Terraform

Infrastructure as Code

Ashish K Thakur
DevOps Engineer, IBM

What is Terraform

```
A tool for building, changing, and versioning infrastructure safely and efficiently
```

Key Features

- Infrastructure as Code
- Execution Plans
- Resource Graph
- Change Automation

Infrastructure as Code

www.ioccc.org/2018/anderson/prog.c(https://www.ioccc.org/2018/anderson/prog.c)

```
#include<stdio.h>
int a = 256;int main(){for(char b[a+a+a],
*c=b ,*d=b+ a ,*e=b+a+a,*f,*g=fgets(e,(b[
a]=b [a+a] =a- a,a) , stdin);c[0]=a-a,f=c
,c=d ,d=e ,e=f, f= g,g =0,g = fgets(e,a+a
-a+ a -a+a -a+ a- +a,stdin ),f +a-a ; pu\
tchar(+10)) { for( int h= 1,i=1,j, k=0 ,l
=e[0]==32,m,n=0,o=c [ 0]== 32, p, q=0;d[q
];j=k,k=l,m=n,n=o,p=(j)+(k* 2 )+(l =(i =
e[ q]&&i ) &&e[q +1 ]== 32,l*4)+(m* 8 )+(
16*  n  )+(  o  =(h =c[ q]&&h)&&c[q+1]==
32,o* (16+16) )+0-0 +0, putchar(" ......"
/*\  (  |||  )  |/|/ / */".')|)\\\\\\\\'"
"" "|||"   "|||" "|'" ")|)\\\\\\\\'/|/(/"
"(/'/|/\\|\\|'/|/(/(/'/|/\\|\\|"[d[q++]==
32?p:0]));}}/* typographic tributaries */
```

Infrastructure as Code (intended use)

- Collaborate & share
- Evolve your infrastructure
- Automation friendly

```
resource "ibm_container_cluster" "demo_cluster" {
  name         = "test"
  datacenter   = "dal10"
  hardware     = "dedicated"
  machine_type = "free"
}
```

Execution Plans

- One Safe Workflow Across Providers

Change Automation

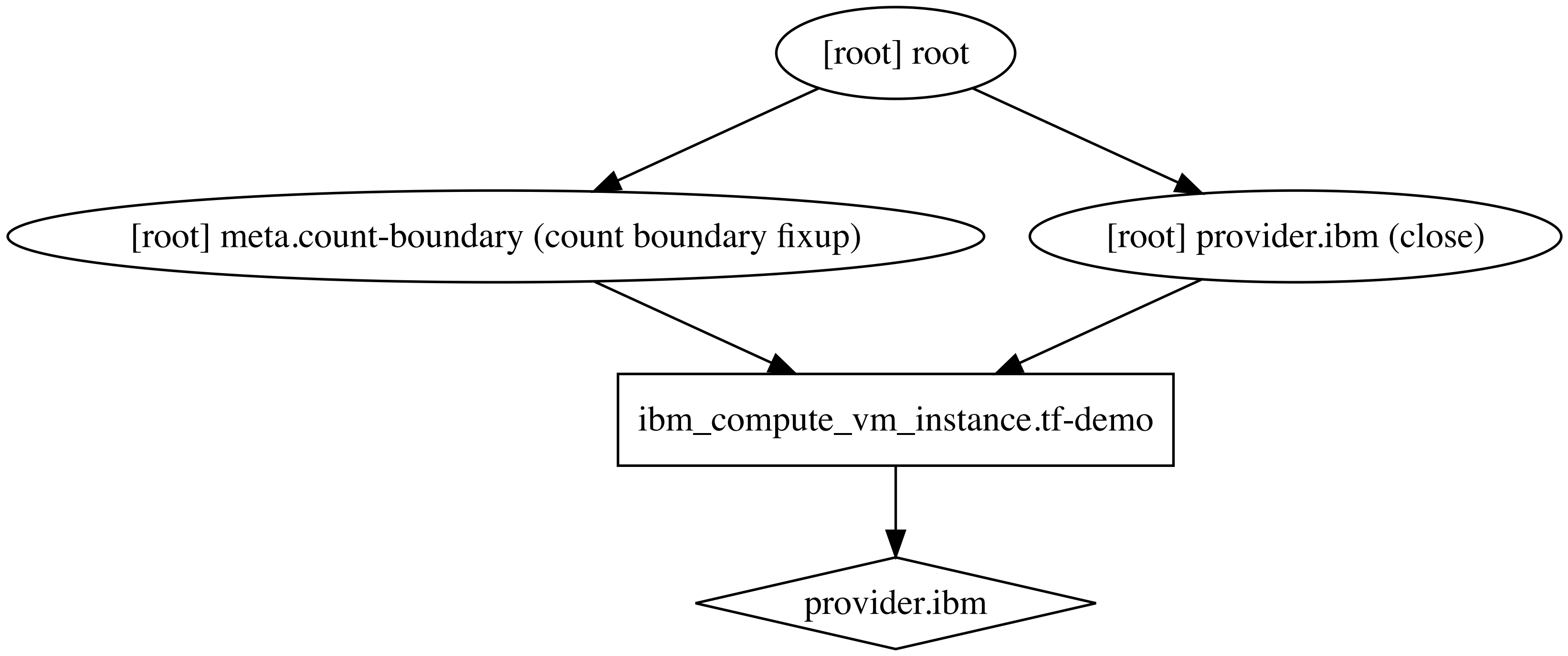
- Reproducible Infrastructure

Build infrastructure

```
resource "ibm_compute_vm_instance" "tf-demo" {
  hostname          = "terraform"
  domain            = "ibm.com"
  os_reference_code = "DEBIAN_8_64"
  datacenter        = "wdc01"
  hourly_billing    = true
  cores             = 1
  memory            = 1024
}
```

Resource Graph

Generated via `terraform graph | dot -Tsvg > graph.svg`


[root] root
[root] meta.count-boundary (count boundary fixup)
[root] provider.ibm (close)
ibm_compute_vm_instance.tf-demo
provider.ibm

Terraform Plan

```
terraform plan
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:

  + ibm_compute_vm_instance.tf-demo
      id:                           <computed>
      block_storage_ids.#:          <computed>
      cores:                        "1"
      datacenter:                   "wdc01"
      domain:                       "ibm.com"
      hostname:                     "terraform"
      hourly_billing:               "true"
      ipv4_address:                 <computed>
      local_disk:                   "true"
      memory:                       "1024"
      network_speed:                "100"
      os_reference_code:            "DEBIAN_8_64"
      private_subnet:               <computed>
      private_subnet_id:            <computed>
      private_vlan_id:              <computed>
      ...
Plan: 1 to add, 0 to change, 0 to destroy.
```

Terraform Apply

```
ibm_compute_vm_instance.tf-demo: Creating...
  block_storage_ids.#:           "" => "<computed>"
  cores:                         "" => "1"
  datacenter:                    "" => "wdc01"
  domain:                        "" => "ibm.com"
  hostname:                      "" => "terraform"
  hourly_billing:                "" => "true"
  ipv6_address:                  "" => "<computed>"
  ipv6_address_id:               "" => "<computed>"
  ipv6_enabled:                  "" => "false"
  ipv6_static_enabled:           "" => "false"
  memory:                        "" => "1024"
  network_speed:                 "" => "100"
  os_reference_code:             "" => "DEBIAN_8_64"
  public_bandwidth_unlimited:    "" => "false"
  public_subnet:                 "" => "<computed>"

ibm_compute_vm_instance.tf-demo: Still creating... (10s elapsed)
...
ibm_compute_vm_instance.tf-demo: Still creating... (4m0s elapsed)
ibm_compute_vm_instance.tf-demo: Creation complete after 4m7s (ID: 61066829)

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

Terraform Show

```
ibm_compute_vm_instance.tf-demo:
  id = 61066829
  block_storage_ids.# = 0
  cores = 1
  disks.# = 1
  disks.0 = 25
  ipv4_address = 108.168.183.12
  ipv4_address_private = 10.108.177.82
  os_reference_code = DEBIAN_8_64
  private_interface_id = 35144391
  private_network_only = false
  private_security_group_ids.# = 0
  private_subnet = 10.108.177.64/26
  private_subnet_id = 646776
  private_vlan_id = 2434895
  public_bandwidth_unlimited = false
  public_interface_id = 35144393
  public_subnet = 108.168.183.8/29
  public_subnet_id = 600454
  public_vlan_id = 2434893
  ...
```

Terraform Change

```
resource "ibm_compute_vm_instance" "tf-demo" {
  hostname          = "terraform"
  domain            = "ibm.com"
  os_reference_code = "DEBIAN_8_64"
  datacenter        = "wdc01"
  hourly_billing    = true
  cores             = 1
  memory            = 2048
}
```

Terraform Plan after Change

```
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.

ibm_compute_vm_instance.tf-demo: Refreshing state... (ID: 61066829)

------------------------------------------------------------------------

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

Terraform will perform the following actions:

  ~ ibm_compute_vm_instance.tf-demo
      memory: "1024" => "2048"


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

------------------------------------------------------------------------

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
```

Terraform Apply after Change

```
ibm_compute_vm_instance.tf-demo: Refreshing state... (ID: 61066829)
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  ~ ibm_compute_vm_instance.tf-demo
      memory: "1024" => "2048"

Plan: 0 to add, 1 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

ibm_compute_vm_instance.tf-demo: Modifying... (ID: 61066829)
  memory: "1024" => "2048"
ibm_compute_vm_instance.tf-demo: Still modifying... (ID: 61066829, 10s elapsed)
...
ibm_compute_vm_instance.tf-demo: Modifications complete after 1m46s (ID: 61066829)
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

Terraform Destroy

Bring existing infra under Terraform management

```
resource "ibm_compute_vm_instance" "vm_created_outside_tf" {}
```

`terraform import ibm_compute_vm_instance.vm_created_outside_tf <vm-id>`

State

- Mapping to the Real World
- Keeps track of metadata
- Stored by default in a local file named "terraform.tfstate"
- JSON
- Workspaces

```
-  equivalent to renaming your state file
-  usage - a developer working on a complex set of infrastructure changes might create a new temporary
```

- Remote state

```
- state held in memory when used by Terraform.
- may be encrypted at rest depending on backend used.
- good option for storing sensitive data
```

Backends

- Determines how state is loaded and how an operation such as apply is executed
- Work efficiently in a team
- Keeping sensitive information off disk
- Remote operations - Turn off your computer and still get the job done

Provisioners

- execute scripts on a local or remote machine

Terraform vs Ansible

- Configuration Management vs Orchestration
- Procedural vs Declarative

Reference (https://blog.gruntwork.io/why-we-use-terraform-and-not-chef-puppet-ansible-saltstack-or-cloudformation-7989dad2865c)

```
- ec2:
    count: 10
    image: ami-v1
    instance_type: t2.micro
```

```
resource "aws_instance" "example" {
  count = 10
  ami = "ami-v1"
  instance_type = "t2.micro"
}
```

IBM Provider Templates

- github.com/Cloud-Schematics (https://github.com/Cloud-Schematics)

Setting up on Windows

- www.ibm.com/blogs/bluemix/2018/01/setting-terraform-ibm-cloud-provider-windows/

  (https://www.ibm.com/blogs/bluemix/2018/01/setting-terraform-ibm-cloud-provider-windows/)

Terraform Provider for IBM Cloud

- github.com/IBM-Cloud/terraform-provider-ibm (https://github.com/IBM-Cloud/terraform-provider-ibm)
- github.com/IBM-Cloud/terraform-provider-ibm/tree/master/examples (https://github.com/IBM-Cloud/terraform-provider-ibm/tree/master/examples)

Thank you

Ashish K Thakur
DevOps Engineer, IBM
ashishth@in.ibm.com (mailto:ashishth@in.ibm.com)