» gridscale_ip

Get data of an IP address resource. This can be used to link ip addresses to a server.

» Example Usage

Using ip datasource for the creation of a server:

```
data "gridscale_ipv4" "ipv4name"{
    resource_id = "xxxx-xxxx-xxxx-xxxx"
}

data "gridscale_ipv6" "ipv6name"{
    resource_id = "xxxx-xxxx-xxxx-xxxx"
}

resource "gridscale_server" "servername"{
    name = "terra-server"
    cores = 2
    memory = 4
    ipv4 = data.gridscale_ipv4.ipv4name.id
    ipv6 = data.gridscale_ipv6.ipv6name.id
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the IP address.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the ip.
- `ip` - Defines the IP Address (v4 or v6) the ip.
- `prefix` - The IP prefix of the ip.
- `location_uuid` - The UUID of the location, that helps to identify which datacenter an object belongs to.
- `failover` - failover mode of this ip. If true, then this IP is no longer available for DHCP and can no longer be related to any server..
- `status` - The status of the ip.
- `reverse_dns` - The reverse DNS of the ip.

- `location_iata` - The IATA airport code, which works as a location identifier.
- `location_country` - The human-readable name of the country of the ip.
- `location_name` - The human-readable name of the location of the ip.
- `create_time` - The date and time the ip was initially created.
- `change_time` - The date and time of the last ip change.
- `delete_block` - Defines if the ip is administratively blocked.
- `usage_in_minutes` - Total minutes the ip has been running.
- `current_price` - The price for the current period since the last bill.
- `labels` - The list of labels.

» gridscale_network

Get data of a network resource. This can be used to link networks to a server.

» Example Usage

Using the network datasource for the creation of a server:

```
data "gridscale_network" "networkname"{
    resource_id = "xxxx-xxxx-xxxx-xxxx"
}

resource "gridscale_server" "servername"{
    name = "terra-server"
    cores = 2
    memory = 4
    network {
        object_uuid = data.gridscale_network.networkname.id
        bootdevice = true
    }
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the network.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the network.
- `name` - The UUID of the network.
- `location_uuid` - The UUID of the location, that helps to identify which datacenter the network belongs to.
- `l2security` - Defines information about MAC spoofing protection.
- `status` - The status of the network.
- `network_type` - The type of the network.
- `location_country` - The human-readable name of the country where the network is located.
- `location_iata` - The IATA airport code, which works as a location identifier.
- `location_name` - The human-readable name of the location where the network is located.
- `delete_block` - Defines if the network is administratively blocked.
- `create_time` - Defines the date and time the network was initially created.
- `change_time` - Defines the date and time of the last network change.
- `labels` - The list of labels.

» gridscale_loadbalancer

Get data of an loabalancer resource.

» Example Usage

```
data "gridscale_loadbalancer" "foo" {
    resource_id   = "xxxx-xxxx-xxxx-xxxx"
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the loadbalancer.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the loadbalancer.
- `name` - The human-readable name of the loadbalancer.
- `algorithm` - The algorithm used to process requests.
- `status` - The status of the loadbalancer.

- `redirect_http_to_https` - Whether the Load balancer is forced to redirect requests from HTTP to HTTPS.
- `listen_ipv4_uuid` - The UUID of the IPv4 address the loadbalancer will listen to for incoming requests.
- `listen_ipv6_uuid` - The UUID of the IPv6 address the loadbalancer will listen to for incoming requests.
- `forwarding_rule` - The forwarding rules of the loadbalancer.
- `backend_server` - The servers that the loadbalancer can communicate with.
- `labels` - The list of labels.

» gridscale_paas_securityzone

Get data of a security zone.

» Example Usage

Using the security zone datasource for the creation of a paas:

```
data "gridscale_paas_securityzone" "foo"{
    resource_id = "xxxx-xxxx-xxxx-xxxx"
}


resource "gridscale_paas" "foo"{
    name = "terra-paas-test"
    service_template_uuid = "f9625726-5ca8-4d5c-b9bd-3257e1e2211a"
    security_zone_uuid = data.gridscale_paas_securityzone.foo.id
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the security zone.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the security zone.
- `name` - The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.

- `location_uuid` - Helps to identify which datacenter an object belongs to.
- `location_country` - The human-readable name of the location's country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The human-readable name of the location.
- `create_time` - Defines the date and time the object was initially created.
- `change_time` - Defines the date and time of the last object change.
- `status` - Status indicates the status of the object.
- `labels` - List of labels.
- `relations` - List of PaaS services' UUIDs relating to the security zone.

» gridscale_paas

Get a PaaS resource based on given UUID.

» Example Usage

Retrieving the PaaS datasource:

```
resource "gridscale_paas" "foo" {
  name = "foo"
  service_template_uuid = "f9625726-5ca8-4d5c-b9bd-3257e1e2211a"
}

data "gridscale_paas" "foo" {
    resource_id   = gridscale_paas.foo.id
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the PaaS service.

» Attributes Reference

The following attributes are exported:

- `name` - The human-readable name of the object.
- `username` - Username for PaaS service.
- `password` - Password for PaaS service.
- `listen_port` - Ports that PaaS service listens to.
  - `name` - Name of a port.

    - `listen_port` - Port number.
- `security_zone_uuid` - The UUID of the security zone that the service is running in.
- `network_uuid` - Network UUID containing security zone.
- `service_template_uuid` - The template used to create the service.
- `usage_in_minute` - Number of minutes that PaaS service is in use.
- `current_price` - Current price of PaaS service.
- `change_time` - Time of the last change.
- `create_time` - Time of the creation.
- `status` - Current status of PaaS service.
- `parameter` - Contains the service parameters for the service.
    - `param` - Name of parameter.
    - `value` - Value of the corresponding parameter.
    - `type` - Primitive type of the parameter.
- `resource_limit` - A list of service resource limits.
    - `resource` - The name of the resource you would like to cap.
    - `limit` - The maximum number of the specific resource your service can use.
- `labels` - List of labels in the format ["label1", "label2"].

» gridscale_server

Get data of a server by its UUID.

» Example

```
resource "gridscale_ipv4" "foo1" {
  name   = "newname"
}
resource "gridscale_network" "foo" {
  name   = "newname"
}
resource "gridscale_storage" "foo1" {
  name   = "newname"
  capacity = 1
}
resource "gridscale_server" "foo" {
  name   = "newname"
  cores = 1
  memory = 1
  power = true
  ipv4 = gridscale_ipv4.foo1.id
  network {
```

```
        object_uuid = gridscale_network.foo.id
        rules_v4_in {
                order = 0
                protocol = "tcp"
                action = "drop"
                dst_port = "20:80"
                comment = "test"
        }
        rules_v6_in {
                order = 1
                protocol = "tcp"
                action = "drop"
                dst_port = "10:20"
                comment = "test1"
        }
    }
  storage {
    object_uuid = gridscale_storage.foo1.id
  }
}


data "gridscale_server" "foo" {
    resource_id   = gridscale_server.foo.id
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the firewall.

» Attributes Reference

This resource exports the following attributes:

- `id` - UUID of the server.
- `name` - The name of the server.
- `cores` - The number of server cores.
- `memory` - The amount of server memory in GB.
- `location_uuid` - Helps to identify which datacenter an object belongs to. The location of the resource depends on the location of the project.
- `labels` - List of labels in the format [”label1”, ”label2”].
- `hardware_profile` - The hardware profile of the Server.
- `storage` - Connects a storage to the server.

  - `object_uuid` - The object UUID or id of the storage.
  - `storage_type` - Indicates the speed of the storage. This may be (storage, storage_high or storage_insane).
  - `bootdevice` - True is the storage is a boot device.
  - `object_name` - Name of the storage.
  - `create_time` - Defines the date and time the object was initially created.
  - `capacity` - Capacity of the storage (GB).
  - `controller` - Defines the SCSI controller id. The SCSI defines transmission routes such as Serial Attached SCSI (SAS), Fibre Channel and iSCSI.
  - `bus` - The SCSI bus id. The SCSI defines transmission routes like Serial Attached SCSI (SAS), Fibre Channel and iSCSI. Each SCSI device is addressed via a specific number. Each SCSI bus can have multiple SCSI devices connected to it.
  - `target` - Defines the SCSI target ID. The target ID is a device (e.g. disk).
  - `lun` - Is the common SCSI abbreviation of the Logical Unit Number. A lun is a unique identifier for a single disk or a composite of disks.
  - `license_product_no` - If a template has been used that requires a license key (e.g. Windows Servers) this shows the product_no of the license (see the /prices endpoint for more details).
  - `last_used_template` - Indicates the UUID of the last used template on this storage (inherited from snapshots).
- `network` - Connects a network to the server.
  - `object_uuid` - The object UUID or id of the network.
  - `bootdevice` - Make this network the boot device. This can only be set for one network.
  - `object_name` - Name of the network.
  - `ordering` - Defines the ordering of the network interfaces. Lower numbers have lower PCI-IDs.
  - `create_time` - Defines the date and time the object was initially created.
  - `network_type` - One of network, network_high, network_insane.
  - `mac` - network_mac defines the MAC address of the network interface.
  - `firewall_template_uuid` - The UUID of firewall template.
  - `rules_v4_in` - Firewall template rules for inbound traffic - covers ipv4 addresses.
    - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    - `action` - This defines what the firewall will do. Either accept or drop.

    * `protocol` - Either 'udp' or 'tcp'.
    * `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `comment` - Comment.
- `rules_v4_out` - Firewall template rules for outbound traffic - covers ipv4 addresses.
    * `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    * `action` - This defines what the firewall will do. Either accept or drop.
    * `protocol` - Either 'udp' or 'tcp'.
    * `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `comment` - Comment.
- `rules_v6_in` - Firewall template rules for inbound traffic - covers ipv6 addresses.
    * `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    * `action` - This defines what the firewall will do. Either accept or drop.
    * `protocol` - Either 'udp' or 'tcp'.
    * `dst_port` - A Number between 1 and 65535, port ranges are

separated by a colon for FTP.
        * `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
        * `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
        * `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
        * `comment` - Comment.
    - `rules_v6_out` - Firewall template rules for outbound traffic - covers ipv6 addresses.
        * `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
        * `action` - This defines what the firewall will do. Either accept or drop.
        * `protocol` - Either 'udp' or 'tcp'.
        * `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
        * `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
        * `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
        * `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
        * `comment` - Comment.
- `ipv4` - The UUID of the IPv4 address of the server.
- `ipv6` - The UUID of the IPv6 address of the server.
- `isoimage` - The UUID of an ISO image in gridscale.
- `power` - The power state of the server.
- `availability_zone` - Defines which Availability-Zone the Server is placed.
- `auto_recovery` - If the server should be auto-started in case of a failure.
- `console_token` - The token used by the panel to open the websocket VNC connection to the server console.
- `legacy` - Legacy-Hardware emulation instead of virtio hardware. If enabled, hotplugging cores, memory, storage, network, etc. will not work, but the server will most likely run every x86 compatible operating system. This mode comes with a performance penalty, as emulated hardware does not benefit from the virtio driver infrastructure.
- `status` - Status indicates the status of the object.

- `usage_in_minutes_memory` - Total minutes of memory used.
- `usage_in_minutes_cores` - Total minutes of cores used.
- `create_time` - Defines the date and time the object was initially created.
- `change_time` - Defines the date and time of the last object change.
- `current_price` - The price for the current period since the last bill.

» gridscale_public_network

Get data of the public network. Use this to link your servers to the public network easily.

» Example Usage

Using the public network datasource for the creation of a server:

```
data "gridscale_public_network" "pubnet"{
}

resource "gridscale_server" "servername"{
    name = "terra-server"
    cores = 2
    memory = 4
    network {
        object_uuid = data.gridscale_public_network.pubnet.id
        bootdevice = true
    }
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the public network.

» Attributes Reference

The following attributes are exported:

- `name` - The UUID of the network.
- `location_uuid` - The UUID of the location, that helps to identify which datacenter the network belongs to.
- `l2security` - Defines information about MAC spoofing protection.
- `status` - The status of the network.

- `network_type` - The type of the network.
- `location_country` - The human-readable name of the country where the network is located.
- `location_iata` - The IATA airport code, which works as a location identifier.
- `location_name` - The human-readable name of the location where the network is located.
- `delete_block` - Defines if the network is administratively blocked.
- `create_time` - Defines the date and time the network was initially created.
- `change_time` - Defines the date and time of the last network change.
- `labels` - The list of labels.

» gridscale_sshkey

Get data of an sshkey resource. This can be used to link SSH keys to a storage when an official template is used.

» Example Usage

Using the sshkey datasource for the creation of a storage:

```
data "gridscale_sshkey" "sshkey-john"{
    resource_id = "xxxx-xxxx-xxxx-xxxx"
}

data "gridscale_sshkey" "sshkey-jane"{
    resource_id = "xxxx-xxxx-xxxx-xxxx"
}

resource "gridscale_storage" "storagename"{
    name = "terraform-storage"
    capacity = 10
    template {
        sshkeys = [
            data.gridscale_sshkey.sshkey-john.id,
            data.gridscale_sshkey.sshkey-jane.id
        ]
        template_uuid = "4db64bfc-9fb2-4976-80b5-94ff43b1233a"
    }
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the SSH key.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the sshkey.
- `name` - The human-readable name of the sshkey.
- `sshkey` - The OpenSSH public key string of the sshkey.
- `status` - The status of the sshkey.
- `create_time` - The date and time of the sshkey was initially created.
- `change_time` - The date and time of the last sshkey change.
- `labels` - The list of labels.

» gridscale_storage

Get data of a storage resource. This can be used to link storages to a server.

» Example Usage

Using the storage datasource for the creation of a server:

```
data "gridscale_storage" "storagename"{
    resource_id = "xxxx-xxxx-xxxx-xxxx"
}

resource "gridscale_server" "servername"{
    name = "terra-server"
    cores = 2
    memory = 4
    storage {
        object_uuid = data.gridscale_storage.storagename.id
        bootdevice = true
    }
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the storage.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the storage.
- `change_time` - Defines the date and time of the last storage change.
- `location_iata` - The IATA airport code of the location where storage locates.
- `status` - The status of the storage.
- `license_product_no` - The license key (e.g. Windows Servers), if the template used by the storage requires.
- `location_country` - The human-readable name of the country where the storage locates.
- `usage_in_minutes` - Total minutes the the storage has been running.
- `last_used_template` - The UUID of the last used template on the storage.
- `current_price` - The price for the current period since the last bill.
- `capacity` - The capacity (GB) of the storage.
- `location_uuid` - The UUID of the location where the storage locates.
- `storage_type` - The type of the storage.
- `parent_uuid` - The UUID of the parent of the storage.
- `name` - The human-readable name of the storage.
- `location_name` - The human-readable name of the location where the storage locates.
- `create_time` - Defines the date and time the storage was initially created.
- `labels` - The list of labels.

» gridscale_snapshot

Get data of a storage snapshot resource.

» Example Usage

```
resource "gridscale_storage" "foo" {
  name   = "storage"
  capacity = 1
}
resource "gridscale_snapshot" "foo" {
  name = "snapshot"
  storage_uuid = gridscale_storage.foo.id
}
```

```
data "gridscale_snapshot" "foo" {
    resource_id   = gridscale_snapshot.foo.id
    storage_uuid = gridscale_storage.foo.id
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) ID of a resource (UUID of snapshot).
- `storage_uuid` - (Required) UUID of the storage used to create this snapshot.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the snapshot.
- `storage_uuid` - The UUID of the storage used to create this snapshot.
- `name` - The name of the snapshot.
- `status` - The status of the snapshot.
- `location_uuid` - The UUID of the location, that helps to identify which datacenter an object belongs to.
- `location_iata` - The IATA airport code, which works as a location identifier.
- `location_country` - The human-readable name of the country of the snapshot.
- `location_name` - The human-readable name of the location of the snapshot.
- `create_time` - The date and time the ip was initially created.
- `change_time` - The date and time of the last snapshot change.
- `usage_in_minutes` - Total minutes the ip has been running.
- `current_price` - The price for the current period since the last bill.
- `capacity` - The capacity of the snapshot in GB.
- `license_product_no` - If a template has been used that requires a license key (e.g. Windows Servers) this shows the product_no of the license (see the /prices endpoint for more details).
- `labels` - The list of labels.

» gridscale_snapshotschedule

Gets data of a storage snapshot schedule.

» Example Usage

```
resource "gridscale_storage" "foo" {
  name   = "storage"
  capacity = 1
}
resource "gridscale_snapshotschedule" "foo" {
  name = "snapshotschedule"
  storage_uuid = gridscale_storage.foo.id
  keep_snapshots = 1
  run_interval = 60
  next_runtime = "2025-12-30 15:04:05"
}
data "gridscale_snapshotschedule" "foo" {
    resource_id   = gridscale_snapshotschedule.foo.id
    storage_uuid   = gridscale_storage.foo.id
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) UUID of the snapshot schedule.
- `storage_uuid` - (Required) UUID of the storage that the snapshot schedule belongs to.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the snapshot schedule.
- `storage_uuid` - UUID of the storage that the snapshot schedule belongs to.
- `status` - The status of the snapshot schedule.
- `name` - The human-readable name of the snapshot schedule.
- `next_runtime` - The date and time that the snapshot schedule will be run.
- `keep_snapshots` - The amount of Snapshots to keep before overwriting the last created Snapshot.
- `run_interval` - The interval at which the schedule will run (in minutes).
- `create_time` - The date and time the snapshot schedule was initially created.
- `change_time` - The date and time of the last snapshot schedule change.
- `labels` - The list of labels.
- `snapshot` - Related snapshots.

  - `name` - Name of the snapshot.
  - `object_uuid` - UUID of the snapshot.
  - `create_time` - The date and time the snapshot was initially created.

» gridscale_template

Get data of a template with a specific name. This can be used to make it more visible which template is being used for new storages.

An error is triggered if the template name does not exist.

» Example Usage

Get the template:

```
data "gridscale_template" "ubuntu" {
  name = "Ubuntu 18.04 LTS"
}
```

Using the template datasource for the creation of a storage:

```
resource "gridscale_storage" "storage-test"{
    name = "terra-storage-test"
    capacity = 10
    template {
        sshkeys = [ "e17e8fd2-0797-4a00-a85d-eb9a612a6e4e" ]
        template_uuid = data.gridscale_template.ubuntu.id
    }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The exact name of the template as show in the expert panel of gridscale.

» Attributes Reference

The following attributes are exported:

- `name` - The name of the template.
- `id` - The UUID of the template.
- `location_uuid` - Helps to identify which datacenter an object belongs to.

- `location_country` - Formatted by the 2 digit country code (ISO 3166-2) of the host country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The human-readable name of the location. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `status` - Status indicates the status of the object.
- `ostype` - The operating system installed in the template.
- `version` - The version of the template.
- `private` - The object is private, the value will be true. Otherwise the value will be false.
- `license_product_no` - If a template has been used that requires a license key (e.g. Windows Servers) this shows the product_no of the license (see the /prices endpoint for more details).
- `create_time` - The date and time the object was initially created.
- `change_time` - The date and time of the last object change.
- `distro` - The OS distrobution that the Template contains.
- `description` - Description of the Template.
- `usage_in_minutes` - Total minutes the object has been running.
- `capacity` - The capacity of a storage/ISO Image/template/snapshot in GB.
- `current_price` - Defines the price for the current period since the last bill.
- `labels` - List of labels.

» gridscale_object_storage_accesskey

Get data of an access key resource of an object storage.

» Example Usage

```
resource "gridscale_object_storage_accesskey" "foo" {
}

data "gridscale_object_storage_accesskey" "foo" {
    resource_id   = "${gridscale_object_storage_accesskey.foo.id}"
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) ID of a resource (access key of an object storage).

» Attributes Reference

The following attributes are exported:

- `id` - The access key of the object storage.
- `access_key` - Access key of an object storage.
- `secret_key` - Secret key of an object storage.

» gridscale_isoimage

Get data of an ISO Image by its UUID.

» Example Usage

```
resource "gridscale_isoimage" "foo" {
  name   = "name"
  source_url = "http://tinycorelinux.net/10.x/x86/release/TinyCore-current.iso"
}

data "gridscale_isoimage" "foo" {
    resource_id   = gridscale_isoimage.foo.id
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the ISO Image.

» Attributes Reference

The following attributes are exported:

- `name` - The name of the ISO Image.
- `source_url` - Contains the source URL of the ISO Image that it was originally fetched from.
- `server` - The information about servers which are related to this ISO Image.
  - `object_uuid` - The object UUID or id of the server.

    - `object_name` - Name of the server.
    - `create_time` - The date and time the object was initially created.
    - `bootdevice` - True if the ISO Image is a boot device of this server.
- `id` - The UUID of the ISO Image.
- `location_uuid` - Helps to identify which datacenter an object belongs to.
- `location_country` - Formatted by the 2 digit country code (ISO 3166-2) of the host country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The human-readable name of the location. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `status` - Status indicates the status of the object.
- `version` - The version of the ISO Image.
- `private` - The object is private, the value will be true. Otherwise the value will be false.
- `create_time` - The date and time the object was initially created.
- `change_time` - The date and time of the last object change.
- `description` - Description of the Template.
- `usage_in_minutes` - Total minutes the object has been running.
- `capacity` - The capacity of a storage/ISO Image/ISO Image/snapshot in GB.
- `current_price` - Defines the price for the current period since the last bill.
- `labels` - List of labels.

» gridscale_firewall

Get data of a firewall by its UUID.

» Example Usage

```
resource "gridscale_firewall" "foo" {
  name   = "example-firewall"
  rules_v4_in {
    order = 0
    protocol = "tcp"
    action = "drop"
    dst_port = "20:80"
    comment = "some comments"
  }
  rules_v6_in {
    order = 0
    protocol = "tcp"
```

```
    action = "drop"
    dst_port = "2000:3000"
    comment = "some comments"
  }
}

data "gridscale_firewall" "foo" {
    resource_id   = gridscale_firewall.foo.id
}
```

» Argument Reference

The following arguments are supported:

- `resource_id` - (Required) The UUID of the firewall.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the firewall.
- `name` - The name of the firewall.
- `rules_v4_in` - Firewall template rules for inbound traffic - covers ipv4 addresses.
  - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - This defines what the firewall will do. Either accept or drop.
  - `protocol` - Either 'udp' or 'tcp'.
  - `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - Comment.
- `rules_v4_out` - Firewall template rules for outbound traffic - covers ipv4 addresses.
  - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other

rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - This defines what the firewall will do. Either accept or drop.
  - `protocol` - Either 'udp' or 'tcp'.
  - `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - Comment.
- `rules_v6_in` - Firewall template rules for inbound traffic - covers ipv6 addresses.
  - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - This defines what the firewall will do. Either accept or drop.
  - `protocol` - Either 'udp' or 'tcp'.
  - `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - Comment.
- `rules_v6_out` - Firewall template rules for outbound traffic - covers ipv6 addresses.
  - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - This defines what the firewall will do. Either accept or drop.
  - `protocol` - Either 'udp' or 'tcp'.
  - `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.

  - `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - Comment.
- `network` - The information about networks which are related to this firewall.
  - `object_uuid` - The object UUID or id of the firewall.
  - `object_name` - Name of the firewall.
  - `network_uuid` - The object UUID or id of the network.
  - `network_name` - Name of the network.
  - `create_time` - The date and time the object was initially created.
- `location_name` - The human-readable name of the location. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `status` - Status indicates the status of the object.
- `private` - The object is private, the value will be true. Otherwise the value will be false.
- `create_time` - The date and time the object was initially created.
- `change_time` - The date and time of the last object change.
- `description` - Description of the firewall.
- `labels` - List of labels.

» gridscale_ipv4

Provides an IPv4 address resource. This can be used to create, modify and delete IPv4 addresses.

» Example Usage

The following example shows how one might use this resource to add an IPv4 address to gridscale:

```
resource "gridscale_ipv4" "terra-ipv4-test" {
    name = "terra-test"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `failover` - (Optional) Sets failover mode for this IP. If true, then this IP is no longer available for DHCP and can no longer be related to any server.

- `reverse_dns` - (Optional) Defines the reverse DNS entry for the IP Address (PTR Resource Record).
- `labels` - (Optional) List of labels in the format ["label1", "label2"].

» Attributes

This resource exports the following attributes:

- `name` - See Argument Reference above.
- `location_uuid` - See Argument Reference above.
- `failover` - See Argument Reference above.
- `reverse_dns` - See Argument Reference above.
- `labels` - See Argument Reference above.
- `ip` - Defines the IP Address.
- `prefix` - The network address and the subnet.
- `status` - status indicates the status of the object.
- `create_time` - The time the object was created.
- `change_time` - Defines the date and time of the last object change.
- `location_country` - Formatted by the 2 digit country code (ISO 3166-2) of the host country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The location name.
- `delete_block` - Defines if the object is administratively blocked. If true, it can not be deleted by the user.
- `usage_in_minutes` - The amount of minutes the IP address has been in use.
- `current_price` - The price for the current period since the last bill.

» gridscale_ipv6

Provides an IPv6 address resource. This can be used to create, modify and delete IPv6 addresses.

» Example Usage

The following example shows how one might use this resource to add an IPv6 address to gridscale:

```
resource "gridscale_ipv6" "terra-ipv6-test" {
    name = "terra-test"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `failover` - (Optional) Sets failover mode for this IP. If true, then this IP is no longer available for DHCP and can no longer be related to any server.
- `reverse_dns` - (Optional) Defines the reverse DNS entry for the IP Address (PTR Resource Record).
- `labels` - (Optional) List of labels in the format ["label1", "label2"].

» Attributes

This resource exports the following attributes:

- `name` - See Argument Reference above.
- `location_uuid` - Helps to identify which datacenter an object belongs to. The location of the resource depends on the location of the project.
- `failover` - See Argument Reference above.
- `reverse_dns` - See Argument Reference above.
- `labels` - See Argument Reference above.
- `ip` - Defines the IP Address.
- `prefix` - The network address and the subnet.
- `status` - status indicates the status of the object.
- `create_time` - The time the object was created.
- `change_time` - Defines the date and time of the last object change.
- `location_country` - Formatted by the 2 digit country code (ISO 3166-2) of the host country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The location name.
- `delete_block` - Defines if the object is administratively blocked. If true, it can not be deleted by the user.
- `usage_in_minutes` - The amount of minutes the IP address has been in use.
- `current_price` - The price for the current period since the last bill.

» gridscale_loadbalancer

Provides a loadbalancer resource. This can be used to create, modify and delete loadbalancers.

» Example Usage

```
resource "gridscale_loadbalancer" "foo" {
    name   = "%s"
    algorithm = "%s"
    redirect_http_to_https = false
    listen_ipv4_uuid = gridscale_ipv4.lb.id
    listen_ipv6_uuid = gridscale_ipv6.lb.id
    labels = []
    backend_server {
        weight = 100
        host   = gridscale_ipv4.server.ip
    }
    forwarding_rule {
        listen_port =  80
        mode        =  "http"
        target_port =  80
    }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `redirect_http_to_https` - (Required) Whether the loadbalancer is forced to redirect requests from HTTP to HTTPS.
- `listen_ipv4_uuid` - (Required) The UUID of the IPv4 address the loadbalancer will listen to for incoming requests.
- `listen_ipv6_uuid` - (Required) The UUID of the IPv6 address the loadbalancer will listen to for incoming requests.
- `algorithm` - (Required) The algorithm used to process requests. Accepted values: roundrobin/leastconn.
- `labels` - (Optional) List of labels in the format [”label1”, ”label2”].

» Attributes

This resource exports the following attributes:

- `id` - The UUID of the loadbalancer.

- `location_uuid` - Helps to identify which datacenter an object belongs to. The location of the resource depends on the location of the project.
- `name` - The human-readable name of the loadbalancer.
- `algorithm` - The algorithm used to process requests.
- `status` - The status of the loadbalancer.
- `redirect_http_to_https` - Whether the Load balancer is forced to redirect requests from HTTP to HTTPS.
- `listen_ipv4_uuid` - The UUID of the IPv4 address the loadbalancer will listen to for incoming requests.
- `listen_ipv6_uuid` - The UUID of the IPv6 address the loadbalancer will listen to for incoming requests.
- `forwarding_rule` - The forwarding rules of the loadbalancer.
- `backend_server` - The servers that the loadbalancer can communicate with.
- `labels` - The list of labels.

» gridscale_network

Provides a network resource. This can be used to create, modify and delete networks.

» Example Usage

The following example shows how one might use this resource to add a network to gridscale:

```
resource "gridscale_network" "networkname"{
    name = "terraform-network"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `l2security` - (Optional) Defines information about MAC spoofing protection (filters layer2 and ARP traffic based on MAC source). It can only be (de-)activated on a private network - the public network always has l2security enabled. It will be true if the network is public, and false if the network is private.
- `labels` - (Optional) List of labels in the format [”label1”, ”label2”].

» Attributes

This resource exports the following attributes:

- `name` - See Argument Reference above.
- `location_uuid` - Helps to identify which datacenter an object belongs to. The location of the resource depends on the location of the project.
- `l2security` - See Argument Reference above.
- `labels` - See Argument Reference above.
- `status` - status indicates the status of the object.
- `create_time` - The time the object was created.
- `change_time` - Defines the date and time of the last object change.
- `network_type` - The type of this network, can be mpls, breakout or network.
- `location_country` - Formatted by the 2 digit country code (ISO 3166-2) of the host country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The location name.
- `public_net` - Is the network public or not.
- `delete_block` - If deleting this network is allowed.

» gridscale_paas_securityzone

Provides a security zone resource. This can be used to create, modify and delete security zones.

» Example Usage

The following example shows how one might use this resource to add a security zone to gridscale:

```
resource "gridscale_paas_securityzone" "foo" {
  name = "test"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `location_uuid` - (Optional) Helps to identify which datacenter an object belongs to.

» Attributes

This resource exports the following attributes:

- `id` - The UUID of the security zone.
- `name` - The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `location_uuid` - Helps to identify which datacenter an object belongs to.
- `location_country` - The human-readable name of the location's country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The human-readable name of the location.
- `create_time` - Defines the date and time the object was initially created.
- `change_time` - Defines the date and time of the last object change.
- `status` - Status indicates the status of the object.
- `labels` - List of labels.
- `relations` - List of PaaS services' UUIDs relating to the security zone.

» gridscale_paas

Provides a PaaS resource. This can be used to create, modify and delete PaaS.

» Example

The following example shows how one might use this resource to add a PaaS to gridscale:

```
resource "gridscale_paas" "terra-paas-test" {
  name = "terra-paas-test"
  service_template_uuid = "f9625726-5ca8-4d5c-b9bd-3257e1e2211a"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `service_template_uuid` - (Required) The template used to create the service.
- `labels` - (Optional) List of labels in the format ["label1", "label2"].

- `security_zone_uuid` - (Optional) The UUID of the security zone that the service is running in.
- `parameters` - (Optional) Contains the service parameters for the service.
  - `param` - (Required) Name of parameter.
  - `value` - (Required) Value of the corresponding parameter.
- `resource_limit` - (Optional) A list of service resource limits..
  - `resource` - (Required) The name of the resource you would like to cap.
  - `limit` - (Required) The maximum number of the specific resource your service can use.
  - `type` - (Required) Primitive type of the parameter: bool, int (better use float for int case), float, string.

» Attributes

This resource exports the following attributes:

- `name` - See Argument Reference above.
- `username` - Username for PaaS service.
- `password` - Password for PaaS service.
- `listen_port` - Ports that PaaS service listens to.
  - `name` - Name of a port.
  - `listen_port` - Port number.
- `security_zone_uuid` - See Argument Reference above.
- `network_uuid` - Network UUID containing security zone.
- `service_template_uuid` - See Argument Reference above.
- `usage_in_minute` - Number of minutes that PaaS service is in use.
- `current_price` - Current price of PaaS service.
- `change_time` - Time of the last change.
- `create_time` - Time of the creation.
- `status` - Current status of PaaS service.
- `parameter` - See Argument Reference above.
  - `param` - See Argument Reference above.
  - `value` - See Argument Reference above.
  - `type` - See Argument Reference above.
- `resource_limit` - See Argument Reference above.
  - `resource` - See Argument Reference above.
  - `limit` - See Argument Reference above.
- `labels` - See Argument Reference above.

» gridscale_server

Provides a server resource. This can be used to create, modify and delete servers.

» Example

The following example shows how one might use this resource to add a server to gridscale:

```
resource "gridscale_server" "terra-server-test"{
    name = "terra-server-test"
    cores = 2
    memory = 1
    storage {
        object_uuid = gridscale_storage.terra-storage-test.id
        bootdevice = true
    }
    storage {
            object_uuid = "UUID of storage 2",
        }
    network {
        object_uuid = gridscale_network.terra-network-test.id
        bootdevice = true
    }
    network {
            object_uuid = "UUID of network 2"
    }
    ipv4 = gridscale_ipv4.terra-ipv4-test.id}
    ipv6 = "UUID of ipv6 address"
    isoimage = "9be3e0a3-42ac-4207-8887-3383c405724d"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `cores` - (Required) The number of server cores.
- `memory` - (Required) The amount of server memory in GB.
- `labels` - (Optional) List of labels in the format [”label1”, ”label2”].
- `auto_recovery` - (Optional) If the server should be auto-started in case of a failure (default=true).

- `hardware_profile` - (Optional, ForceNew) The hardware profile of the Server. Options are default, legacy, nested, cisco_csr, sophos_utm, f5_bigip and q35 at the moment of writing. Check the
- `ipv4` - (Optional) The UUID of the IPv4 address of the server. (***NOTE: The server will NOT automatically be connected to the public network; to give it access to the internet, please add server to the public network.)
- `ipv6` - (Optional) The UUID of the IPv6 address of the server. (***NOTE: The server will NOT automatically be connected to the public network; to give it access to the internet, please add server to the public network.)
- `isoimage` - (Optional) The UUID of an ISO image in gridscale. The server will automatically boot from the ISO if one was added. The UUIDs of ISO images can be found in the expert panel.
- `power` - (Optional, Computed) The power state of the server. Set this to true to will boot the server, false will shut it down.
- `availability_zone` - (Optional, Computed) Defines which Availability-Zone the Server is placed.
- `storage` - (Optional) Connects a storage to the server.
  - `object_uuid` - (Required) The object UUID or id of the storage.
- `network` - (Optional) Connects a network to the server.
  - `object_uuid` - (Required) The object UUID or id of the network.
  - `bootdevice` - (Optional, Computed) Make this network the boot device. This can only be set for one network.
  - `firewall_template_uuid` - (Optional) The UUID of firewall template.
  - `rules_v4_in` - (Optional) Firewall template rules for inbound traffic - covers ipv4 addresses.
    - `order` - (Required) The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    - `action` - (Required) This defines what the firewall will do. Either accept or drop.
    - `protocol` - (Required) Either 'udp' or 'tcp'.
    - `dst_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    - `src_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.

    * `src_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `dst_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `comment` - (Optional) Comment.

- `rules_v4_out` - (Optional) Firewall template rules for outbound traffic - covers ipv4 addresses.

    * `order` - (Required) The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    * `action` - (Required) This defines what the firewall will do. Either accept or drop.
    * `protocol` - (Required) Either 'udp' or 'tcp'.
    * `dst_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `dst_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `comment` - (Optional) Comment.

- `rules_v6_in` - (Optional) Firewall template rules for inbound traffic - covers ipv6 addresses.

    * `order` - (Required) The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    * `action` - (Required) This defines what the firewall will do. Either accept or drop.
    * `protocol` - (Required) Either 'udp' or 'tcp'.
    * `dst_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.

    - `src_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    - `dst_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    - `comment` - (Optional) Comment.
  - `rules_v6_out` - (Optional) Firewall template rules for outbound traffic - covers ipv6 addresses.
    - `order` - (Required) The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    - `action` - (Required) This defines what the firewall will do. Either accept or drop.
    - `protocol` - (Required) Either 'udp' or 'tcp'.
    - `dst_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    - `src_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    - `src_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    - `dst_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    - `comment` - (Optional) Comment.

» Attributes

This resource exports the following attributes:

- `id` - UUID of the server.
- `name` - The name of the server.
- `cores` - The number of server cores.
- `memory` - The amount of server memory in GB.
- `location_uuid` - Helps to identify which datacenter an object belongs to. The location of the resource depends on the location of the project.
- `labels` - List of labels in the format ["label1", "label2"].
- `hardware_profile` - The hardware profile of the Server.
- `storage` - Connects a storage to the server.
  - `object_uuid` - The object UUID or id of the storage.

  - `storage_type` - Indicates the speed of the storage. This may be (storage, storage_high or storage_insane).
  - `bootdevice` - True is the storage is a boot device.
  - `object_name` - Name of the storage.
  - `create_time` - Defines the date and time the object was initially created.
  - `capacity` - Capacity of the storage (GB).
  - `controller` - Defines the SCSI controller id. The SCSI defines transmission routes such as Serial Attached SCSI (SAS), Fibre Channel and iSCSI.
  - `bus` - The SCSI bus id. The SCSI defines transmission routes like Serial Attached SCSI (SAS), Fibre Channel and iSCSI. Each SCSI device is addressed via a specific number. Each SCSI bus can have multiple SCSI devices connected to it.
  - `target` - Defines the SCSI target ID. The target ID is a device (e.g. disk).
  - `lun` - Is the common SCSI abbreviation of the Logical Unit Number. A lun is a unique identifier for a single disk or a composite of disks.
  - `license_product_no` - If a template has been used that requires a license key (e.g. Windows Servers) this shows the product_no of the license (see the /prices endpoint for more details).
  - `last_used_template` - Indicates the UUID of the last used template on this storage (inherited from snapshots).
- `network` - Connects a network to the server.
  - `object_uuid` - The object UUID or id of the network.
  - `bootdevice` - Make this network the boot device. This can only be set for one network.
  - `object_name` - Name of the network.
  - `ordering` - Defines the ordering of the network interfaces. Lower numbers have lower PCI-IDs.
  - `create_time` - Defines the date and time the object was initially created.
  - `network_type` - One of network, network_high, network_insane.
  - `mac` - network_mac defines the MAC address of the network interface.
  - `firewall_template_uuid` - The UUID of firewall template.
  - `rules_v4_in` - Firewall template rules for inbound traffic - covers ipv4 addresses.
    - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    - `action` - This defines what the firewall will do. Either accept or drop.
    - `protocol` - Either 'udp' or 'tcp'.

    * `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `comment` - Comment.
- `rules_v4_out` - Firewall template rules for outbound traffic - covers ipv4 addresses.
    * `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    * `action` - This defines what the firewall will do. Either accept or drop.
    * `protocol` - Either 'udp' or 'tcp'.
    * `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
    * `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
    * `comment` - Comment.
- `rules_v6_in` - Firewall template rules for inbound traffic - covers ipv6 addresses.
    * `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
    * `action` - This defines what the firewall will do. Either accept or drop.
    * `protocol` - Either 'udp' or 'tcp'.
    * `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.

        * `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
        * `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
        * `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
        * `comment` - Comment.
    - `rules_v6_out` - Firewall template rules for outbound traffic - covers ipv6 addresses.
        * `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
        * `action` - This defines what the firewall will do. Either accept or drop.
        * `protocol` - Either 'udp' or 'tcp'.
        * `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
        * `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
        * `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
        * `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
        * `comment` - Comment.
- `ipv4` - The UUID of the IPv4 address of the server.
- `ipv6` - The UUID of the IPv6 address of the server.
- `isoimage` - The UUID of an ISO image in gridscale.
- `power` - The power state of the server.
- `availability_zone` - Defines which Availability-Zone the Server is placed.
- `auto_recovery` - If the server should be auto-started in case of a failure.
- `console_token` - The token used by the panel to open the websocket VNC connection to the server console.
- `legacy` - Legacy-Hardware emulation instead of virtio hardware. If enabled, hotplugging cores, memory, storage, network, etc. will not work, but the server will most likely run every x86 compatible operating system. This mode comes with a performance penalty, as emulated hardware does not benefit from the virtio driver infrastructure.
- `status` - Status indicates the status of the object.
- `usage_in_minutes_memory` - Total minutes of memory used.

- `usage_in_minutes_cores` - Total minutes of cores used.
- `create_time` - Defines the date and time the object was initially created.
- `change_time` - Defines the date and time of the last object change.
- `current_price` - The price for the current period since the last bill.

» gridscale_sshkey

Provides an SSH public key resource. This can be used to create, modify and delete SSH public keys.

» Example Usage

The following example shows how one might use this resource to add an SSH public key to gridscale:

```
resource "gridscale_sshkey" "sshkey-john"{
    name = "john's computer"
    sshkey = "an ssh public key"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `sshkey` - (Required) This is the OpenSSH public key string (all key types are supported => ed25519, ecdsa, dsa, rsa, rsa1).
- `labels` - (Optional) List of labels in the format [”label1”, ”label2”].

» Attributes

This resource exports the following attributes:

- `name` - See Argument Reference above.
- `sshkey` - See Argument Reference above.
- `labels` - See Argument Reference above.
- `status` - status indicates the status of the object.
- `create_time` - The time the object was created.
- `change_time` - Defines the date and time of the last object change.

» gridscale_storage

Provides a storage resource. This can be used to create, modify and delete storages.

» Example Usage

The following example shows how one might use this resource to add a storage to gridscale:

```
resource "gridscale_storage" "storage-john"{
    name = "john's storage"
    capacity = 10
    storage_type = "storage_high"
    template {
        template_uuid = "4db64bfc-9fb2-4976-80b5-94ff43b1233a"
        password = var.gridscale_password-john
        password_type = "plain"
        hostname = "Ubuntu"
    }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `capacity` - (Required) required (integer - minimum: 1 - maximum: 4096).
- `storage_type` - (Optional) (one of storage, storage_high, storage_insane).
- `labels` - (Optional) List of labels in the format ["label1", "label2"].
- `template` - (Optional) List of labels in the format ["label1", "label2"].
  - `template_uuid` - (Required) The UUID of a template. This can be found in the expert panel by clicking more on the template or by using a gridscale_template datasource.
  - `password` - (Optional) The root (Linux) or Administrator (Windows) password to set for the installed storage. Valid only for public templates. The password has to be either plain-text or a crypt string (modular crypt format - MCF).

  - `password_type` - (Optional) (one of plain, crypt) Required if password is set (ignored for private templates and public Windows templates).
  - `sshkeys` - (Optional) (array of any - minItems: 0) Public Linux templates only! The UUIDs of SSH keys to be added for the root user.
  - `hostname` - (Optional) The hostname of the installed server (ignored for private templates and public windows templates).

Note When using official templates using either a password and password_type or at least one SSH public key is required. This is not the case when using custom templates. For official templates password authentication for SSH is enabled by default, so be sure to pick a strong password.

» Attributes

This resource exports the following attributes:

- `name` - See Argument Reference above.
- `capacity` - See Argument Reference above.
- `storage_type` - See Argument Reference above.
- `location_uuid` - Helps to identify which datacenter an object belongs to. The location of the resource depends on the location of the project.
- `labels` - See Argument Reference above.
- `status` - status indicates the status of the object.
- `create_time` - The time the object was created.
- `change_time` - Defines the date and time of the last object change.
- `location_country` - Formatted by the 2 digit country code (ISO 3166-2) of the host country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The location name.
- `license_product_no` - If a template has been used that requires a license key (e.g. Windows Servers) this shows the product_no of the license (see the /prices endpoint for more details).
- `last_used_template` - Indicates the UUID of the last used template on this storage (inherited from snapshots).
- `usage_in_minutes` - The amount of minutes the IP address has been in use.
- `current_price` - The price for the current period since the last bill.

» gridscale_snapshot

Provides a storage snapshot resource. This can be used to create, modify and delete storage snapshots.

» Example Usage

```
resource "gridscale_storage" "foo" {
  name   = "storage"
  capacity = 1
}
resource "gridscale_snapshot" "foo" {
  name = "snapshot"
  storage_uuid = gridscale_storage.foo.id
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the snapshot.
- `storage_uuid` - (Required) UUID of the storage used to create this snapshot.
- `labels` - (Optional) The list of labels.
- `rollback` - (Optional) Returns a storage to the state of the selected Snapshot.
  - `id` - (Required) ID of the rollback request. It can be any string value. Each rollback request has to have a UNIQUE id.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the snapshot.
- `storage_uuid` - See Argument Reference above.
- `name` - See Argument Reference above.
- `status` - The status of the snapshot.
- `location_uuid` - The UUID of the location, that helps to identify which datacenter an object belongs to.
- `location_iata` - The IATA airport code, which works as a location identifier.

- `location_country` - The human-readable name of the country of the snapshot.
- `location_name` - The human-readable name of the location of the snapshot.
- `create_time` - The date and time the ip was initially created.
- `change_time` - The date and time of the last snapshot change.
- `usage_in_minutes` - Total minutes the ip has been running.
- `current_price` - The price for the current period since the last bill.
- `capacity` - The capacity of the snapshot in GB.
- `license_product_no` - If a template has been used that requires a license key (e.g. Windows Servers) this shows the product_no of the license (see the /prices endpoint for more details).
- `rollback` - See Argument Reference above.
  - `id` - See Argument Reference above.
  - `rollback_time` - The time when rollback request is fulfilled.
  - `status` - Status of the rollback request.
- `labels` - See Argument Reference above.

» gridscale_snapshotschedule

Provides a storage snapshot schedule resource. This can be used to create, modify and delete snapshot schedules.

» Example Usage

```
resource "gridscale_storage" "foo" {
  name   = "storage"
  capacity = 1
}
resource "gridscale_snapshotschedule" "foo" {
  name = "snapshotschedule"
  storage_uuid = gridscale_storage.foo.id
  keep_snapshots = 1
  run_interval = 60
  next_runtime = "2025-12-30 15:04:05"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) UUID of the snapshot schedule.

- `storage_uuid` - (Required) UUID of the storage that the snapshot schedule belongs to.
- `labels` - (Optional) The list of labels.
- `next_runtime` - (Optional) The date and time that the snapshot schedule will be run.
- `keep_snapshots` - (Required) The amount of Snapshots to keep before overwriting the last created Snapshot (>=1).
- `run_interval` - (Required) The interval at which the schedule will run (in minutes, >=60).

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the snapshot schedule.
- `storage_uuid` - See Argument Reference above.
- `status` - The status of the snapshot schedule.
- `name` - See Argument Reference above.
- `next_runtime` - See Argument Reference above.
- `keep_snapshots` - See Argument Reference above.
- `run_interval` - See Argument Reference above.
- `create_time` - The date and time the snapshot schedule was initially created.
- `change_time` - The date and time of the last snapshot schedule change.
- `labels` - See Argument Reference above.
- `snapshot` - Related snapshots.
  - `name` - Name of the snapshot.
  - `object_uuid` - UUID of the snapshot.
  - `create_time` - The date and time the snapshot was initially created.

» gridscale_template

Provides a template resource. This can be used to create, modify and delete template.

» Example Usage

The following example shows how one might use this resource to add a template to gridscale:

```
resource "gridscale_storage" "foo" {
  name   = "newname"
  capacity = 1
}

resource "gridscale_snapshot" "foo" {
  name = "newname"
  storage_uuid = gridscale_storage.foo.id
}

resource "gridscale_template" "foo" {
  name   = "newname"
  snapshot_uuid = gridscale_snapshot.foo.id
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The exact name of the template as show in the expert panel of gridscale.
- `snapshot_uuid` - (Required) Snapshot uuid for template.
- `labels` - (Optional) List of labels.

» Attributes Reference

The following attributes are exported:

- `name` - The name of the template.
- `id` - The UUID of the template.
- `location_uuid` - Helps to identify which datacenter an object belongs to.
- `location_country` - Formatted by the 2 digit country code (ISO 3166-2) of the host country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The human-readable name of the location. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `status` - Status indicates the status of the object.
- `ostype` - The operating system installed in the template.
- `version` - The version of the template.
- `private` - The object is private, the value will be true. Otherwise the value will be false.

- `license_product_no` - If a template has been used that requires a license key (e.g. Windows Servers) this shows the product_no of the license (see the /prices endpoint for more details).
- `create_time` - The date and time the object was initially created.
- `change_time` - The date and time of the last object change.
- `distro` - The OS distrobution that the Template contains.
- `description` - Description of the Template.
- `usage_in_minutes` - Total minutes the object has been running.
- `capacity` - The capacity of a storage/ISO Image/template/snapshot in GB.
- `current_price` - Defines the price for the current period since the last bill.
- `labels` - List of labels.

» gridscale_object_storage_accesskey

Provides an access key resource of an object storage. This can be used to create, modify and delete object storages' access keys.

» Example Usage

```
resource "gridscale_object_storage_accesskey" "foo" {
}
```

» Attributes Reference

The following attributes are exported:

- `id` - The access key of the object storage.
- `access_key` - Access key of an object storage.
- `secret_key` - Secret key of an object storage.

» gridscale_isoimage

Provides an ISO Image resource. This can be used to create, modify and delete ISO Images.

» Example Usage

```
resource "gridscale_isoimage" "foo" {
```

```
  name   = "newname"
  source_url = "http://tinycorelinux.net/10.x/x86/release/TinyCore-current.iso"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `source_url` - (Required) Contains the source URL of the ISO Image that it was originally fetched from.
- `labels` - (Optional) List of labels in the format ["label1", "label2"].

» Attributes Reference

The following attributes are exported:

- `name` - The name of the ISO Image.
- `source_url` - Contains the source URL of the ISO Image that it was originally fetched from.
- `server` - The information about servers which are related to this ISO Image.
  - `object_uuid` - The object UUID or id of the server.
  - `object_name` - Name of the server.
  - `create_time` - The date and time the object was initially created.
  - `bootdevice` - True if the ISO Image is a boot device of this server.
- `id` - The UUID of the ISO Image.
- `location_uuid` - Helps to identify which datacenter an object belongs to.
- `location_country` - Formatted by the 2 digit country code (ISO 3166-2) of the host country.
- `location_iata` - Uses IATA airport code, which works as a location identifier.
- `location_name` - The human-readable name of the location. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `status` - Status indicates the status of the object.
- `version` - The version of the ISO Image.
- `private` - The object is private, the value will be true. Otherwise the value will be false.
- `create_time` - The date and time the object was initially created.
- `change_time` - The date and time of the last object change.
- `description` - Description of the Template.
- `usage_in_minutes` - Total minutes the object has been running.

- `capacity` - The capacity of a storage/ISO Image/ISO Image/snapshot in GB.
- `current_price` - Defines the price for the current period since the last bill.
- `labels` - List of labels.

» gridscale_firewall

Provides a firewall resource. This can be used to create, modify and delete firewalls.

» Example Usage

```
resource "gridscale_firewall" "foo" {
  name   = "example-firewall"
  rules_v4_in {
    order = 0
    protocol = "tcp"
    action = "drop"
    dst_port = "20:80"
    comment = "some comments"
  }
  rules_v6_in {
    order = 0
    protocol = "tcp"
    action = "drop"
    dst_port = "2000:3000"
    comment = "some comments"
  }
}
```

» Argument Reference

The following arguments are supported:

**Note: 'Optional'* means there is at least 1 rule in the firewall. Otherwise, an error will be returned.

- `name` - (Required) The human-readable name of the object. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `rules_v4_in` - (Optional*) Firewall template rules for inbound traffic - covers ipv4 addresses.

  - `order` - (Required) The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - (Required) This defines what the firewall will do. Either accept or drop.
  - `protocol` - (Required) Either 'udp' or 'tcp'.
  - `dst_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - (Optional) Comment.

- `rules_v4_out` - (Optional*) Firewall template rules for outbound traffic - covers ipv4 addresses.

  - `order` - (Required) The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - (Required) This defines what the firewall will do. Either accept or drop.
  - `protocol` - (Required) Either 'udp' or 'tcp'.
  - `dst_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - (Optional) Comment.

- `rules_v6_in` - (Optional*) Firewall template rules for inbound traffic - covers ipv6 addresses.

  - `order` - (Required) The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - (Required) This defines what the firewall will do. Either accept or drop.
  - `protocol` - (Required) Either 'udp' or 'tcp'.
  - `dst_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - (Optional) Comment.

- `rules_v6_out` - (Optional*) Firewall template rules for outbound traffic - covers ipv6 addresses.

  - `order` - (Required) The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - (Required) This defines what the firewall will do. Either accept or drop.
  - `protocol` - (Required) Either 'udp' or 'tcp'.
  - `dst_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - (Optional) A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - (Optional) Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - (Optional) Comment.

- `labels` - (Optional) List of labels in the format ["label1", "label2"].

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the firewall.
- `name` - The name of the firewall.
- `rules_v4_in` - Firewall template rules for inbound traffic - covers ipv4 addresses.
  - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - This defines what the firewall will do. Either accept or drop.
  - `protocol` - Either 'udp' or 'tcp'.
  - `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - Comment.
- `rules_v4_out` - Firewall template rules for outbound traffic - covers ipv4 addresses.
  - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - This defines what the firewall will do. Either accept or drop.
  - `protocol` - Either 'udp' or 'tcp'.
  - `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - Comment.
- `rules_v6_in` - Firewall template rules for inbound traffic - covers ipv6 addresses.
  - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either

allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - This defines what the firewall will do. Either accept or drop.
  - `protocol` - Either 'udp' or 'tcp'.
  - `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - Comment.
- `rules_v6_out` - Firewall template rules for outbound traffic - covers ipv6 addresses.
  - `order` - The order at which the firewall will compare packets against its rules, a packet will be compared against the first rule, it will either allow it to pass or block it and it won t be matched against any other rules. However, if it does no match the rule, then it will proceed onto rule 2. Packets that do not match any rules are blocked by default.
  - `action` - This defines what the firewall will do. Either accept or drop.
  - `protocol` - Either 'udp' or 'tcp'.
  - `dst_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_port` - A Number between 1 and 65535, port ranges are separated by a colon for FTP.
  - `src_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `dst_cidr` - Either an IPv4/6 address or and IP Network in CIDR format. If this field is empty then this service has access to all IPs.
  - `comment` - Comment.
- `network` - The information about networks which are related to this firewall.
  - `object_uuid` - The object UUID or id of the firewall.
  - `object_name` - Name of the firewall.
  - `network_uuid` - The object UUID or id of the network.
  - `network_name` - Name of the network.
  - `create_time` - The date and time the object was initially created.
- `location_name` - The human-readable name of the location. It supports the full UTF-8 charset, with a maximum of 64 characters.
- `status` - Status indicates the status of the object.
- `private` - The object is private, the value will be true. Otherwise the value will be false.
- `create_time` - The date and time the object was initially created.
- `change_time` - The date and time of the last object change.

- `description` - Description of the firewall.
- `labels` - List of labels.