HashiCorp
Terraform

Terraform Workflows

Agenda

- Run Workflows
- Terraform Modules
- Workspaces
- Variables
- Git Repo Structure
- Q+A

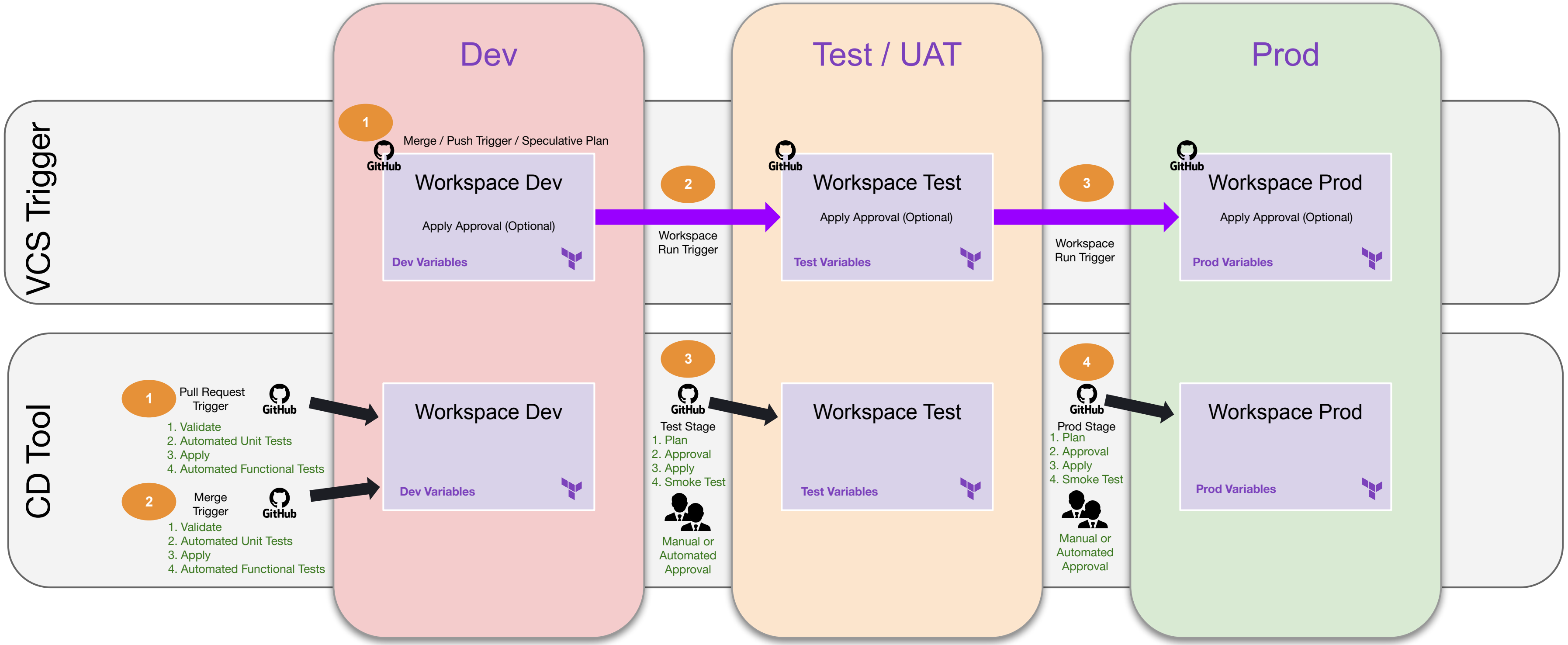
Run Workflows

Run Workflows

- **UI-Driven Runns** - manually trigger runs from the TFC web UI.
- **VCS-Driven Runns** - easiest integration, directly connects a Git Repo to a Terraform Workspace, with automatic runs on Git Commit and Pull Request code changes.
- **CLI-Driven Runs** - easy to use, single CLI command to trigger runs, takes files in the local folder, creates a .zip file, and sends the contents to the TFC API.
- **SDK-Driven Runs** - calls to the TFC API, using a Language Specific integration, available for Golang, Python, and .NET.
- **API-Driven Runs** - full control, all features available to the web UI have an API call, but requires custom coding JSON REST HTTP API calls.

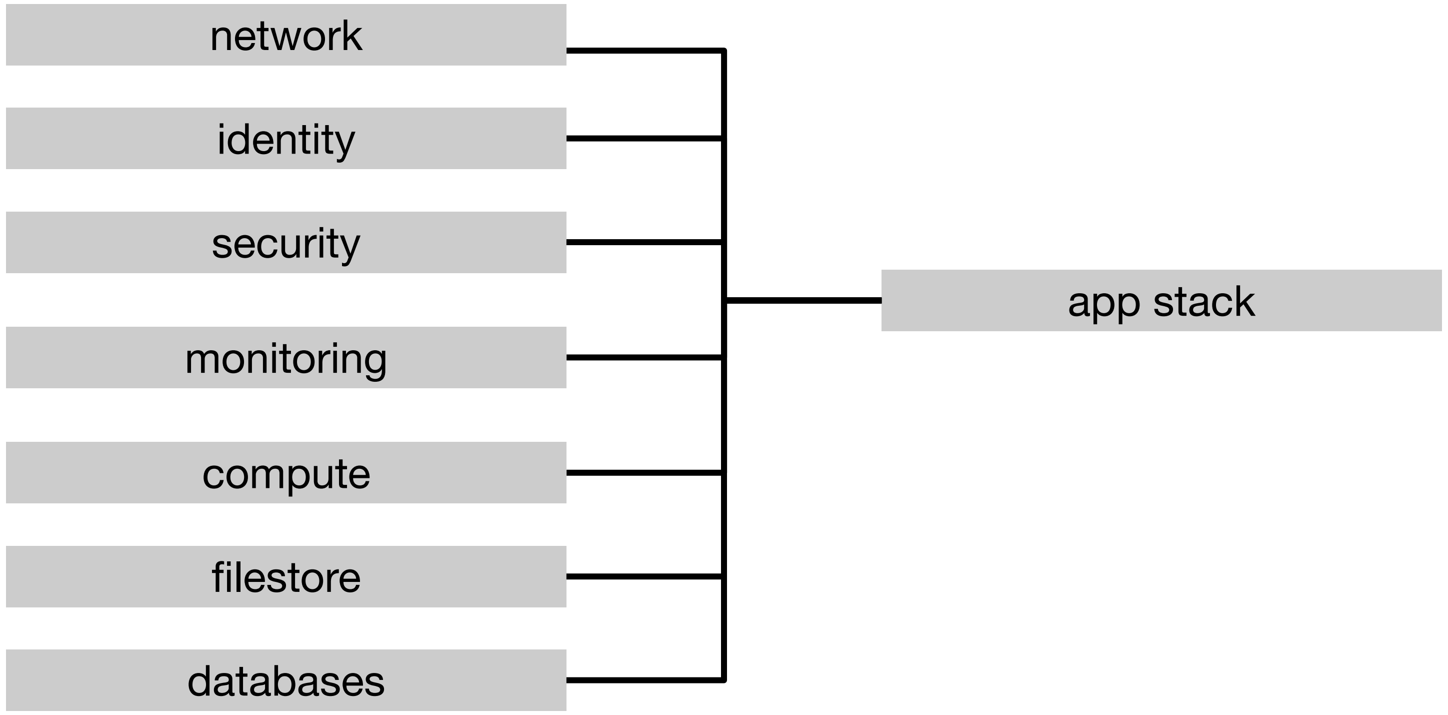
Workflow Types

VCS Trigger vs CLI / API in CD Tool



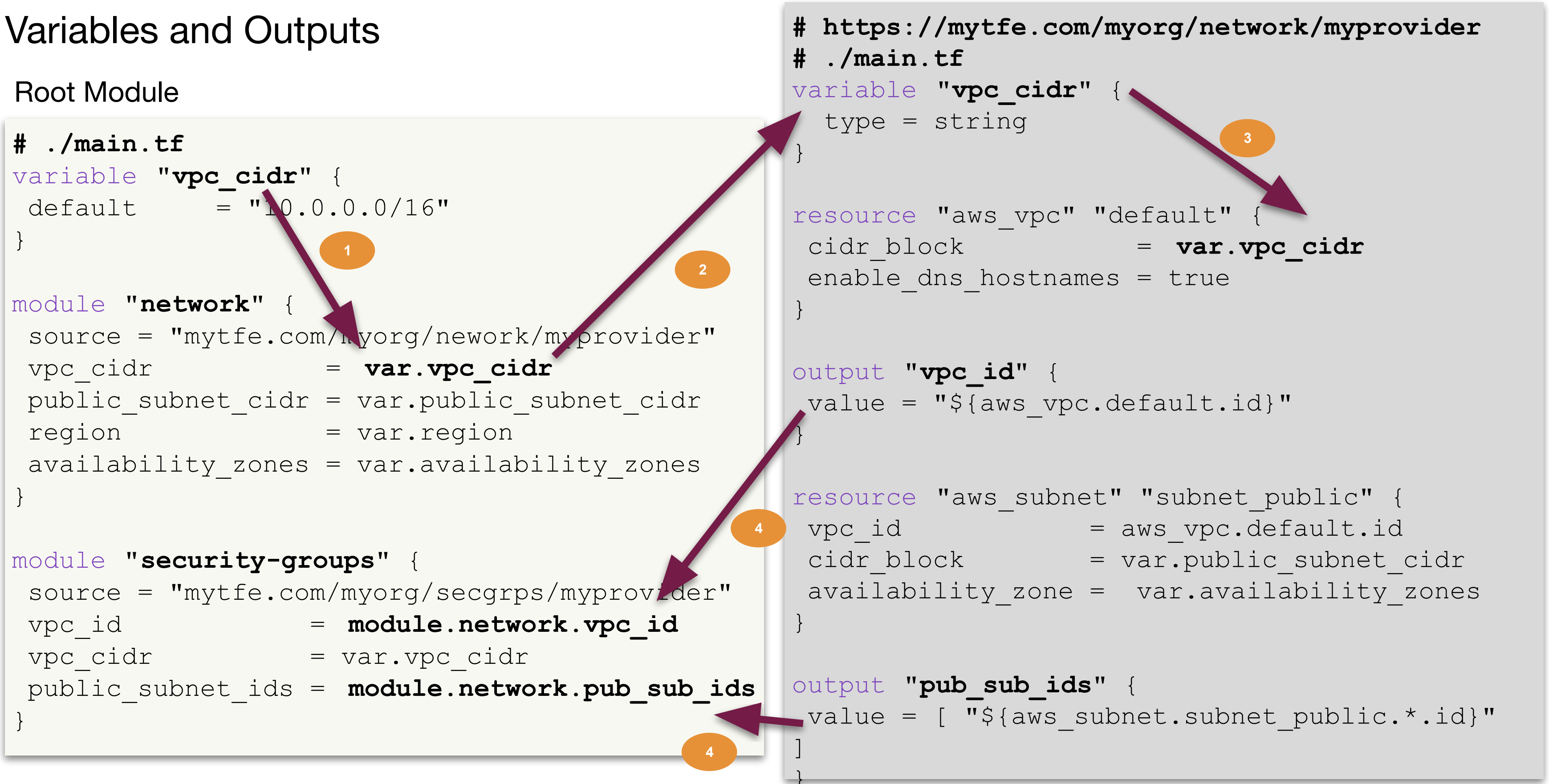
Terraform Modules

Architecture



Code Layout

Variables and Outputs



Root Module

```
# ./main.tf
variable "vpc_cidr" {
 default      = "10.0.0.0/16"
}

module "network" {
 source = "mytfe.com/myorg/nework/myprovider"
 vpc_cidr              = var.vpc_cidr
 public_subnet_cidr = var.public_subnet_cidr
 region                = var.region
 availability_zones = var.availability_zones
}

module "security-groups" {
 source = "mytfe.com/myorg/secgrps/myprovider"
 vpc_id               = module.network.vpc_id
 vpc_cidr             = var.vpc_cidr
 public_subnet_ids = module.network.pub_sub_ids
}
```

Private Sub Module

```
# https://mytfe.com/myorg/network/myprovider
# ./main.tf
variable "vpc_cidr" {
  type = string
}

resource "aws_vpc" "default" {
 cidr_block              = var.vpc_cidr
 enable_dns_hostnames = true
}

output "vpc_id" {
 value = "${aws_vpc.default.id}"
}

resource "aws_subnet" "subnet_public" {
 vpc_id                = aws_vpc.default.id
 cidr_block            = var.public_subnet_cidr
 availability_zone =  var.availability_zones
}

output "pub_sub_ids" {
 value = [ "${aws_subnet.subnet_public.*.id}"
]
}
```

Network

AWS
Route 53 DNS, TLS/SSL Certs, Regions, Availability Zones, VPC, Internet Gateway, Public Subnet, Private Subnet, Route Table, Network ACL, Direct

Azure
VNet, Network Gateway, NAT Gateway, Route Table, Express Route (on-prem), Public IP, Application Gateway

GCP
VPC, Subnet, Cloud NAT, Compute Route, Cloud Interconnect (on-prem), Public IP, API Gateway

VMware
Infoblox DNS / BIND, Verisign / Microsoft AD / Cloud Foundry CA TLS/SSL certs, Regions, Availability Zones, VLAN, Palo Alto / Checkpoint Firewall, DMZ, Internal VLANs, Cisco / Juniper / HP / Dell Route Table, Network ACL, WAN Link / Dark fiber, VMware ESXi / Tanzu NSX Firewall Rules, VMware vLAN

Security

AWS
AWS Config (resource), AWS GuardDuty (NIDS), AWS Macie (S3), VPC Flow Logs

Azure
Azure PolicySets, Network Security Groups, Azure AD Policies

GCP
GCP Security Command Center

VMware
Palo Alto Prisma (resource), Splunk (NIDS), SFlow / NetFlow / Cisco Network Flow Logs, Qualys / Tenable Nessus / Rapid7 Nexpose / Checkpoint (VM, container), Tripwire / OSSEC (FIM)

Identity

AWS
IAM Group, IAM Role, IAM User, IAM Policy (customer-managed)

Azure
Azure AD (Active Directory), Azure Resource Group

GCP
Service Account, Folder, Roles, Policy

VMware
Microsoft Active Directory, LDAP, SAML, Okta

Monitoring

AWS
AWS CloudTrail (cli/sdk), CloudWatch, CloudWatch Metrics

Azure
Azure Network Watcher Flow Log, Monitor

GCP
Network Telemetry, VPC Flow Logs, Cloud Audit Logs

VMware
DataDog / SignalFX / Nagios / SolarWinds, Splunk / ELK / SumoLogic, HP OpenView

Compute

AWS
Load Balancer (ALB, ELB, NLB), Auto-scaling Group + Launch Config + Resource Group + EC2, EKS (K8S), ECS, FarGate (hosted ECS), AWS Lambda

Azure
Traffice Manager (global LB), Scale Set + Launch Config + Resource Group + VM, Azure K8S / AKS

GCP
Load Balancer, Managed Instance Group (MIG) + Instance Template + Stateful Configuration + Compute, GCP EKS / K8S

VMware
F5 / HAProxy / nginx Load Balancers, VMware vRealize, VMware Pivotal Cloud Foundry (PKS, PCS) / K8S

Filestore

AWS
S3, CloudFront (CDN)

Azure
Blob Storage, Content Delivery Network

GCP
Cloud Storage, Cloud CDN

VMware
SAN, NAS, GlusterFS, Minio / Ceph / Dell EMC ECS S3-compatible, Akamai

SQL Databases

AWS
RDS (MySQL, Aurora, Postgresql, MSSQL, Oracle)

Azure
MSSQL, Oracle, MySQL, Postgres

GCP
Cloud SQL (PostgreSQL, MySQL, SQL Server)

VMware
MS SQL Server, Oracle DB, Sybase DB, DB2, MySQL, Postgresql

NoSQL Databases

AWS
ElasticSearch, MongoDB, DocumentDB, Hadoop, DynamoDB

Azure
ElasticSearch, MongoDB, Azure HDInsight Hadoop

GCP
BigQuery, ElasticSearch, MongoDB Atlas, BigTable

VMware
ElasticSearch, MongoDB, Hadoop

In-memory Databases

AWS
ElastiCache (Memcached, Redis)

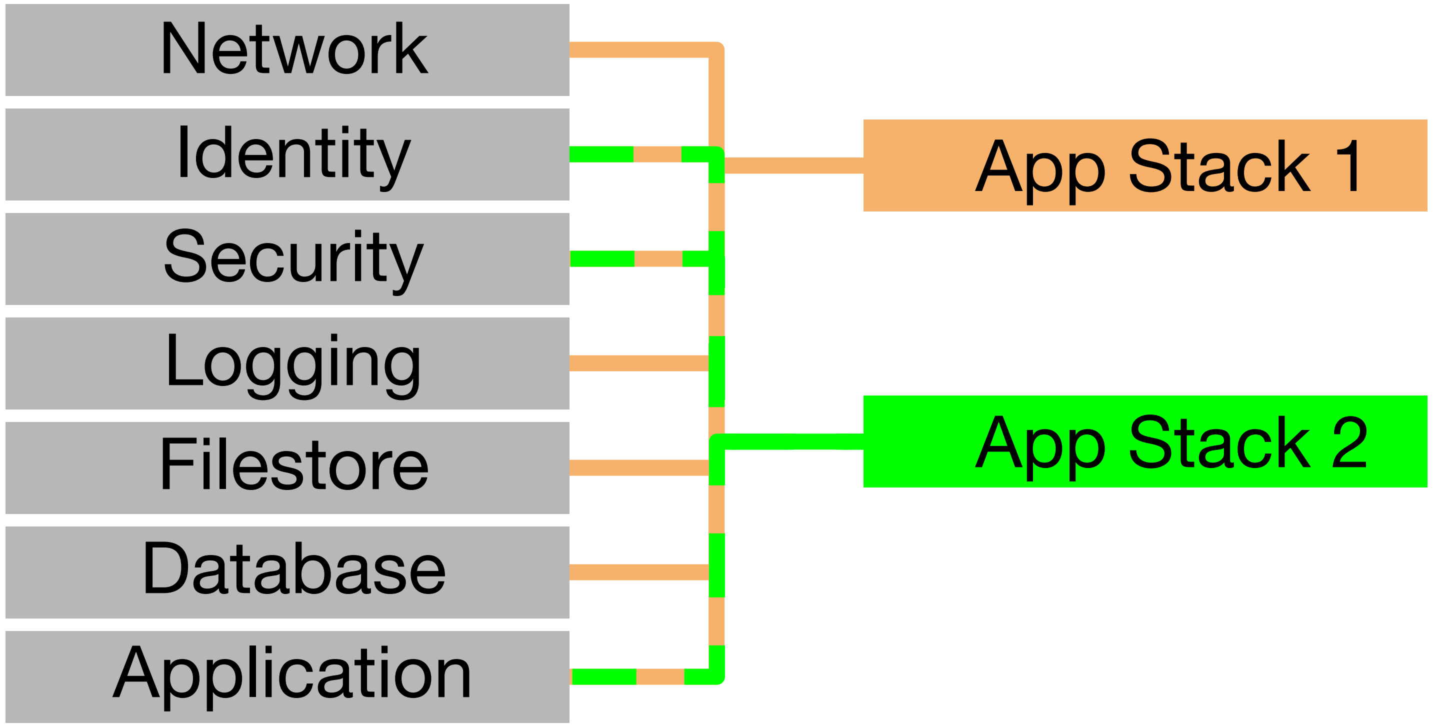
Azure
Azure Cache for Redis

GCP
GCP Memorystore (Redis, Memcached)

VMware
Memcached, Redis

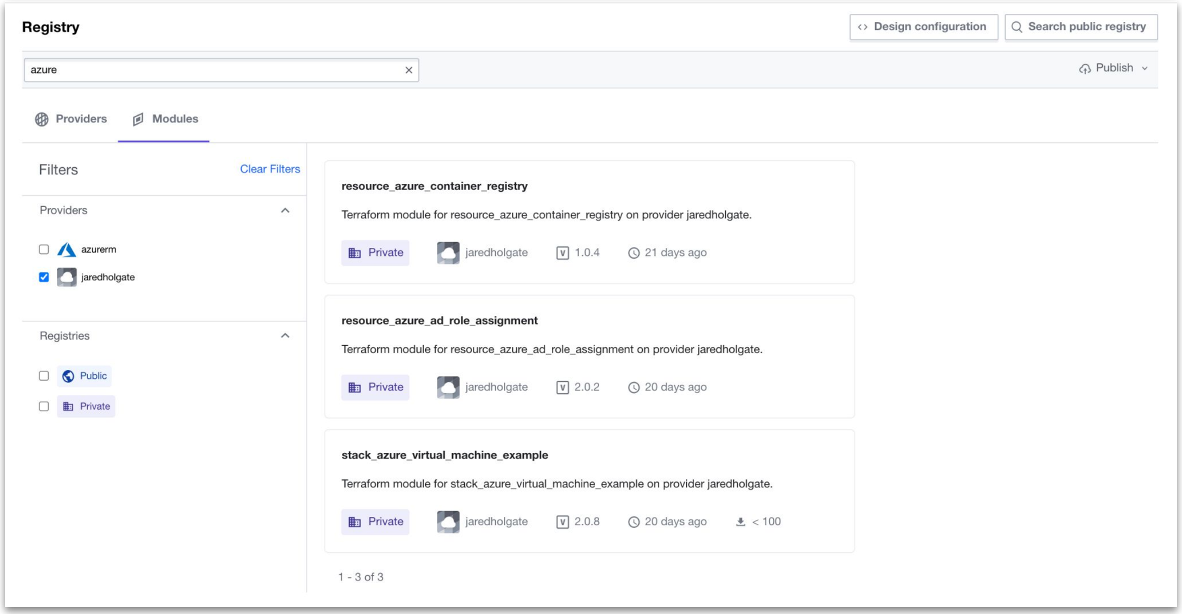
Module Registry

Module Sharing



Private Module Registry

- VCS integration
- Versioning based on VCS tags
- Restrict using Sentinel
- Must follow a specific convention



resource_azure_container_registry

Terraform module for resource_azure_container_registry on provider jaredholgate.

Private jaredholgate 1.0.4 21 days ago

resource_azure_ad_role_assignment

Terraform module for resource_azure_ad_role_assignment on provider jaredholgate.

Private jaredholgate 2.0.2 20 days ago

stack_azure_virtual_machine_example

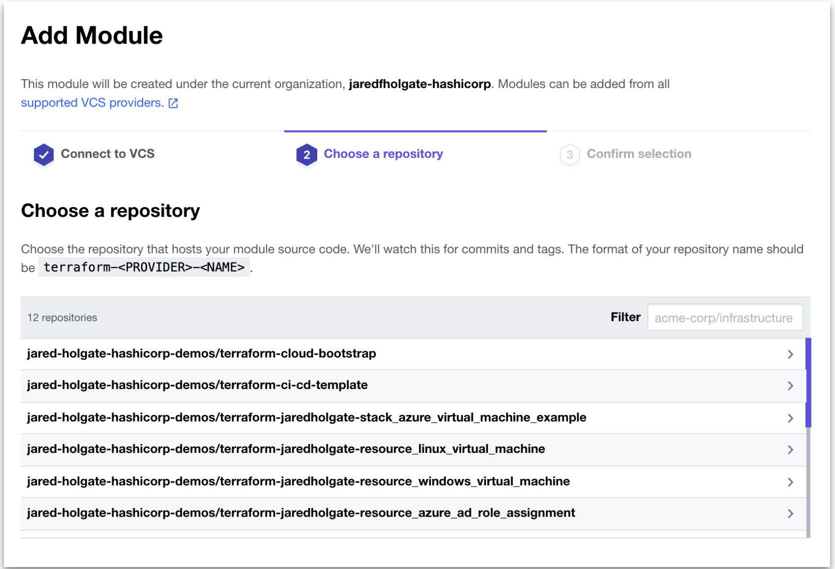
Terraform module for stack_azure_virtual_machine_example on provider jaredholgate.

Private jaredholgate 2.0.8 20 days ago < 100

1 - 3 of 3

Private Module Registry

- Repo name must follow the convention: terraform-<provider>-<module name>
  e.g. terraform-myorganisation-azure_network
- Must have a README.md
- Must have a main.tf file
- Must have a version tag in x.y.z format



Add Module

This module will be created under the current organization, **jaredfholgate-hashicorp**. Modules can be added from all supported VCS providers.

Choose a repository

Choose the repository that hosts your module source code. We'll watch this for commits and tags. The format of your repository name should be `terraform-<PROVIDER>-<NAME>`.

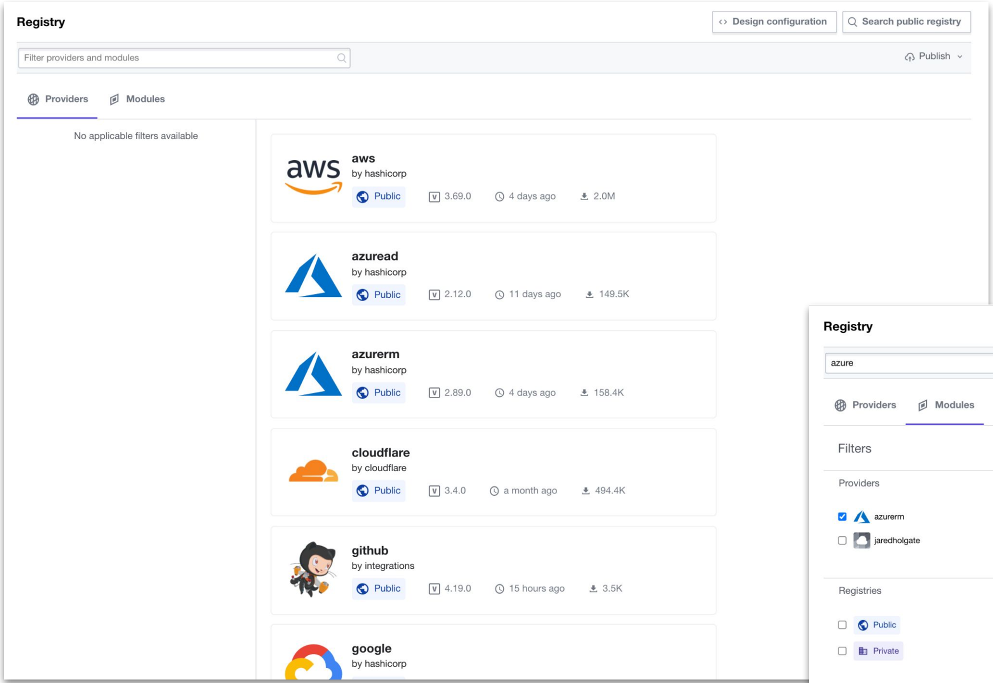
12 repositories

Filter acme-corp/infrastructure

- jared-holgate-hashicorp-demos/terraform-cloud-bootstrap
- jared-holgate-hashicorp-demos/terraform-ci-cd-template
- jared-holgate-hashicorp-demos/terraform-jaredholgate-stack_azure_virtual_machine_example
- jared-holgate-hashicorp-demos/terraform-jaredholgate-resource_linux_virtual_machine
- jared-holgate-hashicorp-demos/terraform-jaredholgate-resource_windows_virtual_machine
- jared-holgate-hashicorp-demos/terraform-jaredholgate-resource_azure_ad_role_assignment

Public Providers and Modules

- Specify which providers and modules are recommended
- Restrict using Sentinel



Registry

Design configuration | Search public registry

Filter providers and modules

Publish

Providers | Modules

No applicable filters available

aws
by hashicorp
Public | 3.69.0 | 4 days ago | 2.0M

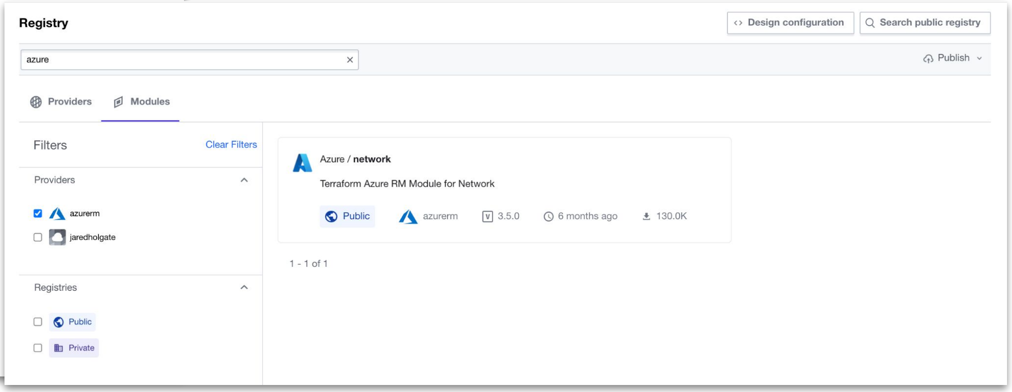
azuread
by hashicorp
Public | 2.12.0 | 11 days ago | 149.5K

azurerm
by hashicorp
Public | 2.89.0 | 4 days ago | 158.4K

cloudflare
by cloudflare
Public | 3.4.0 | a month ago | 494.4K

github
by integrations
Public | 4.19.0 | 15 hours ago | 3.5K

google
by hashicorp



Registry

Design configuration | Search public registry

azure

Publish

Providers | Modules

Filters | Clear Filters

Providers
- [✓] azurerm
- [ ] jaredholgate

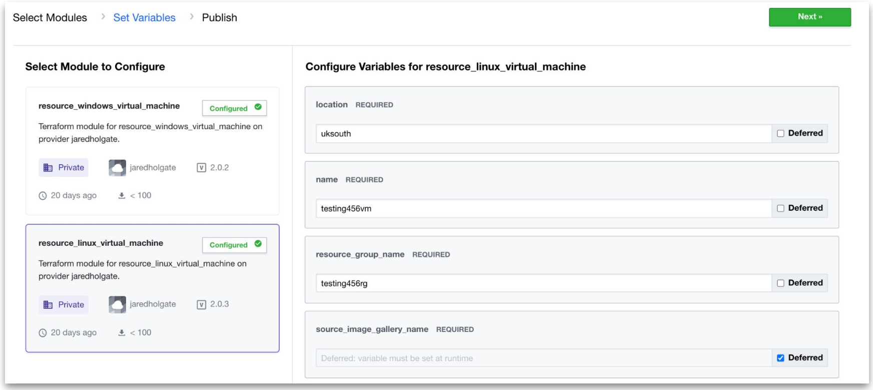
Registries
- [ ] Public
- [ ] Private

Azure / **network**
Terraform Azure RM Module for Network
Public | azurerm | 3.5.0 | 6 months ago | 130.0K

1 - 1 of 1

Configuration Designer

- Helps to write HCL
- Still need to source control and have a workspace



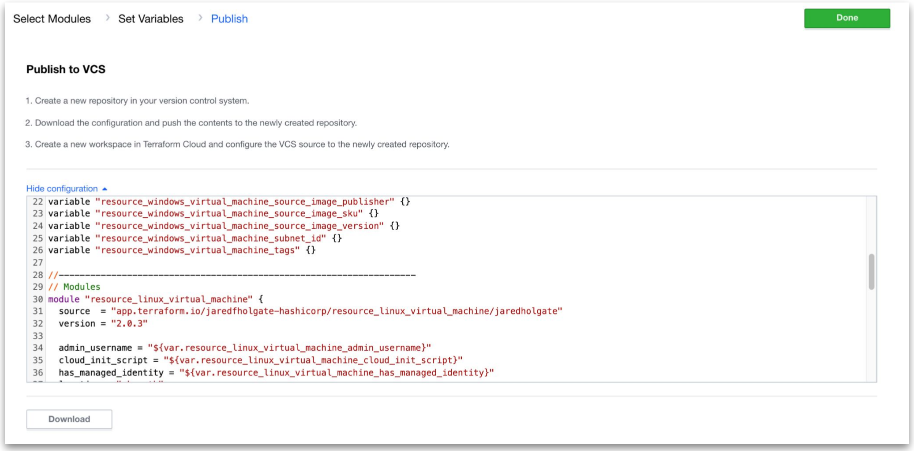
Select Module to Configure

resource_windows_virtual_machine — Configured ✓
Terraform module for resource_windows_virtual_machine on provider jaredholgate.
Private · jaredholgate · 2.0.2
20 days ago · < 100

resource_linux_virtual_machine — Configured ✓
Terraform module for resource_linux_virtual_machine on provider jaredholgate.
Private · jaredholgate · 2.0.3
20 days ago · < 100

Configure Variables for resource_linux_virtual_machine

location REQUIRED	uksouth	☐ Deferred
name REQUIRED	testing456vm	☐ Deferred
resource_group_name REQUIRED	testing456rg	☐ Deferred
source_image_gallery_name REQUIRED	Deferred: variable must be set at runtime	☑ Deferred



Publish to VCS

1. Create a new repository in your version control system.
2. Download the configuration and push the contents to the newly created repository.
3. Create a new workspace in Terraform Cloud and configure the VCS source to the newly created repository.

Hide configuration ▲

```
variable "resource_windows_virtual_machine_source_image_publisher" {}
variable "resource_windows_virtual_machine_source_image_sku" {}
variable "resource_windows_virtual_machine_source_image_version" {}
variable "resource_windows_virtual_machine_subnet_id" {}
variable "resource_windows_virtual_machine_tags" {}

//-------------------------------------------------------------------
// Modules
module "resource_linux_virtual_machine" {
  source  = "app.terraform.io/jaredfholgate-hashicorp/resource_linux_virtual_machine/jaredholgate"
  version = "2.0.3"

  admin_username = "${var.resource_linux_virtual_machine_admin_username}"
  cloud_init_script = "${var.resource_linux_virtual_machine_cloud_init_script}"
  has_managed_identity = "${var.resource_linux_virtual_machine_has_managed_identity}"
```

Resources

- https://learn.hashicorp.com/tutorials/terraform/module-private-registry-add?in=terraform/modules
- https://learn.hashicorp.com/tutorials/terraform/module-private-registry-share?in=terraform%2Fmodules
- https://www.terraform.io/docs/registry/index.html
- https://www.terraform.io/docs/cloud/registry/publish.html
- https://www.terraform.io/docs/cloud/registry/add.html

Workspaces

Considerations

- **Blast-Radius**: Do not put everything in one place.
- **Least Privilege**: Divide cloud resources into multiple Workspaces so that a Team cannot change another Team's cloud resources.
- **Rate of Change**: The Networking layer will not change as often as the Compute layer. Common changes should not affect uncommonly changing resources.
- **Ease of Maintenance**: Group similar resources to ensure maintenance changes don't affect other components, ex: upgrading all instances of Postgres / MySQL / MS SQL / Oracle / ElasticSearch should not affect the Networking resource.

1. Monolithic Workspace

Production	Staging	QA	Dev
network	network	network	network
security	security	security	security
identity	identity	identity	identity
compute	compute	compute	compute
filestore	filestore	filestore	filestore
sql	sql	sql	sql

2. Production vs. Non-production

Production	Staging	QA	Dev
network	network	network	network
security	security	security	security
identity	identity	identity	identity
compute	compute	compute	compute
filestore	filestore	filestore	filestore
sql	sql	sql	sql

3. Prod vs. Non-prod w/ Landing Zones

Production	Staging	QA	Dev
network	network	network	network
security	security	security	security
identity	identity	identity	identity
compute	compute	compute	compute
filestore	filestore	filestore	filestore
sql	sql	sql	sql

4. Divided by Environments (Envs)

Production	Staging	QA	Dev
network	network	network	network
security	security	security	security
identity	identity	identity	identity
compute	compute	compute	compute
filestore	filestore	filestore	filestore
sql	sql	sql	sql

5. Isolated Envs w/ Landing Zones (LZs)

Production	Staging	QA	Dev
network	network	network	network
security	security	security	security
identity	identity	identity	identity
compute	compute	compute	compute
filestore	filestore	filestore	filestore
sql	sql	sql	sql

6. Isolated Envs w/ LZs and App Layers

Production	Staging	QA	Dev
network security identity	network security identity	network security identity	network security identity
compute	compute	compute	compute
filestore	filestore	filestore	filestore
sql	sql	sql	sql

7. Isolated Envs w/ Shared App Layers

Production	Staging	QA	Dev
network	network	network	network
security	security	security	security
identity	identity	identity	identity
compute	compute	compute	compute
filestore	filestore	filestore	filestore
sql	sql	sql	sql

8. Isolated Envs w/ Isolated Layers

Production	Staging	QA	Dev
network	network	network	network
security	security	security	security
identity	identity	identity	identity
compute	compute	compute	compute
filestore	filestore	filestore	filestore
sql	sql	sql	sql

Terraform tfe Provider

Automate Terraform Cloud Configuration

- https://registry.terraform.io/providers/hashicorp/tfe/
- tfe = Terraform Enterprise
- Works with Terraform Cloud and Terraform Enterprise
- Requires a Token argument, which is the API Token
- Comprehensive resource and data source coverage

Workspace Creation Automation

```
# Configure a TF Workspace Variable called
# “tf_token” with the TFE API Token
terraform {
  required_providers {
    tfe = {
      source = "hashicorp/tfe"
      version = "~> 0.25.3"
    }
    null = {
      source = "hashicorp/null"
      version = "~> 3.1.0"
    }
  }
}
# https://registry.terraform.io/providers/hashicorp/tfe/latest/docs
provider "tfe" {
  hostname = var.tf_hostname
  token    = var.tf_token
}
```

Workspace Creation Automation

```
variable "tf_organization" {
  type = string
  default = "Pyrocumulus"
}

variable "tf_workspaces" {
  type = set(string)
  default = ["workspaceA", "workspaceB",
    "workspaceC"]
}

resource "tfe_workspace" "test" {
  for_each = var.tf_workspaces
  name = each.key
  organization = var.tf_organization
}

output "tf_workspace_ids" {
  value = { for k, v in tfe_workspace.test :
    k => v.id }
}
```

```
resource "tfe_variable" "test" {
  for_each = { for k, v in
tfe_workspace.test:
    k => v.id }
  key = "test_key_name"
  value = "test_value_name"
  category = "terraform"
  workspace_id = each.value
}

resource "tfe_team" "test" {
  name = "test-team-name"
  organization = var.tf_organization
}

resource "tfe_team_access" "test" {
  for_each = { for k, v in
tfe_workspace.test:
    k => v.id }
  access = "read"
  team_id = tfe_team.test.id
  workspace_id = each.value
}
```

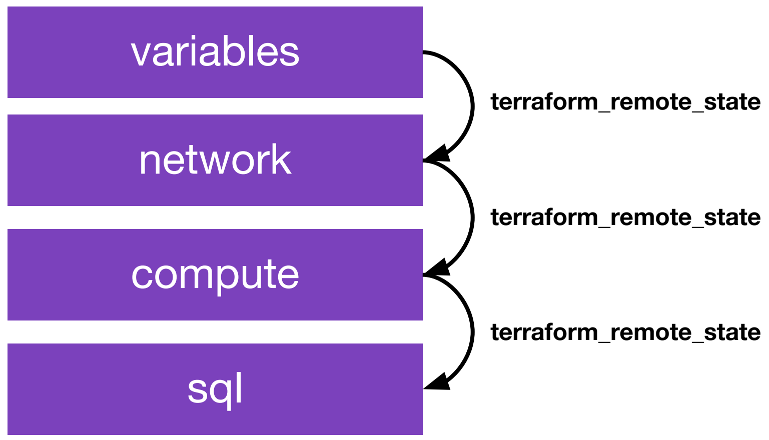
Workspace Variables

Workspaces, Secrets / Credentials

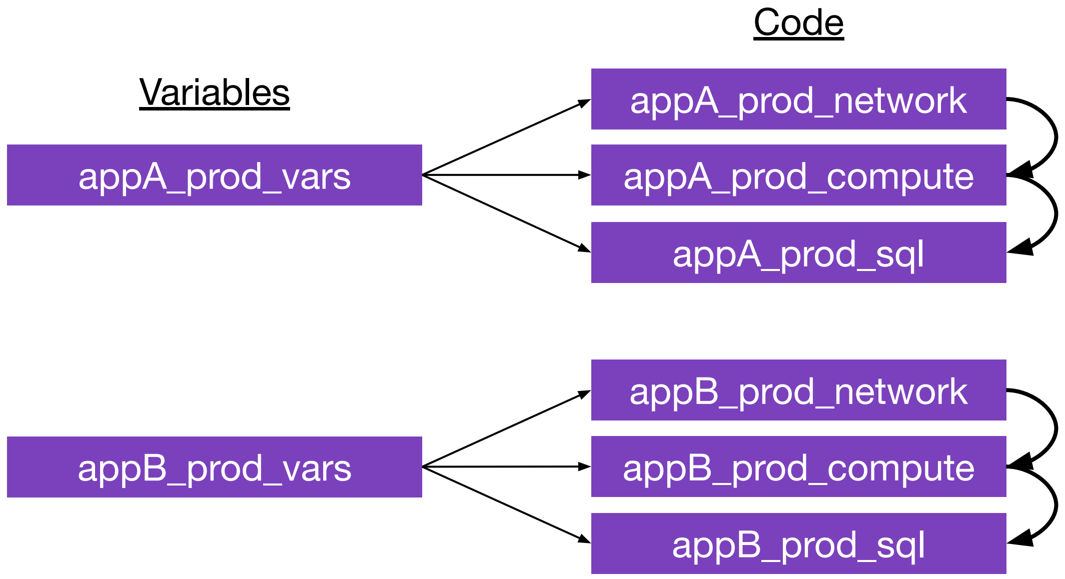
1. Vault Enterprise
2. Vault Open Source
3. Cloud Agents, with Cloud Identity Credentials (ex: AWS IAM Instance Profile)
4. Variable Sets (beta)
5. **terraform_remote_state** data source, read between Workspaces
6. Workspace Variable, Sensitive
7. Workspace Environment Variable, Sensitive
8. CI/CD Inject Credentials at Run-time

https://www.hashicorp.com/blog/managing-credentials-in-terraform-cloud-and-enterprise

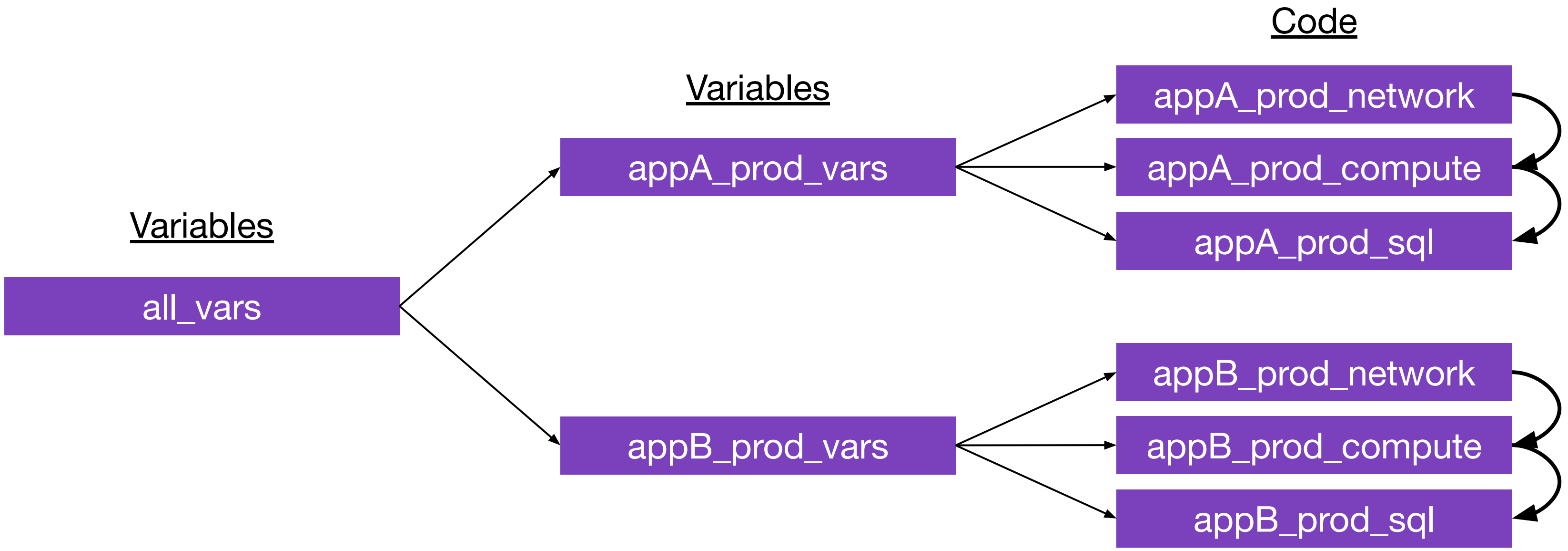
Changes Across Workspaces with Run Triggers



Changes Across Workspaces with Run Triggers



Changes Across Workspaces with Run Triggers



Git Repository Structure

MonoRepo vs MultiRepo

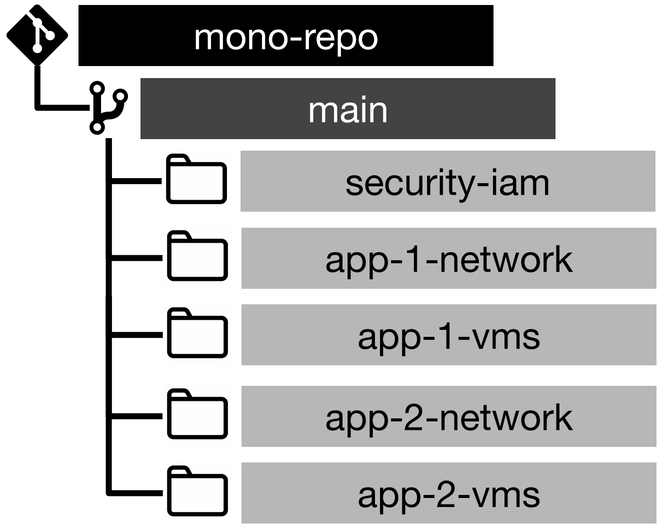
TFE Supports both models;

- **MonoRepo:** Single Repo organised by folder
- **MultiRepo:** Repo per application / component

- For Private Registry Modules you must use a repo per module when using VCS integration.

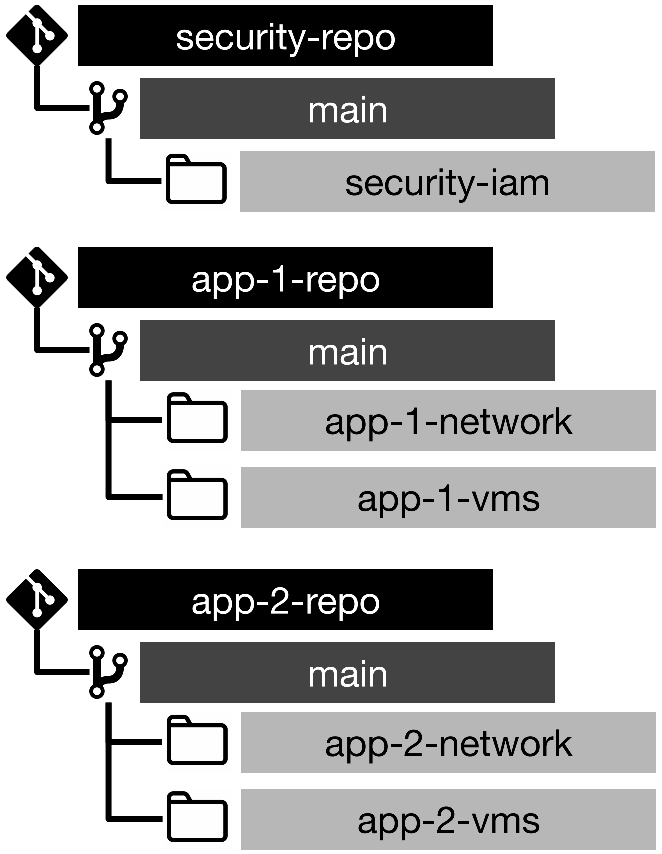
Caution: Managing a large MonoRepo can be complex and may impact performance

Repository Structure MonoRepo



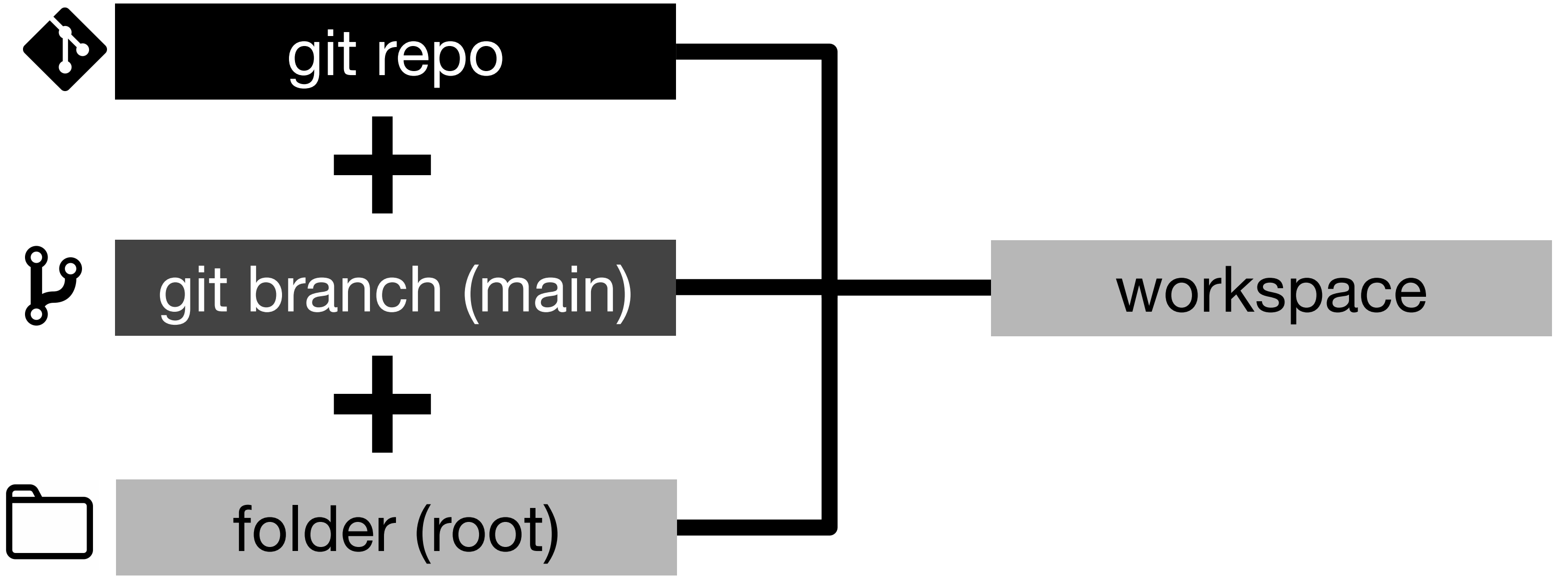
- Workspace per Env
- Single Git Repo
- Large git clones
- git tag / version is applied across the whole repo

Repository Structure MultiRepo



- Workspace per Env
- Multi Git Repo
- Small git clones
- git tag / version is per application

Components, for VCS-Driven Runs

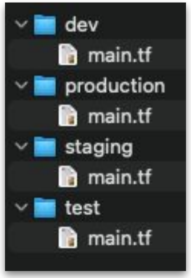


Code Structure for Environments

Three main options

Many Code copies with hard coded variables

Do not use variables, simply hard code the differences between environments directly in the HCL files.

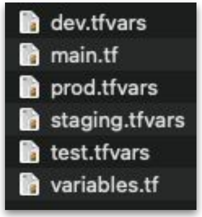


```
dev
    main.tf
production
    main.tf
staging
    main.tf
test
    main.tf
```

Single Code copy with variable files (API / CLI)

Use variables in the code and store any non-secret variables in source control per environment.

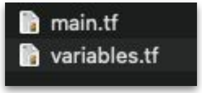
This pattern can be used for API / CLI driven workflows. You can specify the tfvars file on the command line or copy it to a *.auto.tfvars file.



```
dev.tfvars
main.tf
prod.tfvars
staging.tfvars
test.tfvars
variables.tf
```

Single Code copy with deploy time variables

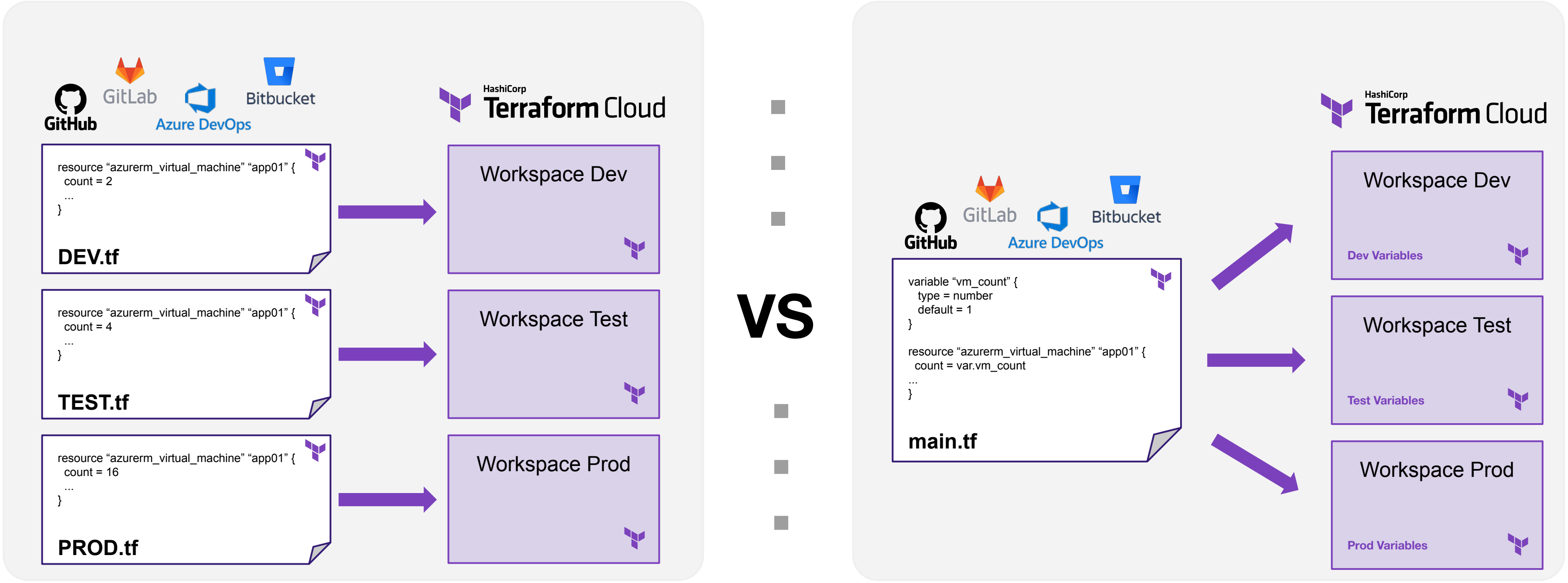
Use variables in the code and inject from the workspace or CD tool at deploy time.



```
main.tf
variables.tf
```

Code Structure for Environments

Many Copies vs Single Copy



```
resource “azurerm_virtual_machine” “app01” {
  count = 2
  ...
}
```

```
resource “azurerm_virtual_machine” “app01” {
  count = 4
  ...
}
```

```
resource “azurerm_virtual_machine” “app01” {
  count = 16
  ...
}
```

```
variable “vm_count” {
   type = number
   default = 1
}

resource “azurerm_virtual_machine” “app01” {
  count = var.vm_count
  ...
}
```

Branching Strategies

Main options for Branching with TFE

Trunk based

Trunk based branching refers to having a single branch that you deploy from.

Changes get into the trunk branch via a Pull Request from a short lived feature branch.

Run Triggers or a CD tool are used to promote to environments.

Git Flow or similar

More complicated structure, you may have a dev branch that deploys to dev environment and release and hotfix branches that go to staging and production.

You would still likely use Run Triggers or a CD tool to get from staging to production.

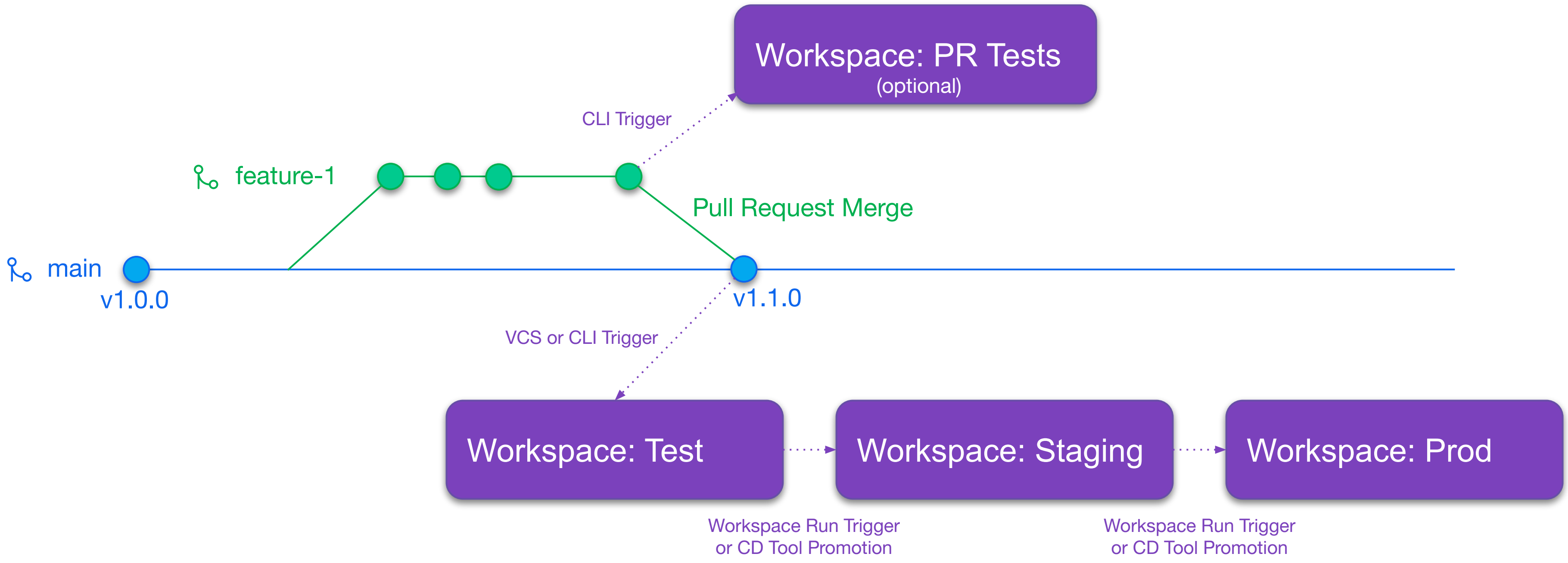
Branch per Environment

Each environment has its own branch associated with a workspace.

To kick off runs to different environments, the branches are merged into each other going up the chain.

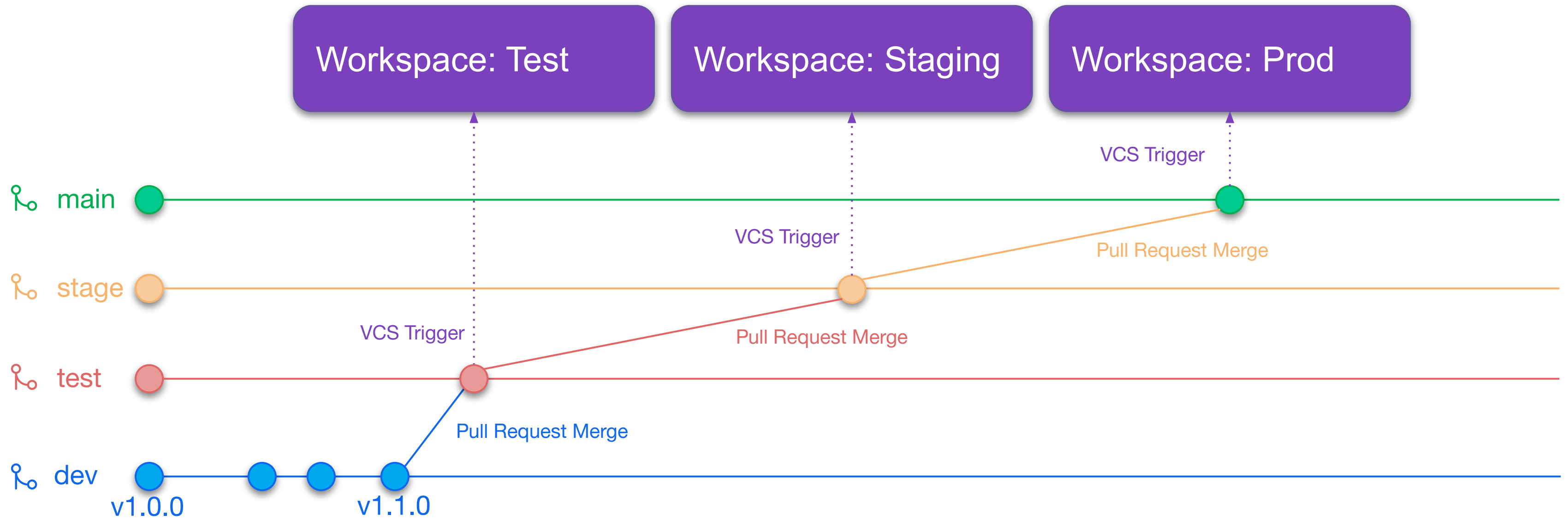
Branching Strategies

Trunk Based



Branching Strategies

Branch per Environment



How refactor a Git Monorepo

1. Refactor to use Terraform Modules
2. Create Git Repos for each Terraform Module
3. Created Git Repos of Terraform Code for each App
4. Migrate the code from the MonoRepo folder into the new repository
5. Update the Workspace VCS configuration

Next Steps

Need Additional Help?

Customer Success

Contact our Customer Success Management team with any questions. We will help coordinate the right resources for you to get your questions answered.

customer.success@hashicorp.com

Technical Support

Something not working quite right? Engage with HashiCorp Technical Support by opening a new ticket for your issue at support.hashicorp.com.

Upcoming Onboarding Webinars

May 31:

Webinar: **Terraform Enterprise Arch Deep Dive (Active/Active) and Community Office Hours**

An interactive open forum to discuss specific questions about your environment and Use Cases. Please bring your questions.

June 7:

Webinar: **Terraform Cloud Agents, RBAC & Sentinel / Architecture**

June 14:

Program Closing

Q & A

Thank You

customer.success@hashicorp.com
www.hashicorp.com