

HashiCorp

Terraform laC

Infrastructure as Code



4 parties

Les bases de Terraform

Implémentation d'un provider 

Déployer une “google cloud function” 

Terraform en entreprise

Les bases de Terraform

Terraform CLI

Configuration files / State file

HCL Configuration

Dépendances

State file

Plan - CRUD

“Blueprints”

Terraform CLI

Terraform <command>

<command>:

Plan: prévisualisation des changements d'infrastructure

Apply: applique les changements

Destroy: détruit l'infrastructure créée

Init: installe les plugins nécessaire à la modification d'infrastructure

Import: importe des ressources existantes dans la configuration Terraform

State: affiche ou modifie l'état courant de l'infrastructure gérée par Terraform

...

<https://developer.hashicorp.com/terraform/cli>

Configuration files / State file



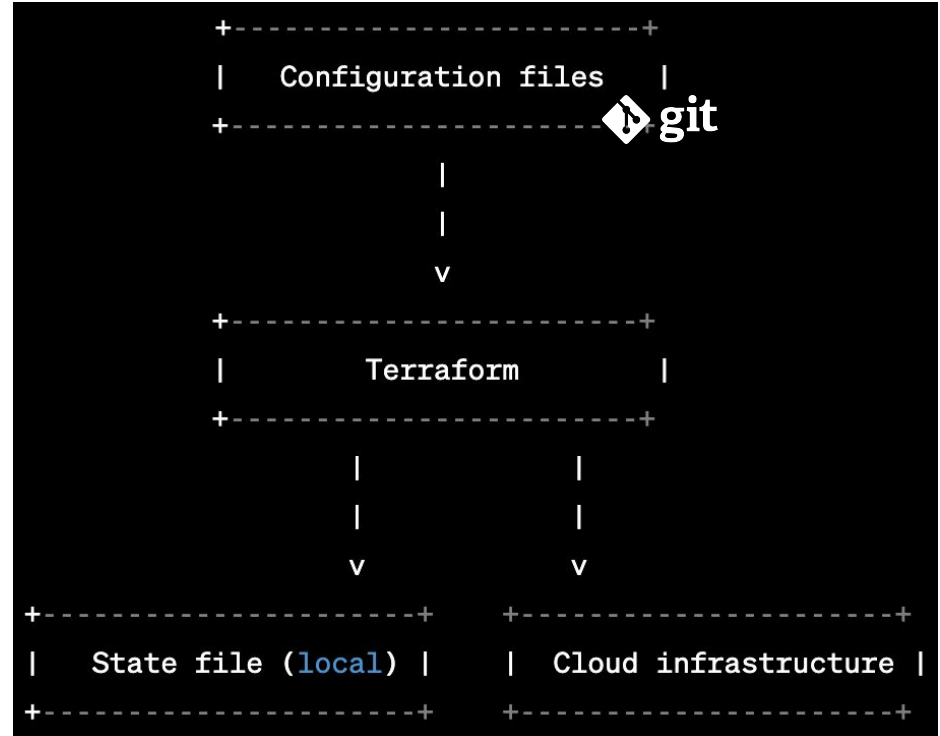
Voici un schéma qui représente la relation entre le fichier d'état, la configuration Terraform et l'infrastructure cloud :

La configuration Terraform est utilisée pour créer et gérer l'infrastructure cloud.

Terraform crée également un fichier d'état local qui contient des informations sur l'infrastructure créée et gérée.

Ce fichier d'état est utilisé pour suivre les modifications apportées à l'infrastructure au fil du temps, et pour garantir que les modifications apportées par les utilisateurs sont cohérentes avec l'état actuel de l'infrastructure.

Les modifications sont ensuite appliquées à l'infrastructure cloud et le fichier d'état est mis à jour en conséquence.



HCL Configuration

```
locals {  
  computer_models = [  
    "First",  
    "Second",  
    "Third",  
  ]  
}  
  
resource "computer-database_company" "my_company" {  
  id = "cotf"  
  name = "My Terraformed Company"  
  computer_models = toset([ for model_name in local.computer_models:  
    {  
      id = format("cotf%s", lower(model_name))  
      name = model_name  
      release = 2023  
    }  
  ])  
}
```

Hashicorp Configuration Language

<https://github.com/hashicorp/hcl>

<https://developer.hashicorp.com/terraform/language>

Langage déclaratif en “blocs”

Logique fonctionnelle et valeurs immuables
(english: “immutable”)

Blocs:

Resource: crée et gère une ressource
d'infrastructure

Data: récupère des données externes sur une
ressource d'infrastructure non gérée

locals: définit une variable locale

variable: définit une variable (utile pour un module)

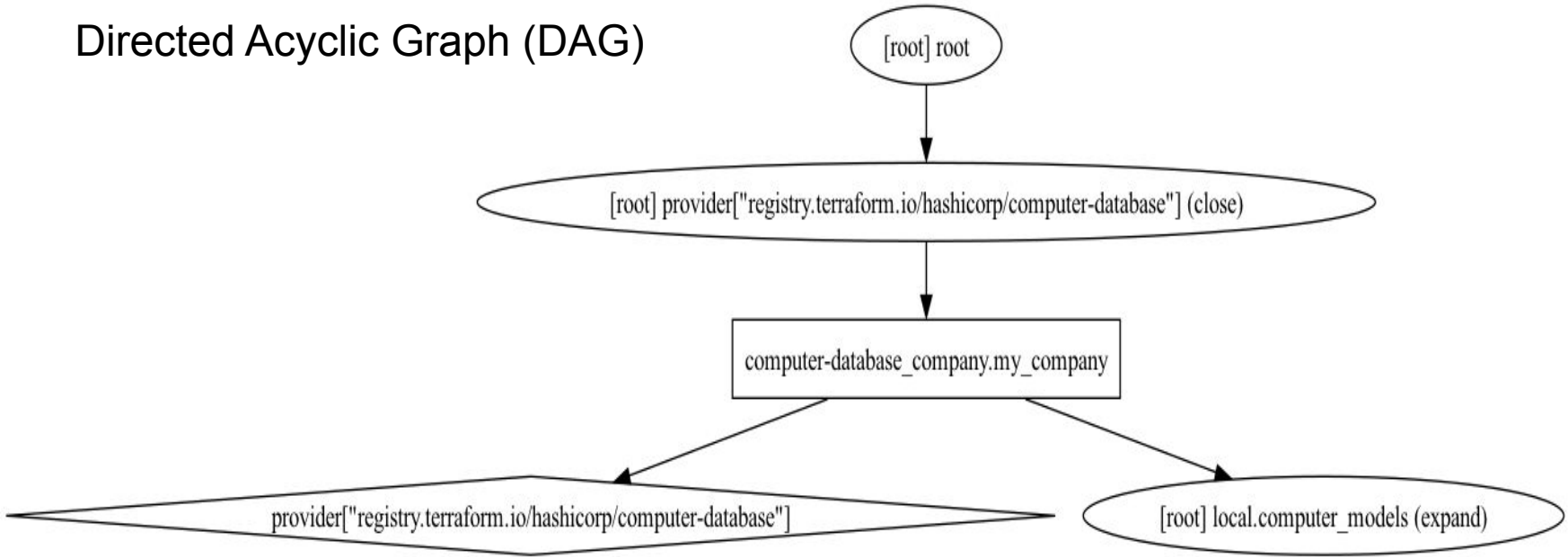
output: expose une valeur

module: permet de réutiliser une configuration hcl

provider: configure le fournisseur de cloud/service 6

Dépendances

Directed Acyclic Graph (DAG)



terraform graph | dot -Tpng > graph.png

State file

Un simple fichier JSON*

```
$> terraform state list
```

```
computer-database_company.my_company
```

```
"version": 4,  
"terraform_version": "1.4.0",  
"serial": 5,  
"lineage": "af6c9a36-f361-8caa-19c4-bf55d9c1e703",  
"outputs": {},  
"resources": [  
  {  
    "mode": "managed",  
    "type": "computer-database_company",  
    "name": "my_company",  
    "provider": "provider[\"registry.terraform.io/hashicorp/computer-database\"]",  
    "instances": [  
      {  
        "schema_version": 0,  
        "attributes": {  
          "computer_models": [  
            {  
              "id": "cotffirst",  
              "name": "First",  
              "release": "2023"  
            },  
            {  
              "id": "cotfsecond",  
              "name": "Second",  
              "release": "2023"  
            },  
            {  
              "id": "cotfthird",  
              "name": "Third",  
              "release": "2023"  
            }  
          ],  
          "id": "cotf",  
          "location": "global",  
          "name": "My Terraformed Company"  
        },  
        "sensitive_attributes": []  
      }  
    ]  
  },  
  {  
    "check_results": null  
  }  
]
```

JSON*: JavaScript Object Notation, more commonly known by the acronym JSON, is an open data interchange format that is both human and machine-readable.


```
~ update in-place
-/+ destroy and then create replacement
```

PLAN - CRUD

Terraform will perform the following actions:

```
# module.cf_helloworld.google_cloudfunctions_function.function will be updated in-place
~ resource "google_cloudfunctions_function" "function" {
    id          = "projects/oxyl-terraform-tn-april-23/locations/europe-we
    name        = "helloworld"
    ~ source_archive_object = "cf_helloworld-zrUI1DQH28DvTIRCBZn3jSm2MQEBE9et3j1vJAZNVU4=.zip"
    # (18 unchanged attributes hidden)
}

# module.cf_helloworld.google_storage_bucket_object.remote_func_archive must be replaced
-/+ resource "google_storage_bucket_object" "remote_func_archive" {
    ~ content_type = "application/zip" -> (known after apply)
    ~ crc32c       = "rCmJvw==" -> (known after apply)
    ~ detect_md5hash = "eMu190Tfx3j6y98nP7LLZg==" -> "different hash" # forces replacement
    - event_based_hold = false -> null
    ~ id              = "oxyl-tn-2023-04-functions-cf_helloworld-zrUI1DQH28DvTIRCBZn3jSm2MQEBE9et3j1vJAZNVU4=.zip" -> (known after apply)
    + kms_key_name    = (known after apply)
    ~ md5hash        = "eMu190Tfx3j6y98nP7LLZg==" -> (known after apply)
    ~ media_link      = "https://storage.googleapis.com/download/storage/v1/b/oxyl-tn-2023-04-functions-cf_helloworld-zrUI1DQH28DvTIRCBZn3jSm2MQEBE9et3j1vJAZNVU4=.zip?generation=1682602411314466&alt=media" -> (known after apply)
    - metadata       = {} -> null
    ~ name           = "cf_helloworld-zrUI1DQH28DvTIRCBZn3jSm2MQEBE9et3j1vJAZNVU4=.zip" -> (known after apply)
    ~ output_name     = "cf_helloworld-zrUI1DQH28DvTIRCBZn3jSm2MQEBE9et3j1vJAZNVU4=.zip" -> (known after apply)
    ~ self_link       = "https://www.googleapis.com/storage/v1/b/oxyl-tn-2023-04-functions-cf_helloworld-zrUI1DQH28DvTIRCBZn3jSm2MQEBE9et3j1vJAZNVU4=.zip" -> (known after apply)
    ~ storage_class   = "STANDARD" -> (known after apply)
    - temporary_hold  = false -> null
    # (2 unchanged attributes hidden)
}
```

Plan: 1 to add, 1 to change, 1 to destroy.

“Blueprints”

<https://cloud.google.com/docs/terraform/blueprints/terraform-blueprints>

Un “Blueprint” (plan de construction) est un ensemble de *modules* terraform.

Un *module* c’est une configuration terraform
avec un ensemble de *variables* configurable en entrée
et un ensemble de valeurs de sortie “*output*”.

Implémentation d'un provider

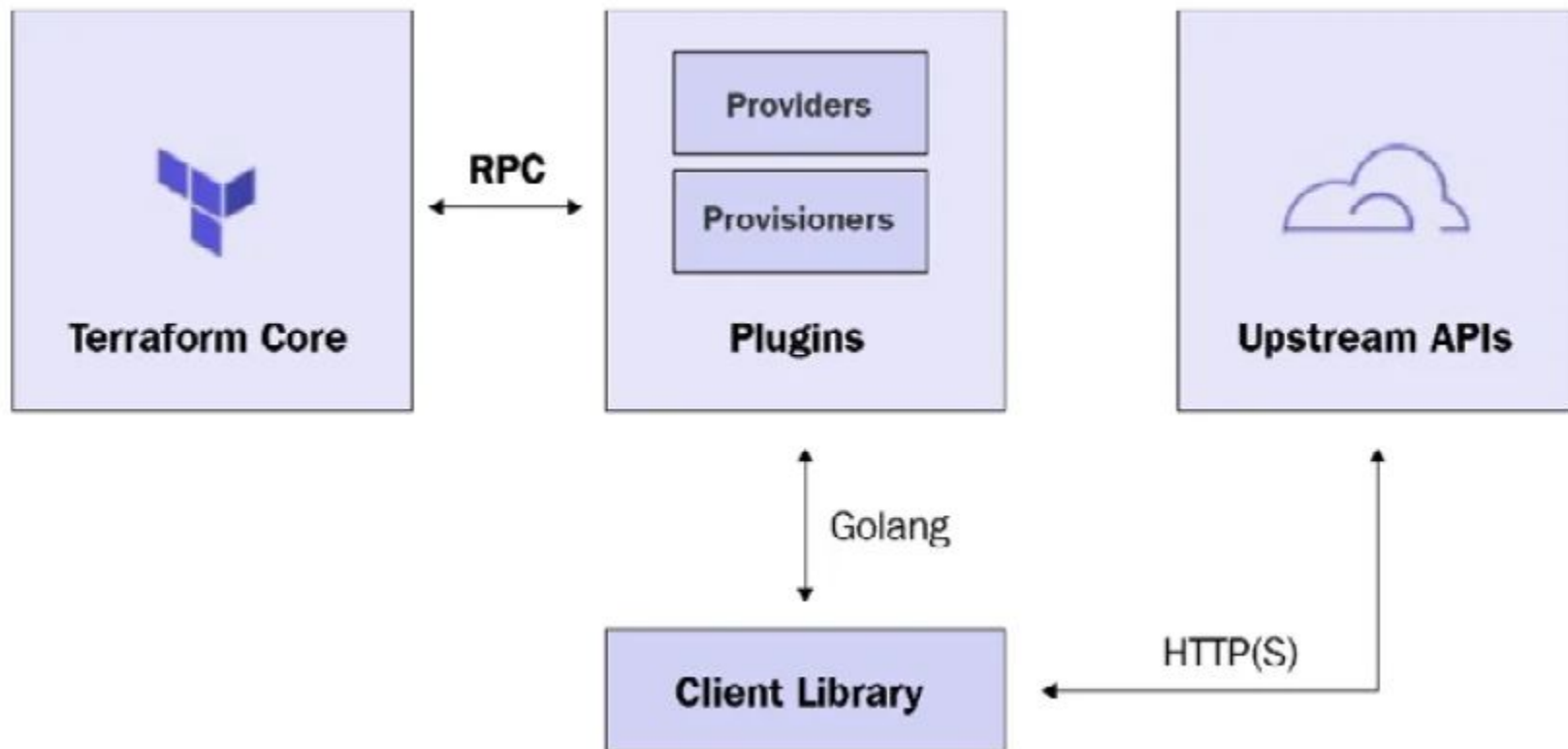
Plugins Terraform

Définition d'un provider

Description d'une "resource"

Mon provider "Computer Database"

Plugins Terraform



Définition d'un provider

Un petit nom qui sera le préfixe des “resources”.

Une manière de le configurer.

Une collection de “resources” et une collection de “data”.

Description d'une "resource"

Un petit nom préfixé par le nom du provider

Un Schéma - Une structure de donnée

"CRUD methods"

Mon provider “Computer Database”



<http://127.0.0.1:8080/api/v1/companies/>

à essayer <https://registry.terraform.io/providers/Mastercard/restapi/latest/docs>

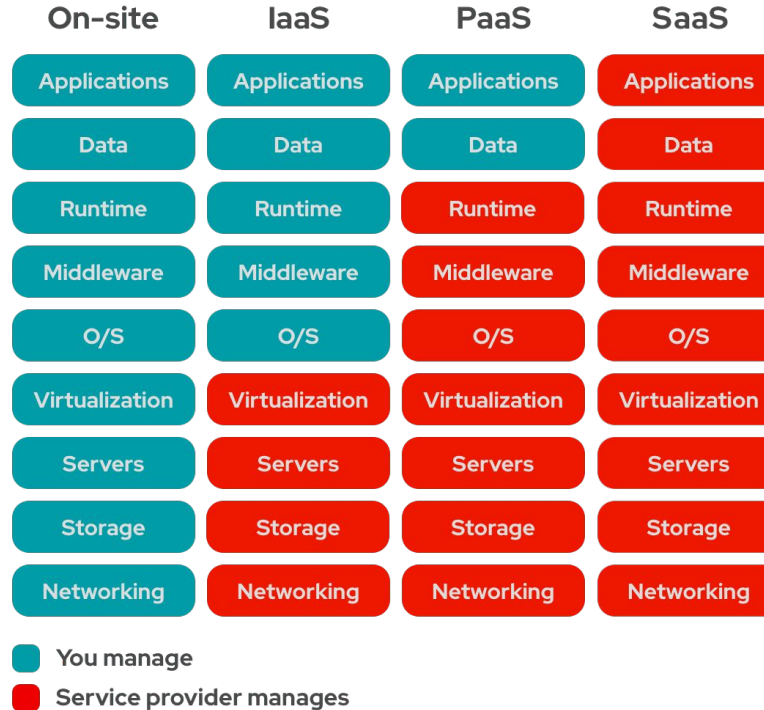
Déployer une “google cloud function”

L'Infrastructure As A Service (IAAS)

Cloud Service Provider (CSP)

Déploiement d'un “Hello World”

L'Infrastructure As A Service (IAAS)



Cloud Service Provider (CSP)

Services (on demand*):

Network and Virtual Machines

Managed Clusters (Kubernetes, Spark/Hadoop)

Databases

Serverless Runtimes (Python, Nodejs)

Observability (logs, audit)

Security

Documentation

SLA/SLO/SLI (<https://www.atlassian.com/fr/incident-management/kpis/sla-vs-slo-vs-sli>)

Software Development Kit (SDK)

Librairies (Javascript / Python / Java / Golang / etc...)

APIs REST (interface de programmation web)

*on demand: payer à la seconde, à la minute ... en fonction du service en question.

Service Health

This page provides status information on the services that are part of Google Cloud. Check back here to view the current status of the services listed below. If you are experiencing an issue not listed here, please [contact Support](#). Learn more about what's posted on the dashboard in [this FAQ](#). For additional information on these services, please visit <https://cloud.google.com/>.

✓ Available
i Service information
! Service disruption
✗ Service outage

Incident affecting Google BigQuery, Cloud Run, Cloud Workflows, Operations, Cloud Spanner, Cloud Armor, Google Compute Engine, Google Kubernetes Engine, Cloud Memorystore, Google Cloud Bigtable, Cloud Logging, Persistent Disk, Google Cloud Dataflow, Data Catalog, Google Cloud Storage, Google Cloud Networking, Google Cloud Console, Dataplex, Identity and Access Management, Google Cloud Pub/Sub, Google Cloud SQL, Cloud Filestore, Managed Service for Microsoft Active Directory (AD), Database Migration Service

Multiple Google Cloud services in the europe-west9 region are impacted.

Incident began at **2023-04-25 19:00** (all times are **US/Pacific**).

Currently affected location(s)

Paris (europe-west9)

DATE	TIME	DESCRIPTION
		<p>Summary: Multiple Google Cloud services in the europe-west9 region are impacted.</p> <p>Description: Water intrusion in europe-west9-a has caused a multi-cluster failure and has led to an emergency shutdown of multiple zones. We expect general unavailability of the europe-west9 region. There is no current ETA for recovery of operations in the europe-west9 region at this time, but it is expected to be an extended outage. Customers are advised to failover to other regions if they are impacted.</p>
✗ 26 Apr 2023	00:35 PDT	<p>We will provide an update by Wednesday, 2023-04-26 02:00 US/Pacific with current details.</p> <p>We apologize to all who are affected by the disruption.</p> <p>Diagnosis: Customers may be unable to access Cloud resources in europe-west9 region</p> <p>Workaround: Customers can failover to zones in other regions</p>



Déploiement d'un "Hello World"



<https://console.cloud.google.com/functions/list?project=oxyl-terraform-tn-april-23>

Terraform en entreprise

Travailler à plusieurs - Terraform Backend

Travailler à plusieurs - Contribuer

Terraform chez les clients

Le risque vendor lock-in

“Caveats”

Travailler à plusieurs - Terraform Backend

Protéger une modification en cours avec un “lock”.

```
terraform force-unlock
```

Quid des configurations terraform sur différentes branches ?

```
workspace
```

```
-target resource.name
```

Travailler à plusieurs - Contribuer (1)

Structure des projets Git

Des projets isolés administrés par des admins/architectes
ou

Des parties de configuration terraform couplées au code

IAM - Identity and Access Management

Être capable d'appliquer le principe du moindre privilège

Travailler à plusieurs - Contribuer (2)

Responsabilité des DevOps / exploitation~production

Difficile d'apprendre quand on peut tout casser 🛠️🛠️

Connaissance du cloud

Gérer précisément les droits des contributeurs

bonne connaissance de la gestion des accès et des besoins utilisateurs

ou

Limiter le nombre de contributeurs

intégration continue amoindrie

Terraform chez les clients

Figaro Classifieds

Contexte: Applications web (kubernetes)

Code terraform: isolé et maintenu(validé) par

Equipe d'exploitation dédiée

Contribution possible

Exemple du rachat de Viadéo
par Figaro Classifieds (coût de
maintenance élevé, difficile de
savoir quel élément
retirer/scale down sans tout
casser)




Club Med

Contexte: Data engineering

Code terraform: couplé

Autonomie des prestataires

Le risque vendor lock-in

<div>AWS Lambda Amazon</div> <div>Learn More</div> <div>Update Features</div> <div>Serverless Features<ul style="list-style-type: none">API Proxy ✓Application Integration ✓Data Stores ✓Developer Tooling ✓Orchestration ✓Reporting / Analytics ✓Serverless Computing ✓Storage ✓</div> <div>Integrations<ul style="list-style-type: none">AWS App Mesh ✓AWS CodeDeploy ✓AWS CodeStar ✓</div>	<div>Alibaba Function Compute Alibaba</div> <div>Learn More</div> <div>Update Features</div> <div>Serverless Features<ul style="list-style-type: none">API Proxy ✓Application Integration ✓Data Stores ✓Developer Tooling ✓Orchestration ✓Reporting / Analytics ✓Serverless Computing ✓Storage ✓</div> <div>Integrations<ul style="list-style-type: none">AWS App Mesh ✓AWS CodeDeploy ✓AWS CodeStar ✓</div>	<div>Cloud Functions Google</div> <div>Learn More</div> <div>Update Features</div> <div>Serverless Features<ul style="list-style-type: none">API Proxy ✓Application Integration ✓Data Stores ✓Developer Tooling ✓Orchestration ✓Reporting / Analytics ✓Serverless Computing ✓Storage ✓</div> <div>Integrations<ul style="list-style-type: none">AWS App Mesh ✓AWS CodeDeploy ✓AWS CodeStar ✓</div>
--	---	---

<https://sourceforge.net/software/compare/AWS-Lambda-vs-Alibaba-Function-Compute-vs-Cloud-Functions/>

Caveats

Sensibilité du fichier “state”

Problématique de performance quand le fichier grandit

Incompatibilité qui peuvent survenir sur windows par exemple (eg: réseau)

In place update VS Destroy

Redéploiement de cloud function - Redémarrage de base de données

Destroy and Create pas toujours facile (effets de bords)

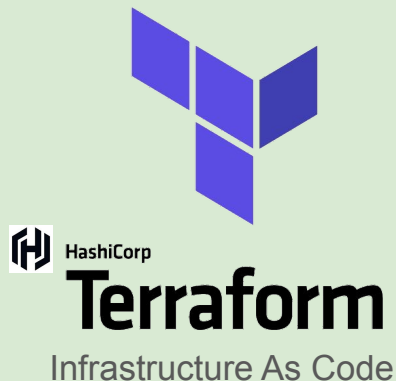
Conclusion

<https://developer.hashicorp.com/terraform/tutorials>

Ça permet de passer du clic au code. Wordpress 🐼

Le principe est super simple.

Les conditions d'apprentissage sont particulières. 📦



Sondage: Qui a du terraform en mission 🖐️ ? allianz, rexel

Question: Quelles alternatives 🛠️ ? pulumi, cloudformation pour AWS

Action: Applaudir et accueillir les managers pour les chiffres 📈.

