Terraform

Diego Pacheco

About Me

- Cat's Father
- Principal Software Architect
- Agile Coach
- SOA/Microservices Expert
- DevOps Practitioner
- Speaker
- Author

diegopacheco

@diego_pacheco

http://diego-pacheco.blogspot.com.br/

Terraforming or terraformation (literally, "Earth-shaping") of a planet, moon, or other body is the hypothetical process of deliberately modifying its atmosphere, temperature, surface topography or ecology to be similar to the environment of Earth to make it habitable by Earth-like life.

Terraforming - Wikipedia

https://en.wikipedia.org/wiki/Terraforming

Sobre este resultado Feedback

- ❑ Infrastructure as Code
- ❑ Launch / Creates Infrastructure
- ❑ Change Infrastructure
- ❑ Do down scale as well
- ❑ Similar to AWS Cloud Formation but more generic ☺
- ❑ Show what will do before doing it
- ❑ Based on config file

```
$ terraform
usage: terraform [--version] [--help] <command> [<args>]

Available commands are:
    apply      Builds or changes infrastructure
    destroy    Destroy Terraform-managed infrastructure
    get        Download and install modules for the configuration
    graph      Create a visual graph of Terraform resources
    init       Initializes Terraform configuration from a module
    output     Read an output from a state file
    plan       Generate and show an execution plan
    refresh    Update local state file against real resources
    remote     Configure remote state storage
    show       Inspect Terraform state or plan
    taint      Manually mark a resource for recreation
    version    Prints the Terraform version
```

Example Configuration

```
resource "aws_elb" "frontend" {
    name = "frontend-load-balancer"
    listener {
        instance_port = 8000
        instance_protocol = "http"
        lb_port = 80
        lb_protocol = "http"
    }

    instances = ["${aws_instance.app.*.id}"]
}

resource "aws_instance" "app" {
    count = 5

    ami = "ami-043a5034"
    instance_type = "m1.small"
}
```

Example Configuration

```
resource "digitalocean_droplet" "web" {
    name = "tf-web"
    size = "512mb"
    image = "centos-5-8-x32"
    region = "sfo1"
}

resource "dnsimple_record" "hello" {
    domain = "example.com"
    name = "test"
    value = "${digitalocean_droplet.web.ipv4_address}"
    type = "A"
}
```

$ terraform apply

main.tf

```
# Specify the provider and access details
provider "aws" {
  region = "${var.aws_region}"
}

resource "aws_elb" "web" {
  name = "terraform-example-elb"

  # The same availability zone as our instances
  availability_zones = ["${aws_instance.web.*.availability_zone}"]

  listener {
    instance_port = 80
    instance_protocol = "http"
    lb_port = 80
    lb_protocol = "http"
  }

  # The instances are registered automatically
  instances = ["${aws_instance.web.*.id}"]
}


resource "aws_instance" "web" {
  instance_type = "m1.small"
  ami = "${lookup(var.aws_amis, var.aws_region)}"

  # This will create 4 instances
  count = 4
}
```

$ terraform apply

main.tf

```
# Create our Heroku application. Heroku will
# automatically assign a name.
resource "heroku_app" "web" {}

# Create our DNSimple record to point to the
# heroku application.
resource "dnsimple_record" "web" {
  domain = "${var.dnsimple_domain}"

  name = "terraform"

  # heroku_hostname is a computed attribute on the heroku
  # application we can use to determine the hostname
  value = "${heroku_app.web.heroku_hostname}"

  type = "CNAME"
  ttl = 3600
}

# The Heroku domain, which will be created and added
# to the heroku application after we have assigned the domain
# in DNSimple
resource "heroku_domain" "foobar" {
  app = "${heroku_app.web.name}"
  hostname = "${dnsimple_record.web.hostname}"
}
```

variables.tf

```
variable "dnsimple_domain" {
  description = "The domain we are creating a record for."
}
```

Install Terraform

install-terraform.sh

```
1  #!/bin/bash
2
3  wget https://releases.hashicorp.com/terraform/0.11.11/terraform_0.11.11_linux_amd64.zip
4  unzip terraform_0.11.11_linux_amd64.zip
5  rm -rf terraform_0.11.11_linux_amd64.zip
6  ./terraform
```

SAMPLES

https://github.com/terraform-providers/terraform-provider-aws/tree/master/examples

https://github.com/terraform-providers/terraform-provider-google/tree/master/examples

Terraform

Diego Pacheco