Introducción a DevOps y Metodologías Afines

Infrastructure as code (IaC)

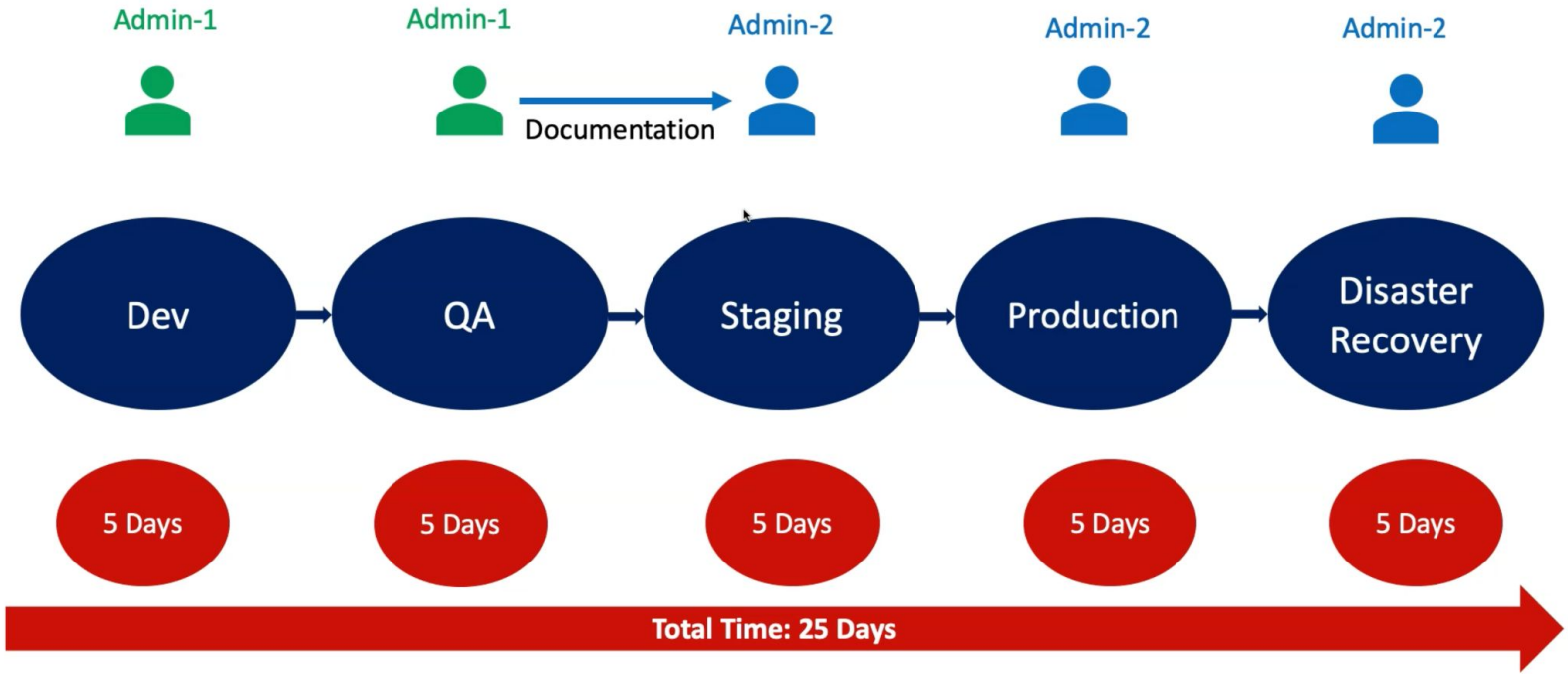
Federico Barceló – Profesor Adjunto
Escuela de Tecnología – Facultad de Ingeniería
UNIVERSIDAD ORT Uruguay
federico@barcelo.com.uy

AGENDA

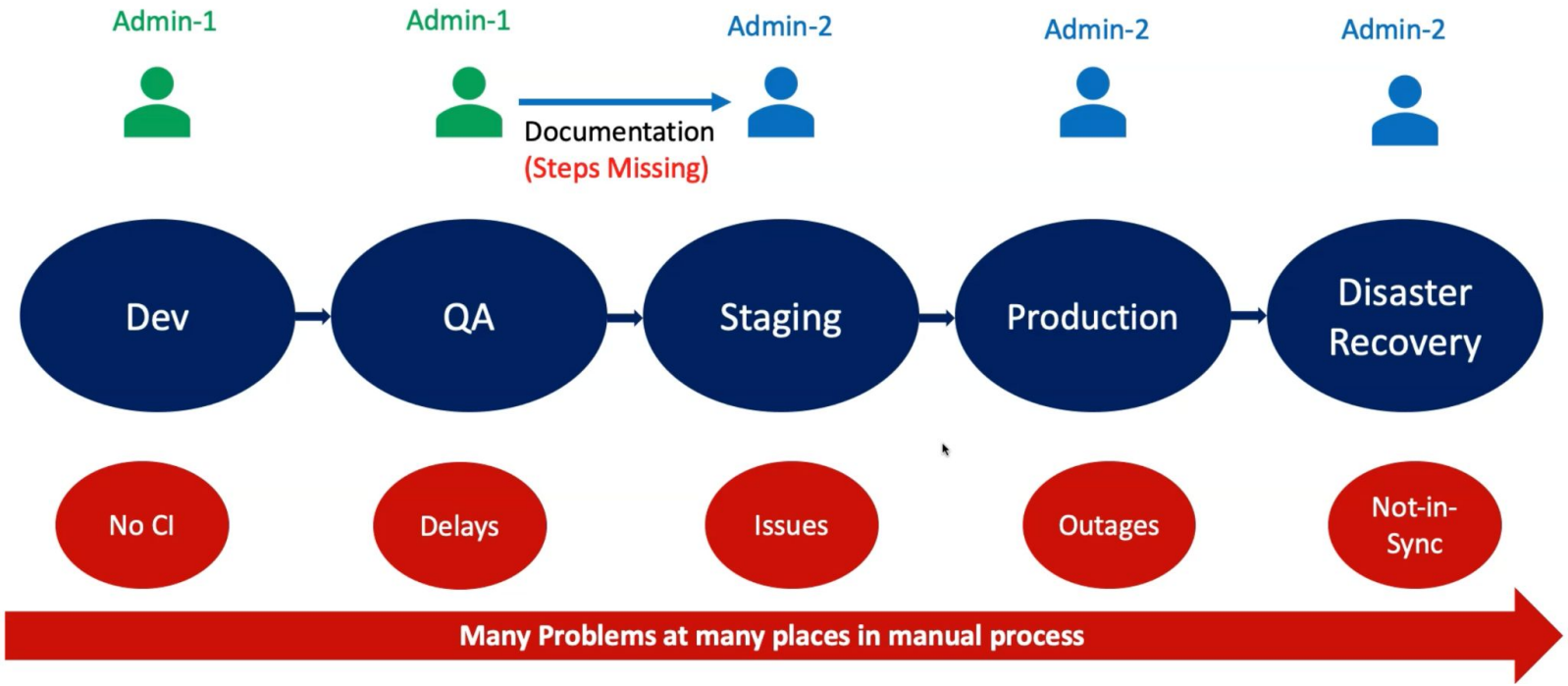
1. Old infra managing way vs new infra managing way
2. ¿Qué es IaC?
3. Herramientas
4. Terraform

Old infra managing way
vs
new infra managing way

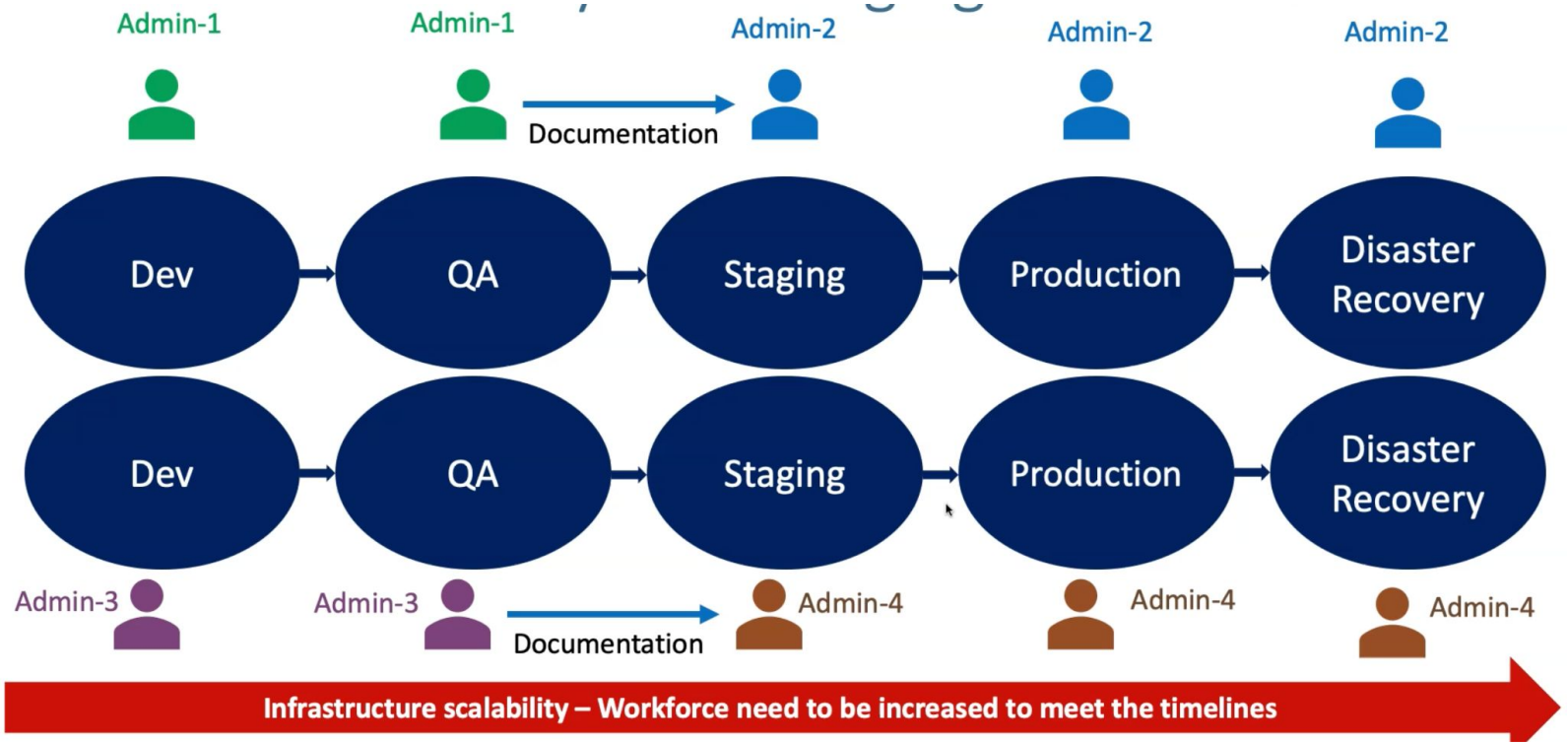
Old infra managing way



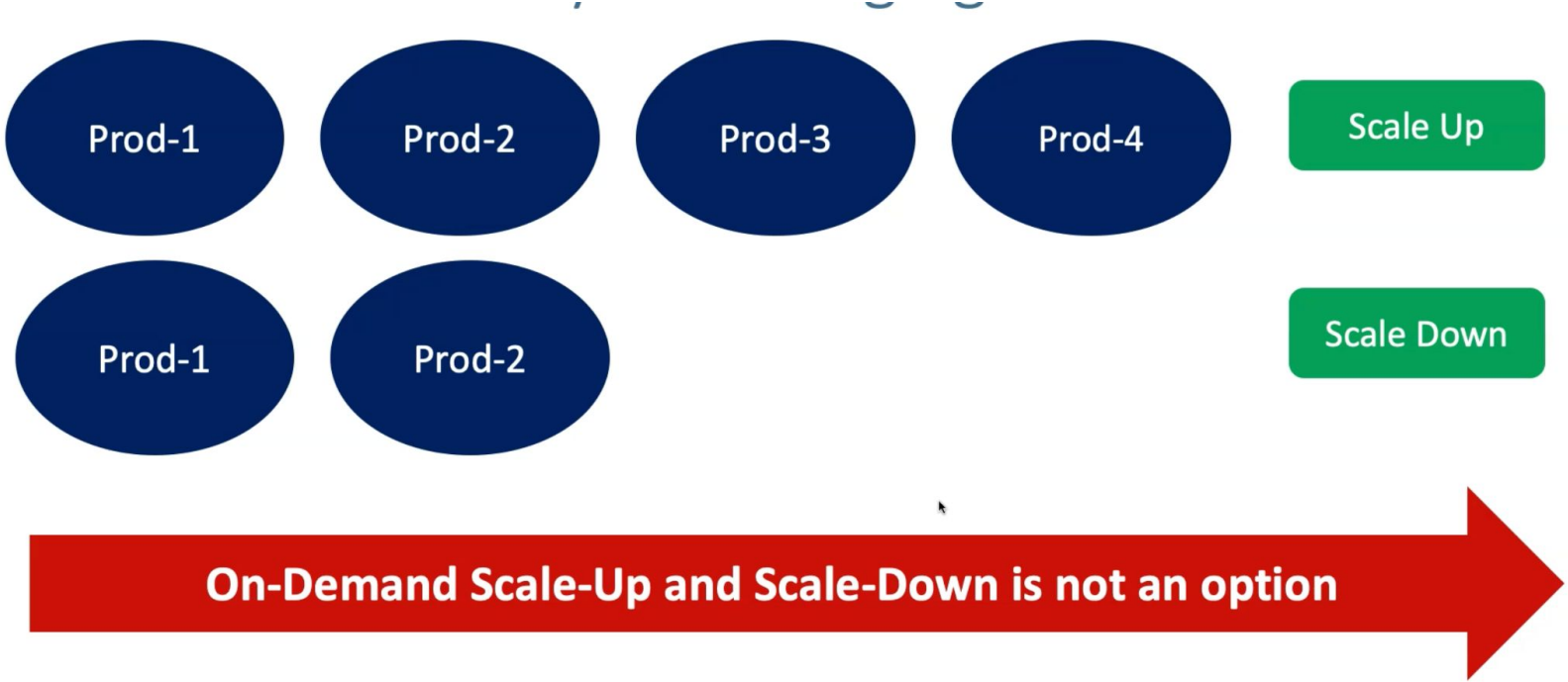
Old infra managing way



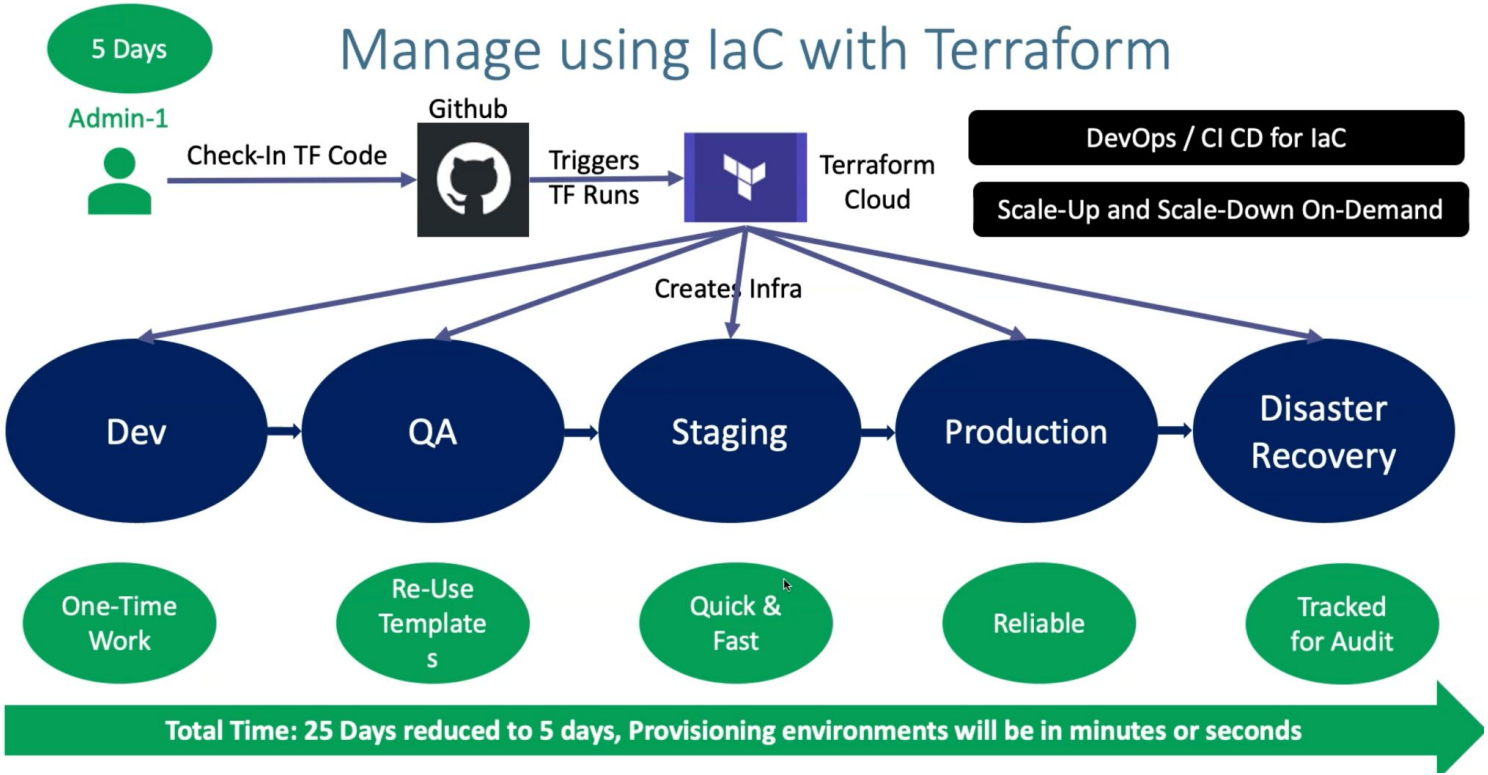
Old infra managing way



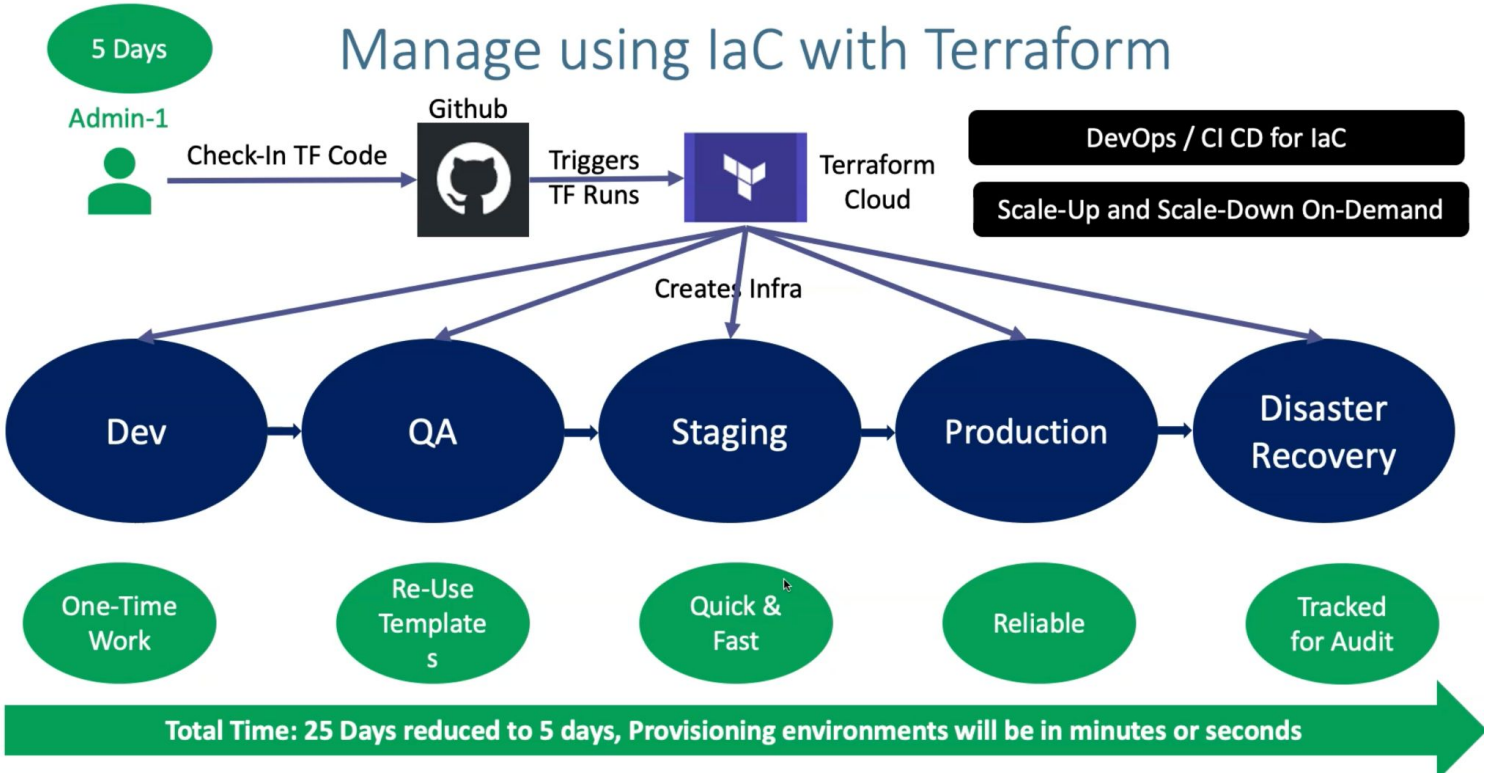
Old infra managing way



New infra managing way



New infra managing way



New infra managing way

Visibility	IaC serves as a very clear reference of what resources we created, and what their settings are. We don't have to navigate to the web console to check the parameters.
Stability	If you accidentally change the wrong setting or delete the wrong resource in the web console you can break things. IaC helps solve this, especially when it is combined with version control, such as Git.
Scalability	With IaC we can write it once and then reuse it many times. This means that one well written template can be used as the basis for multiple services, in multiple regions around the world, making it much easier to horizontally scale.
Security	Once again IaC gives you a unified template for how to deploy our architecture. If we create one well secured architecture we can reuse it multiple times, and know that each deployed version is following the same settings.
Audit	Terraform not only creates resources it also maintains the record of what is created in real world cloud environments using its State files.

¿Qué es IaC?

Infrastructure as Code (IaC)

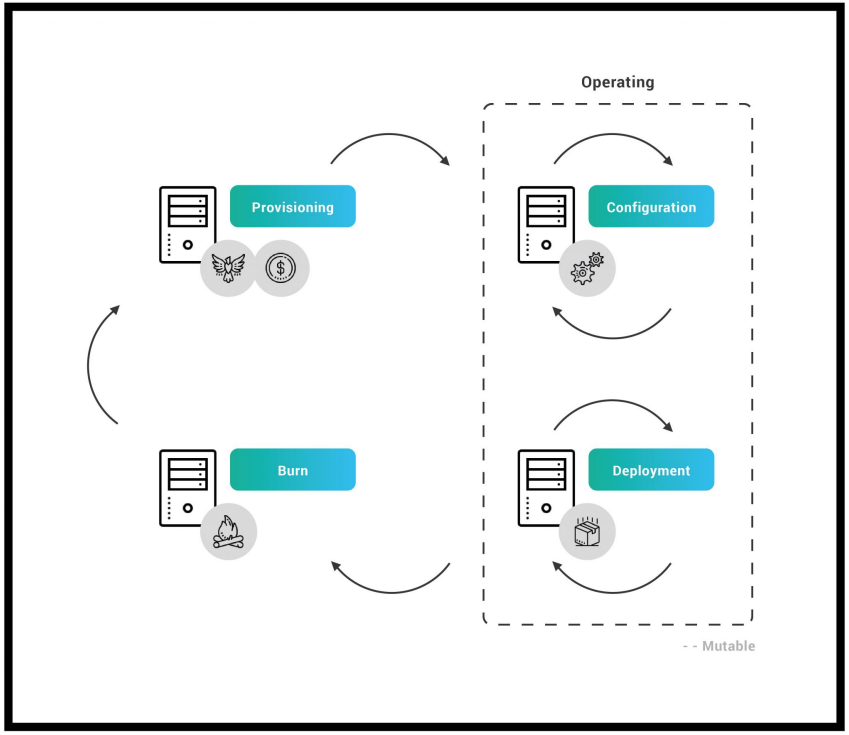
Es el proceso de gestionar la infraestructura de forma automatizada, aplicando los mismos principios y prácticas que los equipos de desarrollo aplican a la hora de escribir código.

Recursos

Grandes empresas

NETFLIX

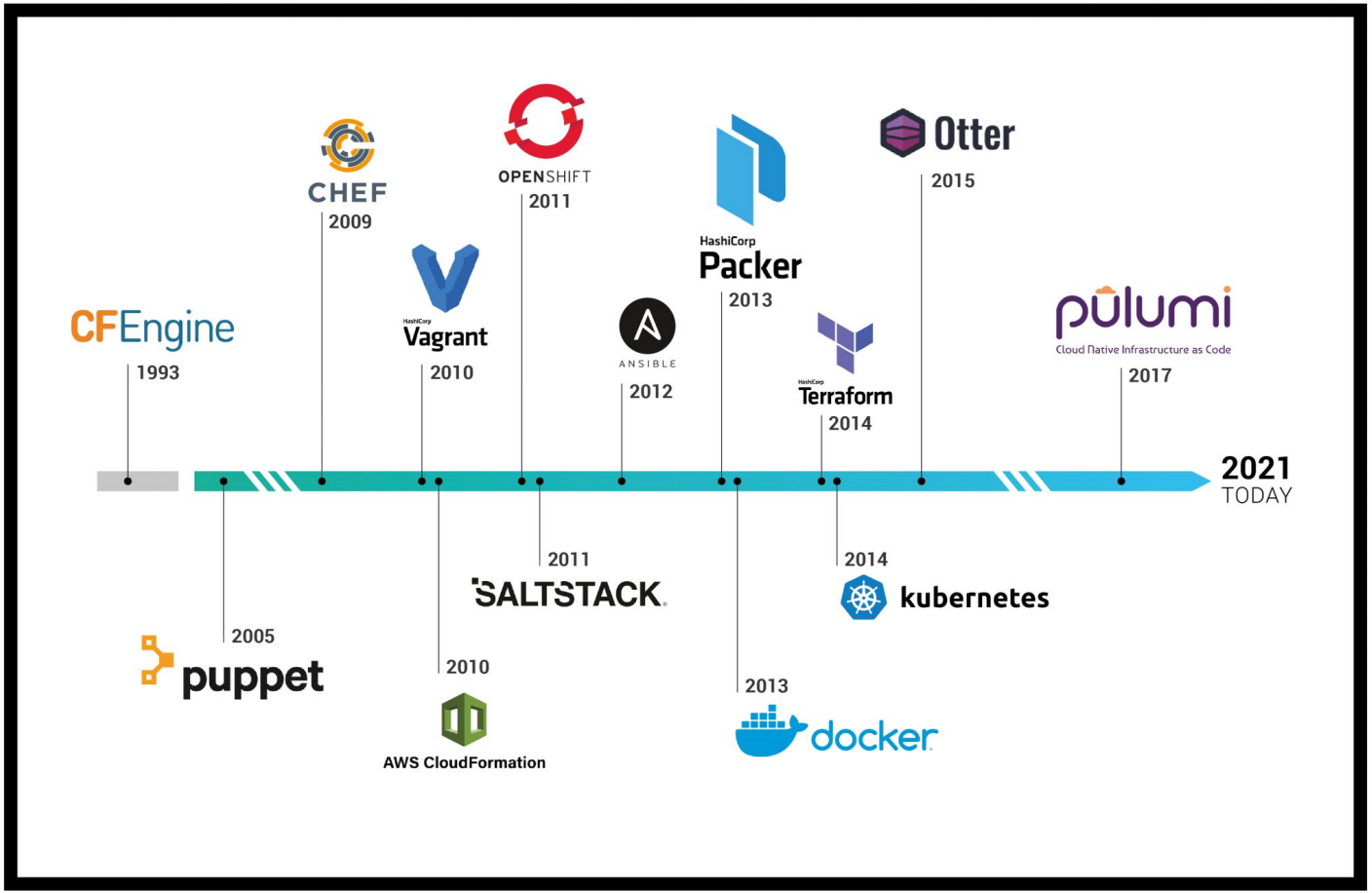
Aprovisionamiento, configuración y implementación



Aprovisionamiento, configuración y implementación

„The result is a unique snowflake – good for a ski resort, bad for a data center.“ – Martin Fowler

„A server should be like a phoenix, regularly rising from the ashes“ – Martin Fowler


CFEngine
1993
CHEF
2009
Vagrant
2010
OPENSHIFT
2011
ANSIBLE
2012
HashiCorp
Packer
2013
HashiCorp
Terraform
2014
Otter
2015
pulumi
Cloud Native Infrastructure as Code
2017
2021
TODAY
2005
puppet
2010
AWS CloudFormation
2011
SALTSTACK
2013
docker
2014
kubernetes

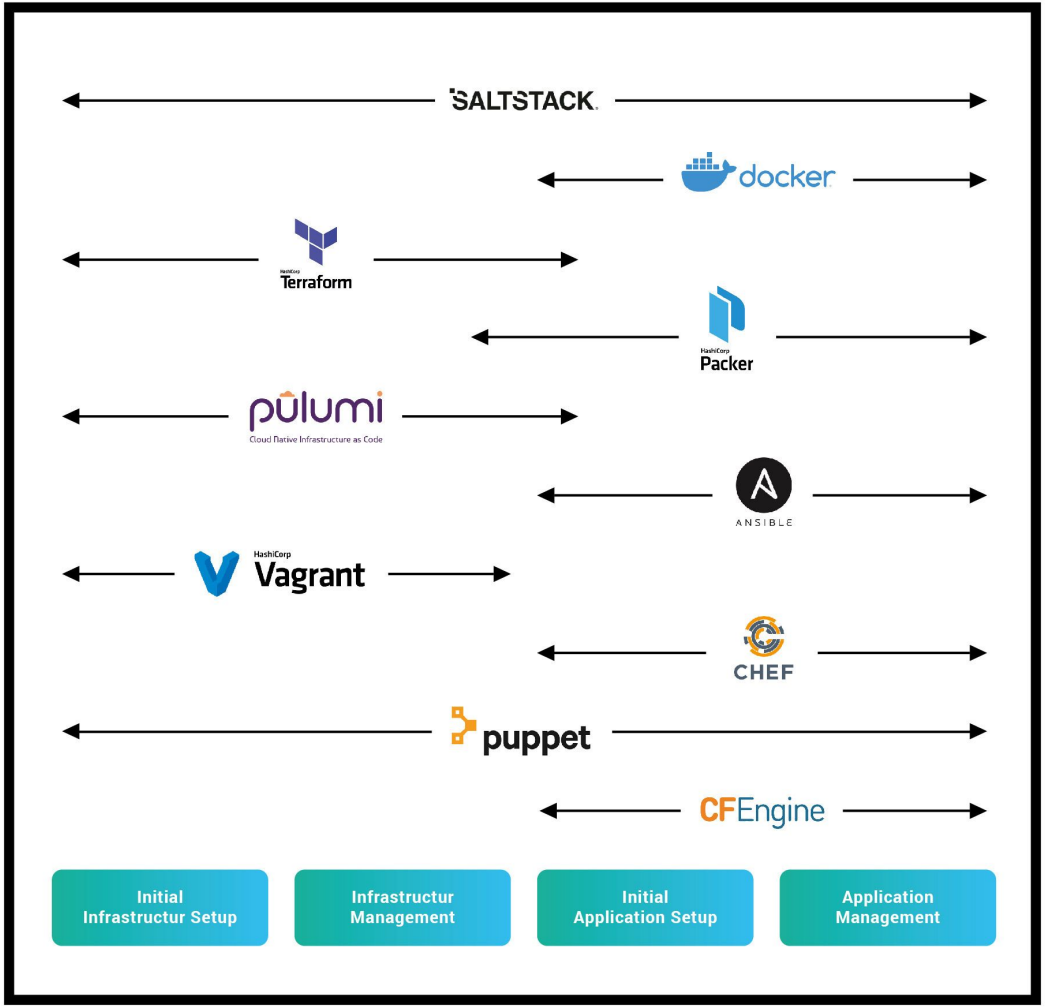
Las diferentes fases

Initial setup phase

- Provisionar la infraestructura
- Configuración de la infraestructura
- Instalación inicial de software
- Configuración inicial del software

Maintaining phase

- Ajustes en la infraestructura
- Eliminación y adición de componentes
- Actualización de software
- Reconfiguración de software


SALTSTACK
docker
Terraform
Packer
pulumi
Cloud Native Infrastructure as Code
ANSIBLE
HashiCorp
Vagrant
CHEF
puppet
CFEngine
Initial Infrastructur Setup
Infrastructur Management
Initial Application Setup
Application Management

Las diferentes fases

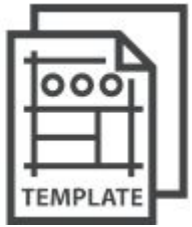
Initial setup phase

- Provisionar la infraestructura
- Configuración de la infraestructura
- Instalación inicial de software
- Configuración inicial del software

Maintaining phase

- Ajustes en la infraestructura
- Eliminación y adición de componentes
- Actualización de software
- Reconfiguración de software

Los diferentes tipos de IaC

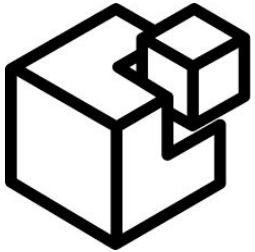


Scripts

```
1
2  #!/bin/bash
3
4  # Update Package Manager
5  sudo apt-get update
6
7  # Install Apache
8  sudo apt-get install -y apache2
9
10 # Start Apache
11 sudo service apache2 start
12
```

Configuration management tools

```
1
2 - hosts: apache
3 sudo: yes
4 tasks:
5 - name: install apache2
6 apt: name=apache2 update_cache=yes state=latest
7
```



Templating tools

```
1
2 FROM ubuntu:latest
3 RUN apt-get -y update &amp;&amp; \
4     apt-get install -y apache2
5 ENTRYPOINT ["/usr/sbin/apache2"]
6 CMD ["-D", "FOREGROUND"]
7
```






Herramientas

Implementaciones IaC

Examples of "Infrastructure as Code" implementations		
vRealize Automation 	Template: Cloud template Target: Multiple Clouds Cloud Agnostic	resources: example-vm: type: Cloud.AWS.EC2.Instance properties: image: Ubuntu-1

AWS CloudFormation 	Template: CFT
	Target: AWS
		"Resources" : {
		"example-vm": {
		"Type": "AWS::EC2::Instance",
		"Properties": {
		"ImageId" {

Azure Resource Manager 	Template: ARM template Target: Azure	"resources": [{ "type": "Microsoft.Compute/virtualMachines", "name": "example-vm",

Terraform 	Template: Terraform configuration
	Target: Multiple Clouds
		resource "aws_instance" "example-vm" {
		ami = "ami-0c55b172cbfefa1f0"
		instance_type = "t2.micro"
		}

Implementaciones Configuration management

Examples of "Configuration Management" implementations		
vRealize Automation SaltStack Config 	Config: Template	nginx: service.running: - enable: True

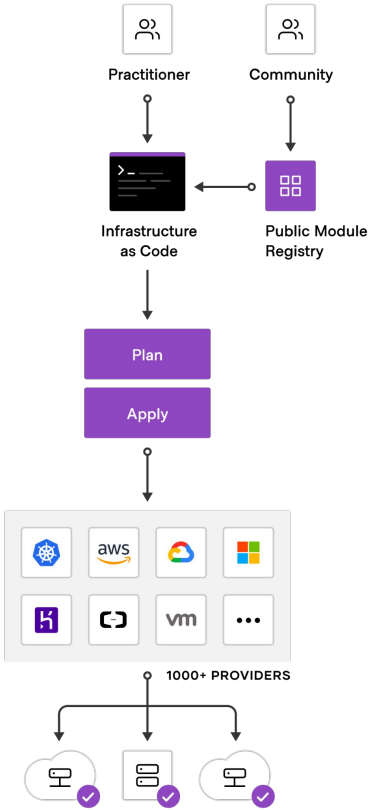
Puppet 	Config: Manifest	class running_service {
		service { 'nginx':
		ensure => 'running',
		}

Chef 	Config: Recipe	service "nginx" do supports :status => true action :start End
Ansible 	Config: Playbook	handlers: - name: start nginx service: name: nginx state: present

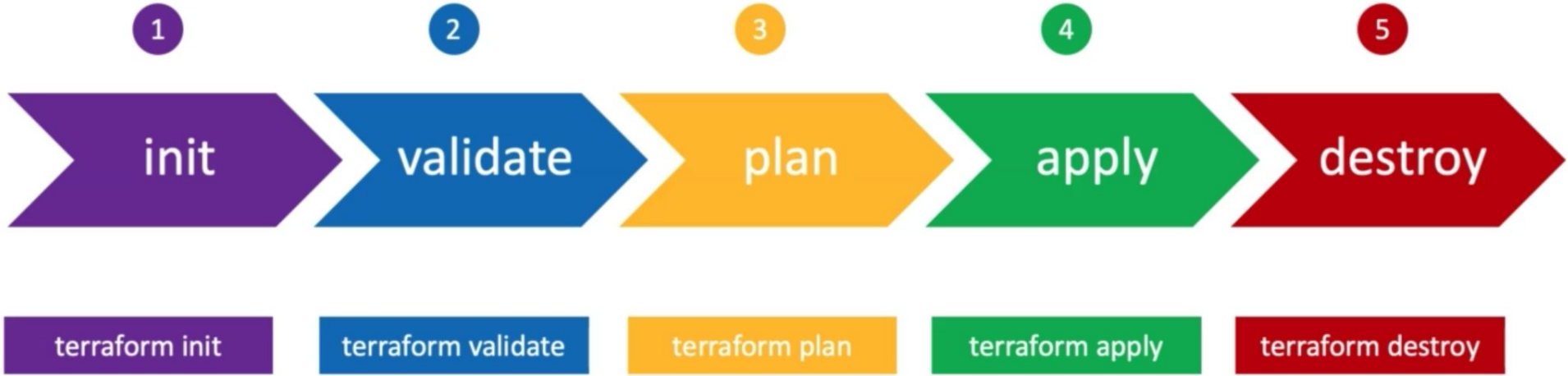
Templating tools

Terraform

¿Por qué Terraform?



Terraform Workflow



Terraform Workflow