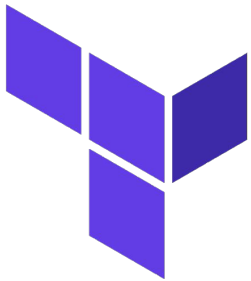
ADC
Applaudo
Developers
Conference
2020

Powered by ApplaudoStudios

Infrastructure as Code with



HashiCorp

Terraform

César Amaya

Technologies

HashiCorp Terraform

aws

@cesaramaya

cesarem

cesaramaya

Agenda

1. The problem
2. Definitions
3. IaC Tools
4. Terraform
5. Demo
6. Benefits
7. What's next
8. Q&A

Infrastructure

Infrastructure is all the software and hardware that support applications.

_Machines
_Cables
_Racks
_Cooling system
_OS
_Deployment pipelines
_Configurations

Servers

_bigs
_noisy
_expensive
_a lot of power needed
_people needed
_dificult to purchase
_racking
_cabling
_cooling
_installing
_configuring

Infrastructure as Code

Is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.[1]

[1] Wittig, Andreas; Wittig, Michael (2016). Amazon Web Services in Action. Manning Press. p. 93.

IaC Tools

Ad Hoc Scripts

Python, bash, PowerShell, etc.

Configuration management tools

Ansible, Pupet, Chef, SaltStack

Server templating tools

Docker, Packer, Vagrant

Orchestration tools

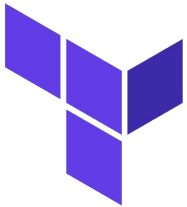
Kubernetes, Amazon ECS, Mesos, Docker Swarn, Rancher, Nomad

Provisioning tools

Terraform, CloudFormation, Azure Resource Manager, Google Deployment Manager.

Some Features

- Developed by HashiCorp
- Open Source
- Written in Go
- Released v0.1.0 on July/28/2014
- Current Stable v0.13.3 on Sep/16/2020
- Modular
- Extensible
- Flavors
    - Terraform CLI
    - Terraform Cloud
    - Terraform Enterprise
- AWS, Azure, GCP, Cisco, PaloAlto, 200+ vendors & services

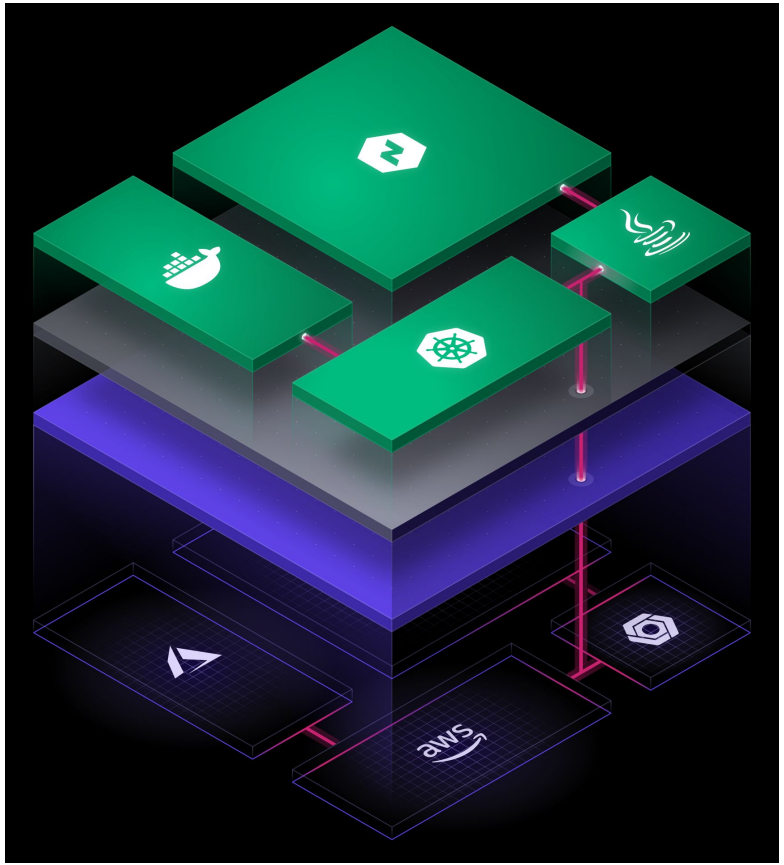


HashiCorp
Terraform

Terraform

Write and execute code to define, deploy, update, and destroy your infrastructure at will.

Terraform: One Tool To Rule Them All!



Terraform Demo

Demo Challenge

Write Terraform code to deploy the following infrastructure in AWS

> create an AWS S3 bucket with two files: test1.txt and test2.txt. \
> the content of these files must be the creation timestamp \
> a cluster of 2 EC2 instances behind an ALB running Traefik to proxy \
> the connections to the files in a S3 bucket \
> the cluster must be deployed in a new VPC \
> this VPC must have only one public subnet \
> protect the files in the S3 bucket so only the EC2 instances using IAM roles \
> can access them

Demo Components

AWS

Terraform

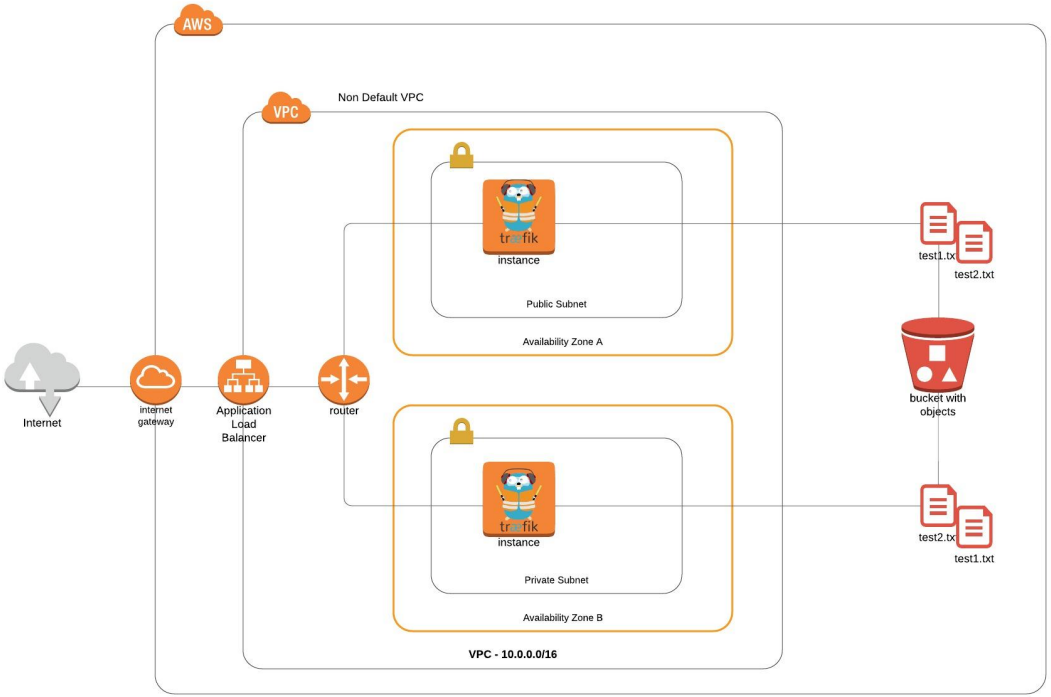
Packer

Docker

Traefik

pottava/s3-proxy

Solution Architecture



Workflow

```
# create ec2 instance image using Packer
$ packer build packer.json

# init state, download providers and modules
$ terraform init

# check the deployment plan
$ terraform plan

# apply infrastructure changes
$ terraform apply

# don’t forget destroy to avoid costs
$ terraform destroy
```

Terraform Benefits

- Any cloud, infrastructure, or service
- Declarative configuration files
- Collaborate and share configurations
- Evolve and version your infrastructure
- Automate provisioning
- Plan and predict changes
- Clearly mapped resource dependencies
- Separation of plan and apply
- Consistent, repeatable workflow
- Reproducible production, staging, and development environments
- Shared modules for common infrastructure patterns
- Combine multiple providers consistently

Terraform to the next level

- Implement Automated Tests
  - Kitchen Terraform, Terratest
- Running Terraform in Automation
  - CI/CD
- Terraform Team Collaboration
  - Terraform Cloud / Terraform Enterprise
- Develop Terraform Modules
  - https://registry.terraform.io/
- Multi Cloud Deployments

Reference

1. https://terraform.io
2. *Terraform: Up & Running Second Edition - Yevgeniy Brikman - O'Reilly Media*
3. https://github.com/cesarem/terraform-webinar-demo-101
4. https://www.oreilly.com/library/view/cloud-native-infrastructure/9781491984291/ch01.html

Applaudo
Developers
Conference
2020

Powered by ApplaudoStudios