Terraform

Emanuele Rocca

2015-10-05

Launch infrastructure

Terraform provides a common configuration to launch infrastructure

- Virtual
- Physical
- DNS providers
- Email

Outline

- Configuration files
- Providers
- Commands
- Provisioners
- DNS

Configuration files

Files ending in .tf

https://terraform.io/docs/configuration/syntax.html

```
provider "digitalocean" {
    token = "THIS_IS_A_BANANA"
}

resource "digitalocean_droplet" "lb2" {
    image = "ubuntu-14-04-x64"
    name = "lb2"
    region = "nyc2"
    size = "512mb"
    ssh_keys = [135751,595595]
}
```

Providers: Digital Ocean

- Digital-Ocean specific docs:
  https://terraform.io/docs/providers/do/index.html
- Find Digital Ocean SSH key IDs:
  https://api.digitalocean.com/v1/ssh_keys/

Providers: AWS

```
provider "aws" {
    access_key = "ACCESS_KEY_HERE"
    secret_key = "SECRET_KEY_HERE"
    region = "us-east-1"
}

resource "aws_instance" "lb1" {
    ami = "ami-408c7f28"
    instance_type = "t1.micro"
    key_name = "whatever"
}
```

AWS specific docs:
https://terraform.io/docs/providers/aws/index.html

Other providers

- Azure
- Google Cloud
- OpenStack

Commands

- plan - Generate and show an execution plan
- apply - Builds or changes infrastructure
- show - Inspect Terraform state or plan
- destroy - Destroy Terraform-managed infrastructure

Execution plan

```
$ ./terraform plan
Refreshing Terraform state prior to plan...
[...]
+ digitalocean_droplet.lb2
    image:                  "" => "ubuntu-14-04-x64"
    ipv4_address:           "" => "<computed>"
    ipv4_address_private:   "" => "<computed>"
    ipv6_address:           "" => "<computed>"
    ipv6_address_private:   "" => "<computed>"
    locked:                 "" => "<computed>"
    name:                   "" => "lb2"
    region:                 "" => "nyc2"
    size:                   "" => "512mb"
    ssh_keys.#:             "" => "2"
    ssh_keys.0:             "" => "135751"
    ssh_keys.1:             "" => "595595"
    status:                 "" => "<computed>"

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

Build infrastructure

```
$ ./terraform apply
digitalocean_droplet.lb2: Creating...
  image:                  "" => "ubuntu-14-04-x64"
  ipv4_address:           "" => "<computed>"
  ipv4_address_private:   "" => "<computed>"
  ipv6_address:           "" => "<computed>"
  ipv6_address_private:   "" => "<computed>"
  locked:                 "" => "<computed>"
  name:                   "" => "lb2"
  region:                 "" => "nyc2"
  size:                   "" => "512mb"
  ssh_keys.#:             "" => "2"
  ssh_keys.0:             "" => "135751"
  ssh_keys.1:             "" => "595595"
  status:                 "" => "<computed>"
```

Issues with apply

```
Error applying plan:

1 error(s) occurred:

* digitalocean_droplet.lb2: Error waiting for droplet (7990105)
```

- Terraform does not automatically rollback in the face of errors
- State file gets updated with any resources that were successfully created

Success with apply

```
digitalocean_droplet.lb2: Creation complete

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

The state of your infrastructure has been saved to the path
below. This state is required to modify and destroy your
infrastructure, so keep it safe. To inspect the complete state
use the ‘terraform show‘ command.

State path: terraform.tfstate
```

State file

terraform.tfstate


```
"digitalocean_droplet.lb2": {
    "type": "digitalocean_droplet",
    "primary": {
        "id": "7990105",
        "attributes": {
            "id": "7990105",
            "image": "ubuntu-14-04-x64",
            "ipv4_address": "162.243.96.161",
            "locked": "true",
            "name": "lb2",
            "region": "nyc2",
            "size": "512mb",
            "ssh_keys.#": "2",
            "ssh_keys.0": "135751",
            "ssh_keys.1": "595595",
            "status": "new"
        }
    }
}
```

Inspect Terraform state

```
$ ./terraform show
digitalocean_droplet.lb2:
  id = 7994458
  image = ubuntu-14-04-x64
  ipv4_address = 107.170.94.8
  locked = false
  name = lb2
  region = nyc2
  size = 512mb
  ssh_keys.# = 2
  ssh_keys.0 = 135751
  ssh_keys.1 = 595595
  status = active
```

Update infrastructure

After bumping the droplet size from 512mb to 1024mb:

```
$ ./terraform plan
Refreshing Terraform state prior to plan...

digitalocean_droplet.lb2: Refreshing state... (ID: 7994458)

[...]

~ digitalocean_droplet.lb2
    size: "512mb" => "1024mb"


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

Update results

```
* digitalocean_droplet.lb2: Error resizing droplet (7994458):
Error processing droplet action:
API Error: unprocessable_entity: Size can't be blank
```

- Terraform said the resource would be changed "in-place"
- The server was rebooted without asking for confirmation
- It failed (of course "Size" wasn't blank)

Commands: destroy

```
$ ./terraform destroy
Do you really want to destroy?
  Terraform will delete all your managed infrastructure.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

digitalocean_droplet.lb2: Refreshing state... (ID: 7990105)
digitalocean_droplet.lb2: Destroying...
digitalocean_droplet.lb2: Destruction complete

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

Providers

Can be executed to initialize resources after creation.

- Add resources to an inventory management system
- Run a configuration management tool

Provisioners (chef)

A chef provisioner is available, allowing to specify:

- server_url
- validation_client_name
- validation_key_path
- secret_key_path
- version
- attributes
- node_name
- run_list
- environment

Provisioners (remote-exec)

Arbitrary commands can be specified to provision the resource:

```
resource "digitalocean_droplet" "lb2" {
    [...]
    provisioner "remote-exec" {
        inline = [
        "puppet apply",
        "consul join ${digitalocean_droplet.lb2.ipv4_address}"
        ]
    }
}
```

Note that different providers call the same thing differently.

- AWS: private_ip
- Digital Ocean: ipv4_address_private

DNS

Terraform allows to modify DNS records using different providers.

- Cloudflare
- DNSimple
- DNSMadeEasy

Cloudflare

```
provider "cloudflare" {
    email = "pear@example.org"
    token = "THIS_IS_A_TOKEN"
}

resource "cloudflare_record" "lb2" {
    depends_on = [ "digitalocean_droplet.lb2" ]
    domain = "example.org"
    name = "lb2"
    value = "${digitalocean_droplet.lb2.ipv4_address}"
    type = "A"
    ttl = 3600
}
```

Note that it is possible to reference attributes of other resources!

TYPE.NAME.ATTRIBUTE

Conclusions

- Lots of good ideas
- The Digital Ocean provider seems to be very unreliable (2 failures out of 4 commands)
- Reboots happen without asking for confirmation
- Play with it :-)