

Terraform

Benoît Charroux

`benoit.charroux@efrei.fr`

What is Terraform ?

What is Terraform ?

Terraform is an infrastructure as code (IaC) tool that build, change, and version infrastructure safely and efficiently.

This includes:

- low-level components such as compute instances, storage, and networking
- high-level components such as DNS entries, SaaS features, etc.

Terraform can manage both existing cloud service providers and custom in-house solutions.

Terraform is an open-source software tool created by HashCorp.

Two approaches for IaC

Declarative vs Imperative

- The declarative approach (functional) focuses on what the eventual target configuration should be. The declarative approach defines the desired state and the system executes what needs to happen to achieve that desired state.
- The imperative approach (procedural) focuses on how the infrastructure is to be changed to meet this. Imperative defines specific commands that need to be executed in the appropriate order to end with the desired conclusion.
- Terraform uses the declarative approach

CCA tools

| Tool | Released by | Approach |
|------------------|------------------|----------------------------|
| <u>Chef</u> | Chef | Declarative and imperative |
| <u>CFEngine</u> | Northern.tech | Declarative |
| <u>Terraform</u> | <u>HashiCorp</u> | Declarative |
| <u>Ansible</u> | <u>Red Hat</u> | Declarative and imperative |

Key features

Key features of Terraform

- Infrastructure as Code: You describe your infrastructure using Terraform's high-level configuration language in human-readable, declarative configuration files.
- Execution Plans: Terraform generates an execution plan describing what it will do and asks for your approval.
- Resource Graph: Terraform builds a resource graph.
- Change Automation: Terraform can apply complex changesets to your infrastructure with minimal human interaction => it will apply only changes.

The Terraform language

The Terraform language

- Terraform language is declaring resources, which represent infrastructure objects
- Example:

```
resource "google_compute_instance" "terraform" {  
  project    = "<PROJECT_ID>"  
  name       = "terraform"  
  machine_type = "n1-standard-1"  
  zone       = "us-central1-a"  
  boot_disk {  
    initialize_params {  
      image = "debian-cloud/debian-9"  
    }  
  }  
  network_interface {  
    network = "default"  
    access_config {  
    }  
  }  
}
```