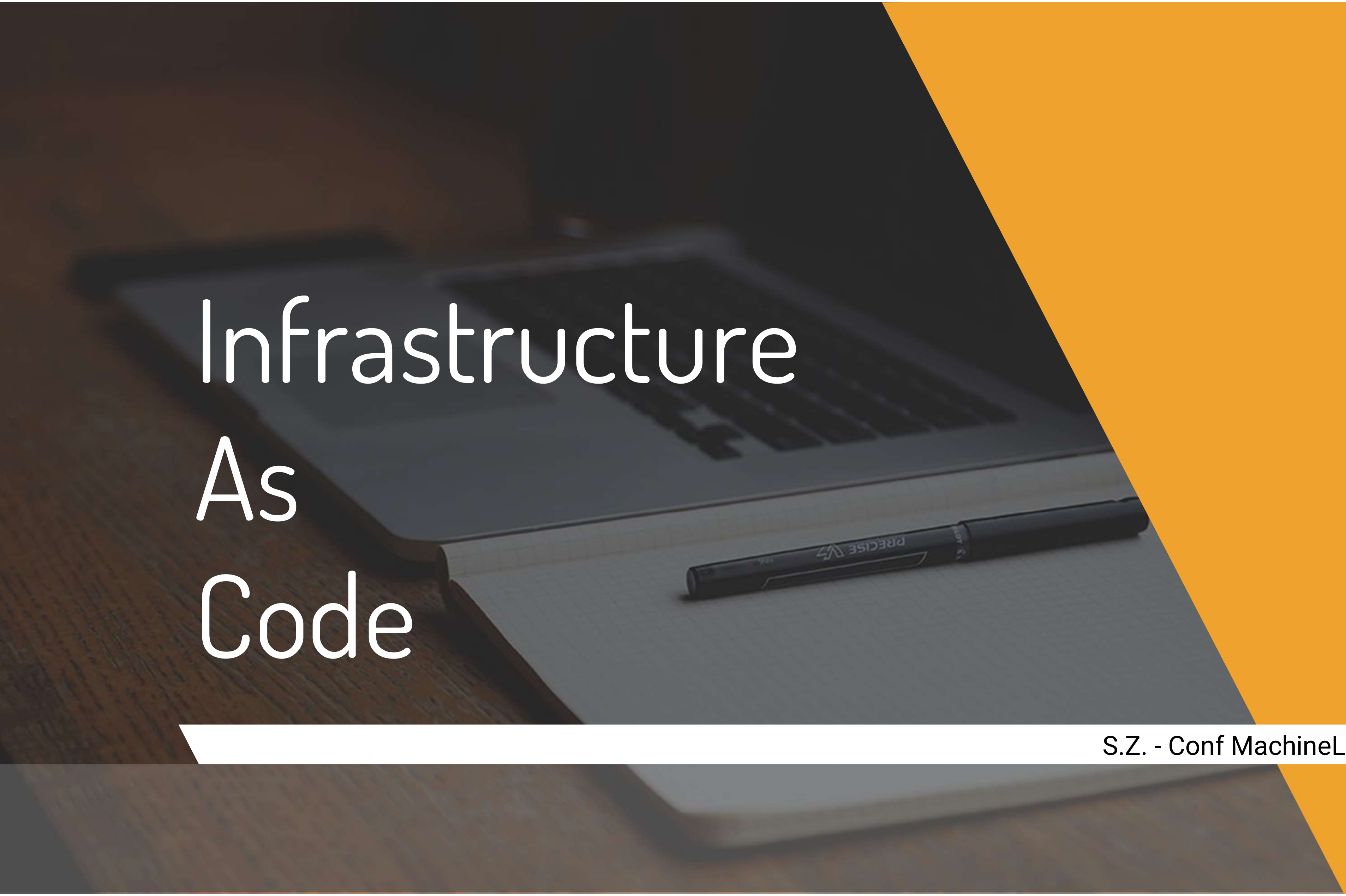
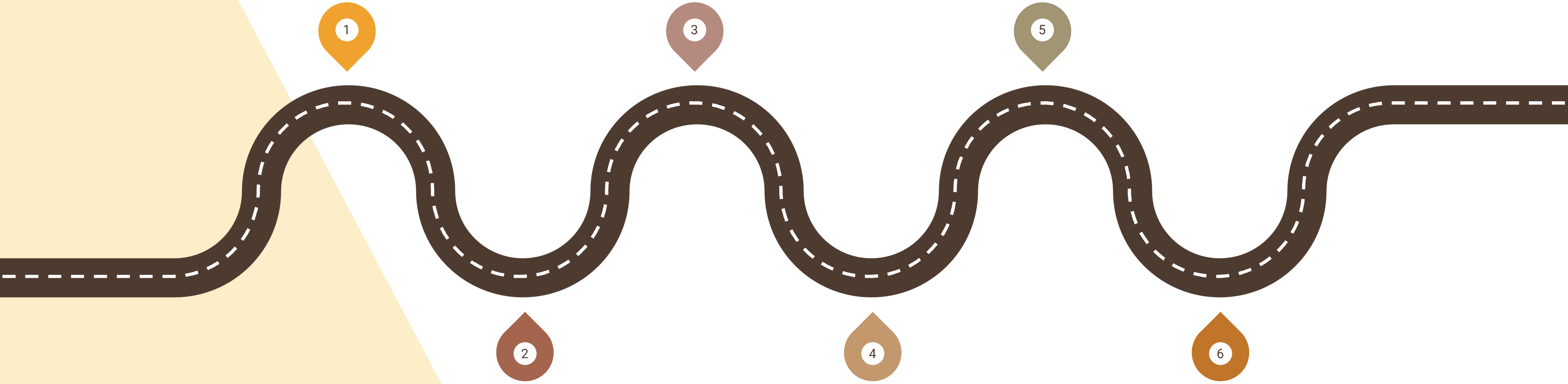


Infrastructure As Code

S.Z. - Conf MachineLearnia – 07-01-22

IaC



Setup initial
Infrastructure

Maintenance
Infrastucture

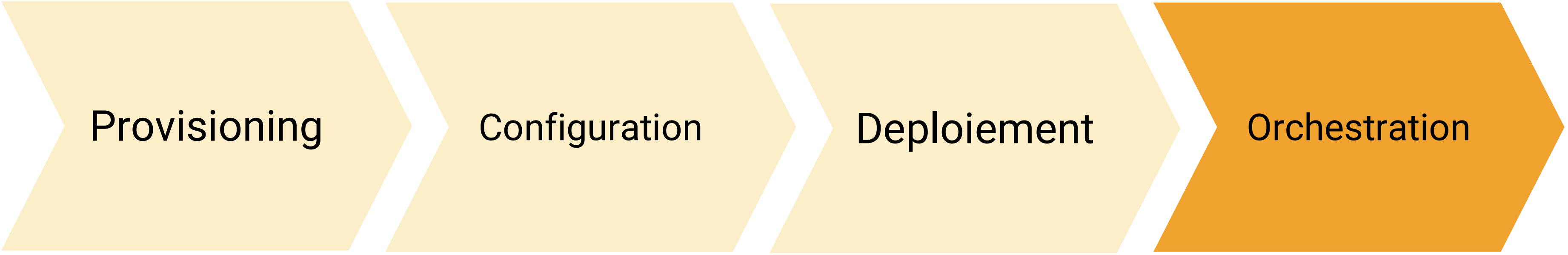
Setup initial
applications

Maintenance
applications

Nouveau setup
Infrastucture

Nouveau setup
applications

IaC



Provisioning

Configuration

Deploiement

Orchestration

IaC

Avant l’automatisation du Devops

Setup

- Setup serveurs
- Users
- Networking
- Firewall
- Softwares / libs
- BDD
- Etc.

Maintenance

- Update versions
- Nouvelles releases
- BDD backups / updates
- Recovery
- Etc..

Coût en temps

Coût en RH

Risques d’erreur

IaC

=> Automatisation du Devops

- Setup
- Maintenance

Automatisation end to end

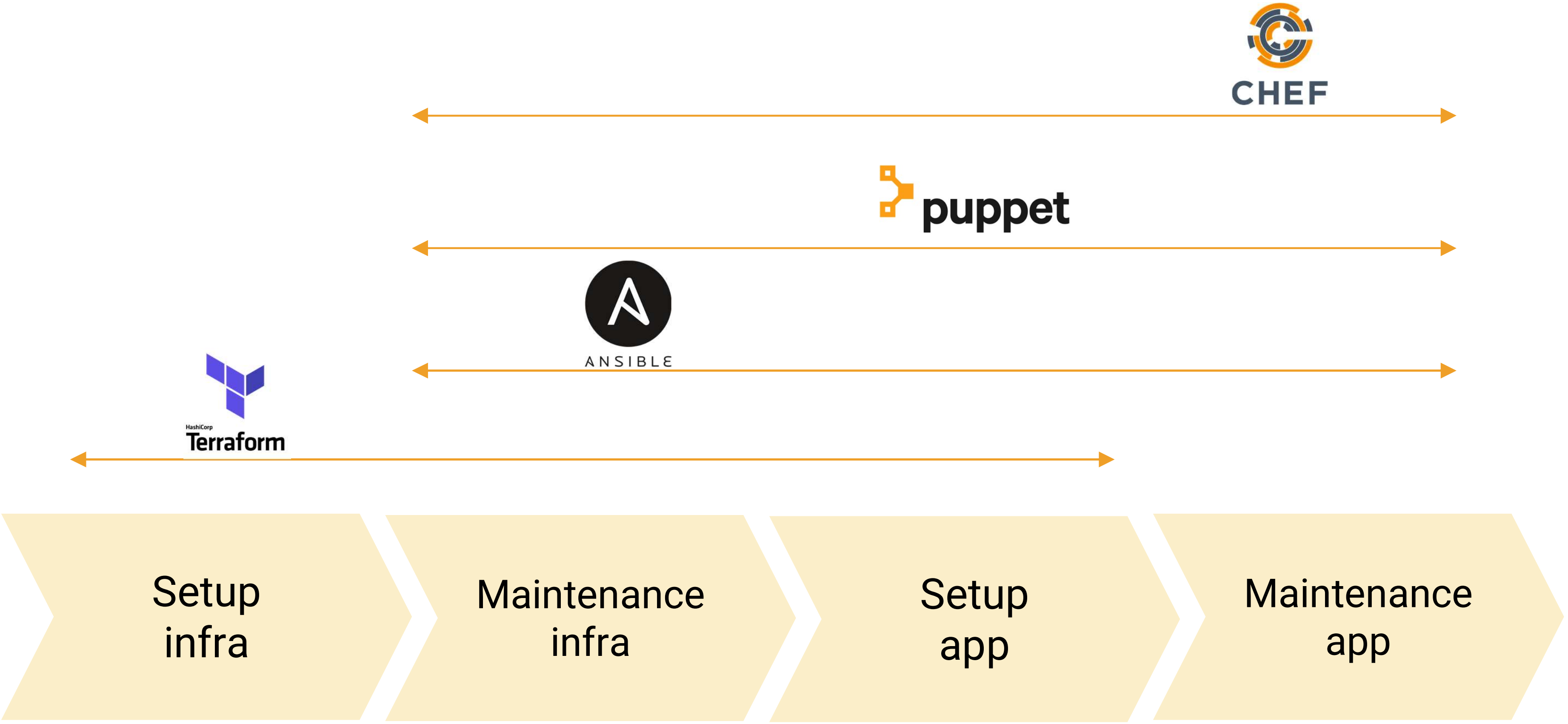
HashiCorp
Terraform

ANSIBLE

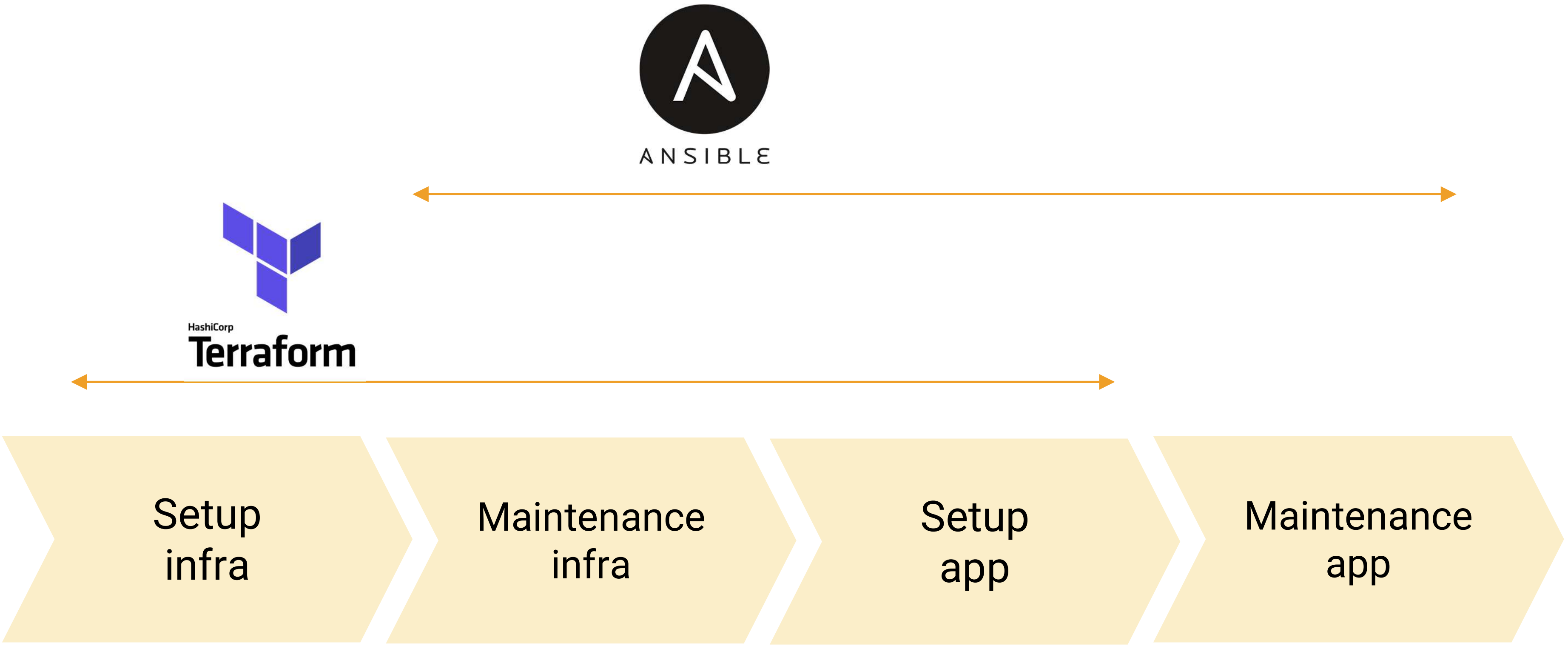
puppet

CHEF

IaC


CHEF
puppet
ANSIBLE
HashiCorp
Terraform
Setup infra
Maintenance infra
Setup app
Maintenance app

IaC



Setup infra

Maintenance infra

Setup app

Maintenance app

IaC

Automatiser le déploiement

Configurer son architecture

Gérer les services sur sa plateforme

Open Source Déclaratif

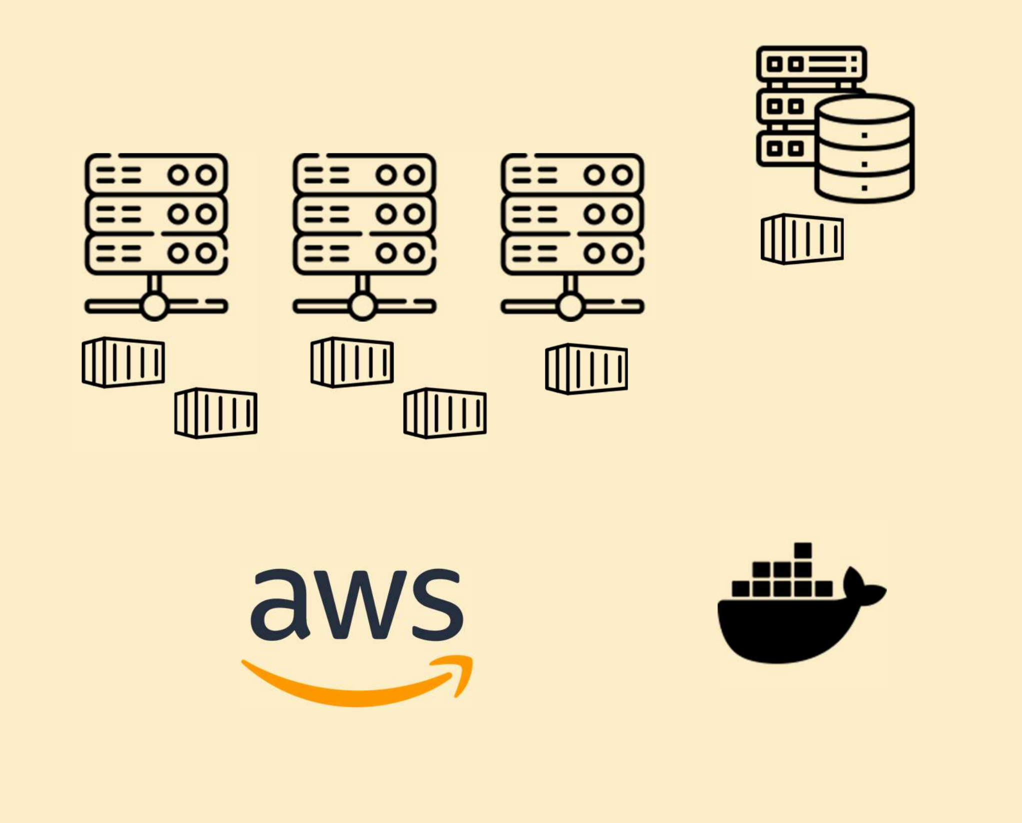
Terraform

Automatiser son déploiement

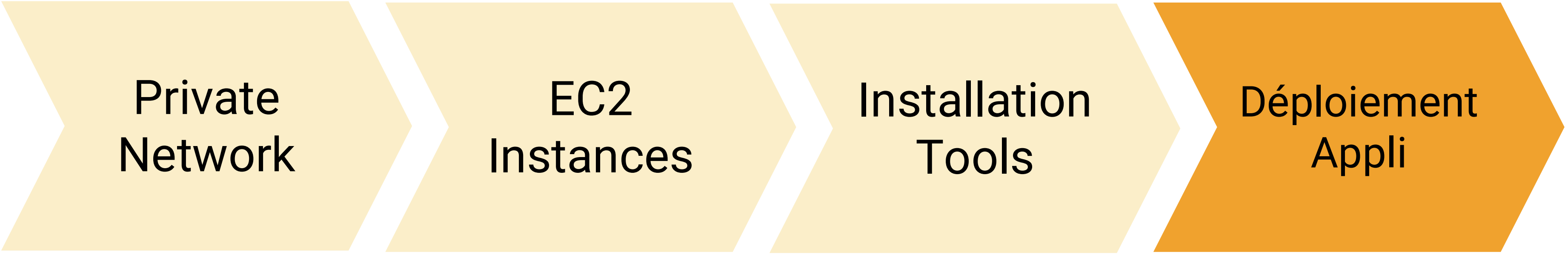
Terraform

Provisioning

- Micro services
  - Conteneurs Docker
- 3 serveurs
- 1 serveur BDD
- AWS



Cas « classique »



Private Network

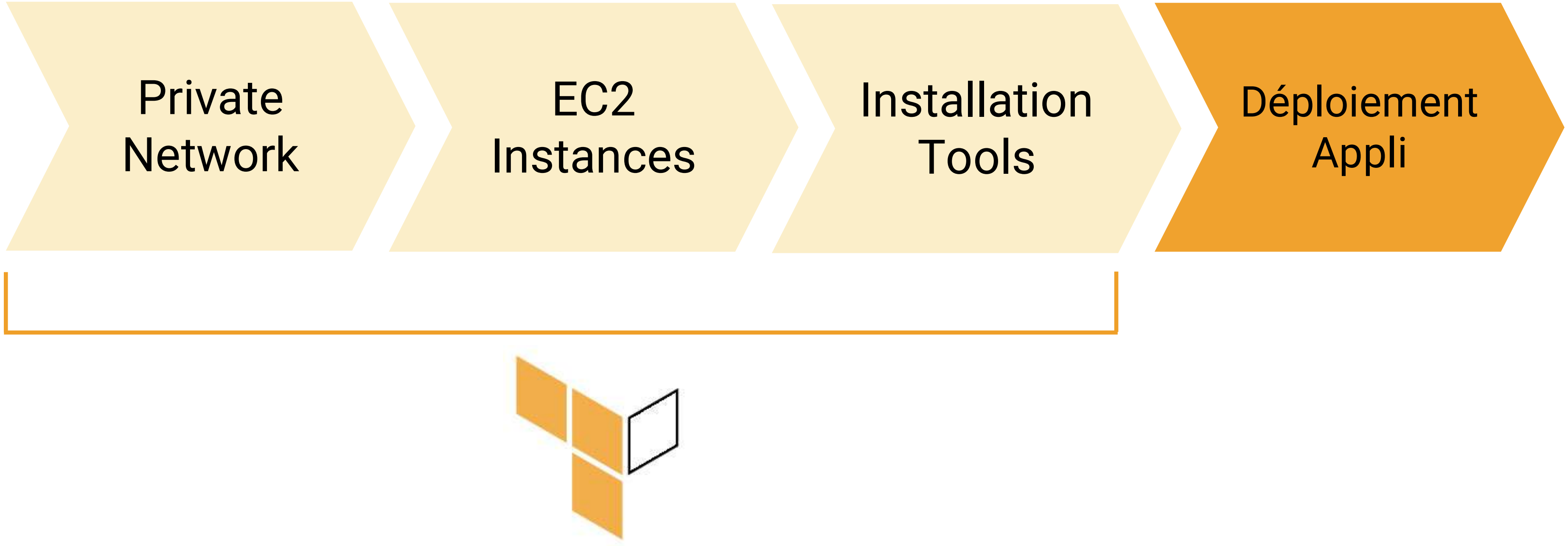
EC2 Instances

Installation Tools

Déploiement Appli

Terraform

Avec Terraform



Private Network

EC2 Instances

Installation Tools

Déploiement Appli

Terraform

DNS

EC2

Network FW

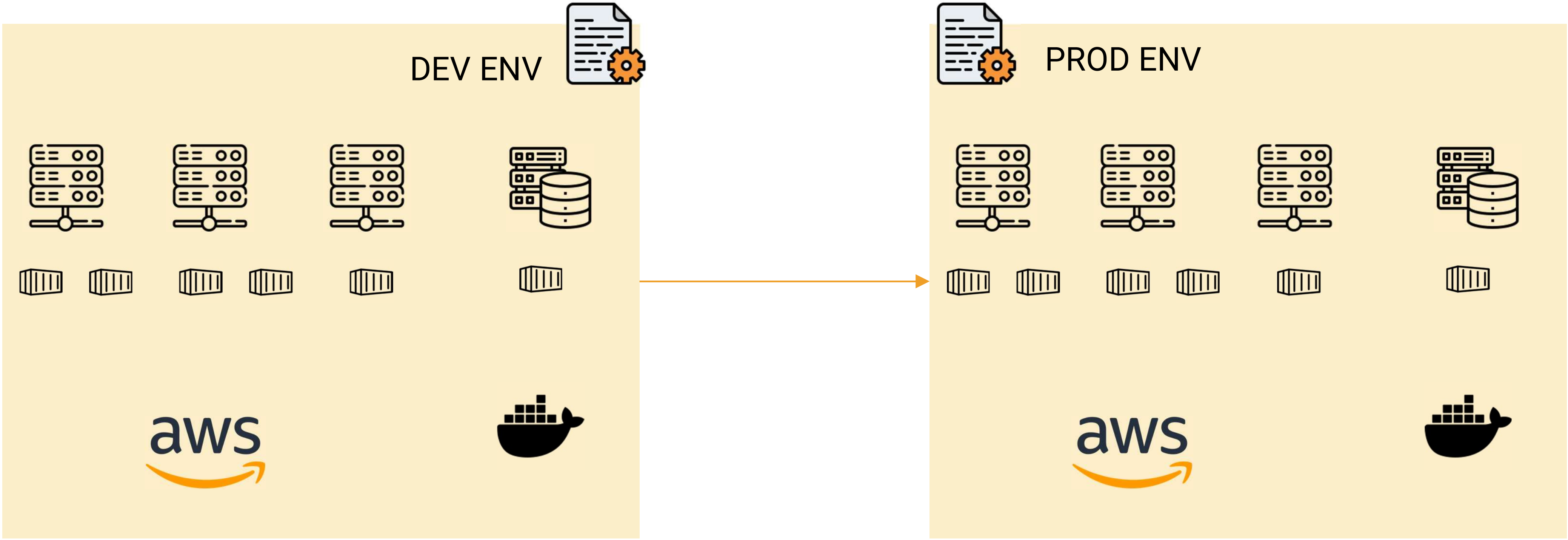
Backups

Kinesis

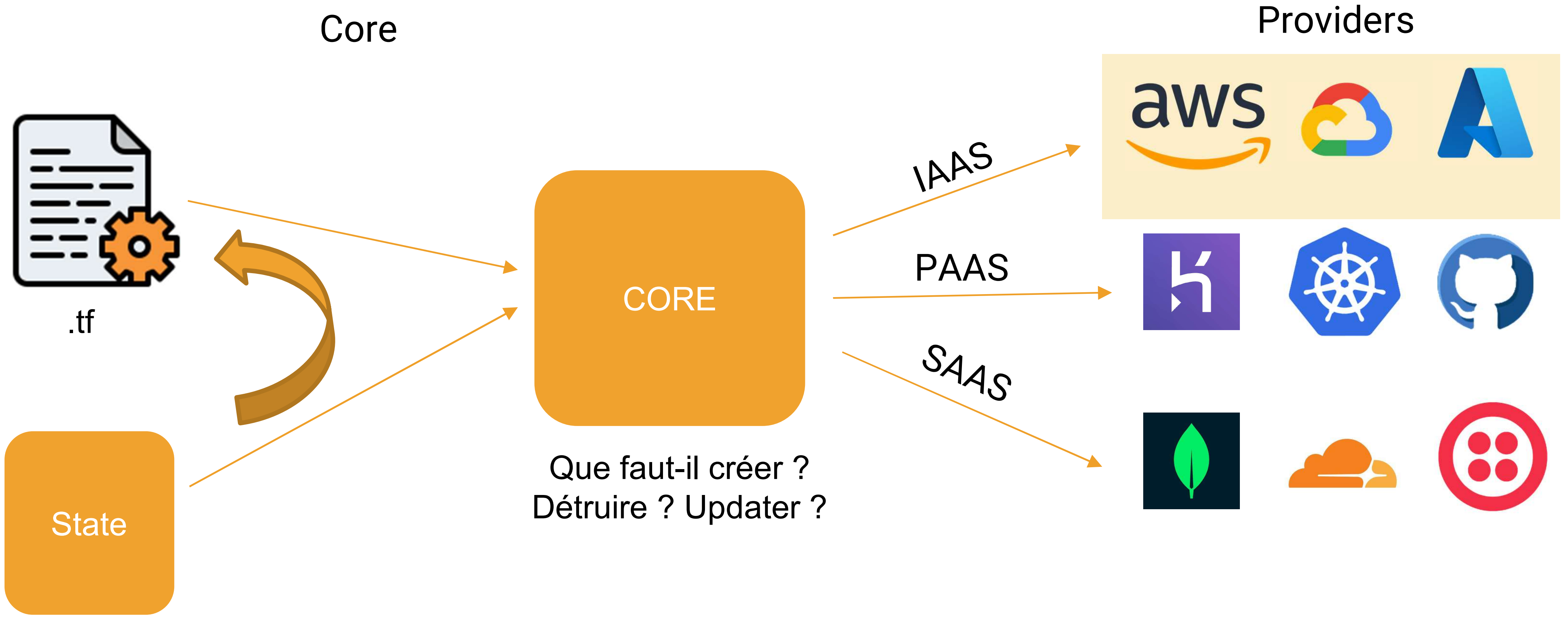
etc

https://registry.terraform.io/providers/hashicorp/aws/latest/docs

Terraform – Réplication automatisée



Terraform - Composants



Core

Providers

.tf

State

CORE

IAAS

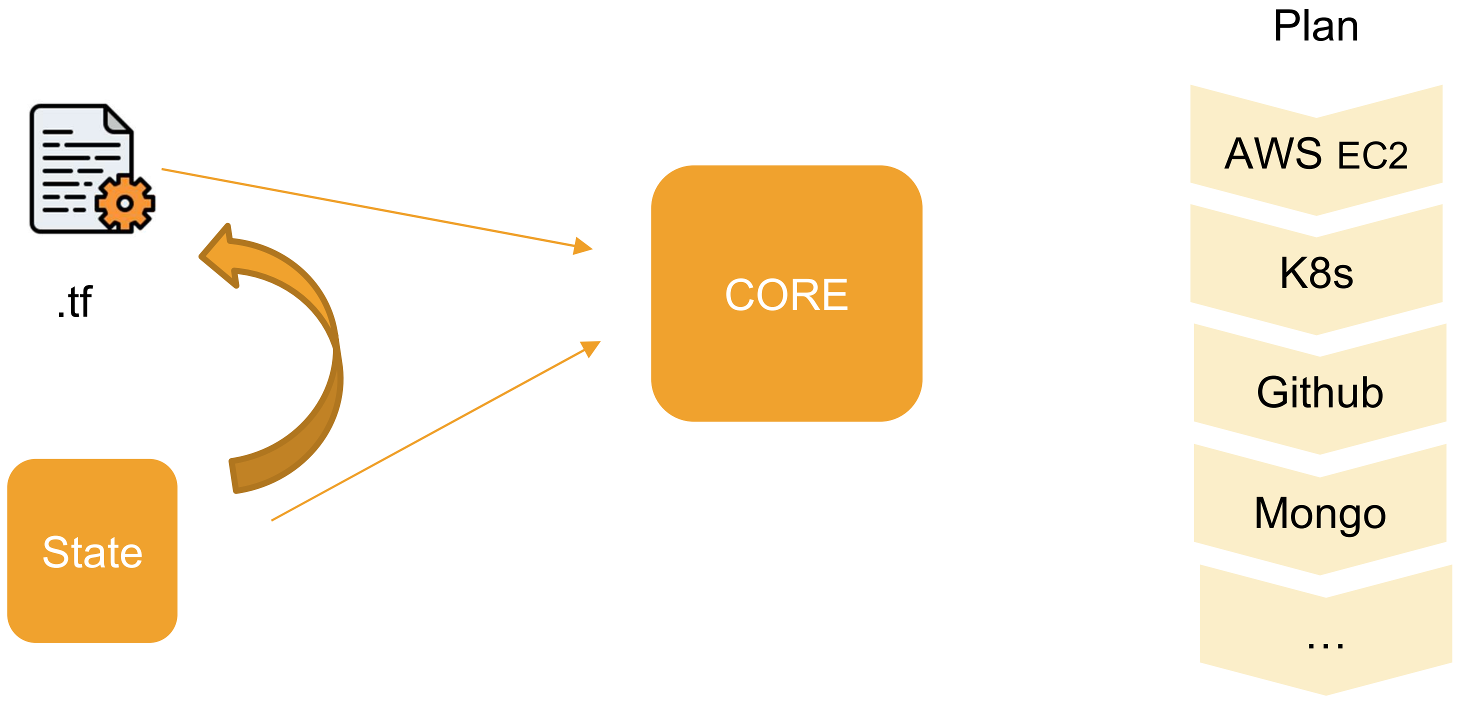
PAAS

SAAS

Que faut-il créer ?
Détruire ? Updater ?

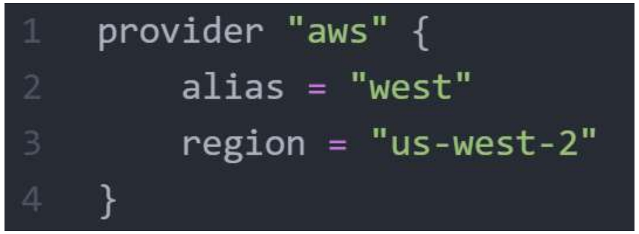
~100 providers + community

Terraform



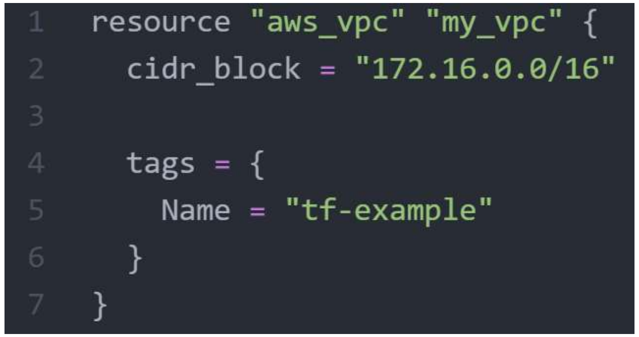
Terraform – .tf files

Provider configuration



```
provider "aws" {
    alias = "west"
    region = "us-west-2"
}
```

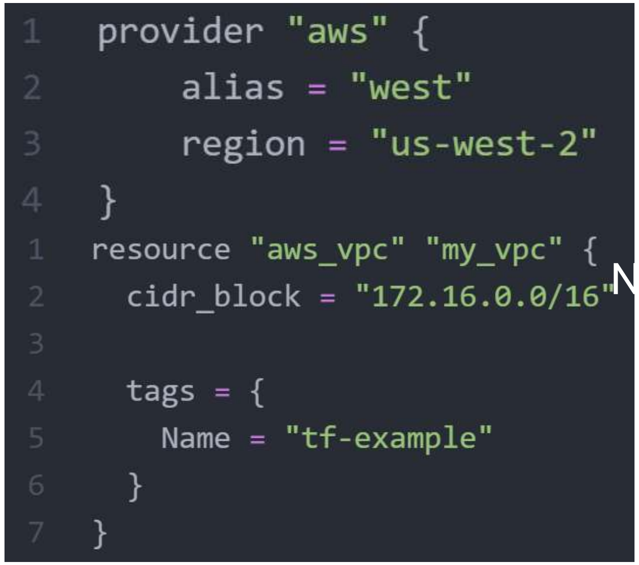
VPC creation



```
resource "aws_vpc" "my_vpc" {
  cidr_block = "172.16.0.0/16"

  tags = {
    Name = "tf-example"
  }
}
```

Terraform - Stages

Current state



```
provider "aws" {
    alias = "west"
    region = "us-west-2"
}
resource "aws_vpc" "my_vpc" {
  cidr_block = "172.16.0.0/16"

  tags = {
    Name = "tf-example"
  }
}
```

Commandes

Refresh

Récupère le state actuel (remote)

Plan

Preview des modifications prévues

Apply

Mise en place effective des modifications

Destroy

Destructions des ressources et de l'infra

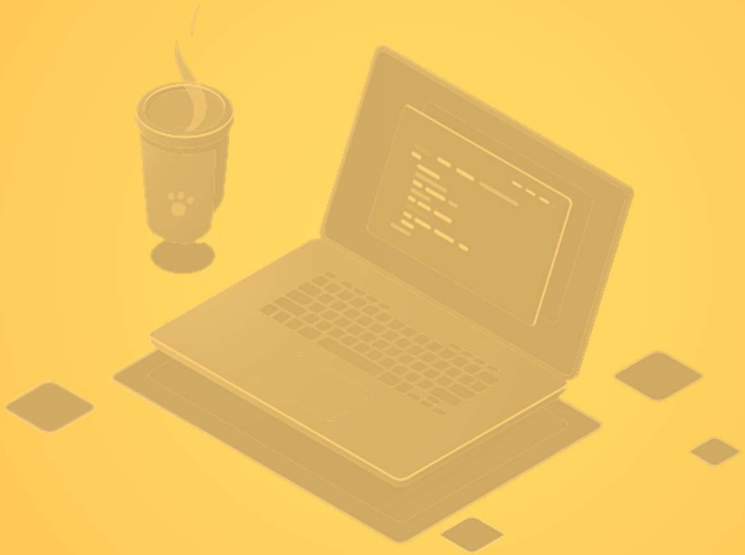
Infrastructure As code

HashiCorp Terraform

ANSIBLE

Pulumi

Infrastructure As code

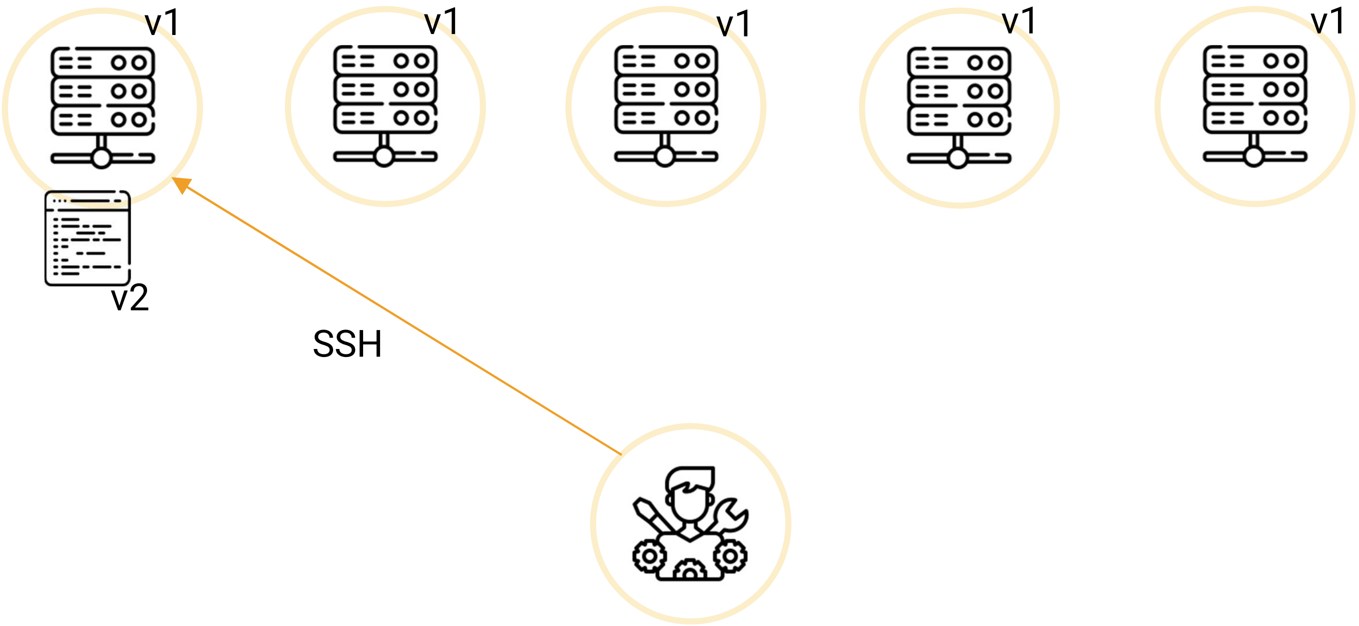


It’s Demo Time !

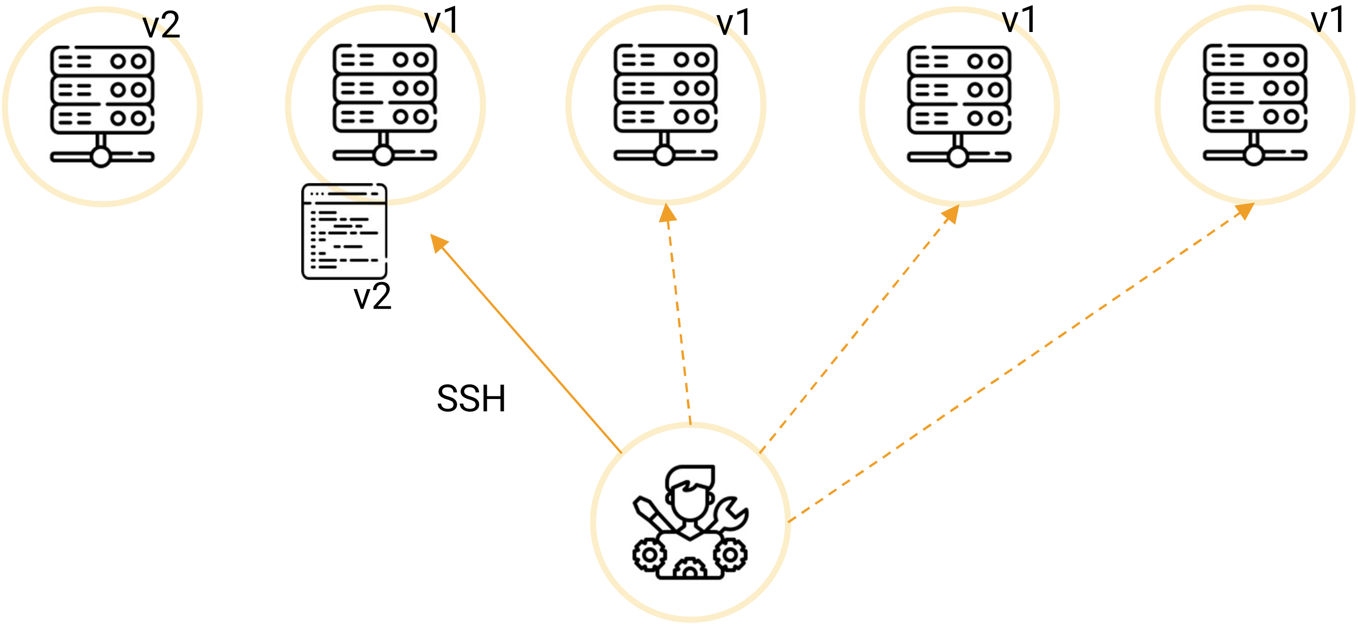
Ansible

Automatiser les tâches IT

Ansible - Updates



Ansible



Ansible – Tâches répétitives

- User creation
- Group management
- Updates
- Backups
- Firewall rules
- Weekly reboot

Ansible

- Plus efficace
- Moins chronophage
- Moins de risques d’erreurs humaines
- Agentless

Local

Tâches exécutées à partir de sa propre machine

Single steps

Config / install / deploiement = 1 fichier YAML

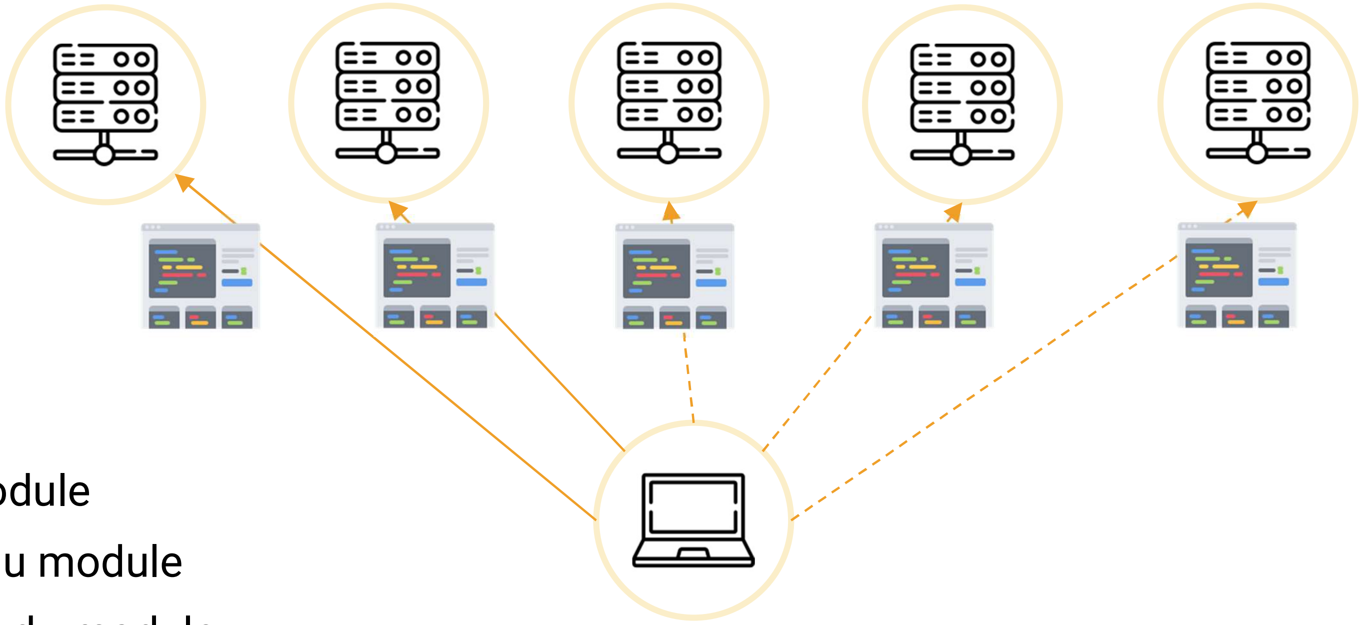
Ré-utilisable

1 script peut être réutilisé sur plusieurs environnements

Moins d’erreurs

Moins de manipulations humaines, moins de scripts

Ansible - Modules



- Push du module
- Exécution du module
- Destruction du module

Ansible – Modules

Créer 1 user

Installation Apache

Créer 1 vhost

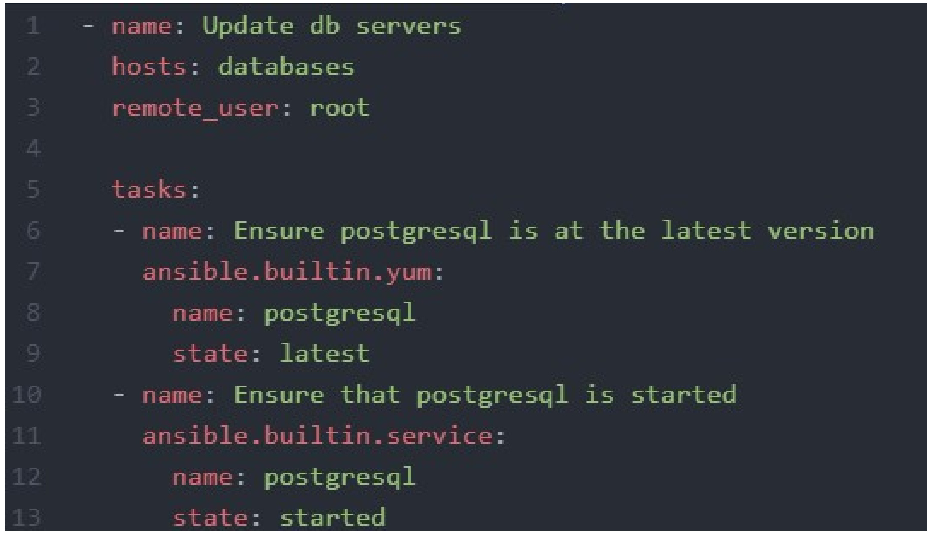
Start Apache

Firewall rules

Création backup

https://docs.ansible.com/ansible/2.9/modules/modules_by_category.html

Ansible – Playbooks



```yaml
- name: Update db servers
  hosts: databases
  remote_user: root

  tasks:
  - name: Ensure postgresql is at the latest version
    ansible.builtin.yum:
      name: postgresql
      state: latest
  - name: Ensure that postgresql is started
    ansible.builtin.service:
      name: postgresql
      state: started
```

Nom du module

Arguments

Tâches

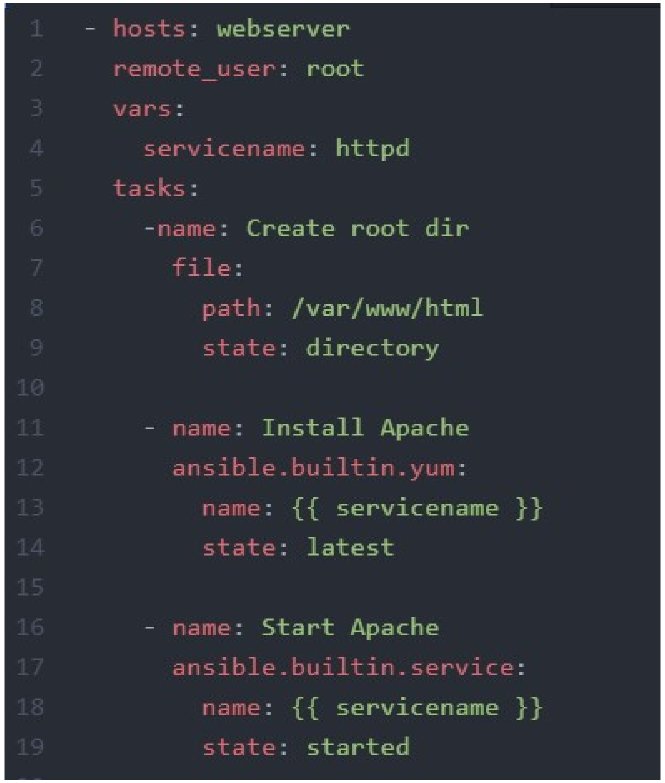
Ansible – Playbooks

Quel hôte ?

Quel user ?

Quelles tâches ?

« Play » =>



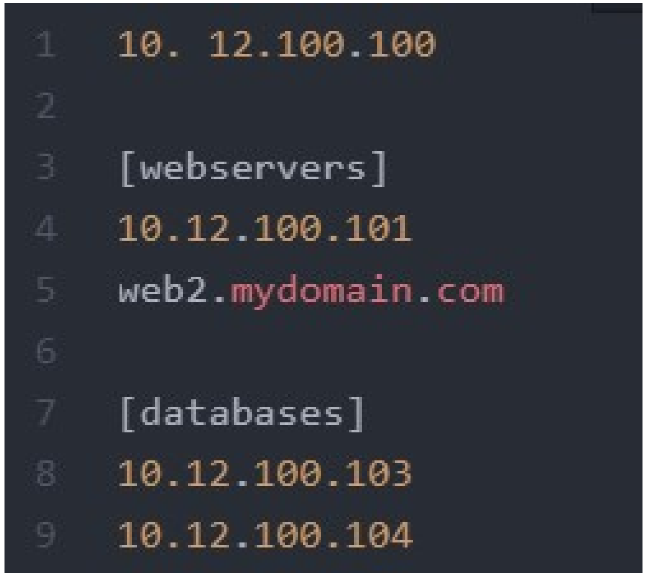
```yaml
- hosts: webserver
  remote_user: root
  vars:
    servicename: httpd
  tasks:
    -name: Create root dir
      file:
        path: /var/www/html
        state: directory

    - name: Install Apache
      ansible.builtin.yum:
        name: {{ servicename }}
        state: latest

    - name: Start Apache
      ansible.builtin.service:
        name: {{ servicename }}
        state: started
```

Ansible - Playbooks

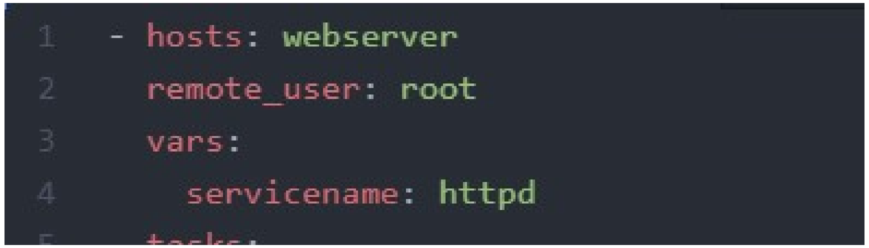
Playbook

- Quels « Plays »
- Dans quel ordre ?
- Quand et sur quelle machine ?
- Quels modules exécuter ?

Ansible – Inventory

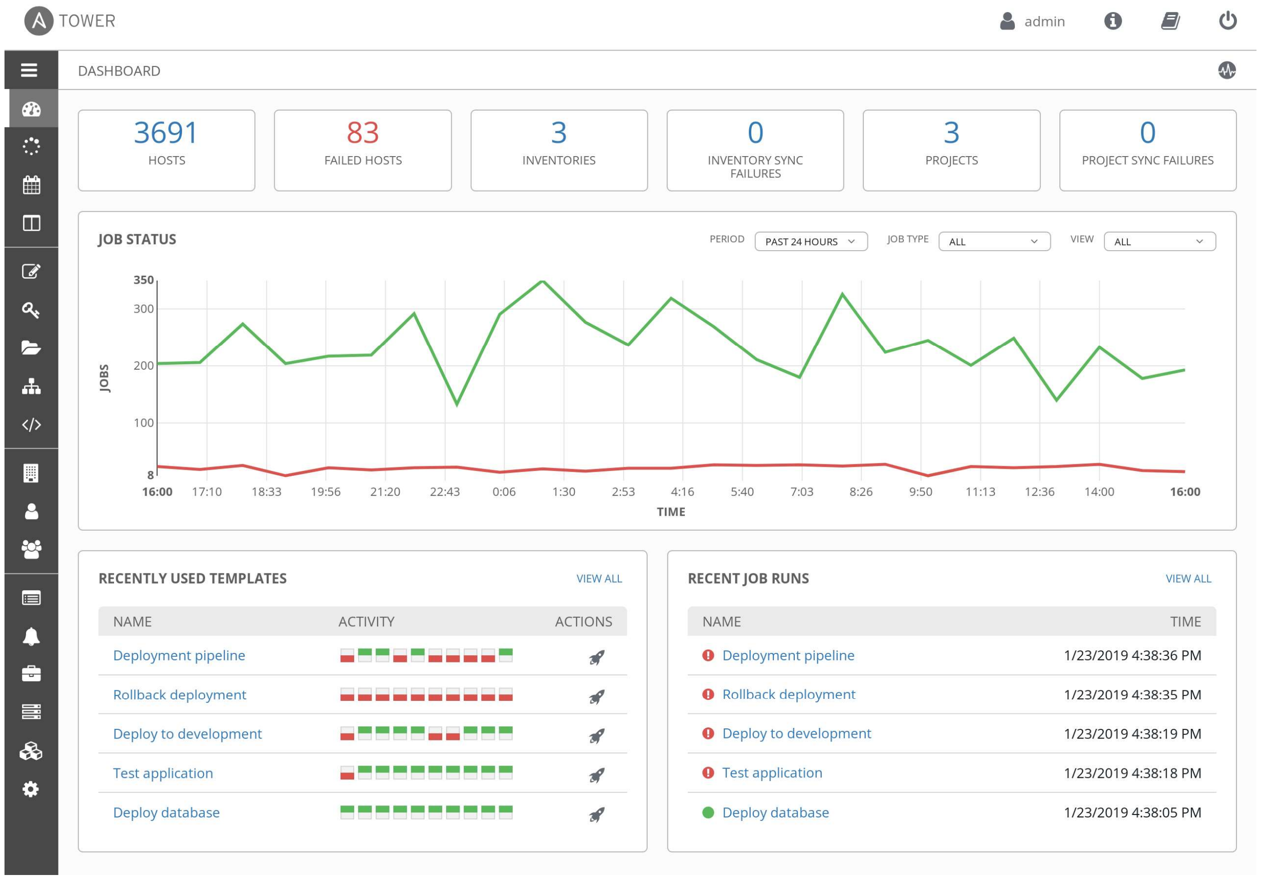


```
10. 12.100.100

[webservers]
10.12.100.101
web2.mydomain.com

[databases]
10.12.100.103
10.12.100.104
```



```
- hosts: webserver
  remote_user: root
  vars:
    servicename: httpd
  tasks:
```

Ansible – Tower



3691 HOSTS	83 FAILED HOSTS	3 INVENTORIES	0 INVENTORY SYNC FAILURES	3 PROJECTS	0 PROJECT SYNC FAILURES

PERIOD: PAST 24 HOURS   JOB TYPE: ALL   VIEW: ALL

RECENTLY USED TEMPLATES   VIEW ALL

NAME	ACTIVITY	ACTIONS
Deployment pipeline	🟥🟩🟩🟥🟩🟥🟥🟥🟥🟩	
Rollback deployment	🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥	
Deploy to development	🟥🟩🟩🟩🟩🟥🟥🟩🟩🟩	
Test application	🟥🟩🟩🟩🟩🟩🟩🟩🟩🟩	
Deploy database	🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩	

RECENT JOB RUNS   VIEW ALL

NAME	TIME
Deployment pipeline	1/23/2019 4:38:36 PM
Rollback deployment	1/23/2019 4:38:35 PM
Deploy to development	1/23/2019 4:38:19 PM
Test application	1/23/2019 4:38:18 PM
● Deploy database	1/23/2019 4:38:05 PM

Pulumi

Automatiser en Python, JS, Ruby, etc.

Pulumi

JSON

YAML

HCL

Conditions ?

Loops ?

Functions ?

Re-use code ?

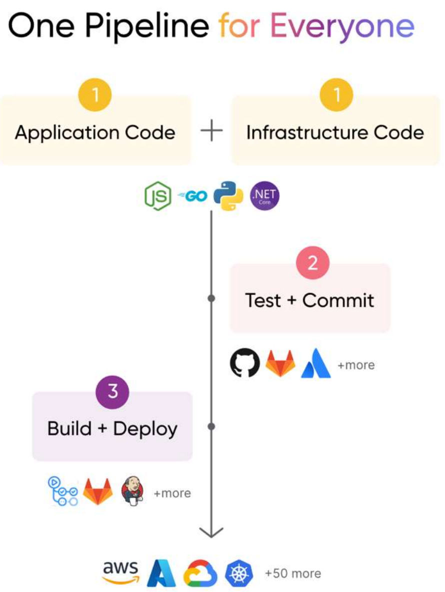
Pulumi

Pulumi

GO
node
JS
.NET
Core

https://www.pulumi.com/docs/intro/languages/

Pulumi



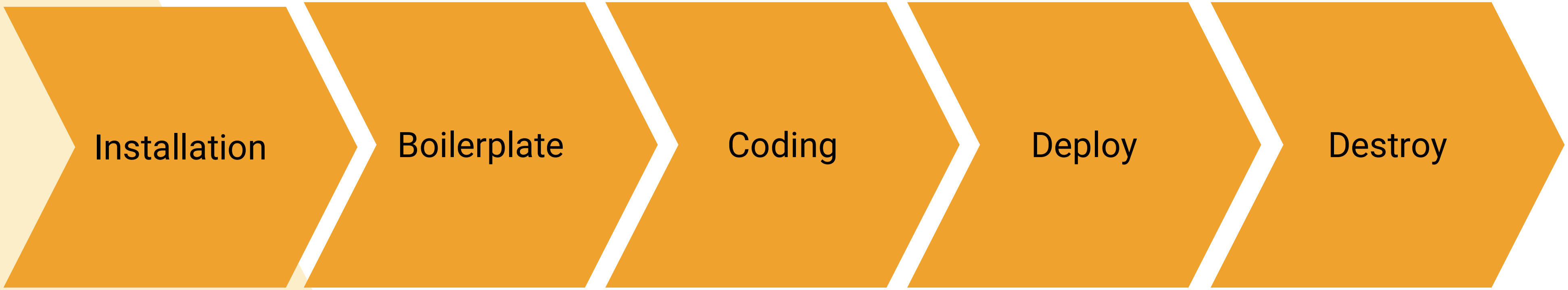
Ecosystème connu

Tests unitaires / d’intégration

CI / CD

https://www.pulumi.com/docs/intro/languages/

Pulumi



Installation

Boilerplate

Coding

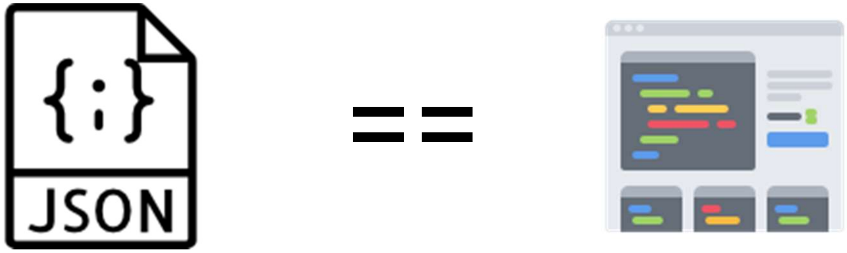
Deploy

Destroy

https://www.pulumi.com/

Pulumi - states

- Que faut-il créer ?
- Que faut-il changer ?
- Que faut-il supprimer ?

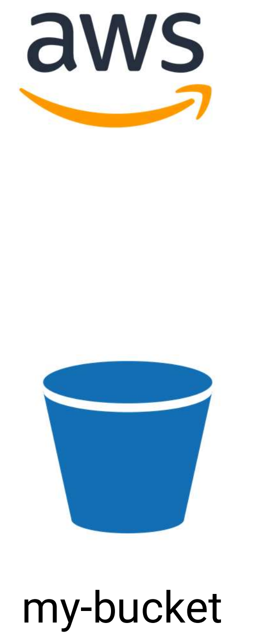


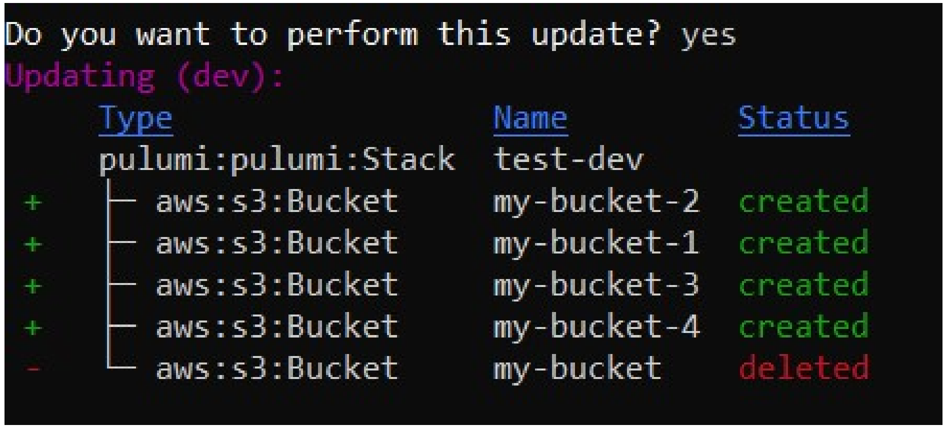
State actuel

State recherché

Pulumi - states

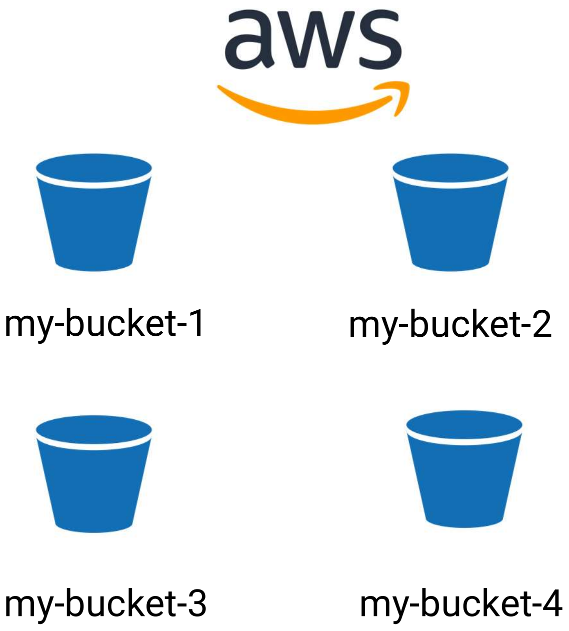
State actuel





```
Do you want to perform this update? yes
Updating (dev):
     Type                 Name         Status
     pulumi:pulumi:Stack  test-dev
 +   ├─ aws:s3:Bucket     my-bucket-2  created
 +   ├─ aws:s3:Bucket     my-bucket-1  created
 +   ├─ aws:s3:Bucket     my-bucket-3  created
 +   ├─ aws:s3:Bucket     my-bucket-4  created
 -   └─ aws:s3:Bucket     my-bucket    deleted
```

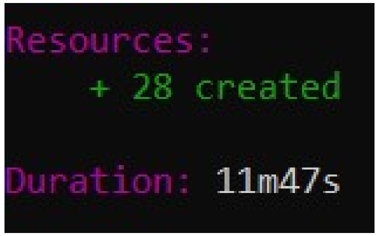
State recherché



Pulumi – EKS cluster

Recherché

Amazon EKS



```
Resources:
    + 28 created

Duration: 11m47s
```

Plan

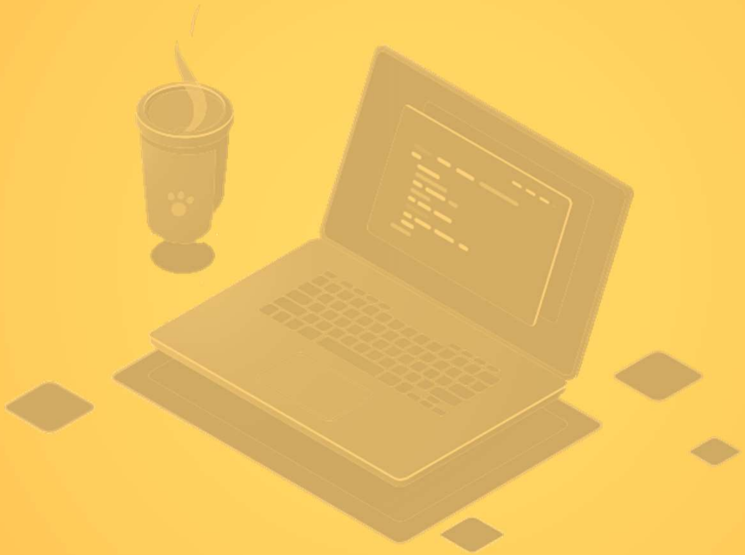
Amazon EKS

AWS IAM

VPC

AWS EC2

Infrastructure As code



It's Demo Time !

Bonus

Terraform++

Terraform - Bonus

- Remote state
- Lock state
- CI/CD
- Backup state

Infrastructure As code

HashiCorp Terraform

ANSIBLE

Pulumi