_engine_model.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_pubsub_subscription

A named resource representing the stream of messages from a single, specific topic, to be delivered to the subscribing application.

To get more information about Subscription, see:

- API documentation
- How-to Guides
  - Managing Subscriptions

» Example Usage - Pubsub Subscription Push

```
resource "google_pubsub_topic" "example" {
  name = "example-topic"
}

resource "google_pubsub_subscription" "example" {
  name  = "example-subscription"
  topic = google_pubsub_topic.example.name

  ack_deadline_seconds = 20

  labels = {
    foo = "bar"
  }

  push_config {
    push_endpoint = "https://example.com/push"

    attributes = {
      x-goog-version = "v1"
    }
  }
}
```

» Example Usage - Pubsub Subscription Pull

```
resource "google_pubsub_topic" "example" {
  name = "example-topic"
}

resource "google_pubsub_subscription" "example" {
  name  = "example-subscription"
  topic = google_pubsub_topic.example.name

  labels = {
    foo = "bar"
  }
```

```
  # 20 minutes
  message_retention_duration = "1200s"
  retain_acked_messages      = true

  ack_deadline_seconds = 20

  expiration_policy {
    ttl = "300000.5s"
  }
}
```

» Example Usage - Pubsub Subscription Different Project

```
resource "google_pubsub_topic" "example" {
  project = "topic-project"
  name    = "example-topic"
}

resource "google_pubsub_subscription" "example" {
  project = "subscription-project"
  name    = "example-subscription"
  topic   = google_pubsub_topic.example.name
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the subscription.
- `topic` - (Required) A reference to a Topic resource.

- `labels` - (Optional) A set of key/value label pairs to assign to this Subscription.
- `push_config` - (Optional) If push delivery is used with this subscription, this field is used to configure it. An empty pushConfig signifies that the subscriber will pull and ack messages using API methods. Structure is documented below.
- `ack_deadline_seconds` - (Optional) This value is the maximum time after a subscriber receives a message before the subscriber should acknowledge the message. After message delivery but before the ack deadline expires and before the message is acknowledged, it is an outstanding message and will not be delivered again during that time (on a best-effort

basis). For pull subscriptions, this value is used as the initial value for the ack deadline. To override this value for a given message, call subscriptions.modifyAckDeadline with the corresponding ackId if using pull. The minimum custom deadline you can specify is 10 seconds. The maximum custom deadline you can specify is 600 seconds (10 minutes). If this parameter is 0, a default value of 10 seconds is used. For push delivery, this value is also used to set the request timeout for the call to the push endpoint. If the subscriber never acknowledges the message, the Pub/Sub system will eventually redeliver the message.

- `message_retention_duration` - (Optional) How long to retain unacknowledged messages in the subscription's backlog, from the moment a message is published. If retainAckedMessages is true, then this also configures the retention of acknowledged messages, and thus configures how far back in time a subscriptions.seek can be done. Defaults to 7 days. Cannot be more than 7 days (`"604800s"`) or less than 10 minutes (`"600s"`). A duration in seconds with up to nine fractional digits, terminated by 's'. Example: `"600.5s"`.

- `retain_acked_messages` - (Optional) Indicates whether to retain acknowledged messages. If `true`, then messages are not expunged from the subscription's backlog, even if they are acknowledged, until they fall out of the messageRetentionDuration window.

- `expiration_policy` - (Optional) A policy that specifies the conditions for this subscription's expiration. A subscription is considered active as long as any connected subscriber is successfully consuming messages from the subscription or is issuing operations on the subscription. If expirationPolicy is not set, a default policy with ttl of 31 days will be used. If it is set but ttl is "", the resource never expires. The minimum allowed value for expirationPolicy.ttl is 1 day. 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.

The `push_config` block supports:

- `oidc_token` - (Optional) If specified, Pub/Sub will generate and attach an OIDC JWT token as an Authorization header in the HTTP request for every pushed message. Structure is documented below.

- `push_endpoint` - (Required) A URL locating the endpoint to which messages should be pushed. For example, a Webhook endpoint might use "https://example.com/push".

- `attributes` - (Optional) Endpoint configuration attributes. Every endpoint has a set of API supported attributes that can be used to control different aspects of the message delivery. The currently supported attribute is x-goog-version, which you can use to change the format of the pushed message. This attribute indicates the version of the data expected

by the endpoint. This controls the shape of the pushed message (i.e., its fields and metadata). The endpoint version is based on the version of the Pub/Sub API. If not present during the subscriptions.create call, it will default to the version of the API used to make such call. If not present during a subscriptions.modifyPushConfig call, its value will not be changed. subscriptions.get calls will always return a valid version, even if the subscription was created without this attribute. The possible values for this attribute are:

  - v1beta1: uses the push format defined in the v1beta1 Pub/Sub API.
  - v1 or v1beta2: uses the push format defined in the v1 Pub/Sub API.

The `oidc_token` block supports:

- `service_account_email` - (Required) Service account email to be used for generating the OIDC token. The caller (for subscriptions.create, subscriptions.patch, and subscriptions.modifyPushConfig RPCs) must have the iam.serviceAccounts.actAs permission for the service account.
- `audience` - (Optional) Audience to be used when generating OIDC token. The audience claim identifies the recipients that the JWT is intended for. The audience value is a single case-sensitive string. Having multiple values (array) for the audience field is not supported. More info about the OIDC JWT token audience here: https://tools.ietf.org/html/rfc7519#section-4.1.3 Note: if not specified, the Push endpoint URL will be used.

The `expiration_policy` block supports:

- `ttl` - (Required) Specifies the "time-to-live" duration for an associated resource. The resource expires if it is not active for a period of ttl. If ttl is not set, the associated resource never expires. A duration in seconds with up to nine fractional digits, terminated by 's'. Example - "3.5s".

» Attributes Reference

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

- `path`: Path of the subscription in the format `projects/{project}/subscriptions/{name}`

» 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

Subscription can be imported using any of these accepted formats:

```
$ terraform import google_pubsub_subscription.default projects/{{project}}/subscriptions/{{
$ terraform import google_pubsub_subscription.default {{project}}/{{name}}
$ terraform import google_pubsub_subscription.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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 G Suite domain (primary, instead of alias) 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 G Suite domain (primary, instead of alias) 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 G Suite domain (primary, instead of alias) 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_pubsub_topic

A named resource to which messages are sent by publishers.

To get more information about Topic, see:

- API documentation
- How-to Guides
  - Managing Topics

» Example Usage - Pubsub Topic Basic

```
resource "google_pubsub_topic" "example" {
  name = "example-topic"

  labels = {
    foo = "bar"
  }
}
```

» Example Usage - Pubsub Topic Cmek

```
resource "google_pubsub_topic" "example" {
  name         = "example-topic"
  kms_key_name = google_kms_crypto_key.crypto_key.self_link
}

resource "google_kms_crypto_key" "crypto_key" {
  name     = "example-key"
  key_ring = google_kms_key_ring.key_ring.self_link
}

resource "google_kms_key_ring" "key_ring" {
  name     = "example-keyring"
  location = "global"
}
```

» Example Usage - Pubsub Topic Geo Restricted

```
resource "google_pubsub_topic" "example" {
  name = "example-topic"

  message_storage_policy {
    allowed_persistence_regions = [
      "europe-west3",
    ]
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the topic.

- `kms_key_name` - (Optional) The resource name of the Cloud KMS CryptoKey to be used to protect access to messages published on this topic. Your project's PubSub service account (`service-{{PROJECT_NUMBER}}@gcp-sa-pubsub.iam.gservicea` must have `roles/cloudkms.cryptoKeyEncrypterDecrypter` to use this feature. The expected format is `projects/*/locations/*/keyRings/*/cryptoKeys/*`
- `labels` - (Optional) A set of key/value label pairs to assign to this Topic.
- `message_storage_policy` - (Optional) Policy constraining the set of Google Cloud Platform regions where messages published to the topic may be stored. If not present, then no constraints are in effect. 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.

The `message_storage_policy` block supports:

- `allowed_persistence_regions` - (Required) A list of IDs of GCP regions where messages that are published to the topic may be persisted in storage. Messages published by publishers running in non-allowed GCP regions (or running outside of GCP altogether) will be routed for storage in one of the allowed regions. An empty list means that no regions are allowed, and is not a valid configuration.

» 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

Topic can be imported using any of these accepted formats:

```
$ terraform import google_pubsub_topic.default projects/{{project}}/topics/{{name}}
$ terraform import google_pubsub_topic.default {{project}}/{{name}}
$ terraform import google_pubsub_topic.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for PubsubTopic

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

resource "google_pubsub_topic_iam_policy" "editor" {
  project = "${google_pubsub_topic.example.project}"
  topic = "${google_pubsub_topic.example.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_pubsub_topic_iam_binding

```
resource "google_pubsub_topic_iam_binding" "editor" {
  project = "${google_pubsub_topic.example.project}"
  topic = "${google_pubsub_topic.example.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_pubsub_topic_iam_member

```
resource "google_pubsub_topic_iam_member" "editor" {
  project = "${google_pubsub_topic.example.project}"
  topic = "${google_pubsub_topic.example.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `topic` - (Required) Used to find the parent resource to bind the IAM policy to

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/topics/{{name}}
- {{project}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Pubsub topic IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_pubsub_topic_iam_member.editor "projects/{{project}}/topics/{{topic}} roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_pubsub_topic_iam_binding.editor "projects/{{project}}/topics/{{topic}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_pubsub_topic_iam_policy.editor projects/{{project}}/topics/{{topic}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for PubsubTopic

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

resource "google_pubsub_topic_iam_policy" "editor" {
  project = "${google_pubsub_topic.example.project}"
  topic = "${google_pubsub_topic.example.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_pubsub_topic_iam_binding

```
resource "google_pubsub_topic_iam_binding" "editor" {
  project = "${google_pubsub_topic.example.project}"
  topic = "${google_pubsub_topic.example.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_pubsub_topic_iam_member

```
resource "google_pubsub_topic_iam_member" "editor" {
  project = "${google_pubsub_topic.example.project}"
  topic = "${google_pubsub_topic.example.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `topic` - (Required) Used to find the parent resource to bind the IAM policy to

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/topics/{{name}}
- {{project}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Pubsub topic IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_pubsub_topic_iam_member.editor "projects/{{project}}/topics/{{topic}} roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_pubsub_topic_iam_binding.editor "projects/{{project}}/topics/{{topic}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_pubsub_topic_iam_policy.editor projects/{{project}}/topics/{{topic}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for PubsubTopic

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

resource "google_pubsub_topic_iam_policy" "editor" {
  project = "${google_pubsub_topic.example.project}"
  topic = "${google_pubsub_topic.example.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_pubsub_topic_iam_binding

```
resource "google_pubsub_topic_iam_binding" "editor" {
  project = "${google_pubsub_topic.example.project}"
  topic = "${google_pubsub_topic.example.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_pubsub_topic_iam_member

```
resource "google_pubsub_topic_iam_member" "editor" {
  project = "${google_pubsub_topic.example.project}"
  topic = "${google_pubsub_topic.example.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `topic` - (Required) Used to find the parent resource to bind the IAM policy to

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/topics/{{name}}
- {{project}}/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Pubsub topic IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_pubsub_topic_iam_member.editor "projects/{{project}}/topics/{{topic}} roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_pubsub_topic_iam_binding.editor "projects/{{project}}/topics/{{topic}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_pubsub_topic_iam_policy.editor projects/{{project}}/topics/{{topic}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_redis_instance

A Google Cloud Redis instance.

To get more information about Instance, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Redis Instance Basic

```
resource "google_redis_instance" "cache" {
  name           = "memory-cache"
  memory_size_gb = 1
```

```
}
```

» Example Usage - Redis Instance Full

```
resource "google_redis_instance" "cache" {
  name           = "ha-memory-cache"
  tier           = "STANDARD_HA"
  memory_size_gb = 1

  location_id             = "us-central1-a"
  alternative_location_id = "us-central1-f"

  authorized_network = google_compute_network.auto-network.self_link

  redis_version     = "REDIS_3_2"
  display_name      = "Terraform Test Instance"
  reserved_ip_range = "192.168.0.0/29"

  labels = {
    my_key    = "my_val"
    other_key = "other_val"
  }
}

resource "google_compute_network" "auto-network" {
  name = "authorized-network"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The ID of the instance or a fully qualified identifier for the instance.
- `memory_size_gb` - (Required) Redis memory size in GiB.

- `alternative_location_id` - (Optional) Only applicable to STANDARD_HA tier which protects the instance against zonal failures by

provisioning it across two zones. If provided, it must be a different zone from the one provided in [locationId].

- `authorized_network` - (Optional) The full name of the Google Compute Engine network to which the instance is connected. If left unspecified, the default network will be used.
- `display_name` - (Optional) An arbitrary and optional user-provided name for the instance.
- `labels` - (Optional) Resource labels to represent user provided metadata.
- `redis_configs` - (Optional) Redis configuration parameters, according to http://redis.io/topics/config. Please check Memorystore documentation for the list of supported parameters: https://cloud.google.com/memorystore/docs/redis/reference/rest/v1/projects.locations.instances#Instance.FIELDS.redis_configs
- `location_id` - (Optional) The zone where the instance will be provisioned. If not provided, the service will choose a zone for the instance. For STANDARD_HA tier, instances will be created across two zones for protection against zonal failures. If [alternativeLocationId] is also provided, it must be different from [locationId].
- `redis_version` - (Optional) The version of Redis software. If not provided, latest supported version will be used. Currently, the supported values are:
  - REDIS_4_0 for Redis 4.0 compatibility
  - REDIS_3_2 for Redis 3.2 compatibility
- `reserved_ip_range` - (Optional) The CIDR range of internal addresses that are reserved for this instance. If not provided, the service will choose an unused /29 block, for example, 10.0.0.0/29 or 192.168.0.0/29. Ranges must be unique and non-overlapping with existing subnets in an authorized network.
- `tier` - (Optional) The service tier of the instance. Must be one of these values:
  - BASIC: standalone instance
  - STANDARD_HA: highly available primary/replica instances
- `region` - (Optional) The name of the Redis region of the instance.
- `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:

- `create_time` - The time the instance was created in RFC3339 UTC "Zulu" format, accurate to nanoseconds.
- `current_location_id` - The current zone where the Redis endpoint is placed. For Basic Tier instances, this will always be the same as the [locationId] provided by the user at creation time. For Standard Tier instances, this can be either [locationId] or [alternativeLocationId] and can change after a failover event.
- `host` - Hostname or IP address of the exposed Redis endpoint used by clients to connect to the service.
- `port` - The port number of the exposed Redis endpoint.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 10 minutes.
- `update` - Default is 10 minutes.
- `delete` - Default is 10 minutes.

» Import

Instance can be imported using any of these accepted formats:

```
$ terraform import google_redis_instance.default projects/{{project}}/locations/{{region}}/i
$ terraform import google_redis_instance.default {{project}}/{{region}}/{{name}}
$ terraform import google_redis_instance.default {{region}}/{{name}}
$ terraform import google_redis_instance.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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.

» Import

Runtime Configs can be imported using the `name` or full config name, e.g.

```
$ terraform import google_runtimeconfig_config.myconfig myconfig
```

```
$ terraform import google_runtimeconfig_config.myconfig projects/my-gcp-project/configs/myco
```

When importing using only the name, the provider project must be set.

» 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".

» Import

Runtime Config Variables can be imported using the `name` or full variable name, e.g.

```
$ terraform import google_runtimeconfig_variable.myvariable myconfig/myvariable
```

```
$ terraform import google_runtimeconfig_variable.myvariable projects/my-gcp-project/configs/
```

When importing using only the name, the provider project must be set.

» google_scc_source

A Cloud Security Command Center's (Cloud SCC) finding source. A finding source is an entity or a mechanism that can produce a finding. A source is like a container of findings that come from the same scanner, logger, monitor, etc.

To get more information about Source, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Scc Source Basic

```
resource "google_scc_source" "custom_source" {
  display_name = "My Source"
  organization = "123456789"
  description  = "My custom Cloud Security Command Center Finding Source"
}
```

» Argument Reference

The following arguments are supported:

- `display_name` - (Required) The source's display name. A source's display name must be unique amongst its siblings, for example, two sources with the same parent can't 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 32 characters.
- `organization` - (Required) The organization whose Cloud Security Command Center the Source lives in.

- `description` - (Optional) The description of the source (max of 1024 characters).

» Attributes Reference

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

- `name` - The resource name of this source, in the format `organizations/{{organization}}/sources/{{so`

» 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

Source can be imported using any of these accepted formats:

```
$ terraform import google_scc_source.default organizations/{{organization}}/sources/{{name}}
$ terraform import google_scc_source.default {{organization}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» google_service_networking_connection

Manages a private VPC connection with a GCP service provider. For more information see the official documentation and API.

» Example usage

```
resource "google_compute_network" "peering_network" {
  name = "peering-network"
}

resource "google_compute_global_address" "private_ip_alloc" {
  name          = "private-ip-alloc"
  purpose       = "VPC_PEERING"
  address_type  = "INTERNAL"
  prefix_length = 16
  network       = google_compute_network.peering_network.self_link
}

resource "google_service_networking_connection" "foobar" {
  network                 = google_compute_network.peering_network.self_link
  service                 = "servicenetworking.googleapis.com"
  reserved_peering_ranges = [google_compute_global_address.private_ip_alloc.name]
}
```

» Argument Reference

The following arguments are supported:

- `network` - (Required) Name of VPC network connected with service producers using VPC peering.
- `service` - (Required) Provider peering service that is managing peering connectivity for a service provider organization. For Google services that support this functionality it is 'servicenetworking.googleapis.com'.
- `reserved_peering_ranges` - (Required) Named IP address range(s) of PEERING type reserved for this service provider. Note that invoking this method with a different range when connection is already established will not reallocate already provisioned service producer subnetworks.

» google_sourcerepo_repository

A repository (or repo) is a Git repository storing versioned source content.

To get more information about Repository, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Sourcerepo Repository Basic

```
resource "google_sourcerepo_repository" "my-repo" {
  name = "my/repository"
}
```

» Example Usage - Sourcerepo Repository Full

```
resource "google_service_account" "test-account" {
  account_id   = "my-account"
  display_name = "Test Service Account"
}

resource "google_pubsub_topic" "topic" {
  name     = "my-topic"
}

resource "google_sourcerepo_repository" "my-repo" {
  name = "my-repository"
  pubsub_configs {
      topic = google_pubsub_topic.topic.id
      message_format = "JSON"
      service_account_email = google_service_account.test-account.email
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Resource name of the repository, of the form `{{repo}}`. The repo name may contain slashes. eg, `name/with/slash`

- `pubsub_configs` - (Optional) How this repository publishes a change in the repository through Cloud Pub/Sub. Keyed by the topic names. 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.

The `pubsub_configs` block supports:

- `topic` - (Required) The identifier for this object. Format specified above.
- `message_format` - (Required) The format of the Cloud Pub/Sub messages.
  - PROTOBUF: The message payload is a serialized protocol buffer of SourceRepoEvent.
  - JSON: The message payload is a JSON string of SourceRepoEvent.
- `service_account_email` - (Optional) Email address of the service account used for publishing Cloud Pub/Sub messages. This service account needs to be in the same project as the PubsubConfig. When added, the caller needs to have iam.serviceAccounts.actAs permission on this service account. If unspecified, it defaults to the compute engine default service account.

» Attributes Reference

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

- `url` - URL to clone the repository from Google Cloud Source Repositories.
- `size` - The disk usage of the repo, in bytes.

» 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

Repository can be imported using any of these accepted formats:

```
$ terraform import google_sourcerepo_repository.default projects/{{project}}/repos/{{name}}
$ terraform import google_sourcerepo_repository.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for SourceRepoRepository

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

- `google_sourcerepo_repository_iam_policy`: Authoritative. Sets the IAM policy for the repository and replaces any existing policy already attached.
- `google_sourcerepo_repository_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 repository are preserved.
- `google_sourcerepo_repository_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the repository are preserved.

Note: `google_sourcerepo_repository_iam_policy` **cannot** be used in conjunction with `google_sourcerepo_repository_iam_binding` and `google_sourcerepo_repository_iam_member` or they will fight over what your policy should be.

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

» google_sourcerepo_repository_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
```

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

resource "google_sourcerepo_repository_iam_policy" "editor" {
  project = "${google_sourcerepo_repository.my-repo.project}"
  repository = "${google_sourcerepo_repository.my-repo.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_sourcerepo_repository_iam_binding

```
resource "google_sourcerepo_repository_iam_binding" "editor" {
  project = "${google_sourcerepo_repository.my-repo.project}"
  repository = "${google_sourcerepo_repository.my-repo.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_sourcerepo_repository_iam_member

```
resource "google_sourcerepo_repository_iam_member" "editor" {
  project = "${google_sourcerepo_repository.my-repo.project}"
  repository = "${google_sourcerepo_repository.my-repo.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `repository` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_sourcerepo_repository_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`

- `policy_data` - (Required only by `google_sourcerepo_repository_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/repos/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

SourceRepo repository IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_sourcerepo_repository_iam_member.editor "projects/{{project}}/repos/{{repository}} roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_sourcerepo_repository_iam_binding.editor "projects/{{project}}/repos/{{repository}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_sourcerepo_repository_iam_policy.editor projects/{{project}}/repos/{{repository}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for SourceRepoRepository

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

- `google_sourcerepo_repository_iam_policy`: Authoritative. Sets the IAM policy for the repository and replaces any existing policy already attached.
- `google_sourcerepo_repository_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 repository are preserved.
- `google_sourcerepo_repository_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the repository are preserved.

Note: `google_sourcerepo_repository_iam_policy` **cannot** be used in conjunction with `google_sourcerepo_repository_iam_binding` and `google_sourcerepo_repository_iam_member` or they will fight over what your policy should be.

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

» google_sourcerepo_repository_iam_policy

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

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

resource "google_sourcerepo_repository_iam_policy" "editor" {
  project = "${google_sourcerepo_repository.my-repo.project}"
  repository = "${google_sourcerepo_repository.my-repo.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_sourcerepo_repository_iam_binding

```
resource "google_sourcerepo_repository_iam_binding" "editor" {
  project = "${google_sourcerepo_repository.my-repo.project}"
  repository = "${google_sourcerepo_repository.my-repo.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_sourcerepo_repository_iam_member

```
resource "google_sourcerepo_repository_iam_member" "editor" {
  project = "${google_sourcerepo_repository.my-repo.project}"
  repository = "${google_sourcerepo_repository.my-repo.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `repository` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_sourcerepo_repository_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_sourcerepo_repository_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/repos/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

SourceRepo repository IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import`

`google_sourcerepo_repository_iam_member.editor "projects/{{project}}/repos/{{repository}} roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_sourcerepo_repository_iam_binding.editor "projects/{{project}}/repos/{{repository}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_sourcerepo_repository_iam_policy.editor projects/{{project}}/repos/{{repository}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for SourceRepoRepository

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

- `google_sourcerepo_repository_iam_policy`: Authoritative. Sets the IAM policy for the repository and replaces any existing policy already attached.
- `google_sourcerepo_repository_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 repository are preserved.
- `google_sourcerepo_repository_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the repository are preserved.

Note: `google_sourcerepo_repository_iam_policy` **cannot** be used in conjunction with `google_sourcerepo_repository_iam_binding` and `google_sourcerepo_repository_iam_member` or they will fight over what your policy should be.

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

» google_sourcerepo_repository_iam_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/viewer"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_sourcerepo_repository_iam_policy" "editor" {
  project = "${google_sourcerepo_repository.my-repo.project}"
  repository = "${google_sourcerepo_repository.my-repo.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_sourcerepo_repository_iam_binding

```
resource "google_sourcerepo_repository_iam_binding" "editor" {
  project = "${google_sourcerepo_repository.my-repo.project}"
  repository = "${google_sourcerepo_repository.my-repo.name}"
  role = "roles/viewer"
  members = [
    "user:jane@example.com",
  ]
}
```

» google_sourcerepo_repository_iam_member

```
resource "google_sourcerepo_repository_iam_member" "editor" {
  project = "${google_sourcerepo_repository.my-repo.project}"
  repository = "${google_sourcerepo_repository.my-repo.name}"
  role = "roles/viewer"
  member = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `repository` - (Required) Used to find the parent resource to bind the IAM policy to

- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_sourcerepo_repository_iam_binding` can be used per role. Note that custom roles must be of the format `[projects|organizations]/{parent-name}/roles/{role`
- `policy_data` - (Required only by `google_sourcerepo_repository_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 IAM policy.

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- projects/{{project}}/repos/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

SourceRepo repository IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_sourcerepo_repository_iam_member.editor "projects/{{project}}/repos/{{repository}} roles/viewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_sourcerepo_repository_iam_binding.editor "projects/{{project}}/repos/{{repository}} roles/viewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_sourcerepo_repository_iam_policy.editor projects/{{project}}/repos/{{repository}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_spanner_database

A Cloud Spanner Database which is hosted on a Spanner instance.

To get more information about Database, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Spanner Database Basic

```
resource "google_spanner_instance" "main" {
  config       = "regional-europe-west1"
  display_name = "main-instance"
}
```

```
resource "google_spanner_database" "database" {
  instance = google_spanner_instance.main.name
  name     = "my-database"
  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:

- `name` - (Required) A unique identifier for the database, which cannot be changed after the instance is created. Values are of the form [a-z][-a-z0-9]*[a-z0-9].
- `instance` - (Required) The instance to create the database on.

- `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.
- `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:

- `state` - An explanation of the status of the database.

» Timeouts

This resource provides the following Timeouts configuration options:

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

» Import

Database can be imported using any of these accepted formats:

```
$ terraform import google_spanner_database.default projects/{{project}}/instances/{{instance
$ terraform import google_spanner_database.default instances/{{instance}}/databases/{{name}}
$ terraform import google_spanner_database.default {{project}}/{{instance}}/{{name}}
$ terraform import google_spanner_database.default {{instance}}/{{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for Spanner databases

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

- `google_spanner_database_iam_policy`: Authoritative. Sets the IAM policy for the database and replaces any existing policy already attached.

Warning: It's entirely possibly to lock yourself out of your database using `google_spanner_database_iam_policy`. Any permissions granted by default will be removed unless you include them in your config.

- `google_spanner_database_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 database are preserved.
- `google_spanner_database_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the database are preserved.

Note: `google_spanner_database_iam_policy` **cannot** be used in conjunction with `google_spanner_database_iam_binding` and `google_spanner_database_iam_member` or they will fight over what your policy should be.

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

» google_spanner_database_iam_policy

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

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

resource "google_spanner_database_iam_policy" "database" {
  instance    = "your-instance-name"
  database    = "your-database-name"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_spanner_database_iam_binding

```
resource "google_spanner_database_iam_binding" "database" {
  instance = "your-instance-name"
  database = "your-database-name"
  role     = "roles/compute.networkUser"

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

» google_spanner_database_iam_member

```
resource "google_spanner_database_iam_member" "database" {
  instance = "your-instance-name"
  database = "your-database-name"
  role     = "roles/compute.networkUser"
  member   = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `database` - (Required) The name of the Spanner database.
- `instance` - (Required) The name of the Spanner instance the database belongs 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 G Suite domain (primary, instead of alias) 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_spanner_database_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_spanner_database_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `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:

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- {{project}}/{{instance}}/{{database}}
- {{instance}}/{{database}} (project is taken from provider project)

IAM member imports use space-delimited identifiers; the resource in question, the role, and the member identity, e.g.

```
$ terraform import google_spanner_database_iam_member.database "project-name/instance-name/d
```

IAM binding imports use space-delimited identifiers; the resource in question and the role, e.g.

```
$ terraform import google_spanner_database_iam_binding.database "project-name/instance-name/
```

IAM policy imports use the identifier of the resource in question, e.g.

```
$ terraform import google_spanner_database_iam_policy.database project-name/instance-name/da
```

» IAM policy for Spanner databases

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

- `google_spanner_database_iam_policy`: Authoritative. Sets the IAM policy for the database and replaces any existing policy already attached.

Warning: It's entirely possibly to lock yourself out of your database using `google_spanner_database_iam_policy`. Any permissions granted by default will be removed unless you include them in your config.

- `google_spanner_database_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 database are preserved.
- `google_spanner_database_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the database are preserved.

Note: `google_spanner_database_iam_policy` **cannot** be used in conjunction with `google_spanner_database_iam_binding` and `google_spanner_database_iam_member` or they will fight over what your policy should be.

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

» google_spanner_database_iam_policy

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

    members = [
```

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

resource "google_spanner_database_iam_policy" "database" {
  instance    = "your-instance-name"
  database    = "your-database-name"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_spanner_database_iam_binding

```
resource "google_spanner_database_iam_binding" "database" {
  instance = "your-instance-name"
  database = "your-database-name"
  role     = "roles/compute.networkUser"

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

» google_spanner_database_iam_member

```
resource "google_spanner_database_iam_member" "database" {
  instance = "your-instance-name"
  database = "your-database-name"
  role     = "roles/compute.networkUser"
  member   = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `database` - (Required) The name of the Spanner database.
- `instance` - (Required) The name of the Spanner instance the database belongs 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 G Suite domain (primary, instead of alias) 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_spanner_database_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_spanner_database_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `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:

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- {{project}}/{{instance}}/{{database}}
- {{instance}}/{{database}} (project is taken from provider project)

IAM member imports use space-delimited identifiers; the resource in question, the role, and the member identity, e.g.

```
$ terraform import google_spanner_database_iam_member.database "project-name/instance-name/d
```

IAM binding imports use space-delimited identifiers; the resource in question and the role, e.g.

```
$ terraform import google_spanner_database_iam_binding.database "project-name/instance-name/
```

IAM policy imports use the identifier of the resource in question, e.g.

```
$ terraform import google_spanner_database_iam_policy.database project-name/instance-name/da
```

» IAM policy for Spanner databases

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

- `google_spanner_database_iam_policy`: Authoritative. Sets the IAM policy for the database and replaces any existing policy already attached.

Warning: It's entirely possibly to lock yourself out of your database using `google_spanner_database_iam_policy`. Any permissions granted by default will be removed unless you include them in your config.

- `google_spanner_database_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 database are preserved.
- `google_spanner_database_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the database are preserved.

Note: `google_spanner_database_iam_policy` **cannot** be used in conjunction with `google_spanner_database_iam_binding` and `google_spanner_database_iam_member` or they will fight over what your policy should be.

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

» google_spanner_database_iam_policy

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

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

resource "google_spanner_database_iam_policy" "database" {
  instance    = "your-instance-name"
```

```
  database    = "your-database-name"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_spanner_database_iam_binding

```
resource "google_spanner_database_iam_binding" "database" {
  instance = "your-instance-name"
  database = "your-database-name"
  role     = "roles/compute.networkUser"

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

» google_spanner_database_iam_member

```
resource "google_spanner_database_iam_member" "database" {
  instance = "your-instance-name"
  database = "your-database-name"
  role     = "roles/compute.networkUser"
  member   = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `database` - (Required) The name of the Spanner database.
- `instance` - (Required) The name of the Spanner instance the database belongs 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 G Suite domain (primary, instead of alias) 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_spanner_database_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_spanner_database_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `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:

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- {{project}}/{{instance}}/{{database}}
- {{instance}}/{{database}} (project is taken from provider project)

IAM member imports use space-delimited identifiers; the resource in question, the role, and the member identity, e.g.

```
$ terraform import google_spanner_database_iam_member.database "project-name/instance-name/d
```

IAM binding imports use space-delimited identifiers; the resource in question and the role, e.g.

```
$ terraform import google_spanner_database_iam_binding.database "project-name/instance-name/
```

IAM policy imports use the identifier of the resource in question, e.g.

```
$ terraform import google_spanner_database_iam_policy.database project-name/instance-name/da
```

» google_spanner_instance

An isolated set of Cloud Spanner resources on which databases can be hosted.

To get more information about Instance, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Spanner Instance Basic

```
resource "google_spanner_instance" "example" {
  config       = "regional-us-central1"
  display_name = "Test Spanner Instance"
  num_nodes    = 2
  labels = {
    "foo" = "bar"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A unique identifier for the instance, which cannot be changed after the instance is created. The name must be between 6 and 30 characters in length.

If not provided, a random string starting with `tf-` will be selected.

- `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. Must be unique per project and between 4 and 30 characters in length.

- `num_nodes` - (Optional) The number of nodes allocated to this instance.
- `labels` - (Optional) An object containing a list of "key": value pairs. Example: { "name": "wrench", "mass": "1.3kg", "count": "3" }.
- `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:

- `state` - Instance status: `CREATING` or `READY`.

» 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

Instance can be imported using any of these accepted formats:

```
$ terraform import google_spanner_instance.default projects/{{project}}/instances/{{name}}
$ terraform import google_spanner_instance.default {{project}}/{{name}}
$ terraform import google_spanner_instance.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for Spanner Instances

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

- `google_spanner_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.

Warning: It's entirely possibly to lock yourself out of your instance using `google_spanner_instance_iam_policy`. Any permissions granted by default will be removed unless you include them in your config.

- `google_spanner_instance_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 instance are preserved.
- `google_spanner_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_spanner_instance_iam_policy` **cannot** be used in conjunction with `google_spanner_instance_iam_binding` and `google_spanner_instance_iam_member` or they will fight over what your policy should be.

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

» google_spanner_instance_iam_policy

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

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

resource "google_spanner_instance_iam_policy" "instance" {
  instance    = "your-instance-name"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_spanner_instance_iam_binding

```
resource "google_spanner_instance_iam_binding" "instance" {
  instance = "your-instance-name"
  role     = "roles/compute.networkUser"

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

» google_spanner_instance_iam_member

```
resource "google_spanner_instance_iam_member" "instance" {
  instance = "your-instance-name"
  role     = "roles/compute.networkUser"
  member   = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name of the instance.
- `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 G Suite domain (primary, instead of alias) 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_spanner_instance_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_spanner_instance_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `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:

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- {{project}}/{{name}}
- {{name}} (project is taken from provider project)

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account, e.g.

```
$ terraform import google_spanner_instance_iam_member.instance "project-name/instance-name r
```

IAM binding imports use space-delimited identifiers; the resource in question and the role, e.g.

```
$ terraform import google_spanner_instance_iam_binding.instance "project-name/instance-name
```

IAM policy imports use the identifier of the resource in question, e.g.

```
$ terraform import google_spanner_instance_iam_policy.instance project-name/instance-name
```

» IAM policy for Spanner Instances

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

- `google_spanner_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.

Warning: It's entirely possibly to lock yourself out of your instance using `google_spanner_instance_iam_policy`. Any permissions granted by default will be removed unless you include them in your config.

- `google_spanner_instance_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 instance are preserved.
- `google_spanner_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_spanner_instance_iam_policy` **cannot** be used in conjunction with `google_spanner_instance_iam_binding` and `google_spanner_instance_iam_member` or they will fight over what your policy should be.

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

» google_spanner_instance_iam_policy

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

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

resource "google_spanner_instance_iam_policy" "instance" {
  instance    = "your-instance-name"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_spanner_instance_iam_binding

```
resource "google_spanner_instance_iam_binding" "instance" {
  instance = "your-instance-name"
  role     = "roles/compute.networkUser"

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

» google_spanner_instance_iam_member

```
resource "google_spanner_instance_iam_member" "instance" {
  instance = "your-instance-name"
  role     = "roles/compute.networkUser"
  member   = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name of the instance.
- `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 G Suite domain (primary, instead of alias) 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_spanner_instance_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_spanner_instance_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `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:

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- {{project}}/{{name}}
- {{name}} (project is taken from provider project)

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account, e.g.

```
$ terraform import google_spanner_instance_iam_member.instance "project-name/instance-name r
```

IAM binding imports use space-delimited identifiers; the resource in question and the role, e.g.

```
$ terraform import google_spanner_instance_iam_binding.instance "project-name/instance-name
```

IAM policy imports use the identifier of the resource in question, e.g.

```
$ terraform import google_spanner_instance_iam_policy.instance project-name/instance-name
```

» IAM policy for Spanner Instances

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

- `google_spanner_instance_iam_policy`: Authoritative. Sets the IAM policy for the instance and replaces any existing policy already attached.

Warning: It's entirely possibly to lock yourself out of your instance using `google_spanner_instance_iam_policy`. Any permissions granted by default will be removed unless you include them in your config.

- `google_spanner_instance_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 instance are preserved.
- `google_spanner_instance_iam_member`: Non-authoritative. Updates the IAM policy to grant a role to a new member. Other members for the role for the instance are preserved.

Note: `google_spanner_instance_iam_policy` **cannot** be used in conjunction with `google_spanner_instance_iam_binding` and `google_spanner_instance_iam_member` or they will fight over what your policy should be.

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

» google_spanner_instance_iam_policy

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

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

resource "google_spanner_instance_iam_policy" "instance" {
  instance    = "your-instance-name"
  policy_data = data.google_iam_policy.admin.policy_data
}
```

» google_spanner_instance_iam_binding

```
resource "google_spanner_instance_iam_binding" "instance" {
  instance = "your-instance-name"
  role     = "roles/compute.networkUser"

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

» google_spanner_instance_iam_member

```
resource "google_spanner_instance_iam_member" "instance" {
  instance = "your-instance-name"
  role     = "roles/compute.networkUser"
  member   = "user:jane@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `instance` - (Required) The name of the instance.
- `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 G Suite domain (primary, instead of alias) 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_spanner_instance_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_spanner_instance_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `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:

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- {{project}}/{{name}}
- {{name}} (project is taken from provider project)

IAM member imports use space-delimited identifiers; the resource in question, the role, and the account, e.g.

```
$ terraform import google_spanner_instance_iam_member.instance "project-name/instance-name r
```

IAM binding imports use space-delimited identifiers; the resource in question and the role, e.g.

```
$ terraform import google_spanner_instance_iam_binding.instance "project-name/instance-name
```

IAM policy imports use the identifier of the resource in question, e.g.

```
$ terraform import google_spanner_instance_iam_policy.instance project-name/instance-name
```

» google_sql_database

Represents a SQL database inside the Cloud SQL instance, hosted in Google's cloud.

» Example Usage - Sql Database Basic

```
resource "google_sql_database" "database" {
  name     = "my-database"
  instance = google_sql_database_instance.instance.name
}

resource "google_sql_database_instance" "instance" {
  name   = "my-database-instance"
  region = "us-central"
  settings {
    tier = "D0"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the database in the Cloud SQL instance. This does not include the project ID or instance name.

- `instance` - (Required) The name of the Cloud SQL instance. This does not include the project ID.

- `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 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 only support a value of `en_US.UTF8` at creation time.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 15 minutes.
- `update` - Default is 10 minutes.
- `delete` - Default is 10 minutes.

» Import

Database can be imported using any of these accepted formats:

```
$ terraform import google_sql_database.default projects/{{project}}/instances/{{instance}}/d
$ terraform import google_sql_database.default instances/{{instance}}/databases/{{name}}
$ terraform import google_sql_database.default {{project}}/{{instance}}/{{name}}
$ terraform import google_sql_database.default {{instance}}/{{name}}
$ terraform import google_sql_database.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_sql_database_instance

Creates a new Google SQL Database Instance. For more information, see the official documentation, or the JSON API.

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 "random_id" "db_name_suffix" {
  byte_length = 4
}

resource "google_sql_database_instance" "master" {
  name             = "master-instance-${random_id.db_name_suffix.hex}"
  database_version = "MYSQL_5_7"

  # 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_11"
  region           = "us-central1"

  settings {
    # Second-generation instance tiers are based on the machine
    # type. See argument reference below.
    tier = "db-f1-micro"
  }
}
```

» Granular restriction of network access

```
resource "google_compute_instance" "apps" {
  count        = 8
  name         = "apps-${count.index + 1}"
  machine_type = "f1-micro"

  boot_disk {
    initialize_params {
      image = "ubuntu-os-cloud/ubuntu-1804-lts"
    }
  }

  network_interface {
    network = "default"

    access_config {
      // Ephemeral IP
    }
  }
}

resource "random_id" "db_name_suffix" {
  byte_length = 4
}

locals {
  onprem = ["192.168.1.2", "192.168.2.3"]
}

resource "google_sql_database_instance" "postgres" {
  name             = "postgres-instance-${random_id.db_name_suffix.hex}"
  database_version = "POSTGRES_11"

  settings {
    tier = "db-f1-micro"

    ip_configuration {

      dynamic "authorized_networks" {
        for_each = google_compute_instance.apps
        iterator = apps

        content {
          name  = apps.value.name
```

```
          value = apps.value.network_interface.0.access_config.0.nat_ip
        }
      }

      dynamic "authorized_networks" {
        for_each = local.onprem
        iterator = onprem

        content {
          name  = "onprem-${onprem.key}"
          value = onprem.value
        }
      }
    }
  }
}
```

» Private IP Instance

NOTE: For private IP instance setup, note that the `google_sql_database_instance` does not actually interpolate values from `google_service_networking_connection`. You must explicitly add a `depends_on`reference as shown below.

```
resource "google_compute_network" "private_network" {
  provider = google-beta

  name = "private-network"
}

resource "google_compute_global_address" "private_ip_address" {
  provider = google-beta

  name          = "private-ip-address"
  purpose       = "VPC_PEERING"
  address_type  = "INTERNAL"
  prefix_length = 16
  network       = google_compute_network.private_network.self_link
}

resource "google_service_networking_connection" "private_vpc_connection" {
  provider = google-beta

  network                 = google_compute_network.private_network.self_link
  service                 = "servicenetworking.googleapis.com"
  reserved_peering_ranges = [google_compute_global_address.private_ip_address.name]
```

```
}

resource "random_id" "db_name_suffix" {
  byte_length = 4
}

resource "google_sql_database_instance" "instance" {
  provider = google-beta

  name   = "private-instance-${random_id.db_name_suffix.hex}"
  region = "us-central1"

  depends_on = [google_service_networking_connection.private_vpc_connection]

  settings {
    tier = "db-f1-micro"
    ip_configuration {
      ipv4_enabled    = false
      private_network = google_compute_network.private_network.self_link
    }
  }
}

provider "google-beta" {
  region = "us-central1"
  zone   = "us-central1-a"
}
```

» 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. A valid region must be provided to use this resource. If a region is not provided in the resource definition, the provider region will be used instead, but this will be an apply-time error for all first-generation instances *and* for second-generation instances if the provider region is not supported with Cloud SQL. If you choose not to provide the `region` argument for this resource, make sure you understand this.
- `settings` - (Required) The settings to use for the database. The configuration is detailed below.

- `database_version` - (Optional, Default: `MYSQL_5_6`) The MySQL, PostgreSQL or SQL Server (beta) version to use. Supported values include `MYSQL_5_6`, `MYSQL_5_7`, `POSTGRES_9_6`,`POSTGRES_11`, `SQLSERVER_2017_STANDARD`, `SQLSERVER_2017_ENTERPRISE`, `SQLSERVER_2017_EXPRESS`, `SQLSERVER_2017_WEB`. Database Version Policies includes an up-to-date reference of supported versions. First-generation instances support `MYSQL_5_5` or `MYSQL_5_6`.
- `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.
- `root_password` - (Optional, Beta) Initial root password. Required for MS SQL Server, ignored by MySQL and PostgreSQL.

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. Note that future

`terraform apply` calls will attempt to resize the disk to the value specified in `disk_size` - if this is set, do not set `disk_size`.

- `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`.
- `user_labels` - (Optional) A set of key/value user label pairs to assign to the instance.

The optional `settings.database_flags` sublist supports:

- `name` - (Required) Name of the flag.
- `value` - (Required) Value of the flag.

The optional `settings.backup_configuration` subblock supports:

- `binary_log_enabled` - (Optional) True if binary logging is enabled. If `settings.backup_configuration.enabled` is false, this must be as well. Cannot be used with Postgres.
- `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) Whether this Cloud SQL instance should be assigned a public IPV4 address. Either `ipv4_enabled` must be enabled or a `private_network` must be configured.
- `private_network` - (Optional) The VPC network from which the Cloud SQL instance is accessible for private IP. For example, projects/myProject/global/networks/default. Specifying a network enables private IP. Either `ipv4_enabled` must be enabled or a `private_network` must be configured. This setting can be updated, but it cannot be removed after it is set.
- `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` - (Required) 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:

- `self_link` - The URI of the created resource.
- `connection_name` - The connection name of the instance to be used in connection strings. For example, when connecting with Cloud SQL Proxy.
- `service_account_email_address` - The service account email address assigned to the instance. This property is applicable only to Second Generation instances.
- `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.
- `ip_address.0.type` - The type of this IP address.
  - A `PRIMARY` address is an address that can accept incoming connections.
  - An `OUTGOING` address is the source address of connections originating from the instance, if supported.
  - A `PRIVATE` address is an address for an instance which has been configured to use private networking see: Private IP.
- `first_ip_address` - The first IPv4 address of any type assigned. This is to support accessing the first address in the list in a terraform output when the resource is configured with a `count`.
- `public_ip_address` - The first public (`PRIMARY`) IPv4 address assigned. This is a workaround for an issue fixed in Terraform 0.12 but also provides a convenient way to access an IP of a specific type without performing filtering in a Terraform config.
- `private_ip_address` - The first private (`PRIVATE`) IPv4 address assigned. This is a workaround for an issue fixed in Terraform 0.12 but also provides a convenient way to access an IP of a specific type without performing filtering in a Terraform config.
- `settings.version` - Used to make sure changes to the `settings` block are atomic.

- `server_ca_cert.0.cert` - The CA Certificate used to connect to the SQL Instance via SSL.
- `server_ca_cert.0.common_name` - The CN valid for the CA Cert.
- `server_ca_cert.0.create_time` - Creation time of the CA Cert.
- `server_ca_cert.0.expiration_time` - Expiration time of the CA Cert.
- `server_ca_cert.0.sha1_fingerprint` - SHA Fingerprint of the CA Cert.

» Timeouts

`google_sql_database_instance` provides the following Timeouts configuration options:

- `create` - Default is 20 minutes.
- `update` - Default is 20 minutes.
- `delete` - Default is 20 minutes.

» Import

Database instances can be imported using one of any of these accepted formats:

```
$ terraform import google_sql_database_instance.master projects/{{project}}/instances/{{name
$ terraform import google_sql_database_instance.master {{project}}/{{name}}
$ terraform import google_sql_database_instance.master {{name}}
```

NOTE: Some fields (such as `replica_configuration`) won't show a diff if they are unset in config and set on the server. When importing, double-check that your config has all the fields set that you expect- just seeing no diff isn't sufficient to know that your config could reproduce the imported resource.

» google_sql_ssl_cert

Creates a new Google SQL SSL Cert on a Google SQL Instance. For more information, see the official documentation, or the JSON API.

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

» Example Usage

Example creating a SQL Client Certificate.

```
resource "random_id" "db_name_suffix" {
  byte_length = 4
}

resource "google_sql_database_instance" "master" {
  name = "master-instance-${random_id.db_name_suffix.hex}"

  settings {
    tier = "D0"
  }
}

resource "google_sql_ssl_cert" "client_cert" {
  common_name = "client-name"
  instance    = google_sql_database_instance.master.name
}
```

» 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.
- `common_name` - (Required) The common name to be used in the certificate to identify the client. Constrained to [a-zA-Z.-_]+. 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:

- `sha1_fingerprint` - The SHA1 Fingerprint of the certificate.
- `private_key` - The private key associated with the client certificate.
- `server_ca_cert` - The CA cert of the server this client cert was generated from.
- `cert` - The actual certificate data for this client certificate.
- `cert_serial_number` - The serial number extracted from the certificate data.
- `create_time` - The time when the certificate was created in RFC 3339 format, for example 2012-11-15T16:19:00.094Z.

- `expiration_time` - The time when the certificate expires in RFC 3339 format, for example 2012-11-15T16:19:00.094Z.

» Import

Since the contents of the certificate cannot be accessed after its creation, this resource cannot be imported.

» 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 "random_id" "db_name_suffix" {
  byte_length = 4
}

resource "google_sql_database_instance" "master" {
  name = "master-instance-${random_id.db_name_suffix.hex}"

  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 MySQL instances. Don't set this field for PostgreSQL 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 MySQL databases can be imported using the `project`, `instance`, `host` and `name`, e.g.

```
$ terraform import google_sql_user.users my-project/master-instance/my-domain.com/me
```

SQL users for PostgreSQL databases can be imported using the `project`, `instance` and `name`, e.g.

```
$ terraform import google_sql_user.users my-project/master-instance/me
```

» google_logging_billing_account_exclusion

Manages a billing account logging exclusion. For more information see the official documentation and Excluding Logs.

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_exclusion" "my-exclusion" {
  name             = "my-instance-debug-exclusion"
```

```
  billing_account = "ABCDEF-012345-GHIJKL"

  description = "Exclude GCE instance debug logs"

  # Exclude all DEBUG or lower severity messages relating to instances
  filter = "resource.type = gce_instance AND severity <= DEBUG"
}
```

» Argument Reference

The following arguments are supported:

- `billing_account` - (Required) The billing account to create the exclusion for.
- `name` - (Required) The name of the logging exclusion.
- `description` - (Optional) A human-readable description.
- `disabled` - (Optional) Whether this exclusion rule should be disabled or not. This defaults to false.
- `filter` - (Required) The filter to apply when excluding logs. Only log entries that match the filter are excluded. See Advanced Log Filters for information on how to write a filter.

» Import

Billing account logging exclusions can be imported using their URI, e.g.

```
$ terraform import google_logging_billing_account_exclusion.my_exclusion billingAccounts/my-
```

» 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 You must have the "Logs Configuration Writer" IAM role (`roles/logging.configWriter`) granted on the billing account to the credentials used with Terraform. IAM roles granted on a billing account are separate from the typical IAM roles granted on a project.

» 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 bigquery
  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.
- `bigquery_options` - (Optional) Options that affect sinks exporting data to BigQuery. Structure documented below.

The `bigquery_options` block supports:

- `use_partitioned_tables` - (Required) Whether to use BigQuery's partition tables. By default, Logging creates dated tables based on the log entries' timestamps, e.g. syslog_20170523. With partitioned tables the

date suffix is no longer present and special query syntax has to be used instead. In both cases, tables are sharded based on UTC timezone.

» 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

Billing account logging sinks can be imported using this format:

```
$ terraform import google_logging_billing_account_sink.my_sink billingAccounts/{{billing_acc
```

» google_logging_folder_exclusion

Manages a folder-level logging exclusion. For more information see the official documentation and Excluding Logs.

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_exclusion" "my-exclusion" {
  name   = "my-instance-debug-exclusion"
  folder = google_folder.my-folder.name

  description = "Exclude GCE instance debug logs"

  # Exclude all DEBUG or lower severity messages relating to instances
  filter = "resource.type = gce_instance AND severity <= DEBUG"
}

resource "google_folder" "my-folder" {
  display_name = "My folder"
  parent       = "organizations/123456"
}
```

» Argument Reference

The following arguments are supported:

- `folder` - (Required) The folder to be exported to the sink. Note that either [FOLDER_ID] or "folders/[FOLDER_ID]" is accepted.
- `name` - (Required) The name of the logging exclusion.
- `description` - (Optional) A human-readable description.
- `disabled` - (Optional) Whether this exclusion rule should be disabled or not. This defaults to false.
- `filter` - (Required) The filter to apply when excluding logs. Only log entries that match the filter are excluded. See Advanced Log Filters for information on how to write a filter.

» Import

Folder-level logging exclusions can be imported using their URI, e.g.

```
$ terraform import google_logging_folder_exclusion.my_exclusion folders/my-folder/exclusions
```

» 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 bigquery
  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.
- `bigquery_options` - (Optional) Options that affect sinks exporting data to BigQuery. Structure documented below.

The `bigquery_options` block supports:

- `use_partitioned_tables` - (Required) Whether to use BigQuery's partition tables. By default, Logging creates dated tables based on the log

entries' timestamps, e.g. syslog_20170523. With partitioned tables the date suffix is no longer present and special query syntax has to be used instead. In both cases, tables are sharded based on UTC timezone.

» 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

Folder-level logging sinks can be imported using this format:

```
$ terraform import google_logging_folder_sink.my_sink folders/{{folder_id}}/sinks/{{sink_id}
```

» google_logging_organization_exclusion

Manages an organization-level logging exclusion. For more information see the official documentation and Excluding Logs.

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_exclusion" "my-exclusion" {
  name   = "my-instance-debug-exclusion"
  org_id = "123456789"

  description = "Exclude GCE instance debug logs"

  # Exclude all DEBUG or lower severity messages relating to instances
  filter = "resource.type = gce_instance AND severity <= DEBUG"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the logging exclusion.
- `org_id` - (Required) The organization to create the exclusion in.
- `description` - (Optional) A human-readable description.
- `disabled` - (Optional) Whether this exclusion rule should be disabled or not. This defaults to false.
- `filter` - (Required) The filter to apply when excluding logs. Only log entries that match the filter are excluded. See Advanced Log Filters for information on how to write a filter.

» Import

Organization-level logging exclusions can be imported using their URI, e.g.

```
$ terraform import google_logging_organization_exclusion.my_exclusion organizations/my-orga
```

» 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 bigquery
  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_member" "log-writer" {
  role = "roles/storage.objectCreator"

  member = 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.
- `bigquery_options` - (Optional) Options that affect sinks exporting data to BigQuery. Structure documented below.

The `bigquery_options` block supports:

- `use_partitioned_tables` - (Required) Whether to use BigQuery's partition tables. By default, Logging creates dated tables based on the log entries' timestamps, e.g. syslog_20170523. With partitioned tables the date suffix is no longer present and special query syntax has to be used instead. In both cases, tables are sharded based on UTC timezone.

» 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

Organization-level logging sinks can be imported using this format:

```
$ terraform import google_logging_organization_sink.my_sink organizations/{{organization_id}
```

» google_logging_metric

Logs-based metric can also be used to extract values from logs and create a a distribution of the values. The distribution records the statistics of the extracted values along with an optional histogram of the values as specified by the bucket options.

To get more information about Metric, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Logging Metric Basic

```
resource "google_logging_metric" "logging_metric" {
  name   = "my-(custom)/metric"
  filter = "resource.type=gae_app AND severity>=ERROR"
  metric_descriptor {
    metric_kind = "DELTA"
    value_type  = "DISTRIBUTION"
    unit        = "1"
    labels {
      key         = "mass"
      value_type  = "STRING"
      description = "amount of matter"
    }
    labels {
      key         = "sku"
      value_type  = "INT64"
```

```
      description = "Identifying number for item"
    }
    display_name = "My metric"
  }
  value_extractor = "EXTRACT(jsonPayload.request)"
  label_extractors = {
    "mass" = "EXTRACT(jsonPayload.request)"
    "sku"  = "EXTRACT(jsonPayload.id)"
  }
  bucket_options {
    linear_buckets {
      num_finite_buckets = 3
      width              = 1
      offset             = 1
    }
  }
}
```

» Example Usage - Logging Metric Counter Basic

```
resource "google_logging_metric" "logging_metric" {
  name   = "my-(custom)/metric"
  filter = "resource.type=gae_app AND severity>=ERROR"
  metric_descriptor {
    metric_kind = "DELTA"
    value_type  = "INT64"
  }
}
```

» Example Usage - Logging Metric Counter Labels

```
resource "google_logging_metric" "logging_metric" {
  name   = "my-(custom)/metric"
```

```
  filter = "resource.type=gae_app AND severity>=ERROR"
  metric_descriptor {
    metric_kind = "DELTA"
    value_type  = "INT64"
    labels {
      key         = "mass"
      value_type  = "STRING"
      description = "amount of matter"
    }
  }
  label_extractors = {
    "mass" = "EXTRACT(jsonPayload.request)"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The client-assigned metric identifier. Examples - "error_count", "nginx/requests". Metric identifiers are limited to 100 characters and can include only the following characters A-Z, a-z, 0-9, and the special characters _-.,+!*',()%/. The forward-slash character (/) denotes a hierarchy of name pieces, and it cannot be the first character of the name.
- `filter` - (Required) An advanced logs filter (https://cloud.google.com/logging/docs/view/advanced-filters) which is used to match log entries.
- `metric_descriptor` - (Required) The metric descriptor associated with the logs-based metric. Structure is documented below.

The `metric_descriptor` block supports:

- `unit` - (Optional) The unit in which the metric value is reported. It is only applicable if the valueType is `INT64`, `DOUBLE`, or `DISTRIBUTION`. The supported units are a subset of The Unified Code for Units of Measure standard
- `value_type` - (Required) Whether the measurement is an integer, a floating-point number, etc. Some combinations of metricKind and valueType might not be supported. For counter metrics, set this to INT64.
- `metric_kind` - (Required) Whether the metric records instantaneous values, changes to a value, etc. Some combinations of metricKind and valueType might not be supported. For counter metrics, set this to DELTA.

- `labels` - (Optional) The set of labels that can be used to describe a specific instance of this metric type. For example, the appengine.googleapis.com/http/server/response_latencies metric type has a label for the HTTP response code, response_code, so you can look at latencies for successful responses or just for responses that failed. Structure is documented below.
- `display_name` - (Optional) A concise name for the metric, which can be displayed in user interfaces. Use sentence case without an ending period, for example "Request count". This field is optional but it is recommended to be set for any metrics associated with user-visible concepts, such as Quota.

The `labels` block supports:

- `key` - (Required) The label key.
- `description` - (Optional) A human-readable description for the label.
- `value_type` - (Optional) The type of data that can be assigned to the label.

- `description` - (Optional) A description of this metric, which is used in documentation. The maximum length of the description is 8000 characters.
- `label_extractors` - (Optional) A map from a label key string to an extractor expression which is used to extract data from a log entry field and assign as the label value. Each label key specified in the LabelDescriptor must have an associated extractor expression in this map. The syntax of the extractor expression is the same as for the valueExtractor field.
- `value_extractor` - (Optional) A valueExtractor is required when using a distribution logs-based metric to extract the values to record from a log entry. Two functions are supported for value extraction - EXTRACT(field) or REGEXP_EXTRACT(field, regex). The argument are 1. field - The name of the log entry field from which the value is to be extracted. 2. regex - A regular expression using the Google RE2 syntax (https://github.com/google/re2/wiki/Syntax) with a single capture group to extract data from the specified log entry field. The value of the field is converted to a string before applying the regex. It is an error to specify a regex that does not include exactly one capture group.
- `bucket_options` - (Optional) The bucketOptions are required when the logs-based metric is using a DISTRIBUTION value type and it describes the bucket boundaries used to create a histogram of the extracted values. 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.

The `bucket_options` block supports:

- `linear_buckets` - (Optional) Specifies a linear sequence of buckets that all have the same width (except overflow and underflow). Each bucket represents a constant absolute uncertainty on the specific value in the bucket. Structure is documented below.
- `exponential_buckets` - (Optional) Specifies an exponential sequence of buckets that have a width that is proportional to the value of the lower bound. Each bucket represents a constant relative uncertainty on a specific value in the bucket. Structure is documented below.
- `explicit_buckets` - (Optional) Specifies a set of buckets with arbitrary widths. Structure is documented below.

The `linear_buckets` block supports:

- `num_finite_buckets` - (Optional) Must be greater than 0.
- `width` - (Optional) Must be greater than 0.
- `offset` - (Optional) Lower bound of the first bucket.

The `exponential_buckets` block supports:

- `num_finite_buckets` - (Optional) Must be greater than 0.
- `growth_factor` - (Optional) Must be greater than 1.
- `scale` - (Optional) Must be greater than 0.

The `explicit_buckets` block supports:

- `bounds` - (Required) The values must be monotonically increasing.

» 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

Metric can be imported using any of these accepted formats:

```
$ terraform import google_logging_metric.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_logging_project_exclusion

Manages a project-level logging exclusion. For more information see the official documentation and Excluding Logs.

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_exclusion" "my-exclusion" {
  name = "my-instance-debug-exclusion"

  description = "Exclude GCE instance debug logs"

  # Exclude all DEBUG or lower severity messages relating to instances
  filter = "resource.type = gce_instance AND severity <= DEBUG"
}
```

» Argument Reference

The following arguments are supported:

- `filter` - (Required) The filter to apply when excluding logs. Only log entries that match the filter are excluded. See Advanced Log Filters for information on how to write a filter.
- `name` - (Required) The name of the logging exclusion.
- `description` - (Optional) A human-readable description.
- `disabled` - (Optional) Whether this exclusion rule should be disabled or not. This defaults to false.

- `project` - (Optional) The project to create the exclusion in. If omitted, the project associated with the provider is used.

» Import

Project-level logging exclusions can be imported using their URI, e.g.

```
$ terraform import google_logging_project_exclusion.my_exclusion projects/my-project/exclusi
```

» 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: You must have granted the "Logs Configuration Writer" IAM role (`roles/logging.configWriter`) to the credentials used with terraform.

Note You must enable the Cloud Resource Manager API

» Example Usage

```
resource "google_logging_project_sink" "my-sink" {
  name = "my-pubsub-instance-sink"

  # Can export to pubsub, cloud storage, or bigquery
  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-9"
    }
  }

  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_

  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.
- `bigquery_options` - (Optional) Options that affect sinks exporting data to BigQuery. Structure documented below.

The `bigquery_options` block supports:

- `use_partitioned_tables` - (Required) Whether to use BigQuery's partition tables. By default, Logging creates dated tables based on the log entries' timestamps, e.g. syslog_20170523. With partitioned tables the date suffix is no longer present and special query syntax has to be used instead. In both cases, tables are sharded based on UTC timezone.

» 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_monitoring_alert_policy

A description of the conditions under which some aspect of your system is considered to be "unhealthy" and the ways to notify people or services about this state.

To get more information about AlertPolicy, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Monitoring Alert Policy Basic

```
resource "google_monitoring_alert_policy" "alert_policy" {
  display_name = "My Alert Policy"
  combiner     = "OR"
  conditions {
    display_name = "test condition"
    condition_threshold {
      filter     = "metric.type=\"compute.googleapis.com/instance/disk/write_bytes_count\" A
      duration   = "60s"
      comparison = "COMPARISON_GT"
      aggregations {
        alignment_period   = "60s"
        per_series_aligner = "ALIGN_RATE"
      }
    }
  }

  user_labels = {
    foo = "bar"
  }
}
```

» Argument Reference

The following arguments are supported:

- `display_name` - (Required) A short name or phrase used to identify the policy in dashboards, notifications, and incidents. To avoid confusion, don't use the same display name for multiple policies in the same project. The name is limited to 512 Unicode characters.
- `combiner` - (Required) How to combine the results of multiple conditions to determine if an incident should be opened.

- `conditions` - (Required) A list of conditions for the policy. The conditions are combined by AND or OR according to the combiner field. If the combined conditions evaluate to true, then an incident is created. A policy can have from one to six conditions. Structure is documented below.

The `conditions` block supports:

- `condition_absent` - (Optional) A condition that checks that a time series continues to receive new data points. Structure is documented below.
- `name` - The unique resource name for this condition. Its syntax is: projects/[PROJECT_ID]/alertPolicies/[POLICY_ID]/conditions/[CONDITION_ID] [CONDITION_ID] is assigned by Stackdriver Monitoring when the condition is created as part of a new or updated alerting policy.
- `condition_threshold` - (Optional) A condition that compares a time series against a threshold. Structure is documented below.
- `display_name` - (Required) A short name or phrase used to identify the condition in dashboards, notifications, and incidents. To avoid confusion, don't use the same display name for multiple conditions in the same policy.

The `condition_absent` block supports:

- `aggregations` - (Optional) Specifies the alignment of data points in individual time series as well as how to combine the retrieved time series together (such as when aggregating multiple streams on each resource to a single stream for each resource or when aggregating streams across all members of a group of resources). Multiple aggregations are applied in the order specified. Structure is documented below.
- `trigger` - (Optional) The number/percent of time series for which the comparison must hold in order for the condition to trigger. If unspecified, then the condition will trigger if the comparison is true for any of the time series that have been identified by filter and aggregations. Structure is documented below.
- `duration` - (Required) The amount of time that a time series must fail to report new data to be considered failing. Currently, only values that are a multiple of a minute--e.g. 60s, 120s, or 300s --are supported.
- `filter` - (Optional) A filter that identifies which time series should be compared with the threshold.The filter is similar to the one that is specified in the MetricService.ListTimeSeries request (that call is useful to verify the time series that will be retrieved / processed) and must specify the metric type and optionally may contain restrictions on resource type, resource labels, and metric labels. This field may not exceed 2048 Unicode characters in length.

The `aggregations` block supports:

- `per_series_aligner` - (Optional) The approach to be used to align individual time series. Not all alignment functions may be applied to all time series, depending on the metric type and value type of the original time series. Alignment may change the metric type or the value type of the time series.Time series data must be aligned in order to perform cross- time series reduction. If crossSeriesReducer is specified, then perSeriesAligner must be specified and not equal ALIGN_NONE and alignmentPeriod must be specified; otherwise, an error is returned.
- `group_by_fields` - (Optional) The set of fields to preserve when crossSeriesReducer is specified. The groupByFields determine how the time series are partitioned into subsets prior to applying the aggregation function. Each subset contains time series that have the same value for each of the grouping fields. Each individual time series is a member of exactly one subset. The crossSeriesReducer is applied to each subset of time series. It is not possible to reduce across different resource types, so this field implicitly contains resource.type. Fields not specified in groupByFields are aggregated away. If groupByFields is not specified and all the time series have the same resource type, then the time series are aggregated into a single output time series. If crossSeriesReducer is not defined, this field is ignored.
- `alignment_period` - (Optional) The alignment period for per-time series alignment. If present, alignmentPeriod must be at least 60 seconds. After per-time series alignment, each time series will contain data points only on the period boundaries. If perSeriesAligner is not specified or equals ALIGN_NONE, then this field is ignored. If perSeriesAligner is specified and does not equal ALIGN_NONE, then this field must be defined; otherwise an error is returned.
- `cross_series_reducer` - (Optional) The approach to be used to combine time series. Not all reducer functions may be applied to all time series, depending on the metric type and the value type of the original time series. Reduction may change the metric type of value type of the time series.Time series data must be aligned in order to perform cross- time series reduction. If crossSeriesReducer is specified, then perSeriesAligner must be specified and not equal ALIGN_NONE and alignmentPeriod must be specified; otherwise, an error is returned.

The `trigger` block supports:

- `percent` - (Optional) The percentage of time series that must fail the predicate for the condition to be triggered.
- `count` - (Optional) The absolute number of time series that must fail the predicate for the condition to be triggered.

The `condition_threshold` block supports:

- `threshold_value` - (Optional) A value against which to compare the time series.
- `denominator_filter` - (Optional) A filter that identifies a time series that should be used as the denominator of a ratio that will be compared with the threshold. If a denominator_filter is specified, the time series specified by the filter field will be used as the numerator.The filter is similar to the one that is specified in the MetricService.ListTimeSeries request (that call is useful to verify the time series that will be retrieved / processed) and must specify the metric type and optionally may contain restrictions on resource type, resource labels, and metric labels. This field may not exceed 2048 Unicode characters in length.
- `denominator_aggregations` - (Optional) Specifies the alignment of data points in individual time series selected by denominatorFilter as well as how to combine the retrieved time series together (such as when aggregating multiple streams on each resource to a single stream for each resource or when aggregating streams across all members of a group of resources).When computing ratios, the aggregations and denominator_aggregations fields must use the same alignment period and produce time series that have the same periodicity and labels.This field is similar to the one in the MetricService.ListTimeSeries request. It is advisable to use the ListTimeSeries method when debugging this field. Structure is documented below.
- `duration` - (Required) The amount of time that a time series must violate the threshold to be considered failing. Currently, only values that are a multiple of a minute--e.g., 0, 60, 120, or 300 seconds--are supported. If an invalid value is given, an error will be returned. When choosing a duration, it is useful to keep in mind the frequency of the underlying time series data (which may also be affected by any alignments specified in the aggregations field); a good duration is long enough so that a single outlier does not generate spurious alerts, but short enough that unhealthy states are detected and alerted on quickly.
- `comparison` - (Required) The comparison to apply between the time series (indicated by filter and aggregation) and the threshold (indicated by threshold_value). The comparison is applied on each time series, with the time series on the left-hand side and the threshold on the right-hand side. Only COMPARISON_LT and COMPARISON_GT are supported currently.
- `trigger` - (Optional) The number/percent of time series for which the comparison must hold in order for the condition to trigger. If unspecified, then the condition will trigger if the comparison is true for any of the time series that have been identified by filter and aggregations, or by the ratio, if denominator_filter and denominator_aggregations are specified. Structure is documented below.

- `aggregations` - (Optional) Specifies the alignment of data points in individual time series as well as how to combine the retrieved time series together (such as when aggregating multiple streams on each resource to a single stream for each resource or when aggregating streams across all members of a group of resources). Multiple aggregations are applied in the order specified.This field is similar to the one in the MetricService.ListTimeSeries request. It is advisable to use the ListTimeSeries method when debugging this field. Structure is documented below.
- `filter` - (Optional) A filter that identifies which time series should be compared with the threshold.The filter is similar to the one that is specified in the MetricService.ListTimeSeries request (that call is useful to verify the time series that will be retrieved / processed) and must specify the metric type and optionally may contain restrictions on resource type, resource labels, and metric labels. This field may not exceed 2048 Unicode characters in length.

The `denominator_aggregations` block supports:

- `per_series_aligner` - (Optional) The approach to be used to align individual time series. Not all alignment functions may be applied to all time series, depending on the metric type and value type of the original time series. Alignment may change the metric type or the value type of the time series.Time series data must be aligned in order to perform cross- time series reduction. If crossSeriesReducer is specified, then perSeriesAligner must be specified and not equal ALIGN_NONE and alignmentPeriod must be specified; otherwise, an error is returned.
- `group_by_fields` - (Optional) The set of fields to preserve when crossSeriesReducer is specified. The groupByFields determine how the time series are partitioned into subsets prior to applying the aggregation function. Each subset contains time series that have the same value for each of the grouping fields. Each individual time series is a member of exactly one subset. The crossSeriesReducer is applied to each subset of time series. It is not possible to reduce across different resource types, so this field implicitly contains resource.type. Fields not specified in groupByFields are aggregated away. If groupByFields is not specified and all the time series have the same resource type, then the time series are aggregated into a single output time series. If crossSeriesReducer is not defined, this field is ignored.
- `alignment_period` - (Optional) The alignment period for per-time series alignment. If present, alignmentPeriod must be at least 60 seconds. After per-time series alignment, each time series will contain data points only on the period boundaries. If perSeriesAligner is not specified or equals ALIGN_NONE, then this field is ignored. If perSeriesAligner is specified and does not equal ALIGN_NONE, then this field must be defined; otherwise an error is returned.

- `cross_series_reducer` - (Optional) The approach to be used to combine time series. Not all reducer functions may be applied to all time series, depending on the metric type and the value type of the original time series. Reduction may change the metric type of value type of the time series.Time series data must be aligned in order to perform cross- time series reduction. If crossSeriesReducer is specified, then perSeriesAligner must be specified and not equal ALIGN_NONE and alignmentPeriod must be specified; otherwise, an error is returned.

The `trigger` block supports:

- `percent` - (Optional) The percentage of time series that must fail the predicate for the condition to be triggered.
- `count` - (Optional) The absolute number of time series that must fail the predicate for the condition to be triggered.

The `aggregations` block supports:

- `per_series_aligner` - (Optional) The approach to be used to align individual time series. Not all alignment functions may be applied to all time series, depending on the metric type and value type of the original time series. Alignment may change the metric type or the value type of the time series.Time series data must be aligned in order to perform cross- time series reduction. If crossSeriesReducer is specified, then perSeriesAligner must be specified and not equal ALIGN_NONE and alignmentPeriod must be specified; otherwise, an error is returned.
- `group_by_fields` - (Optional) The set of fields to preserve when crossSeriesReducer is specified. The groupByFields determine how the time series are partitioned into subsets prior to applying the aggregation function. Each subset contains time series that have the same value for each of the grouping fields. Each individual time series is a member of exactly one subset. The crossSeriesReducer is applied to each subset of time series. It is not possible to reduce across different resource types, so this field implicitly contains resource.type. Fields not specified in groupByFields are aggregated away. If groupByFields is not specified and all the time series have the same resource type, then the time series are aggregated into a single output time series. If crossSeriesReducer is not defined, this field is ignored.
- `alignment_period` - (Optional) The alignment period for per-time series alignment. If present, alignmentPeriod must be at least 60 seconds. After per-time series alignment, each time series will contain data points only on the period boundaries. If perSeriesAligner is not specified or equals ALIGN_NONE, then this field is ignored. If perSeriesAligner is specified and does not equal ALIGN_NONE, then this field must be defined; otherwise an error is returned.
- `cross_series_reducer` - (Optional) The approach to be used to combine

time series. Not all reducer functions may be applied to all time series, depending on the metric type and the value type of the original time series. Reduction may change the metric type of value type of the time series.Time series data must be aligned in order to perform cross- time series reduction. If crossSeriesReducer is specified, then perSeriesAligner must be specified and not equal ALIGN_NONE and alignmentPeriod must be specified; otherwise, an error is returned.

- `enabled` - (Optional) Whether or not the policy is enabled. The default is true.
- `notification_channels` - (Optional) Identifies the notification channels to which notifications should be sent when incidents are opened or closed or when new violations occur on an already opened incident. Each element of this array corresponds to the name field in each of the NotificationChannel objects that are returned from the notificationChannels.list method. The syntax of the entries in this field is `projects/[PROJECT_ID]/notificationChannels/[CHANNEL_ID]`
- `user_labels` - (Optional) This field is intended to be used for organizing and identifying the AlertPolicy objects.The field can contain up to 64 entries. Each key and value is limited to 63 Unicode characters or 128 bytes, whichever is smaller. Labels and values can contain only lowercase letters, numerals, underscores, and dashes. Keys must begin with a letter.
- `documentation` - (Optional) A short name or phrase used to identify the policy in dashboards, notifications, and incidents. To avoid confusion, don't use the same display name for multiple policies in the same project. The name is limited to 512 Unicode characters. 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.

The `documentation` block supports:

- `content` - (Optional) The text of the documentation, interpreted according to mimeType. The content may not exceed 8,192 Unicode characters and may not exceed more than 10,240 bytes when encoded in UTF-8 format, whichever is smaller.
- `mime_type` - (Optional) The format of the content field. Presently, only the value "text/markdown" is supported.

» Attributes Reference

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

- `name` - The unique resource name for this policy. Its syntax is: projects/[PROJECT_ID]/alertPolicies/[ALERT_POLICY_ID]
- `creation_record` - A read-only record of the creation of the alerting policy. If provided in a call to create or update, this field will be ignored. Structure is documented below.

The `creation_record` block contains:

- `mutate_time` - When the change occurred.
- `mutated_by` - The email address of the user making the change.

» 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

AlertPolicy can be imported using any of these accepted formats:

```
$ terraform import google_monitoring_alert_policy.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_monitoring_group

The description of a dynamic collection of monitored resources. Each group has a filter that is matched against monitored resources and their associated

metadata. If a group's filter matches an available monitored resource, then that resource is a member of that group.

To get more information about Group, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Monitoring Group Basic

```
resource "google_monitoring_group" "basic" {
  display_name = "tf-test MonitoringGroup"

  filter = "resource.metadata.region=\"europe-west2\""
}
```

» Example Usage - Monitoring Group Subgroup

```
resource "google_monitoring_group" "parent" {
  display_name = "tf-test MonitoringParentGroup"
  filter       = "resource.metadata.region=\"europe-west2\""
}

resource "google_monitoring_group" "subgroup" {
  display_name = "tf-test MonitoringSubGroup"
  filter       = "resource.metadata.region=\"europe-west2\""
  parent_name  =  google_monitoring_group.parent.name
}
```

» Argument Reference

The following arguments are supported:

- `display_name` - (Required) A user-assigned name for this group, used only for display purposes.
- `filter` - (Required) The filter used to determine which monitored resources belong to this group.

- `parent_name` - (Optional) The name of the group's parent, if it has one. The format is "projects/{project_id_or_number}/groups/{group_id}". For groups with no parent, parentName is the empty string, "".
- `is_cluster` - (Optional) If true, the members of this group are considered to be a cluster. The system can perform additional analysis on groups that are clusters.
- `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` - A unique identifier for this group. The format is "projects/{project_id_or_number}/groups/{grou

» 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

Group can be imported using any of these accepted formats:

```
$ terraform import google_monitoring_group.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_monitoring_notification_channel

A NotificationChannel is a medium through which an alert is delivered when a policy violation is detected. Examples of channels include email, SMS, and third-party messaging applications. Fields containing sensitive information like authentication tokens or contact info are only partially populated on retrieval.

Notification Channels are designed to be flexible and are made up of a supported `type` and labels to configure that channel. Each `type` has specific labels that need to be present for that channel to be correctly configured. The labels that are required to be present for one channel `type` are often different than those required for another. Due to these loose constraints it's often best to set up a channel through the UI and import to Terraform when setting up a brand new channel type to determine which labels are required.

A list of supported channels per project the `list` endpoint can be accessed programmatically or through the api explorer at https://cloud.google.com/monitoring/api/ref_v3/rest/v3/projects.notificationChannelDescriptors/list . This provides the channel type and all of the required labels that must be passed.

To get more information about NotificationChannel, see:

- API documentation
- How-to Guides
  - Notification Options
  - Monitoring API Documentation

» Example Usage - Notification Channel Basic

```
resource "google_monitoring_notification_channel" "basic" {
  display_name = "Test Notification Channel"
  type         = "email"
  labels = {
    email_address = "fake_email@blahblah.com"
  }
}
```

» Argument Reference

The following arguments are supported:

- `type` - (Required) The type of the notification channel. This field matches the value of the NotificationChannelDescriptor.type field. See https://cloud.google.com/monitoring/api/ref_v3/rest/v3/projects.notificationChannelDescriptors/list to get the list of valid values such as "email", "slack", etc...
- `display_name` - (Required) An optional human-readable name for this notification channel. It is recommended that you specify a non-empty and unique name in order to make it easier to identify the channels in your project, though this is not enforced. The display name is limited to 512 Unicode characters.

- `labels` - (Optional) Configuration fields that define the channel and its behavior. The permissible and required labels are specified in the NotificationChannelDescriptor corresponding to the type field. **Note**: Some NotificationChannelDescriptor labels are sensitive and the API will return an partially-obfuscated value. For example, for `"type": "slack"` channels, an `auth_token` label with value "SECRET" will be obfuscated as "**CRET". In order to avoid a diff, Terraform will use the state value if it appears that the obfuscated value matches the state value in length/unobfuscated characters. However, Terraform will not detect a a diff if the obfuscated portion of the value was changed outside of Terraform.
- `user_labels` - (Optional) User-supplied key/value data that does not need to conform to the corresponding NotificationChannelDescriptor's schema, unlike the labels field. This field is intended to be used for organizing and identifying the NotificationChannel objects.The field can contain up to 64 entries. Each key and value is limited to 63 Unicode characters or 128 bytes, whichever is smaller. Labels and values can contain only lowercase letters, numerals, underscores, and dashes. Keys must begin with a letter.
- `description` - (Optional) An optional human-readable description of this notification channel. This description may provide additional details, beyond the display name, for the channel. This may not exceed 1024 Unicode characters.
- `enabled` - (Optional) Whether notifications are forwarded to the described channel. This makes it possible to disable delivery of notifications to a particular channel without removing the channel from all alerting policies that reference the channel. This is a more convenient approach when the

change is temporary and you want to receive notifications from the same set of alerting policies on the channel at some point in the future.

- `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` - The full REST resource name for this channel. The syntax is: projects/[PROJECT_ID]/notificationChannels/[CHANNEL_ID] The [CHANNEL_ID] is automatically assigned by the server on creation.
- `verification_status` - Indicates whether this channel has been verified or not. On a ListNotificationChannels or GetNotificationChannel operation, this field is expected to be populated.If the value is UNVERIFIED, then it indicates that the channel is non-functioning (it both requires verification and lacks verification); otherwise, it is assumed that the channel works.If the channel is neither VERIFIED nor UNVERIFIED, it implies that the channel is of a type that does not require verification or that this specific channel has been exempted from verification because it was created prior to verification being required for channels of this type.This field cannot be modified using a standard UpdateNotificationChannel operation. To change the value of this field, you must call VerifyNotificationChannel.

» 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

NotificationChannel can be imported using any of these accepted formats:

```
$ terraform import google_monitoring_notification_channel.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» google_monitoring_uptime_check_config

This message configures which resources and services to monitor for availability.

To get more information about UptimeCheckConfig, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Uptime Check Config Http

```
resource "google_monitoring_uptime_check_config" "http" {
  display_name = "http-uptime-check"
  timeout      = "60s"

  http_check {
    path = "/some-path"
    port = "8010"
  }

  monitored_resource {
    type = "uptime_url"
    labels = {
      project_id = "my-project-name"
      host       = "192.168.1.1"
    }
  }

  content_matchers {
    content = "example"
  }
}
```

» Example Usage - Uptime Check Config Https

```
resource "google_monitoring_uptime_check_config" "https" {
  display_name = "https-uptime-check"
  timeout = "60s"

  http_check {
    path = "/some-path"
    port = "443"
    use_ssl = true
    validate_ssl = true
  }

  monitored_resource {
    type = "uptime_url"
    labels = {
      project_id = "my-project-name"
      host = "192.168.1.1"
    }
  }

  content_matchers {
    content = "example"
  }
}
```

» Example Usage - Uptime Check Tcp

```
resource "google_monitoring_uptime_check_config" "tcp_group" {
  display_name = "tcp-uptime-check"
  timeout      = "60s"

  tcp_check {
```

```
    port = 888
  }

  resource_group {
    resource_type = "INSTANCE"
    group_id      = google_monitoring_group.check.name
  }
}

resource "google_monitoring_group" "check" {
  display_name = "uptime-check-group"
  filter       = "resource.metadata.name=has_substring(\"foo\")"
}
```

» Argument Reference

The following arguments are supported:

- `display_name` - (Required) A human-friendly name for the uptime check configuration. The display name should be unique within a Stackdriver Workspace in order to make it easier to identify; however, uniqueness is not enforced.
- `timeout` - (Required) The maximum amount of time to wait for the request to complete (must be between 1 and 60 seconds). Accepted formats https://developers.google.com/protocol-buffers/docs/reference/google.protobuf#google.protobuf.Duration

---

- `period` - (Optional) How often, in seconds, the uptime check is performed. Currently, the only supported values are 60s (1 minute), 300s (5 minutes), 600s (10 minutes), and 900s (15 minutes). Optional, defaults to 300s.
- `content_matchers` - (Optional) The expected content on the page the check is run against. Currently, only the first entry in the list is supported, and other entries will be ignored. The server will look for an exact match of the string in the page response's content. This field is optional and should only be specified if a content match is required. Structure is documented below.
- `selected_regions` - (Optional) The list of regions from which the check will be run. Some regions contain one location, and others contain more than one. If this field is specified, enough regions to include a minimum of 3 locations must be provided, or an error message is returned. Not specifying this field will result in uptime checks running from all regions.

- `http_check` - (Optional) Contains information needed to make an HTTP or HTTPS check. Structure is documented below.
- `tcp_check` - (Optional) Contains information needed to make a TCP check. Structure is documented below.
- `resource_group` - (Optional) The group resource associated with the configuration. Structure is documented below.
- `monitored_resource` - (Optional) The monitored resource (https://cloud.google.com/monitoring/api/resources) associated with the configuration. The following monitored resource types are supported for uptime checks: uptime_url gce_instance gae_app aws_ec2_instance aws_elb_load_balancer 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.

The `content_matchers` block supports:

- `content` - (Required) String or regex content to match (max 1024 bytes)

The `http_check` block supports:

- `auth_info` - (Optional) The authentication information. Optional when creating an HTTP check; defaults to empty. Structure is documented below.
- `port` - (Optional) The port to the page to run the check against. Will be combined with host (specified within the MonitoredResource) and path to construct the full URL. Optional (defaults to 80 without SSL, or 443 with SSL).
- `headers` - (Optional) The list of headers to send as part of the uptime check request. If two headers have the same key and different values, they should be entered as a single header, with the value being a comma-separated list of all the desired values as described at https://www.w3.org/Protocols/rfc2616/rfc2616.txt (page 31). Entering two separate headers with the same key in a Create call will cause the first to be overwritten by the second. The maximum number of headers allowed is 100.
- `path` - (Optional) The path to the page to run the check against. Will be combined with the host (specified within the MonitoredResource) and port to construct the full URL. Optional (defaults to "/").
- `use_ssl` - (Optional) If true, use HTTPS instead of HTTP to run the check.
- `validate_ssl` - (Optional) Boolean specifying whether to include SSL certificate validation as a part of the Uptime check. Only applies to checks where monitoredResource is set to uptime_url. If useSsl is false, setting validateSsl to true has no effect.

- `mask_headers` - (Optional) Boolean specifying whether to encrypt the header information. Encryption should be specified for any headers related to authentication that you do not wish to be seen when retrieving the configuration. The server will be responsible for encrypting the headers. On Get/List calls, if mask_headers is set to True then the headers will be obscured with ******.

The `auth_info` block supports:

- `password` - (Required) The password to authenticate.
- `username` - (Required) The username to authenticate.

The `tcp_check` block supports:

- `port` - (Required) The port to the page to run the check against. Will be combined with host (specified within the MonitoredResource) to construct the full URL.

The `resource_group` block supports:

- `resource_type` - (Optional) The resource type of the group members.
- `group_id` - (Optional) The group of resources being monitored. Should be the `name` of a group

The `monitored_resource` block supports:

- `type` - (Required) The monitored resource type. This field must match the type field of a MonitoredResourceDescriptor (https://cloud.google.com/monitoring/api/ref_v3/rest/v3/projects.monitoredResourceDescriptors#MonitoredResourceDescriptor) object. For example, the type of a Compute Engine VM instance is gce_instance. For a list of types, see Monitoring resource types (https://cloud.google.com/monitoring/api/resources) and Logging resource types (https://cloud.google.com/logging/docs/api/v2/resource-list).
- `labels` - (Required) Values for all of the labels listed in the associated monitored resource descriptor. For example, Compute Engine VM instances use the labels "project_id", "instance_id", and "zone".

» Attributes Reference

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

- `name` - A unique resource name for this UptimeCheckConfig. The format is projects/[PROJECT_ID]/uptimeCheckConfigs/[UPTIME_CHECK_ID].
- `uptime_check_id` - The id of the uptime check

» 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

UptimeCheckConfig can be imported using any of these accepted formats:

```
$ terraform import google_monitoring_uptime_check_config.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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.

Note: If the project id is not set on the resource or in the provider block it will be dynamically determined which will require enabling the compute 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, Default: 'STANDARD') The Storage Class of the new bucket. Supported values include: `STANDARD`, `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.
- `retention_policy` - (Optional) Configuration of the bucket's data retention policy for how long objects in the bucket should be retained. 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.
- `encryption` - (Optional) The bucket's encryption configuration.
- `requester_pays` - (Optional, Default: false) Enables Requester Pays on a storage bucket.
- `bucket_policy_only` - (Optional, Default: false) Enables Bucket Policy Only access to a bucket.

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.
- `with_state` - (Optional) Match to live and/or archived objects. Unversioned buckets have only live objects. Supported values include: `"LIVE"`, `"ARCHIVED"`, `"ANY"`.
- `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` - (Required) 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 `retention_policy` block supports:

- `is_locked` - (Optional) If set to `true`, the bucket will be locked and permanently restrict edits to the bucket's retention policy. Caution: Locking a bucket is an irreversible action.
- `retention_period` - (Optional) The period of time, in seconds, that objects in the bucket must be retained and cannot be deleted, overwritten, or archived. The value must be less than 3,155,760,000 seconds.

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 this bucket's name.

The `encryption` block supports:

- `default_kms_key_name`: A Cloud KMS key that will be used to encrypt objects inserted into this bucket, if no encryption method is specified. You must pay attention to whether the crypto key is available in the location that this bucket is created in. See the docs for more details.

» 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` or `project/name`. If the project is not passed to the import command it will be inferred from the provider block

or environment variables. If it cannot be inferred it will be queried from the Compute API (this will fail if the API is not enabled).

e.g.

```
$ terraform import google_storage_bucket.image-store image-store-bucket
$ terraform import google_storage_bucket.image-store tf-test-project/image-store-bucket
```

Note: Terraform will import this resource with `force_destroy` set to `false` in state. If you've set it to `true` in config, run `terraform apply` to update the value set in state. If you delete this resource before updating the value, objects in the bucket will not be destroyed.

» google_storage_bucket_access_control

The BucketAccessControls resource represents the Access Control Lists (ACLs) for buckets within Google Cloud Storage. ACLs let you specify who has access to your data and to what extent.

There are three roles that can be assigned to an entity:

READERs can get the bucket, though no acl property will be returned, and list the bucket's objects. WRITERs are READERs, and they can insert objects into the bucket and delete the bucket's objects. OWNERs are WRITERs, and they can get the acl property of a bucket, update a bucket, and call all BucketAccessControls methods on the bucket. For more information, see Access Control, with the caveat that this API uses READER, WRITER, and OWNER instead of READ, WRITE, and FULL_CONTROL.

» Example Usage - Storage Bucket Access Control Public Bucket

```
resource "google_storage_bucket_access_control" "public_rule" {
  bucket = google_storage_bucket.bucket.name
  role   = "READER"
  entity = "allUsers"
}

resource "google_storage_bucket" "bucket" {
  name = "static-content-bucket"
```

```
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket.
- `entity` - (Required) The entity holding the permission, in one of the following forms: user-userId user-email group-groupId group-email domain-domain project-team-projectId allUsers allAuthenticatedUsers Examples: The user liz@example.com would be user-liz@example.com. The group example@googlegroups.com would be group-example@googlegroups.com. To refer to all members of the Google Apps for Business domain example.com, the entity would be domain-example.com.

- `role` - (Optional) The access permission for the entity.

» Attributes Reference

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

- `domain` - The domain associated with the entity.
- `email` - The email address associated with the entity.

» 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

BucketAccessControl can be imported using any of these accepted formats:

```
$ terraform import google_storage_bucket_access_control.default {{bucket}}/{{entity}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» 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 StorageBucket

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

- `google_storage_bucket_iam_policy`: Authoritative. Sets the IAM policy for the bucket and replaces any existing policy already attached.
- `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 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 bucket are preserved.

Note: `google_storage_bucket_iam_policy` **cannot** be used in conjunction with `google_storage_bucket_iam_binding` and `google_storage_bucket_iam_member` or they will fight over what your policy should be.

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_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/storage.admin"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_storage_bucket_iam_policy" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/storage.admin"
    members = [
      "user:jane@example.com",
    ]
```

```
    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_storage_bucket_iam_policy" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_storage_bucket_iam_binding

```
resource "google_storage_bucket_iam_binding" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_storage_bucket_iam_binding" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_storage_bucket_iam_member

```
resource "google_storage_bucket_iam_member" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  member = "user:jane@example.com"
```

```
}
```

With IAM Conditions (beta):

```
resource "google_storage_bucket_iam_member" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_storage_bucket_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_storage_bucket_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- b/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Storage bucket IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import`

`google_storage_bucket_iam_member.editor "b/{{bucket}}?projection=full roles/storage.objectViewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_storage_bucket_iam_binding.editor "b/{{bucket}} roles/storage.objectViewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_storage_bucket_iam_policy.editor b/{{bucket}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for StorageBucket

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

- `google_storage_bucket_iam_policy`: Authoritative. Sets the IAM policy for the bucket and replaces any existing policy already attached.
- `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 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 bucket are preserved.

Note: `google_storage_bucket_iam_policy` **cannot** be used in conjunction with `google_storage_bucket_iam_binding` and `google_storage_bucket_iam_member` or they will fight over what your policy should be.

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_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/storage.admin"
```

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

resource "google_storage_bucket_iam_policy" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/storage.admin"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
  }
}

resource "google_storage_bucket_iam_policy" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_storage_bucket_iam_binding

```
resource "google_storage_bucket_iam_binding" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_storage_bucket_iam_binding" "editor" {
  bucket = "${google_storage_bucket.default.name}"
```

```
  role = "roles/storage.admin"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_storage_bucket_iam_member

```
resource "google_storage_bucket_iam_member" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_storage_bucket_iam_member" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_storage_bucket_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_storage_bucket_iam_policy`) The policy data generated by a `google_iam_policy` data source.
- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- b/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Storage bucket IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_storage_bucket_iam_member.editor "b/{{bucket}}?projection=full roles/storage.objectViewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_storage_bucket_iam_binding.editor "b/{{bucket}} roles/storage.objectViewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_storage_bucket_iam_policy.editor b/{{bucket}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» IAM policy for StorageBucket

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

- `google_storage_bucket_iam_policy`: Authoritative. Sets the IAM policy for the bucket and replaces any existing policy already attached.
- `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 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 bucket are preserved.

Note: `google_storage_bucket_iam_policy` **cannot** be used in conjunction with `google_storage_bucket_iam_binding` and `google_storage_bucket_iam_member` or they will fight over what your policy should be.

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_policy

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/storage.admin"
    members = [
      "user:jane@example.com",
    ]
  }
}

resource "google_storage_bucket_iam_policy" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

With IAM Conditions (beta):

```
data "google_iam_policy" "admin" {
  binding {
    role = "roles/storage.admin"
    members = [
      "user:jane@example.com",
    ]

    condition {
      title       = "expires_after_2019_12_31"
      description = "Expiring at midnight of 2019-12-31"
      expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
    }
```

```
  }
}

resource "google_storage_bucket_iam_policy" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  policy_data = "${data.google_iam_policy.admin.policy_data}"
}
```

» google_storage_bucket_iam_binding

```
resource "google_storage_bucket_iam_binding" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  members = [
    "user:jane@example.com",
  ]
}
```

With IAM Conditions (beta):

```
resource "google_storage_bucket_iam_binding" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  members = [
    "user:jane@example.com",
  ]

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» google_storage_bucket_iam_member

```
resource "google_storage_bucket_iam_member" "editor" {
  bucket = "${google_storage_bucket.default.name}"
  role = "roles/storage.admin"
  member = "user:jane@example.com"
}
```

With IAM Conditions (beta):

```
resource "google_storage_bucket_iam_member" "editor" {
  bucket = "${google_storage_bucket.default.name}"
```

```
  role = "roles/storage.admin"
  member = "user:jane@example.com"

  condition {
    title       = "expires_after_2019_12_31"
    description = "Expiring at midnight of 2019-12-31"
    expression  = "request.time < timestamp(\"2020-01-01T00:00:00Z\")"
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) Used to find the parent resource to bind the IAM policy to
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the project will be parsed from the identifier of the parent resource. If no project is provided in the parent identifier and no project is specified, 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 G Suite domain (primary, instead of alias) 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_storage_bucket_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_storage_bucket_iam_policy`) The policy data generated by a `google_iam_policy` data source.

- `condition` - (Optional, Beta) An IAM Condition for a given binding. Structure is documented below.

The `condition` block supports:

- `expression` - (Required) Textual representation of an expression in Common Expression Language syntax.
- `title` - (Required) A title for the expression, i.e. a short string describing its purpose.
- `description` - (Optional) An optional description of the expression. This is a longer text which describes the expression, e.g. when hovered over it in a UI.

Warning: Terraform considers the `role` and condition contents (`title`+`description`+`expression`) as the identifier for the binding. This means that if any part of the condition is changed out-of-band, Terraform will consider it to be an entirely different resource and will treat it as such.

» Attributes Reference

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

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

» Import

For all import syntaxes, the "resource in question" can take any of the following forms:

- b/{{name}}
- {{name}}

Any variables not passed in the import command will be taken from the provider configuration.

Storage bucket IAM resources can be imported using the resource identifiers, role, and member.

IAM member imports use space-delimited identifiers: the resource in question, the role, and the member identity, e.g. `$ terraform import google_storage_bucket_iam_member.editor "b/{{bucket}}?projection=full roles/storage.objectViewer jane@example.com"`

IAM binding imports use space-delimited identifiers: the resource in question and the role, e.g. `$ terraform import google_storage_bucket_iam_binding.editor "b/{{bucket}} roles/storage.objectViewer"`

IAM policy imports use the identifier of the resource in question, e.g. `$ terraform import google_storage_bucket_iam_policy.editor b/{{bucket}}`

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.

» 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. If you're interpolating the name of this object, see `output_name` instead.

One of the following is required:

- `content` - (Optional, Sensitive) Data as `string` to be uploaded. Must be defined if `source` is not. **Note**: The `content` field is marked as sensitive. To view the raw contents of the object, please define an output.

- `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.
- `self_link` - (Computed) A url reference to this object.
- `output_name` - (Computed) The name of the object. Use this field in interpolations with `google_storage_object_acl` to recreate `google_storage_object_acl` resources when your `google_storage_bucket_object` is recreated.

» google_storage_default_object_access_control

The DefaultObjectAccessControls resources represent the Access Control Lists (ACLs) applied to a new object within a Google Cloud Storage bucket when no ACL was provided for that object. ACLs let you specify who has access to your bucket contents and to what extent.

There are two roles that can be assigned to an entity:

READERs can get an object, though the acl property will not be revealed. OWNERs are READERs, and they can get the acl property, update an object, and call all objectAccessControls methods on the object. The owner of an object is always an OWNER. For more information, see Access Control, with the caveat that this API uses READER and OWNER instead of READ and FULL_CONTROL.

To get more information about DefaultObjectAccessControl, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Storage Default Object Access Control Public

```
resource "google_storage_default_object_access_control" "public_rule" {
  bucket = google_storage_bucket.bucket.name
  role   = "READER"
  entity = "allUsers"
}

resource "google_storage_bucket" "bucket" {
  name = "static-content-bucket"
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket.
- `entity` - (Required) The entity holding the permission, in one of the following forms:
  - user-{{userId}}
  - user-{{email}} (such as "user-liz@example.com")
  - group-{{groupId}}
  - group-{{email}} (such as "group-example@googlegroups.com")
  - domain-{{domain}} (such as "domain-example.com")
  - project-team-{{projectId}}

  - allUsers
  - allAuthenticatedUsers

- `role` - (Required) The access permission for the entity.

- `object` - (Optional) The name of the object, if applied to an object.

» Attributes Reference

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

- `domain` - The domain associated with the entity.
- `email` - The email address associated with the entity.
- `entity_id` - The ID for the entity
- `generation` - The content generation of the object, if applied to an object.
- `project_team` - The project team associated with the entity Structure is documented below.

The `project_team` block contains:

- `project_number` - (Optional) The project team associated with the entity
- `team` - (Optional) The team.

» 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

DefaultObjectAccessControl can be imported using any of these accepted formats:

```
$ terraform import google_storage_default_object_access_control.default {{bucket}}/{{entity}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» google_storage_default_object_acl

Authoritatively manages the default object ACLs for a Google Cloud Storage bucket without managing the bucket itself.

Note that for each object, its creator will have the `"OWNER"` role in addition to the default ACL that has been defined.

For more information see the official documentation and API.

Want fine-grained control over default object ACLs? Use `google_storage_default_object_access_control` to control individual role entity pairs.

» 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` - (Optional) List of role/entity pairs in the form `ROLE:entity`. See GCS Object ACL documentation for more details. Omitting the field is the same as providing an empty list.

» 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.

In order to enable notifications, a special Google Cloud Storage service account unique to the project must have the IAM permission "projects.topics.publish" for a Cloud Pub/Sub topic in the project. To get the service account's email address, use the `google_storage_project_service_account` datasource's `email_address` value, and see below for an example of enabling notifications by granting the correct IAM permission. See the notifications documentation for more details.

» Example Usage

```
resource "google_storage_notification" "notification" {
  bucket         = google_storage_bucket.bucket.name
  payload_format = "JSON_API_V1"
  topic          = google_pubsub_topic.topic.name
  event_types    = ["OBJECT_FINALIZE", "OBJECT_METADATA_UPDATE"]
  custom_attributes = {
    new-attribute = "new-attribute-value"
  }
  depends_on = [google_pubsub_topic_iam_binding.binding]
}

// Enable notifications by giving the correct IAM permission to the unique service account.

data "google_storage_project_service_account" "gcs_account" {
}

resource "google_pubsub_topic_iam_binding" "binding" {
  topic   = google_pubsub_topic.topic.name
  role    = "roles/pubsub.publisher"
  members = ["serviceAccount:${data.google_storage_project_service_account.gcs_account.email
}

// End enabling notifications

resource "google_storage_bucket" "bucket" {
  name = "default_bucket"
}

resource "google_pubsub_topic" "topic" {
```

```
  name = "default_topic"
}
```

» 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. Expects either the topic name, assumed to belong to the default GCP provider project, or the project-level name, i.e. `projects/my-gcp-project/topics/my-topic` or `my-topic`.

- `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:

- `notification_id` - The ID of the created notification.
- `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_access_control

The ObjectAccessControls resources represent the Access Control Lists (ACLs) for objects within Google Cloud Storage. ACLs let you specify who has access to your data and to what extent.

There are two roles that can be assigned to an entity:

READERs can get an object, though the acl property will not be revealed. OWNERs are READERs, and they can get the acl property, update an object, and call all objectAccessControls methods on the object. The owner of an object is always an OWNER. For more information, see Access Control, with the caveat that this API uses READER and OWNER instead of READ and FULL_CONTROL.

To get more information about ObjectAccessControl, see:

- API documentation
- How-to Guides
  - Official Documentation

» Example Usage - Storage Object Access Control Public Object

```
resource "google_storage_object_access_control" "public_rule" {
  object = google_storage_bucket_object.object.output_name
  bucket = google_storage_bucket.bucket.name
  role   = "READER"
  entity = "allUsers"
}

resource "google_storage_bucket" "bucket" {
  name = "static-content-bucket"
}

resource "google_storage_bucket_object" "object" {
  name   = "public-object"
  bucket = google_storage_bucket.bucket.name
  source = "../static/img/header-logo.png"
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket.
- `entity` - (Required) The entity holding the permission, in one of the following forms:
  - user-{{userId}}
  - user-{{email}} (such as "user-liz@example.com")
  - group-{{groupId}}
  - group-{{email}} (such as "group-example@googlegroups.com")
  - domain-{{domain}} (such as "domain-example.com")
  - project-team-{{projectId}}
  - allUsers
  - allAuthenticatedUsers
- `object` - (Required) The name of the object to apply the access control to.
- `role` - (Required) The access permission for the entity.

» Attributes Reference

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

- `domain` - The domain associated with the entity.
- `email` - The email address associated with the entity.
- `entity_id` - The ID for the entity
- `generation` - The content generation of the object, if applied to an object.
- `project_team` - The project team associated with the entity Structure is documented below.

The `project_team` block contains:

- `project_number` - (Optional) The project team associated with the entity
- `team` - (Optional) The team.

» 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

ObjectAccessControl can be imported using any of these accepted formats:

```
$ terraform import google_storage_object_access_control.default {{bucket}}/{{object}}/{{enti
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» google_storage_object_acl

Authoritatively manages the access control list (ACL) for an object in a Google Cloud Storage (GCS) bucket. Removing a `google_storage_object_acl` sets the acl to the `private` predefined ACL.

For more information see the official documentation and API.

Want fine-grained control over object ACLs? Use `google_storage_object_access_control` to control individual role entity pairs.

» 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.output_name

  role_entity = [
```

```
    "OWNER:user-my.email@gmail.com",
    "READER:group-mygroup",
  ]
}
```

» Argument Reference

- `bucket` - (Required) The name of the bucket the object is stored in.
- `object` - (Required) The name of the object to apply the acl to.

- `predefined_acl` - (Optional) The "canned" predefined 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.

The object's creator will always have `OWNER` permissions for their object, and any attempt to modify that permission would return an error. Instead, Terraform automatically adds that role/entity pair to your `terraform plan` results when it is omitted in your config; `terraform plan` will show the correct final state at every point except for at `Create` time, where the object role/entity pair is omitted if not explicitly set.

» Attributes Reference

Only the arguments listed above are exposed as attributes.

» google_storage_transfer_job

Creates a new Transfer Job in Google Cloud Storage Transfer.

To get more information about Google Cloud Storage Transfer, see:

- Overview
- API documentation
- How-to Guides
  - Configuring Access to Data Sources and Sinks

» Example Usage

Example creating a nightly Transfer Job from an AWS S3 Bucket to a GCS bucket.

```
data "google_storage_transfer_project_service_account" "default" {
  project = var.project
}

resource "google_storage_bucket" "s3-backup-bucket" {
  name          = "${var.aws_s3_bucket}-backup"
  storage_class = "NEARLINE"
  project       = var.project
}

resource "google_storage_bucket_iam_member" "s3-backup-bucket" {
  bucket     = google_storage_bucket.s3-backup-bucket.name
  role       = "roles/storage.admin"
  member     = "serviceAccount:${data.google_storage_transfer_project_service_account.defaul
  depends_on = [google_storage_bucket.s3-backup-bucket]
}

resource "google_storage_transfer_job" "s3-bucket-nightly-backup" {
  description = "Nightly backup of S3 bucket"
  project     = var.project

  transfer_spec {
    object_conditions {
      max_time_elapsed_since_last_modification = "600s"
      exclude_prefixes = [
        "requests.gz",
      ]
    }
    transfer_options {
      delete_objects_unique_in_sink = false
    }
    aws_s3_data_source {
      bucket_name = var.aws_s3_bucket
      aws_access_key {
        access_key_id     = var.aws_access_key
        secret_access_key = var.aws_secret_key
      }
    }
    gcs_data_sink {
      bucket_name = google_storage_bucket.s3-backup-bucket.name
    }
```

```
  }

  schedule {
    schedule_start_date {
      year  = 2018
      month = 10
      day   = 1
    }
    schedule_end_date {
      year  = 2019
      month = 1
      day   = 15
    }
    start_time_of_day {
      hours   = 23
      minutes = 30
      seconds = 0
      nanos   = 0
    }
  }

  depends_on = [google_storage_bucket_iam_member.s3-backup-bucket]
}
```

» Argument Reference

The following arguments are supported:

- `description` - (Required) Unique description to identify the Transfer Job.
- `transfer_spec` - (Required) Transfer specification. Structure documented below.
- `schedule` - (Required) Schedule specification defining when the Transfer Job should be scheduled to start, end and and what time to run. Structure documented below.

- `project` - (Optional) The project in which the resource belongs. If it is not provided, the provider project is used.
- `status` - (Optional) Status of the job. Default: `ENABLED`. **NOTE: The effect of the new job status takes place during a subsequent job run. For example, if you change the job status from ENABLED to DISABLED, and an operation spawned by the transfer is running, the status change would not affect the current operation.**

The `transfer_spec` block supports:

- `gcs_data_sink` - (Required) A Google Cloud Storage data sink. Structure documented below.
- `object_conditions` - (Optional) Only objects that satisfy these object conditions are included in the set of data source and data sink objects. Object conditions based on objects' `last_modification_time` do not exclude objects in a data sink. Structure documented below.
- `transfer_options` - (Optional) Characteristics of how to treat files from datasource and sink during job. If the option `delete_objects_unique_in_sink` is true, object conditions based on objects' `last_modification_time` are ignored and do not exclude objects in a data source or a data sink. Structure documented below.
- `gcs_data_source` - (Optional) A Google Cloud Storage data source. Structure documented below.
- `aws_s3_data_source` - (Optional) An AWS S3 data source. Structure documented below.
- `http_data_source` - (Optional) An HTTP URL data source. Structure documented below.

The `schedule` block supports:

- `schedule_start_date` - (Required) The first day the recurring transfer is scheduled to run. If `schedule_start_date` is in the past, the transfer will run for the first time on the following day. Structure documented below.
- `schedule_end_date` - (Optional) The last day the recurring transfer will be run. If `schedule_end_date` is the same as `schedule_start_date`, the transfer will be executed only once. Structure documented below.
- `start_time_of_day` - (Optional) The time in UTC at which the transfer will be scheduled to start in a day. Transfers may start later than this time. If not specified, recurring and one-time transfers that are scheduled to run today will run immediately; recurring transfers that are scheduled to run on a future date will start at approximately midnight UTC on that date. Note that when configuring a transfer with the Cloud Platform Console, the transfer's start time in a day is specified in your local timezone. Structure documented below.

The `object_conditions` block supports:

- `max_time_elapsed_since_last_modification` - (Optional) A duration in seconds with up to nine fractional digits, terminated by 's'. Example: "3.5s".
- `min_time_elapsed_since_last_modification` - (Optional) A duration in seconds with up to nine fractional digits, terminated by 's'. Example:

"3.5s".

- `include_prefixes` - (Optional) If `include_refixes` is specified, objects that satisfy the object conditions must have names that start with one of the `include_prefixes` and that do not start with any of the `exclude_prefixes`. If `include_prefixes` is not specified, all objects except those that have names starting with one of the `exclude_prefixes` must satisfy the object conditions. See Requirements.
- `exclude_prefixes` - (Optional) `exclude_prefixes` must follow the requirements described for `include_prefixes`. See Requirements.

The `transfer_options` block supports:

- `overwrite_objects_already_existing_in_sink` - (Optional) Whether overwriting objects that already exist in the sink is allowed.
- `delete_objects_unique_in_sink` - (Optional) Whether objects that exist only in the sink should be deleted. Note that this option and `delete_objects_from_source_after_transfer` are mutually exclusive.
- `delete_objects_from_source_after_transfer` - (Optional) Whether objects should be deleted from the source after they are transferred to the sink. Note that this option and `delete_objects_unique_in_sink` are mutually exclusive.

The `gcs_data_sink` block supports:

- `bucket_name` - (Required) Google Cloud Storage bucket name.

The `gcs_data_source` block supports:

- `bucket_name` - (Required) Google Cloud Storage bucket name.

The `aws_s3_data_source` block supports:

- `bucket_name` - (Required) S3 Bucket name.
- `aws_access_key` - (Required) AWS credentials block.

The `aws_access_key` block supports:

- `access_key_id` - (Required) AWS Key ID.
- `secret_access_key` - (Required) AWS Secret Access Key.

The `http_data_source` block supports:

- `list_url` - (Required) The URL that points to the file that stores the object list entries. This file must allow public access. Currently, only URLs with HTTP and HTTPS schemes are supported.

The `schedule_start_date` and `schedule_end_date` blocks support:

- `year` - (Required) Year of date. Must be from 1 to 9999.

- `month` - (Required) Month of year. Must be from 1 to 12.
- `day` - (Required) Day of month. Must be from 1 to 31 and valid for the year and month.

The `start_time_of_day` blocks support:

- `hours` - (Required) Hours of day in 24 hour format. Should be from 0 to 23
- `minutes` - (Required) Minutes of hour of day. Must be from 0 to 59.
- `seconds` - (Optional) Seconds of minutes of the time. Must normally be from 0 to 59.
- `nanos` - (Required) Fractions of seconds in nanoseconds. Must be from 0 to 999,999,999.

» Attributes Reference

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

- `name` - The name of the Transfer Job.
- `creation_time` - When the Transfer Job was created.
- `last_modification_time` - When the Transfer Job was last modified.
- `deletion_time` - When the Transfer Job was deleted.

» Import

Storage buckets can be imported using the Transfer Job's `project` and `name` without the `transferJob/` prefix, e.g.

```
$ terraform import google_storage_transfer_job.nightly-backup-transfer-job my-project-1asd32
```

» google_vpc_access_connector

Serverless VPC Access connector resource.

Warning: This resource is in beta, and should be used with the terraform-provider-google-beta provider. See Provider Versions for more details on beta resources.

To get more information about Connector, see:

- API documentation

- How-to Guides
  - Configuring Serverless VPC Access

» Example Usage - VPC Access Connector

```
provider "google-beta" {
}

resource "google_vpc_access_connector" "connector" {
  name          = "my-connector"
  provider      = google-beta
  region        = "us-central1"
  ip_cidr_range = "10.8.0.0/28"
  network       = "default"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the resource (Max 25 characters).
- `network` - (Required) Name of a VPC network.
- `ip_cidr_range` - (Required) The range of internal addresses that follows RFC 4632 notation. Example: `10.132.0.0/28`.
- `region` - (Required) Region where the VPC Access connector resides

---

- `min_throughput` - (Optional) Minimum throughput of the connector in Mbps. Default and min is 200.
- `max_throughput` - (Optional) Maximum throughput of the connector in Mbps, must be greater than `min_throughput`. Default is 1000.
- `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:

- `state` - State of the VPC access connector.
- `self_link` - The fully qualified name of this VPC connector

» Timeouts

This resource provides the following Timeouts configuration options:

- `create` - Default is 6 minutes.
- `delete` - Default is 10 minutes.

» Import

Connector can be imported using any of these accepted formats:

```
$ terraform import -provider=google-beta google_vpc_access_connector.default projects/{{proj
$ terraform import -provider=google-beta google_vpc_access_connector.default {{project}}/{{r
$ terraform import -provider=google-beta google_vpc_access_connector.default {{region}}/{{na
$ terraform import -provider=google-beta google_vpc_access_connector.default {{name}}
```

If you're importing a resource with beta features, make sure to include `-provider=google-beta` as an argument so that Terraform uses the correct provider to import your resource.

» User Project Overrides

This resource supports User Project Overrides.