

Oracle Cloud Infrastructure Classic Provider

The Oracle Cloud Infrastructure Classic provider (formerly know as the Oracle Public Cloud provider) is used to interact with the many resources supported by the Oracle Cloud Infrastructure Classic (<http://cloud.oracle.com/classic>) and Oracle Cloud at Customer (<https://cloud.oracle.com/cloud-at-customer>) infrastructure services. The provider needs to be configured with credentials for the Oracle Cloud Account.

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

Example Usage

```
# Configure the Oracle Cloud Infrastructure Classic provider
provider "opc" {
  user          = "..."
  password      = "..."
  identity_domain = "..."
  endpoint      = "..."
}

# Create an IP Reservation
resource "opc_compute_ip_reservation" "production" {
  parent_pool = "/oracle/public/ippool"
  permanent  = true
}
```

Argument Reference

The following arguments are supported:

- `user` - (Optional) The username to use, generally your email address. It can also be sourced from the `OPC_USERNAME` environment variable.
- `password` - (Optional) The password associated with the username to use. It can also be sourced from the `OPC_PASSWORD` environment variable.
- `identity_domain` - (Optional) The Identity Domain name (for Traditional accounts) or Identity Service ID (for IDCS accounts) of the environment to use. It can also be sourced from the `OPC_IDENTITY_DOMAIN` environment variable.
- `endpoint` - (Optional) The Compute Classic API endpoint to use, associated with your Oracle Cloud Account. This is known as the REST Endpoint within the Oracle portal. It can also be sourced from the `OPC_ENDPOINT` environment variable.
- `storage_endpoint` - (Optional) The Storage Classic API endpoint to use, associated with your Oracle Storage Cloud account. This is known as the REST Endpoint within the Oracle portal. Can also be set via the `OPC_STORAGE_ENDPOINT` environment variable.
- `storage_service_id` - (Optional) The Storage Service ID for authentication with the `storage_endpoint` If not set the `identity_domain` value is used. Can also be set via the `OPC_STORAGE_SERVICE_ID` environment variable.
- `max_retries` - (Optional) The maximum number of tries to make for a successful response when operating on resources. It can also be sourced from the `OPC_MAX_RETRIES` environment variable. Defaults to 1.

- `insecure` - (Optional) Skips TLS Verification for using self-signed certificates. Should only be used if absolutely needed.
Can also via setting the `OPC_INSECURE` environment variable to `true`.

Testing

Credentials must be provided via the `OPC_USERNAME`, `OPC_PASSWORD`, `OPC_IDENTITY_DOMAIN` and `OPC_ENDPOINT` environment variables in order to run acceptance tests.

opc_compute_image_list_entry

Use this data source to access the configuration of an Image List Entry.

Example Usage

```
data "opc_compute_image_list_entry" "foo" {
  image_list = "my_image_list"
  version = "version_of_my_list"
}

output "machine_images" {
  value = "${data.opc_compute_image_list_entry.foo.machine_images}"
}

output "single_image" {
  value = "${data.opc_compute_image_list_entry.foo.machine_images[1]}"
}
```

Argument Reference

- `image_list` - (Required) - The name of the image list to lookup.
- `version` - (Required) - The version (integer) of the Image List to use.
- `entry` - (Optional) - Which machine image to use. See `Entry` below for more details

Entry

The `entry` argument is fully optional when configuring the Data Source. If specified, however, the returned array of machine images will have a length of 1, and only contain the desired image.

Thus, if `"my_image_list"` is an image list that contains the following images:

```
["image1", "image2", "image3", "image4", "image5"]
```

Then specifying an `entry` of 3, the returned `machine_images` array will have a single element: `"image3"`. If `entry` was omitted, or set to 0, the returned `machine_images` array will contain all of the images for that image list version.

If the supplied `entry` value is invalid for the image list, Terraform will exit with an error, that the desired image list entry was not found.

Attributes Reference

- `dns` - Array of DNS servers for the interface.
- `attributes` - JSON object of all of the image list's attributes

- `machine_images` - An array of machine images as strings
- `uri` - The URI of the image list

opc_compute_ip_address_reservation

Use this data source to access the attributes of an existing IP Network IP Address Reservation.

Example Usage

```
data "opc_compute_ip_address_reservation" "example" {
  name = "/Compute-${var.domain}/${var.user}/ipaddressreservation1"
}

output "public_ip_address" {
  value = "${data.opc_compute_ip_address_reservation.example.ip_address}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the ip address reservation.

Attributes Reference

- `ip_address_pool` - The IP address pool from which the IP address is allocated.
- `description` - A description of the ip address reservation.
- `tags` - List of tags that applied to the IP address reservation.
- `ip_address` - The reserved IPv4 Public IP address.
- `uri` - The Uniform Resource Identifier of the ip address reservation

opc_compute_ip_reservation

Use this data source to access the attributes of an existing Shared Network IP Reservation.

Example Usage

```
data "opc_compute_ip_reservation" "example" {
  name = "/Compute-${var.domain}/${var.user}/309dd783-552d-4b8a-a3aa-a71109f703df"
}

output "public_ip_address" {
  value = "${data.opc_compute_ip_reservation.example.ip}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the IP Reservation.

Attributes Reference

- `ip` - The reserved IPv4 Public IP address.
- `permanent` - Whether the IP address remains reserved even when it is no longer associated with an instance.
- `parent_pool` - The pool from which to allocate the IP address.
- `tags` - List of tags applied to the IP reservation.
- `used` - indicates that the IP reservation is associated with an instance.

opc_compute_machine_image

Use this data source to access the configuration of an Machine Image.

Example Usage

```
data "opc_compute_machine_image" "foo" {
  account = "/Compute-${var.domain}/cloud_storage"
  name    = "Microsoft_Windows_Server_2012_R2"
}

output "attributes" {
  value = "${data.opc_compute_machine_image.foo.attributes}"
}
```

Argument Reference

- **account** - (Required) The two part name of the compute object storage account in the format `/Compute-{identity_domain}/cloud_storage`
- **name** - (Required) The name of the Machine Image.

Attributes Reference

- **file** - The name of the Machine Image .tar.gz file in the `compute_images` storage container.
- **description** - A description of the Machine Image.
- **attributes** - An optional JSON object of arbitrary attributes to be made available to the instance. These are user-defined tags. After defining attributes, you can view them from within an instance at <http://192.0.0.192/> (<http://192.0.0.192/>)
- **error_reason** - Description of the state of the machine image if there is an error.
- **hypervisor** - Dictionary of hypervisor-specific attributes.
- **image_format** - The format of the image.
- **platform** - The OS platform of the image.
- **state** - The state of the uploaded machine image.
- **uri** - The Uniform Resource Identifier for the Machine Image.

opc_compute_network_interface

Use this data source to access the configuration of an instance's network interface

Example Usage

```
data "opc_compute_network_interface" "foo" {
  instance_id   = "${opc_compute_instance.my_instance.id}"
  instance_name = "${opc_compute_instance.my_instance.name}"
  interface     = "eth0"
}

output "mac_address" {
  value = "${data.opc_compute_network_interface.foo.mac_address}"
}

output "vnic" {
  value = "${data.opc_compute_network_interface.foo.vnic}"
}
```

Argument Reference

- `instance_name` is the name of the instance.
- `instance_id` is the id of the instance.
- `interface` is the name of the attached interface. `eth0`, `eth1`, ... `eth9`.

Attributes Reference

- `dns` - Array of DNS servers for the interface.
- `ip_address` - IP Address assigned to the interface.
- `ip_network` - The IP Network assigned to the interface.
- `is_default_gateway` - Whether or not the the interface is the default gateway.
- `mac_address` - The MAC address of the interface.
- `model` - The model of the NIC card used.
- `name_servers` - Array of name servers for the interface.
- `nat` - The IP Reservation (in IP Networks) associated with the interface.
- `search_domains` - The search domains that are sent through DHCP as option 119.
- `sec_lists` - The security lists the interface is added to.
- `shared_network` - Whether or not the interface is inside the Shared Network or an IP Network.

- `vnic` - The name of the vNIC created for the IP Network.
- `vnic_sets` - The array of vNIC Sets the interface was added to.

opc_compute_ssh_key

Use this data source to access the attributes of an SSH Key.

Example Usage

```
data "opc_compute_ssh_key" "test" {
  name = "/Compute-${var.domain}/${var.user}/test-key"
}

output "public_ssh_key" {
  value = "${data.opc_compute_ssh_key.test.key}"
}
```

Argument Reference

- `name` - (Required) The unique (within this identity domain) name of the SSH key.

Attributes Reference

- `key` - The public SSH key.
- `enabled` - Whether or not the key is enabled.

opc_compute_storage_volume_snapshot

Use this data source to access the configuration of a storage volume snapshot.

Example Usage

```
data "opc_compute_storage_volume_snapshot" "foo" {
  name = "my-storage-volume/my-storage-volume-snapshot-name"
}
```

Argument Reference

- `name` is the name of the storage volume snapshot.

Attributes Reference

- `account` - Account of the snapshot.
- `collocated` - Boolean specifying whether the snapshot is collocated or remote.
- `description` - The description of the storage volume snapshot.
- `machine_image_name` - The name of the machine image that's used in the boot volume from which this snapshot is taken.
- `parent_volume_bootable` - Boolean specifying whether or not the snapshot's parent volume was bootable.
- `property` - Where the snapshot is stored, whether collocated, or in the Oracle Storage Cloud Service instance.
- `platform` - The OS platform this snapshot is compatible with
- `size` - The size of the snapshot in GB.
- `snapshot_timestamp` - Timestamp of the storage snapshot, generated by storage server. The snapshot will contain data written to the original volume before this time.
- `snapshot_id` - The Oracle ID of the snapshot.
- `start_timestamp` - Timestamp when the snapshot was started.
- `status` - Status of the snapshot.
- `status_detail` - Details about the latest state of the storage volume snapshot.
- `status_timestamp` - Indicates the time that the current view of the storage volume snapshot was generated.
- `tags` - Comma-separated strings that tag the storage volume.
- `uri` - Uniform Resource Identifier
- `volume_name` - The name of the storage volume that the snapshot was created from

opc_compute_vnic

Use this data source to access the configuration of a Virtual NIC.

Example Usage

```
data "opc_compute_vnic" "current" {
  name = "my_vnic_name"
}

output "mac_address" {
  value = "${data.opc_compute_vnic.current.mac_address}"
}
```

Argument Reference

- `name` is the name of the Virtual NIC.

Attributes Reference

- `description` is a description of the Virtual NIC.
- `mac_address` is the MAC Address of the Virtual NIC.
- `tags` is a list of Tags associated with the Virtual NIC.
- `transit_flag` is true if the Virtual NIC is of the type `transit`.
- `uri` is the Unique Resource Locator of the Virtual NIC.

opc_compute_acl

The `opc_compute_acl` resource creates and manages an ACL in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_acl" "default" {
  name      = "ACL1"
  description = "This is a description for an acl"
  tags      = ["tag1", "tag2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the ACL.
- `enabled` - (Optional) Enables or disables the ACL. Set to true by default.
- `description` - (Optional) A description of the ACL.
- `tags` - (Optional) List of tags that may be applied to the ACL.

In addition to the above, the following values are exported:

- `uri` - The Uniform Resource Identifier for the ACL

Import

ACL's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_acl.acl1 example
```

opc_compute_image_list

The `opc_compute_image_list` resource creates and manages an Image List in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_image_list" "test" {
  name      = "imagelist1"
  description = "This is a description of the Image List"
  default   = 21
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Image List.
- `description` - (Required) A description of the Image List.
- `default` - (Required) The image list entry to be used, by default, when launching instances using this image list. Defaults to 1.

Import

Image List's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_image_list.imagelist1 example
```

opc_compute_image_list_entry

The `opc_compute_image_list_entry` resource creates and manages an Image List Entry in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_image_list" "test" {
  name          = "imagelist1"
  description   = "This is a description of the Image List"
  default       = 21
}

resource "opc_compute_image_list_entry" "test" {
  name          = "${opc_compute_image_list.test.name}"
  machine_images = [ "/oracle/public/oel_6.7_apaaS_16.4.5_1610211300" ]
  version       = 1
  attributes    = <<JSON
{
  "foo": "bar"
}
JSON
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Image List.
- `machine_images` - (Required) An array of machine images.
- `version` - (Required) The unique version of the image list entry, as an integer.
- `attributes` - (Optional) JSON String of optional data that will be passed to an instance of this machine image when it is launched.

Attributes Reference

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

- `uri` - The Unique Resource Identifier for the Image List Entry.

Import

Image List's can be imported using the Name of the Image List, along with the Version of the Image List Entry, delimited via the `|` character, e.g.


```
$ terraform import opc_compute_image_list_entry.entry1 my_image_list|2
```

opc_compute_instance

The `opc_compute_instance` resource creates and manages an instance in an Oracle Cloud Infrastructure Compute Classic identity domain.

Caution: The `opc_compute_instance` resource can completely delete your instance just as easily as it can create it. To avoid costly accidents, consider setting `prevent_destroy` (/docs/configuration/resources.html#prevent_destroy) on your instance resources as an extra safety measure.

Example Usage

```
resource "opc_compute_ip_network" "test" {
  name           = "internal-network"
  description    = "Terraform Provisioned Internal Network"
  ip_address_prefix = "10.0.1.0/24"
  public_napt_enabled = false
}

resource "opc_compute_storage_volume" "test" {
  name = "internal"
  size = 100
}

resource "opc_compute_instance" "test" {
  name      = "instance1"
  label    = "Terraform Provisioned Instance"
  shape     = "oc3"
  image_list = "/oracle/public/oel_6.7_apaas_16.4.5_1610211300"

  storage {
    volume = "${opc_compute_storage_volume.test.name}"
    index  = 1
  }

  networking_info {
    index      = 0
    nat        = ["ippool:/oracle/public/ippool"]
    shared_network = true
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the instance.
- `shape` - (Required) The shape of the instance, e.g. `oc4`.
- `instance_attributes` - (Optional) A JSON string of custom attributes. See [Attributes](#) below for more information.
- `boot_order` - (Optional) The index number of the bootable storage volume, presented as a list, that should be used to

boot the instance. The only valid value is [1]. If you set this attribute, you must also specify a bootable storage volume with index number 1 in the volume sub-parameter of `storage_attachments`. When you specify `boot_order`, you don't need to specify the `imagelist` attribute, because the instance is booted using the image on the specified bootable storage volume. If you specify both `boot_order` and `imagelist`, the `imagelist` attribute is ignored.

- `hostname` - (Optional) The host name assigned to the instance. On an Oracle Linux instance, this host name is displayed in response to the `hostname` command. Only relative DNS is supported. The domain name is suffixed to the host name that you specify. The host name must not end with a period. If you don't specify a host name, then a name is generated automatically.
- `image_list` - (Optional) The `imageList` of the instance, e.g. `/oracle/public/oel_6.4_2GB_v1`.
- `label` - (Optional) The label to apply to the instance.
- `desired_state` - (Optional) Set the desire state of the instance to `running` (default) or `shutdown`. You can use this request to shut down and restart individual instances which use a persistent bootable storage volume.
- `networking_info` - (Optional) Information pertaining to an individual network interface to be created and attached to the instance. If left unspecified, the instance will be created within the `shared_network`. See Networking Info below for more information.
- `storage` - (Optional) Information pertaining to an individual storage attachment to be created during instance creation. Please see Storage Attachments below for more information.
- `reverse_dns` - (Optional) If set to `true` (default), then reverse DNS records are created. If set to `false`, no reverse DNS records are created.
- `ssh_keys` - (Optional) A list of the names of the SSH Keys that can be used to log into the instance.
- `tags` - (Optional) A list of strings that should be supplied to the instance as tags.

Attributes

During instance creation, there are several custom attributes that a user may wish to make available to the instance during instance creation. These attributes can be specified via the `instance_attributes` field, and must be presented as a string in JSON format. The easiest way to populate this field is with a HEREDOC:

```

resource "opc_compute_instance" "foo" {
  name = "test"
  label = "test"
  shape = "oc3"
  imageList = "/oracle/public/oel_6.4_2GB_v1"
  instance_attributes = <<JSON
{
  "foo": "bar",
  "baz": 42,
  "my_obj": {
    "my_key": false,
    "another": true
  }
}
JSON

  sshKeys = ["${opc_compute_ssh_key.key1.name}"]
}

```

This allows the user to have full control over the attributes supplied to an instance during instance creation. There are, as well, some attributes that get populated during instance creation, and the full attributes map can be seen via the exported attributes attribute.

Warning: Due to how Terraform imports resources, the instance_attributes field will *only* be populated when creating a new instance *with terraform*. This requires us to ignore any state diffs on changes to the instance_attributes field. Thus, any configuration changes in the instance_attributes field, will not register a diff during a plan or apply. If a user wishes to make a change solely to the supplied instance attributes, and recreate the instance resource, `terraform taint` is the best solution. You can read more about the taint command here (<https://www.terraform.io/docs/commands/taint.html>)

Networking Info

Each `networking_info` config manages a single network interface for the instance. The attributes are either required or optional depending on whether or not the interface is in the Shared Network, or an IP Network. Some attributes can only be used if the interface is in the Shared Network, and same for an interface in an IP Network.

Note: If `networking_info` is left unspecified, the instance will be booted in the `shared_network`.

The following attributes are supported:

- `index` - (Required) The numerical index of the network interface. Specified as an integer to allow for use of `count`, but directly maps to `ethX`. ie: With `index` set to `0`, the interface `eth0` will be created. Can only be `0-9`.
- `dns` - (Optional, IP Network Only) List of DNS A record names for the instance. You can specify up to eight DNS A record names for each interface on an IP network. These names can be queried by instances on any IP network in the same IP network exchange.
- `ip_address` - (Optional, IP Network Only) IP Address assigned to the interface.
- `ip_network` - (Optional, IP Network Only) The IP Network assigned to the interface.
- `mac_address` - (Optional, IP Network Only) The MAC address of the interface.
- `is_default_gateway` - (Optional, IP Network Only) Specify the interface is to be used as the default gateway for all traffic. Only one interface on an instance can be specified as the default gateway. If the instance has an interface on

the shared network, that interface is always used as the default gateway.

- `model` - (Optional, Shared Network Only) The model of the NIC card used. Must be set to `e1000`.
- `name_servers` - (Optional) Array of name servers for the interface.
- `nat` - (Optional for IP Networks, Required for the Shared Network) The IP Reservations associated with the interface (IP Network). Indicates whether a temporary or permanent public IP address should be assigned to the instance (Shared Network).
- `search_domains` - (Optional) The search domains that are sent through DHCP as option 119.
- `sec_lists` - (Optional, Shared Network Only) The security lists the interface is added to.
- `shared_network` - (Required) Whether or not the interface is inside the Shared Network or an IP Network.
- `vnic` - (Optional, IP Network Only) The name of the vNIC created for the IP Network.
- `vnic_sets` - (Optional, IP Network Only) The array of vNIC Sets the interface was added to.

Storage Attachments

Each Storage Attachment config manages a single storage attachment that is created *during instance creation*. This means that any storage attachments created during instance creation cannot be detached from the instance. Use the `resource_storage_attachment` resource to manage storage attachments for instances if you wish to detach the storage volumes at a later date.

The following attributes are supported:

- `index` - (Required) The Index number of the volume attachment. 1 is the boot volume for the instance. Values 1-10 allowed.
- `volume` - (Required) The name of the storage volume to attach to the instance.

In addition to the above attributes, the following attributes are exported for a storage volume

- `name` - Name of the storage volume attachment.

Attributes Reference

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

- `id` - The id of the instance.
- `attributes` - The full attributes of the instance, as a JSON string.
- `availability_domain` - The availability domain the instance is in.
- `domain` - The default domain to use for the hostname and for DNS lookups.
- `entry` - Imagelist entry number.
- `fingerprint` - SSH server fingerprint presented by the instance.
- `fqdn` - The fully qualified domain name of the instance.

- `image_format` - The format of the image.
- `ip_address` - The IP Address of the instance.
- `placement_requirements` - The array of placement requirements for the instance.
- `platform` - The OS Platform of the instance.
- `priority` - The priority at which the instance was ran.
- `quota_reservation` - Reference to the QuotaReservation, to be destroyed with the instance.
- `relationships` - The array of relationship specifications to be satisfied on instance placement.
- `resolvers` - Array of resolvers to be used instead of the default resolvers.
- `site` - The site the instance is running on.
- `start_time` - The launch time of the instance.
- `state` - The instance's state.
- `vcable_id` - vCable ID for the instance.
- `virtio` - Boolean that determines if the instance is a virtio device.
- `vnc_address` - The VNC address and port of the instance.

Import

Instances can be imported using the Instance's combined Name and ID with a / character separating them. If viewing an instance in the Oracle Web Console, the instance's name and id are the last two fields in the instances fully qualified Name

For example, in the Web Console an instance's fully qualified name is: `/Compute-
<identify>/<user>@<account>/<instance_name>/<instance_id>`

The instance can be imported as such:

```
$ terraform import opc_compute_instance.instance1 instance_name/instance_id
```

Timeouts

`opc_compute_instance` provides the following Timeouts (</docs/configuration/resources.html#timeouts>) configuration options:

- `create` - (Default 20 minutes) Used for Creating Instances.
- `update` - (Default 20 minutes) Used for updating Instances.
- `delete` - (Default 20 minutes) Used for Deleting Instances.

opc_compute_ip_address_association

The `opc_compute_ip_address_association` resource creates and manages an IP address association between an IP address reservation and a virtual NIC in an Oracle Cloud Infrastructure Compute Classic identity domain, for an IP Network.

Example Usage

```
resource "opc_compute_ip_address_association" "default" {
  name                = "IPAddressAssociation1"
  ip_address_reservation = "${opc_compute_ip_address_reservation.default.name}"
  vnic                = "${data.opc_compute_vnic.default.name}"
  tags                = ["tags1", "tags2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the ip address association.
- `ip_address_reservation` - (Optional) The name of the NAT IP address reservation.
- `vnic` - (Optional) The name of the virtual NIC associated with this NAT IP reservation.
- `description` - (Optional) A description of the ip address association.
- `tags` - (Optional) List of tags that may be applied to the ip address association.

In addition to the above, the following variables are exported:

- `uri` - (Computed) The Uniform Resource Identifier of the ip address association.

Import

IP Address Associations can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_ip_address_association.default example
```

opc_compute_ip_address_prefix_set

The `opc_compute_ip_address_prefix_set` resource creates and manages an IP address prefix set in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_ip_address_prefix_set" "default" {  
  name      = "PrefixSet1"  
  prefixes = ["192.168.0.0/16", "172.120.0.0/24"]  
  tags      = ["tags1", "tags2"]  
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the ip address prefix set.
- `prefixes` - (Optional) List of CIDR IPv4 prefixes assigned in the virtual network.
- `description` - (Optional) A description of the ip address prefix set.
- `tags` - (Optional) List of tags that may be applied to the ip address prefix set.

In addition to the above, the following variables are exported:

- `uri` - (Computed) The Uniform Resource Identifier of the ip address prefix set.

Import

IP Address Prefix Set can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_ip_address_prefix_set.default example
```


opc_compute_ip_address_reservation

The `opc_compute_ip_address_reservation` resource creates and manages an IP address reservation in an Oracle Cloud Infrastructure Compute Classic identity domain, for an IP Network.

Example Usage

```
resource "opc_compute_ip_address_reservation" "default" {
  name          = "IPAddressReservation1"
  ip_address_pool = "public-ippool"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the ip address reservation.
- `ip_address_pool` - (Required) The IP address pool from which you want to reserve an IP address. Typically one of either `public-ippool` or `cloud-ippool`.
- `description` - (Optional) A description of the ip address reservation.
- `tags` - (Optional) List of tags that may be applied to the IP address reservation.

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

- `ip_address` - Reserved NAT IPv4 address from the IP address pool.
- `uri` - The Uniform Resource Identifier of the ip address reservation

Import

IP Address Reservations can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_ip_address_reservation.default example
```

opc_compute_ip_association

The `opc_compute_ip_association` resource creates and manages an association between an IP address and an instance in an Oracle Cloud Infrastructure Compute Classic identity domain, for the Shared Network.

Example Usage

```
resource "opc_compute_ip_association" "instance1_reservation1" {  
  vcable      = "${opc_compute_instance.test_instance.vcable}"  
  parent_pool = "ipreservation:${opc_compute_ip_reservation.reservation1.name}"  
}
```

Argument Reference

The following arguments are supported:

- `vcable` - (Required) The vcable of the instance to associate the IP address with.
- `parent_pool` - (Required) The pool from which to take an IP address. To associate a specific reserved IP address, use the prefix `ipreservation:` followed by the name of the IP reservation. To allocate an IP address from a pool, use the prefix `ippool:`, e.g. `ippool:/oracle/public/ippool`.

Attributes Reference

The following attributes are exported:

- `name` The name of the IP Association

Import

IP Associations can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_ip_association.association1 example
```

opc_compute_ip_network

The `opc_compute_ip_network` resource creates and manages an IP Network in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_ip_network" "foo" {
  name           = "my-ip-network"
  description    = "my IP Network"
  ip_address_prefix = "10.0.1.0/24"
  ip_network_exchange = "${opc_compute_ip_exchange.foo.name}"
  public_napt_enabled = false
  tags           = ["tag1", "tag2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the IP Network. Changing this name forces a new resource to be created.
- `ip_address_prefix` - (Required) The IPv4 address prefix, in CIDR format.
- `description` - (Optional) The description of the IP Network.
- `ip_network_exchange` - (Optional) Specify the IP Network exchange to which the IP Network belongs to.
- `public_napt_enabled` - (Optional) If true, enable public internet access using NAPT for VNICs without any public IP Reservation. Defaults to `false`.

Attributes Reference

The following attributes are exported:

- `name` - The name of the IP Network
- `ip_address_prefix` - The IPv4 address prefix, in CIDR format.
- `description` - The description of the IP Network.
- `ip_network_exchange` - The IP Network Exchange for the IP Network
- `public_napt_enabled` - Whether public internet access using NAPT for VNICs without any public IP Reservation or not.
- `uri` - Uniform Resource Identifier for the IP Network

Import

IP Networks can be imported using the resource name, e.g.

```
$ terraform import opc_compute_ip_network.default example
```

opc_compute_ip_network_exchange

The `opc_compute_ip_network_exchange` resource creates and manages an IP network exchange in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_ip_network_exchange" "default" {  
  name = "NetworkExchange1"  
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the ip network exchange.
- `description` - (Optional) A description of the ip network exchange.
- `tags` - (Optional) List of tags that may be applied to the IP network exchange.

Import

IP Network Exchange's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_ip_network_exchange.exchange1 example
```

opc_compute_ip_reservation

The `opc_compute_ip_reservation` resource creates and manages an IP reservation in an Oracle Cloud Infrastructure Compute Classic identity domain for the Shared Network.

Example Usage

```
resource "opc_compute_ip_reservation" "reservation1" {
  parent_pool = "/oracle/public/ippool"
  permanent   = true
  tags        = [ "test" ]
}
```

Argument Reference

The following arguments are supported:

- `permanent` - (Required) Whether the IP address remains reserved even when it is no longer associated with an instance (if true), or may be returned to the pool and replaced with a different IP address when an instance is restarted, or deleted and recreated (if false).
- `parent_pool` - (Optional) The pool from which to allocate the IP address. Defaults to `/oracle/public/ippool`, and is currently the only acceptable input.
- `name` - (Optional) Name of the IP Reservation. Will be generated if unspecified.
- `tags` - (Optional) List of tags that may be applied to the IP reservation.

Attributes Reference

- `ip` - The Public IP address.
- `used` - indicates that the IP reservation is associated with an instance.

Import

IP Reservations can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_ip_reservations.reservation1 example
```

opc_compute_machine_image

The `opc_compute_machine_image` resource creates and manages a machine image template of a virtual hard disk of a specific size with an installed operating system.

Before performing this creating the Machine Image, you must upload your machine image file to Oracle Cloud Infrastructure Object Storage Classic `compute_images` container

Example Usage

```
resource "opc_compute_machine_image" "centos" {
  account      = "/Compute-${var.domain}/cloud_storage"
  name         = "CentOS_7"
  file         = "CentOS-7-x86_64-OracleCloud.raw.tar.gz"
  description  = "CentOS 7"
}
```

Argument Reference

The following arguments are supported:

- `account` - (Required) The two part name of the compute object storage account in the format `/Compute-{identity_domain}/cloud_storage`
- `name` - (Required) The name of the Machine Image.
- `file` - (Required) The name of the Machine Image `.tar.gz` file in the `compute_images` storage container.
- `description` - (Optional) A description of the Machine Image.
- `attributes` - (Optional) An optional JSON object of arbitrary attributes to be made available to the instance. These are user-defined tags. After defining attributes, you can view them from within an instance at `http://192.0.0.192/` (`http://192.0.0.192/`)

In addition to the above, the following values are exported:

- `error_reason` - Description of the state of the machine image if there is an error.
- `hypervisor` - Dictionary of hypervisor-specific attributes.
- `image_format` - The format of the image.
- `platform` - The OS platform of the image.
- `state` - The state of the uploaded machine image.
- `uri` - The Uniform Resource Identifier for the Machine Image.

Import

Machine Images can be imported using the resource name, e.g.

```
$ terraform import opc_compute_machine_image.machine_image1 example
```


opc_compute_orchestrated_instance

The `opc_compute_orchestrated_instance` resource creates and manages an orchestration containing a number of instances in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_orchestrated_instance" "default" {
  name          = "default_orchestration"
  desired_state = "active"

  instance {
    name      = "default-orchestrated-instance"
    label     = "Orchestrated Instance 1 Label"
    shape     = "oc3"
    image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
  }
}
```

Example Usage with Multiple Instances

```
resource "opc_compute_orchestrated_instance" "default" {
  name          = "default_orchestration"
  desired_state = "active"

  instance {
    name      = "default-instance-1"
    label     = "Instance One"
    shape     = "oc3"
    image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
  }

  instance {
    name      = "default-instance-2"
    label     = "Instance Two"
    shape     = "oc3"
    image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
  }
}
```

Example Usage with IP Networking

```

resource "opc_compute_ip_network" "default" {
  name          = "default-ip-network"
  description    = "testing-ip-network-instance"
  ip_address_prefix = "10.1.12.0/24"
}

resource "opc_compute_orchestrated_instance" "default" {
  name          = "default_orchestration"
  desired_state = "active"

  instance {
    name      = "default-instance"
    label     = "Default Instance"
    shape     = "oc3"
    image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"

    networking_info {
      index      = 0
      ip_network  = "${opc_compute_ip_network.default.id}"
      vnic        = "default-ip-network"
      shared_network = false
    }
  }
}

```

Example Usage with Storage

```

resource "opc_compute_storage_volume" "foo" {
  name = "acc-test-orchestration-%d"
  size = 1
}

resource "opc_compute_storage_volume" "bar" {
  name = "acc-test-orchestration-2-%d"
  size = 1
}

resource "opc_compute_orchestrated_instance" "default" {
  name          = "test_orchestration-%d"
  desired_state = "active"

  instance {
    name      = "default-instance"
    label     = "Default Instance"
    shape     = "oc3"
    image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"

    storage {
      volume = "${opc_compute_storage_volume.foo.name}"
      index  = 1
    }

    storage {
      volume = "${opc_compute_storage_volume.bar.name}"
      index  = 2
    }
  }
}

```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the orchestration.
- `desired_state` - (Required) The desired state of the orchestration. Permitted values are:
 - `active`: all resource (instances) declared in the orchestration are created
 - `suspend`: all resources (instances) declared in the orchestration are removed unless the instances has `persistent = true`
 - `inactive`: all resources (instances) declared in the orchestration are removed including the instances that have `persistent = true`
- `instance` - (Required) The information pertaining to creating an instance through the orchestration API.
- `description` - (Optional) The description of the orchestration.

Instance

Instance supports the arguments found in `opc_compute_instance`

(https://www.terraform.io/docs/providers/opc/r/opc_compute_instance.html) and the following:

- `persistent` - (Optional) Determines whether the instance will persist when the orchestration is suspended. Defaults to `false`.

In addition to the above, the following values are exported:

- `uri` - The Uniform Resource Identifier for the Orchestration
- `version` - (Optional) The version of the orchestration.

opc_compute_route

The `opc_compute_route` resource creates and manages a route for an IP Network in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_route" "foo" {
  name           = "my-route"
  description     = "my IP Network route"
  admin_distance = 1
  ip_address_prefix = "10.0.1.0/24"
  next_hop_vnic_set = "${opc_compute_vnic_set.bar.name}"
  tags           = ["tag1", "tag2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the route.
- `description` - (Optional) The description of the route.
- `admin_distance` - (Optional) The route's administrative distance. Defaults to 0.
- `ip_address_prefix` - (Required) The IPv4 address prefix, in CIDR format, of the external network from which to route traffic.
- `next_hop_vnic_set` - (Required) Name of the virtual NIC set to route matching packets to. Routed flows are load-balanced among all the virtual NICs in the virtual NIC set.

Attributes Reference

The following attributes are exported:

- `name` The name of the route
- `description` - The description of the route.
- `admin_distance` - The route's administrative distance. Defaults to 0.
- `ip_address_prefix` - The IPv4 address prefix, in CIDR format, of the external network from which to route traffic.
- `next_hop_vnic_set` - Name of the virtual NIC set to route matching packets to. Routed flows are load-balanced among all the virtual NICs in the virtual NIC set.

Import

Route's can be imported using the `resource` name, e.g.

```
$ terraform import opc_compute_route.route1 example
```

opc_compute_sec_rule

The `opc_compute_sec_rule` resource creates and manages a sec rule in an Oracle Cloud Infrastructure Compute Classic identity domain, which joins together a source security list (or security IP list), a destination security list (or security IP list), and a security application.

Example Usage

```
resource "opc_compute_sec_rule" "test_rule" {
  name          = "test"
  source_list    = "seclist:${opc_compute_security_list.sec-list1.name}"
  destination_list = "seclist:${opc_compute_security_ip_list.sec-ip-list1.name}"
  action         = "permit"
  application    = "${opc_compute_security_application.spring-boot.name}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The unique (within the identity domain) name of the security rule.
- `description` - (Optional) A description for this security rule.
- `source_list` - (Required) The source security list (prefixed with `seclist:`), or security IP list (prefixed with `seclist:`).
 - `destination_list` - (Required) The destination security list (prefixed with `seclist:`), or security IP list (prefixed with `seclist:`).
- `application` - (Required) The name of the application to which the rule applies.
- `action` - (Required) Whether to permit, refuse or deny packets to which this rule applies. This will ordinarily be `permit`.
- `disabled` - (Optional) Whether to disable this security rule. This is useful if you want to temporarily disable a rule without removing it outright from your Terraform resource definition. Defaults to `false`.

In addition to the above, the following values are exported:

- `uri` - The Uniform Resource Identifier of the sec rule.

Import

Sec Rule's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_sec_rule.rule1 example
```

opc_compute_security_application

The `opc_compute_security_application` resource creates and manages a security application in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage (TCP)

```
resource "opc_compute_security_application" "tomcat" {
  name      = "tomcat"
  protocol  = "tcp"
  dport     = "8080"
}
```

Example Usage (ICMP)

```
resource "opc_compute_security_application" "tomcat" {
  name      = "tomcat"
  protocol  = "icmp"
  icmp_type = "echo"
  icmp_code = "protocol"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The unique (within the identity domain) name of the application
- `protocol` - (Required) The protocol to enable for this application. Must be one of `tcp`, `udp`, `ah`, `esp`, `icmp`, `icmpv6`, `igmp`, `ipip`, `gre`, `mplsip`, `ospf`, `pim`, `rdp`, `sctp` or `all`.
- `dport` - (Required) The port, or range of ports, to enable for this application, e.g `8080`, `6000-7000`. This must be set if the `protocol` is set to `tcp` or `udp`.
- `icmp_type` - (Optional) The ICMP type to enable for this application, if the `protocol` is `icmp`. Must be one of `echo`, `reply`, `ttl`, `traceroute`, `unreachable`.
- `icmp_code` - (Optional) The ICMP code to enable for this application, if the `protocol` is `icmp`. Must be one of `admin`, `df`, `host`, `network`, `port` or `protocol`.

Import

Security Application's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_security_application.application1 example
```


opc_compute_security_association

The `opc_compute_security_association` resource creates and manages an association between an instance and a security list in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_security_association" "test_instance_sec_list_1" {  
  name      = "association1"  
  vcable    = "${opc_compute_instance.test_instance.vcable}"  
  seclist   = "${opc_compute_security_list.sec_list1.name}"  
}
```

Argument Reference

The following arguments are supported:

- `name` - (Optional) The Name for the Security Association. If not specified, one is created automatically. Changing this forces a new resource to be created.
- `vcable` - (Required) The `vcable` of the instance to associate to the security list.
- `seclist` - (Required) The name of the security list to associate the instance to.

Import

Security Association's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_security_association.association1 example
```

opc_compute_security_ip_list

The `opc_compute_security_ip_list` resource creates and manages a security IP list in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_security_ip_list" "sec_ip_list1" {  
  name      = "sec-ip-list1"  
  ip_entries = ["217.138.34.4"]  
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The unique (within the identity domain) name of the security IP list.
- `ip_entries` - (Required) The IP addresses to include in the list.
- `description` - (Optional) The description of the security ip list.

Import

IP List's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_ip_list.list1 example
```

opc_compute_security_list

The `opc_compute_security_list` resource creates and manages a security list in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_security_list" "sec_list1" {
  name           = "sec-list-1"
  policy         = "permit"
  outbound_cidr_policy = "deny"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The unique (within the identity domain) name of the security list.
- `policy` - (Required) The policy to apply to instances associated with this list. Must be one of `permit`, `reject` (packets are dropped but a reply is sent) and `deny` (packets are dropped and no reply is sent).
- `output_cidr_policy` - (Required) The policy for outbound traffic from the security list. Must be one of `permit`, `reject` (packets are dropped but a reply is sent) and `deny` (packets are dropped and no reply is sent).

Import

Security List's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_security_list.list1 example
```

opc_compute_security_protocol

The `opc_compute_security_protocol` resource creates and manages a security protocol in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_security_protocol" "default" {
  name      = "security-protocol-1"
  dst_ports = ["2045-2050"]
  src_ports = ["3045-3060"]
  ip_protocol = "tcp"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the security protocol.
- `dst_ports` (Optional) Enter a list of port numbers or port range strings. Traffic is enabled by a security rule when a packet's destination port matches the ports specified here. For TCP, SCTP, and UDP, each port is a destination transport port, between 0 and 65535, inclusive. For ICMP, each port is an ICMP type, between 0 and 255, inclusive. If no destination ports are specified, all destination ports or ICMP types are allowed.
- `src_ports` (Optional) Enter a list of port numbers or port range strings. Traffic is enabled by a security rule when a packet's source port matches the ports specified here. For TCP, SCTP, and UDP, each port is a source transport port, between 0 and 65535, inclusive. For ICMP, each port is an ICMP type, between 0 and 255, inclusive. If no source ports are specified, all source ports or ICMP types are allowed.
- `ip_protocol` (Optional) The protocol used in the data portion of the IP datagram. Permitted values are: `tcp`, `udp`, `icmp`, `igmp`, `ipip`, `rdp`, `esp`, `ah`, `gre`, `icmpv6`, `ospf`, `pim`, `sctp`, `mplsip`, `all`. Traffic is enabled by a security rule when the protocol in the packet matches the protocol specified here. If no protocol is specified, all protocols are allowed.
- `description` - (Optional) A description of the security protocol.
- `tags` - (Optional) List of tags that may be applied to the security protocol.

In addition to the above, the following values are exported:

- `uri` - The Uniform Resource Identifier for the Security Protocol

Import

ACL's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_security_protocol.default example
```


opc_compute_security_rule

The `opc_compute_security_rule` resource creates and manages a security rule in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_security_rule" "default" {
  name           = "SecurityRule1"
  flow_direction = "ingress"
  acl            = "${opc_compute_acl.default.name}"
  security_protocols = ["${opc_compute_security_protocol.default.name}"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the security rule.
- `flow_direction` - (Required) Specify the direction of flow of traffic, which is relative to the instances, for this security rule. Allowed values are `ingress` or `egress`.
- `disabled` - (Optional) Whether to disable this security rule. This is useful if you want to temporarily disable a rule without removing it outright from your Terraform resource definition. Defaults to `false`.
- `acl` - (Optional) Name of the ACL that contains this security rule.
- `dst_ip_address_prefixes` - (Optional) List of IP address prefix set names to match the packet's destination IP address.
- `src_ip_address_prefixes` - (Optional) List of names of IP address prefix set to match the packet's source IP address.
- `dst_vnic_set` - (Optional) Name of virtual NIC set containing the packet's destination virtual NIC.
- `src_vnic_set` - (Optional) Name of virtual NIC set containing the packet's source virtual NIC.
- `security_protocols` - (Optional) List of security protocol object names to match the packet's protocol and port.
- `description` - (Optional) A description of the security rule.
- `tags` - (Optional) List of tags that may be applied to the security rule.

Attributes Reference

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

- `uri` - The Uniform Resource Identifier of the security rule.

Import

Security Rule's can be imported using the `resource` name, e.g.

```
$ terraform import opc_compute_security_rule.rule1 example
```

opc_compute_ssh_key

The `opc_compute_ssh_key` resource creates and manages an SSH key in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_ssh_key" "test" {  
  name      = "test-key"  
  key       = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ Cqw6JwbjIk..."  
  enabled   = true  
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The unique (within this identity domain) name of the SSH key.
- `key` - (Required) The SSH key itself
- `enabled` - (Optional) Whether or not the key is enabled. This is useful if you want to temporarily disable an SSH key, without removing it entirely from your Terraform resource definition. Defaults to `true`

Import

SSH Key's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_ssh_key.key1 example
```


opc_compute_storage_volume

The `opc_compute_storage_volume` resource creates and manages a storage volume in an Oracle Cloud Infrastructure Compute Classic identity domain.

Caution: The `opc_compute_storage_volume` resource can completely delete your storage volume just as easily as it can create it. To avoid costly accidents, consider setting `prevent_destroy` (/docs/configuration/resources.html#prevent_destroy) on your storage volume resources as an extra safety measure.

Example Usage

```
resource "opc_compute_storage_volume" "test" {
  name          = "storageVolume1"
  description   = "Description for the Storage Volume"
  size          = 10
  tags          = ["bar", "foo"]
}
```

Example Usage (Bootable Volume)

```
data "opc_compute_image_list_entry" "test" {
  image_list = "my_image_list"
  version    = 1
}

resource "opc_compute_storage_volume" "test" {
  name          = "storageVolume1"
  description   = "Description for the Bootable Storage Volume"
  size          = 30
  tags          = ["first", "second"]
  bootable      = true
  image_list    = "${data.opc_compute_image_list_entry.test.image_list}"
  image_list_entry = "${data.opc_compute_image_list_entry.test.version}"
}
```

Argument Reference

The following arguments are supported:

- `name` (Required) The name for the Storage Account.
- `description` (Optional) The description of the storage volume.
- `size` (Required) The size of this storage volume in GB. The allowed range is from 1 GB to 2 TB (2048 GB).
- `storage_type` - (Optional) - The Type of Storage to provision. Defaults to `/oracle/public/storage/default`.
- `bootable` - (Optional) Is the Volume Bootable? Defaults to `false`.

- `image_list` - (Optional) Defines an image list.
- `image_list_entry` - (Optional) Defines an image list entry.
- `snapshot` - (Optional) The name of the parent snapshot from which the storage volume is restored or cloned. See Snapshots, below for more information.
- `snapshot_id` - (Optional) The Id of the parent snapshot from which the storage volume is restored or cloned. See Snapshots, below for more information.
- `snapshot_account` - (Optional) The Account of the parent snapshot from which the storage volume is restored. See Snapshots, below for more information.
- `tags` - (Optional) Comma-separated strings that tag the storage volume.

Attributes Reference

The following attributes are exported:

- `hypervisor` - The hypervisor that this volume is compatible with.
- `machine_image` - Name of the Machine Image - available if the volume is a bootable storage volume.
- `managed` - Is this a Managed Volume?
- `platform` - The OS platform this volume is compatible with.
- `readonly` - Can this Volume be attached as readonly?
- `status` - The current state of the storage volume.
- `storage_pool` - The storage pool from which this volume is allocated.
- `uri` - Unique Resource Identifier of the Storage Volume.

Import

Storage Volume's can be imported using the `resource` name, e.g.

```
$ terraform import opc_compute_storage_volume.volume1 example
```

Snapshots

Restoring a storage volume from a snapshot can happen via the use of the `snapshot`, `snapshot_id`, and `snapshot_account` attributes.

Note: There isn't a mechanism to restore a snapshot from a different domain, the `snapshot_account` argument needs to reference the same domain.

There are two different ways to identify the snapshot in which to restore the storage volume:

1. Use the unique `snapshot_id` argument.
2. Use a combination of `snapshot_account` and `snapshot` arguments. `snapshot_account` is in the format of: `/Compute-mydomain/cloud_storage`, and `snapshot` is a concatenated name of the original storage volume name, and it's child snapshot name: `mystorage/mysnapshot`.

Example 1:

```
resource "opc_compute_storage_volume" "foo" {
  name = "from-snapshot"
  size = "10"
  snapshot_account = "/Compute-${var.domain}/cloud_storage"
  snapshot = "my-boot-volume/my-snapshot"
}
```

Example 2:

```
data "opc_compute_storage_volume_snapshot" "foo" {
  name = "my-boot-volume/my-snapshot"
}

resource "opc_compute_storage_volume" "foo" {
  name = "from-snapshot"
  snapshot_id = "${opc_compute_storage_volume_snapshot.foo.snapshot_id}"
  size = "${opc_compute_storage_volume_snapshot.foo.snapshot_id}"
}
```

Colocated Snapshots

Colocated snapshots are stored in the same physical location as the original storage volume. While creating a storage volume from a colocated snapshot, you must specify values for the following parameters:

- `snapshot` the fully qualified multipart name of the snapshot.
- `size` set to the size of the source snapshot volume.
- `storage_type` same type as the parent storage volume of the snapshot.

If you are restoring a bootable storage volume from a snapshot, additionally you must specify the value for `bootable` as `true`. `snapshot_id` and `snapshot_source` are *not* used when creating a volume from a colocated snapshot.

Example:

```
data "opc_compute_storage_volume_snapshot" "snapshot1" {
  name = "my-bootable-storage-volume/my-colocated-snapshot"
}

resource "opc_compute_storage_volume" "volume1" {
  name          = "volume-from-colocated-storage-snapshot"
  snapshot      = "/Compute-${var.domain}/${var.user}/${data.opc_compute_storage_volume_snapshot.snapshot1.name}"
  size          = "${data.opc_compute_storage_volume_snapshot.snapshot1.size}"
  storage_type  = "/oracle/public/storage/default"
  bootable      = "${data.opc_compute_storage_volume_snapshot.snapshot1.parent_volume_bootable}"
}
```

Timeouts

`opc_compute_storage_volume` provides the following Timeouts (</docs/configuration/resources.html#timeouts>) configuration options:

- `create` - (Default 30 minutes) Used for Creating Storage Volumes.
- `update` - (Default 30 minutes) Used for Modifying Storage Volumes.
- `delete` - (Default 30 minutes) Used for Deleting Storage Volumes.

opc_compute_storage_volume_attachment

The `opc_compute_storage_volume_attachment` resource creates and manages a storage volume attachment in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_storage_volume" "default" {
  name = "storage-volume-1"
  size = 1
}

resource "opc_compute_instance" "default" {
  name = "instance-1"
  label = "instance-1"
  shape = "oc3"
  image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
}

resource "opc_compute_storage_attachment" "test" {
  instance = "${opc_compute_instance.default.name}"
  storage_volume = "${opc_compute_storage_volume.default.name}"
  index = 1
}
```

Argument Reference

The following arguments are supported:

- `instance` - (Required) The name of the instance the volume will be attached to.
- `storage_volume` - (Required) The name of the storage volume that will be attached to the instance
- `index` - (Required) The index on the instance that the storage volume will be attached to.

opc_compute_storage_volume_snapshot

The `opc_compute_storage_volume_snapshot` resource creates and manages a storage volume snapshot in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_storage_volume_snapshot" "test" {
  name          = "storageVolume1"
  description   = "Description for the Storage Volume"
  tags          = ["bar", "foo"]
  collocated    = true
  volume_name   = "${opc_compute_storage_volume.foo.name}"
}
```

Argument Reference

The following arguments are supported:

- `volume_name` (Required) The name of the storage volume to create the snapshot from.
- `description` (Optional) The description of the storage volume snapshot.
- `name` (Optional) The name of the storage volume snapshot. Will be generated if unspecified.
- `parent_volume_bootable` (Optional) A string value of whether or not the parent volume is 'bootable' or not. Defaults to "false".
- `collocated` (Optional) Boolean specifying whether the snapshot is collocated or remote. Defaults to `false`.
- `tags` - (Optional) Comma-separated strings that tag the storage volume.

Attributes Reference

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

- `account` - Account to use for snapshots.
- `machine_image_name` - The name of the machine image that's used in the boot volume from which this snapshot is taken.
- `size` - The size of the snapshot in GB.
- `property` - Where the snapshot is stored, whether collocated, or in the Oracle Storage Cloud Service instance.
- `platform` - The OS platform this snapshot is compatible with
- `snapshot_timestamp` - Timestamp of the storage snapshot, generated by storage server. The snapshot will contain data written to the original volume before this time.

- `snapshot_id` - The Oracle ID of the snapshot.
- `start_timestamp` - Timestamp when the snapshot was started.
- `status` - Status of the snapshot.
- `status_detail` - Details about the latest state of the storage volume snapshot.
- `status_timestamp` - Indicates the time that the current view of the storage volume snapshot was generated.
- `uri` - Uniform Resource Identifier

Import

Storage Volume Snapshot's can be imported using the `resource` name, e.g.

```
$ terraform import opc_compute_storage_volume_snapshot.volume1 example
```

Timeouts

`opc_compute_storage_volume_snapshot` provides the following Timeouts (</docs/configuration/resources.html#timeouts>) configuration options:

- `create` - (Default 30 minutes) Used for Creating Storage Volume Snapshots.
- `delete` - (Default 30 minutes) Used for Deleting Storage Volume Snapshots.

opc_compute_vnic_set

The `opc_compute_vnic_set` resource creates and manages a virtual NIC set in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_vnic_set" "test_set" {  
  name          = "test_vnic_set"  
  description   = "My vnic set"  
  applied_acls  = ["acl1", "acl2"]  
  virtual_nics  = ["nic1", "nic2", "nic3"]  
  tags          = ["xyzyz", "quux"]  
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The unique (within this identity domain) name of the virtual nic set.
- `description` - (Optional) A description of the virtual nic set.
- `applied_acls` - (Optional) A list of the ACLs to apply to the virtual nics in the set.
- `virtual_nics` - (Optional) List of virtual NICs associated with this virtual NIC set.
- `tags` - (Optional) A list of tags to apply to the storage volume.

Import

VNIC Set's can be imported using the `resource name`, e.g.

```
$ terraform import opc_compute_vnic_set.set1 example
```


opc_compute_vpn_endpoint_v2

The `opc_compute_vpn_endpoint_v2` resource creates and manages an VPN Endpoint V2 in an Oracle Cloud Infrastructure Compute Classic identity domain.

Example Usage

```
resource "opc_compute_vnic_set" "vnicset1" {
  name = "vpnaas-vnicset1"
}

resource "opc_compute_ip_network" "ipnetwork1" {
  name           = "ipnetwork1"
  ip_address_prefix = "10.0.12.0/24"
}

resource "opc_compute_vpn_endpoint_v2" "vpnaas1" {
  name                = "vpnaas1"
  customer_vpn_gateway = "${var.vpn_endpoint_public_ip}"
  ip_network          = "${opc_compute_ip_network.ipnetwork1.name}"
  pre_shared_key       = "${var.pre_shared_key}"
  reachable_routes     = ["172.16.4.0/24"]
  vnic_sets            = ["${opc_compute_vnic_set.vnicset1.name}"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the VPN Endpoint V2.
- `customer_vpn_gateway` - (Required) The ip address of the VPN gateway in your data center through which you want to connect to the Oracle Cloud VPN gateway.
- `ip_network` - (Required) The name of the IP network on which the cloud gateway is created by VPNaaS.
- `pre_shared_key` - (Required) The pre-shared VPN key
- `reachable_routes` - (Required) A list of routes (CIDR prefixes) that are reachable through this VPN tunnel.
- `vnic_sets` - (Required) A list of vnic sets that traffics is allowed to and from.
- `description` - (Optional) A description of the VPN Endpoint V2.
- `enabled` - (Optional) Enables or disables the VPN Endpoint V2. Set to true by default.
- `ike_identifier` - (Optional) The Internet Key Exchange (IKE) ID. If you don't specify a value, the default value is the public IP address of the cloud gateway.
- `require_perfect_forward_secretcy` - (Optional) Boolean specifying whether Perfect Forward Secrecy is enabled. Set to true by default.
- `phase_one_settings` - (Optional) Settings for the phase one protocol (IKE). Phase One Settings are detailed below.

- `phase_two_settings` - (Optional) Settings for the phase two protocol (IPSEC). Phase Two Settings are detailed below.
- `tags` - (Optional) List of tags that may be applied to the VPN Endpoint V2.

Phase One Settings support the following:

- `encryption` - (Required) IKE Encryption. `aes128`, `aes192` or `aes256`
- `hash` - (Required) IKE Hash. `sha1`, `sha2_256` or `md5`
- `dh_group` - (Required) Diffie-Hellman group for both IKE and ESP. `group2`, `group5`, `group14`, `group22`, `group23`, or `group24`
- `lifetime` - (Optional) IKE Lifetime in seconds.

Phase Two Settings support the following:

- `encryption` - (Required) ESP Encryption. `aes128`, `aes192` or `aes256`
- `hash` - (Required) ESP Hash. `sha1`, `sha2_256` or `md5`
- `lifetime` - (Optional) IPSEC Lifetime in seconds.

In addition to the above, the following values are exported:

- `local_gateway_ip_address` - Public IP Address of the Local Gateway.
- `local_gateway_private_ip_address` - Private IP Address of the Local Gateway.
- `uri` - The Uniform Resource Identifier for the VPN Endpoint V2.

Import

VPN Endpoint V2's can be imported using the `resource` name, e.g.

```
$ terraform import opc_compute_vpn_endpoint_v2.vpnaas1 /Compute-mydomain/user/example
```

opc_lbaas_certificate

The `opc_lbaas_certificate` resource creates and manages an Load Balancer Classic TLS/SSL Digital Certificate.

Server certificates are used to secure the connection between clients and the load balancers. Trusted certificates are used to secure the connection between the load balancer and the origin servers in the server pool.

Example Usage

```
resource "opc_lbaas_certificate" "example" {
  name          = "example-cert1"
  type          = "SERVER"
  private_key   = "${var.private_key_pem}"
  certificate_body = "${var.cert_pem}"
}
```

Argument Reference

- `name` - (Required) The name of the Certificate.
- `certificate_body` - (Required) The Certificate data in PEM format.
- `type` - (Required) Sets the Certificate Type. TRUSTED or SERVER.
- `certificate_chain` - (Optional) PEM encoded bodies of all certificates in the chain up to and including the CA certificate. This is not need when the certificate is self signed.
- `private_key` - (Optional) The private key data in PEM format. Only required for Server Certificates

Additional Attributes

In addition to the above, the following values are exported:

- `state` - The State of the Digital Certificate resource.
- `uri` - The Uniform Resource Identifier for the Certificate resource.

opc_lbaas_listener

The `opc_lbaas_listener` resource creates and manages a Load Balancer Classic Listener for a Load Balancer Classic instance.

Example Usage

```
resource "opc_lbaas_listener" "listener1" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "listener1"

  port          = 80
  balancer_protocol = "HTTP"
  server_protocol  = "HTTP"
  server_pool     = "${opc_lbaas_server_pool.serverpool1.uri}"
  virtual_hosts   = ["mywebapp.example.com"]
  policies       = ["${opc_lbaas_policy.load_balancing_mechanism_policy.uri}"]
}
```

Argument Reference

- `name` - (Required) The name of the Listener.
- `load_balancer` - (Required) The parent Load Balancer the Listener.
- `balancer_protocol` - (Required) transport protocol that will be accepted for all incoming requests to the selected load balancer listener. HTTP or HTTPS. If set to HTTPS then you must also set the `server_certificates`.
- `port` - (Required) The port on which the Load Balancer is listening.
- `server_protocol` - (Required) The protocol to be used for routing traffic to the origin servers in the server pool. HTTP or HTTPS. If set to HTTPS then you must include a Trusted Certificate Policy in the `policies`
- `enabled` - (Optional) Boolean flag to enable or disable the Listener. Default is `true` (enabled).
- `path_prefixes` - (Optional) List of paths to configure the listener to accept only requests that are targeted to a specific path within the URI of the request.
- `policies` - (Optional) List of the Load Balancer Policy URIs to apply to the listener.
- `server_pool` - (Optional) URI of the Server Pool resource to which the load balancer distributes requests.
- `certificates` - (Optional) The URI of the server security certificate.
- `tags` - (Optional) List of tags.
- `virtual_hosts` - (Optional) Configure the listener to only accept URI requests that include the host names listed in this field.

Additional Attributes

In addition to the above, the following values are exported:

- `operational_details` - Description of the operational state.
- `parent_listener` - The parent Listener.
- `state` - State of the Origin Server Pool.
- `uri` - The Uniform Resource Identifier for the Listener.

Import

Listeners can be imported using the a combinable of the resource region, load balancer name and policy name and in the format `region/loadbalancer/name`

```
$ terraform import opc_lbaas_listener.listener1 uscom-central-1/lb1/example-listener1
```

opc_lbaas_load_balancer

The `opc_lbaas_load_balancer` resource creates and manages a Load Balancer Classic instance in an Oracle Cloud Infrastructure Classic Compute region. You must define server pools, create at least one listener, and optionally define the policies for the load balancer before it will be operational.

Example Usage

```
resource "opc_lbaas_load_balancer" "lb1" {
  name          = "example-lb1"
  region        = "uscom-central-1"
  description    = "My Example Load Balancer"
  scheme        = "INTERNET_FACING"

  permitted_methods = ["GET", "HEAD", "POST", "PUT"]
}
```

Argument Reference

- `name` - (Required) The name of the Load Balancer.
- `region` - (Required) The region in which to create the Load Balancer, e.g. `uscom-central-1`
- `scheme` - (Required) Set to either `INTERNET_FACING` or `INTERNAL`
 - `INTERNET_FACING` - Create an internet-facing load balancer in a given IP network.
 - `INTERNAL` - Create an internal load balancer in a given IP network for sole consumption of other clients inside the same network.
- `description` - (Optional) A short description for the load balancer. The description must not exceed 1000 characters.
- `enabled` - (Optional) Boolean flag to enable or disable the Load Balancer. Default is `true` (enabled).
- `ip_network` - (Optional) Fully qualified three part name of the IP network to be associated with the load balancer.
- `parent_load_balancer` - (Optional) Select a parent load balancer if you want to create a dependent load balancer.
- `permitted_clients` - (Optional) List of permitted client IP addresses or CIDR ranges which can connect to this load balancer on the configured Listener ports. If not set all connections are permitted.
- `permitted_methods` - (Optional) List of permitted HTTP methods. e.g. `GET`, `POST`, `PUT`, `PATCH`, `DELETE`, `HEAD` or you can also create your own custom methods. Requests with methods not listed in this field will result in a 403 (unauthorized access) response.
- `tags` - (Optional) List of tags.

Additional Attributes

In addition to the above, the following values are exported:

- `operational_details` - Description of the operational state.
- `state` - State of the Origin Server Pool.
- `status` - Status of the Origin Server Pool.
- `uri` - The Uniform Resource Identifier for the Load Balancer.

Import

Load Balancers can be imported using the a combinable of the resource `region` and name in the format `region/name`

```
$ terraform import opc_lbaas_load_balancer.lb1 uscom-central-1/example-lb1
```

opc_lbaas_policy

The `opc_lbaas_policy` resource creates and manages a Load Balancer Classic Policy for a Load Balancer Classic instance.

The Policy resource supports the definition of distinct Policy types:

- Application Cookie Stickiness Policy
- CloudGate Policy
- Load Balancer Cookie Stickiness Policy
- Load Balancing Mechanism Policy
- Rate Limiting Request Policy
- Redirect Policy
- Resource Access Control Policy
- Set Request Header Policy
- SSL Negotiation Policy
- Trusted Certificate Policy

Example Usage

```
resource "opc_lbaas_policy" "load_balancing_mechanism_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_load_balancing_mechanism_policy"

  load_balancing_mechanism_policy {
    load_balancing_mechanism = "round_robin"
  }
}
```

Argument Reference

- `name` - (Required) The name of the Listener.
- `load_balancer` - (Required) The parent Load Balancer the Listener.

Include one, and only one, of:

- `application_cookie_stickiness_policy` - see Application Cookie Stickiness Policy
- `cloudgate_policy` - see CloudGate Policy
- `load_balancer_cookie_stickiness_policy` - see Load Balancer Cookie Stickiness Policy
- `load_balancing_mechanism_policy` - see Load Balancing Mechanism Policy
- `rate_limiting_request_policy` - see Rate Limiting Request Policy

- `redirect_policy` - see Redirect Policy
- `resource_access_control_policy` - see Resource Access Control Policy
- `set_request_header_policy` - see Set Request Header Policy
- `ssl_negotiation_policy` - see SSL Negotiation Policy
- `trusted_certificate_policy` - see Trusted Certificate Policy

Application Cookie Stickiness Policy

Enable session stickiness (session affinity) for any request based on a given cookie name specified in the policy.

Example

```
resource "opc_lbaas_policy" "application_cookie_stickiness_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_application_cookie_stickiness_policy"

  application_cookie_stickiness_policy {
    cookie_name = "MY_APP_COOKIE"
  }
}
```

Attributes

- `cookie_name` - (Required) Name of the application cookie used to control how long the load balancer will continue to route requests to the same origin server.

CloudGate Policy

protect resources/applications with the help of CloudGate module available in Load Balancer. These headers will enable CloudGate to lookup for the application and the policy present under the appropriate IDCS Tenant containing information for the protection mechanism to be enforced.

Example

```
resource "opc_lbaas_policy" "cloudgate_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_cloudgate_policy"

  cloudgate_policy {
    virtual_hostname_for_policy_attribution = "host1.example.com"
  }
}
```

Attributes

- `virtual_hostname_for_policy_attribution` - (Required) Host name needed by CloudGate to enforce OAuth policies.

Load Balancer Cookie Stickiness Policy

Enables session stickiness (session affinity) for all requests for a given period of time specified in the policy.

Example

```
resource "opc_lbaas_policy" "load_balancer_cookie_stickiness_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_load_balancer_cookie_stickiness_policy"

  load_balancer_cookie_stickiness_policy {
    cookie_expiration_period = 60
  }
}
```

Attributes

- `cookie_expiration_period` - (Required) The time period, in seconds, after which the cookie should be considered stale. If the value is zero or negative the stickiness session lasts for the duration of the browser session.

Load Balancing Mechanism Policy

Specify a load balancing mechanism for distributing client requests across multiple origin servers

Example

```
resource "opc_lbaas_policy" "load_balancing_mechanism_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_load_balancing_mechanism_policy"

  load_balancing_mechanism_policy {
    load_balancing_mechanism = "round_robin"
  }
}
```

Attributes

- `load_balancing_mechanism` - (Required) Supported options are `round_robin`, `least_conn`, and `ip_hash`.

Rate Limiting Request Policy

Limits the number of requests that can be processed per second by the load balancer.

Example

```
resource "opc_lbaas_policy" "rate_limiting_request_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_rate_limiting_request_policy"

  rate_limiting_request_policy {
    requests_per_second      = 1
    burst_size               = 10
    delay_excessive_requests = true
    zone                     = "examplezone"
  }
}
```

Attributes

- `burst_size` - (Required) The number of requests that can be delayed until it exceeds the maximum number specified as burst size in which case the request is terminated.
- `delay_excessive_requests` - (Required) delay excessive requests while requests are being limited.
- `requests_per_second` - (Required) Maximum number of requests per second.
- `zone` - (Required) Name of the shared memory zone.
- `http_error_code` - (Optional) Status code to return in response to rejected requests. You can specify any status code between 405 to 599. Default is 503
- `logging_level` - (Optional) Logging level for cases when the server refuses to process requests due to rate exceeding, or delays request processing. Can be one of info, notice, warn, or error. Default is warn
- `rate_limiting_criteria` - (Optional) Criteria based on which requests will be throttled. Default is server
 - `server` - limit the requests processed by the virtual server
 - `remote_address` - limit the processing rate of requests coming from a single IP address.
 - `host` - limit the processing rate of requests coming from a host.
- `zone_memory_size` - (Optional) Size of the shared memory occupied by the zone. Default is 10

Redirect Policy

Redirects all requests to this load balancer to a specific URI.

Example

```
resource "opc_lbaas_policy" "redirect_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_redirect_policy"

  redirect_policy {
    redirect_uri = "https://redirect.example.com"
    response_code = 306
  }
}
```

Attributes

- `redirect_uri` - (Required) redirected requests to the specified URI.
- `response_code` - (Required) The exact 3xx response code to use when redirecting

Resource Access Control Policy

Controls what clients have access to the load balancer, based on the IP address or the CIDR range of the incoming request.

Example

```
resource "opc_lbaas_policy" "resource_access_control_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_resource_access_control_policy"

  resource_access_control_policy {
    disposition = "DENY_ALL"
    permitted_clients = ["10.0.0.0/16"]
  }
}
```

Attributes

- `disposition` - (Required) Default policy. `DENY_ALL` or `PERMIT_ALL`.
- `denied_clients` - (Optional) List of IP address or CIDR ranges identifying clients from which requests must be accepted by the Load Balancer
- `permitted_clients` - (Optional) IP address or CIDR ranges identifying clients from which requests must be denied by the Load Balancer.

Set Request Header Policy

Inserts additional information into the standard HTTP headers of requests forwarded to a server pool.

Example

```
resource "opc_lbaas_policy" "set_request_header_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_set_request_header_policy"

  set_request_header_policy {
    header_name      = "X-Custom-Header"
    value            = "foo-bar"
    action_when_header_exists = "OVERWRITE"
    action_when_header_value_is = ["bar", "foo"]
  }
}
```

Attributes

- `header_name` - (Required)
- `action_when_header_exists` - (Required) action to be taken when a header exists in the request. Options: NOOP, PREPEND, APPEND, OVERWRITE, CLEAR
- `action_when_header_value_is` - (Optional) List if header values. Action is taken only when the header exists in the request and the header value matches one of the values provided.
- `action_when_header_value_is_not` - (Optional) List if header values. Action is taken only when the header exists in the request and the header value does not match any of the values provided.

SSL Negotiation Policy

Define specific SSL protocols, ciphers, and server order preference for the secure connection

Example

```
resource "opc_lbaas_policy" "ssl_negotiation_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_ssl_negitiation_policy"

  ssl_negotiation_policy {
    port = 8022
    server_order_preference = "ENABLED"
    ssl_protocol = ["SSLv3", "TLSv1.2"]
    ssl_ciphers = ["AES256-SHA"]
  }
}
```

Attributes

- `ssl_protocol` - (Required) Security protocols supported for incoming secure client connections to the associated listener. Supported options are SSLv2, SSLv3, TLSv1, TLSv1.1, TLSv1.2
- `port` - (Optional) The load balancer port for the the SSL protocols and the SSL ciphers.

- `server_order_preference` - (Optional) enable or disable the server order preference for secure connections to associated Listener. ENABLED or DISABLED.
- `ssl_ciphers` - (Optional) List of SSL ciphers supported for incoming secure client connections to the associated Listener.

Trusted Certificate Policy

Identifies a trusted certificate, which the load balancer uses when making a secure connection to the compute instances in the server pool. If you are configuring a secure connection (HTTPS or SSL) between the load balancer and the origin servers, you must add this policy to the load balancer.

Example

```
resource "opc_lbaas_policy" "trusted_certificate_policy" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "example_trusted_certificate_policy"

  trusted_certificate_policy {
    trusted_certificate = "${opc_lbaas_ssl_certificate.cert1.uri}"
  }
}
```

Attributes

- `trusted_certificate` - (Required) URI of the SSL Certificate

Additional Attributes

In addition to the above, the following values are exported:

- `state` - State of the Policy.
- `type` - The Type of the Policy.
- `uri` - The Uniform Resource Identifier for the Policy.

Import

Policies can be imported using the a combinable of the resource region, load balancer name and policy name and in the format `region/loadbalancer/name`

```
$ terraform import opc_lbaas_policy.policy1 uscom-central-1/lb1/example-policy1
```

opc_lbaas_server_pool

The `opc_lbaas_server_pool` resource creates and manages a Load Balancer Classic Origin Server Pool for a Load Balancer Classic instance. The Server Pool defines one or more servers (referred to as origin servers) to which the load balancer can distribute requests.

Example Usage

```
resource "opc_lbaas_server_pool" "serverpool1" {
  load_balancer = "${opc_lbaas_load_balancer.lb1.id}"
  name          = "serverpool1"

  servers = ["129.150.169.179:8080", "129.150.170.162:8080"]

  health_check {
    type = "http"
    path = "/healthcheck"
  }
}
```

Argument Reference

- `name` - (Required) The name of the Server Pool.
- `load_balancer` - (Required) The parent Load Balancer the Origin Server Pool.
- `servers` - (Required) List of servers in the Server Pool. To define the server in the server pool, provide IP address or DNS name of the compute instances, and port for load balancer to direct traffic to, in the format `host:port`
- `enabled` - (Optional) Boolean flag to enable or disable the Server Pool. Default is `true` (enabled).
- `health_check` - (Optional) Enables Load Balancer health check, see Health Check Attributes
- `tags` - (Optional) List of tags.
- `vnic_set` - (Optional) Fully qualified three part name of a vNICSet to be associated with the server pool vNIC. Load Balancer uses this vNICSet to set the right ACLs to allow egress traffic from the load balancer.

Health Check Attributes

The load balancer can perform regular health checks of the origin servers and route inbound traffic to the healthy origin servers.

- `type` - (Optional) Health check mechanism to use to test the origin servers. Default is `http`
 - `http` - sends an HTTP HEAD to the set `path` and checks response is in one of the `accepted_return_codes`
- `path` - (Optional) The path to check. Set the `/'` the check all paths. Default is `/'`
- `accepted_return_codes` - (Optional) List of HTTP response status codes that indicate the origin server is healthy. Accepted return codes can be one or more of the `2xx`, `3xx`, `4xx`, or `5xx` codes. Default is `["2xx", "3xx"]`.

- `enabled` - (Optional) Boolean flag to enable or disable the Health Checks. Default is `true` (enabled).
- `interval` - (Optional) The approximate interval, in seconds, that the load balancer will wait before sending the target request to each origin server, in the range 5 to 300 seconds. Default is 30.
- `timeout` - (Optional) The amount of time, in seconds, that the load balancer will wait without a response before identifying the origin server as unavailable. The timeout value must be less than the `interval` value and it should range between 2 to 60. Default is 20.
- `healthy_threshold` - (Optional) The number of consecutive successful health checks required before moving the origin server to the healthy state. The value of healthy threshold ranges from 2 to 10. Default is 5.
- `unhealthy_threshold` - (Optional) The number of consecutive health check failures required before moving the origin server to the unhealthy state. The value of unhealthy threshold ranges from 2 to 10. Default is 7.

Additional Attributes

In addition to the above, the following values are exported:

- `operational_details` - Description of the operational state.
- `state` - State of the Origin Server Pool.
- `status` - Status of the Origin Server Pool.
- `uri` - The Uniform Resource Identifier for the Server Pool.

Import

Origin Server Pools can be imported using the combination of the resource region, load balancer name, and policy name and in the format `region/loadbalancer/name`

```
$ terraform import opc_lbaas_server_pool.serverpool1 uscom-central-1/lb1/example-serverpool1
```


opc_storage_container

Creates and manages a Container in the Oracle Cloud Infrastructure Storage Classic service. `storage_endpoint` must be set in the provider or environment to manage these resources.

Example Usage

```
resource "opc_storage_container" "default" {
  name = "storage-container-1"
  read_acls = [ ".r:example.com", ".r:listings" ]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Storage Container.
- `read_acls` - (Optional) The list of ACLs that grant read access. See Setting Container ACLs.
- `write_acls` - (Optional) The list of ACLs that grant write access. See Setting Container ACLs.
- `primary_key` - (Optional) The primary secret key value for temporary URLs.
- `secondary_key` - (Optional) The secondary secret key value for temporary URLs.
- `allowed_origins` - (Optional) List of origins that are allowed to make cross-origin requests.
- `exposed_headers` - (Optional) List of headers exposed to the user agent (e.g. browser) in the actual request response
- `max_age` - (Optional) Maximum age in seconds for the origin to hold the preflight results.
- `quota_bytes` - (Optional) Maximum size of the container, in bytes
- `quota_count` - (Optional) Maximum object count of the container
- `metadata` - (Optional) Additional object metadata headers. See Container Metadata below for more information.

Setting Container ACLs

The `read_acl` consists of a list of **roles** or **referrer designations**. The `write_acls` consists of a list of **roles**.

- **roles** can be built-in roles or custom roles. Custom roles are defined in the Users tab in the Oracle Cloud My Services console. For a role that was provisioned as part of another service instance, the format is `domainName.serviceName.roleName`. For a custom role, the format is `domainName.roleName`. Default Storage roles include:
 - `${var.domain}.Storage.Storage_ReadOnlyGroup`
 - `${var.domain}.Storage.Storage_ReadWriteGroup`

- `${var.domain}.Storage.Storage_Administrator`

- **referrer designation** indicates the host (or hosts) for which read access to the container should be allowed or denied. When the server receives a request for the container, it compares the referrer designations specified in the Read ACL with the value of the Referer header in the request, and determines whether access should be allowed or denied. The syntax of the referrer designation is: `.r:value`

`value` indicates the host for which access to the container should be allowed. It can be a specific host name (example: `.r:www.example.com`), a domain (example: `.r:.example.com`), or an asterisk (`.r:*`) to indicate all hosts. Note that if `.r:*` is specified, objects in the container will be publicly readable without authentication.

A minus sign (-) before `value` (example: `.r:-temp.example.com`) indicates that the host specified in the `value` field must be denied access to the container.

By default, read access to a container does not include permission to list the objects in the container. To allow listing of objects as well, include the `.rlistings` directive in the ACL.

Container Metadata

The `metadata` config defines a map of additional meta data header name value pairs. The additional meta data items are set as HTTP Headers on the container in the form `X-Container-Meta-{name}: {value}`, where `{name}` is the name of the metadata item `{value}` is the header content. For example:

```
metadata {
  "Foo-Bar" = "barfoo"
  "Tags" = "abc 123 xyz"
}
```

Import

Container's can be imported using the `resource name`, e.g.

```
$ terraform import opc_storage_container.default example
```

opc_storage_object

Creates and manages a Object in an Oracle Cloud Infrastructure Storage Classic container. `storage_endpoint` must be set in the provider or environment to manage these resources.

Example Usage

```
resource "opc_storage_object" "default" {
  name      = "storage-object-1"
  container = "${opc_storage_container.container.name}"
  file      = "${"./source_file.txt"}"
  etag      = "${md5(file("./source_file.txt"))}"
  content_type = "text/plain;charset=utf-8"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Storage Object.
- `container` - (Required) The name of Storage Container the store the object in.

Note: One of `content`, `file`, or `copy_from` must be specified

- `content` - (Optional) Raw content in string-form of the data.
- `file` - (Optional) File path for the content to use for data.
- `copy_from` - (Optional) name of an existing object used to create the new object as a copy. The value is in form `container/object`. You must UTF-8-encode and then URL-encode the names of the container and object.
- `content_disposition` - (Optional) Set the HTTP Content-Disposition header to specify the override behaviour for the browser, e.g. `inline` or `attachment`.
- `content_encoding` - (Optional) set the HTTP Content-Encoding for the object.
- `content_type` - (Optional) set the MIME type for the object.
- `delete_at` - (Optional) The date and time in UNIX Epoch time stamp format when the system removes the object.
- `etag` - (Optional) MD5 checksum value of the request body. Strongly Recommended.
- `transfer_encoding` - (Optional) Set to `chunked` to enable chunked transfer encoding.
- `metadata` - (Optional) Additional object metadata headers. See [Object Metadata](#) below for more information.

Attributes

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

- `id` - The combined container and object name path of the object.
- `accept_ranges` - Type of ranges that the object accepts.
- `content_length` - Length of the object in bytes.
- `last_modified` - Date and Time that the object was created/modified in ISO 8601.
- `object_manifest` - The dynamic large-object manifest object.
- `timestamp` - Date and Time in UNIX EPOCH when the account, container, or object was initially created at the current version.
- `transaction_id` - Transaction ID of the request.

Object Metadata

The metadata config defines a map of additional meta data header name value pairs. The additional meta data items are set HTTP Headers on the object in the form `X-Object-Meta-{name}: {value}`, where `{name}` is the name of the metadata item `{value}` is the header content. For example:

```
metadata {  
  "Foo-Bar" = "barfoo",  
  "Sha256" = "e91ed4f93637379a7539cb5d8d0b5bca3972755de4f9371ab2e123e7b4c53680"  
}
```

Import

Object's can be imported using the `resource id`, e.g.

```
$ terraform import opc_storage_object.default container/example
```

Please note though, importing a Storage Object does *not* allow a user to modify the content, or attributes for the Storage Object. It is, however, possible to import a Storage Object, and replace the object with new content, or a copy of another Storage Object. It is also possible to import a Storage Object into Terraform in order to delete the object.