» triton_account

The `triton_account` data source queries Triton for Account information.

NOTE: This data source uses the name of the Account currently configured in the Trition provider.

» Example Usages

Find current Account unique identifier:

```
# Declare the data source.
data "triton_account" "main" {}

# Access unique Account ID using output from the data source.
output "account_id" {
  value = "${data.triton_account.main.id}"
}
```

» Argument Reference

There are no arguments available for this data source.

» Attribute Reference

The following attributes are supported:

- `id` - (string) The unique identifier representing the Account in Triton.
- `login` - (string) The login name associated with the Account.
- `email` - (string) An e-mail address that is current set in the Account.
- `cns_enabled` - (boolean) Whether the Container Name Service (CNS) is enabled for the Account.

» triton_datacenter

The `triton_datacenter` data source queries Triton for Data Center information.

NOTE: This data source uses the endpoint URL of the Data Center currently configured in the Trition provider.

» Example Usages

Find current Data Center endpoint URL:

```
# Declare the data source.
data "triton_datacenter" "current" {}

# Access current endpoint URL using output from the data source.
output "endpoint" {
  value = "${data.triton_datacenter.current.endpoint}"
}
```

» Argument Reference

There are no arguments available for this data source.

» Attribute Reference

NOTE: When using the Triton Public Cloud, the `endpoint` attribute might include an old, but still fully supported, domain name "joyentcloud.com" (e.g. https://us-east-1.api.joyentcloud.com), even when the new domain name "joyent.com" has been used to configure the cloud endpoint URL in the Trition provider.

The following attributes are supported:

- `name` - (string) The name of the Data Center.
- `endpoint` - (string) The endpoint URL of the Data Center.

» triton_image

The `triton_image` data source queries the Triton Image API for an image ID based on a variety of different parameters.

» Example Usages

Find the ID of a Base 64 LTS image.

```
data "triton_image" "base" {
  name    = "base-64-lts"
  version = "16.4.1"
}
```

```
output "image_id" {
  value = "${data.triton_image.base.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (string) The name of the image
- `os` - (string) The underlying operating system for the image
- `version` - (string) The version for the image
- `public` - (boolean) Whether to return public as well as private images
- `state` - (string) The state of the image. By default, only `active` images are shown. Must be one of: `active`, `unactivated`, `disabled`, `creating`, `failed` or `all`, though the default is sufficient in almost every case.
- `owner` - (string) The UUID of the account which owns the image
- `type` - (string) The image type. Must be one of: `zone-dataset`, `lx-dataset`, `zvol`, `docker` or `other`.
- `most_recent` - (bool) If more than one result is returned, use the most recent Image.

» Attribute Reference

The following attributes are exported:

- `id` - (string) - The identifier representing the image in Triton.

» triton_network

The `triton_network` data source queries Triton for Network information (e.g., Network ID, etc.) based on the name of the Network.

» Example Usages

Find the Network ID of the `Joyent-SDC-Private` network.

```
# Declare the data source.
data "triton_network" "private" {
  name = "Joyent-SDC-Private"
```

```
}

# Access unique Network ID using output from the data source.
output "private_network_id" {
  value = "${data.triton_network.private.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (string) **Required.** The name of the Network.

» Attribute Reference

The following attributes are supported:

- `id` - (string) The unique identifier of the Network.
- `public` - (boolean) Whether this Network is a public or private RFC1918 network.
- `fabric` - (boolean) Whether this Network is created on a Fabric.

» triton_fabric_vlan

The `triton_fabric_vlan` data source queries Triton for Fabric VLAN information (e.g., VLAN ID, etc.) based either on the name, VLAN ID or description of the Fabric VLAN.

» Example Usages

Find the VLAN ID using the name of the Fabric VLAN as a search filter:

```
# Declare the data source.
data "triton_fabric_vlan" "public" {
  name = "Public-VLAN-Production"
}

# Access unique VLAN ID using output from the data source.
output "public_vlan_id" {
  value = "${data.triton_fabric_vlan.public.vlan_id}"
}
```

Find the VLAN ID using name (with a wildcard match) and description of the Fabric VLAN as a search filters:

```
# Declare the data source, and use a combination of two arguments
# to form a search filter. Use a wildcard match for the name.
data "triton_fabric_vlan" "private_database_vlan" {
  name        = "Private-VLAN-*"
  description = "A secure VLAN for production database servers"
}

# Access unique VLAN ID using output from the data source.
output "private_database_vlan_id" {
  value = "${data.triton_fabric_vlan.private_database_vlan.vlan_id}"
}
```

» Argument Reference

NOTE: The arguments of this data source act as filters when searching for a matching Fabric VLAN and can be combined together, but at lease one of `name`, `vlan_id` or `description` must be assigned.

The following arguments are supported:

- `name` - (string) Optional. The name of the Fabric VLAN.
- `vlan_id` - (integer) Optional. The unique identifier (VLAN ID) of the Fabric VLAN.
- `description` - (string) Optional. The description of the Fabric VLAN.

NOTE: Both the `name` and `description` arguments support a simple wildcard pattern matching using two common wildcards, such as `*` (asterisk) and `?`. There is no support for either ranges or character classes. More details about wildcard patterm matching can be found here.

» Attribute Reference

The following attributes are exported:

- `name` - (string) The name of the Fabric VLAN, if any.
- `vlan_id` - (integer) The unique identifier (VLAN ID) of the Fabric VLAN.
- `description` - (string) The description of the Fabric VLAN, if any.

» triton_fabric_network

The `triton_fabric_network` data source queries Triton for Fabric Network information (e.g., subnet CIDR, gateway, state routes, etc.) based on the name of the Fabric Network and ID of the VLAN on which the network has been created.

» Example Usages

Find the subnet CIDR of a Fabric Network:

```
# Declare the data source to retrieve Fabric VLAN details.
data "triton_fabric_vlan" "private" {
  name = "Private-VLAN-Production"
}

# Declare the data source to retrieve Fabric Network details.
data "triton_fabric_network" "private" {
  name     = "Private-Network-Production"
  vland_id = "${data.triton_fabric_vlan.private.vlan_id}"
}

# Access subnet CIDR using output from the data source.
output "private_network_cidr" {
  value = "${data.triton_fabric_network.private.subnet}"
}
```

» Argument Reference

NOTE: You can use the triton_fabric_vlan data source to retrieve details about a Fabric VLAN for reference.

The following arguments are supported:

- `name` - (string) **Required.** The name of the Fabric Network.
- `vlan_id` - (integer) **Required.** The unique identifier (VLAN ID) of the Fabric VLAN.

» Attribute Reference

- `name` - (string) The name of the Fabric Network.
- `public` - (boolean) Whether this Fabric Network is a public or private RFC1918 network.

- `fabric` - (boolean) Whether this network is created on a Fabric. This is always **true** for a Fabric Network.
- `description` - (string) The description of the Fabric Network, if any.
- `subnet` - (string) A CIDR block used for the Fabric Network.
- `provision_start_ip` - (string) The first IP address on this network that may be assigned.
- `provision_end_ip` - (string) The last IP address on this network that may be assigned.
- `gateway` - (string) An IP address of the gateway on this network, if any.
- `resolvers` - (list) A list of IP addresses of DNS resolvers on this network.
- `routes` - (map) A map of static routes (using the CIDR notation) and corresponding gateways on this network, if any.
- `internet_nat` - (boolean) Whether the gateway on this network is also provisioned with the Internet NAT zone.
- `vlan_id` - (integer) The unique identifier (VLAN ID) of the Fabric VLAN.

» triton_fabric

The `triton_fabric` resource represents an fabric for a Triton account. The fabric is a logical set of interconnected switches.

» Example Usages

» Create a fabric

```
resource "triton_fabric" "dmz" {
  vlan_id            = 100
  name               = "dmz"
  description        = "DMZ Network"
  subnet             = "10.60.1.0/24"
  provision_start_ip = "10.60.1.10"
  provision_end_ip   = "10.60.1.240"
  gateway            = "10.60.1.1"
  resolvers          = ["8.8.8.8", "8.8.4.4"]

  internet_nat = true
}
```

» Argument Reference

The following arguments are supported:

- `name` - (String, Required, Change forces new resource) Network name.
- `description` - (String, Optional, Change forces new resource) Optional description of network.
- `subnet` - (String, Required, Change forces new resource) CIDR formatted string describing network.
- `provision_start_ip` - (String, Required, Change forces new resource) First IP on the network that can be assigned.
- `provision_end_ip` - (String, Required, Change forces new resource) Last assignable IP on the network.
- `gateway` - (String, Optional, Change forces new resource) Optional gateway IP.
- `resolvers` - (List, Optional) Array of IP addresses for resolvers.
- `routes` - (Map, Optional, Change forces new resource) Map of CIDR block to Gateway IP address.
- `internet_nat` - (Bool, Optional, Change forces new resource) If a NAT zone is provisioned at Gateway IP address. Default is `true`.
- `vlan_id` - (Int, Required, Change forces new resource) VLAN id the network is on. Number between 0-4095 indicating VLAN ID.

» Attribute Reference

The following attributes are exported:

- `id` - (string) - The identifier representing the network in Triton.
- `name` - (String) - Network name.
- `public` - (Bool) - Whether or not this is an RFC1918 network.
- `fabric` - (Bool) - Whether or not this network is on a fabric.
- `description` - (String) - Optional description of network.
- `subnet` - (String) - CIDR formatted string describing network.
- `provision_start_ip` - (String) - First IP on the network that can be assigned.
- `provision_end_ip` - (String) - Last assignable IP on the network.
- `gateway` - (String) - Optional gateway IP.
- `resolvers` - (List) - Array of IP addresses for resolvers.
- `routes` - (Map) - Map of CIDR block to Gateway IP address.
- `internet_nat` - (Bool) - If a NAT zone is provisioned at Gateway IP address.

- `vlan_id` - (Int) - VLAN id the network is on. Number between 0-4095 indicating VLAN ID.

» triton_firewall_rule

The `triton_firewall_rule` resource represents a rule for the Triton cloud firewall.

» Example Usages

» Allow web traffic on ports tcp/80 and tcp/443 to machines with the 'www' tag from any source

```
resource "triton_firewall_rule" "www" {
  description = "Allow web traffic on ports tcp/80 and tcp/443 to machines with the 'www' ta
  rule        = "FROM any TO tag \"www\" ALLOW tcp (PORT 80 AND PORT 443)"
  enabled     = true
}
```

» Allow ssh traffic on port tcp/22 to all machines from known remote IPs

```
resource "triton_firewall_rule" "22" {
  description = "Allow ssh traffic on port tcp/22 to all machines from known remote IPs."
  rule        = "FROM (ip w.x.y.z OR ip w.x.y.z) TO all vms ALLOW tcp PORT 22"
  enabled     = true
}
```

» Block IMAP traffic on port tcp/143 to all machines

```
resource "triton_firewall_rule" "imap" {
  description = "Block IMAP traffic on port tcp/143 to all machines."
  rule        = "FROM any TO all vms BLOCK tcp PORT 143"
  enabled     = true
}
```

» Argument Reference

The following arguments are supported:

- `rule` - (string, Required) The firewall rule described using the Cloud API rule syntax defined at https://docs.joyent.com/public-cloud/network/firewall/cloud-firewall-rules-reference. Note: Cloud API will normalize rules based on case-sensitivity, parentheses, ordering of IP addresses, etc. This can result in Terraform updating rules repeatedly if the rule definition differs from the normalized value.
- `enabled` - (boolean, Optional) Default: `false` Whether the rule should be effective.
- `description` - (string, Optional) Description of the firewall rule

» Attribute Reference

The following attributes are exported:

- `id` - (string) - The identifier representing the firewall rule in Triton.

» triton_instance_template

The `triton_instance_template` resource represents a Triton Service Group instance template.

NOTE: Triton Service Groups are in Preview and only supported in specific regions at this time. They will become Generally Available in the near future.

» Example Usages

```
data "triton_image" "base" {
  name    = "base-64-lts"
  version = "16.4.1"
}

data "triton_network" "private" {
  name = "Joyent-SDC-Private"
}

resource "triton_instance_template" "base" {
  template_name    = "Base template"
  image            = "${data.triton_image.base.id}"
  package          = "g4-highcpu-128M"

  firewall_enabled = false
```

```
  networks         = ["${data.triton_network.private.id}"]

  tags {
    hello = "world"
    role  = "database"
  }
}
```

» Argument Reference

The following arguments are supported:

- `template_name` - (string, Required) Friendly name for the instance template.
- `image` - (string, Required) UUID of the image.
- `package` - (string, Required) Package name used for provisioning.
- `firewall_enabled` - (boolean, Optional) Whether to enable the firewall for group instances. Default is `false`.
- `tags` - (map, Optional) Tags for group instances.
- `networks` - (list, Optional) Network IDs for group instances.
- `metadata` - (map, Optional) Metadata for group instances.
- `userdata` - (string, Optional) Data copied to instance on boot.

» Attribute Reference

The following attributes are exported:

- `id` - (string) - The identifier representing the Triton Service Group instance template.

» triton_key

The `triton_key` resource represents an SSH key for a Triton account.

» Example Usages

Create a key

```
resource "triton_key" "example" {
  name = "Example Key"
  key  = "${file("keys/id_rsa")}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (string, Change forces new resource) The name of the key. If this is left empty, the name is inferred from the comment in the SSH key material.
- `key` - (string, Required, Change forces new resource) The SSH key material. In order to read this from a file, use the `file` interpolation.

» triton_machine

The `triton_machine` resource represents a virtual machine or infrastructure container running in Triton.

Note: Starting with Triton 0.2.0, Please note that when you want to specify the networks that you want the machine to be attached to, use the `networks` parameter and not the `nic` parameter.

» Example Usages

» Run a SmartOS base-64 machine.

```
resource "triton_machine" "test-smartos" {
  name    = "test-smartos"
  package = "g3-standard-0.25-smartos"
  image   = "842e6fa6-6e9b-11e5-8402-1b490459e334"

  tags {
    hello = "world"
    role  = "database"
  }

  cns {
    services = ["web", "frontend"]
  }

  metadata {
    hello = "again"
```

```
  }
}
```

» Attaching a Machine to Joyent public network

```
data "triton_image" "image" {
  name    = "base-64-lts"
  version = "16.4.1"
}

data "triton_network" "public" {
  name = "Joyent-SDC-Public"
}

resource "triton_machine" "test" {
  package  = "g4-highcpu-128M"
  image    = "${data.triton_image.image.id}"
  networks = ["${data.triton_network.public.id}"]
}
```

» Run an Ubuntu 14.04 LTS machine.

```
resource "triton_machine" "test-ubuntu" {
  name                 = "test-ubuntu"
  package              = "g4-general-4G"
  image                = "1996a1d6-c0d9-11e6-8b80-4772e39dc920"
  firewall_enabled     = true
  root_authorized_keys = "Example Key"
  user_script          = "#!/bin/bash\necho 'testing user-script' >> /tmp/test.out\nhostname

  tags {
    purpose = "testing ubuntu"
  }
}
```

» Run two SmartOS machine's with placement rules.

```
resource "triton_machine" "test-db" {
  name    = "test-db"
  package = "g4-highcpu-8G"
  image   = "842e6fa6-6e9b-11e5-8402-1b490459e334"

  affinity = ["role!=~web"]
```

```
  tags {
    role = "database"
  }
}

resource "triton_machine" "test-web" {
  name    = "test-web"
  package = "g4-highcpu-8G"
  image   = "842e6fa6-6e9b-11e5-8402-1b490459e334"

  tags {
    role = "web"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (string) The friendly name for the machine. Triton will generate a name if one is not specified.
- `tags` - (map) A mapping of tags to apply to the machine.
- `cns` - (map of CNS attributes, Optional) A mapping of CNS attributes to apply to the machine.
- `metadata` - (map, optional) A mapping of metadata to apply to the machine.
- `package` - (string, Required) The name of the package to use for provisioning.
- `image` - (string, Required) The UUID of the image to provision.
- `networks` - (list, optional) The list of networks to associate with the machine. The network ID will be in hex form, e.g `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`.
- `affinity` - (list of Affinity rules, Optional) A list of valid Affinity Rules to apply to the machine which assist in data center placement. Using this attribute will force resource creation to be serial. NOTE: Affinity rules are best guess and assist in placing instances across a data center. They're used at creation and not referenced after.
- `(Deprecated) locality` - (map of Locality hints, Optional) A mapping of Locality attributes to apply to the machine that assist in data center placement. NOTE: Locality hints are only used at the time of machine

creation and not referenced after. Locality is deprecated as of CloudAPI v8.3.0.

- `firewall_enabled` - (boolean) Default: `false` Whether the cloud firewall should be enabled for this machine.
- `root_authorized_keys` - (string) The public keys authorized for root access via SSH to the machine.
- `user_data` - (string) Data to be copied to the machine on boot. **NOTE:** The content of `user_data` will *not be executed* on boot. The data will only be written to the file on each boot before the content of the script from `user_script` is to be run.
- `user_script` - (string) The user script to run on boot (every boot on SmartMachines). To learn more about both the user script and user data see the metadata API documentation and the Joyent Metadata Data Dictionary specification.
- `administrator_pw` - (string) The initial password for the Administrator user. Only used for Windows virtual machines.
- `cloud_config` - (string) Cloud-init configuration for Linux brand machines, used instead of `user_data`.
- `deletion_protection_enabled` - (bool) Whether an instance is destroyable. Default is `false`.

» Attribute Reference

The following attributes are exported:

- `id` - (string) - The identifier representing the machine in Triton.
- `type` - (string) - The type of the machine (`smartmachine` or `virtualmachine`).
- `state` - (string) - The current state of the machine.
- `dataset` - (string) - The dataset URN with which the machine was provisioned.
- `memory` - (int) - The amount of memory the machine has (in Mb).
- `disk` - (int) - The amount of disk the machine has (in Gb).
- `ips` - (list of strings) - IP addresses of the machine.
- `primaryip` - (string) - The primary (public) IP address for the machine.
- `created` - (string) - The time at which the machine was created.
- `updated` - (string) - The time at which the machine was last updated.

- `compute_node` - (string) - UUID of the server on which the instance is located.
- `nic` - A list of the networks that the machine is attached to. Each network is represented by a `nic`, each of which has the following properties:
- `ip` - The NIC's IPv4 address
- `mac` - The NIC's MAC address
- `primary` - Whether this is the machine's primary NIC
- `netmask` - IPv4 netmask
- `gateway` - IPv4 Gateway
- `network` - The ID of the network to which the NIC is attached
- `state` - The provisioning state of the NIC

The following attributes are used by `cns`:

- `services` - (list of strings) - The list of services that group this instance with others under a shared domain name.
- `disable` - (boolean) - The ability to temporarily disable CNS services domains (optional).

The following attributes are used as `locality` hints:

- `close_to` - (list of strings) - List of container UUIDs that a new instance should be placed alongside, on the same host.
- `far_from` - (list of strings) - List of container UUIDs that a new instance should not be placed onto the same host.

» triton_service_group

The `triton_service_group` resource represents a Triton Service Group.

NOTE: Triton Service Groups are in Preview and only supported in specific regions at this time. They will become Generally Available in the near future.

» Example Usages

```
resource "triton_service_group" "web" {
  group_name = "web_group"
  template   = "${triton_instance_template.base.id}"
  capacity   = 3
}
```

» Argument Reference

The following arguments are supported:

- `group_name` - (string, Required) Friendly name for the service group.
- `template` - (string, Required) Identifier of an instance template.
- `capacity` - (int, Optional) Number of instances to launch and monitor.

» Attribute Reference

The following attributes are exported:

- `id` - (string) - The identifier representing the Triton Service Group.

» triton_snapshot

The `triton_snapshot` resource represents a snapshot of a Triton machine. Snapshots are not usable with other instances; they are a point-in-time snapshot of the current instance. Snapshots can also only be taken of instances that are not of brand `kvm`.

» Example Usages

```
data "triton_image" "ubuntu1604" {
  name    = "ubuntu-16.04"
  version = "20170403"
}

resource "triton_machine" "test" {
  image   = "${data.triton_image.ubuntu1604.id}"
  package = "g4-highcpu-128M"
}

resource "triton_snapshot" "test" {
  name       = "my-snapshot"
  machine_id = "${triton_machine.test.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (string, Required) The name for the snapshot.
- `machine_id` - (string, Required) The ID of the machine of which to take a snapshot.

» Attribute Reference

The following attributes are exported:

- `id` - (string) - The identifier representing the snapshot in Triton.
- `state` - (string) - The current state of the snapshot.

» triton_vlan

The `triton_vlan` resource represents an Triton VLAN. A VLAN provides a low level way to segregate and subdivide the network. Traffic on one VLAN cannot, *on its own*, reach another VLAN.

» Example Usages

» Create a VLAN

```
resource "triton_vlan" "dmz" {
  vlan_id     = 100
  name        = "dmz"
  description = "DMZ VLAN"
}
```

» Argument Reference

The following arguments are supported:

- `vlan_id` - (int, Required, Change forces new resource) Number between 0-4095 indicating VLAN ID
- `name` - (string, Required) Unique name to identify VLAN
- `description` - (string, Optional) Description of the VLAN