



HashiTalks



Marcelo Zambrana

Software Engineer at Microsoft
He/Him

@ch4mbr4s





Como hacer tu primera contribución al proveedor AzureRM

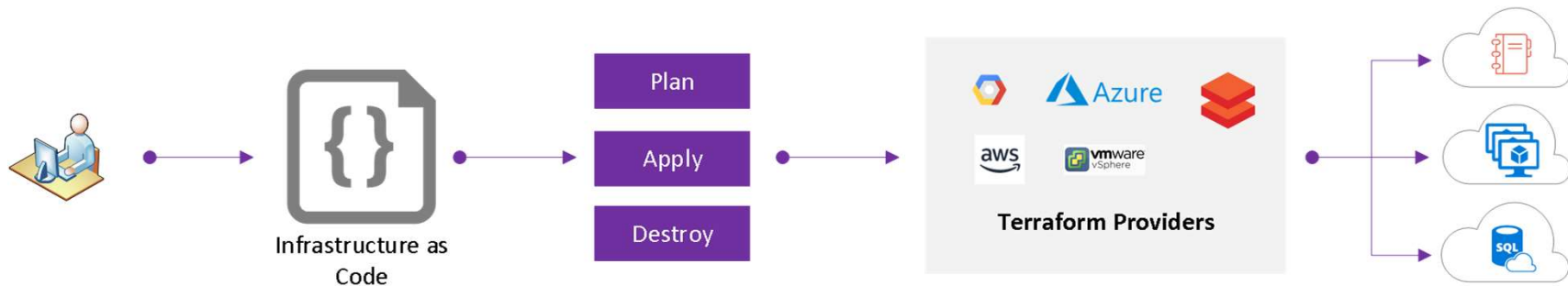
código abierto simplificado.



Descripción general



Flujo de Trabajo de Terraform



Como Funcionan los Proveedores



Estructura del Proyecto



```

TERMINAL
$
├─ examples
├─ internal
│   ├── acceptance
│   ├── clients
│   ├── common
│   ├── features
│   ├── provider
│   ├── resourceproviders
│   └── sdk
├─ services
├─ tf
├─ tools
├─ vendor
└─ website
```

Interacción con Azure



go-azure-sdk

Un [SDK de Go generado por Hashicorp](#) para la interacción con Azure Resource Manager

Hamilton

Para interactuar con Microsoft Graph y obtener un token de autenticación mediante MSAL.

Azure SDK for Go

[SDK de Go](#) generado a partir de los archivos Swagger dentro del repositorio Azure/azure-rest-api-specs para la interacción con Azure Resource Manager.

Giovanni

Para interactuar con las API del plano de datos de Azure Storage.



Creación, Ejecución de pruebas y Depuración del proveedor

Creación

Se puede compilar el proveedor ejecutando *make build*, que compilará el proveedor y colocará el binario en el directorio *\$GOPATH/bin*.

Pruebas

- Pruebas de unidad:
\$ make test
- Pruebas de aceptación:
\$ make acctests SERVICE='<service>' TESTARGS='-run=<nameOfTheTest>' TESTTIMEOUT='60m'

Creación, Ejecución de pruebas y Depuración del proveedor cont.



Depuración

Los registros se pueden ver ejecutando Terraform:

```
$ TF_LOG=INFO terraform apply
```

o las pruebas de aceptación:

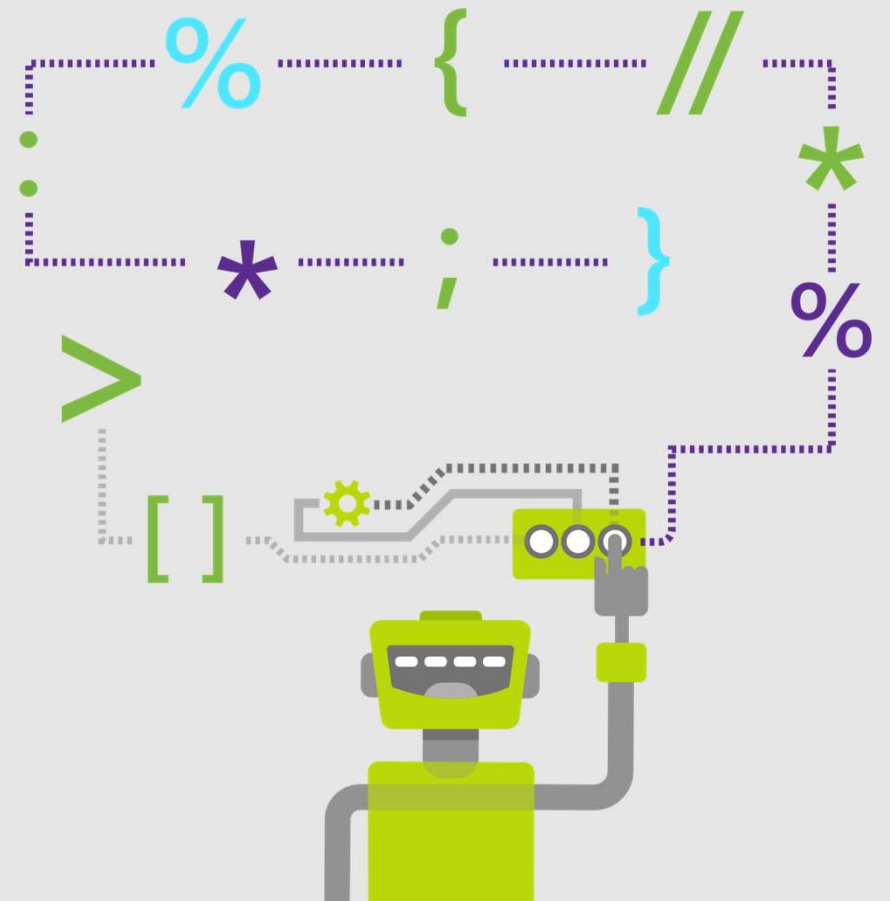
```
$ TF_LOG=DEBUG make acctests SERVICE='<service>' TESTARGS='-run=<nameOfTheTest>'  
TESTTIMEOUT='60m'
```


Algunas Convenciones (Alias)



```
gorepos = cd ~/Repos/  
git st = git status  
git cob = git checkout -b  
git pom = git push origin main  
git pod = git push origin dev  
tfi = terraform init  
tch = terraform fmt + terraform validate  
tfp = terraform plan -out run.plan  
tfa = terraform apply --auto-approve run.plan  
tfo = terraform output  
tfd = terraform destroy -force  
agl = az group list -o table
```

Demo



Creación del Proyecto



```

$ make build
==> Checking that code complies with gofmt requirements...
==> Checking that Custom Timeouts are used...
==> Checking that acceptance test packages are used...
go generate ./internal/services/...
go generate ./internal/provider/
go install
```

Ejecución de pruebas



TERMINAL

```
$ $ make acctests SERVICE='storage' TESTARGS=' -run=TestAccStorageBlob_disappears' TESTTIMEOUT='60m'
==> Checking that code complies with gofmt requirements...
==> Checking that Custom Timeouts are used...
==> Checking that acceptance test packages are used...
TF_ACC=1 go test -v ./internal/services/storage -run=TestAccStorageBlob_disappears -timeout 60m -ldflags="-X=github.com/hashicorp/terraform-provider-azurerm/version.ProviderVersion=acc"
=== RUN   TestAccStorageBlob_disappears
=== PAUSE TestAccStorageBlob_disappears
=== CONT  TestAccStorageBlob_disappears
--- PASS: TestAccStorageBlob_disappears (102.02s)
```

PASS

ok	github.com/hashicorp/terraform-provider-azurerm/internal/services/storage	102.046s
----	---	----------

Depuración



TERMINAL

```
$ $ TF_LOG=INFO
$ terraform validate
| Warning: Provider development overrides are in effect
|
| The following provider development overrides are set in the CLI configuration:
|   - hashicorp/azurerm in /home/marcelo/Repos/GoProjects/bin
|
| The behavior may therefore not match any released version of the provider and applying
changes may cause the state to become incompatible with published releases.
Success! The configuration is valid, but there were some validation warnings as shown above.
```

Conclusiones



- Tener el entorno de desarrollo automatizado.
- La mejor forma de probar es creando el proveedor.
- No todas las contribuciones tienen que ser código.
- Obtener experiencia práctica con proyectos reales con impacto a nivel mundial.
- Aprender nuevas tecnologías, frameworks y flujos de trabajo y al mismo tiempo compartir tu experiencia con otros.
- Impacto y Reconocimiento.

Algunos Recursos



- Get Started – The Go programming language
<https://go.dev/learn/>
- Azure REST API
<https://github.com/Azure/azure-rest-api-specs/tree/main/specification>
- Guías para colaboradores
<https://hashicorp.github.io/terraform-provider-azurerm/>
- Slack Workspace para colaboradores (Pedir Invitación)
<https://terraform-azure.slack.com/>
- Documentación del Proveedor de Azure:
<https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>
- GitHub Actions:
<https://docs.github.com/en/actions/learn-github-actions>



Muchas Gracias



<https://github.com/chambras>



<https://www.linkedin.com/in/marcelozambrana/>



marcelo.zambrana@microsoft.com



[@ch4mbr4s](https://twitter.com/ch4mbr4s)

hugs@hashicorp.com | learn.hashicorp.com | discuss.hashicorp.com