» Data Source: azurestack_network_interface

Use this data source to access the properties of an Azure Network Interface.

» Example Usage

```
data "azurestack_network_interface" "test" {
  name                = "acctest-nic"
  resource_group_name  = "networking"
}

output "network_interface_id" {
  value = "${data.azurestack_network_interface.test.id}"
}
```

» Argument Reference

- `name` - (Required) Specifies the name of the Network Interface.
- `resource_group_name` - (Required) Specifies the name of the resource group the Network Interface is located in.

» Attributes Reference

- `applied_dns_servers` - List of DNS servers applied to the specified network interface.
- `dns_servers` - The list of DNS servers used by the specified network interface.
- `enable_ip_forwarding` - Indicate if IP forwarding is set on the specified network interface.
- `id` - The ID of the virtual network that the specified network interface is associated to.
- `internal_dns_name_label` - The internal dns name label of the specified network interface.
- `internal_fqdn` - The internal FQDN associated to the specified network interface.
- `ip_configuration` - The list of IP configurations associated to the specified network interface.
- `location` - The location of the specified network interface.
- `network_security_group_id` - The ID of the network security group associated to the specified network interface.
- `mac_address` - The MAC address used by the specified network interface.
- `private_ip_address` - The primary private ip address associated to the specified network interface.

- `private_ip_addresses` - The list of private ip addresses associates to the specified network interface.
- `tags` - List the tags assocatied to the specified network interface.
- `virtual_machine_id` - The ID of the virtual machine that the specified network interface is attached to.

» Data Source: azurestack_network_security_group

Use this data source to access the properties of a Network Security Group.

» Example Usage

```
data "azurestack_network_security_group" "test" {
  name                = "${azurestack_network_security_group.test.name}"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

output "location" {
  value = "${data.azurestack_network_security_group.test.location}"
}
```

» Argument Reference

- `name` - (Required) Specifies the Name of the Network Security Group.
- `resource_group_name` - (Required) Specifies the Name of the Resource Group within which the Network Security Group exists

» Attributes Reference

- `id` - The ID of the Network Security Group.
- `location` - The supported Azure location where the resource exists.
- `security_rule` - One or more `security_rule` blocks as defined below.
- `tags` - A mapping of tags assigned to the resource.

The `security_rule` block supports:

- `name` - The name of the security rule.
- `description` - The description for this rule.
- `protocol` - The network protocol this rule applies to.

- `source_port_range` - The Source Port or Range.
- `destination_port_range` - The Destination Port or Range.
- `source_address_prefix` - CIDR or source IP range or * to match any IP.
- `destination_address_prefix` - CIDR or destination IP range or * to match any IP.
- `access` - Is network traffic is allowed or denied?
- `priority` - The priority of the rule
- `direction` - The direction specifies if rule will be evaluated on incoming or outgoing traffic.

» Data Source: azurestack_resource_group

Use this data source to access the properties of an Azure resource group.

» Example Usage

```
data "azurestack_resource_group" "test" {
  name = "dsrg_test"
}

resource "azurestack_managed_disk" "test" {
  name                 = "managed_disk_name"
  location             = "${data.azurestack_resource_group.test.location}"
  resource_group_name  = "${data.azurestack_resource_group.test.name}"
  storage_account_type = "Standard_LRS"
  create_option        = "Empty"
  disk_size_gb         = "1"
}
```

» Argument Reference

- `name` - (Required) Specifies the name of the resource group.

NOTE: If the specified location doesn't match the actual resource group location, an error message with the actual location value will be shown.

» Attributes Reference

- `location` - The location of the resource group.
- `tags` - A mapping of tags assigned to the resource group.

» Data Source: azurestack_storage_account

Gets information about the specified Storage Account.

» Example Usage

```
data "azurestack_storage_account" "test" {
  name                 = "packerimages"
  resource_group_name  = "packer-storage"
}

output "storage_account_tier" {
  value = "${data.azurestack_storage_account.test.account_tier}"
}
```

» Argument Reference

- `name` - (Required) Specifies the name of the Storage Account
- `resource_group_name` - (Required) Specifies the name of the resource group the Storage Account is located in.

» Attributes Reference

- `id` - The ID of the Storage Account.
- `location` - The Azure location where the Storage Account exists
- `account_kind` - (Optional) Defines the Kind of account. Valid option is `Storage`. . Changing this forces a new resource to be created. Defaults to `Storage` currently as per Azure Stack Storage Differences
- `account_tier` - Defines the Tier of this storage account.
- `account_replication_type` - Defines the type of replication used for this storage account.
- `access_tier` - (Required for `BlobStorage` accounts) Defines the access tier for `BlobStorage` accounts. Valid options are `Hot` and `Cold`, defaults to `Hot`. - **`Currently Not Supported on Azure Stack`**

- `account_encryption_source` - The Encryption Source for this Storage Account.
- `custom_domain` - A `custom_domain` block as documented below.
- `tags` - A mapping of tags to assigned to the resource.
- `primary_location` - The primary location of the Storage Account.
- `secondary_location` - The secondary location of the Storage Account.
- `primary_blob_endpoint` - The endpoint URL for blob storage in the primary location.
- `secondary_blob_endpoint` - The endpoint URL for blob storage in the secondary location.
- `primary_queue_endpoint` - The endpoint URL for queue storage in the primary location.
- `secondary_queue_endpoint` - The endpoint URL for queue storage in the secondary location.
- `primary_table_endpoint` - The endpoint URL for table storage in the primary location.
- `secondary_table_endpoint` - The endpoint URL for table storage in the secondary location.
- `primary_file_endpoint` - The endpoint URL for file storage in the primary location.
- `primary_access_key` - The primary access key for the Storage Account.
- `secondary_access_key` - The secondary access key for the Storage Account.
- `primary_connection_string` - The connection string associated with the primary location
- `secondary_connection_string` - The connection string associated with the secondary location
- `primary_blob_connection_string` - The connection string associated with the primary blob location
- `secondary_blob_connection_string` - The connection string associated with the secondary blob location

---

- `custom_domain` supports the following:
- `name` - The Custom Domain Name used for the Storage Account.

» Data Source: azurestack_virtual_network

Use this data source to access the properties of an Azure Virtual Network.

» Example Usage

```
data "azurestack_virtual_network" "test" {
  name                 = "production"
  resource_group_name  = "networking"
}

output "virtual_network_id" {
  value = "${data.azurestack_virtual_network.test.id}"
}
```

» Argument Reference

- `name` - (Required) Specifies the name of the Virtual Network.
- `resource_group_name` - (Required) Specifies the name of the resource group the Virtual Network is located in.

» Attributes Reference

- `id` - The ID of the virtual network.
- `address_spaces` - The list of address spaces used by the virtual network.
- `dns_servers` - The list of DNS servers used by the virtual network.
- `subnets` - The list of name of the subnets that are attached to this virtual network.

» azurestack_resource_group

Creates a new resource group on Azure.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "testResourceGroup1"
  location = "West US"

  tags {
```

```
    environment = "Production"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the resource group. Must be unique on your Azure subscription.
- `location` - (Required) The location where the resource group should be created. For a list of all Azure locations, please consult this link or run `az account list-locations --output table`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

The following attributes are exported:

- `id` - The resource group ID.

» Import

Resource Groups can be imported using the `resource id`, e.g.

```
terraform import azurestack_resource_group.mygroup /subscriptions/00000000-0000-0000-0000-00
```

» azurestack_availability_set

Manages an availability set for virtual machines.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "resourceGroup1"
  location = "West US"
}

resource "azurestack_availability_set" "test" {
  name                = "acceptanceTestAvailabilitySet1"
  location            = "${azurestack_resource_group.test.location}"
```

```
  resource_group_name = "${azurestack_resource_group.test.name}"

  tags {
    environment = "Production"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the availability set. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the availability set. Changing this forces a new resource to be created.
- `location` - (Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
- `platform_update_domain_count` - (Optional) Specifies the number of update domains that are used. Defaults to 5.

NOTE: The number of Update Domains varies depending on which Azure Region you're using - a list can be found here.

- `platform_fault_domain_count` - (Optional) Specifies the number of fault domains that are used. Defaults to 3.

NOTE: The number of Fault Domains varies depending on which Azure Region you're using - a list can be found here.

- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

The following attributes are exported:

- `id` - The virtual Availability Set ID.

» Import

Availability Sets can be imported using the `resource id`, e.g.

```
terraform import azurestack_availability_set.group1 /subscriptions/00000000-0000-0000-0000-0
```

» azurestack_virtual_machine

Manages a virtual machine.

» Example Usage with Unmanaged Disks

```
resource "azurestack_resource_group" "test" {
  name     = "acctestrg"
  location = "West US"
}

resource "azurestack_virtual_network" "test" {
  name                = "acctvn"
  address_space       = ["10.0.0.0/16"]
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

resource "azurestack_subnet" "test" {
  name                 = "acctsub"
  resource_group_name  = "${azurestack_resource_group.test.name}"
  virtual_network_name = "${azurestack_virtual_network.test.name}"
  address_prefix       = "10.0.2.0/24"
}

resource "azurestack_network_interface" "test" {
  name                = "acctni"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"

  ip_configuration {
    name                          = "testconfiguration1"
    subnet_id                     = "${azurestack_subnet.test.id}"
    private_ip_address_allocation = "dynamic"
  }
}

resource "azurestack_storage_account" "test" {
  name                     = "accsa"
  resource_group_name      = "${azurestack_resource_group.test.name}"
  location                 = "${azurestack_resource_group.test.location}"
  account_tier             = "Standard"
  account_replication_type = "LRS"
```

```
  tags {
    environment = "staging"
  }
}

resource "azurestack_storage_container" "test" {
  name                  = "vhds"
  resource_group_name   = "${azurestack_resource_group.test.name}"
  storage_account_name  = "${azurestack_storage_account.test.name}"
  container_access_type = "private"
}

resource "azurestack_virtual_machine" "test" {
  name                  = "acctvm"
  location              = "${azurestack_resource_group.test.location}"
  resource_group_name   = "${azurestack_resource_group.test.name}"
  network_interface_ids = ["${azurestack_network_interface.test.id}"]
  vm_size               = "Standard_F2"

  # Uncomment this line to delete the OS disk automatically when deleting the VM
  # delete_os_disk_on_termination = true

  # Uncomment this line to delete the data disks automatically when deleting the VM
  # delete_data_disks_on_termination = true

  storage_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }

  storage_os_disk {
    name          = "myosdisk1"
    vhd_uri       = "${azurestack_storage_account.test.primary_blob_endpoint}${azurestack_st
    caching       = "ReadWrite"
    create_option = "FromImage"
  }

  # Optional data disks
  storage_data_disk {
    name          = "datadisk0"
    vhd_uri       = "${azurestack_storage_account.test.primary_blob_endpoint}${azurestack_st
    disk_size_gb  = "1023"
    create_option = "Empty"
    lun           = 0
```

```
  }

  os_profile {
    computer_name  = "hostname"
    admin_username = "testadmin"
    admin_password = "Password1234!"
  }

  os_profile_linux_config {
    disable_password_authentication = false
  }

  tags {
    environment = "staging"
  }
}
```

» Example Usage with Unmanaged Disks and Public IP

```
resource "azurestack_resource_group" "test" {
  name     = "acctestrg"
  # This is Azure Stack Region so it will be different per Azure Stack and should not be in
  location = "region1"
}

resource "azurestack_public_ip" "test" {
  name                         = "acceptanceTestPublicIp1"
  location                     = "${azurestack_resource_group.test.location}"
  resource_group_name          = "${azurestack_resource_group.test.name}"
  public_ip_address_allocation = "static"

  tags {
    environment = "Production"
  }
}

resource "azurestack_virtual_network" "test" {
  name                = "acctvn"
  address_space       = ["10.0.0.0/16"]
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

resource "azurestack_subnet" "test" {
  name                 = "acctsub"
```

```
  resource_group_name  = "${azurestack_resource_group.test.name}"
  virtual_network_name = "${azurestack_virtual_network.test.name}"
  address_prefix       = "10.0.2.0/24"
}

resource "azurestack_network_interface" "test" {
  name                = "acctni"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"

  ip_configuration {
    name                          = "testconfiguration1"
    subnet_id                     = "${azurestack_subnet.test.id}"
    private_ip_address_allocation = "dynamic"
    public_ip_address_id          = "${azurestack_public_ip.test.id}"
  }
}

resource "azurestack_storage_account" "test" {
  name                     = "accsa"
  resource_group_name      = "${azurestack_resource_group.test.name}"
  location                 = "${azurestack_resource_group.test.location}"
  account_tier             = "Standard"
  account_replication_type = "LRS"

  tags {
    environment = "staging"
  }
}

resource "azurestack_storage_container" "test" {
  name                  = "vhds"
  resource_group_name   = "${azurestack_resource_group.test.name}"
  storage_account_name  = "${azurestack_storage_account.test.name}"
  container_access_type = "private"
}

resource "azurestack_virtual_machine" "test" {
  name                  = "acctvm"
  location              = "${azurestack_resource_group.test.location}"
  resource_group_name   = "${azurestack_resource_group.test.name}"
  network_interface_ids = ["${azurestack_network_interface.test.id}"]
  vm_size               = "Standard_D2_v2"

  # Uncomment this line to delete the OS disk automatically when deleting the VM
  # delete_os_disk_on_termination = true
```

```
  # Uncomment this line to delete the data disks automatically when deleting the VM
  # delete_data_disks_on_termination = true

  storage_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }

  storage_os_disk {
    name          = "myosdisk1"
    vhd_uri       = "${azurestack_storage_account.test.primary_blob_endpoint}${azurestack_st
    caching       = "ReadWrite"
    create_option = "FromImage"
  }

  # Optional data disks
  storage_data_disk {
    name          = "datadisk0"
    vhd_uri       = "${azurestack_storage_account.test.primary_blob_endpoint}${azurestack_st
    disk_size_gb  = "1023"
    create_option = "Empty"
    lun           = 0
  }

  os_profile {
    computer_name  = "hostname"
    admin_username = "testadmin"
    admin_password = "Password1234!"
  }

  os_profile_linux_config {
    disable_password_authentication = false
  }

  tags {
    environment = "staging"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the virtual machine resource. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the virtual machine.
- `location` - (Required) Specifies the supported Azure Stack Region where the resource exists. Changing this forces a new resource to be created.
- `plan` - (Optional) A plan block as documented below.
- `availability_set_id` - (Optional) The Id of the Availability Set in which to create the virtual machine
- `boot_diagnostics` - (Optional) A boot diagnostics profile block as referenced below.
- `vm_size` - (Required) Specifies the size of the virtual machine.
- `storage_image_reference` - (Optional) A Storage Image Reference block as documented below.
- `storage_os_disk` - (Required) A Storage OS DFlag to enable deletion of the OS disk VHD blob when the VM is deleted, defaults to `false` (not yet supported).
- `storage_data_disk` - (Optional) A list of Storage Data disk blocks as referenced below.
- `delete_data_disks_on_termination` - (Optional) Flag to enable deletion of storage data disk VHD blobs when the VM is deleted, defaults to `false`.
- `os_profile` - (Optional) An OS Profile block as documented below. Required when `create_option` in the `storage_os_disk` block is set to `FromImage`.
- `identity` - (Optional) An identity block as documented below.
- `license_type` - (Optional, when a Windows machine) Specifies the Windows OS license type. If supplied, the only allowed values are `Windows_Client` and `Windows_Server`.
- `os_profile_windows_config` - (Required, when a Windows machine) A Windows config block as documented below.
- `os_profile_linux_config` - (Required, when a Linux machine) A Linux config block as documented below.
- `os_profile_secrets` - (Optional) A collection of Secret blocks as documented below.
- `network_interface_ids` - (Required) Specifies the list of resource IDs for the network interfaces associated with the virtual machine.

- `primary_network_interface_id` - (Optional) Specifies the resource ID for the primary network interface associated with the virtual machine.
- `tags` - (Optional) A mapping of tags to assign to the resource.

For more information on the different example configurations, please check out the azure documentation

`Plan` supports the following:

- `name` - (Required) Specifies the name of the image from the marketplace.
- `publisher` - (Required) Specifies the publisher of the image.
- `product` - (Required) Specifies the product of the image from the marketplace.

`boot_diagnostics` supports the following:

- `enabled`: (Required) Whether to enable boot diagnostics for the virtual machine.
- `storage_uri`: (Required) Blob endpoint for the storage account to hold the virtual machine's diagnostic files. This must be the root of a storage account, and not a storage container.

`storage_image_reference` supports the following:

- `id` - (Optional) Specifies the ID of the (custom) image to use to create the virtual machine, for example:

```
resource "azurestack_image" "test" {
    name = "test"
  ...
}

resource "azurestack_virtual_machine" "test" {
    name = "test"
  ...

    storage_image_reference {
        id = "${azurestack_image.test.id}"
    }

...
```

- `publisher` - (Required, when not using image resource) Specifies the publisher of the image used to create the virtual machine. Changing this forces a new resource to be created.
- `offer` - (Required, when not using image resource) Specifies the offer of the image used to create the virtual machine. Changing this forces a new resource to be created.

- `sku` - (Required, when not using image resource) Specifies the SKU of the image used to create the virtual machine. Changing this forces a new resource to be created.
- `version` - (Optional) Specifies the version of the image used to create the virtual machine. Changing this forces a new resource to be created.

`storage_os_disk` supports the following:

- `name` - (Required) Specifies the disk name.
- `vhd_uri` - (Optional) Specifies the vhd uri. Changing this forces a new resource to be created.
- `create_option` - (Required) Specifies how the virtual machine should be created. Possible value is`FromImage`.
- `caching` - (Optional) Specifies the caching requirements.
- `image_uri` - (Optional) Specifies the image_uri in the form publisherName:offer:skus:version. `image_uri` can also specify the VHD uri of a custom VM image to clone. When cloning a custom disk image the `os_type` documented below becomes required.
- `os_type` - (Optional) Specifies the operating system Type, valid values are windows, linux.
- `disk_size_gb` - (Optional) Specifies the size of the os disk in gigabytes.

`storage_data_disk` supports the following:

- `name` - (Required) Specifies the name of the data disk.
- `vhd_uri` - (Optional) Specifies the uri of the location in storage where the vhd for the virtual machine should be placed.
- `create_option` - (Required) Specifies how the data disk should be created. Possible values are `Attach`, `FromImage` and `Empty`.
- `disk_size_gb` - (Required) Specifies the size of the data disk in gigabytes.
- `caching` - (Optional) Specifies the caching requirements.
- `lun` - (Required) Specifies the logical unit number of the data disk.

`os_profile` supports the following:

- `computer_name` - (Required) Specifies the name of the virtual machine.
- `admin_username` - (Required) Specifies the name of the administrator account.
- `admin_password` - (Required for Windows, Optional for Linux) Specifies the password of the administrator account.
- `custom_data` - (Optional) Specifies custom data to supply to the machine. On linux-based systems, this can be used as a cloud-init script. On other systems, this will be copied as a file on disk. Internally, Terraform will base64 encode this value before sending it to the API. The maximum length of the binary array is 65535 bytes.

NOTE: `admin_password` must be between 6-72 characters long and must satisfy at least 3 of password complexity requirements from the following: 1. Contains an uppercase character 2. Contains a lowercase character 3. Contains a

numeric digit 4. Contains a special character

`identity` supports the following:

- `type` - (Required) Specifies the identity type of the virtual machine. The only allowable value is `SystemAssigned`. To enable Managed Service Identity the virtual machine extension "ManagedIdentityExtensionForWindows" or "ManagedIdentityExtensionForLinux" must also be added to the virtual machine. The Principal ID can be retrieved after the virtual machine has been created, e.g.

```
resource "azurestack_virtual_machine" "test" {
  name                = "test"

  identity = {
    type = "SystemAssigned"
  }
}
resource "azurestack_virtual_machine_extension" "test" {
  name                 = "test"
  resource_group_name  = "${azurestack_resource_group.test.name}"
  location             = "${azurestack_resource_group.test.location}"
  virtual_machine_name = "${azurestack_virtual_machine.test.name}"
  publisher            = "Microsoft.ManagedIdentity"
  type                 = "ManagedIdentityExtensionForWindows"
  type_handler_version = "1.0"

  settings = <<SETTINGS
    {
        "port": 50342
    }
SETTINGS
}
output "principal_id" {
  value = "${lookup(azurestack_virtual_machine.test.identity[0], "principal_id")}"
}
```

`os_profile_windows_config` supports the following:

- `provision_vm_agent` - (Optional) This value defaults to false.
- `enable_automatic_upgrades` - (Optional) This value defaults to false.
- `winrm` - (Optional) A collection of WinRM configuration blocks as documented below.
- `additional_unattend_config` - (Optional) An Additional Unattended Config block as documented below.

`winrm` supports the following:

- `protocol` - (Required) Specifies the protocol of listener

- `certificate_url` - (Optional) Specifies URL of the certificate with which new Virtual Machines is provisioned.

`additional_unattend_config` supports the following:

- `pass` - (Required) Specifies the name of the pass that the content applies to. The only allowable value is `oobeSystem`.
- `component` - (Required) Specifies the name of the component to configure with the added content. The only allowable value is `Microsoft-Windows-Shell-Setup`.
- `setting_name` - (Required) Specifies the name of the setting to which the content applies. Possible values are: `FirstLogonCommands` and `AutoLogon`.
- `content` - (Optional) Specifies the base-64 encoded XML formatted content that is added to the unattend.xml file for the specified path and component.

`os_profile_linux_config` supports the following:

- `disable_password_authentication` - (Required) Specifies whether password authentication should be disabled. If set to `false`, an `admin_password` must be specified.
- `ssh_keys` - (Optional) Specifies a collection of `path` and `key_data` to be placed on the virtual machine.

Note: Please note that the only allowed `path` is `/home/<username>/.ssh/authorized_keys` due to a limitation of Azure.

`os_profile_secrets` supports the following:

- `source_vault_id` - (Required) Specifies the key vault to use.
- `vault_certificates` - (Required) A collection of Vault Certificates as documented below

`vault_certificates` support the following:

- `certificate_url` - (Required) Specifies the URI of the key vault secrets in the format of `https://<vaultEndpoint>/secrets/<secretName>/<secretVersion>`. Stored secret is the Base64 encoding of a JSON Object that which is encoded in UTF-8 of which the contents need to be

```
{
  "data":"<Base64-encoded-certificate>",
  "dataType":"pfx",
  "password":"<pfx-file-password>"
}
```

- `certificate_store` - (Required, on windows machines) Specifies the certificate store on the Virtual Machine where the certificate should be added to.

» Attributes Reference

The following attributes are exported:

- `id` - The virtual machine ID.

» Import

Virtual Machines can be imported using the `resource id`, e.g.

```
terraform import azurestack_virtual_machine.test /subscriptions/00000000-0000-0000-0000-0000
```

» azurestack_virtual_machine_extension

Creates a new Virtual Machine Extension to provide post deployment configuration and run automated tasks.

Please Note: The CustomScript extensions for Linux & Windows require that the `commandToExecute` returns a `0` exit code to be classified as successfully deployed. You can achieve this by appending `exit 0` to the end of your `commandToExecute`.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acctestrg"
  location = "West US"
}

resource "azurestack_virtual_network" "test" {
  name                = "acctvn"
  address_space       = ["10.0.0.0/16"]
  location            = "West US"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

resource "azurestack_subnet" "test" {
  name                 = "acctsub"
  resource_group_name  = "${azurestack_resource_group.test.name}"
  virtual_network_name = "${azurestack_virtual_network.test.name}"
  address_prefix       = "10.0.2.0/24"
}

resource "azurestack_network_interface" "test" {
```

```
  name                = "acctni"
  location            = "West US"
  resource_group_name = "${azurestack_resource_group.test.name}"

  ip_configuration {
    name                          = "testconfiguration1"
    subnet_id                     = "${azurestack_subnet.test.id}"
    private_ip_address_allocation = "dynamic"
  }
}

resource "azurestack_storage_account" "test" {
  name                     = "accsa"
  resource_group_name      = "${azurestack_resource_group.test.name}"
  location                 = "westus"
  account_tier             = "Standard"
  account_replication_type = "LRS"

  tags {
    environment = "staging"
  }
}

resource "azurestack_storage_container" "test" {
  name                  = "vhds"
  resource_group_name   = "${azurestack_resource_group.test.name}"
  storage_account_name  = "${azurestack_storage_account.test.name}"
  container_access_type = "private"
}

resource "azurestack_virtual_machine" "test" {
  name                  = "acctvm"
  location              = "West US"
  resource_group_name   = "${azurestack_resource_group.test.name}"
  network_interface_ids = ["${azurestack_network_interface.test.id}"]
  vm_size               = "Standard_A0"

  storage_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }

  storage_os_disk {
    name          = "myosdisk1"
```

```
    vhd_uri       = "${azurestack_storage_account.test.primary_blob_endpoint}${azurestack_st
    caching       = "ReadWrite"
    create_option = "FromImage"
  }

  os_profile {
    computer_name  = "hostname"
    admin_username = "testadmin"
    admin_password = "Password1234!"
  }

  os_profile_linux_config {
    disable_password_authentication = false
  }

  tags {
    environment = "staging"
  }
}

resource "azurestack_virtual_machine_extension" "test" {
  name                 = "hostname"
  location             = "West US"
  resource_group_name  = "${azurestack_resource_group.test.name}"
  virtual_machine_name = "${azurestack_virtual_machine.test.name}"
  publisher            = "Microsoft.Azure.Extensions"
  type                 = "CustomScript"
  type_handler_version = "2.0"

  settings = <<SETTINGS
    {
        "commandToExecute": "hostname && uptime"
    }
SETTINGS

  tags {
    environment = "Production"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the virtual machine extension peering.

Changing this forces a new resource to be created.

- `location` - (Required) The location where the extension is created. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the virtual network. Changing this forces a new resource to be created.
- `virtual_machine_name` - (Required) The name of the virtual machine. Changing this forces a new resource to be created.
- `publisher` - (Required) The publisher of the extension, available publishers can be found by using the Azure CLI.
- `type` - (Required) The type of extension, available types for a publisher can be found using the Azure CLI.

Note: The `Publisher` and `Type` of Virtual Machine Extensions can be found using the Azure CLI, via: `shell $ az vm extension image list --location westus -o table`

- `type_handler_version` - (Required) Specifies the version of the extension to use, available versions can be found using the Azure CLI.
- `auto_upgrade_minor_version` - (Optional) Specifies if the platform deploys the latest minor version update to the `type_handler_version` specified.
- `settings` - (Required) The settings passed to the extension, these are specified as a JSON object in a string.

Please Note: Certain VM Extensions require that the keys in the `settings` block are case sensitive. If you're seeing unhelpful errors, please ensure the keys are consistent with how Azure is expecting them (for instance, for the `JsonADDomainExtension` extension, the keys are expected to be in `TitleCase`.)

- `protected_settings` - (Optional) The protected_settings passed to the extension, like settings, these are specified as a JSON object in a string.

Please Note: Certain VM Extensions require that the keys in the `protected_settings` block are case sensitive. If you're seeing unhelpful errors, please ensure the keys are consistent with how Azure is expecting them (for instance, for the `JsonADDomainExtension` extension, the keys are expected to be in `TitleCase`.)

» Attributes Reference

The following attributes are exported:

- `id` - The Virtual Machine Extension ID.

» Import

Virtual Machine Extensions can be imported using the `resource id`, e.g.

```
terraform import azurestack_virtual_machine_extension.test /subscriptions/00000000-0000-0000
```

» azurestack_dns_a_record

Enables you to manage DNS A Records within Azure DNS.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acceptanceTestResourceGroup1"
  location = "West US"
}

resource "azurestack_dns_zone" "test" {
  name                = "mydomain.com"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

resource "azurestack_dns_a_record" "test" {
  name                = "test"
  zone_name           = "${azurestack_dns_zone.test.name}"
  resource_group_name = "${azurestack_resource_group.test.name}"
  ttl                 = 300
  records             = ["10.0.180.17"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the DNS A Record.
- `resource_group_name` - (Required) Specifies the resource group where the resource exists. Changing this forces a new resource to be created.
- `zone_name` - (Required) Specifies the DNS Zone where the resource exists. Changing this forces a new resource to be created.
- `TTL` - (Required) The Time To Live (TTL) of the DNS record.
- `records` - (Required) List of IPv4 Addresses.

- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

The following attributes are exported:

- `id` - The DNS A Record ID.

» Import

A records can be imported using the `resource id`, e.g.

```
terraform import azurestack_dns_a_record.test /subscriptions/00000000-0000-0000-0000-000000
```

» azurestack_dns_zone

Enables you to manage DNS zones within Azure DNS. These zones are hosted on Azure's name servers to which you can delegate the zone from the parent domain.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acceptanceTestResourceGroup1"
  location = "West US"
}

resource "azurestack_dns_zone" "test" {
  name                = "mydomain.com"
  resource_group_name = "${azurestack_resource_group.test.name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the DNS Zone. Must be a valid domain name.
- `resource_group_name` - (Required) Specifies the resource group where the resource exists. Changing this forces a new resource to be created.
- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

The following attributes are exported:

- `id` - The DNS Zone ID.
- `max_number_of_record_sets` - (Optional) Maximum number of Records in the zone. Defaults to `1000`.
- `number_of_record_sets` - (Optional) The number of records already in the zone.
- `name_servers` - (Optional) A list of values that make up the NS record for the zone.

» Import

DNS Zones can be imported using the `resource id`, e.g.

```
terraform import azurestack_dns_zone.zone1 /subscriptions/00000000-0000-0000-0000-0000000000
```

» azurestack_local_network_gateway

Manages a local network gateway connection over which specific connections can be configured.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "localNetworkGWTest"
  location = "West US"
}

resource "azurestack_local_network_gateway" "home" {
  name                = "backHome"
  resource_group_name = "${azurestack_resource_group.test.name}"
  location            = "${azurestack_resource_group.test.location}"
  gateway_address     = "12.13.14.15"
  address_space       = ["10.0.0.0/16"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the local network gateway. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the local network gateway.
- `location` - (Required) The location/region where the local network gateway is created. Changing this forces a new resource to be created.
- `gateway_address` - (Required) The IP address of the gateway to which to connect.
- `address_space` - (Required) The list of string CIDRs representing the address spaces the gateway exposes.
- `bgp_settings` - (Optional) A `bgp_settings` block as defined below containing the Local Network Gateway's BGP speaker settings.
- `tags` - (Optional) A mapping of tags to assign to the resource.

`bgp_settings` supports the following:

- `asn` - (Required) The BGP speaker's ASN.
- `bgp_peering_address` - (Required) The BGP peering address and BGP identifier of this BGP speaker.
- `peer_weight` - (Optional) The weight added to routes learned from this BGP speaker.

» Attributes Reference

The following attributes are exported:

- `id` - The local network gateway unique ID within Azure.

» Import

Local Network Gateways can be imported using the `resource id`, e.g.

```
terraform import azurestack_local_network_gateway.lng1 /subscriptions/00000000-0000-0000-000
```

» azurestack_network_interface

Manages a Network Interface located in a Virtual Network, usually attached to a Virtual Machine.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acceptanceTestResourceGroup1"
  location = "West US"
}

resource "azurestack_virtual_network" "test" {
  name                = "acceptanceTestVirtualNetwork1"
  address_space       = ["10.0.0.0/16"]
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

resource "azurestack_subnet" "test" {
  name                 = "testsubnet"
  resource_group_name  = "${azurestack_resource_group.test.name}"
  virtual_network_name = "${azurestack_virtual_network.test.name}"
  address_prefix       = "10.0.2.0/24"
}

resource "azurestack_network_interface" "test" {
  name                = "acceptanceTestNetworkInterface1"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"

  ip_configuration {
    name                          = "testconfiguration1"
    subnet_id                     = "${azurestack_subnet.test.id}"
    private_ip_address_allocation = "dynamic"
  }

  tags {
    environment = "staging"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the network interface. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the network interface. Changing this forces a new resource

to be created.

- `location` - (Required) The location/region where the network interface is created. Changing this forces a new resource to be created.
- `network_security_group_id` - (Optional) The ID of the Network Security Group to associate with the network interface.
- `internal_dns_name_label` - (Optional) Relative DNS name for this NIC used for internal communications between VMs in the same VNet
- `enable_ip_forwarding` - (Optional) Enables IP Forwarding on the NIC. Defaults to `false`.
- `dns_servers` - (Optional) List of DNS servers IP addresses to use for this NIC, overrides the VNet-level server list
- `ip_configuration` - (Required) One or more `ip_configuration` associated with this NIC as documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `ip_configuration` block supports:

- `name` - (Required) User-defined name of the IP.
- `subnet_id` - (Required) Reference to a subnet in which this NIC has been created.
- `private_ip_address` - (Optional) Static IP Address.
- `private_ip_address_allocation` - (Required) Defines how a private IP address is assigned. Options are Static or Dynamic.
- `public_ip_address_id` - (Optional) Reference to a Public IP Address to associate with this NIC
- `load_balancer_backend_address_pools_ids` - (Optional) List of Load Balancer Backend Address Pool IDs references to which this NIC belongs
- `load_balancer_inbound_nat_rules_ids` - (Optional) List of Load Balancer Inbound Nat Rules IDs involving this NIC
- `application_security_group_ids` - (Optional) List of Application Security Group IDs which should be attached to this NIC

Note: Application Security Groups are currently in Public Preview on an opt-in basis. More information, including how you can register for the Preview, and which regions Application Security Groups are available in are available here

- `primary` - (Optional) Is this the Primary Network Interface? If set to `true` this should be the first `ip_configuration` in the array.

» Attributes Reference

The following attributes are exported:

- `id` - The Virtual Network Interface ID.
- `mac_address` - The media access control (MAC) address of the network interface.
- `private_ip_address` - The private ip address of the network interface.
- `virtual_machine_id` - Reference to a VM with which this NIC has been associated.
- `applied_dns_servers` - If the VM that uses this NIC is part of an Availability Set, then this list will have the union of all DNS servers from all NICs that are part of the Availability Set
- `internal_fqdn` - Fully qualified DNS name supporting internal communications between VMs in the same VNet

» Import

Network Interfaces can be imported using the `resource id`, e.g.

```
terraform import azurestack_network_interface.test /subscriptions/00000000-0000-0000-0000-00
```

» azurestack_network_security_group

Manages a network security group that contains a list of network security rules. Network security groups enable inbound or outbound traffic to be enabled or denied.

NOTE on Network Security Groups and Network Security Rules: Terraform currently provides both a standalone Network Security Rule resource, and allows for Network Security Rules to be defined in-line within the Network Security Group resource. At this time you cannot use a Network Security Group with in-line Network Security Rules in conjunction with any Network Security Rule resources. Doing so will cause a conflict of rule settings and will overwrite rules.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acceptanceTestResourceGroup1"
  location = "West US"
}

resource "azurestack_network_security_group" "test" {
```

```
  name                = "acceptanceTestSecurityGroup1"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"

  security_rule {
    name                       = "test123"
    priority                   = 100
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "*"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }

  tags {
    environment = "Production"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the network security group. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the network security group. Changing this forces a new resource to be created.
- `location` - (Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
- `security_rule` - (Optional) One or more `security_rule` blocks as defined below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `security_rule` block supports:

- `name` - (Required) The name of the security rule.
- `description` - (Optional) A description for this rule. Restricted to 140 characters.
- `protocol` - (Required) Network protocol this rule applies to. Can be `Tcp`, `Udp` or `*` to match both.

- `source_port_range` - (Optional) Source Port or Range. Integer or range between `0` and `65535` or `*` to match any.
- `destination_port_range` - (Optional) Destination Port or Range. Integer or range between `0` and `65535` or `*` to match any.
- `source_address_prefix` - (Optional) CIDR or source IP range or * to match any IP. Tags such as 'VirtualNetwork', 'AzureLoadBalancer' and 'Internet' can also be used.
- `destination_address_prefix` - (Optional) CIDR or destination IP range or * to match any IP. Tags such as 'VirtualNetwork', 'AzureLoadBalancer' and 'Internet' can also be used.
- `access` - (Required) Specifies whether network traffic is allowed or denied. Possible values are `Allow` and `Deny`.
- `priority` - (Required) Specifies the priority of the rule. The value can be between 100 and 4096. The priority number must be unique for each rule in the collection. The lower the priority number, the higher the priority of the rule.
- `direction` - (Required) The direction specifies if rule will be evaluated on incoming or outgoing traffic. Possible values are `Inbound` and `Outbound`.

» Attributes Reference

The following attributes are exported:

- `id` - The Network Security Group ID.

» Import

Network Security Groups can be imported using the `resource id`, e.g.

```
terraform import azurestack_network_security_group.group1 /subscriptions/00000000-0000-0000-
```

» azurestack_network_security_rule

Manages a Network Security Rule.

NOTE on Network Security Groups and Network Security Rules: Terraform currently provides both a standalone Network Security Rule resource, and allows for Network Security Rules to be defined in-line within the Network Security Group resource. At this time you cannot use a Network Security Group with in-line Network Security Rules in conjunction with any Network Security

Rule resources. Doing so will cause a conflict of rule settings and will overwrite rules.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acceptanceTestResourceGroup1"
  location = "West US"
}

resource "azurestack_network_security_group" "test" {
  name                = "acceptanceTestSecurityGroup1"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

resource "azurestack_network_security_rule" "test" {
  name                        = "test123"
  priority                    = 100
  direction                   = "Outbound"
  access                      = "Allow"
  protocol                    = "Tcp"
  source_port_range           = "*"
  destination_port_range      = "*"
  source_address_prefix       = "*"
  destination_address_prefix  = "*"
  resource_group_name         = "${azurestack_resource_group.test.name}"
  network_security_group_name = "${azurestack_network_security_group.test.name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the security rule. This needs to be unique across all Rules in the Network Security Group. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the Network Security Rule. Changing this forces a new resource to be created.
- `network_security_group_name` - (Required) The name of the Network Security Group that we want to attach the rule to. Changing this forces a new resource to be created.

- `description` - (Optional) A description for this rule. Restricted to 140 characters.
- `protocol` - (Required) Network protocol this rule applies to. Possible values include `Tcp`, `Udp` or `*` (which matches both).
- `source_port_range` - (Optional) Source Port or Range. Integer or range between `0` and `65535` or `*` to match any.
- `destination_port_range` - (Optional) Destination Port or Range. Integer or range between `0` and `65535` or `*` to match any.
- `source_address_prefix` - (Optional) CIDR or source IP range or * to match any IP. Tags such as 'VirtualNetwork', 'AzureLoadBalancer' and 'Internet' can also be used.
- `destination_address_prefix` - (Optional) CIDR or destination IP range or * to match any IP. Tags such as 'VirtualNetwork', 'AzureLoadBalancer' and 'Internet' can also be used.
- `access` - (Required) Specifies whether network traffic is allowed or denied. Possible values are `Allow` and `Deny`.
- `priority` - (Required) Specifies the priority of the rule. The value can be between 100 and 4096. The priority number must be unique for each rule in the collection. The lower the priority number, the higher the priority of the rule.
- `direction` - (Required) The direction specifies if rule will be evaluated on incoming or outgoing traffic. Possible values are `Inbound` and `Outbound`.

» Attributes Reference

The following attributes are exported:

- `id` - The Network Security Rule ID.

» Import

Network Security Rules can be imported using the `resource id`, e.g.

```
terraform import azurestack_network_security_rule.rule1 /subscriptions/00000000-0000-0000-00
```

» azurestack_public_ip

Manages a Public IP Address.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "resourceGroup1"
  location = "West US"
}

resource "azurestack_public_ip" "test" {
  name                         = "acceptanceTestPublicIp1"
  location                     = "West US"
  resource_group_name          = "${azurestack_resource_group.test.name}"
  public_ip_address_allocation = "static"

  tags {
    environment = "Production"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the Public IP resource . Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the public ip.
- `location` - (Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
- `public_ip_address_allocation` - (Required) Defines whether the IP address is static or dynamic. Options are Static or Dynamic.

Note `Dynamic` Public IP Addresses aren't allocated until they're assigned to a resource (such as a Virtual Machine or a Load Balancer) by design within Azure - more information is available below.

- `idle_timeout_in_minutes` - (Optional) Specifies the timeout for the TCP idle connection. The value can be set between 4 and 30 minutes.
- `domain_name_label` - (Optional) Label for the Domain Name. Will be used to make up the FQDN. If a domain name label is specified, an A DNS record is created for the public IP in the Microsoft Azure DNS system.
- `reverse_fqdn` - (Optional) A fully qualified domain name that resolves to this public IP address. If the reverseFqdn is specified, then a PTR DNS record is created pointing from the IP address in the in-addr.arpa domain to the reverse FQDN.

- `tags` - (Optional) A mapping of tags to assign to the resource.

» Attributes Reference

The following attributes are exported:

- `id` - The Public IP ID.
- `ip_address` - The IP address value that was allocated.

Note `Dynamic` Public IP Addresses aren't allocated until they're attached to a device (e.g. a Virtual Machine/Load Balancer). Instead you can obtain the IP Address once the the Public IP has been assigned via the `azurestack_public_ip` Data Source (not currently available)

- `fqdn` - Fully qualified domain name of the A DNS record associated with the public IP. This is the concatenation of the domainNameLabel and the regionalized DNS zone

» Import

Public IPs can be imported using the `resource id`, e.g.

```
terraform import azurestack_public_ip.myPublicIp /subscriptions/00000000-0000-0000-0000-0000
```

» azurestack_subnet

Manages a subnet. Subnets represent network segments within the IP space defined by the virtual network.

NOTE on Virtual Networks and Subnet's: Terraform currently provides both a standalone Subnet resource, and allows for Subnets to be defined in-line within the Virtual Network resource. At this time you cannot use a Virtual Network with in-line Subnets in conjunction with any Subnet resources. Doing so will cause a conflict of Subnet configurations and will overwrite Subnet's.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acceptanceTestResourceGroup1"
  location = "West US"
}

resource "azurestack_virtual_network" "test" {
  name                = "acceptanceTestVirtualNetwork1"
```

```
  address_space       = ["10.0.0.0/16"]
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

resource "azurestack_subnet" "test" {
  name                 = "testsubnet"
  resource_group_name  = "${azurestack_resource_group.test.name}"
  virtual_network_name = "${azurestack_virtual_network.test.name}"
  address_prefix       = "10.0.1.0/24"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the subnet. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the subnet. Changing this forces a new resource to be created.
- `virtual_network_name` - (Required) The name of the virtual network to which to attach the subnet. Changing this forces a new resource to be created.
- `address_prefix` - (Required) The address prefix to use for the subnet.
- `network_security_group_id` - (Optional) The ID of the Network Security Group to associate with the subnet.
- `route_table_id` - (Optional) The ID of the Route Table to associate with the subnet.

» Attributes Reference

The following attributes are exported:

- `id` - The subnet ID.
- `ip_configurations` - The collection of IP Configurations with IPs within this subnet.
- `name` - The name of the subnet.
- `resource_group_name` - The name of the resource group in which the subnet is created in.
- `virtual_network_name` - The name of the virtual network in which the subnet is created in

- `address_prefix` - The address prefix for the subnet

» Import

Subnets can be imported using the `resource id`, e.g.

```
terraform import azurestack_subnet.testSubnet /subscriptions/00000000-0000-0000-0000-000000
```

» azurestack_virtual_network

Creates a new virtual network including any configured subnets. Each subnet can optionally be configured with a security group to be associated with the subnet.

NOTE on Virtual Networks and Subnet's: Terraform currently provides both a standalone Subnet resource, and allows for Subnets to be defined in-line within the Virtual Network resource. At this time you cannot use a Virtual Network with in-line Subnets in conjunction with any Subnet resources. Doing so will cause a conflict of Subnet configurations and will overwrite Subnet's.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acceptanceTestResourceGroup1"
  location = "West US"
}

resource "azurestack_network_security_group" "test" {
  name                = "acceptanceTestSecurityGroup1"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

resource "azurestack_virtual_network" "test" {
  name                = "virtualNetwork1"
  resource_group_name = "${azurestack_resource_group.test.name}"
  address_space       = ["10.0.0.0/16"]
  location            = "West US"
  dns_servers         = ["10.0.0.4", "10.0.0.5"]

  subnet {
    name           = "subnet1"
    address_prefix = "10.0.1.0/24"
```

```
  }

  subnet {
    name           = "subnet2"
    address_prefix = "10.0.2.0/24"
  }

  subnet {
    name           = "subnet3"
    address_prefix = "10.0.3.0/24"
    security_group = "${azurestack_network_security_group.test.id}"
  }

  tags {
    environment = "Production"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the virtual network. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the virtual network.
- `address_space` - (Required) The address space that is used the virtual network. You can supply more than one address space. Changing this forces a new resource to be created.
- `location` - (Required) The location/region where the virtual network is created. Changing this forces a new resource to be created.
- `dns_servers` - (Optional) List of IP addresses of DNS servers
- `subnet` - (Optional) Can be specified multiple times to define multiple subnets. Each `subnet` block supports fields documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `subnet` block supports:

- `name` - (Required) The name of the subnet.
- `address_prefix` - (Required) The address prefix to use for the subnet.
- `security_group` - (Optional) The Network Security Group to associate with the subnet. (Referenced by `id`, ie. `azurestack_network_security_group.test.id`)

» Attributes Reference

The following attributes are exported:

- `id` - The virtual NetworkConfiguration ID.
- `name` - The name of the virtual network.
- `resource_group_name` - The name of the resource group in which to create the virtual network.
- `location` - The location/region where the virtual network is created
- `address_space` - The address space that is used the virtual network.

» Import

Virtual Networks can be imported using the `resource id`, e.g.

```
terraform import azurestack_virtual_network.testNetwork /subscriptions/00000000-0000-0000-00
```

» azurestack_storage_account

Manages an Azure Storage Account.

» Example Usage

```
resource "azurestack_resource_group" "testrg" {
  name     = "resourceGroupName"
  location = "westus"
}

resource "azurestack_storage_account" "testsa" {
  name                     = "storageaccountname"
  resource_group_name      = "${azurestack_resource_group.testrg.name}"
  location                 = "westus"
  account_tier             = "Standard"
  account_replication_type = "LRS"

  tags {
    environment = "staging"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the storage account. Changing this forces a new resource to be created. This must be unique across the entire Azure service, not just within the resource group.
- `resource_group_name` - (Required) The name of the resource group in which to create the storage account. Changing this forces a new resource to be created.
- `location` - (Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.
- `account_kind` - (Optional) Defines the Kind of account. Valid option is `Storage`. . Changing this forces a new resource to be created. Defaults to `Storage` currently as per Azure Stack Storage Differences
- `account_tier` - (Required) Defines the Tier to use for this storage account. Valid options are `Standard` and `Premium`. Changing this forces a new resource to be created - **`Can be provisioned, but no performance limit or guarantee.`**
- `account_replication_type` - (Required) Defines the type of replication to use for this storage account. Valid option is `LRS` currently as per Azure Stack Storage Differences
- `access_tier` - (Required for `BlobStorage` accounts) Defines the access tier for `BlobStorage` accounts. Valid options are `Hot` and `Cold`, defaults to `Hot`. - **`Currently Not Supported on Azure Stack`**
- `account_encryption_source` - (Optional) The Encryption Source for this Storage Account. Possible values are `Microsoft.Keyvault` and `Microsoft.Storage`. Defaults to `Microsoft.Storage`.
- `custom_domain` - (Optional) A `custom_domain` block as documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

- `custom_domain` supports the following:
- `name` - (Optional) The Custom Domain Name to use for the Storage Account, which will be validated by Azure.
- `use_subdomain` - (Optional) Should the Custom Domain Name be validated by using indirect CNAME validation?

Note: More information on Validation is available here

» Attributes Reference

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

- `id` - The storage account Resource ID.
- `primary_location` - The primary location of the storage account.
- `secondary_location` - The secondary location of the storage account.
- `primary_blob_endpoint` - The endpoint URL for blob storage in the primary location.
- `secondary_blob_endpoint` - The endpoint URL for blob storage in the secondary location.
- `primary_queue_endpoint` - The endpoint URL for queue storage in the primary location.
- `secondary_queue_endpoint` - The endpoint URL for queue storage in the secondary location.
- `primary_table_endpoint` - The endpoint URL for table storage in the primary location.
- `secondary_table_endpoint` - The endpoint URL for table storage in the secondary location.
- `primary_file_endpoint` - The endpoint URL for file storage in the primary location.
- `primary_access_key` - The primary access key for the storage account
- `secondary_access_key` - The secondary access key for the storage account
- `primary_connection_string` - The connection string associated with the primary location
- `secondary_connection_string` - The connection string associated with the secondary location
- `primary_blob_connection_string` - The connection string associated with the primary blob location
- `secondary_blob_connection_string` - The connection string associated with the secondary blob location

» Import

Storage Accounts can be imported using the `resource id`, e.g.

```
terraform import azurestack_storage_account.storageAcc1 /subscriptions/00000000-0000-0000-00
```

» azurestack_storage_container

Manages an Azure Storage Container.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acctestrg"
  location = "westus"
}

resource "azurestack_storage_account" "test" {
  name                     = "acctestsstorageaccount"
  resource_group_name      = "${azurestack_resource_group.test.name}"
  location                 = "westus"
  account_tier             = "Standard"
  account_replication_type = "LRS"

  tags {
    environment = "staging"
  }
}

resource "azurestack_storage_container" "test" {
  name                  = "vhds"
  resource_group_name   = "${azurestack_resource_group.test.name}"
  storage_account_name  = "${azurestack_storage_account.test.name}"
  container_access_type = "private"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the storage container. Must be unique within the storage service the container is located.
- `resource_group_name` - (Required) The name of the resource group in which to create the storage container. Changing this forces a new resource to be created.
- `storage_account_name` - (Required) Specifies the storage account in which to create the storage container. Changing this forces a new resource to be created.
- `container_access_type` - (Optional) The 'interface' for access the container provides. Can be either `blob`, `container` or `private`. Defaults to `private`. Changing this forces a new resource to be created.

» Attributes Reference

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

- `id` - The storage container Resource ID.
- `properties` - Key-value definition of additional properties associated to the storage container

» azurestack_storage_blob

Manages an Azure Storage Blob.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "acctestrg-d"
  location = "westus"
}

resource "azurestack_storage_account" "test" {
  name                     = "acctestaccs"
  resource_group_name      = "${azurestack_resource_group.test.name}"
  location                 = "westus"
  account_tier             = "Standard"
  account_replication_type = "LRS"
}

resource "azurestack_storage_container" "test" {
  name                  = "vhds"
  resource_group_name   = "${azurestack_resource_group.test.name}"
  storage_account_name  = "${azurestack_storage_account.test.name}"
  container_access_type = "private"
}

resource "azurestack_storage_blob" "testsb" {
  name = "sample.vhd"

  resource_group_name    = "${azurestack_resource_group.test.name}"
  storage_account_name   = "${azurestack_storage_account.test.name}"
  storage_container_name = "${azurestack_storage_container.test.name}"

  type = "page"
  size = 5120
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the storage blob. Must be unique within the storage container the blob is located.
- `resource_group_name` - (Required) The name of the resource group in which to create the storage container. Changing this forces a new resource to be created.
- `storage_account_name` - (Required) Specifies the storage account in which to create the storage container. Changing this forces a new resource to be created.
- `storage_container_name` - (Required) The name of the storage container in which this blob should be created.
- `type` - (Optional) The type of the storage blob to be created. One of either `block` or `page`. When not copying from an existing blob, this becomes required.
- `size` - (Optional) Used only for `page` blobs to specify the size in bytes of the blob to be created. Must be a multiple of 512. Defaults to 0.
- `source` - (Optional) An absolute path to a file on the local system. Cannot be defined if `source_uri` is defined.
- `source_uri` - (Optional) The URI of an existing blob, or a file in the Azure File service, to use as the source contents for the blob to be created. Changing this forces a new resource to be created. Cannot be defined if `source` is defined.
- `parallelism` - (Optional) The number of workers per CPU core to run for concurrent uploads. Defaults to `8`.
- `attempts` - (Optional) The number of attempts to make per page or block when uploading. Defaults to `1`.

» Attributes Reference

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

- `id` - The storage blob Resource ID.
- `url` - The URL of the blob

» azurestack_lb

Manages a Load Balancer Resource.

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "LoadBalancerRG"
  location = "West US"
}

resource "azurestack_public_ip" "test" {
  name                         = "PublicIPForLB"
  location                     = "West US"
  resource_group_name          = "${azurestack_resource_group.test.name}"
  public_ip_address_allocation = "static"
}

resource "azurestack_lb" "test" {
  name                = "TestLoadBalancer"
  location            = "West US"
  resource_group_name = "${azurestack_resource_group.test.name}"

  frontend_ip_configuration {
    name                 = "PublicIPAddress"
    public_ip_address_id = "${azurestack_public_ip.test.id}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the LoadBalancer.
- `resource_group_name` - (Required) The name of the resource group in which to create the LoadBalancer.
- `location` - (Required) Specifies the supported Azure location where the resource exists.
- `frontend_ip_configuration` - (Optional) A frontend ip configuration block as documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

`frontend_ip_configuration` supports the following:

- `name` - (Required) Specifies the name of the frontend ip configuration.
- `subnet_id` - (Optional) Reference to subnet associated with the IP Configuration.

- `private_ip_address` - (Optional) Private IP Address to assign to the Load Balancer. The last one and first four IPs in any range are reserved and cannot be manually assigned.
- `private_ip_address_allocation` - (Optional) Defines how a private IP address is assigned. Options are Static or Dynamic.
- `public_ip_address_id` - (Optional) Reference to Public IP address to be associated with the Load Balancer.

» Attributes Reference

The following attributes are exported:

- `id` - The LoadBalancer ID.
- `private_ip_address` - The first private IP address assigned to the load balancer in `frontend_ip_configuration` blocks, if any.
- `private_ip_addresses` - The list of private IP address assigned to the load balancer in `frontend_ip_configuration` blocks, if any.

» Import

Load Balancers can be imported using the `resource id`, e.g.

```
terraform import azurestack_lb.test /subscriptions/00000000-0000-0000-0000-000000000000/reso
```

» azurestack_lb_rule

Manages a Load Balancer Rule.

NOTE When using this resource, the Load Balancer needs to have a FrontEnd IP Configuration Attached

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "LoadBalancerRG"
  location = "West US"
}

resource "azurestack_public_ip" "test" {
  name                         = "PublicIPForLB"
  location                     = "West US"
  resource_group_name          = "${azurestack_resource_group.test.name}"
  public_ip_address_allocation = "static"
```

```
}

resource "azurestack_lb" "test" {
  name                = "TestLoadBalancer"
  location            = "West US"
  resource_group_name = "${azurestack_resource_group.test.name}"

  frontend_ip_configuration {
    name                 = "PublicIPAddress"
    public_ip_address_id = "${azurestack_public_ip.test.id}"
  }
}

resource "azurestack_lb_rule" "test" {
  resource_group_name            = "${azurestack_resource_group.test.name}"
  loadbalancer_id                = "${azurestack_lb.test.id}"
  name                           = "LBRule"
  protocol                       = "Tcp"
  frontend_port                  = 3389
  backend_port                   = 3389
  frontend_ip_configuration_name = "PublicIPAddress"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the LB Rule.
- `resource_group_name` - (Required) The name of the resource group in which to create the resource.
- `loadbalancer_id` - (Required) The ID of the Load Balancer in which to create the Rule.
- `frontend_ip_configuration_name` - (Required) The name of the frontend IP configuration to which the rule is associated.
- `protocol` - (Required) The transport protocol for the external endpoint. Possible values are `Udp` or `Tcp`.
- `frontend_port` - (Required) The port for the external endpoint. Port numbers for each Rule must be unique within the Load Balancer. Possible values range between 1 and 65534, inclusive.
- `backend_port` - (Required) The port used for internal connections on the endpoint. Possible values range between 1 and 65535, inclusive.
- `backend_address_pool_id` - (Optional) A reference to a Backend Address Pool over which this Load Balancing Rule operates.
- `probe_id` - (Optional) A reference to a Probe used by this Load Balancing Rule.

- `enable_floating_ip` - (Optional) Floating IP is pertinent to failover scenarios: a "floating" IP is reassigned to a secondary server in case the primary server fails. Floating IP is required for SQL AlwaysOn.
- `idle_timeout_in_minutes` - (Optional) Specifies the timeout for the Tcp idle connection. The value can be set between 4 and 30 minutes. The default value is 4 minutes. This element is only used when the protocol is set to Tcp.
- `load_distribution` - (Optional) Specifies the load balancing distribution type to be used by the Load Balancer. Possible values are: `Default` – The load balancer is configured to use a 5 tuple hash to map traffic to available servers. `SourceIP` – The load balancer is configured to use a 2 tuple hash to map traffic to available servers. `SourceIPProtocol` – The load balancer is configured to use a 3 tuple hash to map traffic to available servers. Also known as Session Persistence, where the options are called `None`, `Client IP` and `Client IP and Protocol` respectively.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Load Balancer to which the resource is attached.

» Import

Load Balancer Rules can be imported using the `resource id`, e.g.

```
terraform import azurestack_lb_rule.test /subscriptions/00000000-0000-0000-0000-000000000000
```

» azurestack_lb_backend_address_pool

Manages a LoadBalancer Backend Address Pool.

NOTE: When using this resource, the LoadBalancer needs to have a FrontEnd IP Configuration Attached

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "LoadBalancerRG"
  location = "West US"
}

resource "azurestack_public_ip" "test" {
```

```
  name                          = "PublicIPForLB"
  location                      = "West US"
  resource_group_name           = "${azurestack_resource_group.test.name}"
  public_ip_address_allocation = "static"
}

resource "azurestack_lb" "test" {
  name                = "TestLoadBalancer"
  location            = "West US"
  resource_group_name = "${azurestack_resource_group.test.name}"

  frontend_ip_configuration {
    name                 = "PublicIPAddress"
    public_ip_address_id = "${azurestack_public_ip.test.id}"
  }
}

resource "azurestack_lb_backend_address_pool" "test" {
  resource_group_name = "${azurestack_resource_group.test.name}"
  loadbalancer_id     = "${azurestack_lb.test.id}"
  name                = "BackEndAddressPool"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the Backend Address Pool.
- `resource_group_name` - (Required) The name of the resource group in which to create the resource.
- `loadbalancer_id` - (Required) The ID of the LoadBalancer in which to create the Backend Address Pool.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the LoadBalancer to which the resource is attached.

» Import

Load Balancer Backend Address Pools can be imported using the `resource id`, e.g.

```
terraform import azurestack_lb_backend_address_pool.test /subscriptions/00000000-0000-0000-0
```

» azurestack_lb_nat_rule

Manages a LoadBalancer NAT Rule.

NOTE When using this resource, the LoadBalancer needs to have a FrontEnd IP Configuration Attached

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "LoadBalancerRG"
  location = "West US"
}

resource "azurestack_public_ip" "test" {
  name                         = "PublicIPForLB"
  location                     = "West US"
  resource_group_name          = "${azurestack_resource_group.test.name}"
  public_ip_address_allocation = "static"
}

resource "azurestack_lb" "test" {
  name                = "TestLoadBalancer"
  location            = "West US"
  resource_group_name = "${azurestack_resource_group.test.name}"

  frontend_ip_configuration {
    name                 = "PublicIPAddress"
    public_ip_address_id = "${azurestack_public_ip.test.id}"
  }
}

resource "azurestack_lb_nat_rule" "test" {
  resource_group_name            = "${azurestack_resource_group.test.name}"
  loadbalancer_id                = "${azurestack_lb.test.id}"
  name                           = "RDPAccess"
  protocol                       = "Tcp"
  frontend_port                  = 3389
  backend_port                   = 3389
  frontend_ip_configuration_name = "PublicIPAddress"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the NAT Rule.
- `resource_group_name` - (Required) The name of the resource group in which to create the resource.
- `loadbalancer_id` - (Required) The ID of the LoadBalancer in which to create the NAT Rule.
- `frontend_ip_configuration_name` - (Required) The name of the frontend IP configuration exposing this rule.
- `protocol` - (Required) The transport protocol for the external endpoint. Possible values are `Udp` or `Tcp`.
- `frontend_port` - (Required) The port for the external endpoint. Port numbers for each Rule must be unique within the Load Balancer. Possible values range between 1 and 65534, inclusive.
- `backend_port` - (Required) The port used for internal connections on the endpoint. Possible values range between 1 and 65535, inclusive.
- `enable_floating_ip` - (Optional) Enables the Floating IP Capacity, required to configure a SQL AlwaysOn Availability Group.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the LoadBalancer to which the resource is attached.

» Import

Load Balancer NAT Rules can be imported using the `resource id`, e.g.

```
terraform import azurestack_lb_nat_rule.test /subscriptions/00000000-0000-0000-0000-0000000
```

» azurestack_lb_probe

Manages a LoadBalancer Probe Resource.

NOTE When using this resource, the LoadBalancer needs to have a FrontEnd IP Configuration Attached

» Example Usage

```
resource "azurestack_resource_group" "test" {
```

```
  name     = "LoadBalancerRG"
  location = "West US"
}

resource "azurestack_public_ip" "test" {
  name                         = "PublicIPForLB"
  location                     = "West US"
  resource_group_name          = "${azurestack_resource_group.test.name}"
  public_ip_address_allocation = "static"
}

resource "azurestack_lb" "test" {
  name                = "TestLoadBalancer"
  location            = "West US"
  resource_group_name = "${azurestack_resource_group.test.name}"

  frontend_ip_configuration {
    name                 = "PublicIPAddress"
    public_ip_address_id = "${azurestack_public_ip.test.id}"
  }
}

resource "azurestack_lb_probe" "test" {
  resource_group_name = "${azurestack_resource_group.test.name}"
  loadbalancer_id     = "${azurestack_lb.test.id}"
  name                = "ssh-running-probe"
  port                = 22
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the Probe.
- `resource_group_name` - (Required) The name of the resource group in which to create the resource.
- `loadbalancer_id` - (Required) The ID of the LoadBalancer in which to create the NAT Rule.
- `protocol` - (Optional) Specifies the protocol of the end point. Possible values are `Http` or `Tcp`. If Tcp is specified, a received ACK is required for the probe to be successful. If Http is specified, a 200 OK response from the specified URI is required for the probe to be successful.
- `port` - (Required) Port on which the Probe queries the backend endpoint. Possible values range from 1 to 65535, inclusive.
- `request_path` - (Optional) The URI used for requesting health status

from the backend endpoint. Required if protocol is set to Http. Otherwise, it is not allowed.

- `interval_in_seconds` - (Optional) The interval, in seconds between probes to the backend endpoint for health status. The default value is 15, the minimum value is 5.
- `number_of_probes` - (Optional) The number of failed probe attempts after which the backend endpoint is removed from rotation. The default value is 2. NumberOfProbes multiplied by intervalInSeconds value must be greater or equal to 10.Endpoints are returned to rotation when at least one probe is successful.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the LoadBalancer to which the resource is attached.

» Import

Load Balancer Probes can be imported using the `resource id`, e.g.

```
terraform import azurestack_lb_probe.test /subscriptions/00000000-0000-0000-0000-0000000000
```

» azurestack_lb_nat_pool

Manages a Load Balancer NAT pool.

NOTE When using this resource, the Load Balancer needs to have a FrontEnd IP Configuration Attached

» Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "LoadBalancerRG"
  location = "West US"
}

resource "azurestack_public_ip" "test" {
  name                         = "PublicIPForLB"
  location                     = "West US"
  resource_group_name          = "${azurestack_resource_group.test.name}"
  public_ip_address_allocation = "static"
}
```

```
resource "azurestack_lb" "test" {
  name                = "TestLoadBalancer"
  location            = "West US"
  resource_group_name = "${azurestack_resource_group.test.name}"

  frontend_ip_configuration {
    name                 = "PublicIPAddress"
    public_ip_address_id = "${azurestack_public_ip.test.id}"
  }
}

resource "azurestack_lb_nat_pool" "test" {
  resource_group_name            = "${azurestack_resource_group.test.name}"
  loadbalancer_id                = "${azurestack_lb.test.id}"
  name                           = "SampleApplicationPool"
  protocol                       = "Tcp"
  frontend_port_start            = 80
  frontend_port_end              = 81
  backend_port                   = 8080
  frontend_ip_configuration_name = "PublicIPAddress"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the NAT pool.
- `resource_group_name` - (Required) The name of the resource group in which to create the resource.
- `loadbalancer_id` - (Required) The ID of the Load Balancer in which to create the NAT pool.
- `frontend_ip_configuration_name` - (Required) The name of the frontend IP configuration exposing this rule.
- `protocol` - (Required) The transport protocol for the external endpoint. Possible values are `Udp` or `Tcp`.
- `frontend_port_start` - (Required) The first port number in the range of external ports that will be used to provide Inbound Nat to NICs associated with this Load Balancer. Possible values range between 1 and 65534, inclusive.
- `frontend_port_end` - (Required) The last port number in the range of external ports that will be used to provide Inbound Nat to NICs associated with this Load Balancer. Possible values range between 1 and 65534, inclusive.
- `backend_port` - (Required) The port used for the internal endpoint. Pos-

sible values range between 1 and 65535, inclusive.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Load Balancer to which the resource is attached.

» Import

Load Balancer NAT Pools can be imported using the `resource id`, e.g.

```
terraform import azurestack_lb_nat_pool.test /subscriptions/00000000-0000-0000-0000-0000000
```