» Data Source: ncloud_regions

Gets a list of available regions.

» Example Usage

```
data "ncloud_regions" "regions" {}
```

» Argument Reference

The following arguments are supported:

- `code` - (Optional) region code for filtering
- `output_file` - (Optional) The name of file that can save data source after running `terraform plan`.

» Attributes Reference

- `regions` - A List of region
  - `region_no` - Region number
  - `region_code` - Region code
  - `region_name` - Region name

» Data Source: ncloud_zones

Gets a list of available zones.

» Example Usage

```
data "ncloud_zones" "zones" {}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `output_file` - (Optional) The name of file that can save data source after running `terraform plan`.

» Attributes Reference

- `zones` - A List of region
  - `zone_no` - Zone number
  - `zone_code` - Zone code
  - `zone_name` - Zone name
  - `zone_description` - Zone description
  - `region_no` - Region number

» Data Source: ncloud_server_image

To create a server instance (VM), you should select a server image. This data source get a server image.

» Example Usage

- Filter by product name

```
data "ncloud_server_image" "image" {
  product_name_regex = "^Windows Server 2012(.*)"
}
```

- Filter by product type

```
data "ncloud_server_image" "image" {
  product_type_code = "WINNT"
}
```

» Argument Reference

The following arguments are supported:

- `product_name_regex` - (Optional) A regex string to apply to the server image list returned by ncloud.
- `exclusion_product_code` - (Optional) Product code you want to exclude from the list.
- `product_code` - (Optional) Product code you want to view on the list. Use this when searching for 1 product.
- `product_type` - (Optional) Product type code
- `platform_type_code_list` - (Optional) Values required for identifying platforms in list-type. The available values are as follows: Linux 32Bit(LNX32) | Linux 64Bit(LNX64) | Windows 32Bit(WND32) | Windows 64Bit(WND64) | Ubuntu Desktop 64Bit(UBD64) | Ubuntu Server 64Bit(UBS64)

- `block_storage_size` - (Optional) Block storage size.
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `infra_resource_detail_type_code` - (Optional) infra resource detail type code.

» Attributes Reference

- `product_name` - Product name
- `product_description` - Product description
- `infra_resource_type` - Infra resource type code
- `cpu_count` - CPU count
- `memory_size` - Memory size
- `base_block_storage_size` - Base block storage size
- `platform_type` - Platform type code
- `os_information` - OS Information
- `add_block_storage_size` - Additional block storage size

» Data Source: ncloud_server_images

To create a server instance (VM), you should select a server image. This data source gets a list of server images.

» Example Usage

```
data "ncloud_server_images" "all" {
  output_file = "server_images.json"
}
```

» Argument Reference

The following arguments are supported:

- `product_name_regex` - (Optional) A regex string to apply to the server image list returned by ncloud.
- `exclusion_product_code` - (Optional) Product code you want to exclude from the list.
- `product_code` - (Optional) Product code you want to view on the list. Use this when searching for 1 product.
- `platform_type_code_list` - (Optional) Values required for identifying platforms in list-type. The available values are as follows: Linux

32Bit(LNX32) | Linux 64Bit(LNX64) | Windows 32Bit(WND32) | Windows 64Bit(WND64) | Ubuntu Desktop 64Bit(UBD64) | Ubuntu Server 64Bit(UBS64)
- `block_storage_size` - (Optional) Block storage size.
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `infra_resource_detail_type_code` - (Optional) infra resource detail type code.
- `output_file` - (Optional) The name of file that can save data source after running `terraform plan`.

» Attributes Reference

- `server_images` - A List of server image product code

» Data Source: ncloud_server_product

ou should select a server product (server specification) to create a server instance (VM). To this end, we provide data source by which you can search a server product.

» Example Usage

```
data "ncloud_server_product" "product" {
    server_image_product_code = "SPSW0LINUX000032"
}
```

» Argument Reference

The following arguments are supported:

- `server_image_product_code` - (Required) You can get one from `data ncloud_server_images`. This is a required value, and each available server's specification varies depending on the server image product.
- `product_name_regex` - (Optional) A regex string to apply to the Server Product list returned.
- `exclusion_product_code` - (Optional) Enter a product code to exclude from the list.
- `product_code` - (Optional) Enter a product code to search from the list. Use it for a single search.
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.

- `zone` - (Optional) Zone code. You can decide a zone where servers are created. You can decide which zone the product list will be requested at. default : Select the first Zone in the specific region Get available values using the data source `ncloud_zones`.
- `internet_line_type_code` - (Optional) Internet line code. PUBLC(Public), GLBL(Global)

» Attributes Reference

- `product_name` - Product name
- `product_type` - Product type code
- `product_description` - Product description
- `infra_resource_type` - Infra resource type code
- `cpu_count` - CPU count
- `memory_size` - Memory size
- `base_block_storage_size` - Base block storage size
- `platform_type` - Platform type code
- `os_information` - OS Information
- `add_block_storage_size` - Additional block storage size

» Data Source: ncloud_server_products

You should select a server product (server specification) to create a server instance (VM). To this end, we provide data source by which you can search a server product.

» Example Usage

```
data "ncloud_server_products" "all" {
  # server_image_product_code: You can get one from `data ncloud_server_images`
  server_image_product_code = "SPSW0LINUX000032"
}
```

» Argument Reference

The following arguments are supported:

- `server_image_product_code` - (Required) You can get one from `data ncloud_server_images`. This is a required value, and each available server's specification varies depending on the server image product.
- `product_name_regex` - (Optional) A regex string to apply to the Server Product list returned.

- `exclusion_product_code` - (Optional) Enter a product code to exclude from the list.
- `product_code` - (Optional) Enter a product code to search from the list. Use it for a single search.
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `zone` - (Optional) Zone code. You can decide a zone where servers are created. You can decide which zone the product list will be requested at. default : Select the first Zone in the specific region Get available values using the data source `ncloud_zones`.
- `internet_line_type_code` - (Optional) Internet line code. PUBLC(Public), GLBL(Global)

» Attributes Reference

- `server_products` - A List of server product code

» Data Source: ncloud_member_server_image

Gets a member server image.

» Example Usage

```
data "ncloud_member_server_image" "test" {
}
```

» Argument Reference

The following arguments are supported:

- `name_regex` - (Optional) A regex string to apply to the member server image list returned by ncloud
- `no_list` - (Optional) List of member server images to view
- `platform_type_code_list` - (Optional) List of platform codes of server images to view. Linux 32Bit (`LNX32`) | Linux 64Bit (`LNX64`) | Windows 32Bit (`WND32`) | Windows 64Bit (`WND64`) | Ubuntu Desktop 64Bit (`UBD64`) | Ubuntu Server 64Bit (`UBS64`)
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: `KR` region.

» Attributes Reference

- `no` - Member server image no
- `name` - Member server image name
- `description` - Member server image description
- `original_server_instance_no` - Original server instance no
- `original_server_product_code` - Original server product code
- `original_server_name` - Original server name
- `original_base_block_storage_disk_type` - Original base block storage disk type
- `original_server_image_product_code` - Original server image product code
- `original_os_information` - Original os information
- `original_server_image_name` - Original server image name
- `status_name` - Member server image status name
- `status` - Member server image status
- `operation` - Member server image operation
- `platform_type` - Member server image platform type
- `region` - Region info
- `block_storage_total_rows` - Member server image block storage total rows
- `block_storage_total_size` - Member server image block storage total size

» Data Source: ncloud_member_server_images

Gets a list of member server images.

» Example Usage

```
data "ncloud_member_server_images" "member_server_images" {}
```

» Argument Reference

The following arguments are supported:

- `name_regex` - (Optional) A regex string to apply to the member server image list returned by ncloud
- `no_list` - (Optional) List of member server images to view
- `platform_type_code_list` - (Optional) List of platform codes of server images to view. Linux 32Bit (LNX32) | Linux 64Bit (LNX64) | Windows 32Bit (WND32) | Windows 64Bit (WND64) | Ubuntu Desktop 64Bit (UBD64) | Ubuntu Server 64Bit (UBS64)

- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `output_file` - (Optional) The name of file that can save data source after running `terraform plan`.

» Attributes Reference

- `member_server_images` - A list of Member server image no

» Data Source: ncloud_port_forwarding_rule

Get a port forwarding rule. When a server is created for the first time, a public IP address for port forwarding is given per account.

» Example Usage

```
data "ncloud_port_forwarding_rule" "test" {
  port_forwarding_external_port = "4088"
}
```

ncloud_nas_volume

» Argument Reference

The following arguments are supported:

- `internet_line_type_code` - (Optional) Internet line code. PUBLC(Public), GLBL(Global)
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `zone` - (Optional) Zone code. You can decide a zone where servers are created. You can decide which zone the product list will be requested at. default : Select the first Zone in the specific region Get available values using the data source `ncloud_zones`.
- `server_instance_no` - Filter by server instance number
- `port_forwarding_internal_port` - (Optional) Port forwarding internal port.
- `port_forwarding_external_port` - Port forwarding external port.

» Attributes Reference

- `port_forwarding_configuration_no` - Port forwarding configuration number
- `port_forwarding_public_ip` - Port forwarding public ip

» Data Source: ncloud_port_forwarding_rules

Gets a list of port forwarding rules. When a server is created for the first time, a public IP address for port forwarding is given per account.

» Example Usage

```
data "ncloud_port_forwarding_rules" "rules" {
  zone_code = "KR-1"
}
```

» Argument Reference

The following arguments are supported:

- `internet_line_type_code` - (Optional) Internet line code. PUBLC(Public), GLBL(Global)
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `zone` - (Optional) Zone code. You can decide a zone where servers are created. You can decide which zone the product list will be requested at. default : Select the first Zone in the specific region Get available values using the data source `ncloud_zones`.
- `port_forwarding_internal_port` - (Optional) Port forwarding internal port.
- `output_file` - (Optional) The name of file that can save data source after running `terraform plan`.

» Attributes Reference

- `port_forwarding_configuration_no` - Port forwarding configuration number
- `port_forwarding_public_ip` - Port forwarding public ip
- `port_forwarding_rule_list` - Port forwarding rule list
  - `server_instance_no` - Server instance number
  - `port_forwarding_external_port` - Port forwarding external port.

  - `port_forwarding_internal_port` - Port forwarding internal port.

» Data Source: ncloud_public_ip

Get public IP instance.

» Example Usage

```
data "ncloud_public_ip" "public_ip" {
  sorted_by = "publicIp"
  sorting_order = "ascending"
}
```

» Argument Reference

The following arguments are supported:

- `internet_line_type` - (Optional) Internet line type code. `PUBLC` (Public), `GLBL` (Global)
- `is_associated` - (Optional) Indicates whether the public IP address is associated or not.
- `instance_no_list` - (Optional) List of public IP instance numbers to get.
- `list` - (Optional) List of public IP addresses to get.
- `search_filter_name` - (Optional) `publicIp` (Public IP) | `associatedServerName` (Associated server name)
- `search_filter_value` - (Optional) Filter value to search
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `zone` - (Optional) Zone code. You can filter the list of public IP instances by zones. All the public IP addresses in the zone of the region will be selected if the filter is not specified. Get available values using the data source `ncloud_zones`.
- `sorted_by` - (Optional) The column based on which you want to sort the list.
- `sorting_order` - (Optional) Sorting order of the list. `ascending` (Ascending) | `descending` (Descending) [case insensitive]. Default: `ascending` Ascending

» Attributes Reference

- `instance_no` - Public IP instance number
- `public_ip` - Public IP

- `description` - Public IP description
- `create_date` - Creation date of the public ip
- `internet_line_type` - Internet line type
- `instance_status_name` - Public IP instance status name
- `instance_status` - Public IP instance status
- `instance_operation` - Public IP instance operation
- `kind_type` - Public IP kind type
- `server_instance` - Associated server instance
  - `server_instance_no` - Associated server instance number
  - `server_name` - Associated server name
  - `create_date` - Creation date of the server instance

» Data Source: ncloud_nas_volume

Get NAS volume instance

» Example Usage

```
data "ncloud_nas_volume" "vol" {}
```

» Argument Reference

The following arguments are supported:

- `volume_allotment_protocol_type_code` - (Optional) Volume allotment protocol type code. All volume instances will be selected if the filter is not specified. (`NFS` | `CIFS`)
- `is_event_configuration` - (Optional) Indicates whether the event is set. All volume instances will be selected if the filter is not specified. (`true` | `false`)
- `is_snapshot_configuration` - (Optional) Indicates whether a snapshot volume is set. All volume instances will be selected if the filter is not specified. (`true` | `false`)
- `no_list` - (Optional) List of nas volume instance numbers.
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: `KR` region.
- `zone` - (Optional) Zone code. Get available values using the data source `ncloud_zones`.

» Attributes Reference

- `instance_no` - NAS volume instance number

- `volume_name` - Volume name
- `instance_status` - NAS volume instance status code
- `create_date` - Creation date of the NAS Volume instance
- `volume_total_size` - Volume total size
- `volume_size` - Volume size
- `volume_use_size` - Volume use size
- `volume_use_ratio` - Volume use ratio
- `snapshot_volume_size` - Snapshot volume size
- `snapshot_volume_use_size` - Snapshot volume use size
- `snapshot_volume_use_ratio` - Snapshot volume use ratio
- `instance_custom_ip_list` - NAS volume instance custom IP list
- `description` - NAS volume description

» Data Source: ncloud_nas_volumes

Gets a list of NAS volume instances.

» Example Usage

```
data "ncloud_nas_volumes" "nas_volumes" {}
```

» Argument Reference

The following arguments are supported:

- `volume_allotment_protocol_type_code` - (Optional) Volume allotment protocol type code. All volume instances will be selected if the filter is not specified. (`NFS` | `CIFS`)
- `is_event_configuration` - (Optional) Indicates whether the event is set. All volume instances will be selected if the filter is not specified. (`true` | `false`)
- `is_snapshot_configuration` - (Optional) Indicates whether a snapshot volume is set. All volume instances will be selected if the filter is not specified. (`true` | `false`)
- `no_list` - (Optional) List of nas volume instance numbers.
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `zone` - (Optional) Zone code. Get available values using the data source `ncloud_zones`.
- `output_file` - (Optional) The name of file that can save data source after running `terraform plan`.

» Attributes Reference

- `nas_volumes` - A list of NAS Volume Instance no

» Data Source: ncloud_access_control_group

When creating a server instance (VM), you can add an access control group (ACG) that you specified to set firewalls. `ncloud_access_control_group` provides details about a specific access control group (ACG) information.

» Example Usage

- Filter by ACG name

```
data "ncloud_access_control_group" "test" {
    # filter by ACG name
    name = "acg-name"
}
```

» Argument Reference

The following arguments are supported:

- `configuration_no` - (Optional) List of ACG configuration numbers you want to get
- `name` - (Optional) Name of the ACG you want to get
- `is_default_group` - (Optional) Indicates whether to get default group only

Conditional: Requires `configuration_no` or`name` or `is_default_group`.

» Attributes Reference

- `description` - ACG description

» Data Source: ncloud_access_control_groups

When creating a server instance (VM), you can add an access control group (ACG) that you specified to set firewalls. This data source gets a list of access control groups necessary to set firewalls.

» Example Usage

```
data "ncloud_access_control_groups" "acg" {}
```

» Argument Reference

The following arguments are supported:

- `configuration_no_list` - (Optional) List of ACG configuration numbers you want to get
- `is_default_group` - (Optional) Indicates whether to get default groups only
- `name` - (Optional) Name of the ACG you want to get
- `output_file` - (Optional) The name of file that can save data source after running `terraform plan`.

» Attributes Reference

- `access_control_groups` - A List of access control group configuration_no.

» Data Source: ncloud_access_control_rule

Access configuration rule you want to get

» Example Usage

```
data "ncloud_access_control_rule" "test" {
  is_default_group = "true"
  destination_port = "22"
}
```

» Argument Reference

The following arguments are supported:

- `access_control_group_configuration_no` - (Optional) Access control group number to search
- `access_control_group_name` - (Optional) Access control group name to search
- `is_default_group` - (Optional) Whether default group

- `source_name_regex` - (Optional) A regex string to apply to the source access control rule list returned by ncloud

» Attributes Reference

- `source_ip` - Source IP
- `destination_port` - Destination Port
- `protocol_type_code` - Protocol type code
- `configuration_no` - Access control rule configuration no
- `protocol_type` - Protocol type code
- `source_configuration_no` - Source access control rule configuration no
- `source_name` - Source access control rule name
- `description` - Access control rule description

» Data Source: ncloud_access_control_rules

List of access configuration rules you want to get

» Example Usage

```
data "ncloud_access_control_rules" "test" {
    // access_control_group_configuration_no : You can get one from `ncloud_access_control_g
    //      or `ncloud_access_control_groups`
    access_control_group_configuration_no = "123"
}
```

» Argument Reference

The following arguments are supported:

- `access_control_group_configuration_no` - (Required) Access control group configuration number to search
- `source_name_regex` - (Optional) A regex string to apply to the ACG rule list returned by ncloud
- `output_file` - (Optional) The name of file that can save data source after running `terraform plan`.

» Attributes Reference

- `access_control_rules` - A list of access control rules configuration no

» Data Source: ncloud_root_password

Gets the password of a root account with the server's login key.

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

» Example Usage

```
data "ncloud_root_password" "default" {
  server_instance_no = "server_instance_no" # ${ncloud_server.vm.id}
  private_key = "private_key" # ${ncloud_login_key.key.private_key}
}
```

» Argument Reference

The following arguments are supported:

- `server_instance_no` - (Required) Server instance number
- `private_key` - (Required) Server's login key (auth key)

» Attributes Reference

- `root_password` - password of a root account

» ncloud_server

Provides a ncloud server instance resource.

» Example Usage

```
resource "ncloud_server" "server" {
    name = "tf-test-vm1"
    server_image_product_code = "SPSW0LINUX000032"
    server_product_code = "SPSVRSTAND000004"

    tag_list = [
        {
            tag_key   = "samplekey1"
            tag_value = "samplevalue1"
        },
```

```
        {
            tag_key   = "samplekey2"
            tag_value = "samplevalue2"
        },
    ]
}
```

» Argument Reference

The following arguments are supported:

- `server_image_product_code` - (Optional) Server image product code to determine which server image to create. It can be obtained through `data ncloud_server_images`. You are required to select one among two parameters: server image product code (server_image_product_code) and member server image number(member_server_image_no).
- `server_product_code` - (Optional) Server product code to determine the server specification to create. It can be obtained through the getServerProductList action. Default : Selected as minimum specification. The minimum standards are 1. memory 2. CPU 3. basic block storage size 4. disk type (NET,LOCAL)
- `member_server_image_no` - (Optional) Required value when creating a server from a manually created server image. It can be obtained through the getMemberServerImageList action.
- `name` - (Optional) Server name to create. default: Assigned by ncloud
- `description` - (Optional) Server description to create
- `login_key_name` - (Optional) The login key name to encrypt with the public key. Default : Uses the most recently created login key name
- `is_protect_server_termination` - (Optional) You can set whether or not to protect return when creating. default : false
- `internet_line_type` - (Optional) Internet line identification code. PUBLC(Public), GLBL(Global). default : PUBLC(Public)
- `fee_system_type_code` - (Optional) A rate system identification code. There are time plan(MTRAT) and flat rate (FXSUM). Default : Time plan(MTRAT)
- `zone` - (Optional) Zone code. You can determine the ZONE where the server will be created. Default : Assigned by NAVER Cloud Platform. Get available values using the data source `ncloud_zones`.
- `access_control_group_configuration_no_list` - (Optional) You can set the ACG created when creating the server. ACG setting number can be obtained through the getAccessControlGroupList action. Default : Default ACG number
- `user_data` - (Optional) The server will execute the user data script set by the user at first boot. To view the column, it is returned only when viewing the server instance.

- `raid_type_name` - (Optional) Raid Type Name.
- `tag_list` - (Optional) Server instance tag list.
  - `tag_key` - (Required) Instance tag key
  - `tag_value` - (Required) Instance tag value

» Attributes Reference

- `id` - The instance ID.
- `instance_no` - Server instance number
- `cpu_count` - number of CPUs
- `memory_size` - The size of the memory in bytes.
- `base_block_storage_size` - The size of base block storage in bytes
- `platform_type` - Platform type code
- `is_fee_charging_monitoring` - Fee charging monitoring
- `public_ip` - Public IP
- `private_ip` - Private IP
- `server_image_name` - Server image name
- `instance_status` - Server instance status code
- `instance_status_name` - Server instance status name
- `instance_operation` - Server instance operation code
- `port_forwarding_public_ip` - Port forwarding public ip
- `port_forwarding_external_port` - Port forwarding external port
- `port_forwarding_internal_port` - Port forwarding internal port
- `region` - Region code
- `base_block_storage_disk_type` - Base block storage disk type code
- `base_block_storage_disk_detail_type` - Base block storage disk detail type code

» ncloud_block_storage

Provides a ncloud block storage resource.

» Example Usage

```
resource "ncloud_block_storage" "storage" {
    server_instance_no = "812345"
    name = "tf-test-storage1"
    size = "10"
}
```

» Argument Reference

The following arguments are supported:

- `size` - (Required) Enter a block storage size to ceate. You can enter by the unit of GB. Up to 1000GB you can enter.
- `server_instance_no` - (Required) Server instance No. to attach. It is required and you can get a server instance No. by calling getServerInstanceList.
- `name` - (Optional) Block storage name to create default : Ncloud configures it by itself.
- `description` - (Optional) Block storage descriptions
- `disk_detail_type` - (Optional) You can choose a disk detail type code of HDD and SSD. default : HDD

» Attributes Reference

- `instance_no` - Block storage instance no
- `server_name` - Server name
- `type` - Block storage type code
- `device_name` - Device name
- `product_code` - Block storage product code
- `instance_status` - Block storage instance status code
- `instance_operation` - Block storage instance operation
- `instance_status_name` - Block storage instance status name
- `create_date` - Creation date of the block storage
- `disk_type` - Disk type code

» ncloud_block_storage

Provides a ncloud block storage snapshot resource.

» Example Usage

```
resource "ncloud_block_storage_snapshot" "snapshot" {
    block_storage_instance_no = "812345"
    name = "tf-test-snapshot1"
    description = "Terraform test snapshot1"
}
```

» Argument Reference

The following arguments are supported:

- `block_storage_instance_no` - (Required) Block storage instance No for creating snapshot.
- `name` - (Optional) Block storage snapshot name to create. default : Ncloud assigns default values.
- `description` - (Optional) Descriptions on a snapshot to create.

» Attributes Reference

- `instance_no` - Block Storage Snapshot Instance Number
- `volume_size` - Block Storage Snapshot Volume Size
- `original_block_storage_instance_no` - Original Block Storage Instance Number
- `original_block_storage_name` - Original Block Storage Name
- `instance_status` - Block Storage Snapshot Instance Status code
- `instance_status_name` - Block Storage Snapshot Instance Status Name
- `instance_operation` - Block Storage Snapshot Instance Operation code
- `create_date` - Creation date of the block storage snapshot instance
- `server_image_product_code` - Server Image Product Code
- `os_information` - OS Information

» ncloud_public_ip

Provides a ncloud public IP instance resource.

» Example Usage

```
resource "ncloud_public_ip" "public_ip" {
  server_instance_no = "812345"
}
```

» Argument Reference

The following arguments are supported:

- `server_instance_no` - (Optional) Server instance No. to assign after creating a public IP. You can get one by calling getPublicIpTargetServerInstanceList.
- `description` - (Optional) Public IP description.

- `internet_line_type` - (Optional) Internet line code. PUBLC(Public), GLBL(Global)
- `zone` - (Optional) Zone code. You can decide a zone where servers are created. You can decide which zone the product list will be requested at. default : Select the first Zone in the specific region Get available values using the data source `ncloud_zones`.

» Attributes Reference

- `instance_no` - Public IP instance No.
- `public_ip` - Public IP Address.
- `create_date` - Creation date of the public IP instance
- `instance_status_name` - Public IP instance status name
- `instance_status` - Public IP instance status code
- `instance_operation` - Public IP instance operation code
- `kind_type` - Public IP kind type

» ncloud_login_key

Provides a ncloud login key resource.

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

» Example Usage

```
resource "ncloud_login_key" "loginkey" {
  key_name = "sample key name"
}
```

» Argument Reference

The following arguments are supported:

- `key_name` - (Required) Key name to generate. If the generated key name exists, an error occurs.

» Attributes Reference

- `private_key` - Generated private key
- `fingerprint` - Fingerprint of the login key
- `create_date` - Creation date of the login key

» ncloud_nas_volume

Provides a ncloud NAS volume.

» Example Usage

```
resource "ncloud_nas_volume" "test" {
    volume_name_postfix            = "vol"
    volume_size                    = "600"
    volume_allotment_protocol_type = "NFS"
}
```

» Argument Reference

The following arguments are supported:

- `volume_name_postfix` - (Required) Name of a NAS volume to create. Enter a volume name that can be 3-20 characters in length after the name already entered for user identification.
- `volume_size` - (Required) Enter the nas volume size to be created. You can enter in GiB.
- `volume_allotment_protocol_type` - (Required) Volume allotment protocol type code. `NFS` | `CIFS` `NFS`: You can mount the volume in a Linux server such as CentOS and Ubuntu. `CIFS`: You can mount the volume in a Windows server.
- `server_instance_no_list` - (Optional) List of server instance numbers for which access to NFS is to be controlled
- `custom_ip_list` - (Optional) To add a server of another account to the NAS volume, enter a private IP address of the server.
- `cifs_user_name` - (Optional) CIFS user name. The ID must contain a combination of English alphabet and numbers, which can be 6-20 characters in length.
- `cifs_user_password` - (Optional) CIFS user password. The password must contain a combination of at least 2 English letters, numbers and special characters, which can be 8-14 characters in length.
- `description` - (Optional) NAS volume description
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `zone` - (Optional) Zone code. Zone in which you want to create a NAS volume. Default: The first zone of the region. Get available values using the data source `ncloud_zones`.

» Attributes Reference

- `volume_name` - NAS volume name.
- `instance_status` - NAS Volume instance status code
- `create_date` - Creation date of the NAS volume
- `volume_total_size` - Volume total size, in GiB
- `volume_use_size` - Volume use size, in GiB
- `volume_use_ratio` - Volume use ratio
- `snapshot_volume_size` - Snapshot volume size, in GiB
- `snapshot_volume_use_size` - Snapshot volume use size
- `snapshot_volume_use_ratio` - Snapshot volume use ratio
- `is_snapshot_configuration` - Indicates whether a snapshot volume is set.
- `is_event_configuration` - Indicates whether the event is set.
- `instance_custom_ip_list` - NAS volume instance custom IP list

» ncloud_port_forwarding_rule

Provides a ncloud port forwarding rule resource.

» Example Usage

```
resource "ncloud_port_forwarding_rule" "rule" {
   port_forwarding_configuration_no = "1222"
   server_instance_no = "812345"
   port_forwarding_external_port = "2022"
   port_forwarding_internal_port = "22"
}
```

» Argument Reference

The following arguments are supported:

- `server_instance_no` - (Required) Server instance number for which port forwarding is set
- `port_forwarding_external_port` - (Required) External port for port forwarding
- `port_forwarding_internal_port` - (Required) Internal port for port forwarding. Only the following ports are available. [Linux: `22` | Windows: `3389`]
- `port_forwarding_configuration_no` - (Optional) Port forwarding configuration number. You can get by calling `data ncloud_port_forwarding_rules`

» Attributes Reference

- `port_forwarding_public_ip` - Port forwarding Public IP
- `zone` - Zone code

» ncloud_load_balancer

Provides a ncloud load balancer instance resource.

» Example Usage

```
resource "ncloud_load_balancer" "lb" {
  name           = "tftest_lb"
  algorithm_type = "SIPHS"
  description    = "tftest_lb description"

  rule_list = [
    {
      protocol_type        = "HTTP"
      load_balancer_port   = 80
      server_port          = 80
      l7_health_check_path = "/monitor/l7check"
    },
    {
      protocol_type        = "HTTPS"
      load_balancer_port   = 443
      server_port          = 443
      l7_health_check_path = "/monitor/l7check"
      certificate_name     = "cert"
    },
  ]

  server_instance_no_list = ["812345", "812346"]
  internet_line_type      = "PUBLC"
  network_usage_type      = "PBLIP"

  region = "KR"
}
```

» Argument Reference

The following arguments are supported:

- `rule_list` - (Required) Load balancer rules.
  - `protocol_type` - (Required) Protocol type code of load balancer rules. The following codes are available. [HTTP | HTTPS | TCP | SSL]
  - `load_balancer_port` - (Required) Load balancer port of load balancer rules
  - `server_port` - (Required) Server port of load balancer rules
  - `l7_health_check_path` - Health check path of load balancer rules. Required when the `protocol_type` is HTTP/HTTPS.
  - `certificate_name` - Load balancer SSL certificate name. Required when the `protocol_type` value is SSL/HTTPS.
  - `proxy_protocol_use_yn` - (Optional) Use 'Y' if you want to check client IP addresses by enabling the proxy protocol while you select TCP or SSL.
- `name` - (Optional) Name of a load balancer instance. Default: Automatically specified by Ncloud.
- `algorithm_type` - (Optional) Load balancer algorithm type code. The available algorithms are as follows: [ROUND ROBIN (RR) | LEAST_CONNECTION (LC)]. Default: ROUND ROBIN (RR)
- `description` - (Optional) Description of a load balancer instance.
- `server_instance_no_list` - (Optional) List of server instance numbers to be bound to the load balancer
- `internet_line_type` - (Optional) Internet line identification code. PUBLC(Public), GLBL(Global). default : PUBLC(Public)
- `network_usage_type` - (Optional) Network usage identification code. PBLIP(PublicIP), PRVT(PrivateIP). default : PBLIP(PublicIP)
- `region` - (Optional) Region code. Get available values using the data source `ncloud_regions`. Default: KR region.
- `zone` - (Optional) Zone code. Zone in which you want to create a NAS volume. Default: The first zone of the region. Get available values using the data source `ncloud_zones`.

» Attributes Reference

- `instance_no` - Load balancer instance No
- `virtual_ip` - Virtual IP address
- `create_date` - Creation date of the load balancer instance
- `domain_name` - Domain name
- `instance_status_name` - Load balancer instance status name
- `instance_status` - Load balancer instance status code
- `instance_operation` - Load balancer instance operation code
- `is_http_keep_alive` - Http keep alive value [true | false]
- `connection_timeout` - Connection timeout
- `load_balanced_server_instance_list` - Load balanced server instance list

» load_balancer_ssl_certificate

Provides a ncloud load balancer ssl certificate resource.

» Example Usage

```
resource "ncloud_load_balancer_ssl_certificate" "cert" {
  certificate_name      = "tftest_ssl_cert"
  privatekey            = "${file("lbtest.privateKey")}"
  publickey_certificate = "${file("lbtest.crt")}"
  certificate_chain     = "${file("lbtest.chain")}"
}
```

» Argument Reference

The following arguments are supported:

- `certificate_name` - (Required) Name of a certificate
- `privatekey` - (Required) Private key for a certificate
- `publickey_certificate` - (Required) Public key for a certificate
- `certificate_chain` - (Optional) Chainca certificate (Required if the certificate is issued with a chainca)