Azure Stack Provider

The Azure Stack Provider is used to manage resources in Azure Stack via the Azure Resource Manager API's.

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

Creating Credentials

Terraform supports authenticating to Azure Stack using the Azure CLI (/docs/providers/azurestack/auth/azure_cli.html) or a Service Principal (either using a Client Secret (/docs/providers/azurestack/auth/service_principal_client_secret.html) or a Client Certificate (/docs/providers/azurestack/auth/service_principal_client_certificate.html)).

Example Usage

```
# Configure the Azure Stack Provider
provider "azurestack" {
  # NOTE: we recommend pinning the version of the Provider which should be used in the Provider block
  # version = "=0.5.0"
}

# Create a resource group
resource "azurestack_resource_group" "test" {
  name     = "production"
  location = "West US"
}

# Create a virtual network within the resource group
resource "azurestack_virtual_network" "test" {
  name                = "production-network"
  address_space       = ["10.0.0.0/16"]
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"

  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"
  }
}
```

Argument Reference

The following arguments are supported:

- `arm_endpoint` - (Optional) The Azure Resource Manager Endpoint for your Azure Stack instance, for example `https://management.westus.mydomain.com`. This can also be sourceed from the `ARM_ENDPOINT` Environment Variable.
- `client_id` - (Optional) The Client ID which should be used. This can also be sourceed from the `ARM_CLIENT_ID` Environment Variable.
- `subscription_id` - (Optional) The Subscription ID which should be used. This can also be sourced from the `ARM_SUBSCRIPTION_ID` Environment Variable.
- `tenant_id` - (Optional) The Tenant ID which should be used. This can also be sourced from the `ARM_TENANT_ID` Environment Variable.

When authenticating as a Service Principal using a Client Certificate, the following fields can be set:

- `client_certificate_password` - (Optional) The password associated with the Client Certificate. This can also be sourced from the `ARM_CLIENT_CERTIFICATE_PASSWORD` Environment Variable.
- `client_certificate_path` - (Optional) The path to the Client Certificate associated with the Service Principal which should be used. This can also be sourced from the `ARM_CLIENT_CERTIFICATE_PATH` Environment Variable.

More information on how to configure a Service Principal using a Client Certificate can be found in this guide (/docs/providers/azurestack/auth/service_principal_client_certificate.html).

When authenticating as a Service Principal using a Client Secret, the following fields can be set:

- `client_secret` - (Optional) The Client Secret which should be used. This can also be sourced from the `ARM_CLIENT_SECRET` Environment Variable.

More information on how to configure a Service Principal using a Client Secret can be found in this guide (/docs/providers/azurestack/auth/service_principal_client_secret.html).

For some advanced scenarios, such as where more granular permissions are necessary - the following properties can be set:

- `skip_credentials_validation` - (Optional) Should the Azure Stack Provider skip verifying the credentials being used are valid? This can also be sourced from the `ARM_SKIP_CREDENTIALS_VALIDATION` Environment Variable. Defaults to `false`.
- `skip_provider_registration` - (Optional) Should the Azure Stack Provider skip registering any required Resource Providers? This can also be sourced from the `ARM_SKIP_PROVIDER_REGISTRATION` Environment Variable. Defaults to `false`.

Testing

The following Environment Variables must be set to run the acceptance tests:

> **NOTE:** The Acceptance Tests require the use of a Service Principal using a Client Secret.

- `ARM_ENDPOINT` - The Azure Resource Manager API Endpoint for Azure Stack.
- `ARM_SUBSCRIPTION_ID` - The ID of the Azure Subscription in which to run the Acceptance Tests.
- `ARM_CLIENT_ID` - The Client ID of the Service Principal.
- `ARM_CLIENT_SECRET` - The Client Secret associated with the Service Principal.
- `ARM_TENANT_ID` - The Tenant ID to use.
- `ARM_TEST_LOCATION` - The Azure Stack Location to provision resources in for the Acceptance Tests.

Azure Stack Provider: Authenticating using the Azure CLI

Terraform supports authenticating to Azure Stack using the Azure CLI, a Service Principal (either using a Client Secret or a Client Certificate).

Terraform supports authenticating to Azure Stack using the Azure CLI (which is detailed in this guide) or a Service Principal, either using a Client Secret (/docs/providers/azurestack/auth/service_principal_client_secret.html) or using a Client Certificate (/docs/providers/azurestack/auth/service_principal_client_certificate.html).

NOTE: Authenticating via the Azure CLI is only supported when using a User Account. If you're using a Service Principal (for example via `az login --service-principal`) you should instead authenticate via the Service Principal directly (/docs/providers/azurestack/auth/service_principal_client_secret.html).

This guide assumes that the Certificate being used for Azure Stack is valid, or has been trusted on your machine (instructions for trusting the Azure Stack Certificate can be found here (https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-version-profiles-azurecli2#trust-the-azure-stack-ca-root-certificate)).

We need to add the details for your Azure Stack Configuration to the Azure CLI:

```
$ az cloud register -n AzureStack --endpoint-resource-manager "https://management.region.mycloud.com" --s
uffix-storage-endpoint "region.mycloud.com" --suffix-keyvault-dns ".vault.region.mycloud.com"
```

Note: the values used will differ from the ones specified in the example above - as such you may need to contact your Azure Stack Administrator to determine the values required to connect to your Azure Stack instance.

Once that's done we can now switch to using the newly registered `AzureStack` cloud:

```
$ az cloud set --name AzureStack
```

Next you'll need to configure the Profile used for Azure Stack:

```
$ az cloud update --profile 2018-03-01-hybrid
```

NOTE: If you're using a version of Azure Stack prior to build 1808 - you'll need to use the Profile version `2017-03-09-profile`.

At this point we should be able to log into the Azure Stack instance using the Azure CLI:

```
$ az login
```

NOTE: Authenticating via the Azure CLI is only supported when using a User Account. If you're using a Service Principal (for example via `az login --service-principal`) you should instead authenticate via the Service Principal directly (/docs/providers/azurestack/auth/service_principal_client_secret.html).

This will prompt you to open a web browser, as shown below:

```
$ az login
Note, we have launched a browser for you to login. For old experience with device code, use "az login --u
se-device-code
```

Once logged in, it's possible to list the Subscriptions associated with the account via:

```
$ az account list
```

The output (similar to below) will display one or more Subscriptions:

```
[
  {
    "cloudName": "AzureStack",
    "id": "00000000-0000-0000-0000-000000000000",
    "isDefault": true,
    "name": "Example Subscription",
    "state": "Enabled",
    "tenantId": "00000000-0000-0000-0000-000000000000",
    "user": {
      "name": "user@example.com",
      "type": "user"
    }
  }
]
```

In the snippet above, `id` refers to the Subscription ID and `isDefault` refers to whether this Subscription is configured as the default.

Note: When authenticating via the Azure CLI, Terraform will automatically connect to the Default Subscription. Therefore, if you have multiple subscriptions on the account, you may need to set the Default Subscription, via:

```
$ az account set --subscription="SUBSCRIPTION_ID"
```

Configuring the Provider block when using the Azure CLI

When authenticating using the Azure CLI most of the details required can be inferred, as such to use the Default Subscription configured in the Azure CLI you should be able to use the following Provider Block:

```
provider "azurestack" {
  # whilst the version attribute is optional, we recommend pinning the Provider version being used
  version = "=0.5.0"
}
```

If you're using multiple subscriptions (or have access to multiple tenants) - it's possible to specify the Subscription ID to target a specific Subscription:

```
provider "azurestack" {
  # whilst the version attribute is optional, we recommend pinning the Provider version being used
  version         = "=0.5.0"
  subscription_id = "00000000-0000-0000-0000-000000000000"
}
```

More information on the fields supported in the Provider block can be found here (/docs/providers/azurestack/index.html#argument-reference).

NOTE: If you're previously authenticated using a Service Principal (configured via Environment Variables) - you must remove the `ARM_*` prefixed Environment Variables in order to be able to authenticate using the Azure CLI.

Azure Stack Provider: Authenticating using a Service Principal using a Client Certificate

Terraform supports authenticating to Azure Stack using the Azure CLI (/docs/providers/azurestack/auth/azure_cli.html) or a Service Principal, either using a Client Secret (/docs/providers/azurestack/auth/service_principal_client_secret.html) or using a Client Certificate (which is detailed in this guide).

Creating a Service Principal

A Service Principal is an application within Azure Active Directory which can have authentication tokens associated with it. In this example we'll generate a new certificate, then create and assign it to a Service Principal; so that it can be used for authentication.

Creating a Service Principal in the Azure Portal

NOTE: This needs to be completed in the main Azure (Public) Portal - not the Azure Stack Portal.

There are three tasks needed to create a Service Principal via the Azure Portal (https://portal.azure.com):

1. Generating a Certificate which can be used for Authentication
2. Create an Application in Azure Active Directory (which acts as a Service Principal) and then associating the Certificate with it
3. Grant the Application access to manage resources in your Azure Subscription

1. Generating a Certificate

Firstly we need to create a certificate which can be used for authentication. To do that we're going to generate a Certificate Signing Request (also known as a CSR) using `openssl` (this can also be achieved using PowerShell, however that's outside the scope of this document):

```
$ openssl req \
   -newkey rsa:4096 -nodes -keyout "service-principal.key" \
   -out "service-principal.csr"
```

We can now sign that Certificate Signing Request, in this example we're going to self-sign this certificate using the Key we just generated; however it's also possible to do this using a Certificate Authority. In order to do that we're again going to use `openssl`:

```
$ openssl x509 \
  -signkey "service-principal.key" \
  -in "service-principal.csr" \
  -req -days 365 -out "service-principal.crt"
```

Finally we can generate a PFX file which can be used to authenticate with Azure:

```
$ openssl pkcs12 -export -out "service-principal.pfx" -inkey "service-principal.key" -in "service-princip
al.crt"
```

Now that we've generated a certificate, we can create the Azure Active Directory application.

2. Creating an Application in Azure Active Directory

Firstly navigate to the **Azure Active Directory** overview (https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/Overview) within the Azure Portal - then select the **App Registration** blade (https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps/RegisteredApps/Overview) and click **Endpoints** at the top of the **App Registration** blade. A list of URIs will be displayed and you need to locate the URI for **OAUTH 2.0 AUTHORIZATION ENDPOINT** which contains a GUID. This is your Tenant ID / the `tenant_id` field mentioned above.

Next, navigate back to the **App Registration** blade (https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps/RegisteredApps/Overview) - from here we'll create the Application in Azure Active Directory. To do this click **Add** at the top to add a new Application within Azure Active Directory. On this page, set the following values then press **Create**:

- **Name** - this is a friendly identifier and can be anything (e.g. "Terraform")
- **Application Type** - this should be set to "Web app / API"
- **Sign-on URL** - this can be anything, providing it's a valid URI (e.g. https://terra.form (https://terra.form))

Once that's done - select the Application you just created in the **App Registration** blade (https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps/RegisteredApps/Overview). At the top of this page, the "Application ID" GUID is the `client_id` you'll need.

Finally need to associate the Client Certificate with the Azure Active Directory Application - to do this select **Settings** and then **Keys**. This screen displays the Passwords (Client Secrets) and Public Keys (Client Certificates) which are associated with this Azure Active Directory Application.

The Public Key associated with the generated Certificate can be uploaded by selecting **Upload Public Key**, selecting the file which should be uploaded (in the example above, this'd be `service-principal.crt`) - and then hitting **Save**.

3. Granting the Application access to manage resources in your Azure and Azure Stack Subscriptions

Once the Application exists in Azure Active Directory - we can grant it permissions to modify resources in the Subscription. To do this, navigate to the **Subscriptions** blade within the Azure Portal (https://portal.azure.com/#blade/Microsoft_Azure_Billing/SubscriptionsBlade), then select the Subscription you wish to use, then click **Access Control (IAM)**, and finally **Add**.

NOTE: This will only give SPN access to your Azure Subscription - This is **NOT** required to interact with Azure Stack. To allow SPN access to Azure Stack you need to do it under Azure Stack Subscription navigate to the **Subscriptions** blade within the Azure Stack Portal (https://portal.{region}.{domain}/#blade/Microsoft_Azure_Billing/SubscriptionsBlade), then select the Subscription you wish to use, then click **Access Control (IAM)**, and finally **Add**.

Firstly, specify a Role which grants the appropriate permissions needed for the Service Principal (for example, `Contributor` will grant Read/Write on all resources in the Subscription). There's more information about the built in roles available here (https://azure.microsoft.com/en-gb/documentation/articles/role-based-access-built-in-roles/).

Secondly, search for and select the name of the Application created in Azure Active Directory to assign it this role - then press **Save**.

At this point the newly created Azure Active Directory Application should be associated with the Certificate that we generated earlier (which can be used as a Client Certificate) - and should have permissions to the Azure Subscription.

It should then be possible to configure these credentials in Terraform, either by using setting the relevant Environment Variables:

```
export ARM_ENDPOINT="https://management.region.mycloud.com"
export ARM_CLIENT_CERTIFICATE_PASSWORD="hello-world"
export ARM_CLIENT_CERTIFICATE_PATH="/Users/myuser/keys/service-principal.pfx"
export ARM_CLIENT_ID="00000000-0000-0000-0000-000000000000"
export ARM_SUBSCRIPTION_ID="00000000-0000-0000-0000-000000000000"
export ARM_TENANT_ID="00000000-0000-0000-0000-000000000000"
```

and then using the following Provider Block:

```
provider "azurestack" {
  # whilst the `version` attribute is optional, we'd recommend pinning to a particular version
  version = "=0.5.0"
}
```

Alternatively you can define these fields within the Provider Block:

```
provider "azurestack" {
  # whilst the `version` attribute is optional, we'd recommend pinning to a particular version
  version = "=0.5.0"

  arm_endpoint                = "https://management.region.mycloud.com"
  client_id                   = "00000000-0000-0000-0000-000000000000"
  client_certificate_password = "my-password"
  client_certificate_path     = "./service-principal.pfx"
  subscription_id             = "00000000-0000-0000-0000-000000000000"
  tenant_id                   = "00000000-0000-0000-0000-000000000000"
}
```

More information on the fields supported in the Provider block can be found here (/docs/providers/azurestack/index.html#argument-reference).

Azure Stack Provider: Authenticating using a Service Principal using a Client Secret

Terraform supports authenticating to Azure Stack using the Azure CLI (/docs/providers/azurestack/auth/azure_cli.html) or a Service Principal, either using a Client Secret (which is detailed in this guide) or using a Client Certificate (/docs/providers/azurestack/auth/service_principal_client_certificate.html).

Creating a Service Principal

A Service Principal is an application within Azure Active Directory whose authentication tokens can be used as the `client_id`, `client_secret`, and `tenant_id` fields needed by Terraform (`subscription_id` can be independently recovered from your Azure account details).

Creating a Service Principal in the Azure Portal

NOTE: This needs to be completed in the main Azure (Public) Portal - not the Azure Stack Portal.

There are two tasks needed to create a Service Principal via the Azure Portal (https://portal.azure.com):

1. Create an Application in Azure Active Directory (which acts as a Service Principal)
2. Grant the Application access to manage resources in your Azure Subscription

1. Creating an Application in Azure Active Directory

Firstly navigate to the **Azure Active Directory** overview (https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/Overview) within the Azure Portal - then select the **App Registration** blade (https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps/RegisteredApps/Overview) and click **Endpoints** at the top of the **App Registration** blade. A list of URIs will be displayed and you need to locate the URI for **OAUTH 2.0 AUTHORIZATION ENDPOINT** which contains a GUID. This is your Tenant ID / the `tenant_id` field mentioned above.

Next, navigate back to the **App Registration** blade (https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps/RegisteredApps/Overview) - from here we'll create the Application in Azure Active Directory. To do this click **Add** at the top to add a new Application within Azure Active Directory. On this page, set the following values then press **Create**:

- **Name** - this is a friendly identifier and can be anything (e.g. "Terraform")
- **Application Type** - this should be set to "Web app / API"
- **Sign-on URL** - this can be anything, providing it's a valid URI (e.g. https://terra.form (https://terra.form))

Finally need to create a Password for the Azure Active Directory Application - to do this select **Settings** and then **Keys**. This screen displays the Passwords (Client Secrets) and Public Keys (Client Certificates) which are associated with this Azure Active Directory Application.

Enter a description for the Key and select when this password should expire - and then press **Save**. At this point the Password should be displayed - you'll need to copy it now, since it's only displayed once - which is the `client_secret`.

2. Granting the Application access to manage resources in your Azure and Azure Stack Subscriptions

Once the Application exists in Azure Active Directory - we can grant it permissions to modify resources in the Subscription. To do this, navigate to the **Subscriptions** blade within the Azure Portal (https://portal.azure.com/#blade/Microsoft_Azure_Billing/SubscriptionsBlade), then select the Subscription you wish to use, then click **Access Control (IAM)**, and finally **Add**.

NOTE: This will only give SPN access to your Azure Subscription - This is **NOT** required to interact with Azure Stack. To allow SPN access to Azure Stack you need to do it under Azure Stack Subscription navigate to the **Subscriptions** blade within the Azure Stack Portal (https://portal.{region}.{domain}/#blade/Microsoft_Azure_Billing/SubscriptionsBlade), then select the Subscription you wish to use, then click **Access Control (IAM)**, and finally **Add**.

Firstly, specify a Role which grants the appropriate permissions needed for the Service Principal (for example, `Contributor` will grant Read/Write on all resources in the Subscription). There's more information about the built in roles available here (https://azure.microsoft.com/en-gb/documentation/articles/role-based-access-built-in-roles/).

Secondly, search for and select the name of the Application created in Azure Active Directory to assign it this role - then press **Save**.

3. Configuring the Service Principal in Terraform

As we've obtained the credentials for this Service Principal - it's possible to configure it in a few different ways.

When storing the credentials as Environment Variables, for example:

```
$ export ARM_ENDPOINT="00000000-0000-0000-0000-000000000000"
$ export ARM_CLIENT_ID="00000000-0000-0000-0000-000000000000"
$ export ARM_CLIENT_SECRET="00000000-0000-0000-0000-000000000000"
$ export ARM_SUBSCRIPTION_ID="00000000-0000-0000-0000-000000000000"
$ export ARM_TENANT_ID="00000000-0000-0000-0000-000000000000"
```

The following Provider block can be specified - where `0.5.0` is the version of the Azure Stack Provider that you'd like to use:

```
provider "azurestack" {
  # Whilst version is optional, we /strongly recommend/ using it to pin the version of the Provider being
 used
  version = "=0.5.0"
}
```

More information on the fields supported in the Provider block can be found here (/docs/providers/azurestack/index.html#argument-reference).

It's also possible to configure these variables either in-line or from using variables in Terraform (as the `client_secret` is in this example), like so:

NOTE: We'd recommend not defining these variables in-line since they could easily be checked into Source Control.

```
variable "client_secret" {}

provider "azurestack" {
  # Whilst version is optional, we /strongly recommend/ using it to pin the version of the Provider being
 used
  version = "=0.5.0"

  arm_endpoint    = "https://management.region.myazurestack.com"
  subscription_id = "00000000-0000-0000-0000-000000000000"
  client_id       = "00000000-0000-0000-0000-000000000000"
  client_secret   = "${var.client_secret}"
  tenant_id       = "00000000-0000-0000-0000-000000000000"
}
```

More information on the fields supported in the Provider block can be found here (/docs/providers/azurestack/index.html#argument-reference).

Data Source: azurestack_client_config

Use this data source to access the configuration of the Azure Stack provider.

Example Usage

```
data "azurestack_client_config" "current" {}

output "account_id" {
  value = "${data.azurestack_client_config.current.service_principal_application_id}"
}
```

Argument Reference

There are no arguments available for this data source.

Attributes Reference

- `client_id` is set to the Azure Client ID (Application Object ID).
- `tenant_id` is set to the Azure Tenant ID.
- `subscription_id` is set to the Azure Subscription ID.
- `service_principal_application_id` is the Service Principal Application ID.
- `service_principal_object_id` is the Service Principal Object ID.

Note: To better understand "application" and "service principal", please read Application and service principal objects in Azure Active Directory (https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-application-objects).

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_platform_image

Use this data source to access information about a Platform Image.

Example Usage

```
data "azurestack_platform_image" "test" {
  location  = "West Europe"
  publisher = "Canonical"
  offer     = "UbuntuServer"
  sku       = "16.04-LTS"
}

output "version" {
  value = "${data.azurestack_platform_image.test.version}"
}
```

Argument Reference

- `location` - (Required) Specifies the Location to pull information about this Platform Image from.
- `publisher` - (Required) Specifies the Publisher associated with the Platform Image.
- `offer` - (Required) Specifies the Offer associated with the Platform Image.
- `sku` - (Required) Specifies the SKU of the Platform Image.

Attributes Reference

- `id` - The ID of the Platform Image.
- `version` - The latest version of the Platform Image.

Data Source: azurestack_public_ip

Use this data source to access the properties of an existing Azure Public IP Address.

Example Usage (reference an existing)

```
data "azurestack_public_ip" "test" {
  name                = "name_of_public_ip"
  resource_group_name = "name_of_resource_group"
}

output "domain_name_label" {
  value = "${data.azurestack_public_ip.test.domain_name_label}"
}

output "public_ip_address" {
  value = "${data.azurestack_public_ip.test.ip_address}"
}
```

Example Usage (Retrieve the Dynamic Public IP of a new VM)

```
resource "azurestack_resource_group" "test" {
  name     = "test-resources"
  location = "West US 2"
}

resource "azurestack_virtual_network" "test" {
  name                = "test-network"
  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_public_ip" "test" {
  name                         = "test-pip"
  location                     = "${azurestack_resource_group.test.location}"
  resource_group_name          = "${azurestack_resource_group.test.name}"
  public_ip_address_allocation = "Dynamic"
  idle_timeout_in_minutes      = 30

  tags = {
    environment = "test"
  }
}
```

```
resource "azurestack_network_interface" "test" {
  name                = "test-nic"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"

  ip_configuration {
    name                          = "testconfiguration1"
    subnet_id                     = "${azurestack_subnet.test.id}"
    private_ip_address_allocation = "static"
    private_ip_address            = "10.0.2.5"
    public_ip_address_id          = "${azurestack_public_ip.test.id}"
  }
}

resource "azurestack_virtual_machine" "test" {
  name                  = "test-vm"
  location              = "${azurestack_resource_group.test.location}"
  resource_group_name   = "${azurestack_resource_group.test.name}"
  network_interface_ids = ["${azurestack_network_interface.test.id}"]

  # ...
}

data "azurestack_public_ip" "test" {
  name                = "${azurestack_public_ip.test.name}"
  resource_group_name = "${azurestack_virtual_machine.test.resource_group_name}"
}

output "public_ip_address" {
  value = "${data.azurestack_public_ip.test.ip_address}"
}
```

Argument Reference

- `name` - (Required) Specifies the name of the public IP address.
- `resource_group_name` - (Required) Specifies the name of the resource group.

Attributes Reference

- `domain_name_label` - The label for the Domain Name.
- `idle_timeout_in_minutes` - Specifies the timeout for the TCP idle connection.
- `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.
- `ip_address` - The IP address value that was allocated.
- `tags` - A mapping of tags to assigned to the resource.

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_route_table

Gets information about a Route Table

Example Usage

```
data "azurestack_route_table" "test" {
  name                = "myroutetable"
  resource_group_name = "some-resource-group"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the Route Table.
- `resource_group_name` - (Required) The name of the Resource Group in which the Route Table exists.

Attributes Reference

The following attributes are exported:

- `id` - The Route Table ID.
- `location` - The Azure Region in which the Route Table exists.
- `route` - One or more `route` blocks as documented below.
- `subnets` - The collection of Subnets associated with this route table.
- `tags` - A mapping of tags assigned to the Route Table.

The `route` block exports the following:

- `name` - The name of the Route.
- `address_prefix` - The destination CIDR to which the route applies.
- `next_hop_type` - The type of Azure hop the packet should be sent to.
- `next_hop_in_ip_address` - Contains the IP address packets should be forwarded to.

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 (https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-acs-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_subnet

Use this data source to access the properties of an Azure Subnet located within a Virtual Network.

Example Usage

```
data "azurestack_subnet" "test" {
  name                 = "backend"
  virtual_network_name = "production"
  resource_group_name  = "networking"
}

output "subnet_id" {
  value = "${data.azurestack_subnet.test.id}"
}
```

Argument Reference

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

Attributes Reference

- `id` - The ID of the Subnet.
- `address_prefix` - The address prefix used for the subnet.
- `network_security_group_id` - The ID of the Network Security Group associated with the subnet.
- `route_table_id` - The ID of the Route Table associated with this subnet.
- `ip_configurations` - The collection of IP Configurations with IPs within this subnet.

Data Source: azurestack_virtual_network_gateway

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

Example Usage

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

output "virtual_network_gateway_id" {
  value = "${data.azurestack_virtual_network_gateway.test.id}"
}
```

Argument Reference

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

Attributes Reference

- `id` - The ID of the Virtual Network Gateway.
- `location` - The location/region where the Virtual Network Gateway is located.
- `type` - The type of the Virtual Network Gateway.
- `vpn_type` - The routing type of the Virtual Network Gateway.
- `enable_bgp` - Will BGP (Border Gateway Protocol) will be enabled for this Virtual Network Gateway.
- `default_local_network_gateway_id` - The ID of the local network gateway through which outbound Internet traffic from the virtual network in which the gateway is created will be routed (*forced tunneling*). Refer to the Azure documentation on forced tunneling (https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-forced-tunneling-rm).
- `sku` - Configuration of the size and capacity of the Virtual Network Gateway.
- `ip_configuration` - One or two `ip_configuration` blocks documented below.
- `vpn_client_configuration` - A `vpn_client_configuration` block which is documented below.
- `tags` - A mapping of tags assigned to the resource.

The `ip_configuration` block supports:

- `name` - A user-defined name of the IP configuration.
- `private_ip_address_allocation` - Defines how the private IP address of the gateways virtual interface is assigned.
- `subnet_id` - The ID of the gateway subnet of a virtual network in which the virtual network gateway will be created. It is mandatory that the associated subnet is named `GatewaySubnet`. Therefore, each virtual network can contain at most a single Virtual Network Gateway.
- `public_ip_address_id` - The ID of the Public IP Address associated with the Virtual Network Gateway.

The `vpn_client_configuration` block supports:

- `address_space` - The address space out of which ip addresses for vpn clients will be taken. You can provide more than one address space, e.g. in CIDR notation.
- `root_certificate` - One or more `root_certificate` blocks which are defined below. These root certificates are used to sign the client certificate used by the VPN clients to connect to the gateway.
- `revoked_certificate` - One or more `revoked_certificate` blocks which are defined below.

The `bgp_settings` block supports:

- `asn` - The Autonomous System Number (ASN) to use as part of the BGP.
- `peering_address` - The BGP peer IP address of the virtual network gateway. This address is needed to configure the created gateway as a BGP Peer on the on-premises VPN devices.
- `peer_weight` - The weight added to routes which have been learned through BGP peering.

The `root_certificate` block supports:

- `name` - The user-defined name of the root certificate.
- `public_cert_data` - The public certificate of the root certificate authority. The certificate must be provided in Base-64 encoded X.509 format (PEM).

The `root_revoked_certificate` block supports:

- `name` - The user-defined name of the revoked certificate.
- `public_cert_data` - The SHA1 thumbprint of the certificate to be revoked.

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_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 (https://github.com/MicrosoftDocs/azure-docs/blob/master/includes/managed-disks-common-fault-domain-region-list.md).

- `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 (https://github.com/MicrosoftDocs/azure-docs/blob/master/includes/managed-disks-common-fault-domain-region-list.md).

- `managed` - (Optional) Specifies whether the availability set is managed or not. Possible values are `true` (to specify aligned) or `false` (to specify classic). Default is `false`.

- `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/00000000-0000-0000-0000-000000000000/r
esourceGroups/mygroup1/providers/Microsoft.Compute/availabilitySets/webAvailSet
```

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/00000000-0000-0000-0000-000000000000/resourc
eGroups/mygroup1/providers/Microsoft.Network/dnsZones/zone1/A/myrecord1
```

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/00000000-0000-0000-0000-000000000000/resourceGr
oups/mygroup1/providers/Microsoft.Network/dnsZones/zone1
```

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/00000000-0000-0000-0000-000000000000/resourceGroups/group1/providers/Microsoft.Network/loadBalancers/lb1/backendAddressPools/pool1
```

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/resourceGroups/group1/providers/Microsoft.Network/loadBalancers/lb1
```

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. Possible 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/00000000-0000-0000-0000-000000000000/resourceGroups/group1/providers/Microsoft.Network/loadBalancers/lb1/inboundNatPools/pool1
```

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/00000000-0000-0000-0000-000000000000/resource
Groups/group1/providers/Microsoft.Network/loadBalancers/lb1/inboundNatRules/rule1
```

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/00000000-0000-0000-0000-000000000000/resourceGroups/group1/providers/Microsoft.Network/loadBalancers/lb1/probes/probe1
```

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/resourceGrou
ps/group1/providers/Microsoft.Network/loadBalancers/lb1/loadBalancingRules/rule1
```

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/00000000-0000-0000-0000-00000000000
0/resourceGroups/mygroup1/providers/Microsoft.Network/localNetworkGateways/lng1
```

azurestack_managed_disk

Manage a managed disk.

Example Usage with Create Empty

```
resource "azurestack_resource_group" "test" {
  name     = "acctestRG"
  location = "West US 2"
}

resource "azurestack_managed_disk" "test" {
  name                 = "acctestmd"
  location             = "West US 2"
  resource_group_name  = "${azurestack_resource_group.test.name}"
  storage_account_type = "Standard_LRS"
  create_option        = "Empty"
  disk_size_gb         = "1"

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

Example Usage with Create Copy

```
resource "azurestack_resource_group" "test" {
  name     = "acctestRG"
  location = "West US 2"
}

resource "azurestack_managed_disk" "source" {
  name                 = "acctestmd1"
  location             = "West US 2"
  resource_group_name  = "${azurestack_resource_group.test.name}"
  storage_account_type = "Standard_LRS"
  create_option        = "Empty"
  disk_size_gb         = "1"

  tags = {
    environment = "staging"
  }
}

resource "azurestack_managed_disk" "copy" {
  name                 = "acctestmd2"
  location             = "West US 2"
  resource_group_name  = "${azurestack_resource_group.test.name}"
  storage_account_type = "Standard_LRS"
  create_option        = "Copy"
  source_resource_id   = "${azurestack_managed_disk.source.id}"
  disk_size_gb         = "1"

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

Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the managed disk. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the managed disk.
- `location` - (Required) Specified the supported Azure location where the resource exists. Changing this forces a new resource to be created.
- `storage_account_type` - (Required) The type of storage to use for the managed disk. Allowable values are `Standard_LRS` or `Premium_LRS`.
- `create_option` - (Required) The method to use when creating the managed disk. Possible values include:
  - `Import` - Import a VHD file in to the managed disk (VHD specified with `source_uri`).
  - `Empty` - Create an empty managed disk.
  - `Copy` - Copy an existing managed disk or snapshot (specified with `source_resource_id`).
  - `FromImage` - Copy a Platform Image (specified with `image_reference_id`)

- `source_uri` - (Optional) URI to a valid VHD file to be used when `create_option` is `Import`.
- `source_resource_id` - (Optional) ID of an existing managed disk to copy `create_option` is `Copy`.
- `image_reference_id` - (Optional) ID of an existing platform/marketplace disk image to copy when `create_option` is `FromImage`.
- `os_type` - (Optional) Specify a value when the source of an `Import` or `Copy` operation targets a source that contains an operating system. Valid values are `Linux` or `Windows`
- `disk_size_gb` - (Optional, Required for a new managed disk) Specifies the size of the managed disk to create in gigabytes. If `create_option` is `Copy` or `FromImage`, then the value must be equal to or greater than the source's size.
- `tags` - (Optional) A mapping of tags to assign to the resource.

For more information on managed disks, such as sizing options and pricing, please check out the azure documentation (https://docs.microsoft.com/en-us/azure/storage/storage-managed-disks-overview).

Attributes Reference

The following attributes are exported:

- `id` - The managed disk ID.

Import

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

```
terraform import azurestack_managed_disk.test /subscriptions/00000000-0000-0000-0000-000000000000/resourc
eGroups/mygroup1/providers/microsoft.compute/disks/manageddisk1
```

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 (https://docs.microsoft.com/en-us/azure/virtual-network/create-network-security-group-preview)

- `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/00000000-0000-0000-0000-000000000000/re
sourceGroups/mygroup1/providers/microsoft.network/networkInterfaces/nic1
```

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 (/docs/providers/azurestack/r/network_security_rule.html), and allows for Network Security Rules to be defined in-line within the Network Security Group resource (/docs/providers/azurestack/r/network_security_group.html). 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/00000000-0000-0000-0000-00000000
0000/resourceGroups/mygroup1/providers/Microsoft.Network/networkSecurityGroups/mySecurityGroup
```

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 (/docs/providers/azurestack/r/network_security_rule.html), and allows for Network Security Rules to be defined in-line within the Network Security Group resource (/docs/providers/azurestack/r/network_security_group.html). 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/00000000-0000-0000-0000-000000000000/resourceGroups/mygroup1/providers/Microsoft.Network/networkSecurityGroups/mySecurityGroup/securityRules/rule1
```

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/00000000-0000-0000-0000-000000000000/reso
urceGroups/mygroup1/providers/Microsoft.Network/publicIPAddresses/myPublicIpAddress1
```

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 (http://azure.microsoft.com/en-us/regions/) 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/00000000-0000-0000-0000-000000000000/resourceGroups/myresourcegroup
```

azurestack_route

Manages a Route within a Route Table.

Example Usage

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

resource "azurestack_route_table" "test" {
  name                = "acceptanceTestRouteTable1"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"
}

resource "azurestack_route" "test" {
  name                = "acceptanceTestRoute1"
  resource_group_name = "${azurestack_resource_group.test.name}"
  route_table_name    = "${azurestack_route_table.test.name}"
  address_prefix      = "10.1.0.0/16"
  next_hop_type       = "vnetlocal"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the route. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the route. Changing this forces a new resource to be created.
- `route_table_name` - (Required) The name of the route table within which create the route. Changing this forces a new resource to be created.
- `address_prefix` - (Required) The destination CIDR to which the route applies, such as `10.1.0.0/16`
- `next_hop_type` - (Required) The type of Azure hop the packet should be sent to. Possible values are `VirtualNetworkGateway`, `VnetLocal`, `Internet`, `VirtualAppliance` and `None`
- `next_hop_in_ip_address` - (Optional) Contains the IP address packets should be forwarded to. Next hop values are only allowed in routes where the next hop type is `VirtualAppliance`.

Attributes Reference

The following attributes are exported:

- `id` - The Route ID.

Import

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

```
terraform import azurestack_route.testRoute /subscriptions/00000000-0000-0000-0000-000000000000/resourceG
roups/mygroup1/providers/Microsoft.Network/routeTables/mytable1/routes/myroute1
```

azurestack_route_table

Manages a Route Table

Example Usage

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

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

  disable_bgp_route_propagation = false

  route {
    name           = "route1"
    address_prefix = "10.1.0.0/16"
    next_hop_type  = "vnetlocal"
  }

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

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the route table. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the route table. 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.
- `route` - (Optional) Can be specified multiple times to define multiple routes. Each `route` block supports fields documented below.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `route` block supports:

- `name` - (Required) The name of the route.
- `address_prefix` - (Required) The destination CIDR to which the route applies, such as 10.1.0.0/16

- `next_hop_type` - (Required) The type of Azure hop the packet should be sent to. Possible values are `VirtualNetworkGateway`, `VnetLocal`, `Internet`, `VirtualAppliance` and `None`.
- `next_hop_in_ip_address` - (Optional) Contains the IP address packets should be forwarded to. Next hop values are only allowed in routes where the next hop type is `VirtualAppliance`.

Attributes Reference

The following attributes are exported:

- `id` - The Route Table ID.
- `subnets` - The collection of Subnets associated with this route table.

Import

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

```
terraform import azurestack_route_table.test /subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/mygroup1/providers/Microsoft.Network/routeTables/mytable1
```

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 (https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-acs-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 (https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-acs-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 (https://docs.microsoft.com/en-gb/azure/storage/blobs/storage-custom-domain-name)

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/00000000-0000-0000-0000-000000000000/resourceGroups/myresourcegroup/providers/Microsoft.Storage/storageAccounts/myaccount
```

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_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_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 (/docs/providers/azurestack/r/subnet.html), and allows for Subnets to be defined in-line within the Virtual Network resource (/docs/providers/azurestack/r/virtual_network.html). 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/00000000-0000-0000-0000-000000000000/resourc
eGroups/mygroup1/providers/Microsoft.Network/virtualNetworks/myvnet1/subnets/mysubnet1
```

azurestack_template_deployment

Manages a template deployment of resources

Note on ARM Template Deployments: Due to the way the underlying Azure API is designed, Terraform can only manage the deployment of the ARM Template - and not any resources which are created by it. This means that when deleting the `azurestack_template_deployment` resource, Terraform will only remove the reference to the deployment, whilst leaving any resources created by that ARM Template Deployment. One workaround for this is to use a unique Resource Group for each ARM Template Deployment, which means deleting the Resource Group would contain any resources created within it - however this isn't ideal. More information (https://docs.microsoft.com/en-us/rest/api/resources/deployments#Deployments_Delete).

Example Usage

Note: This example uses Storage Accounts (/docs/providers/azurestack/r/storage_account.html) and Public IP's (/docs/providers/azurestack/r/public_ip.html) which are natively supported by Terraform - we'd highly recommend using the Native Resources where possible instead rather than an ARM Template, for the reasons outlined above.

```
resource "azurestack_resource_group" "test" {
  name     = "acctestRG-01"
  location = "West US"
}

resource "azurestack_template_deployment" "test" {
  name                = "acctesttemplate-01"
  resource_group_name = "${azurestack_resource_group.test.name}"

  template_body = <<DEPLOY
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "storageAccountType": {
      "type": "string",
      "defaultValue": "Standard_LRS",
      "allowedValues": [
        "Standard_LRS",
        "Standard_GRS",
        "Standard_ZRS"
      ],
      "metadata": {
        "description": "Storage Account type"
      }
    }
  },
  "variables": {
    "location": "[resourceGroup().location]",
    "storageAccountName": "[concat(uniquestring(resourceGroup().id), 'storage')]",
    "publicIPAddressName": "[concat('myPublicIp', uniquestring(resourceGroup().id))]",
    "publicIPAddressType": "Dynamic",
```

```
    "apiVersion": "2015-06-15",
    "dnsLabelPrefix": "terraform-acctest"
  },
  "resources": [
    {
      "type": "Microsoft.Storage/storageAccounts",
      "name": "[variables('storageAccountName')]",
      "apiVersion": "[variables('apiVersion')]",
      "location": "[variables('location')]",
      "properties": {
        "accountType": "[parameters('storageAccountType')]"
      }
    },
    {
      "type": "Microsoft.Network/publicIPAddresses",
      "apiVersion": "[variables('apiVersion')]",
      "name": "[variables('publicIPAddressName')]",
      "location": "[variables('location')]",
      "properties": {
        "publicIPAllocationMethod": "[variables('publicIPAddressType')]",
        "dnsSettings": {
          "domainNameLabel": "[variables('dnsLabelPrefix')]"
        }
      }
    }
  ],
  "outputs": {
    "storageAccountName": {
      "type": "string",
      "value": "[variables('storageAccountName')]"
    }
  }
}
DEPLOY

  # these key-value pairs are passed into the ARM Template's `parameters` block
  parameters = {
    "storageAccountType" = "Standard_GRS"
  }

  deployment_mode = "Incremental"
}

output "storageAccountName" {
  value = "${azurestack_template_deployment.test.outputs["storageAccountName"]}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the template deployment. Changing this forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the template deployment.
- `deployment_mode` - (Required) Specifies the mode that is used to deploy resources. This value could be either

`Incremental` or `Complete`. Note that you will almost *always* want this to be set to `Incremental` otherwise the deployment will destroy all infrastructure not specified within the template, and Terraform will not be aware of this.

- `template_body` - (Optional) Specifies the JSON definition for the template.

Note: There's a `file` function available (https://www.terraform.io/docs/configuration/functions/file.html) which allows you to read this from an external file, which helps makes this more resource more readable.

- `parameters` - (Optional) Specifies the name and value pairs that define the deployment parameters for the template.
- `parameters_body` - (Optional) Specifies a valid Azure JSON parameters file that define the deployment parameters. It can contain KeyVault references

Note: There's an `file` interpolation function available (https://www.terraform.io/docs/configuration/interpolation.html#file-path-) which allows you to read this from an external file, which helps makes this more resource more readable.

Attributes Reference

The following attributes are exported:

- `id` - The Template Deployment ID.
- `outputs` - A map of supported scalar output types returned from the deployment (currently, Azure Template Deployment outputs of type String, Int and Bool are supported, and are converted to strings - others will be ignored) and can be accessed using `.outputs["name"]`.

Note

Terraform does not know about the individual resources created by Azure using a deployment template and therefore cannot delete these resources during a destroy. Destroying a template deployment removes the associated deployment operations, but will not delete the Azure resources created by the deployment. In order to delete these resources, the containing resource group must also be destroyed. More information (https://docs.microsoft.com/en-us/rest/api/resources/deployments#Deployments_Delete).

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_storage_contain
er.test.name}/myosdisk1.vhd"
    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 = {
```

```
  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/00000000-0000-0000-0000-0000000
00000/resourceGroups/mygroup1/providers/Microsoft.Compute/virtualMachines/myVM/extensions/hostname
```

azurestack_virtual_machine

Manages a virtual machine.

Example Usage with Managed Disks

```
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 the format of
 "West US" etc... those are not the same values
  location = "region1"
}

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_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"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
  }
  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 Managed 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 the format of
 "West US" etc... those are not the same values
  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_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"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
  }
  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

```
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 the format of
 "West US" etc... those are not the same values
  location = "region1"
}

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_storage_contain
er.test.name}/myosdisk1.vhd"
    caching       = "ReadWrite"
    create_option = "FromImage"
  }
  # Optional data disks
  storage_data_disk {
    name          = "datadisk0"
    vhd_uri       = "${azurestack_storage_account.test.primary_blob_endpoint}${azurestack_storage_contain
er.test.name}/datadisk0.vhd"
    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 the format of
 "West US" etc... those are not the same values
  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_storage_contain
er.test.name}/myosdisk1.vhd"
    caching       = "ReadWrite"
    create_option = "FromImage"
  }
  # Optional data disks
  storage_data_disk {
    name          = "datadisk0"
    vhd_uri       = "${azurestack_storage_account.test.primary_blob_endpoint}${azurestack_storage_contain
er.test.name}/datadisk0.vhd"
    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 (https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-size-specs/).
- `storage_image_reference` - (Optional) A Storage Image Reference block as documented below.
- `storage_os_disk` - (Required) A `storage_os_disk` block.
- `storage_data_disk` - (Optional) A list of Storage Data disk blocks as referenced below.
- `delete_os_disk_on_termination` - (Optional) Should the OS Disk be deleted when the Virtual Machine is destroyed? Defaults to `false`.
- `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 (https://msdn.microsoft.com/en-us/library/mt163591.aspx#Anchor_2)

`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` block supports the following:

- `name` - (Required) Specifies the disk name.
- `create_option` - (Required) Specifies how the OS Disk should be created. Possible values are `Attach` (managed disks only) and `FromImage` .
- `caching` - (Optional) Specifies the caching requirements for the OS Disk. Possible values include `None` , `ReadOnly` and `ReadWrite` .
- `image_uri` - (Optional) Specifies the image_uri in the form publisherName:offer:skus:version. `image_uri` can also specify the VHD uri (https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-cli-deploy-templates/#create-a-custom-vm-image) 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 on the OS Disk. Possible values are `Linux` and `Windows` .

- `disk_size_gb` - (Optional) Specifies the size of the os disk in gigabytes.

The following properties apply when using Managed Disks:

- `managed_disk_id` - (Optional) Specifies the ID of an existing Managed Disk which should be attached as the OS Disk of this Virtual Machine. If this is set then the `create_option` must be set to `Attach`.
- `managed_disk_type` - (Optional) Specifies the type of Managed Disk which should be created. Possible values are `Standard_LRS` or `Premium_LRS`.

The following properties apply when using Unmanaged Disks:

- `vhd_uri` - (Optional) Specifies the URI of the VHD file backing this Unmanaged OS Disk. Changing this forces a new resource to be created.

`storage_data_disk` supports the following:

- `name` - (Required) Specifies the name of the data disk.
- `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.

The following properties apply when using Managed Disks:

- `managed_disk_type` - (Optional) Specifies the type of managed disk to create. Possible values are either `Standard_LRS` or `Premium_LRS`.
- `managed_disk_id` - (Optional) Specifies the ID of an Existing Managed Disk which should be attached to this Virtual Machine. When this field is set `create_option` must be set to `Attach`.

The following properties apply when using Unmanaged Disks:

- `vhd_uri` - (Optional) Specifies the URI of the VHD file backing this Unmanaged Data Disk. Changing this forces a new resource to be created.

`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/00000000-0000-0000-0000-000000000000/reso
urceGroups/mygroup1/providers/microsoft.compute/virtualMachines/machine1
```

azurestack_virtual_machine_scale_set

Manages a virtual machine scale set.

Note: All arguments including the administrator login and password will be stored in the raw state as plain-text. Read more about sensitive data in state (/docs/state/sensitive-data.html).

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            = "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_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_scale_set" "test" {
  name                = "mytestscaleset-1"
  location            = "West US"
  resource_group_name = "${azurestack_resource_group.test.name}"
  upgrade_policy_mode = "Manual"
```

```
  sku {
    name     = "Standard_A0"
    tier     = "Standard"
    capacity = 2
  }

  os_profile {
    computer_name_prefix = "testvm"
    admin_username       = "myadmin"
    admin_password       = "Passwword1234"
  }

  os_profile_linux_config {
    disable_password_authentication = true

    ssh_keys {
      path     = "/home/myadmin/.ssh/authorized_keys"
      key_data = "${file("~/.ssh/demo_key.pub")}"
    }
  }

  network_profile {
    name    = "TestNetworkProfile"
    primary = true

    ip_configuration {
      name      = "TestIPConfiguration"
      subnet_id = "${azurestack_subnet.test.id}"
    }
  }

  storage_profile_os_disk {
    name           = "osDiskProfile"
    caching        = "ReadWrite"
    create_option  = "FromImage"
    vhd_containers = ["${azurestack_storage_account.test.primary_blob_endpoint}${azurestack_storage_conta
iner.test.name}"]
  }

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

Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of the virtual machine scale set 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 scale 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.
- `sku` - (Required) A sku block as documented below.
- `upgrade_policy_mode` - (Required) Specifies the mode of an upgrade to virtual machines in the scale set. Possible values, `Manual` or `Automatic`.
- `overprovision` - (Optional) Specifies whether the virtual machine scale set should be overprovisioned. Defaults to `true`.
- `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` - (Required) A Virtual Machine OS Profile block as documented below.
- `os_profile_secrets` - (Optional) A collection of Secret blocks as documented below.
- `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.
- `network_profile` - (Required) A collection of network profile block as documented below.
- `storage_profile_os_disk` - (Required) A storage profile os disk block as documented below
- `storage_profile_data_disk` - (Optional) A storage profile data disk block as documented below
- `storage_profile_image_reference` - (Optional) A storage profile image reference block as documented below.
- `extension` - (Optional) Can be specified multiple times to add extension profiles to the scale set. Each `extension` block supports the fields documented below.
- `plan` - (Optional) A plan block as documented below.
- `priority` - (Optional) Specifies the priority for the virtual machines in the scale set, defaults to `Regular`. Possible values are `Low` and `Regular`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Please Note: Availability Zones are in Preview and only supported in several regions at this time (https://docs.microsoft.com/en-us/azure/availability-zones/az-overview) - as such you must be opted into the Preview to use this functionality. You can opt into the Availability Zones Preview in the Azure Portal (http://aka.ms/azenroll).

`sku` supports the following:

- `name` - (Required) Specifies the size of virtual machines in a scale set.
- `tier` - (Optional) Specifies the tier of virtual machines in a scale set. Possible values, `standard` or `basic`.
- `capacity` - (Required) Specifies the number of virtual machines in the scale set.

`identity` supports the following:

- `type` - (Required) Specifies the identity type to be assigned to the scale set. The only allowable value is `SystemAssigned`. To enable Managed Service Identity (MSI) on all machines in the scale set, an extension with the type "ManagedIdentityExtensionForWindows" or "ManagedIdentityExtensionForLinux" must also be added. The scale

set's Service Principal ID (SPN) can be retrieved after the scale set has been created.

```
resource "azurestack_virtual_machine_scale_set" "test" {
  name                = "vm-scaleset"
  resource_group_name = "${azurestack_resource_group.test.name}"
  location            = "${azurestack_resource_group.test.location}"

  sku {
    name     = "${var.vm_sku}"
    tier     = "Standard"
    capacity = "${var.instance_count}"
  }

  identity {
    type = "systemAssigned"
  }

  extension {
    name                 = "MSILinuxExtension"
    publisher            = "Microsoft.ManagedIdentity"
    type                 = "ManagedIdentityExtensionForLinux"
    type_handler_version = "1.0"
    settings             = "{\"port\": 50342}"
  }

  output "principal_id" {
    value = "${lookup(azurestack_virtual_machine.test.identity[0], "principal_id")}"
  }
}
```

`os_profile` supports the following:

- `computer_name_prefix` - (Required) Specifies the computer name prefix for all of the virtual machines in the scale set. Computer name prefixes must be 1 to 9 characters long for windows images and 1 - 58 for linux. Changing this forces a new resource to be created.
- `admin_username` - (Required) Specifies the administrator account name to use for all the instances of virtual machines in the scale set.
- `admin_password` - (Required) Specifies the administrator password to use for all the instances of virtual machines in a scale set.
- `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.

`os_profile_secrets` supports the following:

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

`vault_certificates` support the following:

- `certificate_url` - (Required) It is the Base64 encoding of a JSON Object that which is encoded in UTF-8 of which the contents need to be `data`, `dataType` and `password`.

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

`os_profile_windows_config` supports the following:

- `provision_vm_agent` - (Optional) Indicates whether virtual machine agent should be provisioned on the virtual machines in the scale set.
- `enable_automatic_upgrades` - (Optional) Indicates whether virtual machines in the scale set are enabled for automatic updates.
- `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. Changing this forces a new resource to be created.
- `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*

`network_profile` supports the following:

- `name` - (Required) Specifies the name of the network interface configuration.
- `primary` - (Required) Indicates whether network interfaces created from the network interface configuration will be the primary NIC of the VM.
- `ip_configuration` - (Required) An ip_configuration block as documented below.

`public_ip_address_configuration` supports the following:

- `name` - (Required) The name of the public ip address configuration
- `idle_timeout` - (Required) The idle timeout in minutes. This value must be between 4 and 32.

- `domain_name_label` - (Required) The domain name label for the dns settings.

`storage_profile_os_disk` supports the following:

- `name` - (Optional) Specifies the disk name. Must be specified when using unmanaged disk ('managed_disk_type' property not set).
- `vhd_containers` - (Optional) Specifies the vhd uri. Cannot be used when `image` or `managed_disk_type` is specified.
- `managed_disk_type` - (Optional) Specifies the type of managed disk to create. Value you must be either `Standard_LRS` or `Premium_LRS`. Cannot be used when `vhd_containers` or `image` is specified.
- `create_option` - (Required) Specifies how the virtual machine should be created. The only possible option is `FromImage`.
- `caching` - (Optional) Specifies the caching requirements. Possible values include: `None` (default), `ReadOnly`, `ReadWrite`.
- `image` - (Optional) Specifies the blob uri for user image. A virtual machine scale set creates an os disk in the same container as the user image. Updating the osDisk image causes the existing disk to be deleted and a new one created with the new image. If the VM scale set is in Manual upgrade mode then the virtual machines are not updated until they have manualUpgrade applied to them. When setting this field `os_type` needs to be specified.
- `os_type` - (Optional) Specifies the operating system Type, valid values are windows, linux.

`storage_profile_data_disk` supports the following:

- `lun` - (Required) Specifies the Logical Unit Number of the disk in each virtual machine in the scale set.
- `create_option` - (Optional) Specifies how the data disk should be created. The only possible options are `FromImage` and `Empty`.
- `caching` - (Optional) Specifies the caching requirements. Possible values include: `None` (default), `ReadOnly`, `ReadWrite`.
- `disk_size_gb` - (Optional) Specifies the size of the disk in GB. This element is required when creating an empty disk.

`storage_profile_image_reference` supports the following:

- `id` - (Optional) Specifies the ID of the (custom) image to use to create the virtual machine scale set, as in the example below.
- `publisher` - (Optional) Specifies the publisher of the image used to create the virtual machines.
- `offer` - (Optional) Specifies the offer of the image used to create the virtual machines.
- `sku` - (Optional) Specifies the SKU of the image used to create the virtual machines.
- `version` - (Optional) Specifies the version of the image used to create the virtual machines.

`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.

`extension` supports the following:

- `name` - (Required) Specifies the name of the extension.
- `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.
- `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 whether or not to use the latest minor version available.
- `settings` - (Required) The settings passed to the extension, these are specified as a JSON object in a string.
- `protected_settings` - (Optional) The protected_settings passed to the extension, like settings, these are specified as a JSON object in a string.

`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.

Example of storage_profile_image_reference with id

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

  #...
}

resource "azurestack_virtual_machine_scale_set" "test" {
  name = "test"

  #...

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

  #...
}
```

Attributes Reference

The following attributes are exported:

- `id` - The virtual machine scale set ID.
- `boot_diagnostics` - A boot diagnostics profile block as referenced below.

Import

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

```
terraform import azurestack_virtual_machine_scale_set.scaleset1 /subscriptions/00000000-0000-0000-0000-00
0000000000/resourceGroups/mygroup1/providers/Microsoft.Compute/virtualMachineScaleSets/scaleset1
```

azurestack_virtual_network_gateway_connection

Manages a connection in an existing Virtual Network Gateway.

Example Usage

Site-to-Site connection

The following example shows a connection between an Azure virtual network and an on-premises VPN device and network.

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

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

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

resource "azurestack_local_network_gateway" "onpremise" {
  name                = "onpremise"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"
  gateway_address     = "168.62.225.23"
  address_space       = ["10.1.1.0/24"]
}

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

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

  type     = "Vpn"
  vpn_type = "RouteBased"

  active_active = false
```

```
  enable_bgp    = false
  sku           = "Basic"

  ip_configuration {
    public_ip_address_id          = "${azurestack_public_ip.test.id}"
    private_ip_address_allocation = "Dynamic"
    subnet_id                     = "${azurestack_subnet.test.id}"
  }
}

resource "azurestack_virtual_network_gateway_connection" "onpremise" {
  name                = "onpremise"
  location            = "${azurestack_resource_group.test.location}"
  resource_group_name = "${azurestack_resource_group.test.name}"

  type                       = "IPsec"
  virtual_network_gateway_id = "${azurestack_virtual_network_gateway.test.id}"
  local_network_gateway_id   = "${azurestack_local_network_gateway.onpremise.id}"

  shared_key = "4-v3ry-53cr37-1p53c-5h4r3d-k3y"
}
```

VNet-to-VNet connection

The following example shows a connection between two Azure virtual network in different locations/regions.

```
resource "azurestack_resource_group" "us" {
  name     = "us"
  location = "East US"
}

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

resource "azurestack_subnet" "us_gateway" {
  name                 = "GatewaySubnet"
  resource_group_name  = "${azurestack_resource_group.us.name}"
  virtual_network_name = "${azurestack_virtual_network.us.name}"
  address_prefix       = "10.0.1.0/24"
}

resource "azurestack_public_ip" "us" {
  name                         = "us"
  location                     = "${azurestack_resource_group.us.location}"
  resource_group_name          = "${azurestack_resource_group.us.name}"
  public_ip_address_allocation = "Dynamic"
}

resource "azurestack_virtual_network_gateway" "us" {
  name                = "us-gateway"
  location            = "${azurestack_resource_group.us.location}"
  resource_group_name = "${azurestack_resource_group.us.name}"
```

```
  type     = "Vpn"
  vpn_type = "RouteBased"
  sku      = "Basic"

  ip_configuration {
    public_ip_address_id          = "${azurestack_public_ip.us.id}"
    private_ip_address_allocation = "Dynamic"
    subnet_id                     = "${azurestack_subnet.us_gateway.id}"
  }
}

resource "azurestack_resource_group" "europe" {
  name     = "europe"
  location = "West Europe"
}

resource "azurestack_virtual_network" "europe" {
  name                = "europe"
  location            = "${azurestack_resource_group.europe.location}"
  resource_group_name = "${azurestack_resource_group.europe.name}"
  address_space       = ["10.1.0.0/16"]
}

resource "azurestack_subnet" "europe_gateway" {
  name                 = "GatewaySubnet"
  resource_group_name  = "${azurestack_resource_group.europe.name}"
  virtual_network_name = "${azurestack_virtual_network.europe.name}"
  address_prefix       = "10.1.1.0/24"
}

resource "azurestack_public_ip" "europe" {
  name                         = "europe"
  location                     = "${azurestack_resource_group.europe.location}"
  resource_group_name          = "${azurestack_resource_group.europe.name}"
  public_ip_address_allocation = "Dynamic"
}

resource "azurestack_virtual_network_gateway" "europe" {
  name                = "europe-gateway"
  location            = "${azurestack_resource_group.europe.location}"
  resource_group_name = "${azurestack_resource_group.europe.name}"

  type     = "Vpn"
  vpn_type = "RouteBased"
  sku      = "Basic"

  ip_configuration {
    public_ip_address_id          = "${azurestack_public_ip.europe.id}"
    private_ip_address_allocation = "Dynamic"
    subnet_id                     = "${azurestack_subnet.europe_gateway.id}"
  }
}

resource "azurestack_virtual_network_gateway_connection" "us_to_europe" {
  name                = "us-to-europe"
  location            = "${azurestack_resource_group.us.location}"
  resource_group_name = "${azurestack_resource_group.us.name}"

  type                            = "Vnet2Vnet"
```

```
  type                            = "Vnet2Vnet"
  virtual_network_gateway_id      = "${azurestack_virtual_network_gateway.us.id}"
  peer_virtual_network_gateway_id = "${azurestack_virtual_network_gateway.europe.id}"

  shared_key = "4-v3ry-53cr37-1p53c-5h4r3d-k3y"
}

resource "azurestack_virtual_network_gateway_connection" "europe_to_us" {
  name                = "europe-to-us"
  location            = "${azurestack_resource_group.europe.location}"
  resource_group_name = "${azurestack_resource_group.europe.name}"

  type                            = "Vnet2Vnet"
  virtual_network_gateway_id      = "${azurestack_virtual_network_gateway.europe.id}"
  peer_virtual_network_gateway_id = "${azurestack_virtual_network_gateway.us.id}"

  shared_key = "4-v3ry-53cr37-1p53c-5h4r3d-k3y"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the connection. Changing the name forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the connection Changing the name forces a new resource to be created.
- `location` - (Required) The location/region where the connection is located. Changing this forces a new resource to be created.
- `type` - (Required) The type of connection. Valid options are `IPsec` (Site-to-Site), `ExpressRoute` (ExpressRoute), and `Vnet2Vnet` (VNet-to-VNet). Each connection type requires different mandatory arguments (refer to the examples above). Changing the connection type will force a new connection to be created.
- `virtual_network_gateway_id` - (Required) The ID of the Virtual Network Gateway in which the connection will be created. Changing the gateway forces a new resource to be created.
- `authorization_key` - (Optional) The authorization key associated with the Express Route Circuit. This field is required only if the type is an ExpressRoute connection.
- `express_route_circuit_id` - (Optional) The ID of the Express Route Circuit when creating an ExpressRoute connection (i.e. when `type` is `ExpressRoute`). The Express Route Circuit can be in the same or in a different subscription.
- `peer_virtual_network_gateway_id` - (Optional) The ID of the peer virtual network gateway when creating a VNet-to-VNet connection (i.e. when `type` is `Vnet2Vnet`). The peer Virtual Network Gateway can be in the same or in a different subscription.
- `local_network_gateway_id` - (Optional) The ID of the local network gateway when creating Site-to-Site connection (i.e. when `type` is `IPsec`).
- `routing_weight` - (Optional) The routing weight. Defaults to `10`.

- `shared_key` - (Optional) The shared IPSec key. A key must be provided if a Site-to-Site or VNet-to-VNet connection is created whereas ExpressRoute connections do not need a shared key.
- `enable_bgp` - (Optional) If `true`, BGP (Border Gateway Protocol) is enabled for this connection. Defaults to `false`.
- `tags` - (Optional) A mapping of tags to assign to the resource.

Attributes Reference

The following attributes are exported:

- `id` - The connection ID.

Import

Virtual Network Gateway Connections can be imported using their `resource id`, e.g.

```
terraform import azurestack_virtual_network_gateway_connection.testConnection /subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myGroup1/providers/Microsoft.Network/connections/myConnection1
```

azurestack_virtual_network_gateway

Manages a Virtual Network Gateway to establish secure, cross-premises connectivity.

Example Usage

```
resource "azurestack_resource_group" "test" {
  name     = "test"
  location = "Azure-stack-region"
}

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

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

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

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

  type     = "Vpn"
  vpn_type = "RouteBased"
  sku      = "Basic"

  ip_configuration {
    public_ip_address_id          = "${azurestack_public_ip.test.id}"
    private_ip_address_allocation = "Dynamic"
    subnet_id                     = "${azurestack_subnet.test.id}"
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the connection. Changing the name forces a new resource to be created.
- `resource_group_name` - (Required) The name of the resource group in which to create the connection Changing the name forces a new resource to be created.
- `location` - (Required) The location/region where the connection is located. Changing this forces a new resource to be created.
- `type` - (Required) The type of the Virtual Network Gateway. Valid options is `Vpn`
- `vpn_type` - (Optional) The routing type of the Virtual Network Gateway. Only valid option is `RouteBased`.
- `enable_bgp` - (Optional) If `true`, BGP (Border Gateway Protocol) is enabled for this connection. Defaults to `false`.
- `sku` - (Required) Configuration of the size and capacity of the virtual network gateway. Valid options are `Basic`, `Standard` and `HighPerformance`.
- `ip_configuration` - (Required) One or two ip_configuration blocks documented below. An active-standby gateway requires exactly one ip_configuration block whereas an active-active gateway requires exactly two ip_configuration blocks.
- `tags` - (Optional) A mapping of tags to assign to the resource.

The `ip_configuration` block supports:

- `name` - (Optional) A user-defined name of the IP configuration. Defaults to vnetGatewayConfig.
- `private_ip_address_allocation` - (Optional) Defines how the private IP address of the gateways virtual interface is assigned. Valid options are Static or Dynamic. Defaults to Dynamic.
- `subnet_id` - (Required) The ID of the gateway subnet of a virtual network in which the virtual network gateway will be created. It is mandatory that the associated subnet is named `GatewaySubnet`. Therefore, each virtual network can contain at most a single Virtual Network Gateway.
- `public_ip_address_id` - (Optional) The ID of the public ip address to associate with the Virtual Network Gateway.

The `bgp_settings` block supports:

- `asn` - (Optional) The Autonomous System Number (ASN) to use as part of the BGP.
- `peering_address` - (Optional) The BGP peer IP address of the virtual network gateway. This address is needed to configure the created gateway as a BGP Peer on the on-premises VPN devices. The IP address must be part of the subnet of the Virtual Network Gateway. Changing this forces a new resource to be created

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Virtual Network Gateway.

Import

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

```
terraform import azurestack_virtual_network_gateway.testGateway /subscriptions/00000000-0000-0000-0000-00
0000000000/resourceGroups/myGroup1/providers/Microsoft.Network/virtualNetworkGateways/myGateway1
```

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 (/docs/providers/azurestack/r/subnet.html), and allows for Subnets to be defined in-line within the Virtual Network resource (/docs/providers/azurestack/r/virtual_network.html). 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 Subnets.

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/00000000-0000-0000-0000-000000000000/resourceGroups/mygroup1/providers/Microsoft.Network/virtualNetworks/myvnet1
```