VMware vRA7 Provider

The VMware vRA7 provider gives Terraform the ability to work with VMware vRealize Automation 7 (https://www.vmware.com/products/vrealize-automation.html). This provider can be used to deploy exisiting blueprints as deployments.

Use the navigation on the left to read about the various resources and data sources supported by the provider.

Example Usage

The following abridged example demonstrates a current basic usage of the provider to launch a virtual machine using the `vra7_deployment` resource (/docs/providers/vra7/r/deployment.html).

```
provider "vra7" {
  username = "${var.username}"
  password = "${var.password}"
  tenant   = "${var.tenant}"
  host     = "${var.host}"
}

resource "vra7_deployment" "machine" {
  count             = 1
  catalog_item_name = "CentOS 7.0 x64"

  resource_configuration = {
    "Linux.cpu" = "2"
  }
}
```

See the sidebar for usage information on all of the resources, which will have examples specific to their own use cases.

Argument Reference

The following arguments are used to configure the VMware vRA7 Provider:

- `username` - (Required) This is the username for vRA7 API operations. Can also be specified with the `VRA7_USERNAME` environment variable.
- `password` - (Required) This is the password for vRA7 API operations. Can also be specified with the `VRA7_PASSWORD` environment variable.
- `tenant` - (Required) This is the vRA tenant ID vRA API operations. Can also be specified with the `VRA7_SERVER` environment variable.
- `host` - (Required) This is the vRA server name for vRA API operations. Can also be specified with the `VRA7_HOST` environment variable.
- `insecure` - (Optional) Boolean that can be set to true to disable SSL certificate verification. This should be used with care as it could allow an attacker to intercept your auth token. If omitted, default value is `false`. Can also be specified with the `VRA7_INSECURE` environment variable.

Debugging options

NOTE: The following options can leak sensitive data and should only be enabled when instructed to do so by HashiCorp for the purposes of troubleshooting issues with the provider, or when attempting to perform your own troubleshooting. Use them at your own risk and do not leave them enabled!

Bug Reports and Contributing

For more information how how to submit bug reports, feature requests, or details on how to make your own contributions to the provider, see the vRA7 provider project page (https://github.com/terraform-providers/terraform-provider-vra7).

vra7_deployment

Provides a VMware vRA7 deployment resource. This can be used to deploy vRA7 catalog items.

Example Usages

Simple deployment of a vSphere machine with custom properties and a network profile:

You can refer to the sample blueprint (here (https://github.com/terraform-providers/terraform-provider-vra7/tree/master/website/docs/r)) to understand how it is translated to following the terraform config

```
resource "vra7_deployment" "my_vra7_deployment" {
  count = 1
  catalog_item_name = "Basic Single Machine"
  description = "Test deployment"
  reasons = "Testing the vRA 7 Terraform plugin"

  deployment_configuration = {
    _leaseDays = "15"
    _number_of_instances = 2
    deployment_property = "custom deployment property"
  }
  resource_configuration = {
    vSphereVM1.cpu = 1
    vSphereVM1.memory = 2048
    vSphereVM1.machine_property = "machine custom property"
  }
  wait_timeout = 20
  businessgroup_name = "Development"
}
```

Argument Reference

The following arguments are supported:

- `businessgroup_id` - (Optional) The id of the vRA business group to use for this deployment.
- `businessgroup_name` - (Optional) The name of the vRA business group to use for this deployment.
- `catalog_item_id` - (Optional) The id of the catalog item to deploy into vRA.
- `catalog_item_name` - (Optional) The name of the catalog item to deploy into vRA.
- `description` - (Optional) Description of the deployment
- `reasons` - (Optional) Reasons for requesting the deployment
- `deployment_configuration` - (Optional) The configuration of the deployment from the catalog item
- `resource_configuration` - (Optional) The configuration of the individual components from the catalog item

Nested Blocks

deployment_configuration

This block contains the deployment level properties including the custom properties. These are not a fixed set of properties but referred from the blueprint. There are generic properties like _leaseDays, _number_of_instances, etc but they are optional and from the example of the BasicSingleMachine blueprint, their is one custom property, called deployment_property which is required at request time. All the properties that are required during request, must be specified in the config file.

resource_configuration

This block contains the machine resource level properties including the custom properties. These are not a fixed set of properties but referred from the blueprint. The sample blueprint has one vSphere machine resource called vSphereVM1. Properties of this machine can be specified in the config in the format "vSphereVM1.property_name". The properties like cpu, memory, storage, etc are generic machine properties and their is a custom property as well, called machine_property in the sample blueprint which is required at request time. There can be any number of machines and same format has to be followed to specify properties of other machines as well. All the properties that are required during request, must be specified in the config file.

More examples

Simple deployment of a CentOS Linux host with 2 CPU's:

```
resource "vra7_deployment" "machine" {
  count            = 1
  catalog_item_name = "CentOS 7.0 x64"
  resource_configuration = {
    Linux.cpu = "2"
  }
}
```

Catalog "multi_machine_catalog" contains Linux, Windows and http (apache) designs:

```
resource "vra7_deployment" "resource_1" {
  count            = 1
  catalog_item_name = "multi_machine_catalog"
  resource_configuration = {
    Windows.cpu = "2"                //Windows Machine CPU
    Linux.cpu = "2"                  //Linux Machine CPU
    http.hostname = "xyz.com"        //HTTP (apache) hostname
    http.network_mode = "bridge"     //HTTP (apache) network mode
  }
}
```

Showing the use of `depends_on` between two deployments:

Here the second resource, machine2 is dependent on the resource, machine1. So, machine2 will be provisioned after

machine1.

```
resource "vra7_deployment" "machine1" {
  count            = 1
  catalog_item_name = "CentOS 7.0 x64"
}

resource "vra7_deployment" "machine2" {
  count            = 1
  catalog_item_name = "CentOS 7.0 x64"
  depends_on = ["vra7_deployment.machine1"]
}
```