

# Automatiser ses déploiements dans Azure avec Terraform

---

Maxime Coquerel - MVP Azure



# Disclaimer

*“Tous les posts de cette présentation ne reflètent que mon opinion et non celle de mes employeurs et clients.”*

# Remerciements

Merci à l'équipe Global Azure BootCamp ainsi qu'à la communauté Azure Québec!



**Communauté Azure Québec**

# # Speaker

Maxime Coquerel

Cloud Solution Architect

Email : [max.coquerel@live.fr](mailto:max.coquerel@live.fr)

Blog : [zigmax.net](http://zigmax.net) (Since 2012)

Github : <https://github.com/zigmax>

Twitter : [@zig\\_max](https://twitter.com/zig_max)

Open Source Contributor (Kubernetes / VSCode).



54 regions worldwide

140 available in 140 countries



\* Two Azure Government Secret region locations undisclosed

# The Azure Periodic Table

Explore the power and possibilities of Azure

 SECURITY CENTER												 AZURE IOT HUB	
 LINUX HUB	 VIRTUAL MACHINES								 AZURE AD B2C	 AZURE AD	 AZURE AD DC	 MULTI-FACTOR	 EVENT HUBS
 SCHEDULER	 SERVICE FABRIC								 MEDIA PLAYER	 CONTENT PROTECTION	 MEDIA ENCODING	 MEDIA STREAMING	 POWERBI
 AUTOMATION	 BATCH	 VPN GATEWAY	 EXPRESSROUTE	 AZURE DNS	 APPLICATION GATEWAY	 AZURE BACKUP	 BIZTALK SERVICES	 CDN	 DATA CATALOG	 DATA FACTORY	 DATA LAKE ANALYTICS	 MACHINE LEARNING	
 OPINSIGHTS	 REMOTEAPP	 RESERVED IP	 VIRTUAL NETWORK	 TRAFFIC MANAGER	 LOAD BALANCER	 SITE RECOVERY	 SERVICE BUS	 MEDIA SERVICES	 HDINSIGHT	 TABLE/BLOB STORAGE	 DATA LAKE STORAGE	 STREAM ANALYTICS	
 KEY VAULT	 CLOUD SERVICES	 PUBLIC IP	 LOGIC APPS	 API APPS	 APP SERVICES	 API MANAGEMENT	 MOBILE APPS	 MOBILE ENGAGEMENT	 WEB APPS	 CUSTOM DOMAIN	 SSL CERTIFICATES	 NOTIFICATION HUBS	
 DEVTEST LABS	 VS APP INSIGHTS	 VS ONLINE	 SQL DATABASE	 SQL DATA WAREHOUSE	 DOCUMENTDB	 CACHE	 SEARCH	 STORAGE	 STORSIMPLE	 IMPORT / EXPORT	 PREMIUM STORAGE	 SQL ELASTIC DB	

# **Infrastructure as a Code**

# Définition - Infrastructure as Code (IAC)

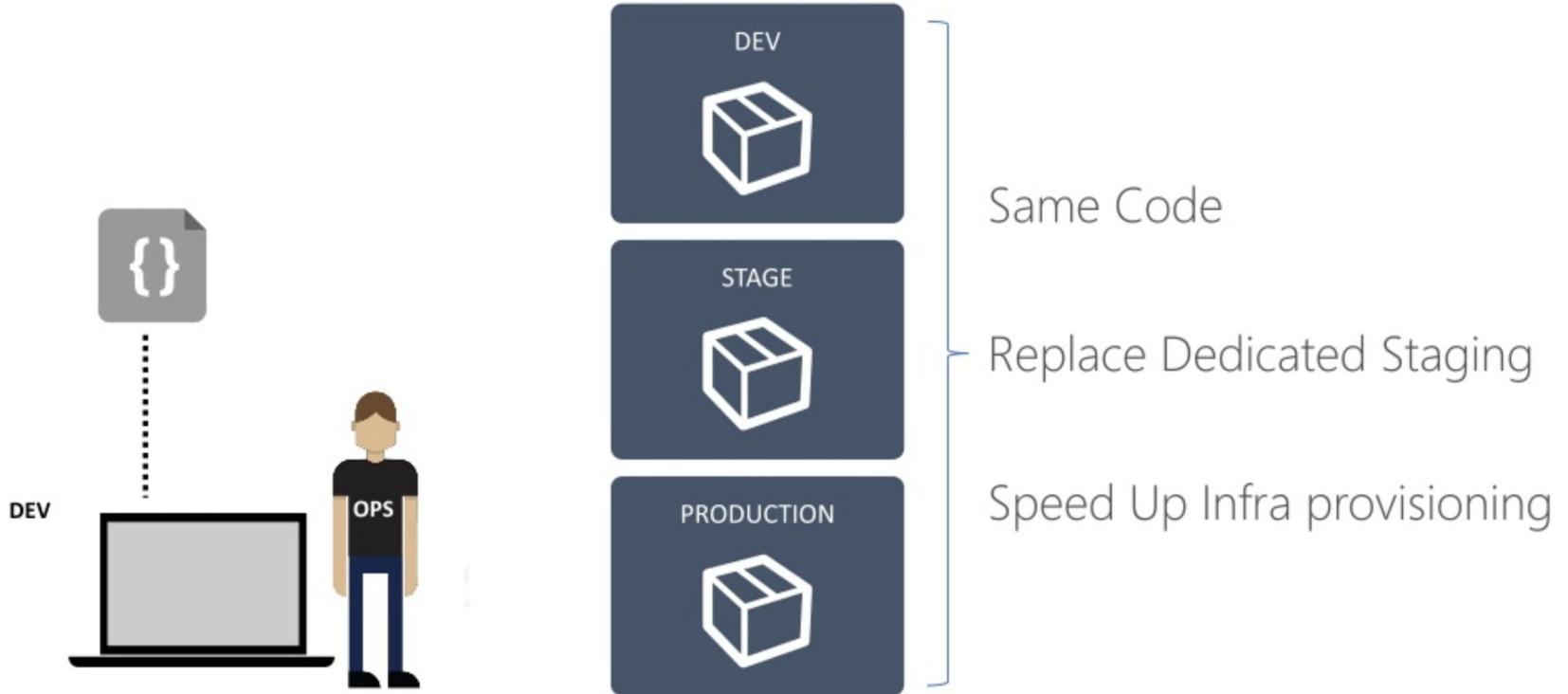
*Infrastructure as Code (IaC) est le concept de **gérer et provisionner des machines / services** au sein d'un centre de données uniquement à l'aide de **fichiers de définition** plutôt qu'une configuration manuelle, à travers des interfaces interactives ou physiquement.*

*Source: Wikipedia*

# Principaux enjeux

- **Impossible de répliquer un environnement existant** (On a une demo la semaine prochaine mais on n'a pas l'environnement ... )
- **Fragile ...** (Tout le monde a peur de faire un changement pendant les vacances de nos Ops)
- **Pas de versionning des changements** (Mais hier cela fonctionnait ... ? Des heures pour retrouver la modification ...)
- **Peu voir pas de documentation des infrastructures** (Mais il a voulu faire quoi ici .... ???)

# Pourquoi IAC ?



# Principaux avantages IAC

- **Consistance**
  - Standardise le déploiement (Toujours le même comportement)
- **Rapidité**
  - Facilite et accélère vos déploiements
- **Réutilisable**
  - Déploiement répétable sans fin ...
- **Extensible**
  - Code modulaire (développement sous formes de modules)

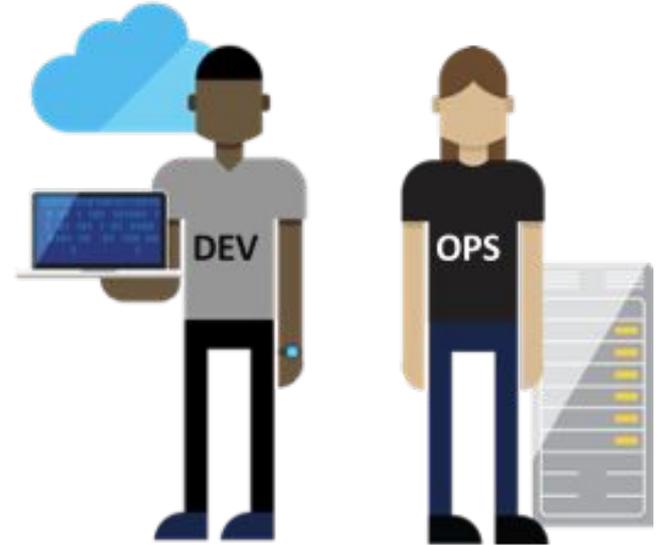
# État de l'art des solutions IAC pour Azure

## Solutions Microsoft :

- ARM Templates
- PowerShell
- Azure CLI

## Solutions tierces :

- Ansible (Module Azure)
- **HashiCorp Terraform**



# HashiCorp



Founded : 2012 - Mitchell Hashimoto Armon Dadgar

## DEVELOP



### Vagrant

Create and configure portable development environments

## PROVISION



### Packer

Create platform specific machine images from a single source



### Terraform

Create, combine and manage infrastructure across multiple providers

## SECURE



### Vault

Centrally store, secure and control access to distributed secrets

## RUN



### Nomad

Cluster manager and scheduler to deploy applications across any infrastructure



### Consul

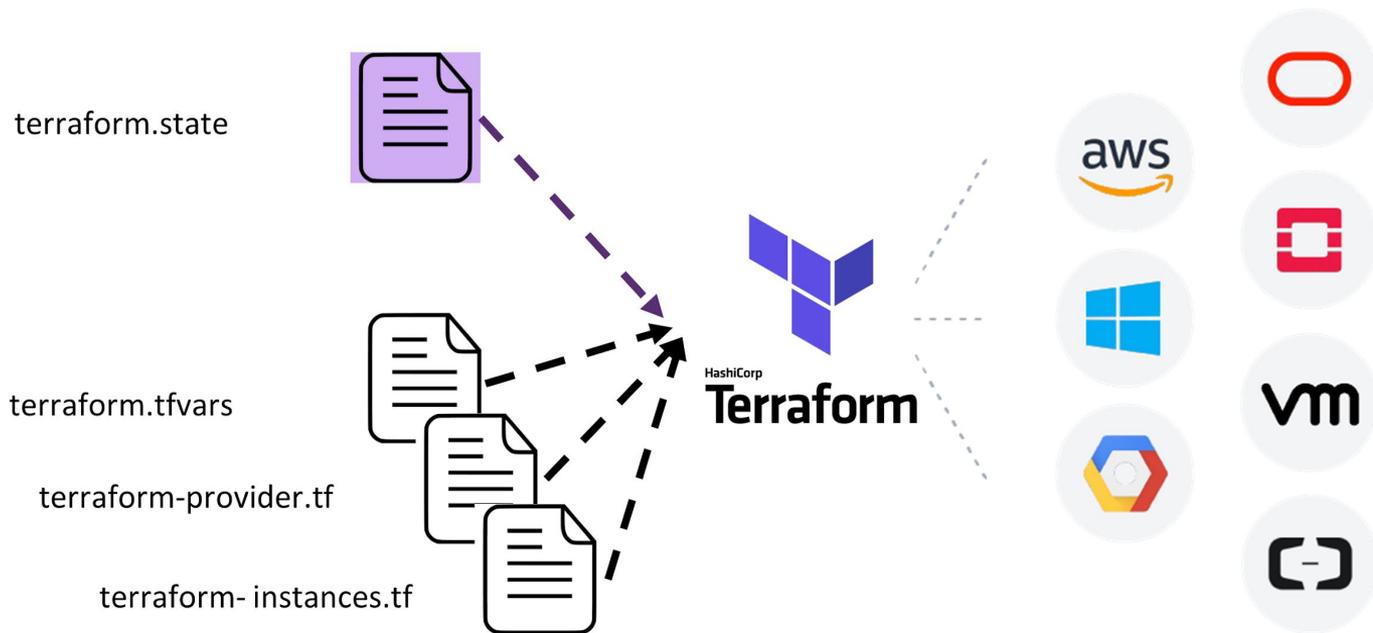
Distributed highly available tool for service discovery, configuration and orchestration

# HashiCorp Terraform

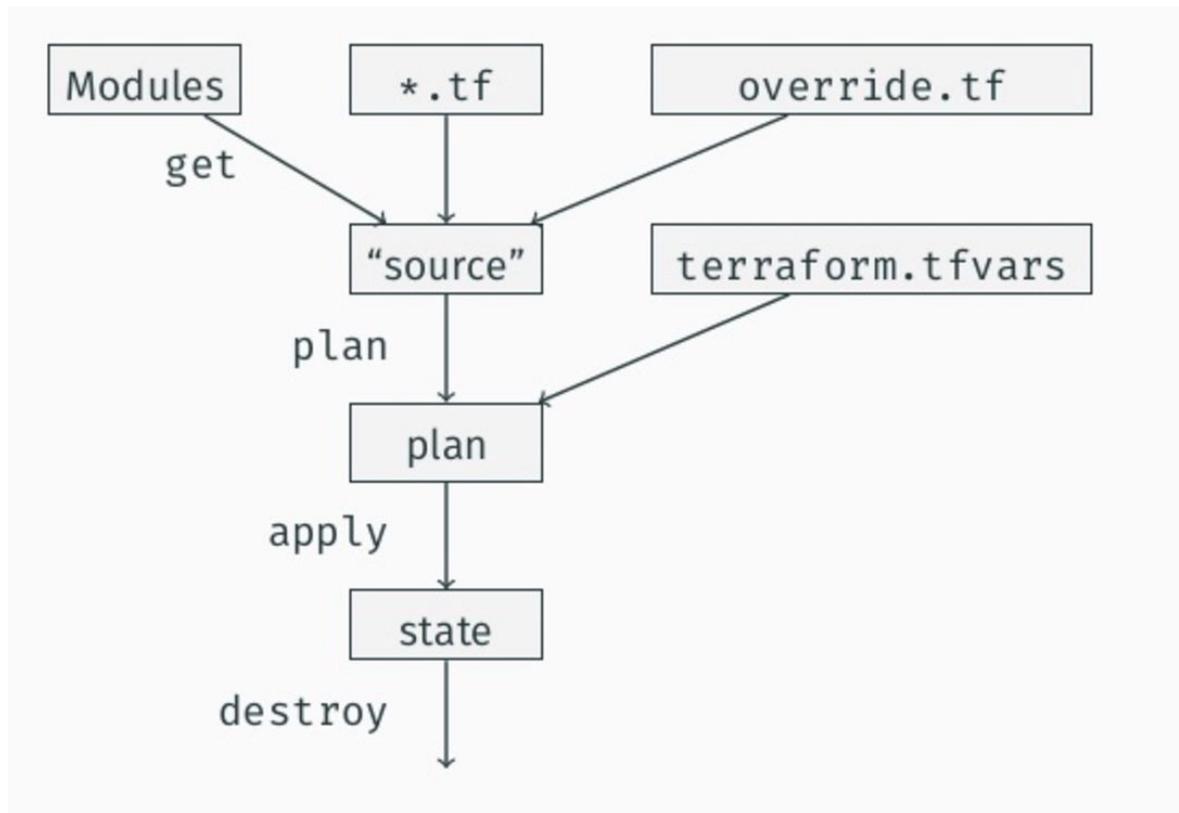
- Open Source
- Mozilla Public Licence 2.0
- Créé par la compagnie HashiCorp (également auteur des produits : vagrant, consul, packer et vault)
- Lancé en 2014
- Développé en Go



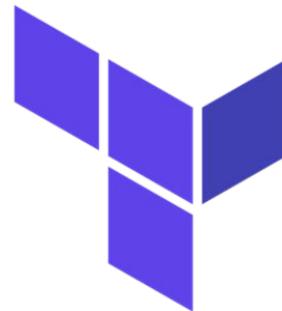
# Terraform Providers



# Terraform - Process



# Flux de travail



## Jour 1



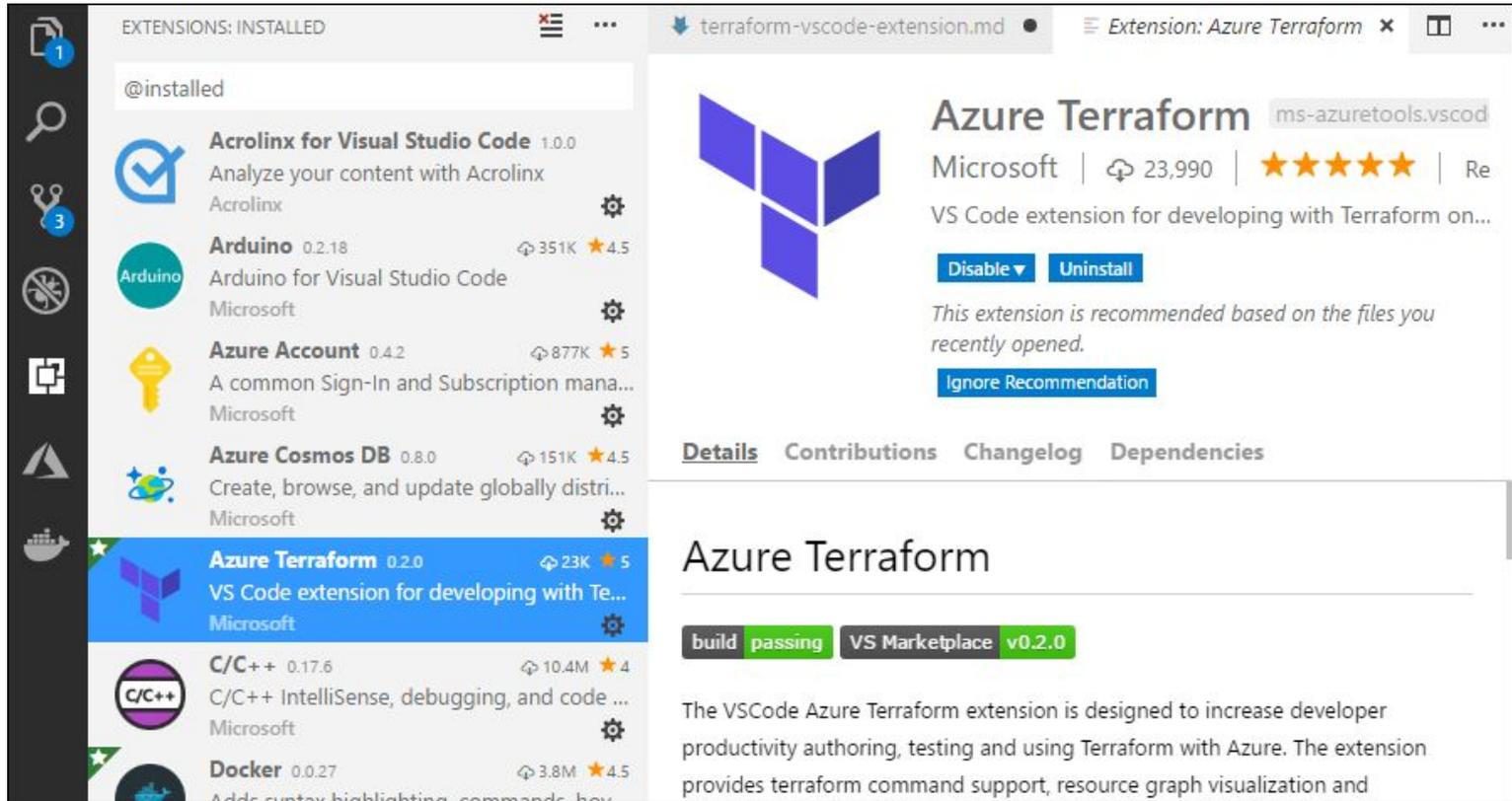
## Jour 1+n



# Exemple de code Terraform

```
1 resource "azurerm_resource_group" "test" {
2   name      = "acctestrg"
3   location = "West US 2"
4 }
5 resource "azurerm_virtual_network" "test" {
6   name                = "acctvn"
7   address_space       = ["10.0.0.0/16"]
8   location             = "${azurerm_resource_group.test.location}"
9   resource_group_name = "${azurerm_resource_group.test.name}"
10 }
11 resource "azurerm_subnet" "test" {
12   name                = "acctsub"
13   resource_group_name = "${azurerm_resource_group.test.name}"
14   virtual_network_name = "${azurerm_virtual_network.test.name}"
15   address_prefix      = "10.0.2.0/24"
16 }
```

# Extension VSCode | Terraform



The screenshot shows the Visual Studio Code interface with the Extensions view open. The left sidebar displays a list of installed extensions, with 'Azure Terraform' highlighted in blue. The main panel shows the details for the 'Azure Terraform' extension by Microsoft. The extension is identified as 'ms-azuretools.vscod' and has 23,990 installations and a 5-star rating. It is recommended based on recently opened files. The interface includes buttons for 'Disable', 'Uninstall', and 'Ignore Recommendation'. Below the extension details, there are tabs for 'Details', 'Contributions', 'Changelog', and 'Dependencies'. The 'Details' tab is active, showing the extension's name 'Azure Terraform' and a status bar indicating 'build passing' and 'VS Marketplace v0.2.0'. The description states that the extension is designed to increase developer productivity by authoring, testing, and using Terraform with Azure, providing terraform command support, resource graph visualization, and other features.

EXTENSIONS: INSTALLED

@installed

- Acrolinx for Visual Studio Code** 1.0.0  
Analyze your content with Acrolinx  
Acrolinx
- Arduino** 0.2.18  
Arduino for Visual Studio Code  
Microsoft
- Azure Account** 0.4.2  
A common Sign-In and Subscription mana...  
Microsoft
- Azure Cosmos DB** 0.8.0  
Create, browse, and update globally distri...  
Microsoft
- Azure Terraform** 0.2.0  
VS Code extension for developing with Te...  
Microsoft
- C/C++** 0.17.6  
C/C++ IntelliSense, debugging, and code ...  
Microsoft
- Docker** 0.0.27  
Adds syntax highlighting, commands, hov...

**Azure Terraform** ms-azuretools.vscod  
Microsoft | 23,990 | ★★★★★ | Re  
VS Code extension for developing with Terraform on...

Disable Uninstall

*This extension is recommended based on the files you recently opened.*

Ignore Recommendation

[Details](#) [Contributions](#) [Changelog](#) [Dependencies](#)

## Azure Terraform

build passing VS Marketplace v0.2.0

The VSCode Azure Terraform extension is designed to increase developer productivity authoring, testing and using Terraform with Azure. The extension provides terraform command support, resource graph visualization and

# Documentation Terraform



[Intro](#) [Learn](#) [Docs](#) [Guides](#) [Extend](#) [Enterprise](#) [Download](#) [GitHub](#)

- › All Providers
- › Azure Providers
  - › Azure Active Directory
  - › Azure
  - › Azure Stack
- › Guides
  - › Azure Provider 2.0 Upgrade Guide
  - › Authenticating using the Azure CLI
  - › Authenticating using Managed Service Identity
  - › Authenticating using a Service Principal with a Client Certificate
  - › Authenticating using a Service Principal with a Client Secret
- › Upcoming Community Events
  - › Community Gardening - Fall 2018
- › Data Sources

## azurerm\_kubernetes\_cluster

Manages a Managed Kubernetes Cluster (also known as AKS / Azure Kubernetes Service)

**Note:** All arguments including the client secret will be stored in the raw state as plain-text. [Read more about sensitive data in state.](#)

## Example Usage

This example provisions a basic Managed Kubernetes Cluster. Other examples of the `azurerm_kubernetes_cluster` resource can be found in the [./examples/kubernetes](#) directory within the [Github Repository](#)

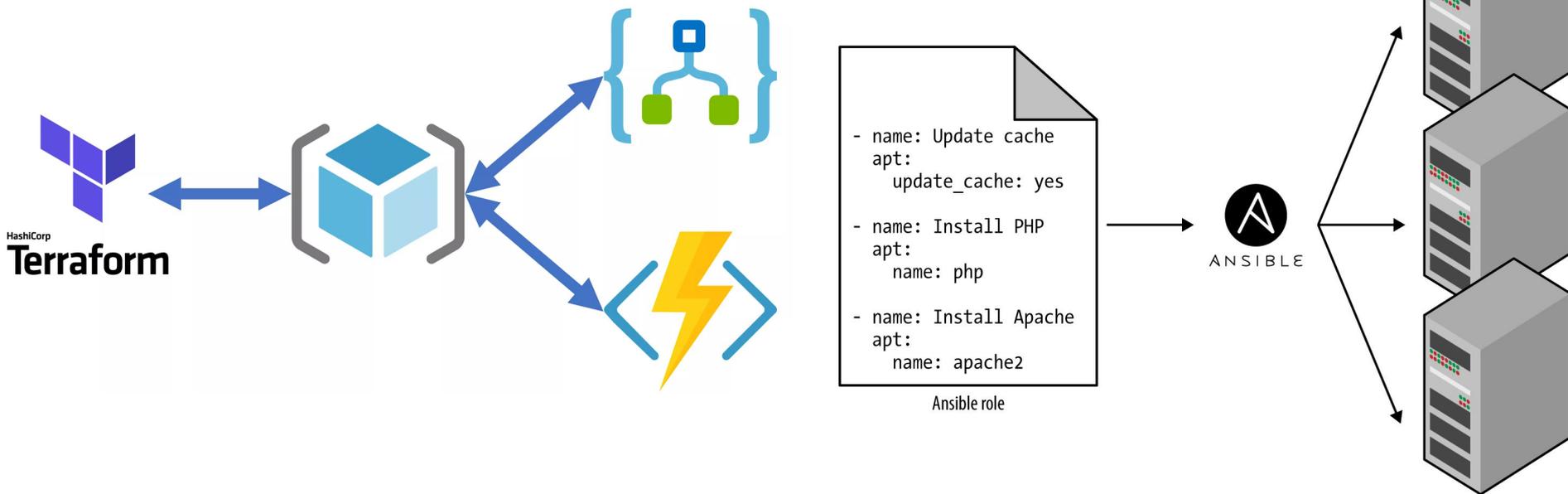
```
resource "azurerm_resource_group" "test" {
  name     = "acctestRG1"
  location = "East US"
}

resource "azurerm_kubernetes_cluster" "test" {
  name                = "acctestaks1"
  location            = "${azurerm_resource_group.test.location}"
  resource_group_name = "${azurerm_resource_group.test.name}"
  dns_prefix          = "acctestagent1"
}
```

# Config Mgmt vs Orchestration

	<b>Chef</b>	<b>Puppet</b>	<b>Ansible</b>	<b>SaltStack</b>	<b>CloudFormation</b>	<b>Terraform</b>
<b>Code</b>	Open source	Open source	Open source	Open source	Closed source	Open source
<b>Cloud</b>	All	All	All	All	AWS only	All
<b>Type</b>	Config Mgmt	Config Mgmt	Config Mgmt	Config Mgmt	Orchestration	Orchestration
<b>Infrastructure</b>	Mutable	Mutable	Mutable	Mutable	Immutable	Immutable
<b>Language</b>	Procedural	Declarative	Procedural	Declarative	Declarative	Declarative
<b>Architecture</b>	Client/Server	Client/Server	Client-Only	Client/Server	Client-Only	Client-Only

# Terraform & Ansible



# Déployer un cluster AKS avec Terraform - Demo



**Retours d'expériences**

-

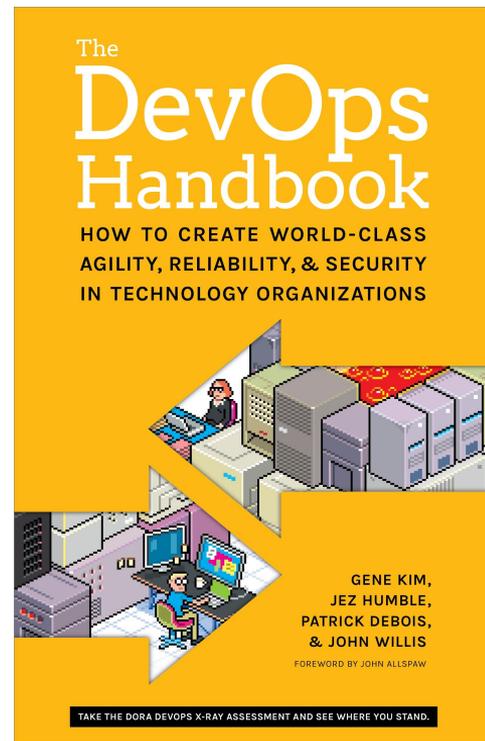
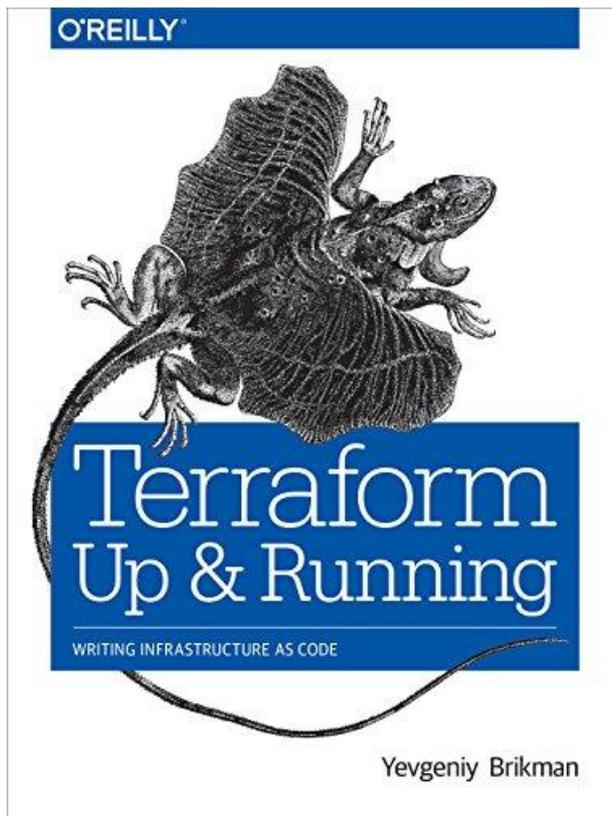
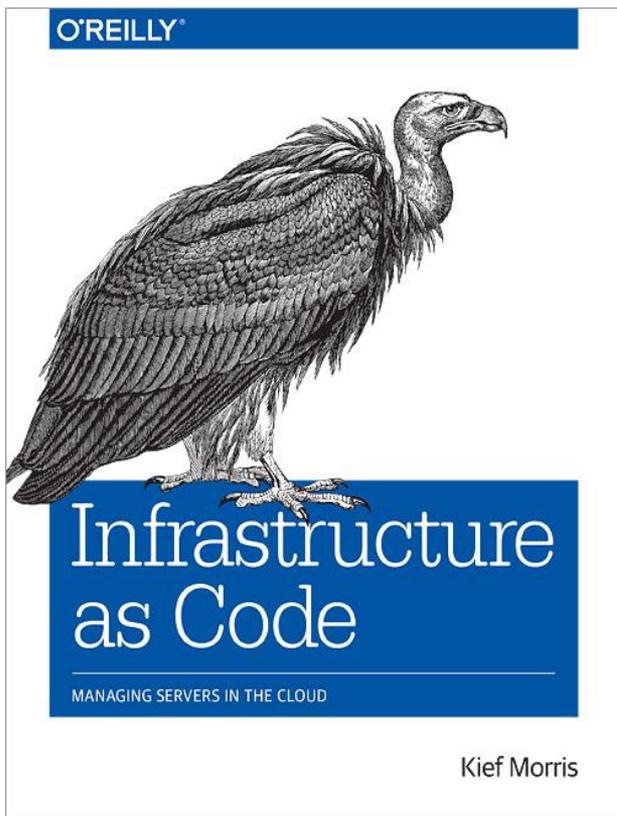
**(Lesson Learned)**

# Conclusion / Recommandations

- > Sélectionner un **projet pilote** “simple” pour commencer
- > Avant de se lancer dans du code, **avoir une architecture et un plan de projet viable ...**
- > **Impliquer vos développeurs** dans l'écriture du code relatif à l'infrastructure
- > Déployer votre code IAC via un outil de **CI / CD (Azure DevOps par exemple)**
- > Have fun :) !



# Livres



# Ressources Techniques

Microsoft Learn - <https://docs.microsoft.com/fr-fr/learn/>

Microsoft Virtual Academy (FR) - <https://stanislas.io/2016/04/26/41/>

Microsoft Technical Community Content

<https://github.com/Microsoft/TechnicalCommunityContent>

Terraform Azure Provider - <https://www.terraform.io/docs/providers/azurerm/index.html>

Maxime Blog - <http://zigmax.net>

Microsoft Ignite 2018 - <https://myignite.techcommunity.microsoft.com/>

Questions / Talks