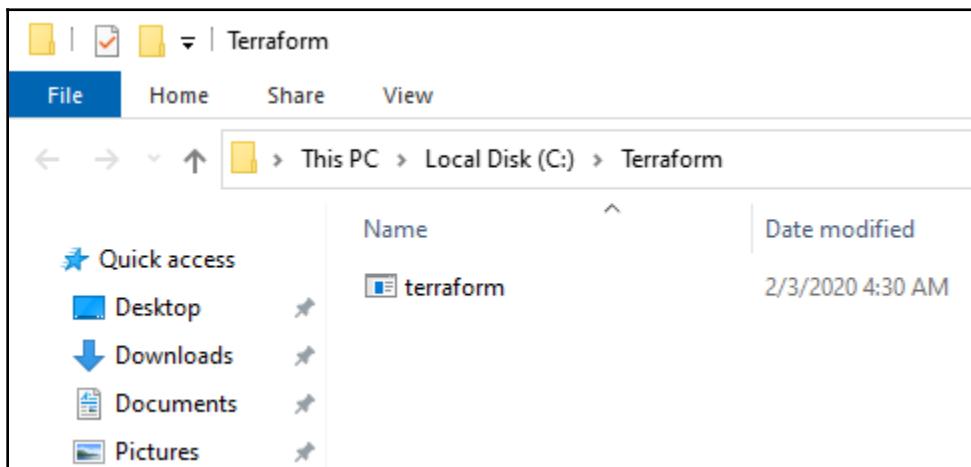
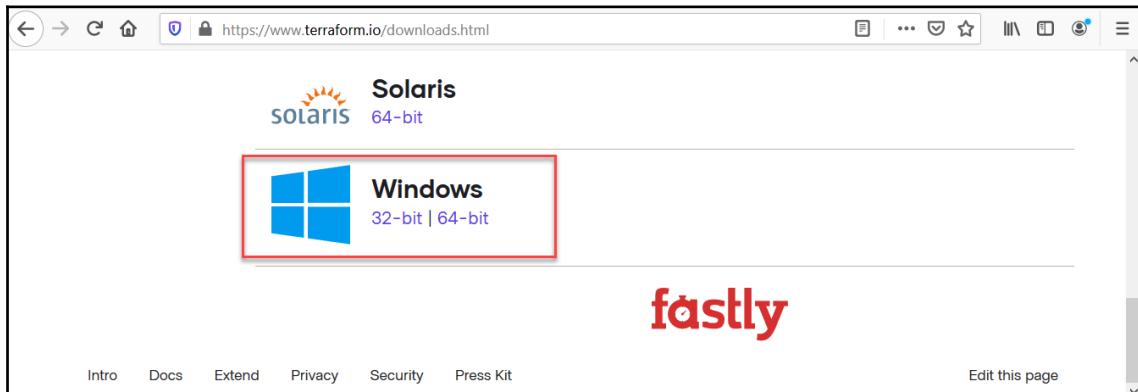
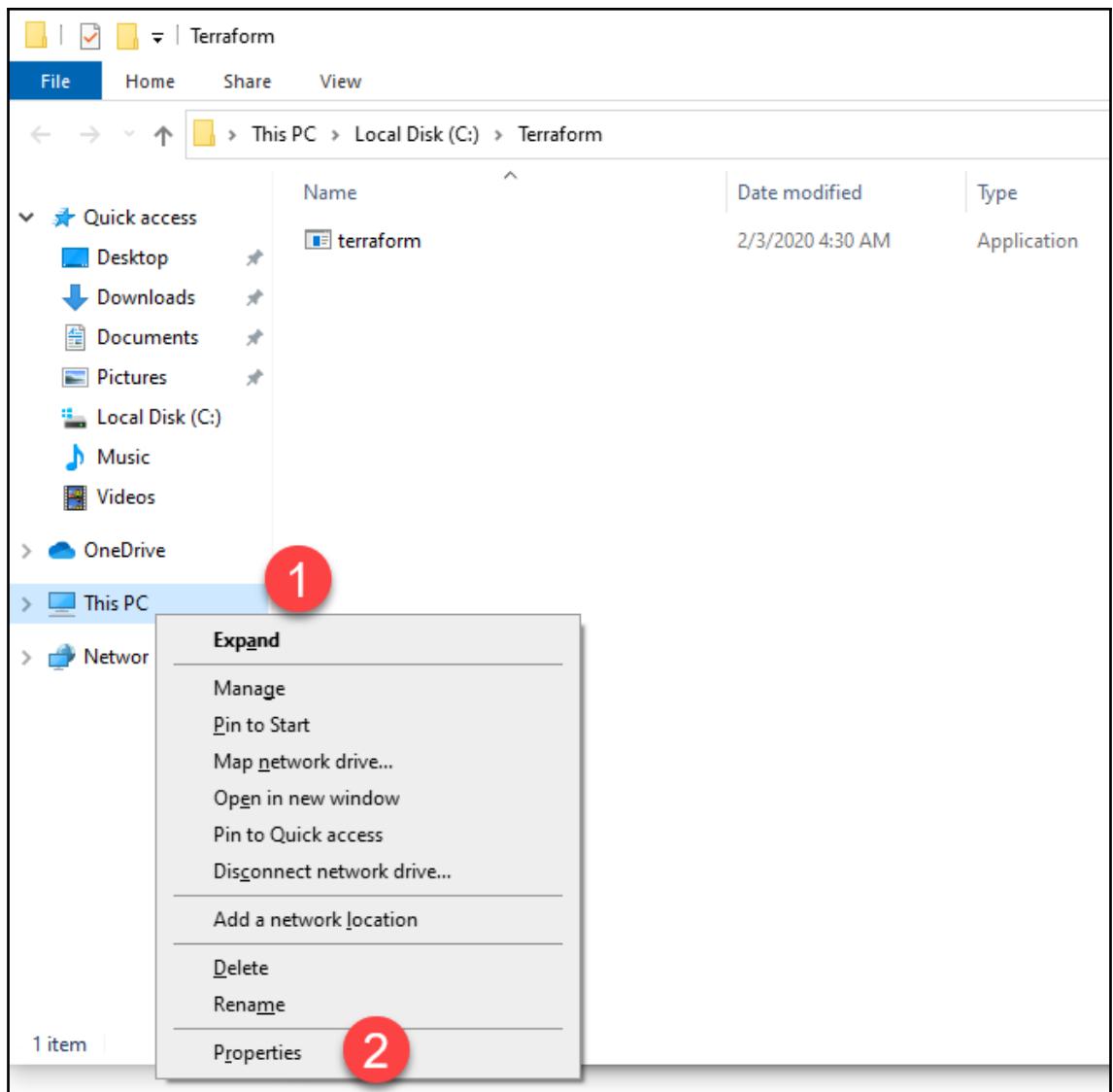
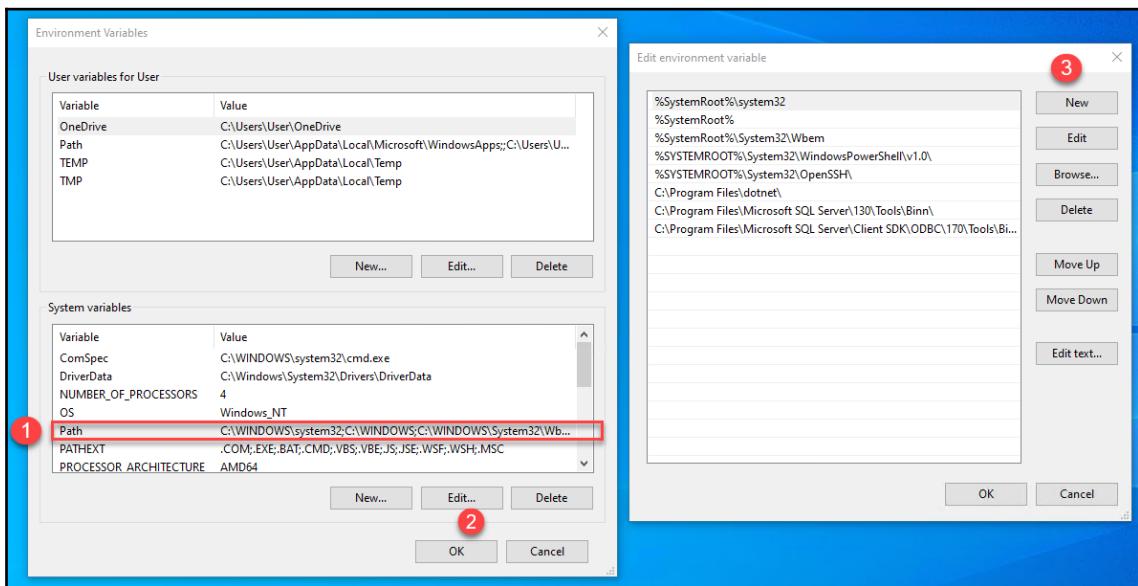
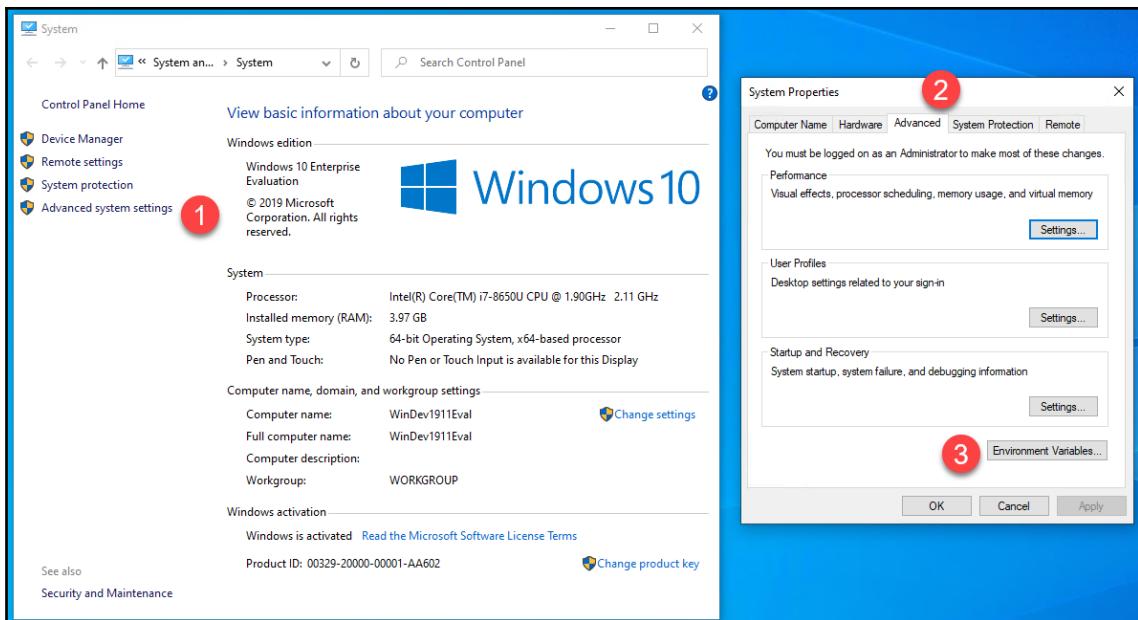
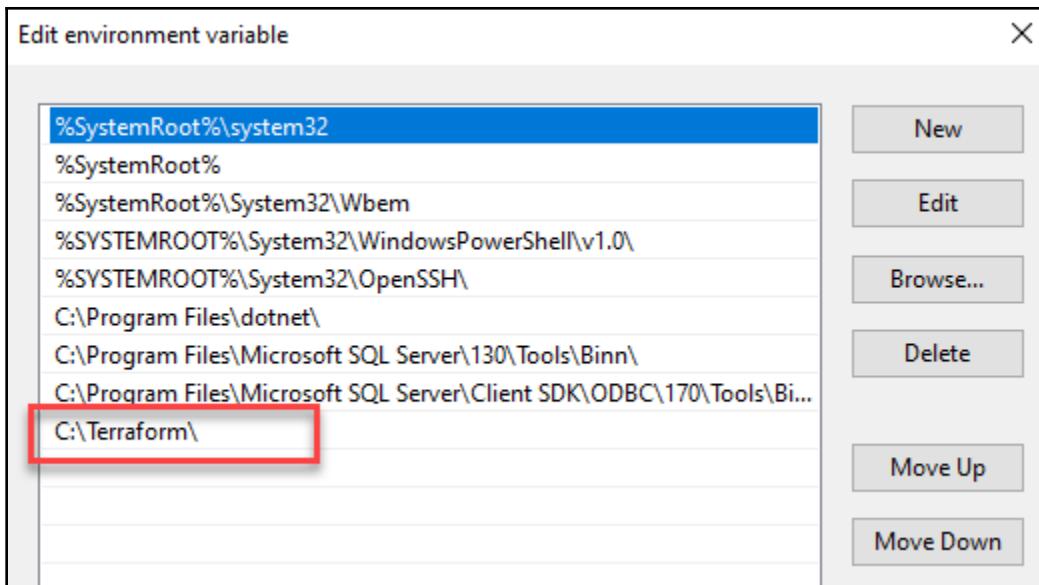


# Chapter 1: Setting Up the Terraform Environment









```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\User> terraform --help
Usage: terraform [-version] [-help] <command> [args]

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.

Common commands:
  apply           Builds or changes infrastructure
  console         Interactive console for Terraform interpolations
  destroy         Destroy Terraform-managed infrastructure
  env             Workspace management
  fmt              Rewrites config files to canonical format
  get              Download and install modules for the configuration
  graph            Create a visual graph of Terraform resources
  import           Import existing infrastructure into Terraform
  init             Initialize a Terraform working directory
  output           Read an output from a state file
  plan             Generate and show an execution plan
  providers        Prints a tree of the providers used in the configuration
  refresh          Update local state file against real resources
  show             Inspect Terraform state or plan
  taint            Manually mark a resource for recreation
  untaint          Manually unmark a resource as tainted
  validate         Validates the Terraform files
  version          Prints the Terraform version
  workspace        Workspace management

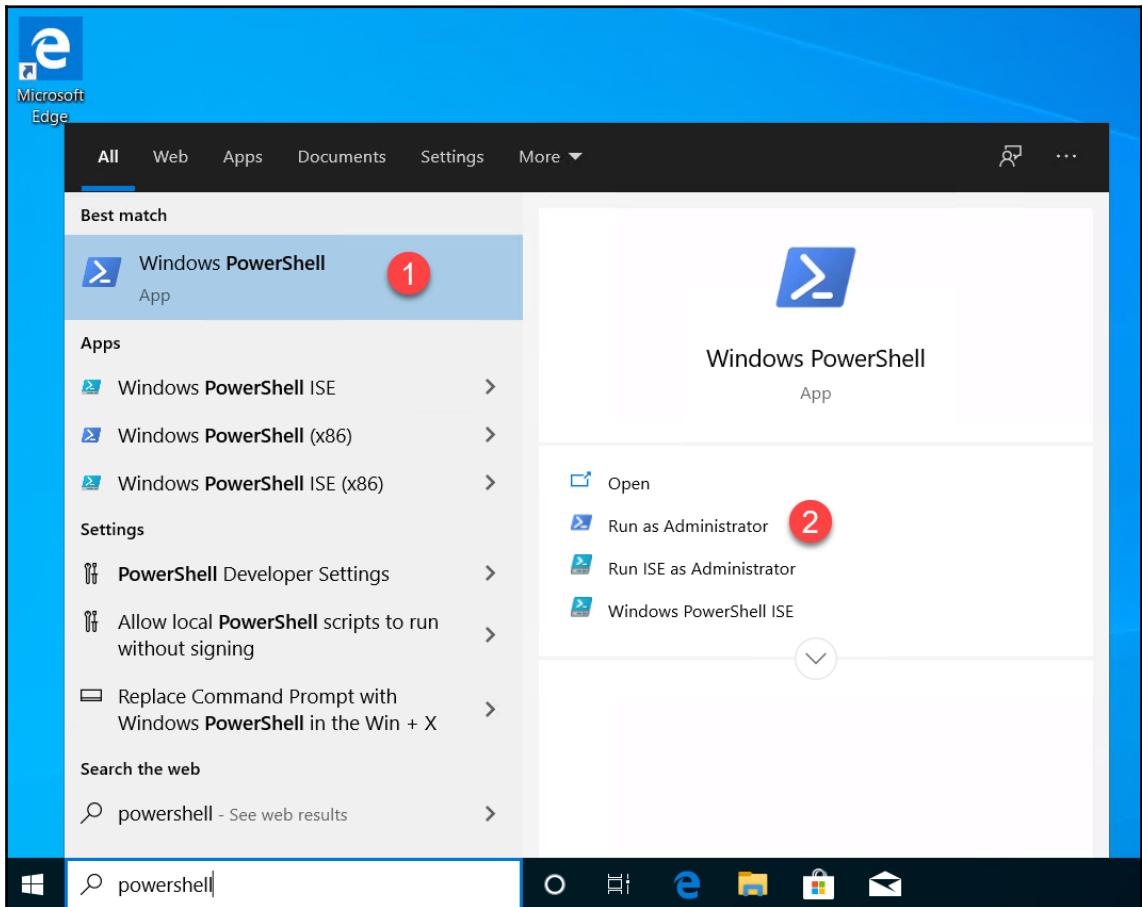
All other commands:
  0.12upgrade     Rewrites pre-0.12 module source code for v0.12
  debug            Debug output management (experimental)
  force-unlock    Manually unlock the terraform state
  push             Obsolete command for Terraform Enterprise legacy (v1)
  state            Advanced state management
```

```
root@LP-FYLZ2X2:/mnt/c/Users/mkrief# terraform --version  
Terraform v0.12.29
```



```
PS C:\Users\mkrief> terraform --version  
Terraform v0.12.28
```

Your version of Terraform is out of date! The latest version is 0.12.29. You can update by downloading from <https://www.terraform.io/downloads.html>



```
Administrator: Windows PowerShell
PS C:\Users\mikael> choco install -y terraform
Chocolatey v0.10.15
Installing the following packages:
terraform
By installing you accept licenses for the packages.
Progress: Downloading terraform 0.12.20... 100%          ↴

terraform v0.12.20 [Approved]
terraform package files install completed. Performing other installation steps.
Removing old terraform plugins
Downloading terraform 64 bit
  From 'https://releases.hashicorp.com/terraform/0.12.20/terraform_0.12.20_windows_amd64.zip'
Progress: 100%  Completed download of C:\Users\mikael\AppData\Local\Temp\chocolatey\terraform\0.12.20\terraform_0.12.20_windows_amd64.zip (15.47 MB).
Download of 'terraform_0.12.20_windows_amd64.zip (15.47 MB)' completed.
Extracting C:\Users\mikael\AppData\Local\Temp\chocolatey\terraform\0.12.20\terraform_0.12.20_windows_amd64.zip to C:\ProgramData\chocolatey\lib\terraform\tools...
C:\ProgramData\chocolatey\lib\terraform\tools
ShimGen has successfully created a shim for terraform.exe
The install of terraform was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\terraform\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Enjoy using Chocolatey! Explore more amazing features to take your
experience to the next level at
https://chocolatey.org/compare
```

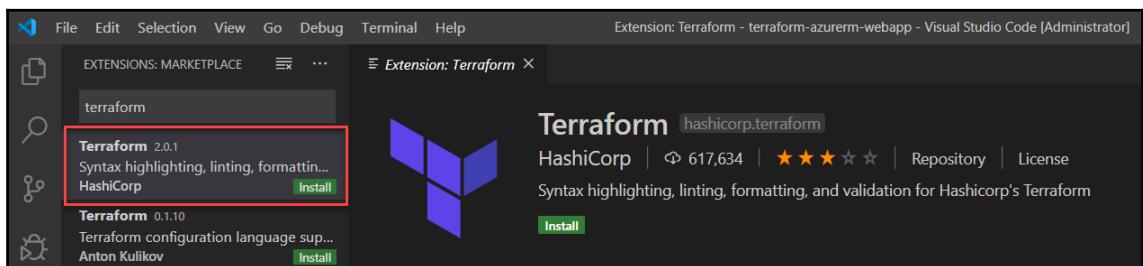
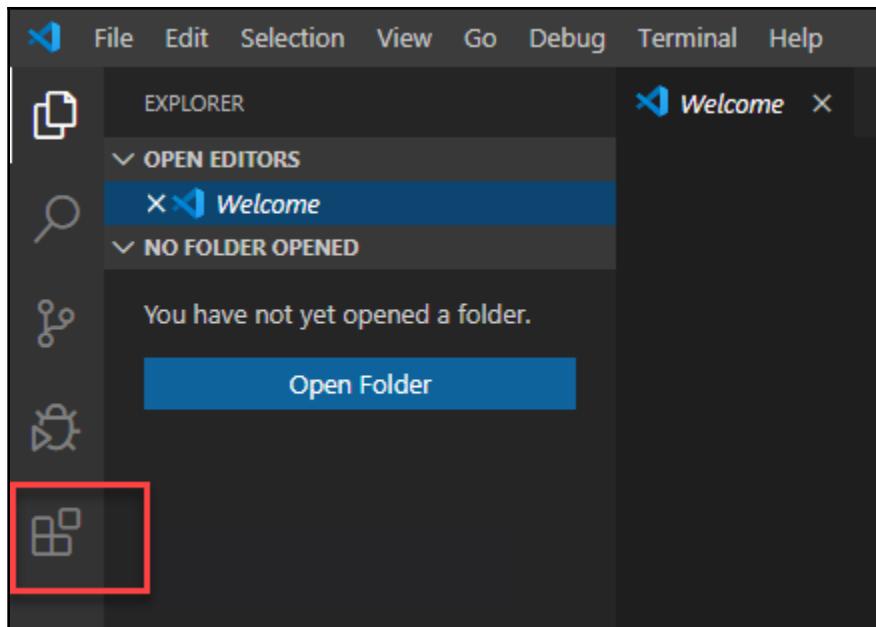
```
mkrief@LP-FYLZ2X2 MINGW64 /c/Program Files/Docker Toolbox
$ docker exec tfapp terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# random_string.random will be created
+ resource "random_string" "random" {
    + id          = (known after apply)
    + length      = 16
    + lower       = true
    + min_lower   = 0
    + min_numeric = 0
    + min_special = 0
    + min_upper   = 0
    + number      = true
    + result      = (known after apply)
    + special     = true
    + upper       = true
}

Plan: 1 to add, 0 to change, 0 to destroy.
```



A screenshot of the Visual Studio Code editor showing a Terraform file named 'main.tf'. The status bar indicates the file is 'main.tf'. The code editor displays the following Terraform configuration:

```
resource "random_id" "randomId" {
    byte_length = 8
}
```

```
0 references
5 resource "azurerm_app_service" "app" [
6
7    ↗ app_service_plan_id          (Required) string
8    ↗ app_service_plan_id
9    ↗ app_service_plan_id (azurerm_app_service)
10   ↗ app_settings
11   ↗ app_settings
12   ↗ app_settings (azurerm_app_service)
13   ↗ auth_settings
14   ↗ auth_settings
15   ↗ backup
16   ↗ backup
17   ↗ client_affinity_enabled
18   ↗ client_affinity_enabled
```

```
• main.tf - CHAP01 - Visual Studio Code [Administrator]
0 references
1 Welcome
2 main.tf
3
4 main.tf > ...
5 0 references
6 1 resource "random_id" "randomId" {
7 2 | byte_length = 8
8 3 }
9
10 0 references
11 5 resource "azurerm_app_service" "app" [
12 | name = "app"
13 ]
14
15 The argument "app_service_plan_id" is required, but no definition was found.
16 The argument "resource_group_name" is required, but no definition was found.
17 The argument "location" is required, but no definition was found.
18 The argument "location" is required, but no definition was found.
19 The argument "resource_group_name" is required, but no definition was found.
20 The argument "app_service_plan_id" is required, but no definition was found.
21
22 Peek Problem No quick fixes available
```

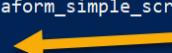
---

```
main.tf > ...
1 references
1   resource "random_id" "randomId" {
2     byte_length = 8
3   }
4
5   0 references
6   resource "azurerm_app_service" "app" {
7     name = "app${random_id.randomId.hex}"
8   }
```

Administrator: Windows PowerShell

```
PS C:\Users\mikael\Documents\terraform-cookbook\CHAP01\terraform_simple_script> terraform 0.12checklist
Looks good! We did not detect any problems that ought to be addressed before upgrading to Terraform v0.12.
```

This tool is not perfect though, so please check the v0.12 upgrade guide for additional guidance, and for next steps:  
<https://www.terraform.io/upgrade-guides/0-12.html>



```
PS C:\Terraform-Cookbook\CHAP01\tf011> terraform 0.12upgrade
```

This command will rewrite the configuration files in the given directory so that they use the new syntax features from Terraform v0.12, and will identify any constructs that may need to be adjusted for correct operation with Terraform v0.12.

We recommend using this command in a clean version control work tree, so that you can easily see the proposed changes as a diff against the latest commit. If you have uncommitted changes already present, we recommend aborting this command and dealing with them before running this command again.

```
Would you like to upgrade the module in the current directory?  
Only 'yes' will be accepted to confirm.
```

```
Enter a value: yes
```

---

```
Upgrade complete!
```

The configuration files were upgraded successfully. Use your version control system to review the proposed changes, make any necessary adjustments, and then commit.

```
PS C:\Terraform-Cookbook\CHAP01\tf011> terraform.exe 0.13upgrade
```

This command will update the configuration files in the given directory to use the new provider source features from Terraform v0.13. It will also highlight any providers for which the source cannot be detected, and advise how to proceed.

We recommend using this command in a clean version control work tree, so that you can easily see the proposed changes as a diff against the latest commit. If you have uncommitted changes already present, we recommend aborting this command and dealing with them before running this command again.

```
Would you like to upgrade the module in the current directory?  
Only 'yes' will be accepted to confirm.
```

```
Enter a value: yes
```

---

```
Upgrade complete!
```

Use your version control system to review the proposed changes, make any necessary adjustments, and then commit.

# Chapter 2: Writing Terraform Configuration

```
PS [REDACTED] \terraform-Cookbook\CHAP02> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

-----
Warning: "public_ip_address_allocation": [DEPRECATED] this property has been deprecated in favor of `allocation_method` to better match the api

on specific-version.tf line 24, in resource "azurerm_public_ip" "pip":
24: resource "azurerm_public_ip" "pip" {
```

```
PS [REDACTED] \terraform-Cookbook\CHAP02> terraform init

Error: Unsupported Terraform Core version

This configuration does not support Terraform version 0.12.28. To proceed,
either choose another supported Terraform version or update the root module's
version constraint. Version constraints are normally set for good reason, so
updating the constraint may lead to other errors or unexpected behavior.
```

```
PS [REDACTED] \terraform-Cookbook\CHAP02> terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "azurerm" (hashicorp/azurerm) 2.20.0...

Terraform has been successfully initialized!
```

```
PS [REDACTED] \terraform-Cookbook\CHAP02> terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "azurerm" (hashicorp/azurerm) 2.10.0...

Terraform has been successfully initialized!
```

```
PS C:\CHAP02\variables> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
```

```
-----
```

An execution plan has been generated and is shown below.  
Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
# azurerm_resource_group.rg will be created
+ resource "azurerm_resource_group" "rg" {
    + id      = (known after apply)
    + location = "westus"
    + name    = "My-RG"
    + tags    = (known after apply)
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

```
mikael@LP-FYLZ2X2: ~ /Terraform-Cookbook/CHAP02/variables$ terraform plan
Error: Invalid value for variable

on main.tf line 19:
19: variable "location" {

The location must be westeurope or westus.

This was checked by the validation rule at main.tf:22,3-13.
```

```

# azurerm_resource_group.rg will be created
+ resource "azurerm_resource_group" "rg" {
  + id      = (known after apply)
  + location = "westeurope"
  + name    = "RG-myappdemo-dev-fr"
  + tags    = (known after apply)
}

```

Plan: 2 to add, 0 to change, 0 to destroy.

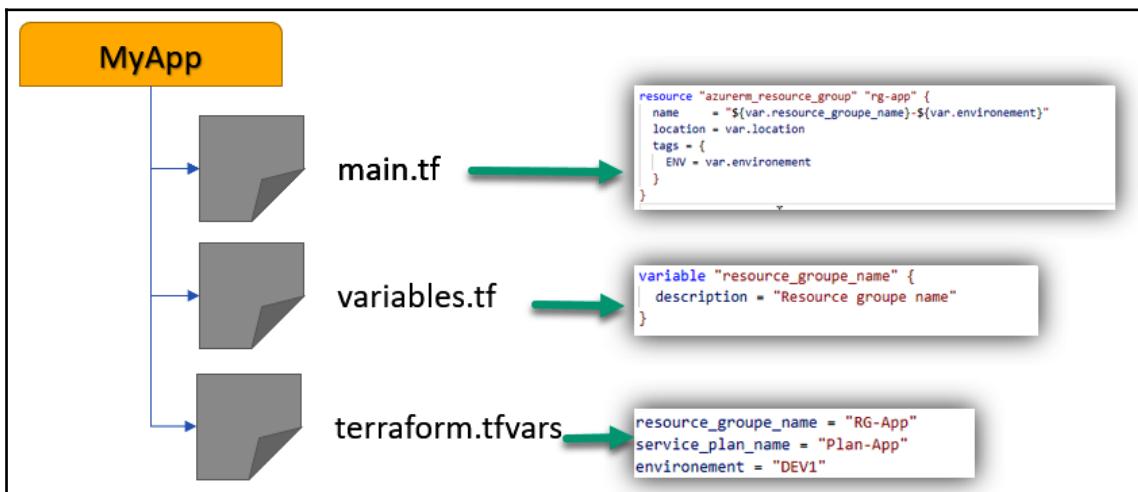
```

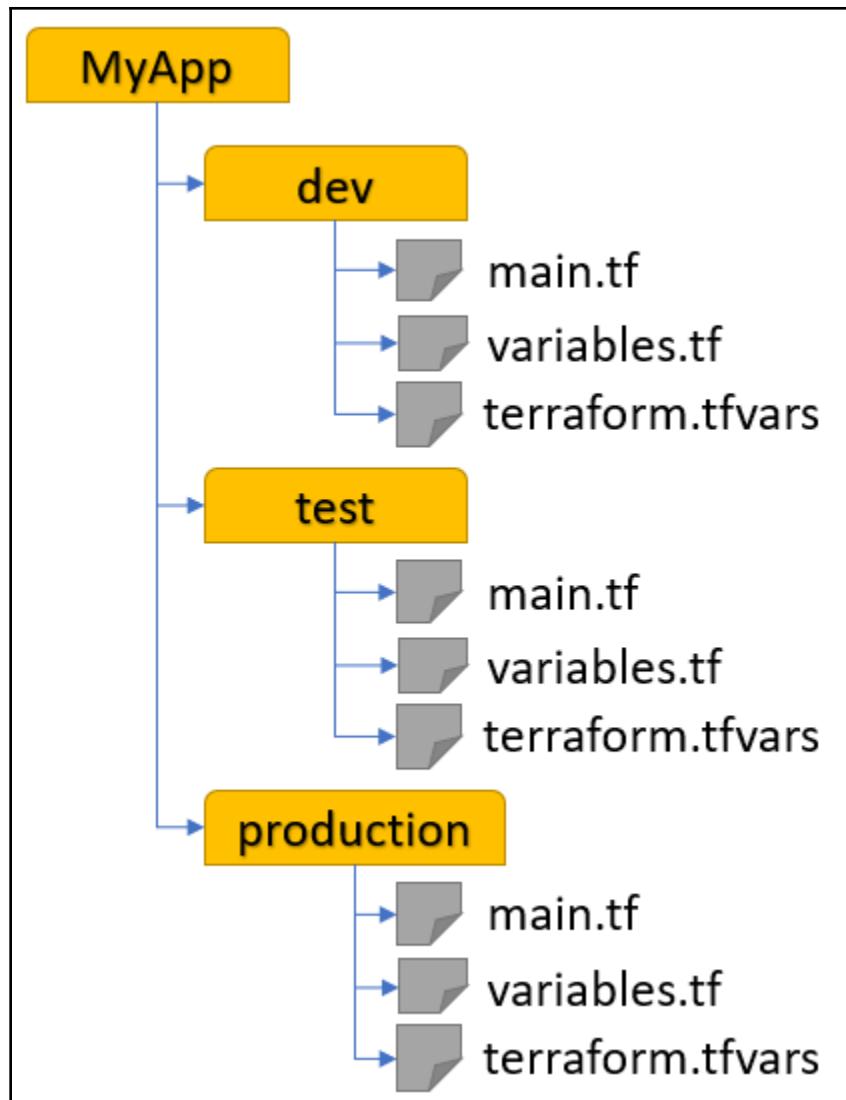
PS C:\...\terraform-Cookbook\CHAP02\sample-app> terraform apply
azurerm_resource_group.rg-app: Refreshing state... [id=/subscriptions/...]/resourceGroup
s/RG-App-demo-DEV1]
azurerm_application_insights.appinsight-app: Refreshing state... [id=/subscriptions/...
/resourceGroups/RG-App-demo-DEV1/providers/microsoft.insights/components/MyApp-demo-DEV1]
azurerm_app_service_plan.plan-app: Refreshing state... [id=/subscriptions/...
/resourceGroups/RG-App-demo-DEV1/providers/Microsoft.Web/serverfarms/Plan-App-demo-DEV1]
azurerm_app_service.app: Refreshing state... [id=/subscriptions/...
/resourceGroups/RG-App-demo-DEV1/providers/Microsoft.Web/sites/MyApp-demo-DEV1]

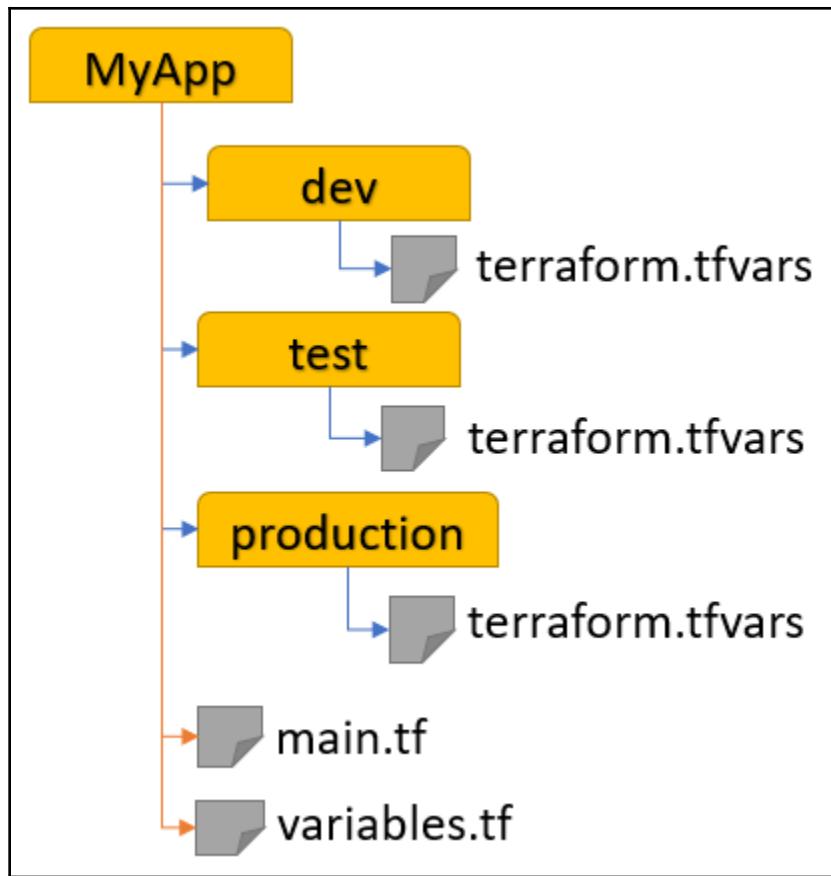
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:
webapp_hostname = myapp-demo-dev1.azurewebsites.net
webapp_name = MyApp-demo-DEV1

```







```

PS C:\Users\...\Desktop\Terraform\CHAP02\data> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.azurerm_app_service_plan.myplan: Refreshing state...

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# azurerm_app_service.app will be created
+ resource "azurerm_app_service" "app" {
  + app_service_plan_id = "/subscriptions/.../resourceGroups/rg-service_plan/providers/Microsoft.web/serverfarms/app-service-plan"
    + app_settings          = (known after apply)
    + client_affinity_enabled = (known after apply)
    + default_site_hostname = (known after apply)
    + enabled                = true
}

```

```
PS C:\CHAP02\data> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.azurerm_app_service_plan.myplan: Refreshing state...

Error: Error: App Service Plan "app-service-plan" (Resource Group "rg-service_plan") was not found

on main.tf line 25, in data "azurerm_app_service_plan" "myplan":
25: data "azurerm_app_service_plan" "myplan" {
```

```
PS C:\REPOSPERSO\Terraform-Cookbook\CHAP02\external> terraform plan -var "environment_name=Dev"
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.external.getlocation: Refreshing state... ←

-----
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# azurerm_resource_group.rg will be created
+ resource "azurerm_resource_group" "rg" {
    + id      = (known after apply)
    + location = "westus" ←
    + name    = "RG-myappdemo-Dev-fr"
    + tags    = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
PS C:\CHAP02\external> terraform plan -var "environment_name=Production"
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.external.getlocation: Refreshing state...

-----
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# azurerm_resource_group.rg will be created
+ resource "azurerm_resource_group" "rg" {
    + id      = (known after apply)
    + location = "westeurope" [+]
    + name    = "RG-myappdemo-Production-fr"
    + tags    = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
PS C:\CHAP02\external> terraform apply
data.external.getlocation: Refreshing state...
azurerm_resource_group.rg: Refreshing state... [id=/subscriptions/[REDACTED]/resourceGroups/RG-myappdemo-Production-fr]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:
locationname = westeurope
```

```
PS C:\CHAP02\fct> terraform plan  
Refreshing Terraform state in-memory prior to plan...  
The refreshed state will be used to calculate this plan, but will not be  
persisted to local or remote state storage.
```

An execution plan has been generated and is shown below.  
Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
# azurerm_resource_group.rg-app will be created  
+ resource "azurerm_resource_group" "rg-app" {  
    + id      = (known after apply)  
    + location = "westeurope"  
    + name    = "RG-MYAPP-DEV"  
    + tags    = (known after apply)  
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

```
PS | \CHAP02\fct> terraform plan  
Refreshing Terraform state in-memory prior to plan...  
The refreshed state will be used to calculate this plan, but will not be  
persisted to local or remote state storage.
```

An execution plan has been generated and is shown below.  
Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
# azurerm_resource_group.rg-app will be created  
+ resource "azurerm_resource_group" "rg-app" {  
    + id      = (known after apply)  
    + location = "westeurope"  
    + name    = "RG-MYAPP"  
    + tags    = (known after apply)  
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

```
PS C:\CHAP02\fct> terraform plan  
Refreshing Terraform state in-memory prior to plan...  
The refreshed state will be used to calculate this plan, but will not be  
persisted to local or remote state storage.
```

An execution plan has been generated and is shown below.  
Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
# azurerm_resource_group.rg-app will be created  
+ resource "azurerm_resource_group" "rg-app" {  
    + id      = (known after apply)  
    + location = "westeurope"  
    + name    = "RG-MYAPP-DEV"  
    + tags    = (known after apply)  
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

---

An execution plan has been generated and is shown below.  
Resource actions are indicated with the following symbols:  
+ create

Terraform will perform the following actions:

```
# local_file myfile will be created
+ resource "local_file" "myfile" {
    + content          = "This is my text"
    + directory_permission = "0777"
    + file_permission   = "0777"
    + filename         = "../mytextfile.txt"
    + id               = (known after apply)
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

```
Plan: 2 to add, 0 to change, 0 to destroy.
```

```
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
```

```
Enter a value: yes
```

```
null_resource.readcontentfile: Creating...
null_resource.readcontentfile: Provisioning with 'local-exec'...
null_resource.readcontentfile (local-exec): Executing: ["PowerShell" "-Command" "Get-Content -Path ../mytextfile.txt"]
local_file myfile: Creating...
local_file myfile: Creation complete after 0s [id=2a29c5a983236a8f5c0fde5c48b7d15a5cb7d47b]
null_resource.readcontentfile (local-exec): This is my text ←
null_resource.readcontentfile: Creation complete after 1s [id=8451305302740975700]
```

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

```
# random_password.password will be created
+ resource "random_password" "password" {
    + id                  = (known after apply)
    + length              = 16
    + lower               = true
    + min_lower           = 0
    + min_numeric         = 0
    + min_special         = 0
    + min_upper            = 0
    + number              = true
    + override_special    = "%@"
    + result              = (sensitive value)
    + special              = true
    + upper                = true
}

```

Plan: 8 to add, 0 to change, 0 to destroy.

# Chapter 3: Building Dynamic Environments with Terraform

```
Terraform will perform the following actions:
```

```
# azurerm_app_service.app[0] will be created ←
+ resource "azurerm_app_service" "app" {
    + app_service_plan_id          = (known after apply)
    + app_settings                 = (known after apply)
    + client_affinity_enabled     = (known after apply)
    + default_site_hostname       = (known after apply)
    + enabled                      = true
    + https_only                   = false
    + id                           = (known after apply)
    + location                     = "westeurope"
    + name                         = "MyApp-DEV1-1" ←
    + outbound_ip_addresses        = (known after apply)
```

```
# azurerm_app_service.app[1] will be created ←
+ resource "azurerm_app_service" "app" {
    + app_service_plan_id          = (known after apply)
    + app_settings                 = (known after apply)
    + client_affinity_enabled     = (known after apply)
    + default_site_hostname       = (known after apply)
    + enabled                      = true
    + https_only                   = false
    + id                           = (known after apply)
    + location                     = "westeurope"
    + name                         = "MyApp-DEV1-2" ←
    + outbound_ip_addresses        = (known after apply)
```

---

```
Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
```

Outputs:

```
app_service_names = [  
    "MyApp-DEV1-1",  
    "MyApp-DEV1-2",  
]
```

```
# azurerm_app_service.app will be created  
+ resource "azurerm_app_service" "app" {  
    + app_service_plan_id          = (known after apply)  
    + app_settings                = {  
        + "KEY1" = "VAL1"  
    }  
    + client_affinity_enabled     = (known after apply)  
    + default_site_hostname       = (known after apply)  
  
# azurerm_resource_group.rg-app will be created  
+ resource "azurerm_resource_group" "rg-app" {  
    + id            = (known after apply)  
    + location      = "westeurope"  
    + name          = "RG-App-DEV1"  
    + tags          = {  
        + "CODE_PROJECT" = "DEMO"  
        + "ENV"           = "DEV1"  
    }  
}
```

---

```
Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
```

Outputs:

```
app_service_names = [  
    "webappdemobook1",  
    "webapptestbook2",  
]
```

```
+ security_rule      = [
+ {
+   access                  = "Allow"
+   description              = ""
+   destination_address_prefix = "*"
+   destination_address_prefixes = []
+   destination_application_security_group_ids = []
+   destination_port_range    = "22"
+   destination_port_ranges   = []
+   direction                = "Inbound"
+   name                     = "rule"
+   priority                 = 110
+   protocol                 = "Tcp"
+   source_address_prefix     = "*"
+   source_address_prefixes   = []
+   source_application_security_group_ids = []
+   source_port_range         = "*"
+   source_port_ranges        = []
},
+
+ {
+   access                  = "Allow"
+   description              = ""
+   destination_address_prefix = "*"
+   destination_address_prefixes = []
+   destination_application_security_group_ids = []
+   destination_port_range    = "80"
+   destination_port_ranges   = []
+   direction                = "Inbound"
+   name                     = "rule1"
+   priority                 = 100
+   protocol                 = "Tcp"
+   source_address_prefix     = "*"
+   source_address_prefixes   = []
+   source_application_security_group_ids = []
+   source_port_range         = "*"
}
```

---

## Chapter 4: Using the Terraform CLI

```
variable "rg_name"{
  description="Name of the resource group"
}

variable "location"{
  description ="location"
  default="westeurope"
}

resource "azurerm_resource_group" "rg-app" [
  name=var.rg_name
  location=var.location
  tags={
    ENV="DEMO"
}
```

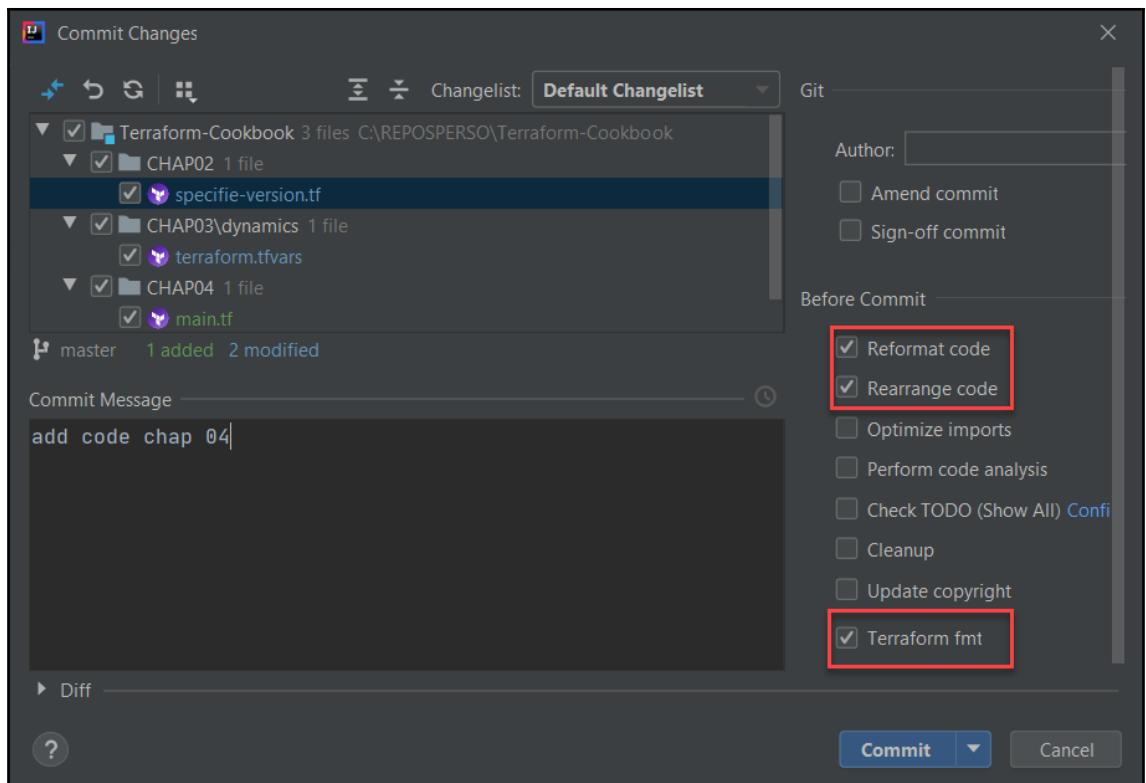
```
PS C:\...\Terraform-Cookbook\CHAP04> terraform fmt
main.tf
```

```
variable "rg_name" {
  description = "Name of the resource group"
}

variable "location" {
  description = "location"
  default     = "westeurope"
}

resource "azurerm_resource_group" "rg-app" {
  name      = var.rg_name
  location = var.location
  tags     = {
    ENV = "DEMO"
  }
}
```

```
PS C:\Terraform-Cookbook\CHAP04> terraform fmt -recursive
main.tf
sub\main.tf ←
```



```
variable "rg_name" {
  description = "Name of the resource group"
}

variable "location" {
  description = "location"
  default     = "westeurope"
}

resource "azurerm_resource_group" "rg-app" {
  name      = var.rg_name
  location  = var.location
  tags      = {
    ENV = var.environment
  }
}
```

```
PS [REDACTED]\Terraform-Cookbook\CHAP04> terraform validate
Error: Reference to undeclared input variable

on main.tf line 19, in resource "azurerm_resource_group" "rg-app":
19:   ENV = var.environment }

An input variable with the name "environment" has not been declared. This
variable can be declared with a variable "environment" {} block.
```

```
PS [REDACTED]\Terraform-Cookbook\CHAP04> terraform validate
Success! The configuration is valid.
```

```

PS C:\REPOS\REPO\Terraform-Cookbook\CHAP04> terraform validate -json
{
  "valid": false,
  "error_count": 1,
  "warning_count": 0,
  "diagnostics": [
    {
      "severity": "error",
      "summary": "Reference to undeclared input variable",
      "detail": "An input variable with the name \"environment\" has not been declared. This variable can be declared with a variable \"environment\""
    }
  ]
}

```

```

PS C:\REPOS\REPO\Terraform-Cookbook\chap04\sample-app> terraform destroy
azurerm_resource_group.rg-app: Refreshing state... [id=/subscriptions/8a7aaece5-          -2c292b6304e5/resourceGroups/RG-App-DEV1]
azurerm_application_insights.appinsight-app: Refreshing state... [id=/subscriptions/8a7aaece5-          -2c292b6304e5/resourceGroups/RG-App-DEV1/providers/
s/microsoft.insights/components/MyApp-DEV1]
azurerm_app_service_plan.plan-app: Refreshing state... [id=/subscriptions/8a7aaece5-          -2c292b6304e5/resourceGroups/RG-App-DEV1/providers/Microsof
t.Web/serverfarms/Plan-APP-DEV1]
azurerm_app_service.app: Refreshing state... [id=/subscriptions/8a7aaece5-          -2c292b6304e5/resourceGroups/RG-App-DEV1/providers/Microsoft.Web/sites/
MyApp-DEV1]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
- destroy

Plan: 0 to add, 0 to change, 4 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

azurerm_application_insights.appinsight-app: Destroying... [id=/subscriptions/8a7aaece5-          -2c292b6304e5/resourceGroups/RG-App-DEV1/providers/micr
osoft.insights/components/MyApp-DEV1]
azurerm_app_service.app: Destroying... [id=/subscriptions/8a7aaece5-          -2c292b6304e5/resourceGroups/RG-App-DEV1/providers/Microsoft.Web/sites/
MyApp-DEV1]
azurerm_resource_group.rg-app: Destruction complete after 1m46s

Destroy complete! Resources: 4 destroyed. ←
PS C:\REPOS\REPO\Terraform-Cookbook\chap04\sample-app>

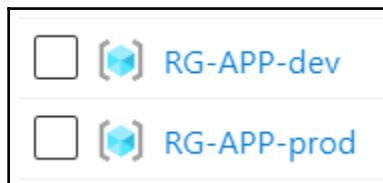
```

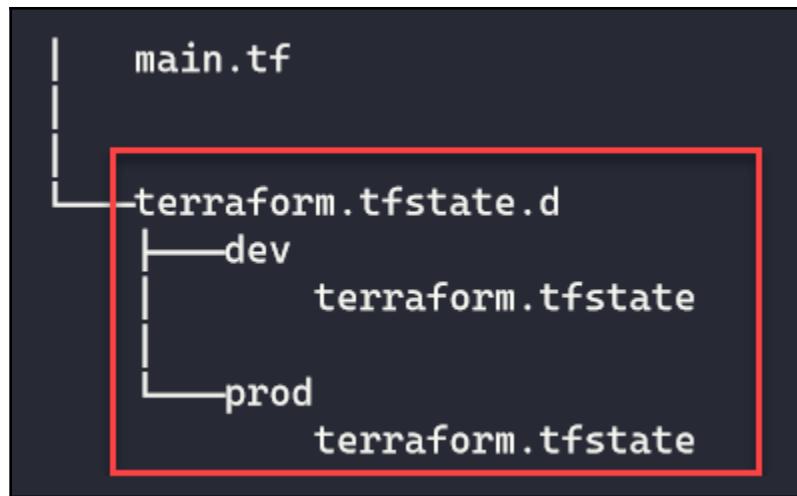
```

PS C:\REPOS\REPO\Terraform-Cookbook\CHAP04\workspaces> terraform workspace new dev
Created and switched to workspace "dev"!

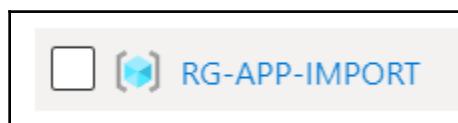
You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.

```





```
PS C:\Terraform-Cookbook\CHAP04\workspaces> terraform workspace list
default
dev
* prod
```



```
PS C:\Terraform-Cookbook\CHAP04\import> terraform apply -auto-approve
azurerm_resource_group.rg-app: Creating ...
Error: A resource with the ID "/subscriptions/8a7aace5-74aa-416f-8d4e-1c77b301ca26/resourceGroups/RG-APP-IMPORT" already exists - to be managed via Terraform this resource needs to be imported into the State. Please see the resource documentation for "azurerm_resource_group" for more information.

on main.tf line 10, in resource "azurerm_resource_group" "rg-app":
 10: resource "azurerm_resource_group" "rg-app" {
```

```
PS C:\Terraform-Cookbook\CHAP04\import> terraform import azurerm_resource_group.rg-app /subscriptions/8a7aace5-74aa-416f-8d4e-1c77b301ca26/resourceGroups/RG-APP-IMPORT
azurerm_resource_group.rg-app: Importing from ID "/subscriptions/8a7aace5-74aa-416f-8d4e-1c77b301ca26/resourceGroups/RG-APP-IMPORT" ...
azurerm_resource_group.rg-app: Import prepared!
Prepared azurerm_resource_group for import
azurerm_resource_group.rg-app: Refreshing state ... [id=/subscriptions/8a7aace5-74aa-416f-8d4e-1c77b301ca26/resourceGroups/RG-APP-IMPORT]

Import successful!
```

The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.

```
PS C:\Terraform-Cookbook\CHAP04\import> terraform plan
Refreshing Terraform state in-memory prior to plan ...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

azurerm_resource_group.rg-app: Refreshing state ... [id=/subscriptions/8a7aace5-74aa-0000-0000-000000000000/resourceGroups/RG-APP-IMPORT]

-----
No changes. Infrastructure is up-to-date.

This means that Terraform did not detect any differences between your
configuration and real physical resources that exist. As a result, no
actions need to be performed.
```

```
mikael@LP-FYLZ2X2: /Terraform-Cookbook/CHAP03/list_map$ terraform output
app_service_names = [
  "webappdemobook1",
  "webapptestbook2",
]
app_service_urls = {
  "webappdemobook1" = "webappdemobook1.azurewebsites.net"
  "webapptestbook2" = "webapptestbook2.azurewebsites.net"
}
```

```
mikael@LP-FYLZ2X2: /Terraform-Cookbook/CHAP03/list_map$ terraform output -json
{
  "app_service_names": {
    "sensitive": false,
    "type": [
      "tuple",
      [
        "string",
        "string"
      ]
    ],
    "value": [
      "webappdemobook1",
      "webapptestbook2"
    ]
  },
  "app_service_urls": {
    "sensitive": false,
    "type": [
      "object",
      {
        "webappdemobook1": "string",
        "webapptestbook2": "string"
      }
    ],
    "value": {
      "webappdemobook1": "webappdemobook1.azurewebsites.net",
      "webapptestbook2": "webapptestbook2.azurewebsites.net"
    }
  }
}
```

```
mikael@LP-FYLZ2X2: /Terraform-Cookbook/CHAP03/list_map$ urlwebappl=$(terraform output -json | jq -r .app_service_urls.value.webappdemobook1)
mikael@LP-FYLZ2X2: /Terraform-Cookbook/CHAP03/list_map$ curl -sL "%{http_code}" -I "$urlwebappl/hostingstart.html"
HTTP/1.1 200 OK ←
Content-Length: 3499
Content-Type: text/html
Last-Modified: Tue, 28 Apr 2020 12:30:10 GMT
Accept-Ranges: bytes
ETag: "c6f6c2581dd61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Set-Cookie: ARRAffinity=0bfcc5d1cc011b1d989c47453347b28e7ed30ce1227470a6d28ed0f9ce4bf3802;Path=/;HttpOnly;Domain=webappdemobook1.azurewebsites.net
Date: Tue, 28 Apr 2020 16:29:27 GMT
```

```
mikael@LP-FYLZ2X2: /Terraform-Cookbook/CHAP03/list_map$ terraform output -json app_service_urls
{"webappdemobook1": "webappdemobook1.azurewebsites.net", "webapptestbook2": "webapptestbook2.azurewebsites.net"}
```

```
PS C:\Terraform-Cookbook\CHAP02\myApp\simple-env> terraform taint azurerm_app_service.app
Resource instance azurerm_app_service.app has been marked as tainted.
```

```
PS C:\Terraform-Cookbook\CHAP02\myApp\simple-env> terraform apply
azurerm_resource_group.rg-app: Refreshing state ... [id=/subscriptions/8a7aac5-... /resourceGroups/RG-App-DEV1]
azurerm_application_insights.appinsight-app: Refreshing state ... [id=/subscriptions/8a7aac5-... /resourceGroups/RG-App-DEV1/providers/microsoft.insights/components/MyApp-DEV1]

Terraform will perform the following actions:

# azurerm_app_service.app is tainted, so must be replaced ←
-/+ resource "azurerm_app_service" "app" {
    app_service_plan_id      = "/subscriptions/8a7aac5-74aa... /resourceGroups/RG-App-DEV1/providers/Microsoft.Web/serverfarms/Plan-App-DEV1"
    app_settings             = {
        - "WEBSITE_NODE_DEFAULT_VERSION" = "6.9.1"
    } → (known after apply)
    client_affinity_enabled = true → (known after apply)
    client_cert_enabled     = false → null
    default_site_hostname   = "myapp-dev1.azurewebsites.net" → (known after apply)
    enabled                 = true
    https_only              = false
}

Apply complete! Resources: 1 added, 0 changed, 1 destroyed. ←
```

```
PS C:\Terraform-Cookbook\CHAP02\myApp\simple-env> terraform state show azurerm_app_service.app
# azurerm_app_service.app: (tainted) ←
resource "azurerm_app_service" "app" {
    app_service_plan_id      = "/subscriptions/... /resourceGroups/RG-App-DEV1/providers/Microsoft.Web/serverfarms/Plan-App-DEV1"
    app_settings             = {
        - "WEBSITE_NODE_DEFAULT_VERSION" = "6.9.1"
    }
    client_affinity_enabled = true
    client_cert_enabled     = false
    default_site_hostname   = "myapp-dev1.azurewebsites.net"
```

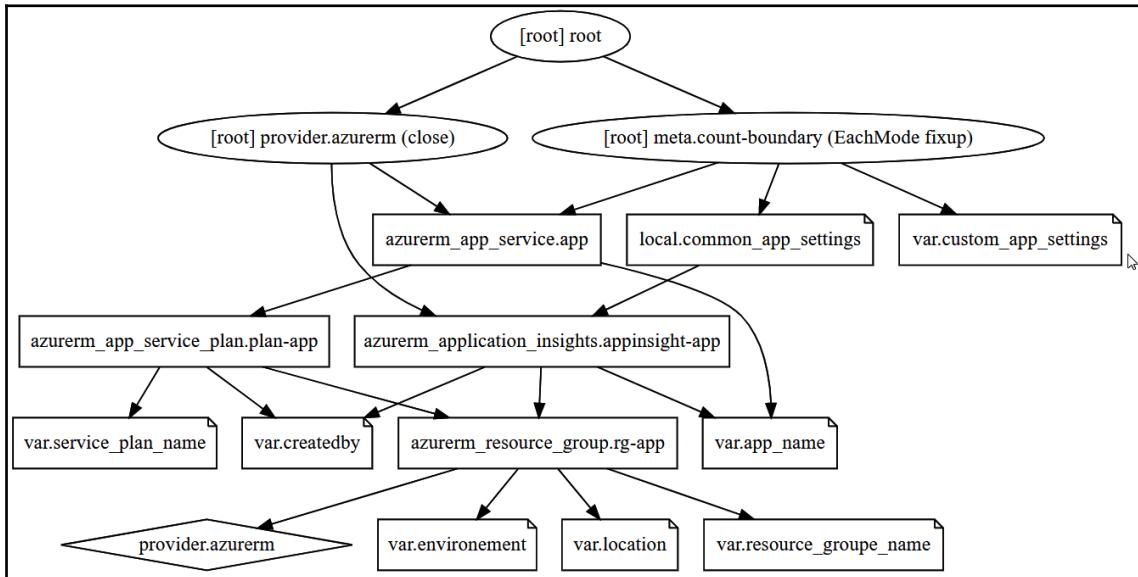
```

PS C:\Terraform-Cookbook\CHAP02\myApp\simple-env> terraform untaint azurerm_app_service.app
Resource instance azurerm_app_service.app has been successfully untainted. ①
PS C:\Terraform-Cookbook\CHAP02\myApp\simple-env> terraform plan ②
Refreshing Terraform state in-memory prior to plan ...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

azurerm_resource_group.rg-app: Refreshing state ... [id=/subscriptions/8a7aac5-74aa-416f-b8e
azurerm_application_insights.appinsight-app: Refreshing state ... [id=/subscriptions/8a7aac5-74aa-416f-b8e
.azureinsights/components/MyApp-DEV1] :/RG-App-DEV1]
azurerm_app_service_plan.plan-app: Refreshing state ... [id=/subscriptions/8a7aac5-74aa-416f-b8e
r Farms/Plan-App-DEV1] :/resourceGroups/RG-App-DEV1/providers/Microsoft.Web/serve
azurerm_app_service.app: Refreshing state ... [id=/subscriptions/8a7aac5-74aa-416f-b8e
1] :/resourceGroups/RG-App-DEV1/providers/Microsoft.Web/sites/MyApp-DEV1

No changes. Infrastructure is up-to-date. ←
This means that Terraform did not detect any differences between your
configuration and real physical resources that exist. As a result, no
actions need to be performed.

```

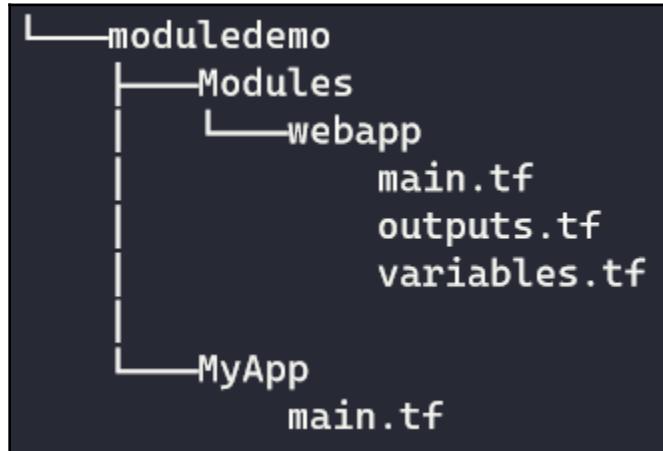


```

: [DEBUG] New state was assigned lineage "66c67b80-04e1-1834-1a70-af4486a8ea99"
[TRACE] Meta.Backend: using default local state only (no backend configuration, and no existing initialized backend)
[TRACE] Meta.Backend: instantiated backend of type <nil>
[DEBUG] checking for provider in "."
  [DEBUG] checking for provider in "C:\\ProgramData\\chocolatey\\lib\\terraform\\tools"
  [DEBUG] checking for provider in ".terraform\\plugins\\windows_amd64"
[DEBUG] found provider "terraform-provider-azurerm_v2.7.0_x5.exe"
[DEBUG] found valid plugin: "azurerm", "2.7.0", "C:\\REPOSERSO\\Terraform-Cookbook\\CHAP02\\myApp\\simple-env\\.terraform\\plugins\\windows_amd64\\azurerm_v2.7.0.x5.exe"
[DEBUG] checking for provisioner in "."

```

# Chapter 5: Sharing Terraform Configuration with Modules



```
PS C:\Terraform-Cookbook\CHAP05\moduledemo\MyApp> terraform init
Initializing modules...
- webapp in ..\Modules\webapp
```

A red arrow points from the text 'webapp in ..\Modules\webapp' to the 'webapp' folder in the file structure diagram above.

```
Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "azurerm" (hashicorp/azurerm) 2.8.0...
```

**Outputs:**

```
webapp_url = myappdemobook.azurewebsites.net
```

The screenshot shows the HashiCorp Terraform Registry interface. At the top, there's a navigation bar with the Terraform logo, the word "Terraform", a "Registry" link, a search bar containing "Search Providers and Modules", and a "Browse" button. Below the navigation, there are two tabs: "Providers" and "Modules", with "Modules" being the active tab. A section titled "Modules" is displayed, with a sub-section header "Azure / network". This module is described as the "Terraform Azure RM Module for Network". It was last updated "2 months ago" and has been downloaded "42 758" times. On the left side of the main content area, there is a "FILTERS" section containing a dropdown menu labeled "Provider" with "azurerm" selected. A red box highlights this dropdown.

This is a detailed view of the "Azure / network" Terraform module. At the top, it features the Microsoft logo followed by the module name "Azure / network" and a blue checkmark icon. Below the name, the description "Terraform Azure RM Module for Network" is provided. Further down, it shows the last update time as "a month ago" and the download count as "37420". To the right of the download count is a blue hexagonal icon with a white triangle and the word "provider".

## Usage

```
resource "azurerm_resource_group" "test" {
    name      = "my-resources"
    location = "West Europe"
}

module "network" {
    source          = "Azure/network/azurerm"
    resource_group_name = azurerm_resource_group.test.name
    address_space     = "10.0.0.0/16"
    subnet_prefixes   = ["10.0.1.0/24", "10.0.2.0/24", "10.0.3.0/24"]
    subnet_names      = ["subnet1", "subnet2", "subnet3"]

    tags = {
        environment = "dev"
        costcenter  = "it"
    }
}
```

The screenshot shows the Terraform Registry interface. At the top, there is a navigation bar with the Terraform logo, a search bar labeled "Search Providers and Modules", and links for "Browse", "Publish", and "Sign-in".

The main content area displays a module named "network" by "AZURERM". The module icon is a blue triangle. The title is "Terraform Azure RM Module for Network". Below the title, there is a summary: "Published May 26, 2020 by Azure", "Module managed by yupwei68", "Total provisions: 42759", and "Source Code: [github.com/Azure/terraform-azurerm-network](https://github.com/Azure/terraform-azurerm-network)".

To the right of the module details, a dropdown menu shows version history:

- Version 3.1.1 (selected, highlighted with a red circle containing the number 1)
- Version 3.0.1
- Version 3.0.0
- Version 2.0.0
- Version 1.1.1
- Show All

On the far right, under the heading "Provision Instructions", there is a code snippet for Terraform configuration:

```
module "network" {  
  source  = "Azure/network/azurerm"  
  version = "3.1.1"  
  # insert the 1 required variable  
}
```

A red circle with the number 2 is positioned near the "Provision Instructions" section.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner

Repository name \*



mikaelkrief ▾

/

terraform-azurerm-webapp



1

Great repository names are short and memorable. Need inspiration? How about **urban-chainsaw**?

Description (optional)

Terraform module for provision Azure Service plan, App service (web app) and Application Insight

Public

Anyone can see this repository. You choose who can commit.

2

Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README 3

This will let you immediately clone the repository to your computer.

Add .gitignore: Terraform ▾

4

Add a license: None ▾

ⓘ

Create repository

5



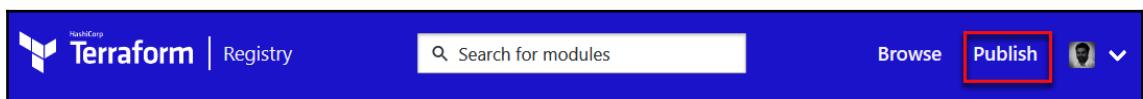
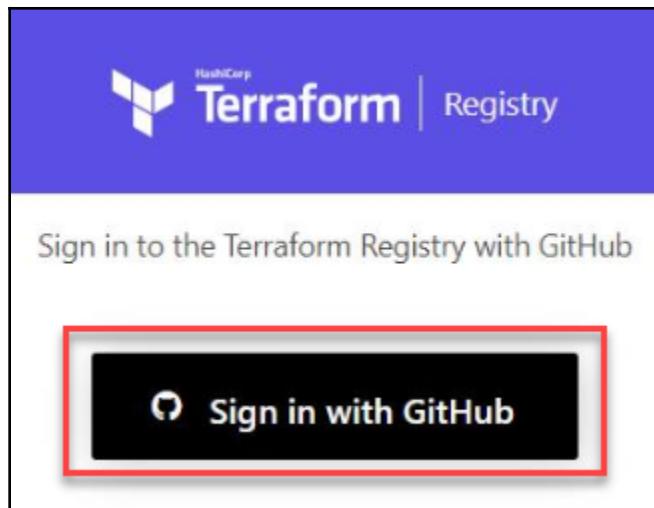
Terraform

| Registry

Search for modules

Browse Publish

Sign-in

A screenshot of the Terraform Registry interface for publishing a module. The top section shows the Terraform logo and "Registry". Below it, a heading says "Select Repository on GitHub" and "How it works". A dropdown menu shows the repository "mikaelkrief/terraform-azurerm-webapp". A "Refresh" button is available. A checkbox labeled "I agree to the Terms of Use." is checked. At the bottom is a large blue "PUBLISH MODULE" button.

The screenshot shows the HashiCorp Terraform Registry interface. At the top, there's a search bar with the placeholder "Search for modules". On the right side of the header, there are "Browse" and "Publish" buttons, along with a user profile icon and a dropdown menu.

The main content area displays a module card for "webapp" by "AZURERM". The card includes:

- A blue icon representing the module.
- The module name: "webapp".
- The provider: "AZURERM".
- A brief description: "Terraform module for provision Azure Service plan, App service (web app) and Application Insight".
- Published date: May 6, 2020, by "mikaelkrief".
- Total provisions: < 100.
- Source: [github.com/mikaelkrief/terraform-azurerm-webapp](https://github.com/mikaelkrief/terraform-azurerm-webapp) (report an issue).

On the right side of the card, there are two buttons: "Manage Module" and "Version 1.0.0". Below these buttons is a "Provision Instructions" section with a code snippet:

```
module "webapp" {  
  source = "mikaelkrief/webapp/azurerm"  
  version = "1.0.0"  
  # insert the 3 required variables here  
}
```

This screenshot is similar to the one above, showing the "webapp" module card. However, a context menu has been opened over the "Manage Module" button. The menu options are:

- Resync Module
- Delete Module Version
- Delete Module Provider
- Delete Module** (highlighted with a pink background)

The "Delete Module" option is currently being selected. To its right, there is a "service (web app)" label. The "Provision Instructions" section and its code snippet are also present on the right side of the card.

```
mikael@LP-FYLZ2X2:~ /Terraform-Cookbook/CHAP05/moduleddemo/Helloworld$ terraform apply  
An execution plan has been generated and is shown below.  
Resource actions are indicated with the following symbols:  
+ create  
Terraform will perform the following actions:  
  
# module.execfile.null_resource.execfile will be created  
+ resource "null_resource" "execfile" {  
    + id = (known after apply)  
}  
  
Plan: 1 to add, 0 to change, 0 to destroy.  
  
Do you want to perform these actions?  
  Terraform will perform the actions described above.  
  Only 'yes' will be accepted to approve.  
  
Enter a value: yes  
  
module.execfile.null_resource.execfile: Creating...  
module.execfile.null_resource.execfile: Provisioning with 'local-exec'...  
module.execfile.null_resource.execfile (local-exec): Executing: ["./bin/bash" "../Modules/execscript/script.sh"]  
module.execfile.null_resource.execfile (local-exec): Hello world  
module.execfile.null_resource.execfile: Creation complete after 0s [id=6621299200818088901]  
  
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

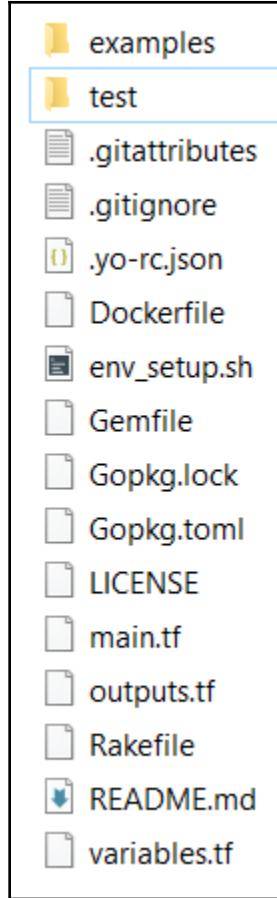
```
module.execfile.null_resource.execfile: Creating...  
module.execfile.null_resource.execfile: Provisioning with 'local-exec'...  
module.execfile.null_resource.execfile (local-exec): Executing: ["./bin/bash" "script.sh"]  
  
Error: Error running command 'script.sh': exec: "/bin/bash": file does not exist. Output:
```

```
PS ~ > yo az-terra-module  
? Terraform module project Name rg  
? The name above already exists on npm, choose another? No  
? What's your name: Mikael Krief  
? Your email (optional): mkrief@.com  
? Your website (optional):  
? Which license do you want to use? MIT  
? Would you like to include the Docker image file? Yes
```

```
PS | > yo az-terra-module
? Terraform module project Name rg
? The name above already exists on npm, choose another? No
? What's your name: Mikael Krief
? Your email (optional): mkrief@████████. com
? Your website (optional):
? Which license do you want to use? MIT
? Would you like to include the Docker image file? Yes
  create LICENSE
  create Dockerfile
  create Gemfile
  create Rakefile
  create env_setup.sh
  create Gopkg.lock
  create Gopkg.toml
  create main.tf
  create outputs.tf
  create variables.tf
  create examples\simple\main.tf
  create examples\simple\outputs.tf
  create .gitattributes
  create .gitignore
  create README.md
  create test\template_test.go
  create test\fixture\main.tf
  create test\fixture\variables.tf
  create test\fixture\outputs.tf
  create test\fixture\terraform.tfvars
Thanks for using module generator for terraform.
```

1

2



```
root@LP-FYLZ2X2:/c/Users/mkrief# curl -L https://github.com/segmentio/terraform-docs/releases/download/v0.9.1/terraform-docs-v0.9.1-linux-amd64 -o terraform-docs-v0.9.1-linux-amd64
% Total    % Received % Xferd  Average Speed   Time     Time   Current
          DLoad  Upload Total Spent   Left Speed
100  644  100  644    0     0  3659      0  --:--:-- 0:00:34  --:--:-- 3659
100 10.6M  100 10.6M   0     0  316k      0  0:00:34  0:00:34  --:--:-- 678k
root@LP-FYLZ2X2:/c/Users/mkrief# tar -xf terraform-docs-v0.9.1-linux-amd64
tar: This does not look like a tar archive
tar: Skipping to next header
tar: Exiting with failure status due to previous errors
root@LP-FYLZ2X2:/c/Users/mkrief# chmod u+x terraform-docs-v0.9.1-linux-amd64
root@LP-FYLZ2X2:/c/Users/mkrief# sudo mv terraform-docs-v0.9.1-linux-amd64 /usr/local/bin/terraform-docs
```

```
PS C:\WINDOWS\system32> choco install terraform-docs -y
Chocolatey v0.10.15
Installing the following packages:
terraform-docs
By installing you accept licenses for the packages.
Error retrieving packages from source 'http://srv-rd-packages.talentsoft.com/nuget/TalentsoftChoco':
Le nom distant n'a pas pu être résolu: 'srv-rd-packages.talentsoft.com'

Terraform-Docs v0.8.2 [Approved]
terraform-docs package files install completed. Performing other installation steps.
Downloading terraform-docs 64 bit
  from 'https://github.com/segmentio/terraform-docs/releases/download/v0.8.2/terraform-docs-v0.8.2-windows-amd64.exe'
Progress: 100% - Completed download of C:\ProgramData\chocolatey\lib\Terraform-Docs\tools\terraform-docs-v0.8.2-windows-amd64.exe (8.71 MB).
Download of terraform-docs-v0.8.2-windows-amd64.exe (8.71 MB) completed.
Hashes match.
C:\ProgramData\chocolatey\lib\Terraform-Docs\tools\terraform-docs-v0.8.2-windows-amd64.exe
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type `refreshenv`).
ShimGen has successfully created a shim for terraform-docs.exe
The install of terraform-docs was successful.
  Software install location not explicitly set, could be in package or
  default install location if installer.

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
```

```
root@LP-FYLZ2X2: /Terraform-Cookbook/CHAP05/moduledemo# terraform-docs --version
terraform-docs version v0.9.1 7f761d3 linux/amd64 BuildDate: 2020-04-02T21:58:38+0000
```

```
root@LP-FYLZ2X2: /Terraform-Cookbook/CHAP05/moduledemo# terraform-docs markdown Modules/webapp/
## Requirements 1
No requirements.

## Providers 2

| Name | Version |
|-----|-----|
| azurerm | n/a |

## Inputs

| Name | Description | Type | Default | Required |
|-----|-----|-----|:-----|
| app\_name | Name of application | `any` | n/a | yes |
| location | Location of Azure resource | `string` | `"West Europe"` | no |
| resource\_groupe\_name | Resource groupe name | `any` | n/a | yes |
| service\_plan\_name | Service plan name | `any` | n/a | yes |

## Outputs
```

## Create new project

X

Project name \*

Terraform-modules



Description

Contain all Terraform Modules

Visibility



Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.



Private

Only people you give access to will be able to view this project.

Advanced

Version control [?](#)

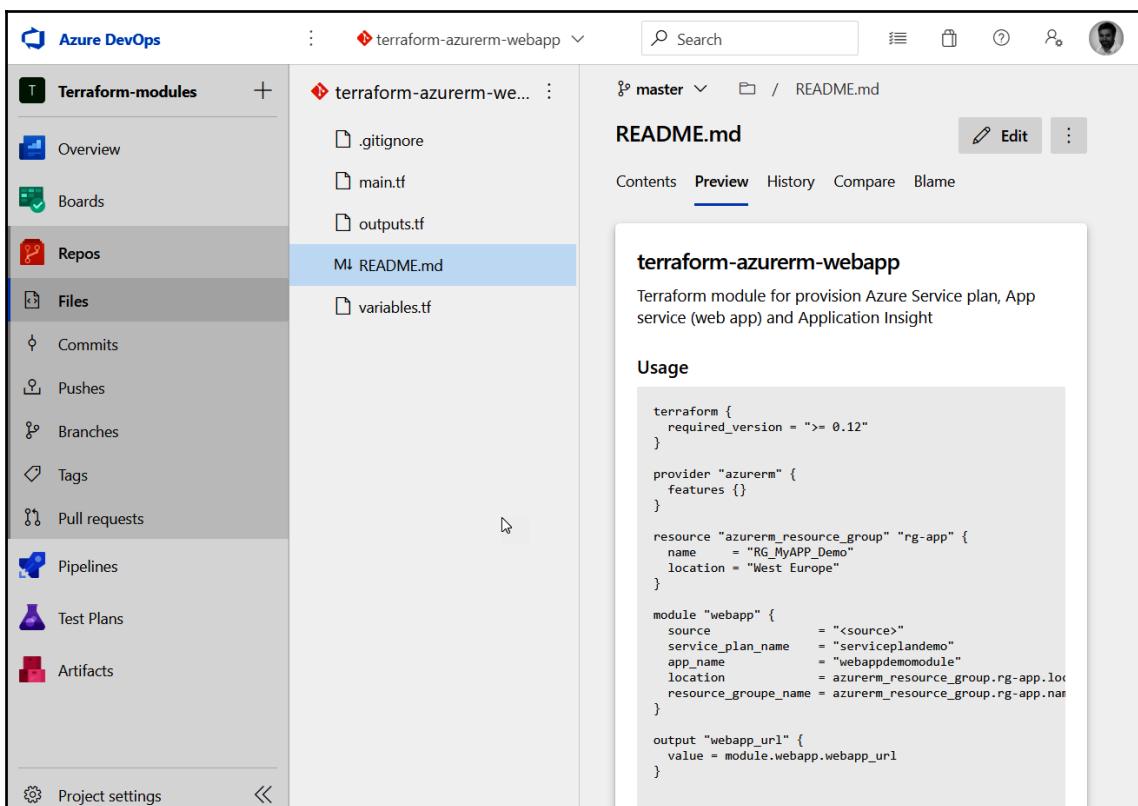
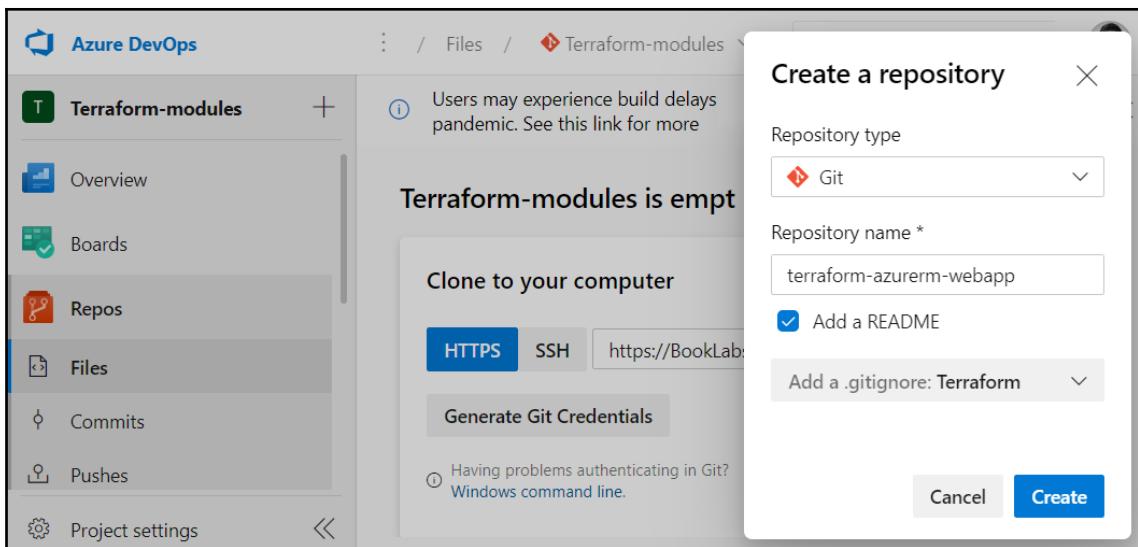
Git

Work item process [?](#)

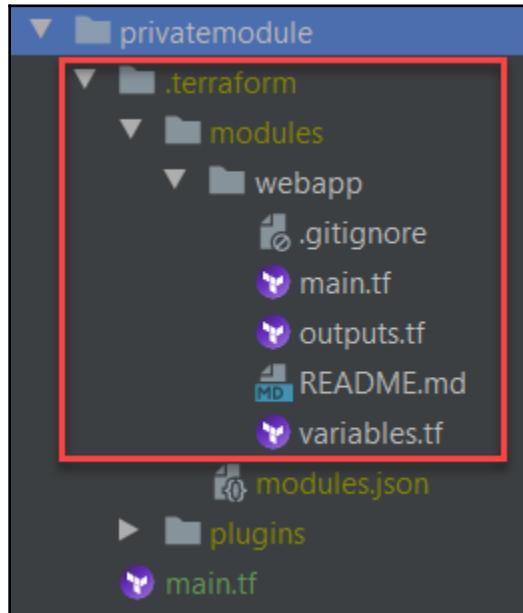
Basic

Cancel

Create



```
PS C:\Terraform-Cookbook\CHAP05\privatemodule> terraform init
Initializing modules...
Downloading git::https://dev.azure.com/BookLabs/Terraform-modules/_git/terraform-azurerm-webapp?ref=v1.0.0 for webapp...
- webapp in .terraform\modules\webapp
```



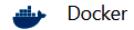
```
root@LP-FYLZ2X2:~/go/src/module/tests# go test -v
== RUN TestTerraformModule
TestTerraformModule 2020-05-19T14:46:29+02:00 retry.go:72: terraform [init -upgrade=false]
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Running command terraform with args [init -upgrade=false] ①
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Initializing modules...
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: - demo in ../..
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: 
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Initializing the backend...
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: 
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Terraform has been successfully initialized!
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: You may now begin working with Terraform. Try running "terraform plan" to see
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: any changes that are required for your infrastructure. All Terraform commands
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: should now work.
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: 
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: If you ever set or change modules or backend configuration for Terraform,
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: rerun this command to reinitialize your working directory. If you forget, other
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: commands will detect it and remind you to do so if necessary.
TestTerraformModule 2020-05-19T14:46:29+02:00 retry.go:72: terraform [get -update]
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Running command terraform with args [get -update]
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: - demo in ../..
TestTerraformModule 2020-05-19T14:46:29+02:00 retry.go:72: terraform [apply -input=false -auto-approve -lock=false]
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Running command terraform with args [apply -input=false -auto-approve -lock=false]
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: 
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Apply complete! Resources: 0 added, 0 changed, 0 destroyed. ②
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: 
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Outputs:
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: 
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: outmodule = This is test of module with TERRATEST
TestTerraformModule 2020-05-19T14:46:29+02:00 retry.go:72: terraform [output -no-color outmodule]
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Running command terraform with args [output -no-color outmodule]
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: This is test of module with TERRATEST ③
TestTerraformModule 2020-05-19T14:46:29+02:00 retry.go:72: terraform [destroy -auto-approve -input=false -lock=false]
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Running command terraform with args [destroy -auto-approve -input=false -lock=false]
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: 
TestTerraformModule 2020-05-19T14:46:29+02:00 logger.go:66: Destroy complete! Resources: 0 destroyed.
--- PASS: TestTerraformModule (0.33s)
PASS
ok    module/tests      0.393s
```

A screenshot of the Azure DevOps Pipelines interface. The left sidebar shows project navigation with 'Terraform-modules' selected. A red circle with the number '1' is on the 'Pipelines' button. The main area features a cartoon illustration of a person working at a laptop, a robot, and a dog. Below the illustration, the text 'Create your first Pipeline' is displayed, followed by the sub-instruction 'Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.' A red circle with the number '2' is on the 'Create Pipeline' button.

A screenshot of the 'Select a repository' step in the pipeline creation wizard. The top navigation bar shows 'Connect', 'Select' (which is highlighted in blue), 'Configure', and 'Review'. The main area is titled 'New pipeline' and contains the heading 'Select a repository'. A search bar at the top right is set to 'Terraform-modules'. Below it, two repositories are listed: 'module-sample' and 'terraform-azurerm-webapp'. A cursor is hovering over the 'module-sample' entry.

New pipeline

## Configure your pipeline



Docker Build a Docker image



Docker

Build and push an image to Azure Container Registry



Deploy to Azure Kubernetes Service

Build and push image to Azure Container Registry; Deploy to Azure Kubernetes Service



Deploy to Kubernetes - Review app with Azure DevSpaces

Build and push image to Azure Container Registry; Deploy to Azure Kuberentes Services and setup Review App with Azure DevSpaces



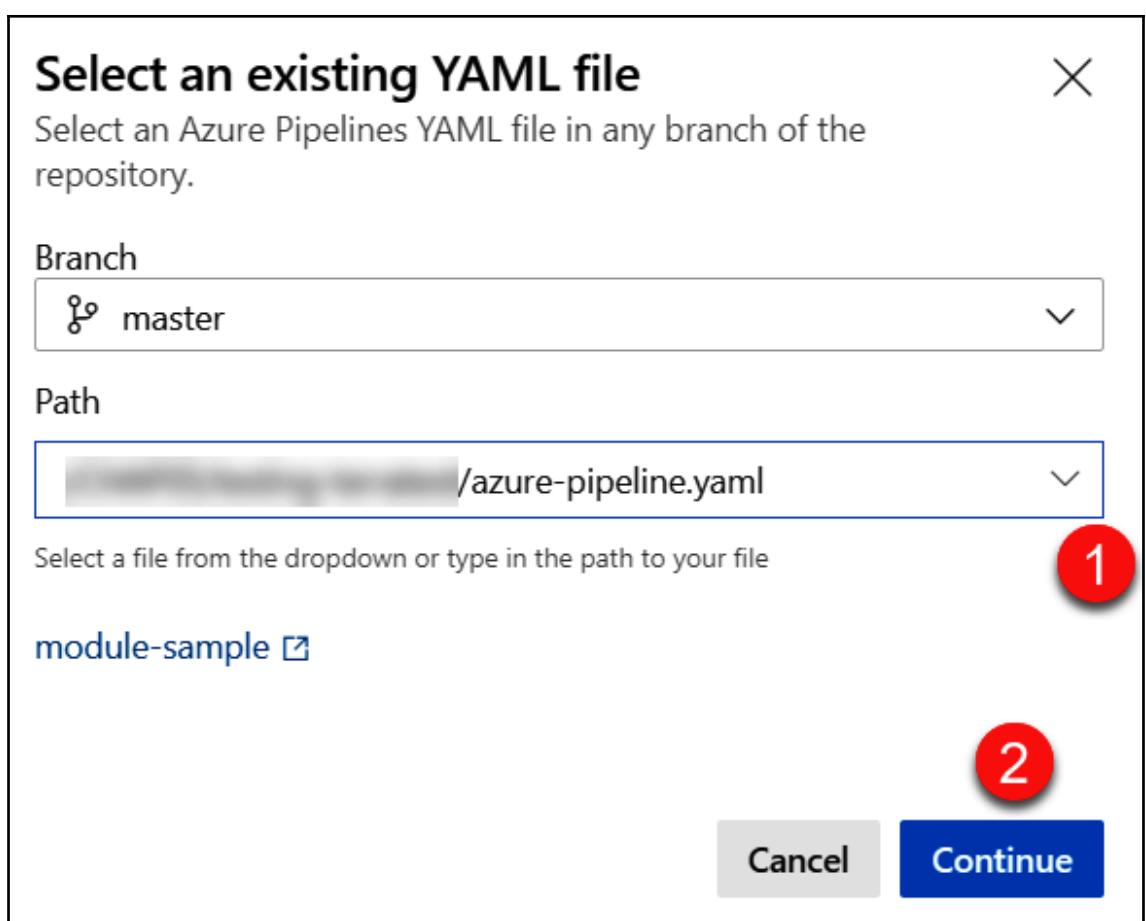
Starter pipeline

Start with a minimal pipeline that you can customize to build and deploy your code.



Existing Azure Pipelines YAML file

Select an Azure Pipelines YAML file in any branch of the repository.



✓ Connect ✓ Select ✓ Configure Review

New pipeline

## Review your pipeline YAML

Variables Run ↗

◆ module-sample / CHAP05/testing-terratest/azure-pipeline.yaml ↗ Show assistant

```
1 trigger:
2 master
```

The screenshot shows the "Review your pipeline YAML" step. It displays the YAML configuration for a pipeline trigger. The YAML code is as follows:

```
trigger:
  master
```

A red box highlights the "Run" button in the top right corner of the review interface.

← Jobs in run #1.0.5

module-sample

Build image

Build	59s
Initialize job	2s
Checkout module-sam...	1s
Docker build	35s
Docker run	18s
Tag code	1s
Post-job: Checkout m...	<1s
Finalize Job	<1s
Report build status	<1s

Docker run

View raw log

```
27 TestTerraformModule 2020-05-20T18:44:50Z logger.go:66: any changes that are required for your infrastructure. All Terraform
28 TestTerraformModule 2020-05-20T18:44:50Z logger.go:66: should now work.
29 TestTerraformModule 2020-05-20T18:44:50Z logger.go:66:
30 TestTerraformModule 2020-05-20T18:44:50Z logger.go:66: If you ever set or change modules or backend configuration for Terraform,
31 TestTerraformModule 2020-05-20T18:44:50Z logger.go:66: rerun this command to reinitialize your working directory. If you forget,
32 TestTerraformModule 2020-05-20T18:44:50Z logger.go:66: commands will detect it and remind you to do so if necessary.
33 TestTerraformModule 2020-05-20T18:44:50Z retry.go:72: terraform [get -update]
34 TestTerraformModule 2020-05-20T18:44:50Z logger.go:66: Running command terraform with args [get -update]
35 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: - demo in ../..
36 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: Running command terraform with args [apply -input=false -auto-approve -lock=false]
37 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: Running command terraform with args [apply -input=false -auto-approve]
38 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: ↵
39 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
40 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66:
41 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: Outputs:
42 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66:
43 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: outmodule - This is test of module with TERRATEST
44 TestTerraformModule 2020-05-20T18:44:51Z retry.go:72: terraform [output -no-color outmodule]
45 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: Running command terraform with args [output -no-color outmodule]
46 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: This is test of module with TERRATEST
47 TestTerraformModule 2020-05-20T18:44:51Z retry.go:72: terraform [destroy -auto-approve -input=false -lock=false]
48 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: Running command terraform with args [destroy -auto-approve -input=false]
49 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66:
50 TestTerraformModule 2020-05-20T18:44:51Z logger.go:66: Destroy complete! Resources: 0 destroyed.
51 --- PASS: TestTerraformModule (0.15s)
52 PASS
53 ok    module-test/tests      0.155s
54
55 Finishing: Docker run
```

The screenshot shows a GitHub repository interface. On the left, a sidebar lists repository navigation options: Overview, Boards, Repos (selected), Files, Commits, Pushes, Branches, Tags (with a red notification badge '1'), and Pull requests. The main area is titled 'Tags' and displays two entries: 'v1.0.5' and 'v1.0.6'. Both entries are highlighted with a red rectangular border.

The screenshot shows the 'Code' tab of the GitHub repository. At the top, there are navigation links: Code (selected), Pull requests, Actions, Projects, Wiki, Security, Insights, and three dots. Below this is a header with a dropdown for the branch ('master'), a 'Go to file' search bar, an 'Add file' button (marked with a red circle '1'), a 'Code' dropdown (marked with a red circle '2'), an 'About' link, and a gear icon. A status message indicates the branch is 4 commits ahead and 2 commits behind 'PacktPublishing:master'. To the right, there is information about the repository: 'Terraform Cookbook by Packt' and a 'Readme' link.

---



### Commit new file

Create integration-tests.yaml

Add an optional extended description...

krief\_mikael@hotmail.com

Choose which email address to associate with this commit

➔ Commit directly to the `master` branch.

⚡ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

**Commit new file** Cancel

1

Code Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Workflows All workflows New workflow

All workflows

Filter workflows

Event Status Branch Actor

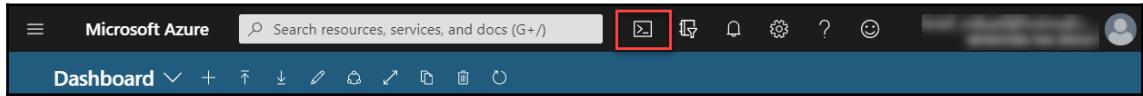
**Create integration-tests.yaml**  
Demo Terraform Module integration test #22: Commit e92b1ba pushed by mikaelkrief

master 28 seconds ago In progress ...

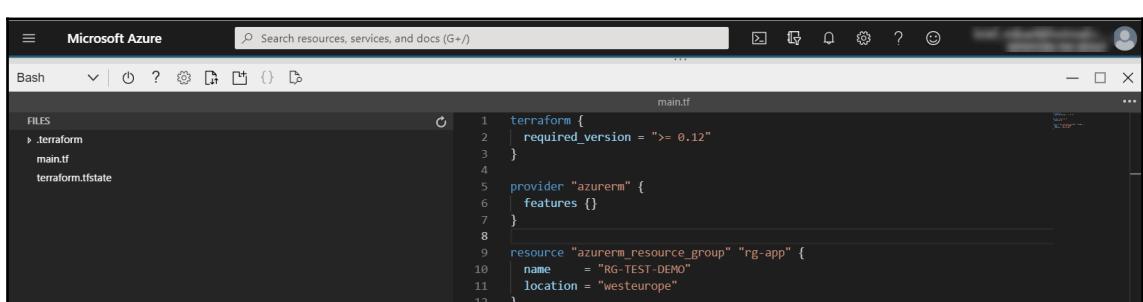
A screenshot of a GitHub repository's Actions page. The pipeline name is "Update integration-tests.yaml". It shows a single job named "Demo Terraform Module integration test / Test Module and publish" which has succeeded 1 minute ago in 1m 34s. The job steps listed are: Set up job, Check out code, Set up Go 1.14, Get Go dependencies, Run Tests, Bump version and push tag, and Complete job.

A screenshot of a GitHub repository's tags page. The current branch is master. A modal window titled "Switch branches/tags" is open, showing a list of tags: v1.0.2, v1.0.1, v1.0.0, v0.0.2, and v0.0.1. The tag v1.0.2 is highlighted with a red box.

# Chapter 6: Provisioning Azure Infrastructure with Terraform



The screenshot shows the Microsoft Azure Cloud Shell interface. The top navigation bar has a search bar and several icons. Below it is a toolbar with icons for file operations like copy, paste, and refresh. The main area is titled "Dashboard" and shows two shell options: "Bash" (selected) and "PowerShell". A message indicates "Cloud Shell.Succeeded. rminal...".

The screenshot shows the Microsoft Azure Cloud Shell interface with the "Bash" tab selected. The main area displays the contents of a "main.tf" file:

```
1 terraform {
2   required_version = ">= 0.12"
3 }
4
5 provider "azurerm" {
6   features {}
7 }
8
9 resource "azurerm_resource_group" "rg-app" {
10   name     = "RG-TEST-DEMO"
11   location = "westeurope"
12 }
```

Below the code editor, the terminal output shows the Terraform plan and execution:

```
# azurerm_resource_group.rg-app will be created
+ resource "azurerm_resource_group" "rg-app" {
+   id      = (known after apply)
+   location = "westeurope"
+   name    = "RG-TEST-DEMO"
}

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

azurerm_resource_group.rg-app: Creating...
azurerm_resource_group.rg-app: Creation complete after 0s [id=/subscriptions/4df82e26-6953-4c19-a43f-777c936a8105/resourceGroups/RG-TEST-DEMO]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
mikael@Azure:~/clouddrive/demotf$
```

```
mikael@Azure:~$ az account list
A few accounts are skipped as they don't have 'Enabled' state. Use '--all' to display them.
[{"cloudName": "AzureCloud",
 "homeTenantId": "REDACTED",
 "id": "1da42ac9-ee3e-4fdbREDACTED",
 "isDefault": false,
 "managedByTenants": [],
 "name": "DEMO",
 "state": "Enabled",
 "tenantId": "REDACTED",
 "user": {"cloudShellID": true,
 "name": "live.com#REDACTED",
 "type": "user"}},
```

The screenshot shows the Microsoft Azure Cloud Shell interface. At the top, there's a search bar and a toolbar with various icons. Below that is a dashboard header with a 'Bash' dropdown and other navigation links. The main area contains a terminal window with the following command and output:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...
mikael@Azure:~$ az ad sp create-for-rbac --name="BookDemoTerraform" --role="Contributor" --scopes="/subscriptions/1da42ac9-REDACTED"
Changing "BookDemoTerraform" to a valid URI of "http://BookDemoTerraform", which is the required format used for service principal names
Creating a role assignment under the scope of "/subscriptions/1da42ac9-ee3e-4fdb-b294-f7a607f589d5"
  Retrying role assignment creation: 1/36
  Retrying role assignment creation: 2/36
{
  "appId": "282936c6-298d-4bbc-890b-REDACTED" . 1
  "displayName": "BookDemoTerraform",
  "name": "http://BookDemoTerraform",
  "password": "62e36b75-16aa-4c33-8fd9-REDACTED" . 2
  "tenant": "2e3a33f9-66b1-4e2a-8b95-REDACTED" . 3
}
mikael@Azure:~$
```

The output shows the creation of a service principal named 'BookDemoTerraform' with a contributor role assigned to it. The terminal window has a red box highlighting the JSON object returned by the command.

The screenshot shows the Microsoft Azure portal's 'Access control (IAM)' blade for a subscription named 'DEMO'. The left sidebar includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Security', 'Events', 'Cost Management', and 'Cost analysis'. The 'Access control (IAM)' section is selected.

The main area displays the following information:

- Number of role assignments for this subscription:** 8 / 2000
- Filtering:** Name (Search by name or email), Type (All), Role (Contributor), Scope (All scopes)
- Role assignments:** 5 items (3 Service Principals, 2 Unknown)
 

Name	Type	Role	Scope
BookDemoTerraform	App	Contributor	All scopes

Microsoft Azure  ...

Dashboard > DEMO | Resource groups > RG-DEMO >

## AppDemoARM | Extensions

App Service

Search (Ctrl+/  
Console  
Advanced Tools  
App Service Editor (Preview)  
Resource explorer  
Extensions

Add

### Extensions

Extensions add functionality to your App Service. Click add to see the list of available extensions.

Name	Version	Update Available
ASP.NET Core 2.2 (x64) Runtime	2.2.0-preview3-35497	No

```
mikael@Azure:~$ az keyvault secret show -n ConnectionStringApp --vault-name keyvdemobook
{
  "attributes": {
    "created": "2020-06-03T09:41:33+00:00",
    "enabled": true,
    "expires": null,
    "notBefore": null,
    "recoveryLevel": "Recoverable+Purgeable",
    "updated": "2020-06-03T09:41:33+00:00"
  },
  "contentType": null,
  "id": "https://keyvdemobook.vault.azure.net/secrets/ConnectionStringApp/af0164c179e647b3952de55b63e4308b",
  "kid": null,
  "managed": null,
  "name": "ConnectionStringApp",
  "tags": null,
  "value": "Data Source=mysever.com;initial catalog=databasedemo;User ID=useradmin;Password=demobook"
}
```

Microsoft Azure  Search resources, services, and docs (G+)

Dashboard > keyvdemobook | Overview > keyvdemobook | Access policies >

## Add access policy

Add access policy

Configure from template (optional)

Key permissions  0 selected

Secret permissions  2 selected 1

Certificate permissions  0 selected

Select principal   
BookDemoTerraform 2

Authorized application  None selected

**Add** 3

```

PS C:\CHAP06\keyvault> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.azurem_key_vault.keyvault: Refreshing state...
data.azurem_key_vault_secret.app-connectionstring: Refreshing state...

Error: Error making Read request on Azure KeyVault Secret secret-sauce: keyvault.BaseClient#GetSecret: Failure respondin
g to request: StatusCode=403 -- Original Error: autorest/azure: Service returned an error. Status=403 Code="Forbidden" M
essage="The user, group or application 'appid=282936c6-298d-40bc-896b-0906b7d9c107;oid=e7f17718-47bd-40fb-8e96-887f20cad
f79;iss=https://sts.windows.net/2e3a33f9-66b1-4e2a-8b95-74102ad857c2/' does not have secrets get permission on key vault
'keydemobook';location='westeurope'. For help resolving this issue, please see https://go.microsoft.com/fwlink/?linkid=2
125287" InnerError={"code":"AccessDenied"}

on main.tf line 17, in data "azurerm_key_vault_secret" "app-connectionstring":
17: data "azurerm_key_vault_secret" "app-connectionstring" {

```

Microsoft Azure

Dashboard > keydemobook | Overview > keydemobook > DEMO | Resource groups > RG-DEMOVAULT >

## demovaultbook | Configuration

App Service

Search (Ctrl+ /) Refresh Save Discard

**Tags**

**Diagnose and solve problems**

**Security**

**Events**

**Deployment**

Quickstart

Deployment slots

Deployment Center

**Settings**

Configuration (1)

**Connection strings**

Connection strings are encrypted at rest and transmitted over an encrypted channel.

New connection string Hide values Advanced edit Filter

Name	Value	Source
WEBSITE_NODE_DEFAULT_VERSION	Hidden value. Click to show value	App Config

**Connection strings**

Name	Type	Deployment...
Database	SQLServer	

```
"mode": "data",
"type": "azurerm_key_vault_secret",
"name": "app-connectionstring",
"provider": "provider.azurerm",
"instances": [
{
    "schema_version": 0,
    "attributes": {
        "content_type": "",
        "id": "https://keyvdemobook.vault.azure.net/secrets/ConnectionStringApp/af0164c179e647b3952de55b63e4308b",
        "key_vault_id": "/subscriptions/1da42ac9-ee3e-4fdb-b294-f7a607f589d5/resourceGroups/rg_key_vault/providers/Microsoft.KeyVault/vaults/keyvdemobook",
        "name": "ConnectionStringApp",
        "tags": {},
        "timeouts": null,
        "value": "Data Source=mysever.com;initial catalog=database;User ID=useradmin;Password=1234567890",
        "version": "af0164c179e647b3952de55b63e4308b"
    }
}
```

```
PS C:\terraform-Cookbook\CHAP06\keyvault> terraform show
# azurerm_app_service:
resource "azurerm_app_service" "app" {
    app_service_plan_id      = "/subscriptions/1da42ac9-ee3e-4fdb-b294-f7a607f589d5/resourceGroups/RG-DEMOVULT/providers/Microsoft.Web/serverfarms/SP-demovault"
    app_settings              = {
        "WEBSITE_NODE_DEFAULT_VERSION" = "6.9.1"
    }
    client_affinity_enabled     = true

    connection_string {
        name   = "Database"
        type   = "SQLServer"
        value  = (sensitive value) ←
    }
}
```

```
PS C:\CHAP06\listresources> terraform output
nsg = {
  "NSG1" = {
    "id" = "/subscriptions/1da42ac9-ee3e-4fdb-b294-f7a607f589d5/resourceGroups/RG-DEMO/providers/Microsoft.Network/networkSecurityGroups/NSG1"
    "location" = "westeurope"
    "name" = "NSG1"
    "tags" = {
      "DEFAULTRULES" = "TRUE"
    }
    "type" = "Microsoft.Network/networkSecurityGroups"
  }
  "NSG2" = {
    "id" = "/subscriptions/1da42ac9-ee3e-4fdb-b294-f7a607f589d5/resourceGroups/RG-DEMO/providers/Microsoft.Network/networkSecurityGroups/NSG2"
    "location" = "westeurope"
    "name" = "NSG2"
    "tags" = {
      "DEFAULTRULES" = "TRUE"
    }
    "type" = "Microsoft.Network/networkSecurityGroups"
  }
}
```

```
mikael@LP-FYLZ2X2:/c/test/terraform$ terraformer --help
Usage:
  [command]

Available Commands:
  help      Help about any command
  import    Import current state to Terraform configuration
  plan      Plan to import current state to Terraform configuration
  version   Print the version number of Terraformer

Flags:
  -h, --help      help for this command
  --version      version for this command

Use " [command] --help" for more information about a command.
```

<input type="text"/> Filter by name...	Location == <b>all</b> <span style="color: red;">X</span>	<span style="color: blue;">+ </span> Add filter
Showing 1 to 8 of 8 records.		
<input type="checkbox"/> <b>Name ↑↓</b>		
<input type="checkbox"/> DefaultResourceGroup-WEU		
<input type="checkbox"/> MyRgRemoteBackend		
<input type="checkbox"/> NetworkWatcherRG		
<input type="checkbox"/> RG-DEMO		
<input type="checkbox"/> RG-DEMOVAULT		
<input type="checkbox"/> RG-VM		
<input type="checkbox"/> rg_keyvault		
<input type="checkbox"/> rg_vmdev		

```
mikael@LP-FVLZ2X2:/c/test/terraformer/generated/azurerm$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.terraform_remote_state.local: Refreshing state...
azurerm_resource_group.tfer--DefaultResourceGroup-002D-WEU: Refreshing state...
[truncated]
azurerm_resource_group.tfer--RG-002D-DEMO: Refreshing state... [id=/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/RG-002D-DEMO]
azurerm_resource_group.tfer--MyRgRemoteBackend: Refreshing state... [id=/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/MyRgRemoteBackend]
azurerm_resource_group.tfer--rg_vmdev: Refreshing state... [id=/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/rg_vmdev]
azurerm_resource_group.tfer--RG-002D-DEMOVAULT: Refreshing state... [id=/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/RG-002D-DEMOVAULT]
azurerm_resource_group.tfer--RG-002D-VM: Refreshing state... [id=/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/RG-002D-VM]
azurerm_resource_group.tfer--rg_keyvault: Refreshing state... [id=/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/rg_keyvault]
azurerm_resource_group.tfer--NetworkWatcherRG: Refreshing state... [id=/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/NetworkWatcherRG]

No changes. Infrastructure is up-to-date.
```

```
mikael@LP-FYLZ2X2:/c/test/terraform$ terraformer import azure --resources=resource_group --compact --path-pattern '{output}/{provider}/'
2020/06/16 18:25:25 Testing if Service Principal / Client Certificate is applicable for Authentication..
2020/06/16 18:25:25 Testing if Multi Tenant Service Principal / Client Secret is applicable for Authentication..
2020/06/16 18:25:25 Testing if Service Principal / Client Secret is applicable for Authentication..
2020/06/16 18:25:25 Using Service Principal / Client Secret for Authentication
2020/06/16 18:25:25 Getting OAuth config for endpoint https://Login.microsoftonline.com/ with tenant 2e3a33f9-XXXXXXXXXX
2020/06/16 18:25:25 azurerm importing... resource_group
2020/06/16 18:25:28 Refreshing state... azurerm_resource_group.tfer--rg_keyvault
2020/06/16 18:25:28 Refreshing state... azurerm_resource_group.tfer--RG-002D-DEMO
2020/06/16 18:25:28 Refreshing state... azurerm_resource_group.tfer--RG-002D-DEMOVULT
2020/06/16 18:25:28 Refreshing state... azurerm_resource_group.tfer--RG-002D-VM
2020/06/16 18:25:28 Refreshing state... azurerm_resource_group.tfer--rg_vmdev
2020/06/16 18:25:28 Refreshing state... azurerm_resource_group.tfer--DefaultResourceGroup-002D-WEU
2020/06/16 18:25:28 Refreshing state... azurerm_resource_group.tfer--MyRgRemoteBackend
2020/06/16 18:25:28 Refreshing state... azurerm_resource_group.tfer--NetworkWatcherRG
2020/06/16 18:25:29 azurerm Connecting....
2020/06/16 18:25:29 azurerm save
2020/06/16 18:25:29 azurerm save tfstate
```

 outputs.tf	16/06/2020 14:01	Fichier TF	1 Ko
 provider.tf	16/06/2020 15:40	Fichier TF	1 Ko
 resources.tf	16/06/2020 14:01	Fichier TF	1 Ko
 terraform.tfstate	16/06/2020 14:01	Fichier TFSTATE	10 Ko
 variables.tf	16/06/2020 14:01	Fichier TF	1 Ko

# Chapter 7: Deep Diving into Terraform

```
Terraform has been successfully initialized!
$$$$$ Finished initializing the Terraform working directory.
$$$$$ Creating the kitchen-terraform-kt-suite-terraform Terraform workspace...
      Created and switched to workspace "kitchen-terraform-kt-suite-terraform"!

$$$$$ Downloading the modules needed for the Terraform configuration...
$$$$$ Finished downloading the modules needed for the Terraform configuration.
$$$$$ Validating the Terraform configuration files...
      Success! The configuration is valid.

$$$$$ Finished validating the Terraform configuration files.
$$$$$ Building the infrastructure based on the Terraform configuration...
local_file.inventory: Creating...
local_file.inventory: Creation complete after 0s [id=c8902b89314e559cd2281e0f9b381677c7a10b16]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

```
-----> Verifying <kt-suite-terraform>...
$$$$$ Reading the Terraform input variables from the Kitchen instance state...
$$$$$ Finished reading the Terraform input variables from the Kitchen instance state.
$$$$$ Reading the Terraform output variables from the Kitchen instance state...
$$$$$ Finished reading the Terraform output variables from the Kitchen instance state.
$$$$$ Verifying the systems...
$$$$$ Verifying the 'basic' system...

Profile: default
Version: (not specified)
Target: local://

[PASS] check_inventory_file: File ./inventory
      [PASS] File ./inventory is expected to exist
      [PASS] File ./inventory size is expected to be > 0

Profile Summary: 1 successful control, 0 control failures, 0 controls skipped
Test Summary: 2 successful, 0 failures, 0 skipped
```

```
Terraform has been successfully initialized!
$$$$$ Finished initializing the Terraform working directory.
$$$$$ Selecting the kitchen-terraform-kt-suite-terraform Terraform workspace...
$$$$$ Finished selecting the kitchen-terraform-kt-suite-terraform Terraform workspace.
$$$$$ Destroying the Terraform-managed infrastructure...
local_file.inventory: Refreshing state... [id=c8902b89314e559cd2281e0f9b381677c7a10b16]
local_file.inventory: Destroying... [id=c8902b89314e559cd2281e0f9b381677c7a10b16]
local_file.inventory: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```

```

# azurerm_application_insights.appinsight-app must be replaced ←
-/+ resource "azurerm_application_insights" "appinsight-app" {
    ~ app_id                      = "4440e9e2-8558-4165-a772-b35942ee0a96" → (known after apply)
    ~ application_type             = "web"
    ~ daily_data_cap_in_gb        = 100 → (known after apply)
    ~ daily_data_cap_notifications_disabled = false → (known after apply)
    ~ disable_ip_masking          = false
    ~ id                           = "/subscriptions/1da42ac9-ee3e-11ea-8d0c-000000000000/resourceGroups/RG-A
pp-DEV1/providers/microsoft.insights/components/MyApp-DEV1" → (known after apply)
    ~ instrumentation_key          = (sensitive value)
    ~ location                     = "westeurope"
    ~ name                         = "MyApp-DEV1" → "MyApp2-DEV1" # forces replacement
    ~ resource_group_name          = "RG-App-DEV1"
    retention_in_days              = 90
    sampling_percentage            = 100
    tags                          = {
        "CreatedBy" = "NA"
        "ENV"       = "DEV1"
    }
}

Plan: 2 to add, 0 to change, 2 to destroy.

```

```

PS C:\CHAP07\preventdestroy> terraform apply
azurerm_resource_group.rg-app: Refreshing state... [id=/subscriptions/1da42ac9-ee3e-11ea-8d0c-000000000000/resourceGroups/RG-App-DEV1]
azurerm_app_service_plan.plan-app: Refreshing state... [id=/subscriptions/1da42ac9-ee3e-11ea-8d0c-000000000000/resourceGroups/RG-App-DEV1/providers/Microsoft.Web/serverfarms/Plan-App-DEV1]
azurerm_application_insights.appinsight-app: Refreshing state... [id=/subscriptions/1da42ac9-ee3e-11ea-8d0c-000000000000/resourceGroups/RG-App-DEV1/providers/microsoft.insights/components/MyApp-DEV1]
azurerm_app_service.app: Refreshing state... [id=/subscriptions/1da42ac9-ee3e-11ea-8d0c-000000000000/resourceGroups/RG-A
pp-DEV1/providers/Microsoft.Web/sites/MyApp-test-DEV1]

Error: Instance cannot be destroyed

on main.tf line 47:
  47: resource "azurerm_application_insights" "appinsight-app" {

Resource azurerm_application_insights.appinsight-app has
lifecycle.prevent_destroy set, but the plan calls for this resource to be
destroyed. To avoid this error and continue with the plan, either disable
lifecycle.prevent_destroy or reduce the scope of the plan using the -target
flag.

```

```

PS C:\CHAP07\preventdestroy> terraform apply

Error: Variables not allowed

on main.tf line 59, in resource "azurerm_application_insights" "appinsight-app":
  59:   prevent_destroy = var.prevent_destroy_ai

Variables may not be used here.

Error: Unsuitable value type

on main.tf line 59, in resource "azurerm_application_insights" "appinsight-app":
  59:   prevent_destroy = var.prevent_destroy_ai

Unsuitable value: value must be known

```

```

# azurerm_app_service.app must be replaced ←
-/+ resource "azurerm_app_service" "app" {
  app_service_plan_id = "/subscriptions/1da42ac9-ee3e-4fdb-XXXXXXXXXX/resourceGroups/RG-App-DEV1/providers/Microsoft.Web/serverFarms
/Plan-App-DEV1"
  ~ app_settings = {
    - "WEBSITE_NODE_DEFAULT_VERSION" = "6.9.1"
    } -> (Known after apply)
  ~ client_affinity_enabled = true -> (Known after apply)
  - client_cert_enabled = false -> null
  ~ default_site_hostname = "myappdemo-dev1.azurewebsites.net" -> (Known after apply)
  enabled = true
  https_only = false
  ~ id = "/subscriptions/1da42ac9-ee3e-4fdb-XXXXXXXXXX/resourceGroups/RG-App-DEV1/providers/Microsoft.Web/sites/MyApp
Demo-DEV1" -> (known after apply)
  location = "westeurope"
  ~ name = "MyAppDemo-DEV1" -> "MyAppDemo-DEV1-demo" # forces replacement
}

```

```

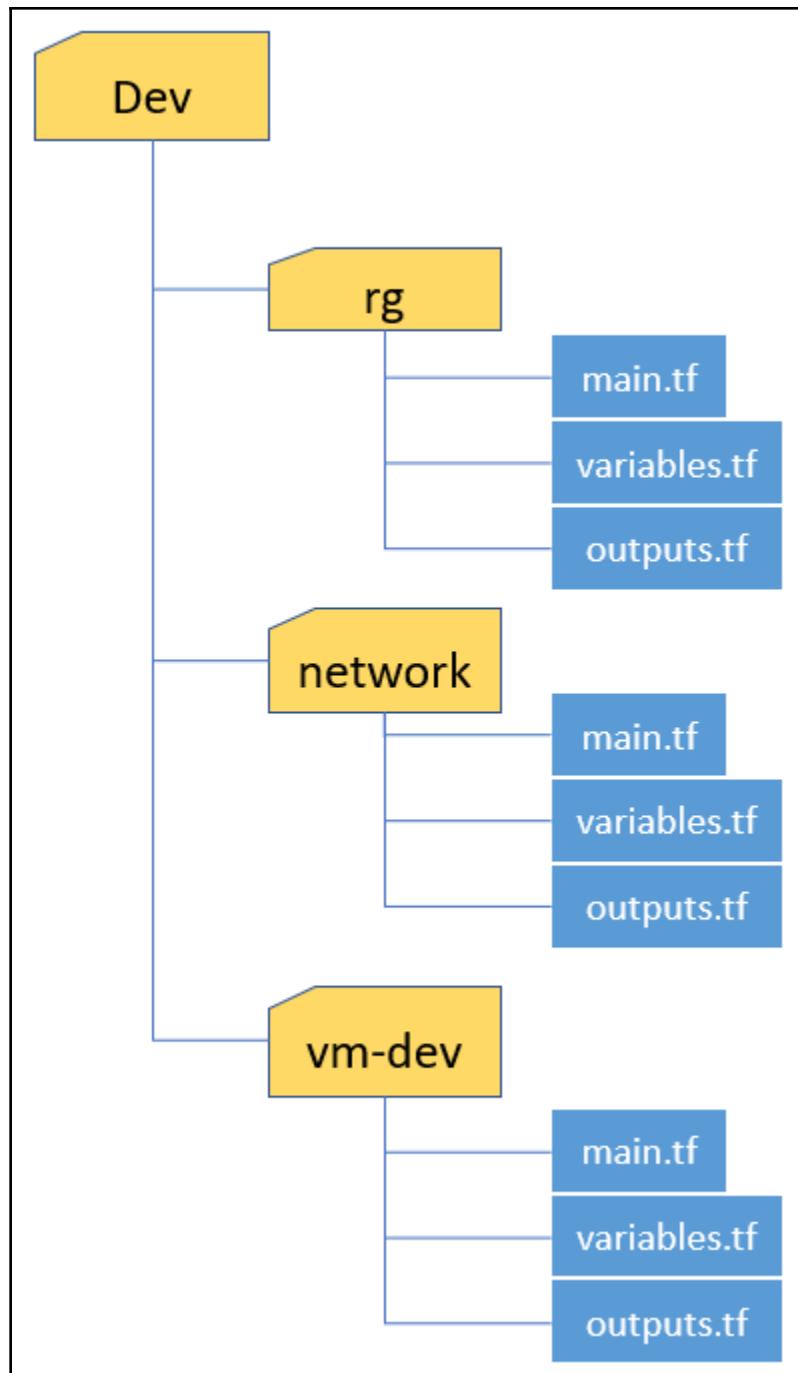
azurerm_app_service.app: Creating... ①
azurerm_app_service.app: Still creating... [10s elapsed]
azurerm_app_service.app: Still creating... [20s elapsed]
azurerm_app_service.app: Creation complete after 28s [id=/subscriptions/1da42ac9-ee3e-4fdb-XXXXXXXXXX/resourceGroups/RG-App-DEV1/providers/Microsoft.Web/sites/MyAppDemo-DEV1]
azurerm_app_service.app: Destroying... [id=/subscriptions/1da42ac9-ee3e-4fdb-XXXXXXXXXX/resourceGroups/RG-App-DEV1/providers/Microsoft.Web/sites/MyAppDemo-DEV1] ②
azurerm_app_service.app: Destruction complete after 2s
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.

```

```

PS \terraform-Cookbook\CHAP07\detectdestroy> $tfplan = terraform show -json tfout.tfplan
PS \terraform-Cookbook\CHAP07\detectdestroy> $actions = $tfplan | jq.exe .resource_changes[].change.actions[]
PS \terraform-Cookbook\CHAP07\detectdestroy> $nbdelete = $actions -match 'delete' | Measure-Object | Select-Object Count
PS \terraform-Cookbook\CHAP07\detectdestroy> Write-Host $nbdelete.Count
1 ←

```



```
PS C:\CHAP07\demogrunt-wrapper> terragrunt init
--version
[terragrunt] 2020/06/29 16:06:10 Reading Terragrunt config file at /CHAP07/demogrunt-wrapper/terragrunt.hcl
[terragrunt] 2020/06/29 16:06:10 Running command: terraform init --backend-config backend.tfvars

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

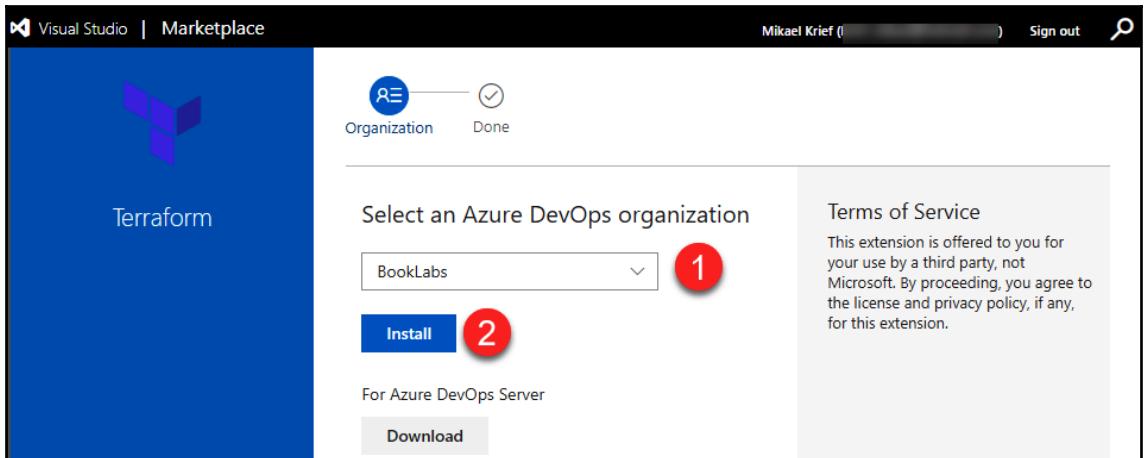
To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.azurerm: version = "~> 2.16"

Terraform has been successfully initialized!

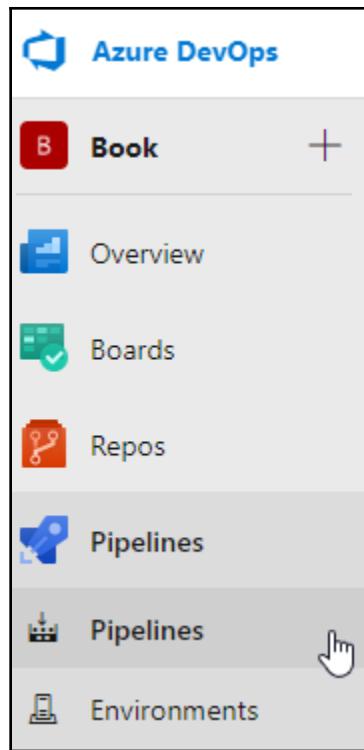
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
```

```
PS C:\CHAP07\demogrunt-wrapper> terragrunt plan
[terragrunt] [ ] \CHAP07\demogrunt-wrapper> 2020/06/29 16:06:18 Running command: terraform --version
[terragrunt] 2020/06/29 16:06:19 Reading Terragrunt config file at /CHAP07/demogrunt-wrapper/terragrunt.hcl
[terragrunt] 2020/06/29 16:06:19 Running command: terraform plan -var-file env-vars.tfvars
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
```



The screenshot shows the Azure DevOps interface for a project named 'DevOpsOpenSource'. The left sidebar contains various project settings like General, Overview, Teams, Permissions, Notifications, Service hooks, Dashboards, Boards, Project configuration, Team configuration, GitHub connections, Repos, Pipelines, and Test management. A red box highlights the 'Service connections\*' option under the Pipelines section.

The main area displays the 'Azure Terraform Demo' service connection details. The 'Overview' tab is selected. The 'Service connection type' is set to 'Azure Resource Manager using service principal authentication'. The 'Scope Level' is set to 'Subscription'. The 'Subscription Id' is '8a7aace5-74aa-4'. The 'Subscription Name' is 'Microsoft Azure Sponsorship'. The 'Authentication' section shows the 'Service Principal Id' as 'ee7f75a0-8553-4e6a-98dd-2a55346d167a'. The 'Credential' section shows 'Service principal key' selected. The 'Service principal key' field contains '\*\*\*\*\*'. At the bottom right of the modal are 'Learn more', 'Cancel', and 'Verify and save' buttons.

A graphic illustration featuring a purple robot holding a blue rocket, a person in a red shirt working on a laptop, and a small brown dog. They are set against a background of clouds and a blue sky.

**Create your first Pipeline**

Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.

**Create Pipeline**

BookLabs / BookDemo / Pipelines

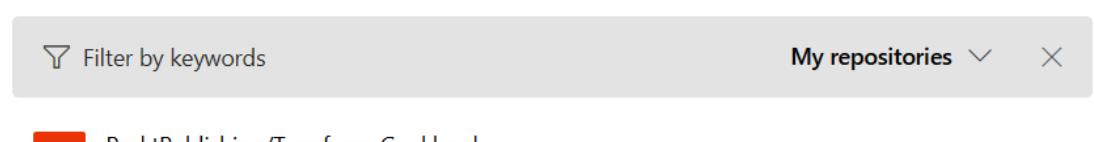
✓ Connect    **Select**    Configure    Review

New pipeline

## Select a repository

Filter by keywords    My repositories

Packt Publishing/Terraform-Cookbook



Azure DevOps : Search

✓ Connect    ✓ Select    ✓ Configure    **Review**

New pipeline

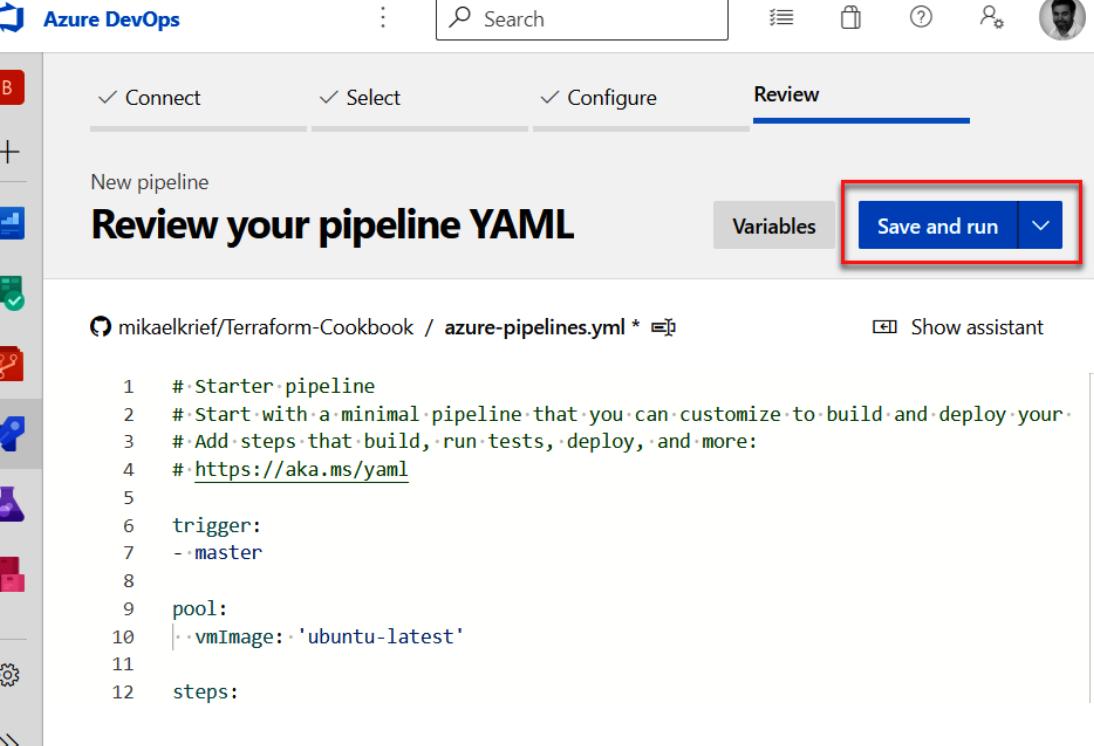
## Review your pipeline YAML

Variables    **Save and run**

mikaelkrief/Terraform-Cookbook / azure-pipelines.yml \* ↗

Show assistant

```
1 # Starter pipeline
2 # Start with a minimal pipeline that you can customize to build and deploy your
3 # Add steps that build, run tests, deploy, and more:
4 # https://aka.ms/yaml
5
6 trigger:
7 - master
8
9 pool:
10   vmImage: 'ubuntu-latest'
11
12 steps:
```



The screenshot shows a CI/CD pipeline interface with a sidebar on the left containing icons for BookLabs, BookDemo, Pipelines, and the current run (mikaelkrieger.Terraform-Cook... / 20200624.21). The main area displays a job log for 'terraform init'. The log output is as follows:

```
1 Starting: terraform init
2 -----
3 Task      : Terraform CLI
4 Description : Execute terraform cli commands
5 Version   : 0.5.12
6 Author    : Charles Zipp
7 Help     :
8 -----
9 /opt/hostedtoolcache/terraform/0.12.24/x64/terraform version
10 Terraform v0.12.24
11
12 Your version of Terraform is out of date! The latest version
13 is 0.12.26. You can update by downloading from https://www.terraform.io/downloads.html
14
15 /opt/hostedtoolcache/terraform/0.12.24/x64/terraform init -backend-config=storage_account_name=storagefbbackendbook -backend-config=container_name=tfsstate -backend=t
16
17 Initializing the backend...
18
19 Successfully configured the backend "azurerm"! Terraform will automatically
20 use this backend unless the backend configuration changes.
```

The screenshot shows a 'Variables' management interface. At the top, there is a search bar labeled 'Search variables' and a '+' button. Below the search bar, there are four listed variables:

- ARM\_TENANT\_ID**: Value = 2e3a33f9-66b1-[REDACTED]
- ARM\_CLIENT\_SECRET**: Value = [REDACTED] (represented by a series of asterisks)
- ARM\_CLIENT\_ID**: Value = 282936c6-298d-[REDACTED]
- ARM\_SUBSCRIPTION\_ID**: Value = 1da42ac9-ee3e-4fdb-[REDACTED]

## Run pipeline

X

Select parameters below and manually run the pipeline

Branch/tag

1

master



dev1



master

## Advanced options

### Variables

This pipeline has no defined variables



### Stages to run

Run as configured



### Resources

Use latest version of all resources



Enable system diagnostics

Cancel

Run

2

← Jobs in run #20200702....  
mikaelkrief.Terraform-Cookbook

Jobs

	Job	26s
✓	Initialize job	2s
✓	Checkout mikaelkrief/T...	2s
✓	Install Terraform 0.12.28	2s
✓	terraform init	4s
✓	Manage Workspaces	5s
✓	terraform plan	4s
✓	terraform apply	4s

Manage Workspaces

```
1 Starting: Manage Workspaces
2 =====
3 Task      : PowerShell
4 Description : Run a PowerShell script on Linux, macOS, or Windows
5 Version   : 2.170.1
6 Author    : Microsoft Corporation
7 Help      : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/powershell
8 -----
9 Generating script.
10 Formatted command: . '/home/vsts/work/1/s/CHMP07/pipeline/ManageWorkspaces.ps1' dev1
11 ====== Starting Command Output ======
12 /usr/bin/pwsh -NoLogo -NoProfile -NonInteractive -Command . '/home/vsts/work/_temp/2c4b7571-2026-4b6c-9575-a5727cee36e7.ps1'
13 Create new Workspace dev1
14 Created and switched to workspace "dev1"!
15
16 You're now on a new, empty workspace. Workspaces isolate their state,
17 so if you run "terraform plan" Terraform will not see any existing state
18 for this configuration.
19
```



# Chapter 8: Using Terraform Cloud to Improve Collaboration

The screenshot shows the 'General Settings' page for a workspace named 'demo-app'. At the top right, there are tabs for 'Runs', 'States', and 'Settings', with 'Settings' being the active tab and having a red circle with the number '1' above it. Below the tabs, the page title is 'General Settings'. There are two sections: 'ID' and 'Execution Mode'. In the 'ID' section, the ID is 'ws-SQR1o56Rw6aYVhuP' with a copy icon. In the 'Name' section, the name is 'demo-app'. Under 'Execution Mode', the 'Local' option is selected (radio button checked) and has a red circle with the number '2' above it. A tooltip for 'Local' states: 'Your plans and applies occur on machines you control. Terraform Cloud is only used to store and synchronize state.' Below these sections is a blue 'Save settings' button with a red circle containing the number '3' to its right.

The screenshot shows the 'API Tokens' page for an organization named 'demoBook'. At the top right, there are tabs for 'Workspaces', 'Modules', and 'Settings', with 'Settings' being the active tab and having a red circle with the number '1' above it. Below the tabs, the page title is 'API Tokens'. On the left, there is a sidebar with 'ORGANIZATION SETTINGS' and a list of items: 'demoBook', 'General', 'Plan & Billing', 'Users', 'Teams', 'API Tokens' (which is highlighted with a red circle containing the number '2'), and 'Authentication'. The main content area is titled 'User Tokens' with a sub-section 'Team Tokens'. A tooltip for 'User Tokens' states: 'A user API token has the same permission level as your user account. It is the only type of token which can be granted access to multiple organizations. Generate user API tokens on your [user settings page](#)'. A tooltip for 'Team Tokens' states: 'Team API tokens are used by services, for example a CI/CD pipeline, to perform plans and applies on a workspace. To generate a team API token, go to the [Teams](#) page in your organization settings, select a'.

demo-app ⓘ

Runs States Settings

New state #sv-3F6fDtPtePHaTWrW  
MikaelKrief triggered from Terraform 16 minutes ago



```
PS C:\CHAP08\app> terraform login
Terraform will request an API token for app.terraform.io using your browser.

If login is successful, Terraform will store the token in plain text in
the following file for use by subsequent commands:
  C:\Users\mkrief\AppData\Roaming\terraform.d\credentials.tfrc.json

Do you want to proceed? (y/n) y
Terraform must now open a web browser to the tokens page for app.terraform.io.

If a browser does not open this automatically, open the following URL to proceed:
  https://app.terraform.io/app/settings/tokens?source=terraform-login
```

---

Generate a token using your browser, and copy-paste it into this prompt.

Terraform will store the token in plain text in the following file for use by subsequent commands:  
C:\Users\mkrief\AppData\Roaming\terraform.d\credentials.tfrc.json

---

Token for app.terraform.io:

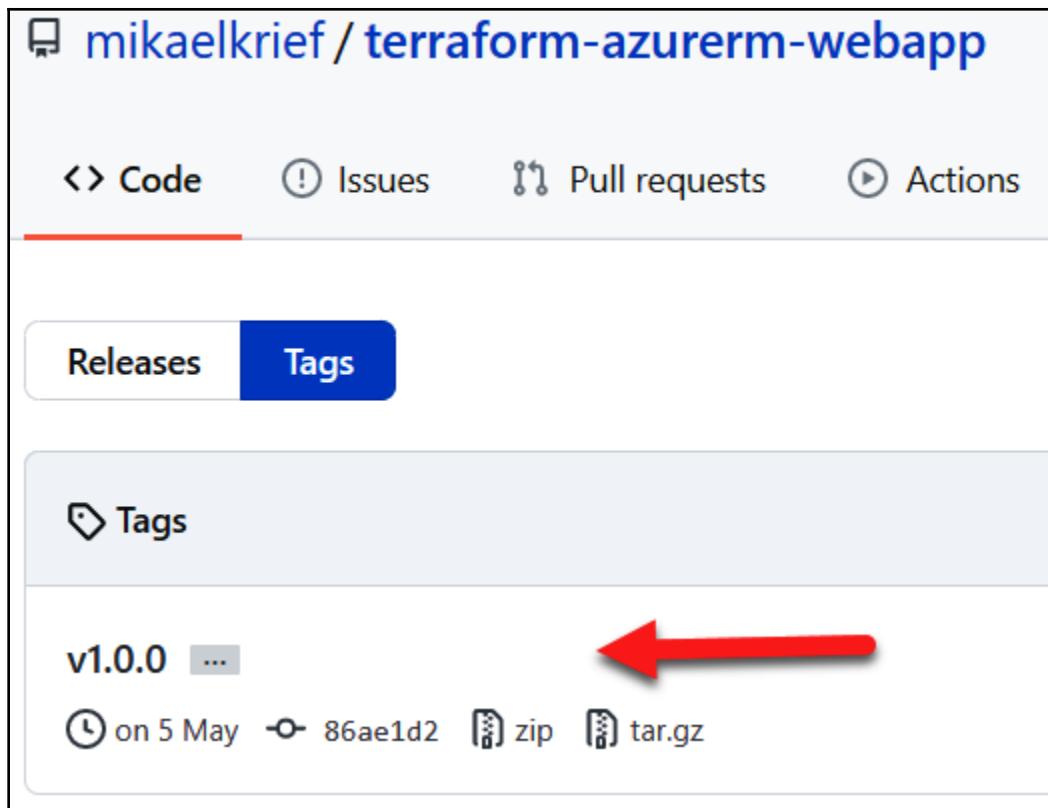
Retrieved token for user MikaelKrief

---

Success! Terraform has obtained and saved an API token.

The new API token will be used for any future Terraform command that must make authenticated requests to app.terraform.io.

The screenshot shows the Terraform Cloud interface. At the top, there's a dark blue header with a logo, the text "Choose an organization ▾", and a close button "X". Below the header is a navigation bar with "Settings / Tokens". On the left, a sidebar titled "USER SETTINGS" lists "Profile", "Organizations", "Password", "Two Factor Authentication", and "Tokens" (which is highlighted with a red circle containing the number 3). In the main content area, the title "Tokens" is displayed. A text block explains that API tokens can be used to access the Terraform Cloud API and lists actions. It includes a link to "API tokens documentation". Below this is a blue button "Create an API token" with a red circle containing the number 4. Underneath, a section titled "Tokens (2)" shows two entries. The first entry is "terraform login2", created 12 minutes ago by user MikaelKrief, with a "Delete" icon. The second entry is partially visible. A user profile sidebar on the right shows "Signed in as MikaelKrief", "Getting Started", "User Settings" (highlighted with a red circle containing the number 2), and "Log Out".



A screenshot of the Terraform Cloud organization settings page for 'demoBook'. The left sidebar lists organization settings like General, Plan & Billing, Users, Teams, API Tokens, Authentication, SSH Keys, and VCS Providers (which is highlighted with a red box). The main content area is titled 'VCS Providers' and shows a single provider entry for 'Terraform Cloud (demoBook)'. It includes fields for Callback URL, HTTP URL, API URL, Created date (Jul 17, 2020 19:29:48 pm), and OAuth Token ID. A note states that a connection was made on Jul 17, 2020 19:30:12 pm via OAuth authentication. Buttons for 'Edit Client' and 'Delete client' are at the bottom right.

The screenshot shows the Terraform Cloud interface. At the top, there is a dark blue header bar with the Terraform logo, the workspace name "demoBook", and several navigation tabs: "Workspaces", "Modules" (which is highlighted with a red box), and "Settings". Below the header, a white sidebar displays the current workspace path: "demoBook / Modules".

This screenshot shows the "Modules" page within the "demoBook" workspace. On the left, there is a sidebar with the workspace path "demoBook / Modules". The main content area features a "Getting started with Terraform Cloud" section containing a checklist. The checklist items are: "Sign up" (checked), "Create organization" (checked), "Create workspace" (checked), and "Plan and apply" (checked). Below the checklist, there are two buttons: "+ Design configuration" and "+ Add module". The "+ Add module" button is highlighted with a red box.

This screenshot shows the "Add Module" wizard. The title is "Add Module". It states that the module will be created under the current organization, "demoBook". It mentions that modules can be added from all supported VCS providers. The process is divided into three steps: 1. Connect to VCS (highlighted with a red box), 2. Choose a repository, and 3. Confirm selection. The first step, "Connect to a version control provider", asks to choose the provider for the module source code. It lists "GitHub" (Terraform Cloud (demoBook)) and "Connect to a different VCS".

# Add Module

This module will be created under the current organization, **demoBook**. Modules can be added from all [supported VCS providers](#).



Connect to VCS



Choose a repository



Confirm selection

## 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>`.

15 repositories

Filter

mikaelkrief/terraform-provider-azuredevops-1 >

mikaelkrief/terraform-azurerm-webapp >

mikaelkrief/terraform-provider-azuredevops >

## Add Module

This module will be created under the current organization, **demoBook**. Modules can be added from all [supported VCS providers](#).

Connect to VCS

Choose a repository

3 Confirm selection

### Confirm selection

Provider GitHub (Terraform Cloud (demoBook))

Repository mikaelkrief/terraform-azurerm-webapp

**Publish module**

Cancel

**Delete module**

**Versions** 1

**Provision Instructions**

Version 1.0.0

Copy and paste into your Terraform configuration and set values for the input variables. Or, [design a configuration](#) to easily use module and workspace outputs as inputs.

```
module "webapp" {  
  source  = "app.terraform.io/demoBook"  
  version = "1.0.0"  
}
```

[navigation arrows]

## Create a new Workspace

Workspaces allow you to organize infrastructure and collaborate on Terraform runs.

1 Connect to VCS

2 Choose a repository

3 Configure settings

### Connect to a version control provider

Choose the version control provider that hosts the Terraform configuration for this workspace.



GitHub

Terraform Cloud (demoBook)

[Connect to a different VCS](#)

## Create a new Workspace

Workspaces allow you to organize infrastructure and collaborate on Terraform runs.

1 Connect to VCS

2 Choose a repository

3 Configure settings

### Choose a repository

Choose the repository that hosts your Terraform source code. We'll watch this for commits and pull requests.

82 repositories

Filter

mikaelkrief/Terraform-Cookbook



# Create a new Workspace

Workspaces allow you to organize infrastructure and collaborate on Terraform runs.

Connect to VCS

Choose a repository

3 Configure settings

## Configure settings

### Workspace Name

demo-app-remote

1

The name of your workspace is unique and used in tools, routing, and UI. Dashes, underscores, and alphanumeric characters are permitted. Learn more about [naming workspaces](#).

Advanced options

### Terraform Working Directory

CHAP08/remote/

2

The directory that Terraform will execute within. This defaults to the root of your repository and is typically set to a subdirectory matching the environment when multiple environments exist within the same repository.

Terraform will change into the `CHAP08/remote/` directory prior to executing any operation. Any modules utilized can be referenced outside of this directory.

### Automatic Run Triggering

Choose when runs should be triggered by VCS changes.

Always trigger runs

Only trigger runs when files in specified paths change

3

CHAP08/remote/

Working Directory



e.g. /modules

Add path

### VCS branch

(default branch)

4

The branch from which to import new versions. This defaults to the value your version control provides as the default branch for this repository.

Include submodules on clone

Checking this box will perform a recursive clone of your repositories submodules, making them available in the resulting slug containing your Terraform configuration. Recursive clone is performed with `--depth 1`.

5 Create workspace

Cancel

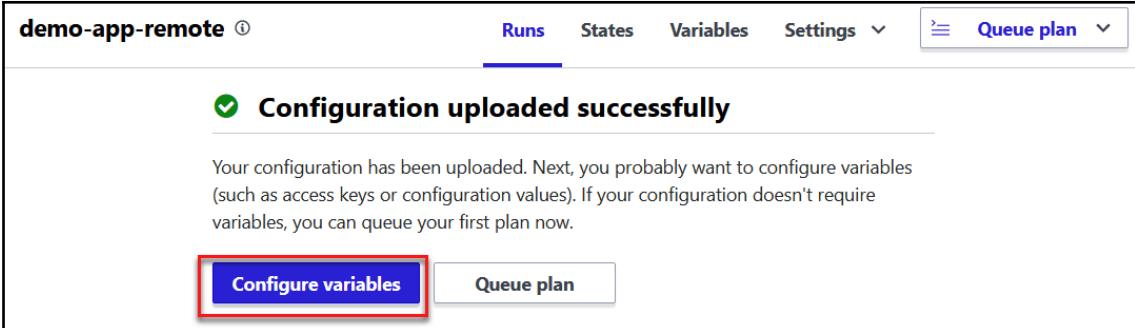
**demo-app-remote** ⓘ

Runs States Variables Settings Queue plan

**Configuration uploaded successfully**

Your configuration has been uploaded. Next, you probably want to configure variables (such as access keys or configuration values). If your configuration doesn't require variables, you can queue your first plan now.

**Configure variables** Queue plan



## Environment Variables

These variables are set in Terraform's shell environment using `export`.

Key	Value	...
ARM_SUBSCRIPTION_ID	8a7aace5-74aa-416f-[REDACTED]	...
ARM_CLIENT_ID	282936c6-298d-40bc-[REDACTED]	...
ARM_CLIENT_SECRET	<i>Sensitive - write only</i>	...
SENSITIVE		
ARM_TENANT_ID	2e3a33f9-66b1-4e2a-[REDACTED]	...
<b>+ Add variable</b>		

The screenshot shows the Terraform Cloud interface. At the top, there's a navigation bar with a logo, the workspace name "demoBook", and links for "Workspaces", "Modules", and "Settings". On the far right are a help icon and a user profile icon.

The main area shows the path "demoBook / Workspaces / demo-app-remote / Variables".

A sidebar on the left contains a "Getting started with Terraform Cloud" section, which includes a checklist with four items: "Sign up" (checked), "Create organization" (checked), "Create workspace" (checked), and "Plan and apply" (checked). Below this is a "Variables" section.

The main content area has tabs for "Runs", "States", "Variables" (which is selected), and "Settings". A "Queue plan" dropdown menu is open, showing the reason "demo book queue plan" (marked with a red circle containing the number 2) and two buttons: "Queue plan" (marked with a red circle containing the number 3) and "Cancel".

**Plan finished** a minute ago Resources: **4** to add, **0** to change, **0** to destroy ^

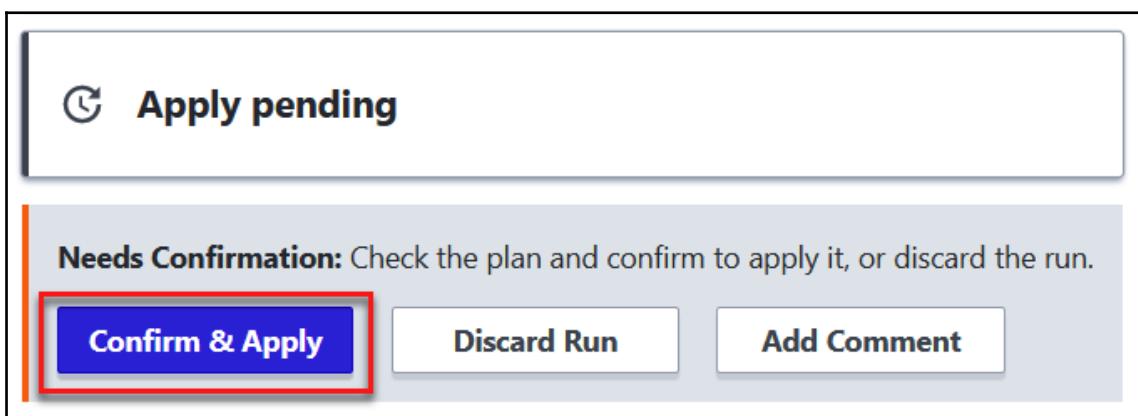
Started a few seconds ago

[Download Sentinel mocks](#) Sentinel mocks can be used for [testing your Sentinel policies](#)

[View raw log](#)  [Top](#)  [Bottom](#)  [Expand](#)  [Full screen](#)

```
# module.webapp.azurerm_application_insights.appinsight-app will be created
+ resource "azurerm_application_insights" "appinsight-app" {
    + app_id                               = (known after apply)
    + application_type                     = "web"
    + daily_data_cap_in_gb                = (known after apply)
    + daily_data_cap_notifications_disabled = (known after apply)
    + disable_ip_masking                  = false
    + id                                    = (known after apply)
    + instrumentation_key                 = (sensitive value)
    + location                             = "westeurope"
    + name                                 = "myappdemobookcloud"
    + resource_group_name                 = "RG_MyAPP_DemoTFCLOUD"
    + retention_in_days                   = 90
    + sampling_percentage                 = 100
}
```

Plan: 4 to add, 0 to change, 0 to destroy.



## Apply pending

**Confirm & Apply:** Add a comment to explain this action.



agree with changes

1

2

Confirm Plan

Close

✓ APPLIED

## demo book queue plan

CURRENT



MikaelKrief triggered a **run** from Terraform Cloud UI an hour ago

Run Details ▾



**Plan finished** 2 hours ago

Resources: **4** to add, **0** to change, **0** to destroy ▾



**Apply finished** a few seconds ago

Resources: **4** added, **0** changed, **0** destroyed ▾



MikaelKrief a few seconds ago  
agree

Run confirmed

**Comment:** Leave feedback or record a decision.

Add Comment

### Apply Method

**Auto apply**

Automatically apply changes when a Terraform plan is successful. Plans that have no changes will not be applied. If this workspace is linked to version control, a push to the default branch of the linked repository will trigger a plan and apply.

**Manual apply**

Require an operator to confirm the result of the Terraform plan before applying. If this workspace is linked to version control, a push to the default branch of the linked repository will only trigger a plan and then wait for confirmation.

### Terraform Version

0.12.28



The version of Terraform to use for this workspace. Upon creating this workspace, the latest version was selected and will be used until it is changed manually. It will **not upgrade automatically**.

### Terraform Working Directory

CHAP08/remote/

The directory to execute Terraform commands in. This defaults to the root of the configuration directory, but can be set to a subdirectory if you use a shared repository for multiple Terraform configurations.

Terraform will change into the `CHAP08/remote/` directory prior to executing any operation.

**Save settings**

**demo-app-remote** ⓘ

Runs States Variables Settings Queue plan

**Destruction and Deletion**

There are two independent steps for destroying this workspace. First, any Terraform infrastructure should be destroyed. Second, including any variables, settings, and alert history can be deleted.

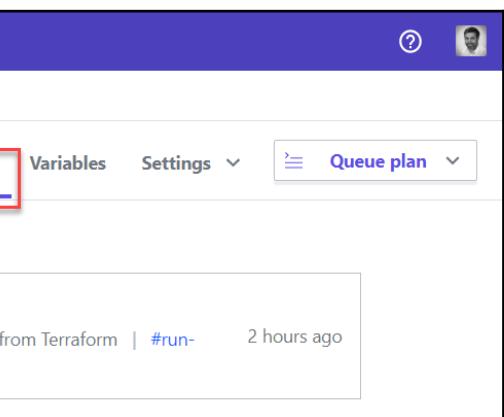
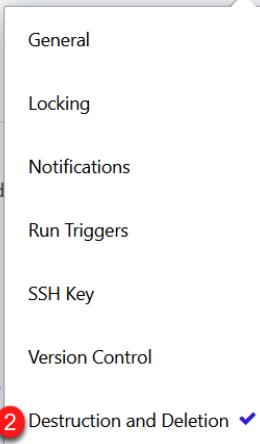
**Destroy infrastructure**

**Allow destroy plans**  
When enabled, this setting allows a destroy plan to be used via the CLI.

**Manually destroy**

Queuing a destroy plan will redirect to a new plan that will destroy all of the infrastructure managed by Terraform. It is equivalent to running `terraform plan -destroy -out=destroy.tfplan` followed by `terraform apply destroy.tfplan` locally.

**Queue destroy plan**



```

PS C:\REPOSPERSO\terraform-Cookbook> terraform plan
Running plan in the remote backend. Output will stream here. Pressing Ctrl-C
will stop streaming the logs, but will not stop the plan running remotely.

Preparing the remote plan...

The remote workspace is configured to work with configuration at
CHAP08/remote/ relative to the target repository.

Terraform will upload the contents of the following directory,
excluding files or directories as defined by a .terraformignore file ①
at C:\REPOSPERSO\terraform-Cookbook/.terraformignore (if it is present),
in order to capture the filesystem context the remote workspace expects:
C:\REPOSPERSO\terraform-Cookbook

To view this run in a browser, visit:
https://app.terraform.io/app/demoBook/demo-app-remote/runs/run-tmEfD2fAUUpVkgms

Waiting for the plan to start...

Terraform v0.12.28
Configuring remote state backend... ②
Initializing Terraform configuration...
2020/07/20 17:06:53 [DEBUG] Using modified User-Agent: Terraform/0.12.28 TFC/9396454cc1
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

```

The screenshot shows the Terraform Cloud user interface. At the top, there's a blue header bar with a logo, a dropdown menu 'Choose an organization', and a user profile icon with a red notification badge containing the number '1'. Below the header, the main content area has a sidebar on the left and a main panel on the right.

- Left Sidebar:** Labeled 'USER SETTINGS' at the top. It contains links: Profile, Organizations, Password, Two Factor Authentication, and Tokens (which is highlighted with a red circle containing the number '3').
- Main Panel - Top Section:** Labeled 'Tokens' at the top. It says: 'Your API tokens can be used to access the Terraform Cloud API and actions your user account is entitled to. For more information, see [tokens documentation](#)'. It includes a 'Create an API token' button (labeled '4' in a red circle) and a 'User Settings' link (labeled '2' in a red circle).
- Main Panel - Bottom Section:** A table titled 'Tokens (4)' showing one token entry:
 

<b>demoBook</b>	Last used <b>15 minutes ago</b>	Created <b>an hour ago</b> by user <b>MikaelKrief</b>	
-----------------	---------------------------------	---	--

 The delete icon is labeled '5' in a red circle.

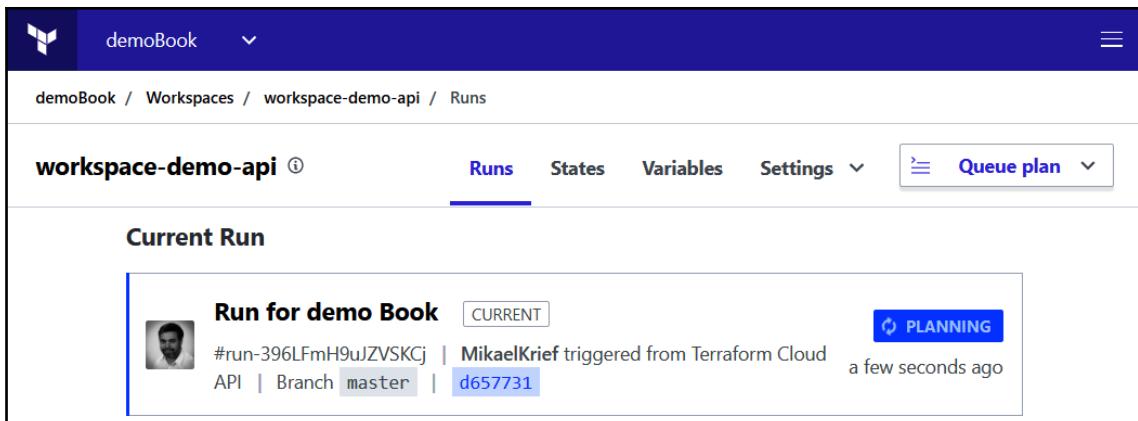
```
PS [REDACTED]\Terraform-Cookbook\CHAP08\api> .\tfcloud-workspaces.ps1 476xcI  
ws-L7oEEUmatRPwP6kS
```

The screenshot shows the Terraform Cloud interface. At the top, there's a navigation bar with tabs for 'demoBook' (selected), 'Workspaces' (highlighted with a red circle labeled '1'), 'Modules', and 'Settings'. Below the navigation is a breadcrumb trail: 'demoBook / Workspaces'. The main area is titled 'Workspaces' with '4 total'. A button '+ New workspace' is visible. Below this is a table with columns: 'WORKSPACE NAME', 'RUN STATUS', 'RUN', 'REPO', and 'LATEST CHANGE'. The rows are: 'demo-app' (status: 0, run: 0, repo: mikaelkrief/Terraform-Cookbook, change: 6 days ago); 'demo-app-remote' (status: 0, run: 1, run status: NEEDS CONFIRMATION, run ID: run-wYpZ8BqR1YgaEJ2R, repo: mikaelkrief/Terraform-Cookbook, change: 2 days ago); 'my-app-demo' (status: 0, run: 0, repo: mikaelkrief/Terraform-Cookbook, change: 6 days ago); and 'workspace-demo-api' (status: 0, run: 0, run status: NEEDS CONFIRMATION, run ID: run-wYpZ8BqR1YgaEJ2R, repo: mikaelkrief/terraform-Cookbook, change: a few seconds ago). The 'workspace-demo-api' row is highlighted with a red box.

The screenshot shows the 'Variables' section of the Terraform Cloud interface. The top navigation bar includes 'demoBook', 'Workspaces' (selected), 'Modules', 'Settings', and a 'Queue plan' dropdown. The breadcrumb trail shows 'demoBook / Workspaces / workspace-demo-api / Variables'. The main title is 'Variables'. Below it is a section 'Environment Variables'. A note says 'These variables are set in Terraform's shell environment using `export`.'. A table lists variables:

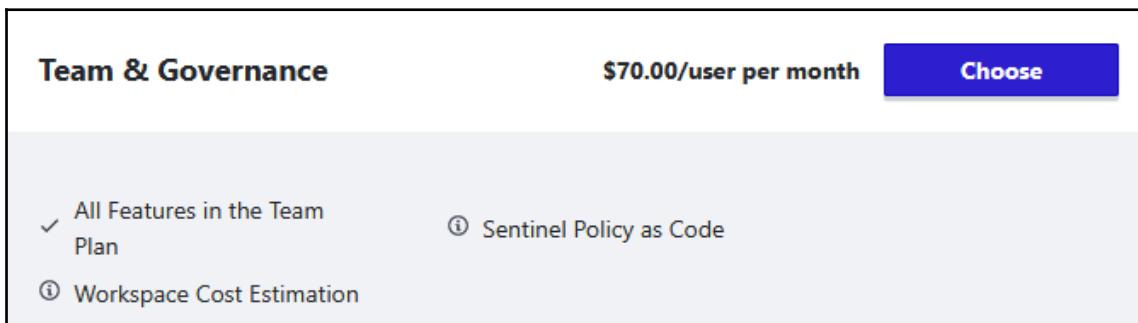
Key	Value	...
ARM_SUBSCRIPTION_ID	8a7aace5-74aa-416f [REDACTED]	...
ARM_CLIENT_ID	282936c6-298d-40bc [REDACTED]	...
ARM_CLIENT_SECRET	Sensitive - write only The Service Principal Client Secret	...
ARM_TENANT_ID	2e3a33f9-66b1-4e2a [REDACTED]	...

A button '+ Add variable' is at the bottom left.



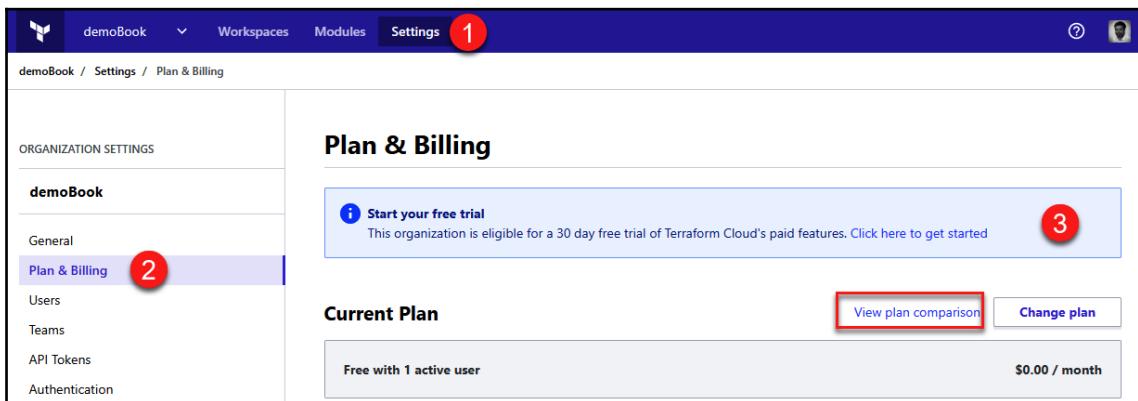
The screenshot shows the 'Runs' tab selected in the 'workspace-demo-api' workspace. A single run is listed under 'Current Run':

- Run for demo Book**: CURRENT
- #run-396LFmH9uJZVSKCj | MikaelKrief triggered from Terraform Cloud API | Branch master | d657731
- PLANNING
- a few seconds ago



The screenshot shows the 'Team & Governance' section. It includes:

- Team & Governance**
- \$70.00/user per month**
- Choose** button
- All Features in the Team Plan**
- SENTINEL Policy as Code**
- Workspace Cost Estimation**



The screenshot shows the 'Plan & Billing' section. It includes:

- ORGANIZATION SETTINGS**
- demoBook**
- General**
- Plan & Billing** (highlighted with a red circle labeled 2)
- Users**
- Teams**
- API Tokens**
- Authentication**
- Plan & Billing** heading
- Start your free trial**: This organization is eligible for a 30 day free trial of Terraform Cloud's paid features. [Click here to get started](#)
- Current Plan**: Free with 1 active user, \$0.00 / month
- View plan comparison** and **Change plan** buttons

The screenshot shows the 'demoBook' settings interface. At the top, there's a navigation bar with 'demoBook', 'Workspaces', 'Modules', 'Settings' (which has a red circle with '1' over it), and a help icon. Below the navigation is a breadcrumb trail: 'demoBook / Settings / Policy Sets'. The main content area is titled 'Policy Sets' with a 'Connect a new policy set' button (circled with '3'). It displays a message: 'There are no policy sets connected' and 'Policy sets are groups of Sentinel policies which may be enforced on workspaces.' Below this, there are links to 'Try using a foundational policy' and 'How to create a policy set'. On the left, a sidebar lists 'ORGANIZATION SETTINGS' under 'demoBook', including 'General', 'Plan & Billing', 'Users', 'Teams', 'API Tokens', 'Authentication', 'SSH Keys', 'Cost Estimation', 'Policies', 'Policy Sets' (which is circled with '2'), and 'VCS Providers'.

# Connect a Policy Set

Policy sets are groups of Sentinel policies which may be enforced on workspaces.

1 Connect to VCS

2 Choose a repository

3 Configure settings

## Connect to a version control provider

Choose the version control provider that hosts the Sentinel policies for this policy set.



GitHub

Terraform Cloud (demoBook)

[Connect to a different VCS](#)

✓ Connect to VCS

2 Choose a repository

3 Configure settings

## Choose a repository

Choose the repository that hosts your Terraform source code. We'll watch this for commits and pull requests.

1 repository of 84 total

Terraform-Cookbook

mikaelkrief/Terraform-Cookbook



**demoBook**

General  
Plan & Billing  
Users  
Teams  
API Tokens  
Authentication  
SSH Keys  
Cost Estimation  
Policies  
**Policy Sets**  
VCS Providers

Policy sets are groups of Sentinel policies which may be enforced on workspaces.

Connect to VCS     Choose a repository    **3 Configure settings**

## Configure settings

Name  1

You can use letters, numbers, dashes (-) and underscores (\_) in your policy set name.

Description  2

Policy Set Source

[^ Hide additional options](#)

#### Policies Path

CHAP08/sentinel-policies/ 3

The path within the repository where the desired policies are present. This directory should include a "sentinel.hcl" configuration file. By default, the repository root is used. The leading "/" is optional.

#### VCS branch

(default branch)

The branch from which to import new versions. This defaults to the value your version control provides as the default branch for this repository.

#### Scope of Policies

- Policies enforced on **all workspaces**
- Policies enforced on **selected workspaces**

**Connect policy set** 5

### Scope of Policies

Policies enforced on **all** workspaces

Policies enforced on **selected** workspaces



## Workspaces

The name of the workspace you wish to add to this policy set.

demo-app-remote



demo-app



Add workspace

! NEEDS CONFIRMATION

Queued manually in Terraform C...

CURRENT



MikaelKrief triggered a **run** from Terraform Cloud UI a few seconds ago

Run Details ▾



**Plan finished** 2 minutes ago

Resources: **4** to add, **0** to change, **0** to destroy ▾



**Policy check passed** 2 minutes ago

Policies: **2** passed, **0** failed ▾

Queued a minute ago > Passed a minute ago

✓ passed Demo\_Policies/allowed-app-service-plan-tiers

✓ passed Demo\_Policies/restrict-app-service-to-ftps

**I NEEDS CONFIRMATION**

## Queued manually in Terrafor...

CURRENT



MikaelKrief triggered a **run** from Terraform Cloud UI a few seconds ago

Run Details ▾



**Plan finished** 8 minutes ago

Resources: **4** to add, **0** to change, **0** to destroy ▾



**Policy check passed** 8 minutes ago

Policies: **2** passed, **0** failed ▾



**Apply pending**

**Needs Confirmation:** Check the plan and confirm to apply it, or discard the run.

**Confirm & Apply**

**Discard Run**

**Add Comment**

Plan finished a few seconds ago Resources: 4 to add, 0 to change, 0 to destroy

Policy check hard failed a few seconds ago Policies: 1 passed, 1 hard failed

Queued 8 minutes ago > Hard failed 8 minutes ago

passed Demo\_Policies/allowed-terraform-version

failed Demo\_Policies/allowed-app-service-plan-tiers

[View raw log](#)

to false. This false was not due to an undefined value or runtime error.

2 policies evaluated.

```
## Policy 1: Demo_Policies/allowed-app-service-plan-tiers (hard-mandatory)

Result: false

Description:
This policy uses the Sentinel tfplan/v2 import to require that all Azure Service plan have tier SKU from an allowed list

Print messages:

module.webapp.azurerm_app_service_plan.plan-app has sku.0.tier with value Premium that is not in the allowed list: [Basic, Standard]

FALSE - ./allowed-app-service-plan-tiers.sentinel:21:1 - Rule "main"
## Policy 2: Demo_Policies/allowed-terraform-version (soft-mandatory)
```

Result: true

— Apply will not run

This screenshot shows the results of a Terraform run. At the top, it says 'Plan finished' and 'Policy check hard failed'. Below that, it lists two policies: one passed and one failed. The failed policy is 'Demo\_Policies/allowed-app-service-plan-tiers'. A red arrow points from the 'hard failed' status to this policy. Another red arrow points from the 'Apply will not run' message at the bottom to the failed policy. The terminal output shows that the 'Result: false' for this policy is due to a 'Premium' tier SKU being present, which is not in the allowed list [Basic, Standard].

## Team & Governance

\$70.00/user per month [Choose](#)

✓ All Features in the Team Plan

ⓘ Sentinel Policy as Code

ⓘ Workspace Cost Estimation

This screenshot shows the Microsoft Sentinel pricing page for the 'Team & Governance' plan. It displays a price of \$70.00/user per month and a 'Choose' button. Below the price, there's a list of features: 'All Features in the Team Plan' (marked with a checkmark) and 'Sentinel Policy as Code' and 'Workspace Cost Estimation' (each marked with an info icon). A red arrow points from the 'All Features in the Team Plan' item to the 'Choose' button.

The screenshot shows the 'demoBook' application interface. At the top, there is a dark blue header bar with the 'demoBook' logo, a dropdown menu, and navigation links for 'Workspaces', 'Modules', and 'Settings'. A red circle with the number '1' is positioned next to the 'Settings' link. Below the header, the URL 'demoBook / Settings / Cost Estimation' is displayed. The main content area is divided into two sections: 'ORGANIZATION SETTINGS' on the left and 'Cost Estimation' on the right.

**ORGANIZATION SETTINGS**

- demoBook** (highlighted with a red circle containing '2')
- General
- Plan & Billing
- Users
- Teams
- API Tokens
- Authentication
- SSH Keys
- Cost Estimation** (highlighted with a red circle containing '2')
- Policies
- Policy Sets

**Cost Estimation**

Enable Cost Estimation for all workspaces (highlighted with a red circle containing '3')

When possible, display an estimated monthly cost for resources provisioned.

**Update settings** (highlighted with a red circle containing '4')