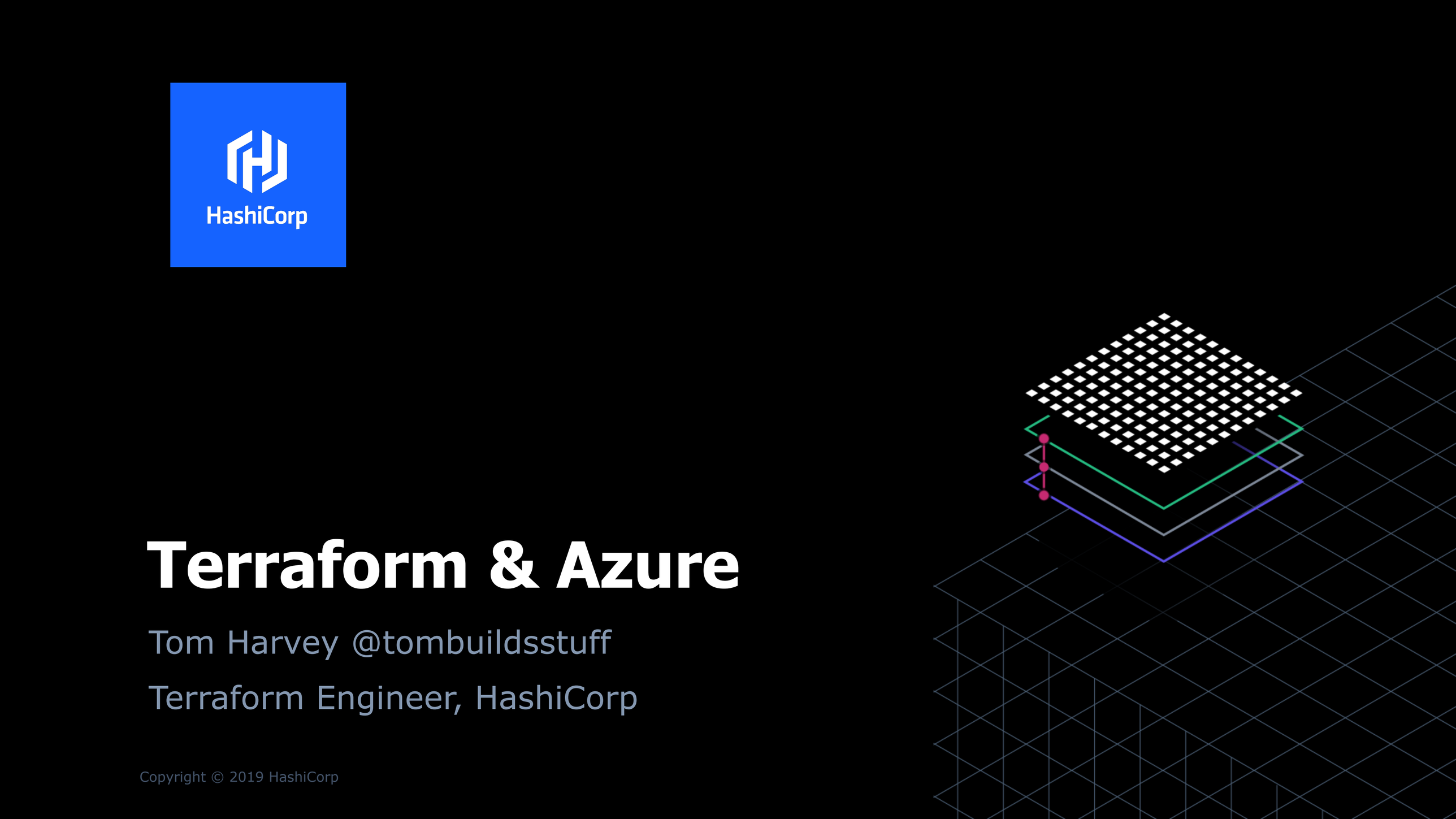
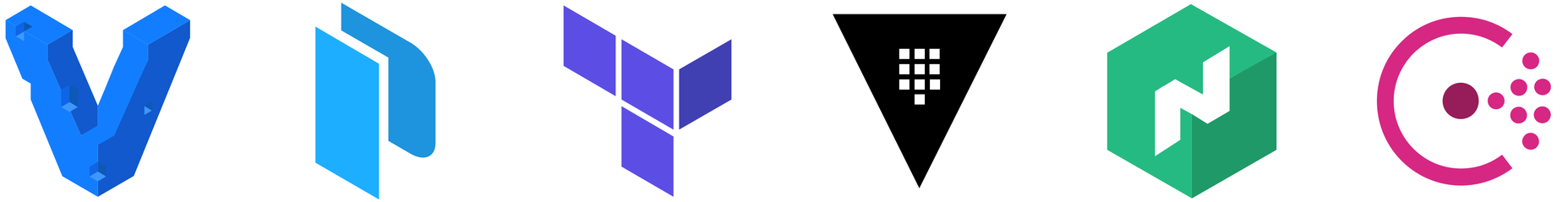


Terraform & Azure

Tom Harvey @tombuildsstuff

Terraform Engineer, HashiCorp

Copyright © 2019 HashiCorp



Azure @ HashiCorp

What’s the state of play?

Copyright © 2019 HashiCorp

Investing deeply in Terraform on Azure

Posted on August 17, 2017

Corey Sanders, Corporate Vice President, Azure

As customers increase their deployed applications in Azure, we are seeing a growing interest in DevOps tooling on Azure. We also see customers looking to deploy applications across multiple environments, including hybrid and multi-cloud deployments while using the same tooling and enabling the same DevOps experiences. In order to meet these growing needs, I am excited to announce that we are greatly increasing our investment in Terraform, partnering closely with HashiCorp, a well-known voice in the DevOps and cloud infrastructure management space.

Our partnership with HashiCorp goes back to early 2016, where we jointly announced plans to bring full support for Azure Resource Manager across many tools in HashiCorp's portfolio including Packer and Terraform. Since then, our customers have found significant value in the HashiCorp support on Azure.

Today, we're extending our partnership and will offer an increasing number of services directly supported by Terraform, including Azure Container Instances, Azure Container Service, Managed Disks, Virtual Machine Scale Sets and others. We want to give additional flexibility to express infrastructure-as-code and to enable many more native Microsoft Azure services to be easily deployed directly through Terraform. Learn more about the Azure provider for Terraform.

I am really excited about our partnership with HashiCorp. They are well-positioned to support the complexity and diversity of this space. They also have a rich portfolio of products that can help our customers adopt DevOps principles

Subscribe

Explore

See where we're heading. Check out upcoming changes to Azure products

Azure updates

Let us know what you think of Azure and what you would like to see in the future

Provide feedback

Topics

Announcements (1930)

Artificial Intelligence (124)

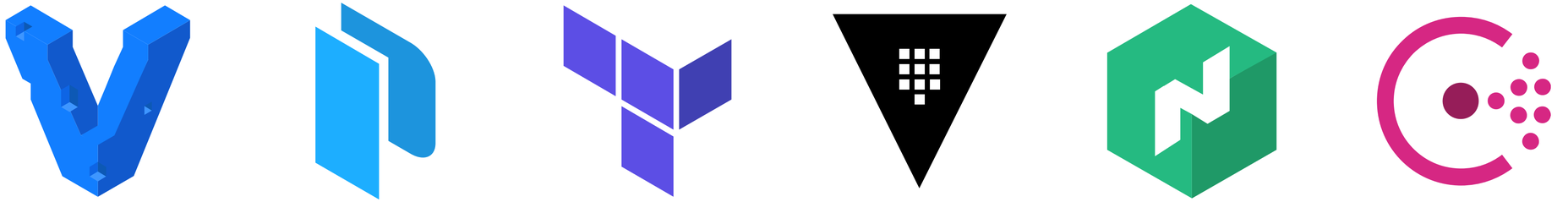
Azure Maps (11)

Azure Marketplace (100)

Big Data (582)

Blockchain (72)

Business Intelligence (99)



Vagrant & Azure

Provisioning a virtual machine in Azure

- Vagrant has an Azure plugin that allows provisioning Vagrant machines in Azure

- https://github.com/Azure/vagrant-azure

Copyright © 2019 HashiCorp

Vagrant & Azure

Provisioning a virtual machine in Azure

```
Vagrant.configure('2') do |config|
  config.vm.box = 'azure'

  # use local ssh key to connect to remote vagrant box
  config.ssh.private_key_path = '~/.ssh/id_rsa'
  config.vm.provider :azure do |azure, override|

    # each of the below values will default to use the env vars
named as below if not specified explicitly
    azure.tenant_id = ENV['AZURE_TENANT_ID']
    azure.client_id = ENV['AZURE_CLIENT_ID']
    azure.client_secret = ENV['AZURE_CLIENT_SECRET']
    azure.subscription_id = ENV['AZURE_SUBSCRIPTION_ID']
  end

end
```

Copyright © 2019 HashiCorp

Packer & Azure

Building Images with Packer on Azure

- Packer Builder for Azure: `azure-arm`
- Can produce either a VHD / Managed Disk
- Authenticating via a Service Principal / MSI

Copyright © 2019 HashiCorp

Packer & Azure

Building Images with Packer on Azure

```
{
  "variables": {},
  "builders": [{
    "type": "azure-arm",
    "resource_group_name": "packer-images",
    "storage_account": "myexamplestoraccount",
    "subscription_id": "00000000-0000-0000-0000-000000000000",
    "os_type": "Linux",
    "image_publisher": "Canonical",
    "image_offer": "UbuntuServer",
    "image_sku": "16.04-LTS",
    "location": "West US",
 }],
  "provisioners": [{
    # ...
  }]
}
```

Copyright © 2019 HashiCorp

Terraform & Azure

Provisioning Resources on Azure using Terraform

- Terraform has multiple Providers supporting Azure:
  - Azure Active Directory
  - Azure Resource Manager
  - Azure Stack
- Now supports 190 Resources & 59 Data Sources

Copyright © 2019 HashiCorp

Terraform & Azure

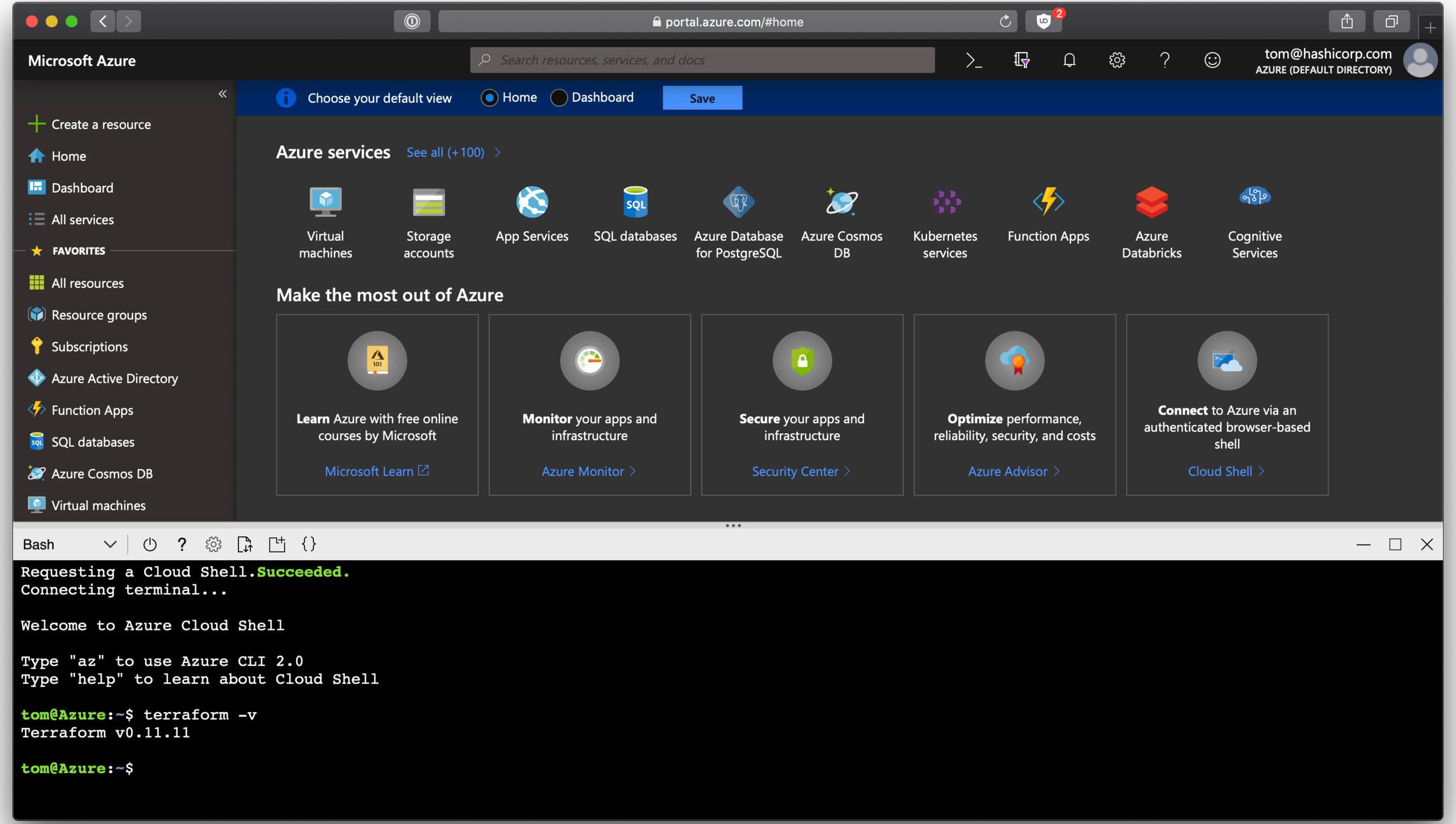
Provisioning Resources on Azure using Terraform

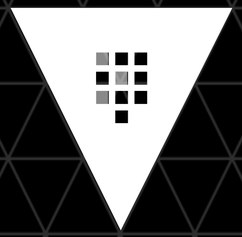
```
provider "azurerm" {
  version = "=1.22.0"
}

resource "azurerm_resource_group" "test" {
  name     = "oslo-hug-resources"
  location = "West Europe"
}

resource "azurerm_virtual_network" "test" {
  name                = "oslo-hug-network"
  resource_group_name = "${azurerm_resource_group.test.name}"
  location            = "${azurerm_resource_group.test.location}"
  address_space       = ["10.0.0.0/16"]
}
```

Copyright © 2019 HashiCorp


portal.azure.com/#home
Microsoft Azure
Search resources, services, and docs
tom@hashicorp.com
AZURE (DEFAULT DIRECTORY)
Create a resource
Home
Dashboard
All services
FAVORITES
All resources
Resource groups
Subscriptions
Azure Active Directory
Function Apps
SQL databases
Azure Cosmos DB
Virtual machines
Choose your default view
Home
Dashboard
Save
Azure services
See all (+100)
Virtual machines
Storage accounts
App Services
SQL databases
Azure Database for PostgreSQL
Azure Cosmos DB
Kubernetes services
Function Apps
Azure Databricks
Cognitive Services
Make the most out of Azure
Learn Azure with free online courses by Microsoft
Microsoft Learn
Monitor your apps and infrastructure
Azure Monitor
Secure your apps and infrastructure
Security Center
Optimize performance, reliability, security, and costs
Azure Advisor
Connect to Azure via an authenticated browser-based shell
Cloud Shell
Bash
Requesting a Cloud Shell.Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell
Type "az" to use Azure CLI 2.0
Type "help" to learn about Cloud Shell
tom@Azure:~$ terraform -v
Terraform v0.11.11
tom@Azure:~$

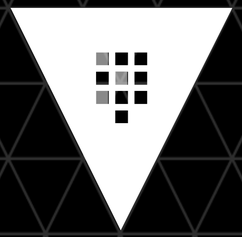


Vault & Azure

Supported Integrations

- Vault supports both an Auth Method and a Secrets Backend
- Allows you to verify that a VM/VM Scale Set exists

Copyright © 2019 HashiCorp

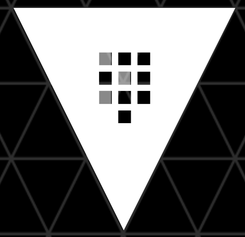


Vault & Azure

Auth Method

```
$ vault write auth/azure/login \
    role="dev-role" \
    jwt="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
    subscription_id="12345-..." \
    resource_group_name="test-group" \
    vm_name="test-vm"
```

Copyright © 2019 HashiCorp



Vault & Azure

Secrets Backend

```
$ vault read azure/creds/my-role

Key                 Value
---                 -----
lease_id            azure/creds/sp_role/1afd0969-ad23-73e2-f974-962f7ac1c2b4
lease_duration      60m
lease_renewable     true
client_id           408bf248-dd4e-4be5-919a-7f6207a307ab
client_secret       ad06228a-2db9-4e0a-8a5d-e047c7f32594
```

Copyright © 2019 HashiCorp

Nomad & Azure

Automatic discovery of cluster members in Azure

- Nomad (and Consul) support automatic discovery of other nodes
- Documentation: https://www.nomadproject.io/docs/configuration/server_join.html

Copyright © 2019 HashiCorp

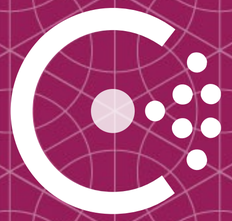
Nomad & Azure

Automatic discovery of cluster members in Azure

```
log_level = "DEBUG"
data_dir = “/var/lib/nomad/server"
datacenter = “westeurope”

server {
  enabled = true
  bootstrap_expect = 3
  retry_join = ["provider=azure tag_name=\"HashiStack\"
tag_value=\"OsloDev\"
client_id=\"00000000-0000-0000-0000-000000000000\"
subscription_id=\"00000000-0000-0000-0000-000000000000\"
secret_access_key=\"00000000-0000-0000-0000-000000000000\"
tenant_id=\"00000000-0000-0000-0000-000000000000\""]
}
```

Copyright © 2019 HashiCorp

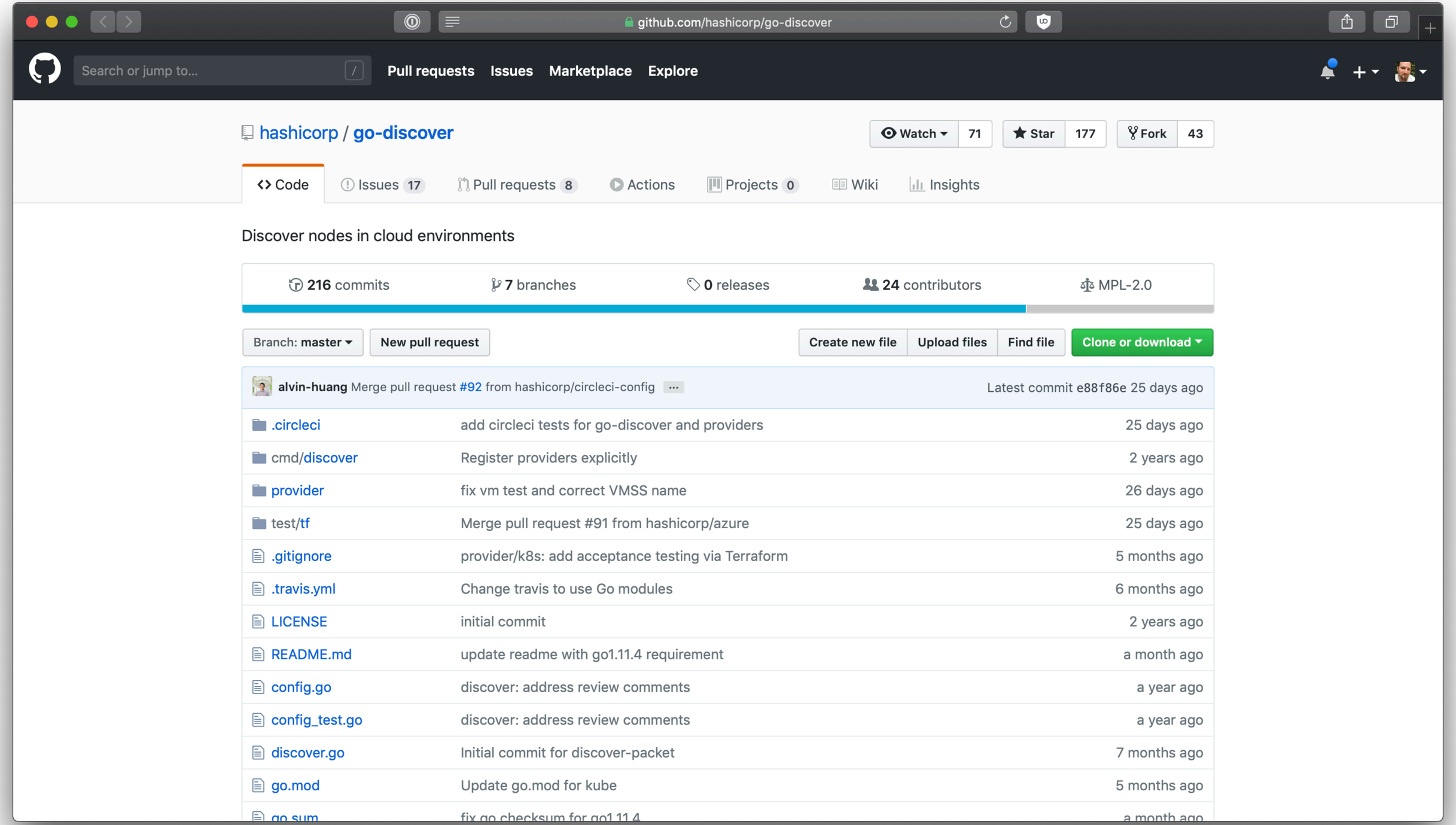


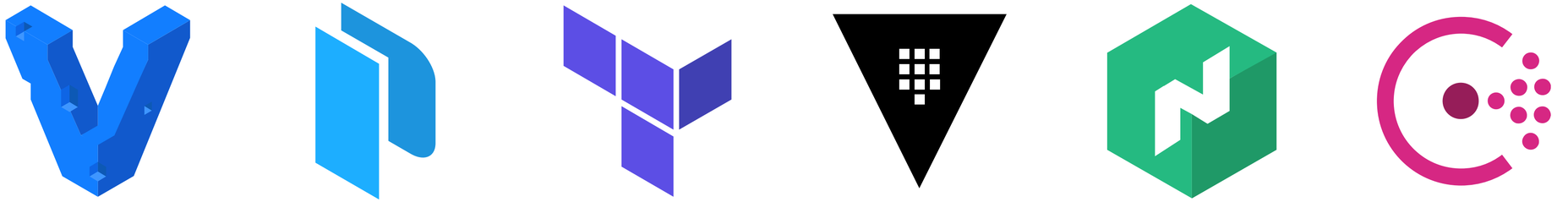
Consul & Azure

Automatic discovery of cluster members in Azure

```
{
    "bootstrap_expect": 3,
    "server": true,
    "encrypt": "zYB6e/vH/P38J8GIgklSlA==",
    "leave_on_terminate": true,
    "log_level": "INFO",
    "rejoin_after_leave": true,
    "datacenter": "westeurope",
    "data_dir": "/var/consul",
    "retry_join": ["provider=azure tag_name=\"HashiStack\"
tag_value=\"OsloDev\"
client_id=\"00000000-0000-0000-0000-000000000000\"
subscription_id=\"00000000-0000-0000-0000-000000000000\"
secret_access_key=\"00000000-0000-0000-0000-000000000000\"
tenant_id=\"00000000-0000-0000-0000-000000000000\""]
}
```

Copyright © 2019 HashiCorp


github.com/hashicorp/go-discover
Search or jump to...
Pull requests
Issues
Marketplace
Explore
hashicorp / go-discover
Watch 71
Star 177
Fork 43
Code
Issues 17
Pull requests 8
Actions
Projects 0
Wiki
Insights
Discover nodes in cloud environments
216 commits
7 branches
0 releases
24 contributors
MPL-2.0
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
alvin-huang Merge pull request #92 from hashicorp/circleci-config
Latest commit e88f86e 25 days ago
.circleci add circleci tests for go-discover and providers 25 days ago
cmd/discover Register providers explicitly 2 years ago
provider fix vm test and correct VMSS name 26 days ago
test/tf Merge pull request #91 from hashicorp/azure 25 days ago
.gitignore provider/k8s: add acceptance testing via Terraform 5 months ago
.travis.yml Change travis to use Go modules 6 months ago
LICENSE initial commit 2 years ago
README.md update readme with go1.11.4 requirement a month ago
config.go discover: address review comments a year ago
config_test.go discover: address review comments a year ago
discover.go Initial commit for discover-packet 7 months ago
go.mod Update go.mod for kube 5 months ago
go.sum fix go checksum for go1.11.4 a month ago



Demo

Consul, Nomad and Terraform on Azure

Copyright © 2019 HashiCorp

Common Pitfalls

Common stumbling blocks in Azure

Copyright © 2019 HashiCorp

Common Pitfalls

Dynamic IP Addresses

- Dynamic IP Addresses in Azure aren’t assigned until the VM/LB etc is Running

Copyright © 2019 HashiCorp

Common Pitfalls

Dynamic IP Addresses

```
resource "azurerm_public_ip" "main" {
  name                = "example-pip"
  resource_group_name = “example-resources"
  location            = “West Europe”
  allocation_method   = “Dynamic"
  tags                = "${var.tags}"
}

output "public_ip_address" {
  value = "${azurerm_public_ip.main.ip_address}"
}
```

Copyright © 2019 HashiCorp

Common Pitfalls

Dynamic IP Addresses

- Dynamic IP Addresses in Azure aren’t assigned until the VM/LB etc is Running
- You can use the Data Source to obtain the IP Address once it’s got an IP Address

Copyright © 2019 HashiCorp

Common Pitfalls

Dynamic IP Addresses

```
resource "azurerm_public_ip" "main" { .. }
resource “azurerm_network_interface" "main" { .. }
resource “azurerm_virtual_machine” "main" { .. }

data “azurerm_public_ip” “main” {
  name                = “${azurerm_public_ip.main.name}”
  resource_group_name = “${azurerm_public_ip.main.resource_group_name}”
  depends_on          = [“azurerm_virtual_machine.main”]
}

output "public_ip_address" {
  value = “${data.azurerm_public_ip.main.ip_address}"
}
```

Copyright © 2019 HashiCorp

Common Pitfalls

Dynamic IP Addresses

- Dynamic IP Addresses in Azure aren’t assigned until the VM/LB etc is Running
- You can use the Data Source to obtain the IP Address once it’s got an IP Address
- Alternatively you can use a Static IP Address

Copyright © 2019 HashiCorp

Common Pitfalls

Resource ID’s

- Azure uses the `name` as the unique identifier

Copyright © 2019 HashiCorp

Common Pitfalls

Resource ID’s

- Azure uses the `name` as the unique identifier
- However Azure’s API’s are Upserts; meaning if a resource already exists it’ll Update it, if not it’ll Create it.

Copyright © 2019 HashiCorp

Common Pitfalls

Resource ID’s

- Azure uses the `name` as the unique identifier
- However Azure’s API’s are Upserts; meaning if a resource already exists it’ll Update it, if not it’ll Create it.
- This means provisioning multiple resources with the same `name` can conflict and end up provisioning the same resource

Copyright © 2019 HashiCorp

Common Pitfalls

Resource ID’s

```
resource “azurerm_resource_group” “test” {
  name     = “example-resources”
  location = “West Europe”
}

resource “azurerm_public_ip” “test” {
  name                = “example-pip”
  location            = “${azurerm_resource_group.test.location}”
  resource_group_name = “${azurerm_resource_group.test.name}”
}
```

Copyright © 2019 HashiCorp

Common Pitfalls

Resource ID’s

```
resource “azurerm_resource_group” “test” {
  name     = “example-resources”
  location = “West Europe”
}

resource “azurerm_public_ip” “test” {
  count               = 3
  name                = “example-pip”
  location            = “${azurerm_resource_group.test.location}”
  resource_group_name = “${azurerm_resource_group.test.name}”
}
```

Copyright © 2019 HashiCorp

Common Pitfalls

Resource ID’s

- Solutions:
  - You can append a unique element (e.g. count) to the `name` to ensure these are unique

Copyright © 2019 HashiCorp

Common Pitfalls

Resource ID’s

```
resource “azurerm_resource_group” “test” {
  name     = “example-resources”
  location = “West Europe”
}

resource “azurerm_public_ip” “test” {
  count               = 3
  name                = “example-pip-${count.index}”
  location            = “${azurerm_resource_group.test.location}”
  resource_group_name = “${azurerm_resource_group.test.name}”
}
```

Copyright © 2019 HashiCorp

Common Pitfalls

Resource ID’s

- Solutions:
  - You can append a unique element (e.g. count) to the `name` to ensure these are unique
  - v2 of Azure Provider solves this via Requires Imports - more details later.

Copyright © 2019 HashiCorp

Common Pitfalls

Monolithic Resources

- Many of Azure’s resources are Monolithic, and only allow one thing to change at once

Copyright © 2019 HashiCorp

Common Pitfalls

Monolithic Resources

- Many of Azure’s resources are Monolithic, and only allow one thing to change at once
  - e.g. Load Balancers, Networks & Virtual Machines

Copyright © 2019 HashiCorp

Common Pitfalls

Monolithic Resources

- Many of Azure’s resources are Monolithic, and only allow one thing to change at once
  - e.g. Load Balancers, Networks & Virtual Machines
- This means that it can be hard to represent these resources in Terraform, since there’s a circular reference

Copyright © 2019 HashiCorp

Common Pitfalls

Monolithic Resources

- Many of Azure’s resources are Monolithic, and only allow one thing to change at once
  - e.g. Load Balancers, Networks & Virtual Machines
- This means that it can be hard to represent these resources in Terraform, since there’s a circular reference
- We’re creating Virtual Resources where possible, which help resolve the circular reference

Copyright © 2019 HashiCorp

Common Pitfalls

Monolithic Resources

```
resource "azurerm_resource_group" "test" { ... }
resource "azurerm_virtual_network" "test" { ... }
resource "azurerm_network_security_group" "test" { ... }

resource "azurerm_subnet" "test" {
  # ...
  network_security_group_id = "${azurerm_network_security_group.test.id}"
}
```

Copyright © 2019 HashiCorp

Common Pitfalls

Monolithic Resources

```
resource "azurerm_resource_group" "test" { ... }
resource "azurerm_virtual_network" "test" { ... }
resource "azurerm_network_security_group" "test" { ... }

resource "azurerm_subnet" "test" {
  # ...
  network_security_group_id = "${azurerm_network_security_group.test.id}"
}

resource "azurerm_subnet_network_security_group_association" "test" {
  subnet_id                 = "${azurerm_subnet.test.id}"
  network_security_group_id = "${azurerm_network_security_group.test.id}"
}
```

Copyright © 2019 HashiCorp

Common Pitfalls

Monolithic Resources

Coming in 2.0

```
resource "azurerm_resource_group" "test" { ... }
resource "azurerm_virtual_network" "test" { ... }
resource "azurerm_network_security_group" "test" { ... }
resource "azurerm_subnet" "test" { ... }

resource "azurerm_subnet_network_security_group_association" "test" {
  subnet_id                 = "${azurerm_subnet.test.id}"
  network_security_group_id = "${azurerm_network_security_group.test.id}"
}
```

Copyright © 2019 HashiCorp

What's supported?

Which resources does Terraform support?

Copyright © 2019 HashiCorp

What's supported in Terraform?

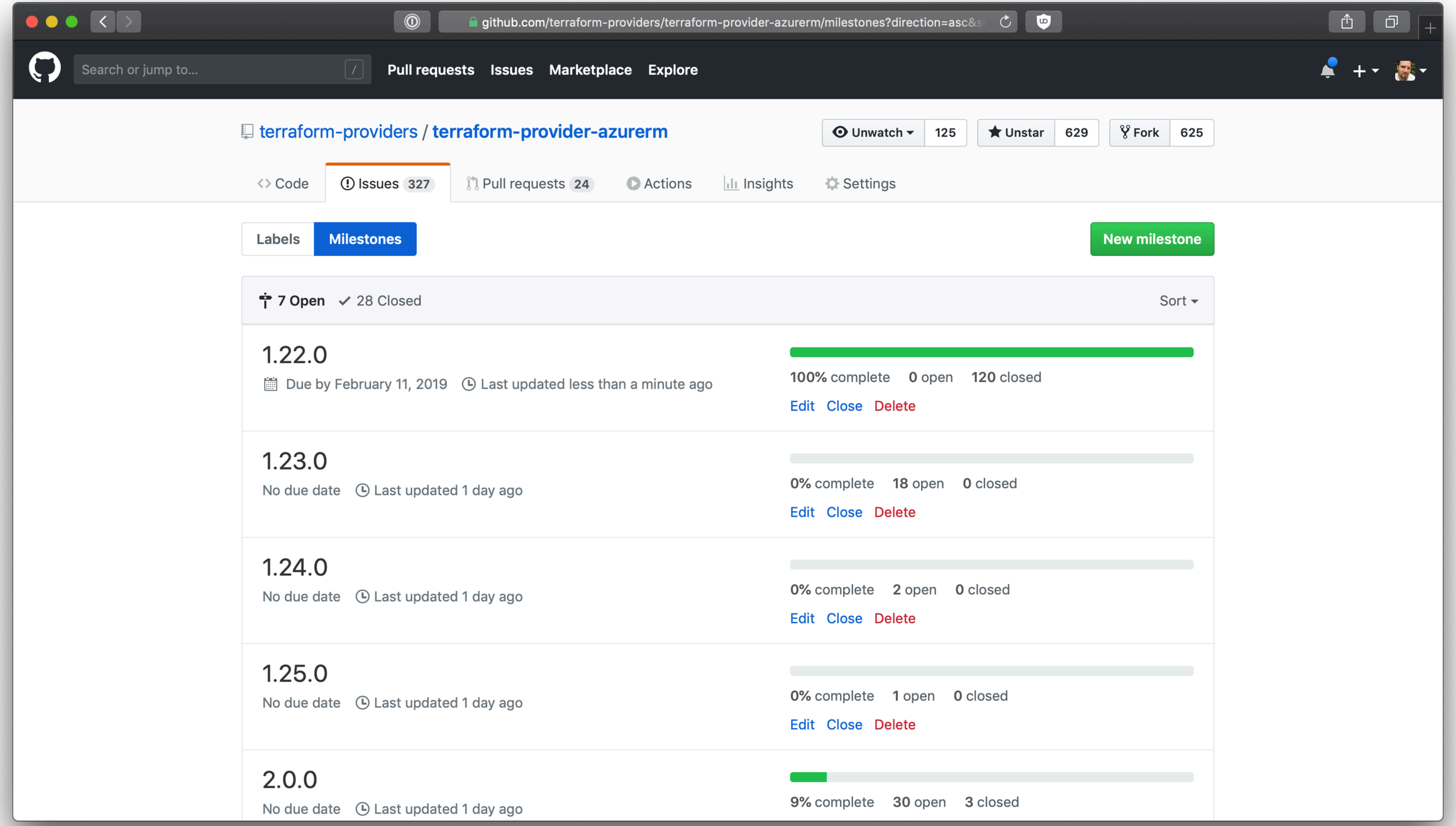
- AKS / Kubernetes
- App Service
- Application Gateway
- Application Insights
- AutoScale Setting
- Automation
- AzureAD
- Azure Monitor
- Batch
- CDN
- Cognitive Services
- Container Instance
- Container Registry
- CosmosDB
- DNS
- Data Lake Analytics
- Data Lake Store
- DataBricks
- Dev Test Labs
- DevSpace
- EventGrid
- EventHub
- Express Route
- Firewall
- Function Apps
- IoTHub
- Key Vault
- Load Balancers
- Local Network Gateway
- Log Analytics / OMS
- Logic Apps
- Management Group
- Management Locks
- MariaDB
- MySQL
- Networks
- Notification Hubs
- Policy
- PostgreSQL
- Recovery Services
- Redis
- Relay
- SQL Azure
- Scheduler Job
- Search
- Security Center
- Service Fabric
- ServiceBus
- SignalR
- Storage
- VM Scale Sets
- Virtual Machines

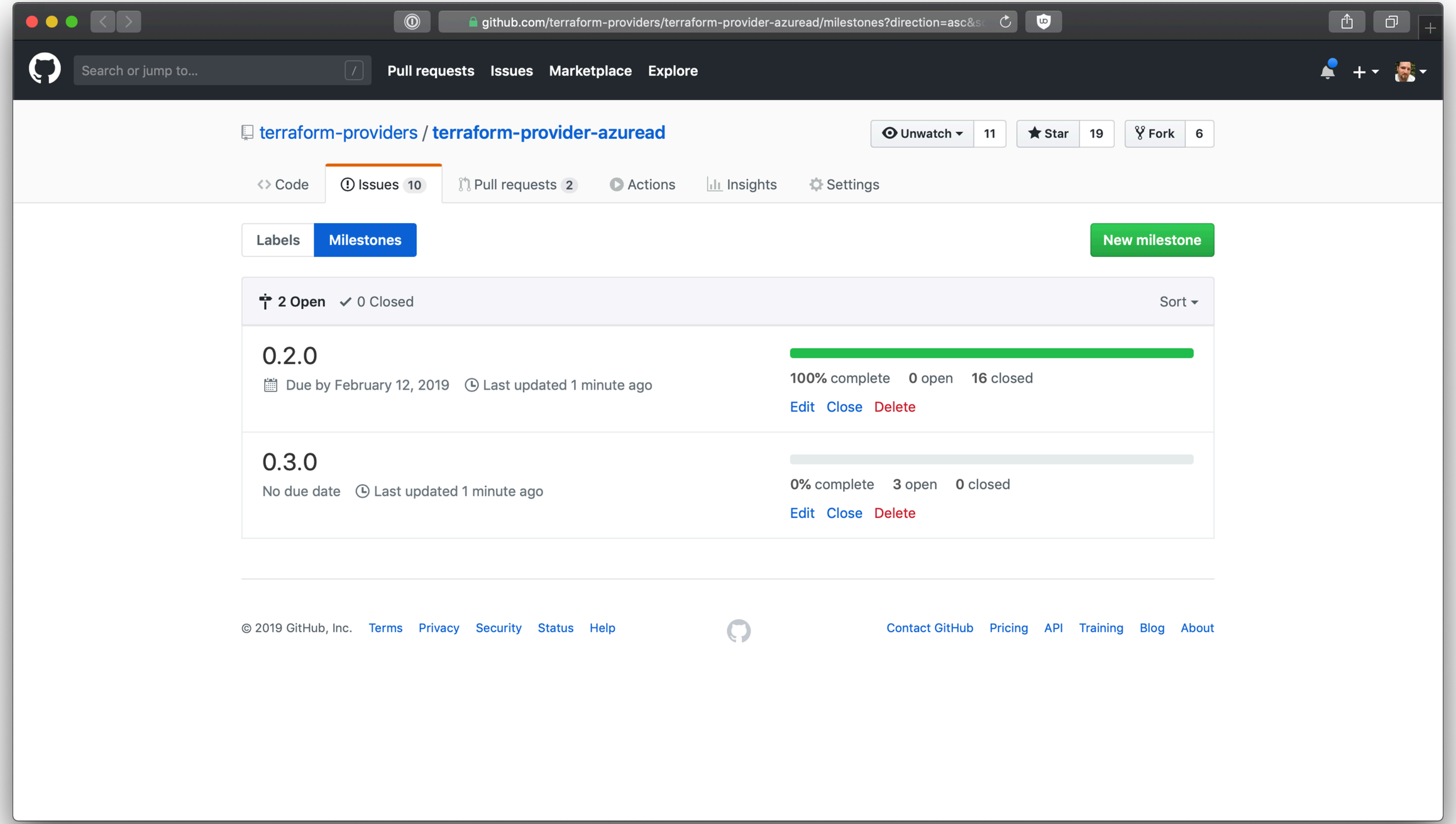
Copyright © 2019 HashiCorp

What’s next?

What’s coming over the next few months?

Copyright © 2019 HashiCorp


github.com/terraform-providers/terraform-provider-azurerm/milestones?direction=asc&s
Search or jump to...
Pull requests
Issues
Marketplace
Explore
terraform-providers / terraform-provider-azurerm
Unwatch 125
Unstar 629
Fork 625
Code
Issues 327
Pull requests 24
Actions
Insights
Settings
Labels
Milestones
New milestone
7 Open
28 Closed
Sort
1.22.0
Due by February 11, 2019
Last updated less than a minute ago
100% complete 0 open 120 closed
Edit Close Delete
1.23.0
No due date
Last updated 1 day ago
0% complete 18 open 0 closed
Edit Close Delete
1.24.0
No due date
Last updated 1 day ago
0% complete 2 open 0 closed
Edit Close Delete
1.25.0
No due date
Last updated 1 day ago
0% complete 1 open 0 closed
Edit Close Delete
2.0.0
No due date
Last updated 1 day ago
9% complete 30 open 3 closed


github.com/terraform-providers/terraform-provider-azuread/milestones?direction=asc&s
Search or jump to...
Pull requests
Issues
Marketplace
Explore
terraform-providers / terraform-provider-azuread
Unwatch 11
Star 19
Fork 6
Code
Issues 10
Pull requests 2
Actions
Insights
Settings
Labels
Milestones
New milestone
2 Open
0 Closed
Sort
0.2.0
Due by February 12, 2019
Last updated 1 minute ago
100% complete
0 open
16 closed
Edit
Close
Delete
0.3.0
No due date
Last updated 1 minute ago
0% complete
3 open
0 closed
Edit
Close
Delete
© 2019 GitHub, Inc.
Terms
Privacy
Security
Status
Help
Contact GitHub
Pricing
API
Training
Blog
About

What’s coming over the next few months?

Provider: Azure Active Directory (0.x)

- Groups (0.2)
- Users (0.3)

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Provider: AzureRM (1.x)

- API Management
- CosmosDB Collections
- CosmosDB Databases
- EventGrid
- HDInsights
- Subscriptions
- Terraform 0.12 support

Copyright © 2019 HashiCorp

Announcement: Upcoming Changes in Version 2.0 of the Azure Provider #2807

Edit New issue

Open **tombuildsstuff** opened this issue 10 days ago · 2 comments

tombuildsstuff commented 10 days ago • edited Member

Terraform initially shipped support for the AzureRM Provider back in December 2015. Since then we've added support for 191 Resources, 58 Data Sources and have launched a couple of related Providers in the form of the Azure Active Directory Provider and the Azure Stack Provider.

Version 2.0 of the AzureRM Provider will be a **Major Release** - in that it will include some larger-scale changes not seen in a regular release. A summary of these changes is outlined below - however a full breakdown will be available on the Terraform Website after the release of v1.22.

Summary

- Existing Resources will be required to be imported
- Custom Timeouts will be available on Resources - this will allow you to specify a custom timeout for provisioning the resource in your Terraform Configuration using the `timeouts` block.
- New resources for Virtual Machines and Virtual Machine Scale Sets
- Removing Fields, Data Sources and Resources which have been deprecated

A brief summary of each item can be found below - more details will be available in the Azure Provider 2.0 upgrade guide on the Terraform Website once v1.22 has been released.

Assignees

No one—assign yourself

Labels

breaking-change

Projects

None yet

Milestone

2.0.0

Notifications

Unsubscribe

You're receiving notifications because you authored the thread.

v2.0 of the AzureRM Provider

Terraform initially shipped support for the AzureRM Provider back in December 2015.

Since then we've added support for 191 Resources, 58 Data Sources and have launched a couple of related Providers in the form of the Azure Active Directory Provider and the Azure Stack Provider.

Version 2.0 of the AzureRM Provider is a major release and as such includes some larger-scale changes which are outlined in this document.

NOTE: This guide is a Work In Progress and additional information may be added to this guide until version 2.0 of the AzureRM Provider is released.

Pinning your Provider Version

We recommend pinning the version of each Provider you use in Terraform - you can do this using the `version` attribute in the `provider` block, either to a specific version of the AzureRM Provider, like so:

```
provider "azurerm" {
  version = "=1.22.0"
}
```

What’s coming over the next few months?

Provider: AzureRM (2.0)

- Custom Timeouts

Copyright © 2019 HashiCorp

Azure Provider 2.0

Custom Timeouts

- At the moment all API calls are hard-limited to an hour
- v2.0 of the Azure Provider will allow you to specify the timeout on resources

Copyright © 2019 HashiCorp

Azure Provider 2.0

Custom Timeouts

```
resource "azurerm_resource_group" "test" {
  name     = "example-resource-group"
  location = "West Europe"
}
```

Copyright © 2019 HashiCorp

Azure Provider 2.0

Custom Timeouts

```
resource "azurerm_resource_group" "test" {
  name     = "example-resource-group"
  location = "West Europe"

  timeouts {
    create = "10m"
    delete = "30m"
  }
}
```

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Provider: AzureRM (2.0)

- Custom Timeouts
- Requiring Imports

Copyright © 2019 HashiCorp

Azure Provider 2.0

Requiring Imports

- As seen earlier - Azure using the `name` as the Resource ID and having API’s which are Upserts means it’s possible to intentionally import an existing resource into Terraform

Copyright © 2019 HashiCorp

Azure Provider 2.0

Requiring Imports

- As seen earlier - Azure using the `name` as the Resource ID and having API's which are Upserts means it's possible to intentionally import an existing resource into Terraform

- However some API's require that a separate Update API is called when the resource is being changed, as such users can see unhelpful error messages

Copyright © 2019 HashiCorp

Azure Provider 2.0

Requiring Imports

- As seen earlier - Azure using the `name` as the Resource ID and having API's which are Upserts means it's possible to intentionally import an existing resource into Terraform
- However some API's require that a separate Update API is called when the resource is being changed, as such users can see unhelpful error messages
- To work around this we're going to check for an existing resource with the same name as each resource is created, and then require that these resources are Imported

Copyright © 2019 HashiCorp

Azure Provider 2.0

Requiring Imports

- As seen earlier - Azure using the `name` as the Resource ID and having API's which are Upserts means it's possible to intentionally import an existing resource into Terraform
- However some API's require that a separate Update API is called when the resource is being changed, as such users can see unhelpful error messages
- To work around this we're going to check for an existing resource with the same name as each resource is created, and then require that these resources are Imported
- You can opt into this behaviour from v1.22 onwards, enhancements to come but will be fully available in 2.0

Copyright © 2019 HashiCorp

What's coming over the next few months?

Provider: AzureRM (2.0)

- Custom Timeouts
- Requiring Imports
- New VM / VM Scale Set Resources

Copyright © 2019 HashiCorp

Azure Provider 2.0

New Virtual Machine /
VM Scale Set Resources

Disclaimer: Early Days / WIP

```
resource “azurerm_network_interface” “test” { ... }
resource “azurerm_managed_disk” “test” { ... }

resource “azurerm_linux_virtual_machine” “test” {
  name     = “example-virtual-machine”
  location = “westeurope”
  resource_group_name = “example-resources”
  network_interfaces = [“${azurerm_network_interface.test.id}”]
  username = “myuser”
  ssh_keys = [
    “${file(“~/.ssh/id_rsa.pub”)”,
  ]

  os_disk {
    id = “${azurerm_managed_disk.test.id”
  }
}
```

Copyright © 2019 HashiCorp

What's coming over the next few months?

Provider: AzureRM (2.0)

- Custom Timeouts
- Requiring Imports
- New VM / VM Scale Set Resources
- Removing deprecated fields/resources

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Terraform 0.12

- Azure Backend Enhancements:

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Terraform 0.12

- Azure Backend Enhancements:
  - Fixes a bug where the lock wouldn’t be released

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Terraform 0.12

- Azure Backend Enhancements:
  - Fixes a bug where the lock wouldn’t be released
  - Authenticate using the Azure CLI

Copyright © 2019 HashiCorp

Terraform 0.12

Authenticating using the Azure CLI

```
terraform {
  backend "azurerm" {
    storage_account_name = "abcd1234"
    container_name       = "tfstate"
    key                  = "prod.terraform.tfstate"
  }
}
```

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Terraform 0.12

- Azure Backend Enhancements:
  - Fixes a bug where the lock wouldn’t be released
  - Authenticate using the Azure CLI
  - Authenticate using a Service Principal with a Client Certificate (soon)

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Terraform 0.12

- Azure Backend Enhancements:
  - Fixes a bug where the lock wouldn’t be released
  - Authenticate using the Azure CLI
  - Authenticate using a Service Principal with a Client Certificate (soon)
  - Authenticate using Managed Service Identity

Copyright © 2019 HashiCorp

Terraform 0.12

Authenticating using Managed Service Identity

```
terraform {
  backend "azurerm" {
    storage_account_name = "abcd1234"
    container_name       = "tfstate"
    key                  = "prod.terraform.tfstate"
    use_msi              = true
    subscription_id      = "00000000-0000-0000-0000-000000000000"
    tenant_id            = "00000000-0000-0000-0000-000000000000"
  }
}
```

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Terraform 0.12

- Azure Backend Enhancements:
  - Fixes a bug where the lock wouldn’t be released
  - Authenticate using the Azure CLI
  - Authenticate using a Service Principal with a Client Certificate (soon)
  - Authenticate using Managed Service Identity
  - Authenticate using a SAS Token

Copyright © 2019 HashiCorp

Terraform 0.12

Authenticating using a Storage Access Token

```
terraform {
  backend "azurerm" {
    storage_account_name = "abcd1234"
    container_name       = "tfstate"
    key                  = "prod.terraform.tfstate"

    # rather than defining this inline, the SAS Token can also be sourced
    # from an Environment Variable - more information is available below.
    sas_token = "abcdefghijklmnopqrstuvwxyz0123456789..."
  }
}
```

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Terraform 0.12

- Azure Backend Enhancements:
  - Fixes a bug where the lock wouldn’t be released
  - Authenticate using the Azure CLI
  - Authenticate using a Service Principal with a Client Certificate (soon)
  - Authenticate using Managed Service Identity
  - Authenticate using a SAS Token
  - Support for Azure Stack

Copyright © 2019 HashiCorp

Terraform 0.12

Support for Azure Stack

```
terraform {
  backend "azurerm" {
    storage_account_name = "abcd1234"
    container_name       = "tfstate"
    key                  = "prod.terraform.tfstate"
    environment          = "stack"
    endpoint             = "https://management.westus.mycloud.com"
  }
}
```

Copyright © 2019 HashiCorp

What’s coming over the next few months?

Terraform 0.12

- Azure Backend Enhancements:
  - Fixes a bug where the lock wouldn’t be released
  - Authenticate using the Azure CLI
  - Authenticate using a Service Principal with a Client Certificate (soon)
  - Authenticate using Managed Service Identity
  - Authenticate using a SAS Token
  - Support for Azure Stack
  - Proxy support

Copyright © 2019 HashiCorp

Thank you.

HashiCorp

www.hashicorp.com hello@hashicorp.com