

SHIFTING SECURITY AND SANITY LEFT BY TESTING TERRAFORM WITH INSPEC

Presented by Annie Hedgpeth

{X}

Hello to all of you out there! This is a pre-recorded video, because it's a time of day that my house is full of children, but there is a high likelihood that I'll be in the chat, so feel free to ask questions, and I hope to get to them! Today, however, I'm still so happy to be here with you today to share with you my ideas about "SHIFTING SECURITY AND SANITY LEFT BY TESTING TERRAFORM WITH INSPEC".



**THE VELOCITY
TEMPTATION**
**LOWER THE
BARRIER TO
ENTRY**

{X}

I am willing to bet that you have been on projects either in the past or even right now where the temptation or even expectation is to go fast and deliver as quickly as possible. And it is possible that testing was seen as an extra or something that you would add later when you have a couple of slow sprints.

And you may even be saying, “What? We test. Isn’t terraform plan or terraform output enough?” “Isn’t anything more kind of a waste of time?” One could certainly argue that, but those of you who have raced to get a Terraform project finished and then gotten stuck in a bottleneck of bugs and manual testing know that you could have done more in the beginning for your own sanity later down the road. Trust me, your future self or future teammates will thank you later for implementing testing EARLY.

And the other thing you might be asking is, “Aren’t there better Terraform testing methods than InSpec? What about Terra-test?” And the truth is...it depends, but I personally think that the barrier to entry is lowest with InSpec. And if you’re stuck in fast mode but wisely know that you need testing, then using something that is easy to onboard to could be the key to getting these tests in the pipeline quickly.

The Site Reliability Workbook

... the later in the product lifecycle a problem is discovered, the more expensive it is to fix . SREs are specifically charged with improving undesirably late problem discovery, yielding benefits for the company as a whole.

And if that isn't a resounding endorsement for early Terraform testing, then I don't know what is.



{X}

10TH MAGNITUDE

Annie Hedgpeth

Senior Cloud Automation Engineer

@anniehedge

anniehedgpeth

www.anniehedge.com

Microsoft
Partner



Gold DevOps
Gold Cloud Platform
Gold Application Development
Gold Datacenter
Gold Data Analytics

Gold
Microsoft
Partner



Azure
Expert
MSP

My name is Annie Hedgpeth, and I'm a cloud automation engineer at 10th Magnitude. My job includes a lot of configuration management as well as infrastructure as code development, so the tools that I have found myself using the most have been Chef and InSpec for configuration management and Terraform for infrastructure as code provisioning.

As you may have deduced from this slide, my examples will be using Azure as my Cloud provider of choice, but rest assured that everything I'm doing can be done in any cloud or even on-prem.

SECURITY & SANITY

USE-CASES FOR TESTING TERRAFORM WITH INSPEC

- Production provisioning validation
- Test Driven Development (TDD)
- Integration Testing and CI/CD
- Compliance, shifting security left

{X}

I titled this talk "SHIFTING SECURITY AND SANITY LEFT BY TESTING TERRAFORM WITH INSPEC" because I'm going to walk you through the following Use-Cases for how to do just that:

1. let's say you're provisioning a production environment. Well, you can run an InSpec profile after that to validate that it's in compliance and in its desired state.

2. you can use InSpec for test driven development (which means that you write a test, watch it fail, and then write the code to remediate that failure)

3. you can use it for integration testing in a CI/CD pipeline by running tests at each pull request and not letting it merge to master without a passing test-kitchen build

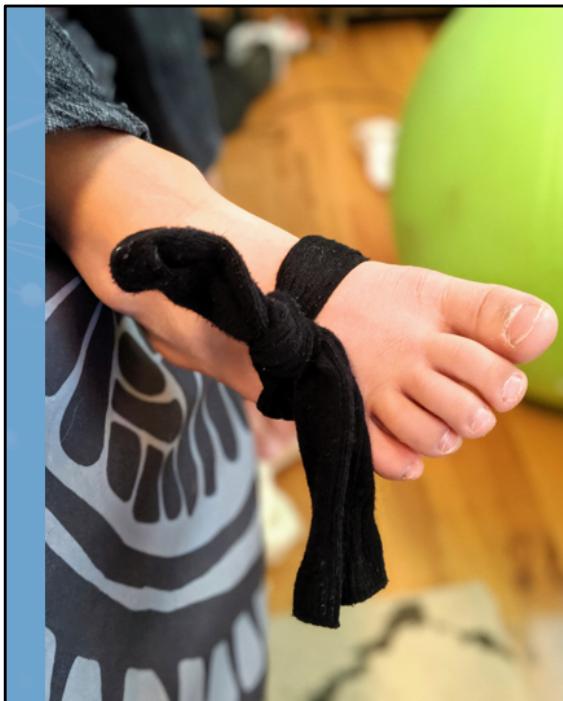
4. you can add all of your compliance testing to these early testing builds so that you don't find out much too late that major changes for the sake of security are required



WHAT IS INSPEC?

{X}

Some of you may have never used InSpec, so I'm going to give you a quick crash course on what it is.



INSPEC (inspec.io)

1. An open-source language for testing and auditing your applications and infrastructure
2. Works by comparing the actual state of your system with the desired state that you express in easy-to-read and easy-to-write Chef InSpec code
3. Detects violations and displays findings in the form of a report, but puts you in control of remediation

Create
a
Profile

Add
Your
Tests

Target
Your
System

{X}

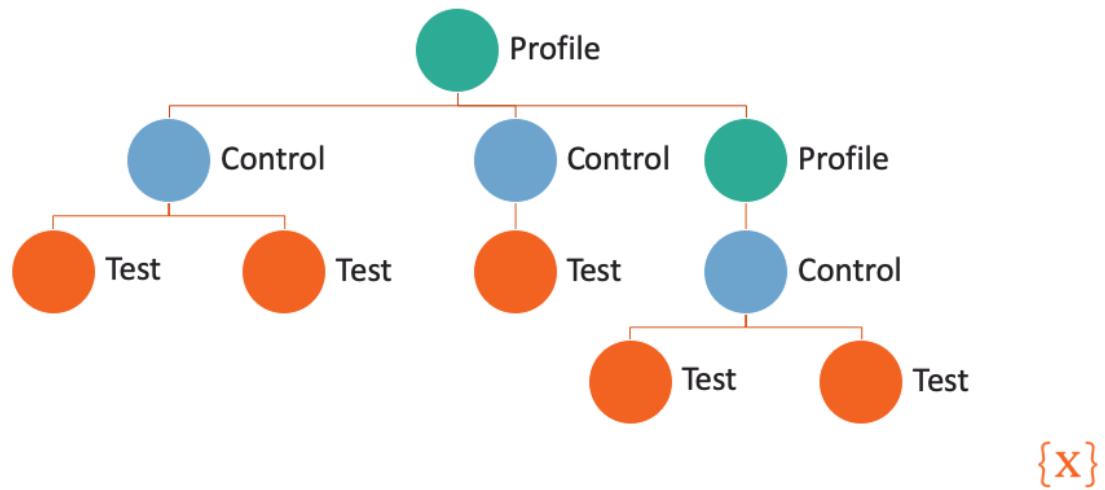
You may have figured out by now that InSpec is an open-source language for testing and auditing your applications and infrastructure.

It compares the actual state of your system with the desired state that you expressed in the....and this is a great thing about it....easy-to-read and easy-to-write code.

If it detects violations, then it will tell you in a report from which you can choose how to remediate.

Kudos to you if by now you have figured out why I added this picture of my son's foot. I told him to put his socks on, and he said he did. Instead of putting it on as one normally does, he tied it around his foot. As you can see, this is clearly not desired state.

INSPEC STRUCTURE



Here's the way that InSpec profiles are structured.

You have a profile that is made up of one or many controls. It can also inherit other profiles and its controls.

And each control is made up of one or many tests.

INSPEC TEST

```
describe docker_container('container') do
  it { should exist }
  it { should be_running }
  its('id') { should_not eq '' }
  its('image') { should eq 'busybox:latest' }
  its('repo') { should eq 'busybox' }
  its('tag') { should eq 'latest' }
  its('ports') { should eq [] }
  its('command') { should eq 'nc -l -p 1234 -e /bin/cat' }
end
```

{X}

Here's what a test looks like, and take note of the ease at which you can figure out what it's testing.

This particular test is using the docker_container resource. There are a myriad of different resources that you can use to test different things.

This one is saying, "Describe my Docker container called 'container'. It should exist. It should be running. Its id should not equal nothing. Its image should be busybox:latest. Etc."

INSPEC CONTROL

```
## Attributes
# Hard code as much as possible because these are tests
resource_group = attribute("resource_group", description: "Azure Resource Group Name")

control 'resources' do
  impact 0.6
  title 'Azure resources'
  desc 'All of the Azure resources exist in the desired state.'
  tag 'azure'

  describe azurerm_virtual_machine(resource_group: resource_group, name: 'my_vm') do
    it { should exist }
    its('type') { should eq 'Microsoft.Compute/virtualMachines' }
  end

  describe azurerm_network_security_group(resource_group: resource_group, name: 'nsg') do
    it { should exist }
    its('type') { should eq 'Microsoft.Network/networkSecurityGroups' }
    its('security_rules') { should_not be_empty }
    its('default_security_rules') { should_not be_empty }
    it { should allow_ssh_from_internet }
  end

  describe azurerm_virtual_network(resource_group: resource_group, name: 'my_network') do
    it { should exist }
    its('location') { should eq 'centralus' }
  end
end
```

METADATA

TEST TEST TEST

CONTROL

{X}

You would take that test and nest it inside of a “control”, which also has some metadata options. Those are especially helpful if you’re testing for compliance, then that metadata would map to the specific audits.

You can also output information from the Terraform run and use it as an attribute in the profile, like here with the resource group name.

INSPEC PROFILE

```
examples/profile
├── README.md
├── controls
|   ├── example.rb
|   └── control_etc.rb
├── libraries
|   └── extension.rb
├── files
|   └── extras.conf
└── inspec.yml
```

Where to Store it:

- Locally
- Source control (git)
- Chef Supermarket
- Chef Automate Server

{X}

So now, we take those controls, put them in a controls directory, and nest them in a profile, where you can also have a README, libraries for custom resources, files if you need them, and an inspec.yml that defines the config of your profile.

INSPEC AS A LOCAL_EXEC

```
resource "null_resource" "inspec" {
  provisioner "local-exec" {
    command = <>EOT
    - inspec exec https://github.com/anniehedgpeth/inspec-azure-demo.git -t azure://${var.subscription_id}
    - inspec exec https://github.com/anniehedgpeth/demo_profile -t ssh://user@ipaddress
    EOT
  }
  depends_on = [
    your.last.resource.provisioned
  ]
}
```

Benefits:

- Good for production validation
- Good for compliance auditing

Problems

- The resources might not be finished provisioning in time for the tests.
- You need separate commands for each InSpec session / target (i.e. vm or subscription).
- If you're developing, then it's cumbersome

{X}

In our first “use-case” I talked about how you can use InSpec to validate a production deployment. One way of doing this is simply to run the InSpec executable directly in the module as a local_exec inside a null resource at the end of the module.

The nice thing about this is that it will validate that it’s in the desired state. If it’s not, then you can alert someone that something went wrong with the deployment. Maybe someone pushed some code that changed the access key to the storage account so that it’s not accessible anymore, and you would sure want to know that right away, so InSpec could catch that and alert you.

The other thing that security teams would appreciate is if you tacked on a profile that defines all of your compliance audits as code, so as soon as you finish deploying, you can show your security team that this environment is in compliance.

Some gotchas that you’d need to be aware of, though, are that:

- 1) Some of your resources may not be completely finished provisioning, but you can try to plan around that depending on what it is.
- 2) An InSpec profile has to run per target, so you’d run the executable against each target separately. In this example, the first command runs it against the Azure

subscription, and the second one runs it against the vm being provisioned. It's going to check that the post-provisioning configuration on that node worked and put the machine in the desired state.

- 3) Another thing to note, is that you wouldn't want to use this method for test driven development. Even if you used a separate tfvars file specific to your test environment, and even if you know that there's no way that you'll interfere with your state file, or ruin anything in production, do you really want to take that risk? Also, it's just cumbersome to run each time you want to test a change because depending on the kind of failure you get and whether it's a Terraform failure or InSpec failure, it's confusing as to what will and won't fail the build and if you want it to fail at that point in testing.

TEST KITCHEN...NOT JUST FOR CHEF



<https://github.com/newcontext-oss/kitchen-terraform>

{X}

A better solution than that for Test Driven Development of Terraform modules is using Test Kitchen. Now, Test Kitchen is most popular for being used with Chef as a way to spin up a test node, run your Chef cookbook on it, and then test it with InSpec. This nice little sandbox provides the perfect, isolated sandbox to do Test Driven Development with your cookbooks.

In the same way, we can provide a sandbox environment for our infrastructure provisioning by using the kitchen-terraform driver.

SIDE BENEFIT: ONBOARDING!



{X}

Another GREAT side benefit of using Test Kitchen as a sandbox is onboarding new team members or users. This is a great way for them to get familiar with the module without breaking anything, and you'll see why the more we go through this.

KITCHEN-TERRAFORM DRIVER

Provides a set of Test Kitchen plugins which enable a system to use Test Kitchen to converge a Terraform configuration and verify the resulting Terraform state with InSpec controls

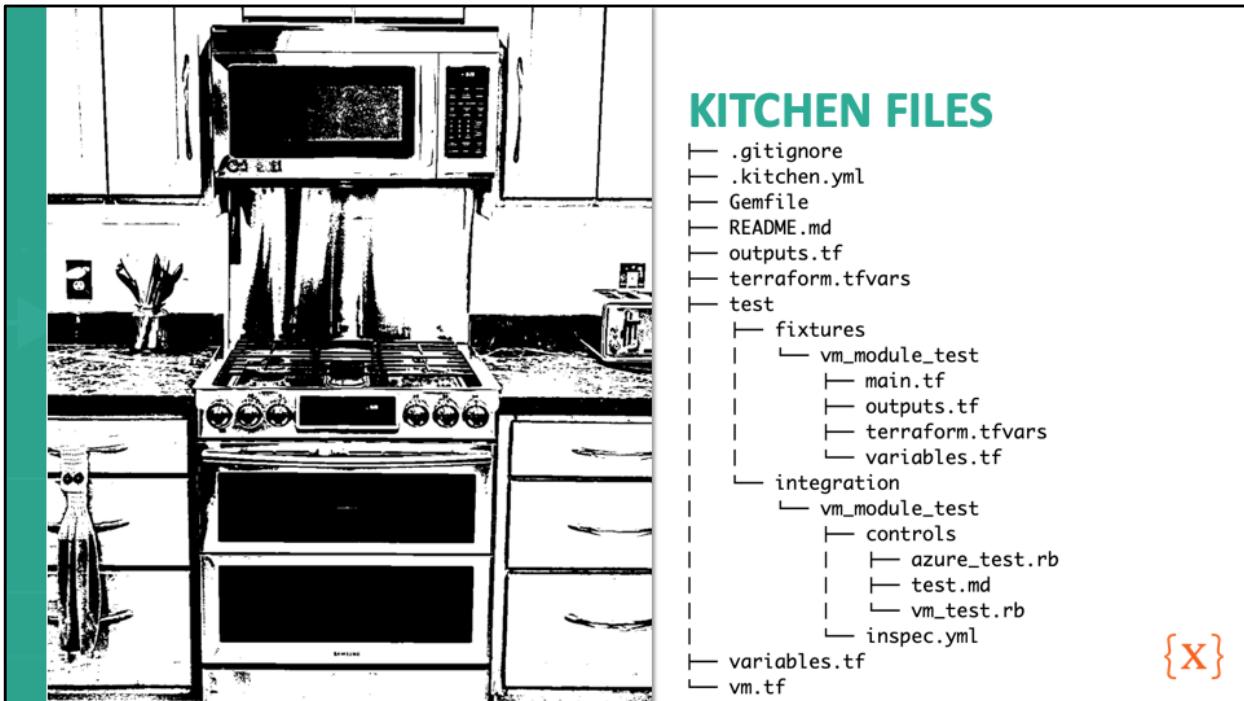
- Built with Ruby
- Use Bundler
- gem install kitchen-terraform

Shout-out to New Context and Aaron Lane for maintaining it!

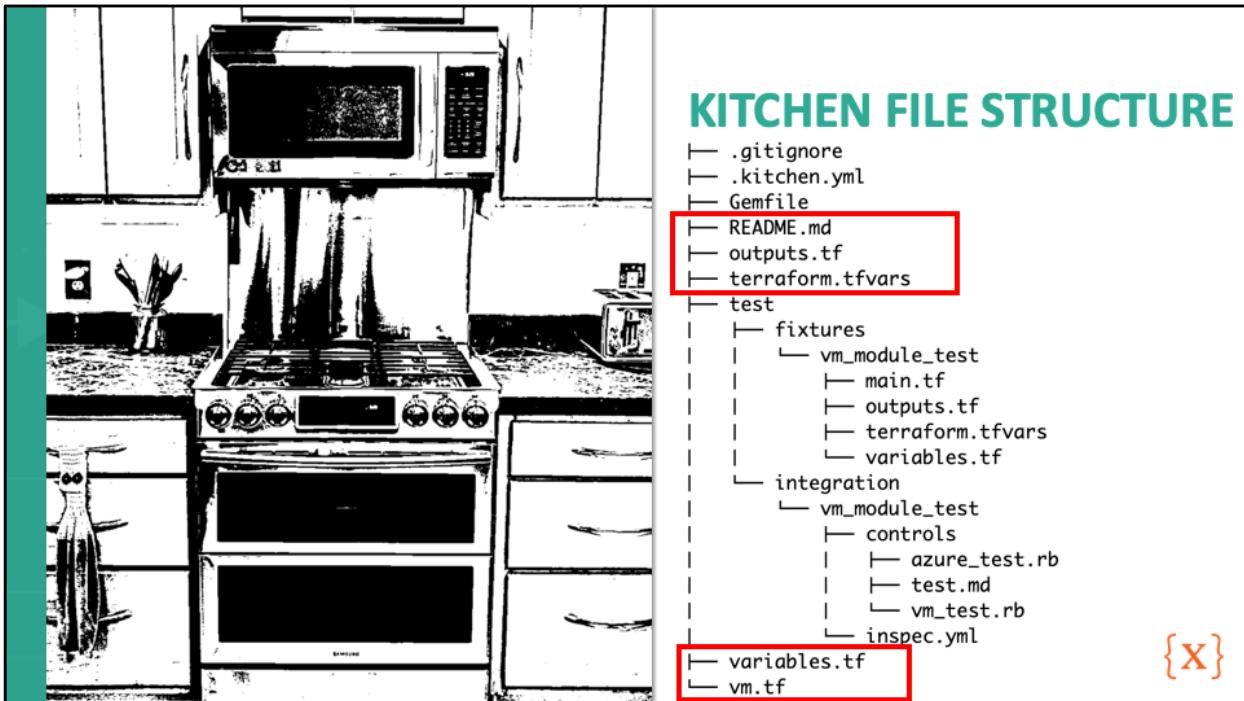
{X}

The “driver” in test kitchen is what will be used to create the resources, so for this, we want to be able to build with Terraform, so we’ll use the kitchen-terraform driver. It’s built with Ruby and uses Bundler, so you’ll need Ruby installed. So then you’d just install the kitchen-terraform gem to use it.

Again, if you’re familiar with Chef or Ruby and already have all of this installed, then this is ideal for you.

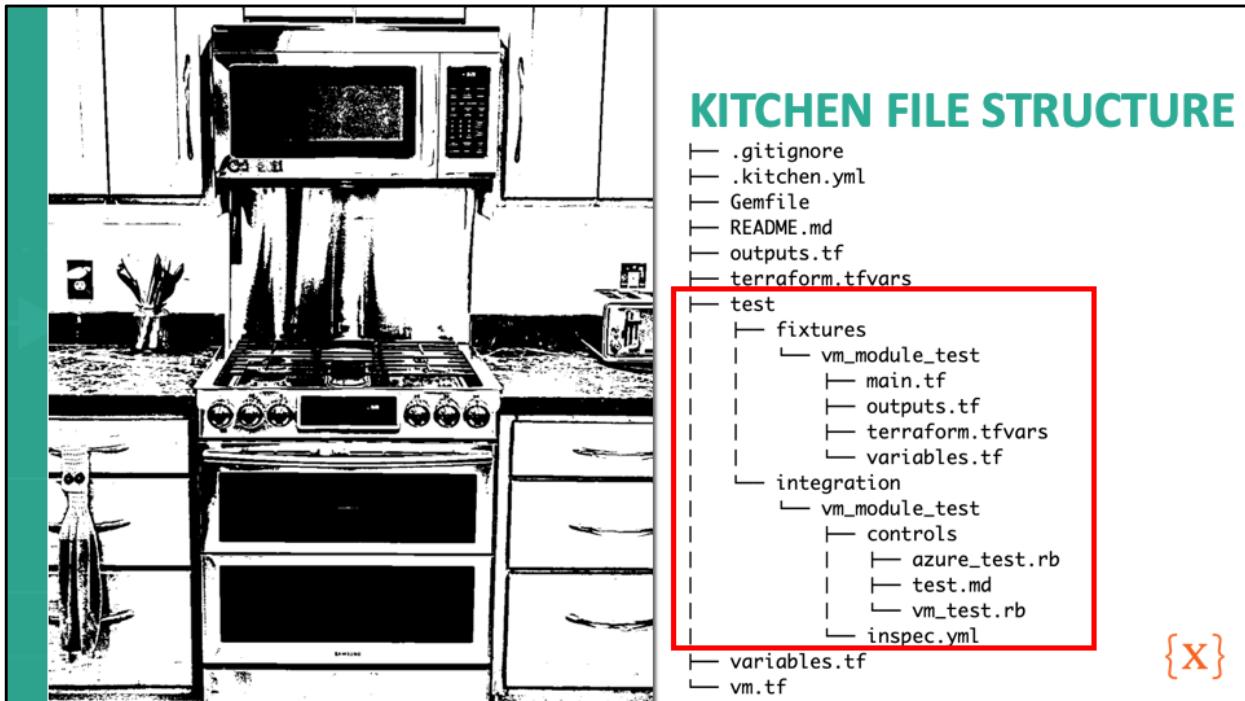


The way that kitchen works is that you basically nest a kitchen module inside the module that you're testing. And the isolation inside that nest insulates it enough to create a nice, little sandbox environment. [NEXT SLIDE]

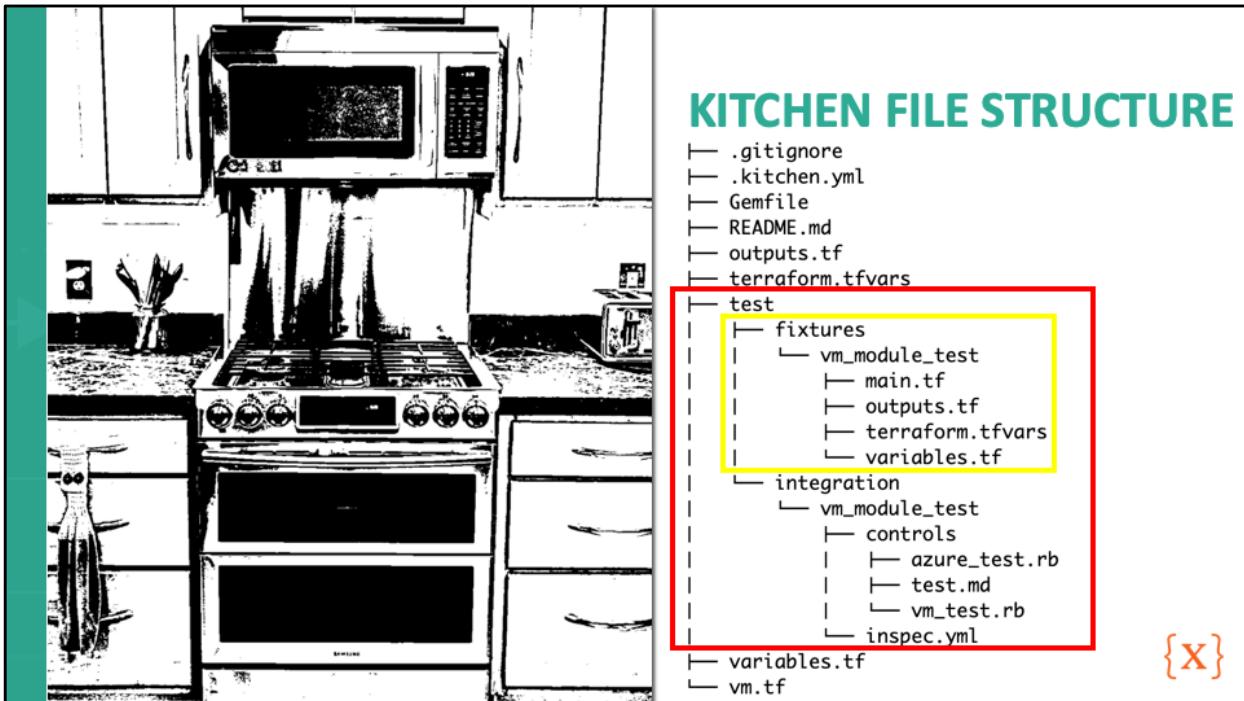


These files would be your main module, and of course this is a very bare bones module. It's very likely that you'd have more files, but basically, this illustrates that the module you're testing sits at the top of the tree. Then nested within this module,

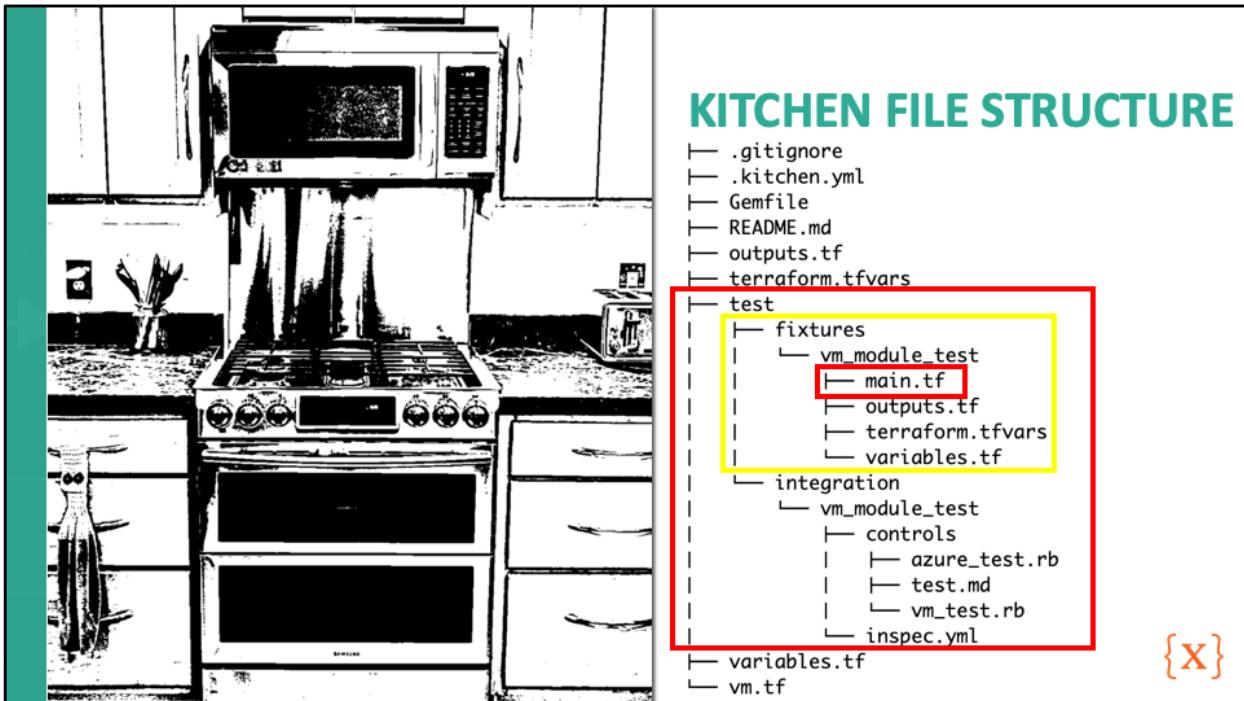
[NEXT SLIDE]



...you'll have your test directory that will have your [NEXT SLIDE]



...fixtures directory that houses your test kitchen module [NEXT SLIDE]



...and your test kitchen module will have all of its own variables and even different dependencies, like a different key vault or vnet, if it needs. Here is a look at this kitchen's main.tf [NEXT SLIDE]

TEST KITCHEN'S MAIN.TF

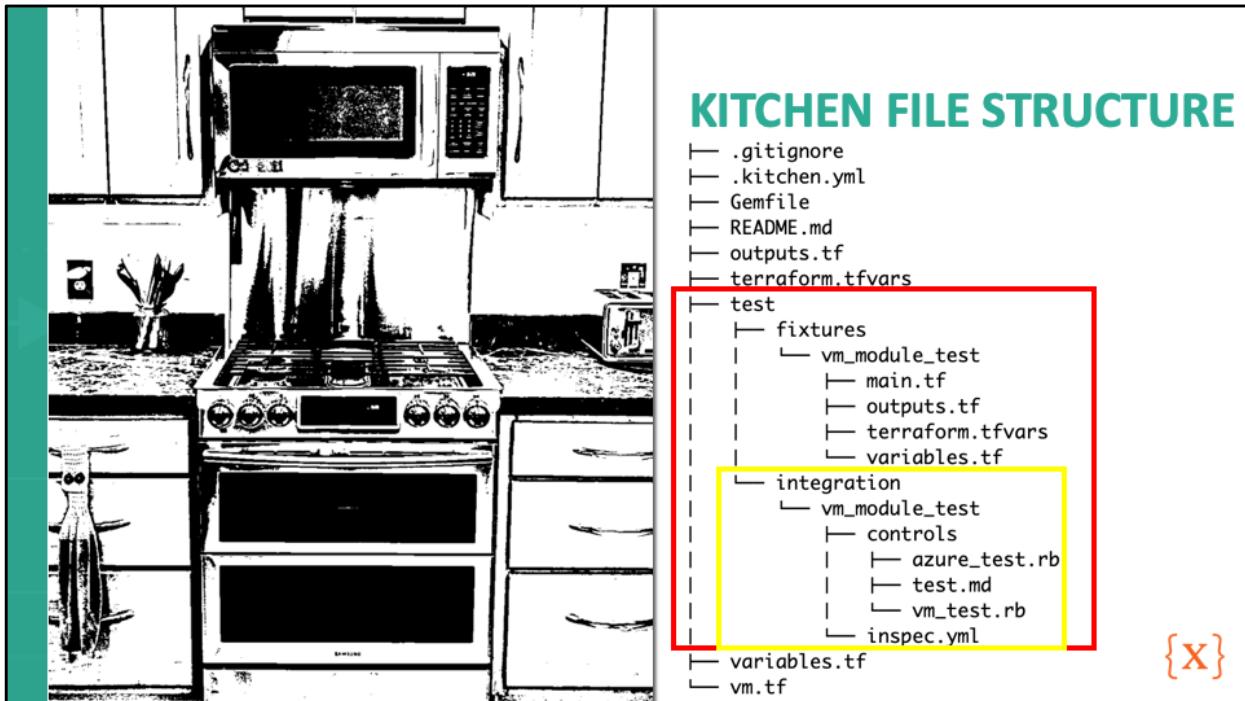
```
resource "random_integer" "prefix" {
  min = 1
  max = 500
}

module "test-kitchen-terraform" {
  source = "../../.."

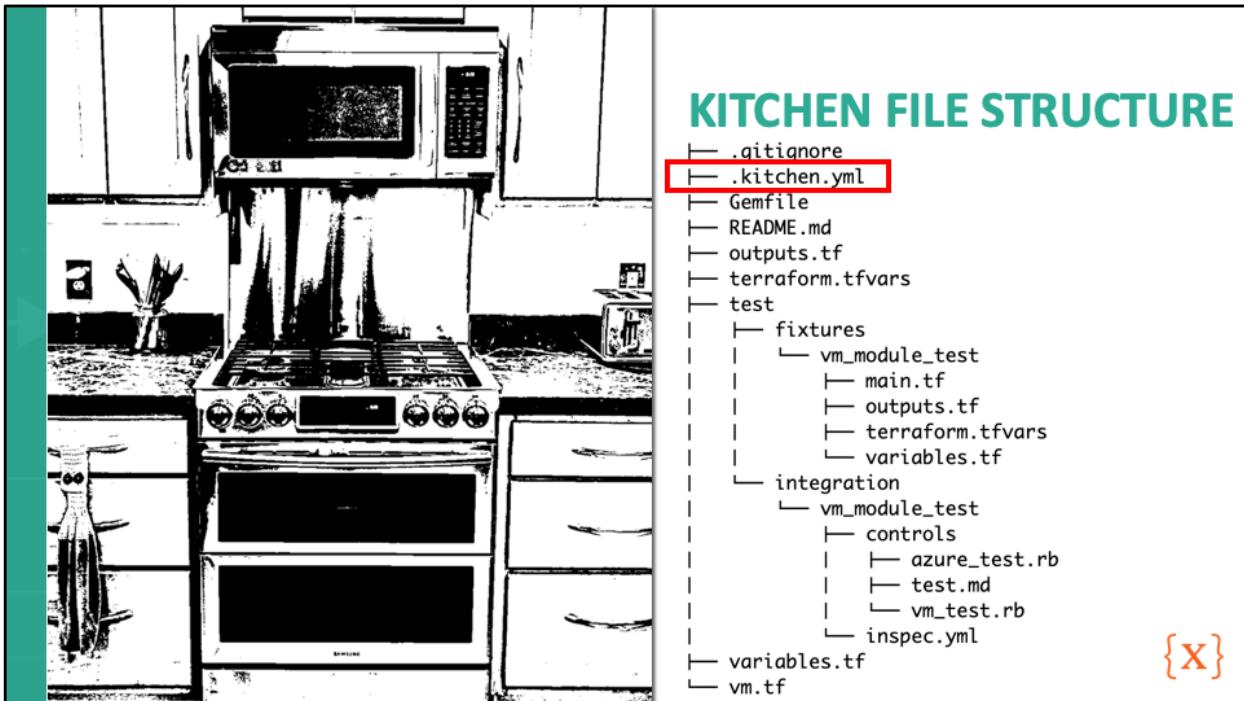
  resource_group  = "test-kitchen-rg-${random_integer.prefix.result}"
  subscription_id = var.subscription_id
  client_id       = var.client_id
  client_secret   = var.client_secret
  tenant_id        = var.tenant_id
  admin_username   = var.admin_username
  admin_password   = var.admin_password
  computer_name    = "tkcomputer"
}
```

{X}

...as you can see, it's just wrapping the root module, which is called "test-kitchen-terraform" and giving it its own variables.



...and below your fixtures directory is your integration folder that houses your InSpec profile. [NEXT SLIDE]



Important to note, though, is that the `kitchen.yml`, the configuration file for kitchen, sits at the top of the tree in the main module's root directory.

.KITCHEN.YML

```
driver:
  name: terraform
  root_module_directory: test/fixtures/vm_module_test
  command_timeout: 1200

provisioner:
  name: terraform

verifier:
  name: terraform
  systems:
    - name: vm
      backend: ssh
      port: 22
      hosts_output: vm_public_ip_address
      user: tkuser
      password: ████████
      controls:
        - vm config
    - name: azure
      backend: azure
      controls:
        - resources

platforms:
  - name: terraform

suites:
  - name: vm_module_test
```

{X}

Here's what your kitchen configuration file will look like. You can see here [NEW SLIDE]

.KITCHEN.YML

```
driver:
  name: terraform
  root_module_directory: test/fixtures/vm_module_test
  command_timeout: 1200

provisioner:
  name: terraform

verifier:
  name: terraform
  systems:
    - name: vm
      backend: ssh
      port: 22
      hosts_output: vm_public_ip_address
      user: tkuser
      password: [REDACTED]
      controls:
        - vm config
    - name: azure
      backend: azure
      controls:
        - resources

platforms:
  - name: terraform

suites:
  - name: vm_module_test
```

{X}

that the driver is “terraform”. The [Terraform driver](#) manages the state of the Terraform root module, and you can see here... [NEW SLIDE]

.KITCHEN.YML

```
driver:
  name: terraform
  root_module_directory: test/fixtures/vm_module_test
  command_timeout: 1200

provisioner:
  name: terraform

verifier:
  name: terraform
  systems:
    - name: vm
      backend: ssh
      port: 22
      hosts_output: vm_public_ip_address
      user: tkuser
      password: [REDACTED]
      controls:
        - vm config
    - name: azure
      backend: azure
      controls:
        - resources

platforms:
  - name: terraform

suites:
  - name: vm_module_test
```

{X}

...that the root module is directed to be the test kitchen that's nested inside the fixtures directory. [NEW SLIDE]

.KITCHEN.YML

```
driver:
  name: terraform
  root_module_directory: test/fixtures/vm_module_test
  command_timeout: 1200

provisioner:
  name: terraform

verifier:
  name: terraform
  systems:
    - name: vm
      backend: ssh
      port: 22
      hosts_output: vm_public_ip_address
      user: tkuser
      password: [REDACTED]
      controls:
        - vm config
    - name: azure
      backend: azure
      controls:
        - resources

platforms:
  - name: terraform

suites:
  - name: vm_module_test
```

{X}

The [Terraform provisioner](#) uses the Terraform driver to apply changes to the Terraform state. [NEW SLIDE]

.KITCHEN.YML

```
driver:
  name: terraform
  root_module_directory: test/fixtures/vm_module_test
  command_timeout: 1200

provisioner:
  name: terraform

verifier:
  name: terraform
  systems:
    - name: vm
      backend: ssh
      port: 22
      hosts_output: vm_public_ip_address
      user: tkuser
      password: ██████████
      controls:
        - vm config
    - name: azure
      backend: azure
      controls:
        - resources

platforms:
  - name: terraform

suites:
  - name: vm_module_test
```

{X}

And the [Terraform verifier](#) uses InSpec to verify the Terraform state. A Kitchen instance is verified by iterating through the systems... [NEW SLIDE]

.KITCHEN.YML

```
driver:
  name: terraform
  root_module_directory: test/fixtures/vm_module_test
  command_timeout: 1200

provisioner:
  name: terraform

verifier:
  name: terraform
  systems:
    - name: vm
      backend: ssh
      port: 22
      hosts_output: vm_public_ip_address
      user: tkuser
      password: (this can be ssh key)
      controls:
        - vm config
    - name: azure
      backend: azure
      controls:
        - resources

platforms:
  - name: terraform

suites:
  - name: vm_module_test
```

{X}

...and executing the associated InSpec controls against each target. So in each of these systems you can see that the backends are different and how to connect to those backends is different. For the first one, we're targeting the actual vm that we created in our Terraform module, and we're going to ssh into it in order to run the necessary tests.

For the second system, we're going to target the Azure subscription, and this uses environment variables to authenticate, so you don't see as much config here for that, just which controls to use.



TRICKS TO TESTING DIFFERENT TARGETS

```
|   └── integration
|     └── vm_module_test
|       ├── controls
|       |   └── azure_test.rb
|       |   └── vm_test.rb
|       └── inspec.yml
|
|   └── variables.tf
└── vm.tf
```

```
--- inspec.yml
name: example-test
title: Example Profile
version: 0.1.0
depends:
  - inspec-azure:
    git: https://github.com/inspec/inspec-azure.git
supports:
  - platform: azure
  - os-family: linux
```

{X}

So I told you that you can test two different systems (as kitchen terraform calls them) or targets (as InSpec calls them). But there is a little trick to getting that to work. You have to do two things [NEXT SLIDE]

Limitations of kitchen-terraform driver:

Can only use one InSpec profile

But you can inherit other profiles, just messy



TRICKS TO TESTING DIFFERENT TARGETS

```
|   └── integration
|     └── vm_module_test
|       ├── controls
|       |   ├── azure_test.rb
|       |   └── vm_test.rb
|       └── inspec.yml
|
|   └── variables.tf
└── vm.tf
```

```
--- inspec.yml
name: example-test
title: Example Profile
version: 0.1.0
depends:
  - inspec-azure:
    git: https://github.com/inspec/inspec-azure.git
supports:
  - platform: azure
  - os-family: linux
```

{X}

First, make sure that in your InSpec config, you're supporting all of the systems that you're testing.



TRICKS TO TESTING DIFFERENT TARGETS

```
|   └── integration
|       └── vm_module_test
|           ├── controls
|           |   ├── azure_test.rb
|           |   └── vm_test.rb
|           └── inspec.yml
|
|   └── variables.tf
└── vm.tf
```

```
--- inspec.yml
name: example-test
title: Example Profile
version: 0.1.0
depends:
  - inspec-azure:
    git: https://github.com/inspec/inspec-azure.git
supports:
  - platform: azure
  - os-family: linux
```

{X}

And then, here in the InSpec profile controls directory, you can see that we lumped both systems' tests into the same profile, but InSpec will fail if you have controls in an InSpec run that aren't for that target or system. So go to your kitchen config [NEXT SLIDE]

.KITCHEN.YML

```
driver:
  name: terraform
  root_module_directory: test/fixtures/vm_module_test
  command_timeout: 1200

provisioner:
  name: terraform

verifier:
  name: terraform
  systems:
    - name: vm
      backend: ssh
      port: 22
      hosts_output: vm_public_ip_address
      user: tkuser
      password: [REDACTED]
      controls:
        - vm_config
    - name: azure
      backend: azure
      controls:
        - resources

platforms:
  - name: terraform

suites:
  - name: vm_module_test
```

Further reading:
Profile inheritance

{X}

...and make sure that each system is ONLY running the controls for that particular system. So you can see here that the “vm_config” InSpec controls are being run on the VM target, and the “resources” InSpec controls are being run on the Azure target. If you left this blank and the verifier defaulted to running the whole profile, then it would fail since it can’t run those controls on those targets.

Another consideration that I don’t have time to explain here but that you can do some further reading on, is profile inheritance. This is a way in which you can run other profiles from this profile.



Now this is the FUN part! You can start running test kitchen and get your sandbox up and testing!

The first two commands that you'll use with kitchen terraform map to terraform commands.

1) You'll run "kitchen create" to do a "terraform init".

2) "kitchen converge" will setup a new workspace, validates the configuration files, and then runs "terraform apply".

Then to run your tests, you'll run "kitchen verify" to run the InSpec executable.

Then after all that, you don't want that environment to hang around, so you'd run "kitchen destroy" to destroy all of your kitchen resources. This will use the kitchen module's tfstate file to run "terraform destroy", which is that added layer of safety that you want.



FAKE DEMO

{X}

```

~/source/personal/azureddevops/test-kitchen-terraform > master •
$ bundle exec kitchen create
-> Starting Kitchen (v1.25.0)
--> Creating <vm-module-test-terra>
Terraform v0.12.19

Your version of Terraform is out of date! Your current version is 0.12.19, but the latest version is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
$$$$$ Running command 'terraform init -input=false -lock=false -no-color -parallelism=1 -reconfigure -upgrade -force-copy -backend=true -get=true -get-plugins=true -verify-plugins=true' in directory /Users/anniehedgpeth/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
Upgrading modules...
- inspec-azure-terraform-demo in .../...
Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "azurerm" (hashicorp/azurerm) 1.43.0...
- Downloading plugin for provider "random" (hashicorp/random) 2.2.1...

The following providers do not have any version constraints in configuration, so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking changes, it is recommended to add version = "..." constraints to the corresponding provider blocks in configuration, with the constraint strings suggested below.

* provider_azurerm: version = "> 1.43"
* provider_random: version = "> 2.2"

Terraform has been successfully initialized!
$$$$$ Running command 'terraform workspace select kitchen-terraform-vm-module-test-terraform' in directory /Users/anniehedgpeth/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
Workspace "kitchen-terraform-vm-module-test-terraform" doesn't exist.

You can create this workspace with the "new" subcommand.
$$$$$ Running command 'terraform workspace new kitchen-terraform-vm-module-test-terraform' in directory /Users/anniehedgpeth/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
Created and switched to workspace "kitchen-terraform-vm-module-test-terraform"

You're now on a new, empty workspace. Workspaces isolate their state, so if you run "terraform plan" Terraform will not see any existing state for this configuration.
Finished creating <vm-module-test-terraform> (0m5.28s).
----> Kitchen is finished. (0m7.42s)

```

The diagram shows a vertical sequence of four steps:

- kitchen create**: terraform init
- kitchen converge**: *terraform workspace select / get / validate
• terraform apply
- kitchen verify**: *runs inspec executable against targets
- kitchen destroy**: *deletes resources using tfstate file

This is what your “kitchen create” will look like, and you can see here that I’m running it inside Ruby’s “bundle exec” AND that I’m running it from the main module’s root directory, NOT the test directory (which is admittedly confusing since my root module is called “test-kitchen-terraform” - whoops). [NEXT SLIDE]

```

~/source/personal/azuredevops/test-kitchen-terraform > master •
$ bundle exec kitchen create
  -> Starting Kitchen (v1.25.0)
  -> Creating <vm-module-test-terraform>...
    Terraform v0.12.19

    Your version of Terraform is out of date! The latest version
    is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
    $$$$$$ Running command 'terraform init --input=false --lock=true --lock-timeout=0s --upgrade --force-copy --backend=true --get=true --get-plugins=true --verify-plugins=true' in directory /Users/anniehedgpeth/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
    Upgrading modules...
      - inspec-azure-terraform-demo in .../...
    Initializing the backend...

    Initializing provider plugins...
      - Checking for available provider plugins...
      - Downloading plugin for provider "azurerm" (hashicorp/azurerm) 1.43.0...
      - Downloading plugin for provider "random" (hashicorp/random) 2.2.1...

    The following providers do not have any version constraints in configuration,
    so the latest version was installed.

    To prevent automatic upgrades to new major versions that may contain breaking
    changes, it is recommended to add version = "..." constraints to the
    corresponding provider blocks in configuration, with the constraint strings
    suggested below.

    * provider_azurerm: version = "> 1.43"
    * provider_random: version = "> 2.2"

    Terraform has been successfully initialized!
    $$$$$$ Running command 'terraform workspace select kitchen-terraform-vm-module-test-terraform' in directory /Users/anniehedgpeth/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
    Workspace "kitchen-terraform-vm-module-test-terraform" doesn't exist.

    You can create this workspace with the "new" subcommand.
    $$$$$$ Running command 'terraform workspace new kitchen-terraform-vm-module-test-terraform' in directory /Users/anniehedgpeth/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
    Created and switched to workspace "kitchen-terraform-vm-module-test-terraform"

    You're now on a new, empty workspace. Workspaces isolate their state,
    so if you run "terraform plan" Terraform will not see any existing state
    for this configuration.
    Finished creating <vm-module-test-terraform> (0m5.28s).

    -----> Kitchen is finished. (0m7.42s)

```

The diagram on the right illustrates the four main commands of the Kitchen tool:

- kitchen create**: A teal chevron pointing down, followed by a white box containing:
 - terraform init
- kitchen converge**: A pink chevron pointing down, followed by a light pink box containing:
 - * terraform workspace select / get / validate
 - terraform apply
- kitchen verify**: An orange chevron pointing down, followed by an orange box containing:
 - * runs inspec executable against targets
- kitchen destroy**: A red chevron pointing down, followed by a light red box containing:
 - * deletes resources using tfstate file

A large orange bracket on the right side groups the four commands together, with a curly brace symbol {X} positioned below it.

And here you can see that “kitchen create” is just a wrapper around “terraform init” which will initialize a working directory containing Terraform configuration files inside our TEST directory. After that runs successfully, you’ll move on to [NEXT SLIDE]

```

~/source/personal/azureddevops/test
$ bundle exec kitchen converge
[...] Starting Kitchen (v1.25.0)
[...] Converging test-kitchen-test-to-form...
Terraform v0.12.19

Your version of Terraform is out of date! The latest version
is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
$$$$$$ Running command 'terraform workspace select kitchen-terraformation-module-test-terraform' in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
$$$$$$ Running command 'terraform get -update' in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
$ terraform init
In ./.terraformrc
$ terraform validate
[...]
Success! The configuration is valid.

$$$$$$ Running command 'terraform apply --lock=true --lock-timeout=6s --input=false --auto-approve=true --parallelism=10 --refresh=true' in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
random_integer.prefix: Creating...
random_integer.prefix: Creation complete after 0s [id=440]
module.inspec-azure-terraformation-module.azurerm_resource_group.my_rg: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraformation-module.azurerm_network_security_group.ng: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraformation-module.azurerm_virtual_network.my_vnet: Creating...
module.inspec-azure-terraformation-module.azurerm_ip_public_ip.publicip: Creation complete after 4s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/publicIPAddresses/publicip]
module.inspec-azure-terraformation-module.azurerm_network_security_group.ng: Still creating... [10s elapsed]
module.inspec-azure-terraformation-module.azurerm_virtual_network.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraformation-module.azurerm_network_interface.my_vnet: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkInterfaces/my_vnet]
module.inspec-azure-terraformation-module.azurerm_subnet.internal: Creating...
module.inspec-azure-terraformation-module.azurerm_network_security_group.ng: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/ng]
module.inspec-azure-terraformation-module.azurerm_network_security_rule.ng: Creating...
module.inspec-azure-terraformation-module.azurerm_subnet.internal: Still creating... [10s elapsed]
module.inspec-azure-terraformation-module.azurerm_network_security_rule.ng: Still creating... [10s elapsed]
module.inspec-azure-terraformation-module.azurerm_subnet.ng: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/ng/securityRules/inbound-tcp-ssh]
module.inspec-azure-terraformation-module.azurerm_subnet.ng: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terraformation-module.azurerm_subnet_network_security_association.ng_assoc: Creating...
module.inspec-azure-terraformation-module.azurerm_network_interface.my_vnet: Creating...
module.inspec-azure-terraformation-module.azurerm_subnet_network_security_group_association.ng_assoc: Still creating... [10s elapsed]
module.inspec-azure-terraformation-module.azurerm_network_interface.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraformation-module.azurerm_subnet_network_security_group_association.ng_assoc: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]

```

"kitchen converge" which you can see here will first run [NEXT SLIDE]

```

~/source/personal/azuredevops/test-kitchen-terraform 3301 19:57:53
$ bundle exec kitchen converge
--> Starting Kitchen (v1.25.0)
--> Converging `kitchen-test-terraforming`...
Terraform v0.12.19

Your version of Terraform is out of date! The latest version
is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
$***** Running command 'terraform workspace select kitchen-terraforming-vm-module-test-terraforming' in directory /Users/annihedgpath/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
$***** Running command 'terraform -update' in directory /Users/annihedgpath/source/personal/azuredevops/test-kitchen-terraforming-vm-module-test-terraforming-vm-module-test
$***** Running command 'terraform validate' in directory /Users/annihedgpath/source/personal/azuredevops/test-kitchen-terraforming/test/fixtures/vm_module_test
Success! The configuration is valid.

$***** Running command 'terraform apply --lock=true --lock-timeout=6s --input=false --auto-approve=true --parallelism=10 -refresh=true' in directory /Users/annihedgpath/source/personal/azuredevops/test-kitchen-terraforming/test/fixtures/vm_module_test
random_integer.prefix: Creating...
random_integer.prefix: Creation complete after 0s [id=440]
module.inspec-azure-terraforming-demo.azurerm_resource_group.my_rg: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraforming-demo.azurerm_network_security_group.my_nsg: Creation...
module.inspec-azure-terraforming-demo.azurerm_virtual_network.my_vnet: Creation...
module.inspec-azure-terraforming-demo.azurerm_virtual_ip_public_ip.my_ip: Creation complete after 4s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/publicIPAddresses/publicip]
module.inspec-azure-terraforming-demo.azurerm_network_security_group.nsg: Still creating... [10s elapsed]
module.inspec-azure-terraforming-demo.azurerm_virtual_network.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraforming-demo.azurerm_network_interface.my_vnet: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/vy_network]
module.inspec-azure-terraforming-demo.azurerm_subnet.internal: Creating...
module.inspec-azure-terraforming-demo.azurerm_network_security_group.nsg: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/nsg]
module.inspec-azure-terraforming-demo.azurerm_network_security_rule.my_nsg_rule: Creating...
module.inspec-azure-terraforming-demo.azurerm_subnet.internal: Still creating... [10s elapsed]
module.inspec-azure-terraforming-demo.azurerm_network_security_rule.my_nsg_rule: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/nsg/securityRules/inbound-tcp-ssh]
module.inspec-azure-terraforming-demo.azurerm_subnet.internal: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terraforming-demo.azurerm_subnet_network_security_group_association.my_nsg_assoc: Creating...
module.inspec-azure-terraforming-demo.azurerm_network_interface.my_vnet: Creating...
module.inspec-azure-terraforming-demo.azurerm_subnet_network_security_group_association.my_nsg_assoc: Still creating... [10s elapsed]
module.inspec-azure-terraforming-demo.azurerm_network_interface.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraforming-demo.azurerm_subnet_network_security_group_association.my_nsg_assoc: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]

```

The diagram on the right maps kitchen commands to Terraform actions:

- kitchen create**: terraform init
- kitchen converge**: terraform workspace select / get / validate
• terraform apply
- kitchen verify**: runs inspec executable against targets
- kitchen destroy**: deletes resources using tfstate file

““terrafom workspace select” which will select that new workspace that we just created in our test kitchen directory so that we can do stuff in it, [NEXT SLIDE]

```

~/source/personal/azureddevops/test-kitchen-terraform master • 3301 19:57:53
bundle exec kitchen converge
--> Starting Kitchen (v1.25.0)
--> Converging <test>/module-test-terraform...
Terraform v0.12.19

Your version of Terraform is out of date! The latest version
is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
$$$$$$ Running command 'terraform workspace select' in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/module-test
$$$$$$ Running command 'terraform get -update' in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/module-test
$$$$$$ Running command 'terraform validate' in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm.module_test
Success! The configuration is valid.

$$$$$$ Running command 'terraform apply --lock=true --lock-timeout=6s --input=false --auto-approve=true --parallelism=10 --refresh=true' in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm.module_test
random_integer.prefix: Creating...
random_integer.prefix: Creation complete after 0s [id=440]
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraform-demo.azurerm_network_security_group.my_nsg: Creating...
module.inspec-azure-terraform-demo.azurerm_network_security_group.my_nsg: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraform-demo.azurerm_virtual_network.my_vnet: Creating...
module.inspec-azure-terraform-demo.azurerm_virtual_network.my_vnet: Creation complete after 4s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/publicIPAddresses/publicip]
module.inspec-azure-terraform-demo.azurerm_network_security_group.my_nsg: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_virtual_network.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_network_interface.my_vnet: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/my_nsg]
module.inspec-azure-terraform-demo.azurerm_subnet.internal: Creating...
module.inspec-azure-terraform-demo.azurerm_subnet.internal: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/internal]
module.inspec-azure-terraform-demo.azurerm_network_security_rule.my_ssh: Creating...
module.inspec-azure-terraform-demo.azurerm_network_security_rule.my_ssh: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_network_security_rule.my_ssh: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/ng/securityRules/inbound-tcp-ssh]
module.inspec-azure-terraform-demo.azurerm_subnet.internal: Creating complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terraform-demo.azurerm_subnet_network_security_group_association.my_nsg_assoc: Creating...
module.inspec-azure-terraform-demo.azurerm_network_interface.my_vnet: Creating...
module.inspec-azure-terraform-demo.azurerm_subnet_network_security_group_association.my_nsg_assoc: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_network_interface.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_subnet_network_security_group_association.my_nsg_assoc: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]

```

The diagram illustrates the four steps of the kitchen workflow:

- kitchen create**: terraform init
- kitchen converge**: terraform workspace select / get / validate
terrafrom apply
- kitchen verify**: runs inspec executable against targets
- kitchen destroy**: deletes resources using tfstate file

Then it'll run “terraform get update” to download and update [modules](#) mentioned in the root module, which in this case is our test kitchen module.

```

~/source/personal/azuredevops/test-kitchen-terraform master • 3301 19:57:53
bundle exec kitchen converge
--> Starting Kitchen (v1.25.0)
--> Converging <vm-module-test>-terraform...
Terraform v0.12.19

Your version of Terraform is out of date! The latest version
is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
$$$$$$ Running command 'terraform workspace select kitchen-terraform-vm-module-test-terraform' in directory /Users/annehedgepath/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
$$$$$$ Running command 'terraform get -update' in directory /Users/annehedgepath/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
$$$$$$ Inspecting terraform...
$$$$$$ Running command 'terraform validate'
Success! The configuration is valid.

$$$$$$ Running command 'terraform apply --lock=true --lock-timeout=6s --input=false --auto-approve=true --parallelism=10' in directory /Users/annehedgepath/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
random_integer.prefix: Creating...
random_integer.prefix: Creation complete after 0s [id=440]
module.inspec-azure-terraformation-demo.azurerm_resource_group.my_rg: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraformation-demo.azurerm_network_security_group.my_nsg: Creation...
module.inspec-azure-terraformation-demo.azurerm_virtual_network.my_vnet: Creating...
module.inspec-azure-terraformation-demo.azurerm_public_ip.publicip: Creation complete after 4s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/publicIPAddresses/publicip]
module.inspec-azure-terraformation-demo.azurerm_network_security_group.nsg: Still creating... [10s elapsed]
module.inspec-azure-terraformation-demo.azurerm_virtual_network.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraformation-demo.azurerm_network_interface.my_vnet: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/nsg]
module.inspec-azure-terraformation-demo.azurerm_network_security_group.nsg: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/nsg]
module.inspec-azure-terraformation-demo.azurerm_subnet.internal: Creating...
module.inspec-azure-terraformation-demo.azurerm_subnet.internal: Still creating... [10s elapsed]
module.inspec-azure-terraformation-demo.azurerm_network_security_rule.ssh: Still creating... [10s elapsed]
module.inspec-azure-terraformation-demo.azurerm_network_security_rule.ssh: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/nsg/securityRules/inbound-tcp-ssh]
module.inspec-azure-terraformation-demo.azurerm_subnet.internal: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terraformation-demo.azurerm_subnet_network_security_group_association.nsg_assoc: Creating...
module.inspec-azure-terraformation-demo.azurerm_network_interface.my_vnet: Creating...
module.inspec-azure-terraformation-demo.azurerm_subnet_network_security_group_association.nsg_assoc: Still creating... [10s elapsed]
module.inspec-azure-terraformation-demo.azurerm_network_interface.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraformation-demo.azurerm_subnet_network_security_group_association.nsg_assoc: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]

```

kitchen create
• terraform init

 kitchen converge
• terraform workspace select / get / validate
• terraform apply

 kitchen verify
• runs inspec executable against targets

 kitchen destroy
• deletes resources using tfstate file

{X}

Then “terraform validate” to verify that the syntax is good and that the variable work, etc.

The diagram illustrates the four steps of the kitchen workflow:

- kitchen create**: terraform init
- kitchen converge**: terraform workspace select / get / validate
terrafrom apply
- kitchen verify**: runs inspec executable against targets
- kitchen destroy**: deletes resources using tfstate file

{X}

```

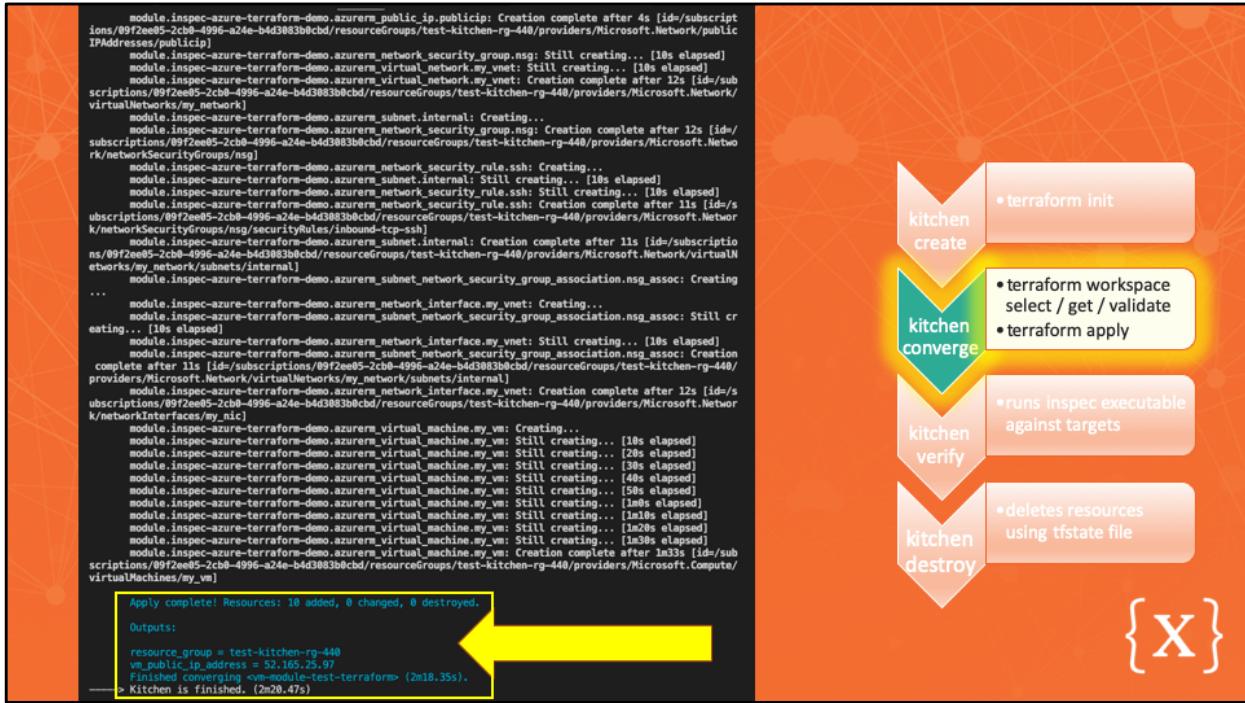
~/source/personal/azureddevops/test-kitchen-terraform master • 3301 19:57:53
bundle exec kitchen converge
--> Starting Kitchen (v1.25.0)
--> Converging <test-kitchen-test>-terraform...
Terraform v0.12.19

Your version of Terraform is out of date! The latest version
is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
$$$$$$ Running command `terraform workspace select kitchen-terraform-vm-module-test-terraform` in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
$$$$$$ Running command `terraform get -update` in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
$$$$$$ Inspecting Terraform In ./...
$$$$$$ Running command `terraform validate` in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
Success! The configuration is valid.

$$$$$$ Running command `terraform apply` to-approve=true -parallelism=10 -refresh=true in directory /Users/annihedgpath/source/personal/azureddevops/test-kitchen-terraform/test/fixtures/vm_module_test
random_integer.prefix: Creating...
random_integer.prefix: Creation complete after 0s [id=440]
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraform-demo.azurerm_network_security_group.ng: Creating...
module.inspec-azure-terraform-demo.azurerm_network_security_group.ng: Creation complete after 1s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraform-demo.azurerm_virtual_network.my_vnet: Creating...
module.inspec-azure-terraform-demo.azurerm_virtual_network.my_vnet: Creation complete after 4s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/publicIPAddresses/publicip]
module.inspec-azure-terraform-demo.azurerm_network_security_group.ng: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_virtual_network.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_network_security_group.ng: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/ng]
module.inspec-azure-terraform-demo.azurerm_network_security_group.ng: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/ng]
module.inspec-azure-terraform-demo.azurerm_network_security_rule.ng: Creating...
module.inspec-azure-terraform-demo.azurerm_network_security_rule.ng: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_network_security_rule.ng: Creation complete after 12s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/ng/securityRules/inbound-tcp-ssh]
module.inspec-azure-terraform-demo.azurerm_subnet.internal: Creating...
module.inspec-azure-terraform-demo.azurerm_subnet.internal: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_subnet.internal: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSubnets/internal]
module.inspec-azure-terraform-demo.azurerm_subnet_network_security_group_association.ng_assoc: Creating...
module.inspec-azure-terraform-demo.azurerm_subnet_network_interface.my_vnet: Creating...
module.inspec-azure-terraform-demo.azurerm_subnet_network_interface.my_vnet: Still creating... [10s elapsed]
module.inspec-azure-terraform-demo.azurerm_subnet_network_interface.my_vnet: Creation complete after 11s [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkInterfaces/my_vnet]

```

And finally it runs “terraform apply” from that same test kitchen directory, and it starts provisioning a test kitchen environment of our root module.



And if we scroll down, we can see here that it finished and gave us our terraform output.

```

~$ source/venv/bin/activate
$ bundle exec kitchen verify
--> Starting Kitchen (v1.25.0)
----> Finished setting up <vm-module-test>-<terraform>...
----> Verifying <vm-module-test>-<terraform>...
$$$$$$ Running command `terraform workspace select kitchen-terraform-vm-module-test` in directory /Users/anniehedgpeth/source/personal/azuredops/test-kitchen-terraform/test/fixtures/vm_module_test
$$$$$$ Running command `terraform output -json` in directory /Users/anniehedgpeth/source/personal/azuredops/test-kitchen-terraform/test/fixtures/vm_module_test
vm: Verifying host 52.165.25.97
Skipping profile: 'inspec-azure' on unsupported platform: 'ubuntu/16.04'.

Profile: Demo VM Profile (demo-vm-test)
Version: 0.1.0
Target: ssh://tkuser@52.165.25.97:22

  ✓ vm config: User tkuser
    ✓ User 'tkuser' home is expected to eq "/home/tkuser"
    ✓ Directory '/home/tkuser' is expected to exist
    ✓ Directory '/home/tkuser' owner is expected to eq "tkuser"

Profile Summary: 1 successful control, 0 control failures, 0 controls skipped
Test Summary: 3 successful, 0 failures, 0 skipped
azure: Verifying

Profile: Demo VM Profile (demo-vm-test)
Version: 0.1.0
Target: azure://09f2ee05-2cb0-4996-a24e-b4d3083b0cbd

  ✓ resources: Azure resources
    ✓ 'my_vm' Virtual Machine is expected to exist
    ✓ 'my_vm' Virtual Machine type is expected to eq "Microsoft.Compute/virtualMachines"
    ✓ 'nsgr' Network Security Group is expected to exist
    ✓ 'nsgr' Network Security Group is expected to allow ssh from internet
    ✓ 'nsgr' Network Security Group type is expected to eq "Microsoft.Network/networkSecurityGroups"
    ✓ 'nsgr' Network Security Group security_rules is expected not to be empty
    ✓ 'nsgr' Network Security Group default_security_rules is expected not to be empty
    ✓ 'my_network' Virtual Network is expected to exist
    ✓ 'my_network' Virtual Network location is expected to eq "centralus"

Profile: Azure Resource Pack (inspec-azure)
Version: 1.12.0
Target: azure://09f2ee05-2cb0-4996-a24e-b4d3083b0cbd

  No tests executed.

Profile Summary: 1 successful control, 0 control failures, 0 controls skipped
Test Summary: 0 successful, 0 failures, 0 skipped
Finished verifying <vm-module-test>-<terraform> (0m5.28s).
--> Kitchen is finished. (0m7.34s)

```

The diagram illustrates the four main steps of the Kitchen workflow:

- kitchen create**: Associated with `terraform init`.
- kitchen converge**: Associated with `terraform workspace select / get / validate` and `terraform apply`.
- kitchen verify**: Associated with running InSpec executable against targets.
- kitchen destroy**: Associated with deleting resources using `tfstate` file.

And now for my FAVORITE part, actually testing our module. Here I'm running "kitchen verify", and remember that it's going to run InSpec against the two different targets that we gave it in the kitchen.yml.

```

~/source/personal/azuredevops/test-kitchen-terraform > master 3302 20:09:33
$ bundle exec kitchen verify
--> Starting Kitchen (v1.25.0)
--> Setting up <vm-module-test-terraform>...
--> Verifying <vm-module-test-terraform>...
$fffff Running command `terraform workspace select kitchen-terraform-vm-module-test` in directory /Users/anniehedgpeth/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
$fffff Running command `terraform output -json` in directory /Users/anniehedgpeth/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
vm: Verifying host 52.165.25.97
Skipping profile: 'inspec-azure' on unsupported platform: 'ubuntu/16.04'.
Profile: Demo VM Profile (demo-vm-test)
Version: 0.1.0
Target: ssh://tkuser@52.165.25.97:22
  ✓ vm config: User tkuser
    ✓ User tkuser home is expected to eq "/home/tkuser"
    ✓ Directory /home/tkuser is expected to exist
    ✓ Directory /home/tkuser owner is expected to eq "tkuser"

Profile Summary: 1 successful control, 0 control failures, 0 controls skipped
Test Summary: 3 successful, 0 failures, 0 skipped
azure: Verifying

Profile: Demo VM Profile (demo-vm-test)
Version: 0.1.0
Target: azure://09f2ee05-2cb0-4996-a24e-b4d3083b0cbd
  ✓ resources: Azure resources
    ✓ 'my_vm' Virtual Machine is expected to exist
    ✓ 'my_vm' Virtual Machine type is expected to eq "Microsoft.Compute/virtualMachines"
    ✓ 'nsgr' Network Security Group is expected to exist
    ✓ 'nsgr' Network Security Group is expected to allow ssh from internet
    ✓ 'nsgr' Network Security Group type is expected to eq "Microsoft.Network/networkSecurityGroups"
    ✓ 'nsgr' Network Security Group security_rules is expected not to be empty
    ✓ 'nsgr' Network Security Group default_security_rules is expected not to be empty
    ✓ 'my_network' Virtual Network is expected to exist
    ✓ 'my_network' Virtual Network location is expected to eq "centralus"

Profile: Azure Resource Pack (inspec-azure)
Version: 1.12.0
Target: azure://09f2ee05-2cb0-4996-a24e-b4d3083b0cbd
  No tests executed.

Profile Summary: 1 successful control, 0 control failures, 0 controls skipped
Test Summary: 0 successful, 0 failures, 0 skipped
Finished verifying <vm-module-test-terraform> (0m5.28s).
--> Kitchen is finished. (0m7.34s)

```

The diagram on the right illustrates the four main steps of the Kitchen workflow:

- kitchen create**: • terraform init
- kitchen converge**: • terraform workspace select / get / validate
• terraform apply
- kitchen verify**: • runs inspec executable against targets
- kitchen destroy**: • deletes resources using tfstate file

A large orange bracket on the right groups the first three steps under the heading {X}.

FIRST, it's going to run all of the `vm_config` InSpec tests against the VM target. So now we have validated that the user called "tkuser" exists and owns the "/home/tkuser" directory. This is super bare bones, obviously, but you can use this to validate any configuration you did to the node, whether it be post-provisioning, maybe in a cloud init script, in Chef or Ansible, or on the image that that vm was created from.

```

~/source/personal/azuredevops/test-kitchen-terraform > master 3302 20:09:33
$ bundle exec kitchen verify
--> Starting Kitchen (v1.25.0)
--> Setting up <vm-module-test-terraform>...
----- FINISHED setting up <vm-module-test-terraform> (0m0.00s).
----- Running <vm-module-test-terraform>...
----- Running command `terraform workspace select kitchen-terraform-vm-module-test` in directory /Users/anniehedgpeth/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
----- Running command `terraform output -json` in directory /Users/anniehedgpeth/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
vm: Verifying host 52.165.25.97
Skipping profile: 'inspec-azure' on unsupported platform: 'ubuntu/16.04'.

Profile: Demo VM Profile (demo-vm-test)
Version: 0.1.0
Target: ssh://tkuser@52.165.25.97:22

  ✓ vm config: User tkuser
    ✓ User tkuser home is expected to eq "/home/tkuser"
    ✓ Directory '/home/tkuser' is expected to exist
    ✓ Directory '/home/tkuser' owner is expected to eq "tkuser"

Profile Summary: 1 successful control, 0 control failures, 0 controls skipped
Test Summary: 3 successful, 0 failures, 0 skipped
azure: Verifying

Profile: Demo VM Profile (demo-vm-test)
Version: 0.1.0
Target: azure://09f2ee05-2cb0-4996-a24e-b4d3083b0cbd

  ✓ resources: Azure resources
    ✓ 'my_vm' Virtual Machine is expected to exist
    ✓ 'my_vm' Virtual Machine type is expected to eq "Microsoft.Compute/virtualMachines"
    ✓ 'nsrg' Network Security Group is expected to exist
    ✓ 'nsrg' Network Security Group is expected to allow ssh from internet
    ✓ 'nsrg' Network Security Group type is expected to eq "Microsoft.Network/networkSecurityGroups"
    ✓ 'nsrg' Network Security Group security_rules is expected not to be empty
    ✓ 'nsrg' Network Security Group default_security_rules is expected not to be empty
    ✓ 'my_network' Virtual Network is expected to exist
    ✓ 'my_network' Virtual Network location is expected to eq "centralus"

Profile: Azure Resource Pack (inspec-azure)
Version: 1.12.0
Target: azure://09f2ee05-2cb0-4996-a24e-b4d3083b0cbd

  No tests executed.

Profile Summary: 1 successful control, 0 control failures, 0 controls skipped
Test Summary: 0 successful, 0 failures, 0 skipped
----- Finished verifying <vm-module-test-terraform> (0m5.20s).

--> Kitchen is finished. (0m7.34s)

```

The diagram illustrates the four stages of the kitchen workflow:

- kitchen create**: terraform init
- kitchen converge**: terraform workspace select / get / validate, terraform apply
- kitchen verify**: runs inspec executable against targets
- kitchen destroy**: deletes resources using tfstate file

And next we target the Azure environment itself and validate that all of the resources that we created exist. Here we can see that the VM exists, and that there are network security groups and rules, that the virtual network that we created exists and that it's in the region we wanted it to be in. This can also be way more complex based on your environment. And like I said earlier, there are a myriad of different InSpec resources to be used for all of the major cloud providers, linux, windows, and docker. Really, anything you can run a command to check, then InSpec can likely check for you.

And now that we have a successful validation, we can destroy our kitchen resources!

```

~/source/personal/azurededevops/test
$ bundle exec kitchen destroy
[1]: Starting Kitchen (V1.25.0)
-> Starting test-kitchen-test
Terraform v0.12.19

Your version of Terraform is out of date! The latest version
is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
$$$$$ Running command terraform init -input=false -lock=true -lock-timeout=0s -force-copy -backend=true -get=true -get-plugins=true -verify-plugins=true in directory /Users/anniehedgpeth/source/personal/azurededevops/test-kitchen-terraform/test/fixtures/vm_module_test
Initializing modules...
Initializing the backend...
Initializing provider plugins...
The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "...", constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.azurevm: version = "> 1.43"
* provider.random: version = "> 2.2"

Terraform has been successfully initialized!
$$$$$ Running command terraform workspace select kitchen-terraform-vm-module-test-terraform in directory /Users/anniehedgpeth/source/personal/azurededevops/test-kitchen-terraform/test/fixtures/vm_module_test
$$$$$ Running command terraform destroy -auto-approve -lock=true -lock-timeout=0s -input=false -parallelism=10 -refresh=true - in directory /Users/anniehedgpeth/source/personal/azurededevops/test-kitchen-terraform/test/fixtures/vm_module_test
random_integer.prefix: Refreshing state... [id=440]
module.inspec-azure-terrafrom-demo.azurerm_resource_group.my_rg: Refreshing state... [id=/subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terrafrom-demo.azurerm_virtual_network.my_vnet: Refreshing state... [id=/subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetworks/my_network]
module.inspec-azure-terrafrom-demo.azurerm_network_security_group.ng: Refreshing state... [id=/subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/ng]
module.inspec-azure-terrafrom-demo.azurerm_ip_publicip: Refreshing state... [id=/subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/publicIPAddresses/publicip]
module.inspec-azure-terrafrom-demo.azurerm_subnet_internal: Refreshing state... [id=/subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terrafrom-demo.azurerm_network_security_rule.ssh: Refreshing state... [id=/subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkSecurityGroups/my_nsg]
module.inspec-azure-terrafrom-demo.azurerm_subnet_network_security_group_association.ng_assoc: Refreshing state... [id=/subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terrafrom-demo.azurerm_network_interface.my_vnet: Refreshing state... [id=/subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkInterfaces/my_nic]

```

So after you're finished, and you've run converge and verify as many times as you needed to, you can destroy your test kitchen resources with "kitchen destroy". [NEXT SLIDE]

The diagram illustrates the Terraform Kitchen workflow, which consists of five main steps:

- kitchen create**: • terraform init
- kitchen converge**: • terraform workspace select / get / validate
• terraform apply
- kitchen verify**: • runs inspec executable against targets
- kitchen destroy**: • deletes resources using tfstate file

A yellow arrow points from the terminal output on the left to the "kitchen converge" step in the diagram.

```

~/source/personal/azuredevops/test-kitchen-terraform > master > 3303 20:48:29
$ bundle exec kitchen destroy
--> Starting Kitchen (v1.25.0)
--> Destroying test-kitchen-test-terraform...
Terraform v0.12.19

Your version of Terraform is out of date! The latest version
is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
$$$$$ Running command: terraform init -input=false -lock=true -lock-timeout=0s -force-copy -backend=true -ge
ttrue -get-plugins=true -verify-plugins=true in directory /Users/anniehedgpeth/source/personal/azuredevops/test
-st-kitchen-terraform/test/fixtures/vm_module_test

Initializing modules...

Initializing the backend...

Initializing provider plugins...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

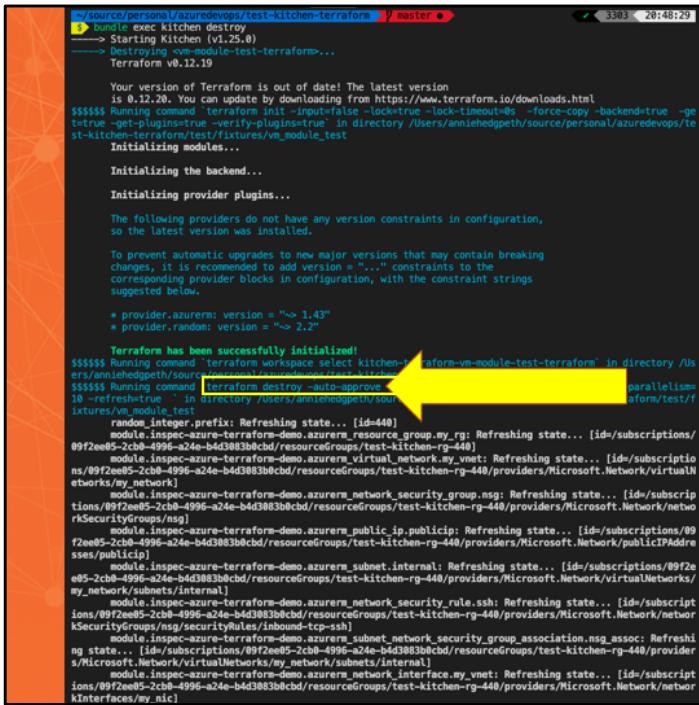
To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "...," constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.azurevm: version = "> 1.43"
* provider.random: version = "> 2.12"

Terraform has been successfully initialized!
$$$$$$ Running command: terraform workspace select -name test-kitchen-test-terraform
$$$$$$ Running command: terraform destroy -auto-approve -lock=true -lock-timeout=0s -input=false -parallelism
10 -refresh=true - in directory /Users/anniehedgpeth/source/personal/azuredevops/test-kitchen-terra
/fixtures/vm_module_test
random_integer.prefix: Refreshing state... [id=440]
module.inspec-azure-azurerm_resource_group.my_rg: Refreshing state... [id=subscriptions/
09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-azurerm_virtual_network.my_vnet: Refreshing state... [id=subscriptio
ns/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualN
etworks/my_network]
module.inspec-azure-azurerm_network_security_group.ng: Refreshing state... [id/subscript
ions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/netwo
rkSecurityGroups/ng]
module.inspec-azure-terraform-demo_azurerm_public_ip.publicip: Refreshing state... [id/subscriptions/09
f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/publicIPAddre
ses/publicip]
module.inspec-azure-terraform-demo_azurerm_subnet_internal: Refreshing state... [id/subscriptions/09f2e
e05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetw
orks/my_network/subnets/internal]
module.inspec-azure-terraform-demo_azurerm_network_security_rule.ssh: Refreshing state... [id/subscript
ions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkS
ecurityRules/ssh]
module.inspec-azure-terraform-demo_azurerm_subnet_network_security_group_association.ng_assoc: Refreshi
ng state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/provider
s/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terraform-demo_azurerm_network_interface.my_vnet: Refreshing state... [id/subscript
ions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/netw
orkInterfaces/my_nic]

```

This will again select your test workspace, keeping it isolated and sandboxed, and then [NEXT SLIDE]



The terminal output shows the execution of the 'kitchen destroy' command:

```

~/source/personal/azuredevops/test-kitchen-terraform > / master • 3303 20:48:29
bundle exec kitchen destroy
--> Starting Kitchen (v1.25.0)
--> Destroying workspace-test-terraform...
Terraform v0.12.19

Your version of Terraform is out of date! The latest version
is 0.12.20. You can update by downloading from https://www.terraform.io/downloads.html
$$$$$ Running command terraform init -input=false -lock=true -lock-timeout=0s -force-copy -backend=true -get=true -get-plugins=true -verify-plugins=true in directory /Users/anniehedghep/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
Initialising modules...

Initializing the backend...
Initializing provider plugins...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "...", constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.azurevm: version = "> 1.43"
* provider.random: version = "> 2.2"

Terraform has been successfully initialized!
$$$$$ Running command terraform workspace select kitchen-terraform-vm-module-test-terraform in directory /Users/anniehedghep/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
$$$$$ Running command terraform destroy -auto-approve
10 -refresh=true in directory /Users/anniehedghep/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
random_integer.prefix: Refreshing state... [id=448]
module.inspec-azure-azurerm_resource_group.my_rg: Refreshing state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-448]
module.inspec-azure-azurerm_virtual_network.my_vnet: Refreshing state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-448/providers/Microsoft.Network/virtualNetworks/my_network]
module.inspec-azure-terraform-demo_azurerm_network_security_group.ng: Refreshing state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-448/providers/Microsoft.Network/networkSecurityGroups/ng]
module.inspec-azure-terraform-demo_azurerm_public_ip.publicip: Refreshing state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-448/providers/Microsoft.Network/publicIPAddresses/publicip]
module.inspec-azure-terraform-demo_azurerm_subnet_internal: Refreshing state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-448/providers/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terraform-demo_azurerm_subnet_my_subnet: Refreshing state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-448/providers/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terraform-demo_azurerm_network_security_rule.ssh: Refreshing state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-448/providers/Microsoft.Network/networkSecurityRules/ssh]
module.inspec-azure-terraform-demo_azurerm_subnet_network_security_group_association.ng_assoc: Refreshing state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-448/providers/Microsoft.Network/virtualNetworks/my_network/subnets/internal]
module.inspec-azure-terraform-demo_azurerm_network_interface.my_vnet: Refreshing state... [id=subscriptions/09f2ee05-2cb8-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-448/providers/Microsoft.Network/networkInterfaces/my_nic]

```



{X}

...run “terraform destroy” from the safety of your sandbox. When all is done..[NEXT SLIDE]

```

module.inspec-azure-terraform-demo.azurerm_virtual_machine.my_vm: Still destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-...icrosoft.Compute/virtualMachines/my_vm, 5m30s elapsed]
module.inspec-azure-terraform-demo.azurerm_virtual_machine.my_vm: Destruction complete after 5m32s
module.inspec-azure-terraform-demo.azurerm_network_interface.my_vnet: Destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/networkInterfaces/my_nic]
module.inspec-azure-terraform-demo.azurerm_network_interface.my_vnet: Destruction complete after 1s
module.inspec-azure-terraform-demo.azurerm_public_ip.publicip: Destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/publicIPAddresses/publicip]
module.inspec-azure-terraform-demo.azurerm_subnet.internal: Destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetworks/my_netwokr/subnets/internal]
module.inspec-azure-terraform-demo.azurerm_subnet.internal: Destruction complete after 0s
module.inspec-azure-terraform-demo.azurerm_virtual_network.my_vnet: Destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440/providers/Microsoft.Network/virtualNetworks/my_network]
module.inspec-azure-terraform-demo.azurerm_public_ip.publicip: Still destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-...oft.Network/publicIPAddresses/publicip, 10s elapsed]
module.inspec-azure-terraform-demo.azurerm_virtual_network.my_vnet: Still destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-...oft.Network/virtualNetworks/my_network, 10s elapsed]
module.inspec-azure-terraform-demo.azurerm_public_ip.publicip: Destruction complete after 10s
module.inspec-azure-terraform-demo.azurerm_virtual_network.my_vnet: Destruction complete after 11s
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-b4d3083b0cbd/resourceGroups/test-kitchen-rg-440]
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Still destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-...cbd/resourceGroups/test-kitchen-rg-440, 10s elapsed]
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Still destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-...cbd/resourceGroups/test-kitchen-rg-440, 20s elapsed]
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Still destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-...cbd/resourceGroups/test-kitchen-rg-440, 30s elapsed]
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Still destroying... [id=/subscriptions/09f2ee05-2cb0-4996-a24e-...cbd/resourceGroups/test-kitchen-rg-440, 40s elapsed]
module.inspec-azure-terraform-demo.azurerm_resource_group.my_rg: Destruction complete after 47s
random_integer.prefix: Destroying... [id=440]
random_integer.prefix: Destruction complete after 0s
Destroy complete! Resources: 10 destroyed.

$$$$$ Running command terraform workspace select default
Switched to workspace "default".
$$$$$ Running command terraform workspace delete kitchen-terraform-vm-module-test-terraform' in directory /Users/anniedhedgpeth/source/personal/azuredevops/test-kitchen-terraform/test/fixtures/vm_module_test
Deleted workspace "kitchen-terraform-vm-module-test-terraform".
Finished destroying <vm-module-test-terraform> (6m40.68s).
--> Kitchen is finished. (6m42.86s)

```

...you don't have to worry about keeping any of these environments around.

SECURITY & SANITY

USE-CASES FOR TESTING TERRAFORM WITH INSPEC

- Production provisioning validation
- Test Driven Development (TDD)
- Integration Testing and CI/CD
- Compliance, shifting security left

{X}

So now that you know a bit about InSpec, we can go back to this slide and match the use-cases with the method for running InSpec.

SECURITY & SANITY

USE-CASES FOR TESTING TERRAFORM WITH INSPEC

- Production provisioning validation
- Test Driven Development (TDD)
- Integration Testing and CI/CD
- Compliance, shifting security left

```
provisioner  
"local-exec" {  
  command =  
    "bundle exec  
    inspec exec
```

{X}

If you want to validate and run compliance audits on a deployment, then you can run an InSpec profile at the end of your module with a local-exec provisioner.

SECURITY & SANITY

USE-CASES FOR TESTING TERRAFORM WITH INSPEC

- Production provisioning validation
- Test Driven Development (TDD)
- Integration Testing and CI/CD
- Compliance, shifting security left

```
provisioner  
"local-exec" {  
  command =  
    "bundle exec  
    inspec exec
```

{X}

If you want to use test driven development practices in the development of your terraform modules, then use those tests in a pull request build, then Test Kitchen is your friend.

SECURITY & SANITY

USE-CASES FOR TESTING TERRAFORM WITH INSPEC

- Production provisioning validation
- Test Driven Development (TDD)
- Integration Testing and CI/CD
- Compliance, shifting security left

```
provisioner  
"local-exec" {  
  command =  
    "bundle exec  
    inspec exec
```

{X}

You can even add those compliance profiles that I keep talking about in those test kitchen tests and pull request builds to make sure nothing gets merged that would push you out of compliance.

RESOURCES

<https://github.com/anniehedgpeth/inpec-azure-terraform-demo>

<http://www.anniehedgie.com/kitchen-terraform-and-inpec>

<https://github.com/newcontext-oss/kitchen-terraform>

{X}

I hope this helps to make your terraforming more sane and enjoyable. Here are some additional resources. This demo repo is in my github, and I also blogged about it there at that second link, so feel free to take a look at those. And of course, you can familiarize yourself with the kitchen terraform plugin there on their github.



10TH MAGNITUDE

Annie Hedgpeth
Senior Cloud Automation Engineer
@anniehedgie
www.anniehedgie.com



Gold DevOps
Gold Cloud Platform
Gold Application Development
Gold Datacenter
Gold Data Analytics



Azure
Expert
MSP

It's been a pleasure sharing this with you today. Feel free to reach out with any questions, and enjoy the rest of the amazing HashiTalks coming at you!