



**Gérer son infra avec  
son langage de  
programmation  
préféré\***

**Julien Briault** - Deezer



**@ju\_hnny5**

# ~#whoami

## Julien Briault

(ex) SecOps consultant chez  **Rudder**

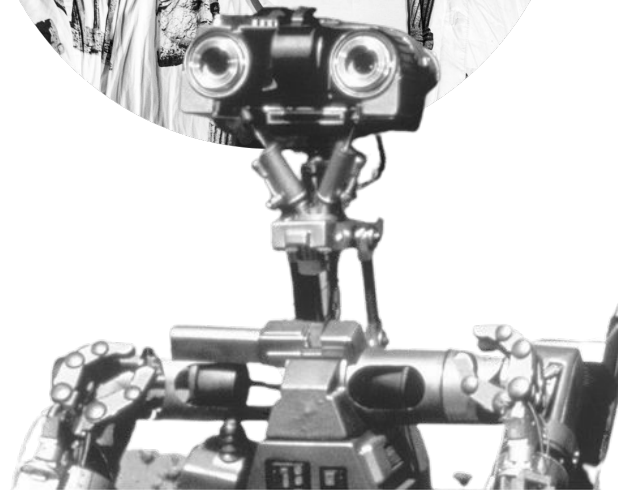
IT/Infrastructure Manager (bénévole) aux  
Network Engineer / SRE chez  **DEEZER**



Auteur principal sur [blog.jbriault.fr](https://blog.jbriault.fr)

#Networking #FOSS #Dev #Music

 [@ju\\_hnny5](https://twitter.com/ju_hnny5)



Et oui, je ne dors pas beaucoup...



@ju\_hnny5

**Julien Briault**

@ju\_hnny5



Actuellement aux @restosducoeur nous recherchons des dons de pc portables, d'écran (avec HDMI), raspberry, etc.

Si jamais vous connaissez des entreprises ouvertes aux dons, n'hésitez pas à me contacter. 🙏

**WE NEED YOUR HELP!**

## Disclaimer

**Ce talk a été produit avant le drama  
d'Hashicorp et la Business Source  
Licence.**



# Pulumi



## Open Source

### Plus 7 fun facts for Terraform users



0:00 / 2:30 • Introduction: Pulumi ♥ Open Source >



Pulumi ♥ Open Source - Plus 7 Fun Facts for Terraform Users

[https://www.youtube.com/watch?v=E\\_F71Niwg\\_8](https://www.youtube.com/watch?v=E_F71Niwg_8)



@ju\_hnny5



A black and white photograph of a crowd of people at a formal event. In the foreground, a person wearing a hoodie and sunglasses is looking down. The background shows many other people, some in formal attire like tuxedos and gowns, looking in various directions.

Disclaimer 2

**Aucun code  
Terraform ne sera maltraité.**

**Retour aux  
sources**



An aerial, high-angle photograph of a city intersection. Several cars are visible, including a dark sedan in the center, a white car to the right, and a dark car further back. Pedestrians are crossing the street at a crosswalk. The image is in black and white, with some color overlays for text.

**Dev - Ops**

@ju\_hnny5

An aerial, high-angle photograph of a busy city intersection. Several cars are visible, including a dark sedan in the center, a white car to the right, and a dark car further back. Pedestrians are crossing the street at a crosswalk. The scene is captured in a high-contrast, almost black and white style with some color saturation. A purple rectangular box is overlaid on the lower-left portion of the image, containing the text 'Agile + DevOps' in white.

# Agile + DevOps



# Pet vs Cattle

@ju\_hnny5

A black and white photograph of a man with a beard, shirtless, standing in a dense forest. He is holding a large, light-colored pipe or tube horizontally across his chest. The background is filled with trees and foliage, creating a textured, dappled light effect. A yellow rectangular box with black text is overlaid on the lower left portion of the image.

**"Infra is hard.."**

**@ju\_hnny5**

# Infra as Code ?

# L'infra as **Code** ?

L'*Infrastructure as Code* (IaC) est **une pratique** qui **consiste à gérer et à provisionner des infrastructures informatiques en utilisant des fichiers de définition.**

# L'infra as **Code** ?

L'*Infrastructure as Code* apporte plusieurs bénéfices :

1. Elle permet de gérer le versionning de l'infrastructure à l'aide de logiciels comme `Git`.
2. Le **Shadow IT** est réduit, c'est-à-dire l'ensemble des systèmes informatiques qui ne sont pas officiellement répertoriés.



# L'infra as Code ?

Infrastructure  
Provisioning

$\neq$

Infrastructure  
(configuration)  
Management



@ju\_hnny5

# L'infra as Code ?



Stateful

vs

Stateless

# L'infra as **Code** ?

**Imperative**

Procedural  
Structured  
Object-oriented

**VS**

**Declarative**

Functional  
Logic

# L'infra as **Code** ?

## Imperative

Step by step  
instructions

Create a server

Add a server

Make this change

**VS**

## Declarative

Declare  
and result

2 servers



**"When devs  
want to do Ops".**



@ju\_hnny5

Pourquoi **OctoDNS** ?  
Parce qu'il fait les choses  
simplement et de manière  
agnostique. 😄



<https://www.youtube.com/watch?v=aZvsplOAd7c>

**On commence ?**





**Legacy solutions**

@ju\_hnny5

# Legacy Solutions : AWS CloudFormation

## Quelques **désavantages** :

1. Complexité de la syntaxe
2. N'est lié qu'à AWS (obviously)
3. Limitation des modèles
4. Ressources orphelines
  - a. Si une mise à jour de template n'inclut pas toutes les ressources existantes, certaines peuvent devenir "orphelines", c'est-à-dire qu'elles ne sont pas supprimées lors de la mise à jour, ce qui peut entraîner des coûts inattendus.



**AWS CloudFormation**

# Legacy Solutions : Azure Bicep

## Quelques **désavantages** :

1. La courbe d'apprentissage
2. N'est lié qu'à Microsoft Azure
3. La maturité
4. Ecosystème et la communauté



Microsoft Azure

# Legacy Solutions : OpenStack Heat

## Quelques désavantages :

1. La complexité au démarrage (lié principalement au DSL)
2. L'héritage de valeurs
3. Le système de modules



# HEAT

*an OpenStack Community Project*

# Legacy Solutions : Terraform

## Quelques **désavantages** :

1. La complexité au démarrage (lié principalement au DSL)
2. L'héritage de valeurs
  - a. Le système de modules
3. Lisible mais logique de dev difficilement applicable



HashiCorp

# Terraform

# Le combat que tout oppose



**Le standard  
de facto**

@ju\_hnny5







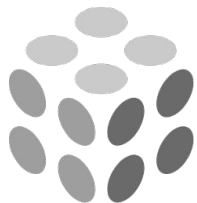
# Pulumi



Le petit  
nouveau\*

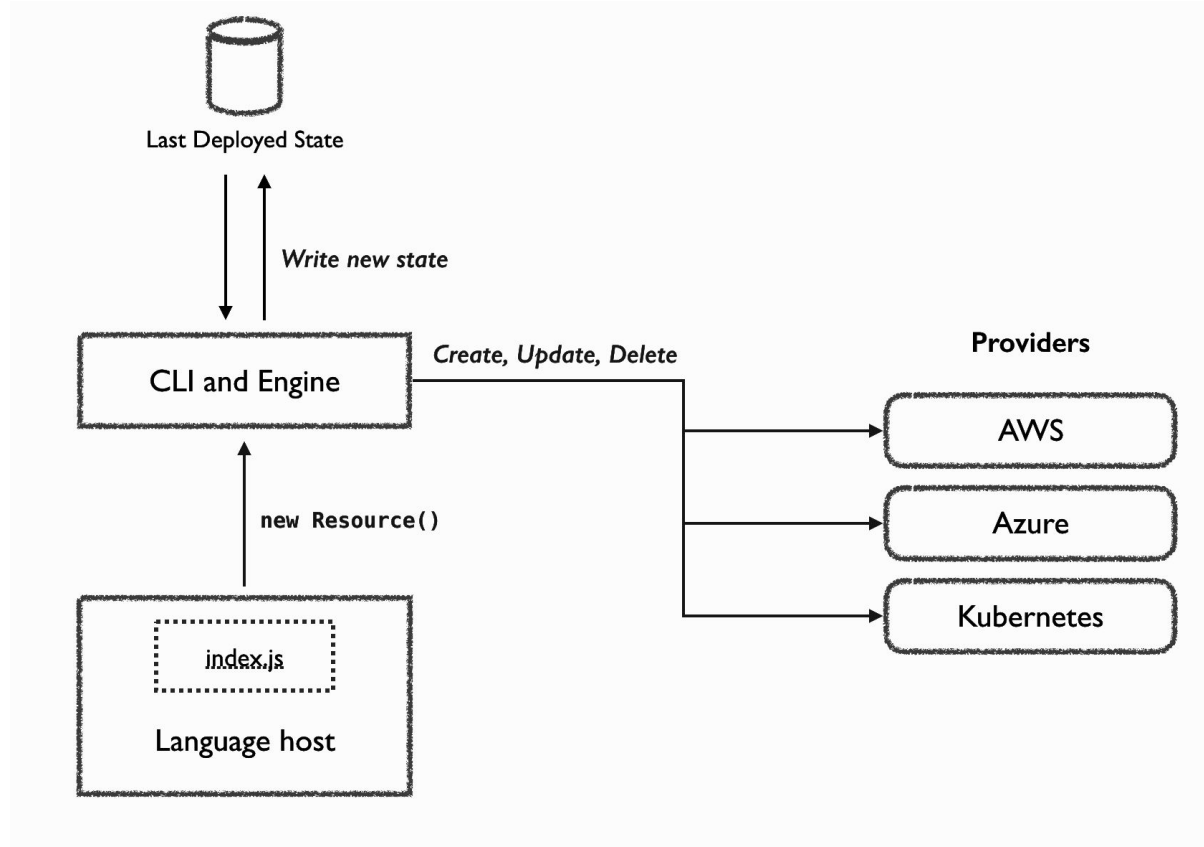
# Pulumi, Késako ?

- La première version stable est sortie le **3 septembre 2019**.
  - Projet très jeune par rapport à Terraform sorti en 2014
- Projet écrit en **Go**(lang)
- Possède une communauté large et active
- Beaucoup de tooling fournit directement par l'outil
  - Vs Terraform où beaucoup d'améliorations sont apportées par des outils tiers (exemple Terragrunt)
- Pour bien commencer :
  - <https://www.pulumi.com/docs/get-started/>

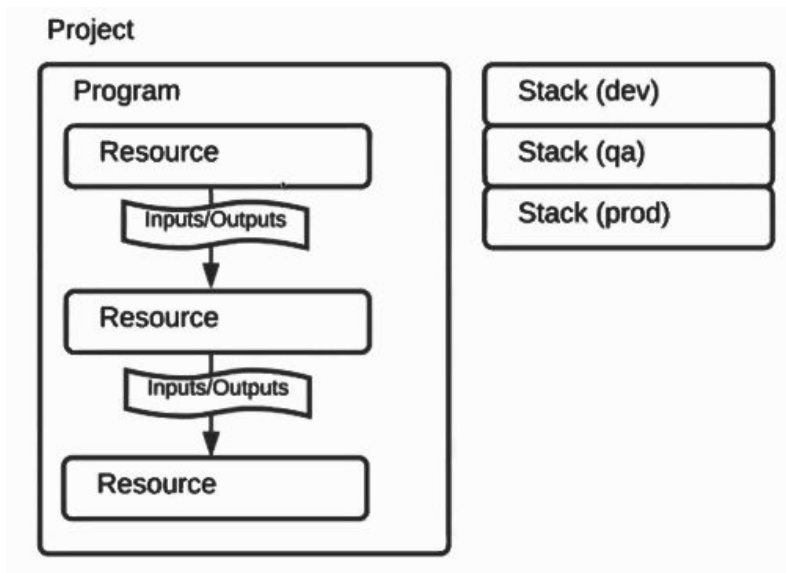


# Pulumi

# Pulumi, Késako ?



# Pulumi, Késako ?



**Infra as Code : Apprendre  
de nouveau langages ?**



# DevXP

```
header(); ?>
```

```
div class="container">
```

```
<h1 class="text-primary">
info"></i> <
<hr>
```

```
<div class="
```

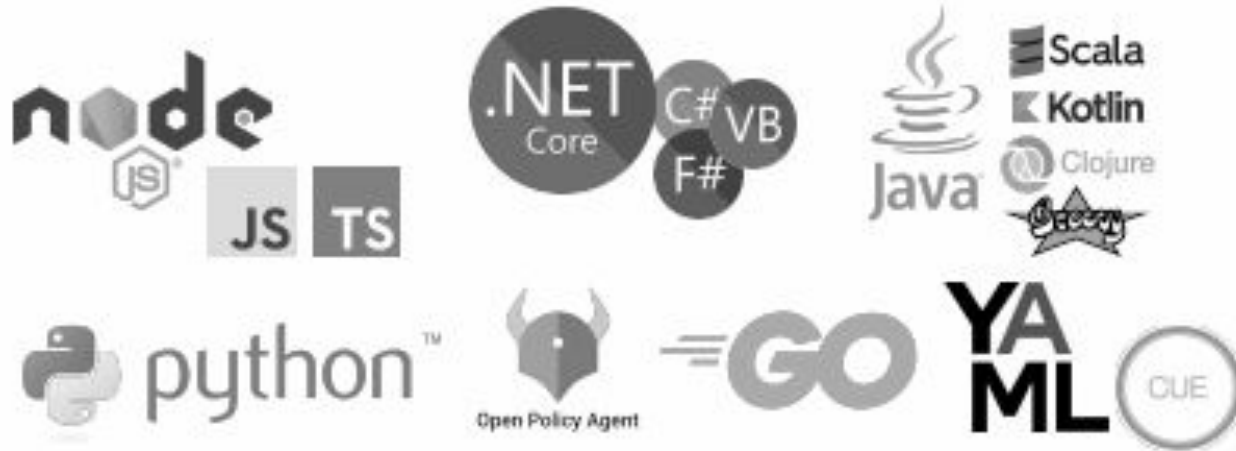
```
<?php
```



@ju\_hnny5

# DevXP : Les langages supportés

A l'inverse de Terraform qui propose d'utiliser son DSL, **Pulumi** propose un SDK permettant d'utiliser plusieurs langages comme :





# DevXP : Le code

```
resource "openstack_compute_instance_v2" "basic" {  
  name = "basic"  
  image_name = "Ubuntu Server 22.04"  
  flavor_name = "m1.small"  
  key_pair = "jlbriault"  
  security_groups = ["default"]  
  network { name = "tenant_network" }  
}
```

Terraform  
HCL

```
instance = openstack.compute.Instance("basic",  
  flavor_name="m1.small",  
  image_name="Ubuntu Server 22.04",  
  key_pair="jlbriault",  
  security_groups=["default"],  
  networks=[{"name": "tenant_network"}],  
)
```

Pulumi  
Python

L'Infrastructure  
as Code, bah...  
c'est du **code** !



# DevXP : Les langages supportés

Des ressources intéressantes :

- <https://www.pulumi.com/docs/concepts/vs/terraform/>
- <https://www.youtube.com/watch?v=PqAP4BunQZU>



Pulumi pour les utilisateurs de Terraform

# DevXP : **L'outillage**

1. Utiliser son environnement de dev préféré
  - a. Pas besoin d'IDE ou de plugin spécifique pour Pulumi
2. Mode pour de la CICD (via l'option `-non-interactive`)  
**directement intégré** à la CLI Pulumi



# DevXP : L'outillage

## IDEs



## Tools



## Packages



# DevXP : L'outillage

L'IA au service du bien.

🗨️ Generate an EC2 instance on AWS.

Python

The following Python code creates a single EC2 instance on AWS using Pulumi. It specifies the Amazon Machine Image (AMI) ID, instance type, and associates a public IP address with the instance for SSH access. For security reasons, it's best to use a specific AMI after checking the latest available in your region, but here's an example using the Amazon Linux 2 AMI.

```
import pulumi
import pulumi_aws as aws

# AMI for Amazon Linux 2 in the us-west-2 region. This should be changed to the one relevant to your needs.
ami_id = "ami-0e34e7b9ca0ace12d"

# Generate a key pair to SSH into the instance. Do this only if you don't already have an RSA key pair.
key_pair = aws.ec2.KeyPair("my-keypair", public_key="ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCs... user@example.com")

# Create a new security group that allows SSH traffic on port 22 from anywhere.
security_group = aws.ec2.SecurityGroup('allow-ssh',
    description='Allow SSH inbound traffic',
    ingress=[
        {
            'description': 'SSH from anywhere',
            'from_port': 22,
            'to_port': 22,
            'protocol': 'tcp',
            'cidr_blocks': ['0.0.0.0/0'],
        }
    ],
)
```

@ju\_hnny5

# DevXP : La modularité

1. Vous pouvez organiser votre code en modules réutilisables, ce qui facilite la création et la gestion de configurations d'infrastructure complexes.
2. Vous pouvez facilement découper votre code sous forme de modules ou non (à l'inverse de Terraform qui n'offre qu'un système de modules).
3. Pas besoin d'outil comme `terragrunt` pour hériter de valeurs.

# DevXP : Une couche d'abstraction

Pulumi permet d'utiliser les mécanismes de chaque langage.

Ainsi il est aisé d'apporter de l'abstraction sur certains éléments de son infrastructure.



# DevXP : Une couche d'abstraction

Exemple, créer des objets sur un pare-feu FortiGate :

```
for key, value in content_services.items():
    if value['type'] == 'TCP':
        serviced = forti.FirewallServiceCustom(
            key,
            app_service_type="disable",
            category="General",
            check_reset_range="default",
            color=0,
            helper="auto",
            iprange="0.0.0.0",
            name=key,
            protocol="TCP/UDP/SCTP",
            protocol_number=6,
            proxy="disable",
            tcp_halfclose_timer=0,
            tcp_halfopen_timer=0,
            tcp_portrange=value['port_range'],
            tcp_timewait_timer=0,
            udp_idle_timer=0,
            visibility=value['visibility'],
        )
    elif value['type'] == 'UDP':
        serviced = forti.FirewallServiceCustom(
            key,
            app_service_type="disable",
            category="General",
            check_reset_range="default",
            color=0,
            helper="auto",
            iprange="0.0.0.0",
            name=key,
            protocol="TCP/UDP/SCTP",
            protocol_number=6,
            proxy="disable",
            udp_halfclose_timer=0,
            udp_halfopen_timer=0,
            udp_portrange=value['port_range'],
            udp_timewait_timer=0,
            udp_idle_timer=0,
            visibility=value['visibility'],
        )
    elif value['type'] == 'SCTP':
        serviced = forti.FirewallServiceCustom(
            key,
            app_service_type="disable",
            category="General",
            check_reset_range="default",
            color=0,
            helper="auto",
            iprange="0.0.0.0",
            name=key,
            protocol="TCP/UDP/SCTP",
            protocol_number=6,
            proxy="disable",
            sctp_halfclose_timer=0,
            sctp_halfopen_timer=0,
            sctp_portrange=value['port_range'],
            sctp_timewait_timer=0,
```



```
# TCP port range
myport:
    type: "TCP" # or UDP/SCTP
    port_range: "223-332"
    visibility: "enable"
    category: "General"
```

```
ssh_port:
    type: "TCP"
    port_range: "22-22"
    visibility: "enable"
    category: "General"
```

<https://github.com/juhnny5/pulumi-fortigate-example>

@ju\_hnny5

A black and white photograph of two laptops. The laptop on the left is in the foreground, and the one on the right is slightly behind it. Both screens display lines of code. A hand is visible on the left, near the first laptop, and another hand is on the right, typing on the second laptop's keyboard.

# The CLI

@ju\_hnny5

# The CLI : pulumi

Une seule ligne de commande pour tout contrôler.

Verbes différents de ceux de Terraform.

- Liste disponible ici :

<https://www.pulumi.com/docs/concepts/vs/terraform/terminology/#commands>

# The CLI : pulumi

## Exemples

Afficher les modifications à apporter :

```
pulumi preview
```

```
terraform plan
```

Appliquer les modifications :

```
pulumi update
```

```
terraform apply
```

Détruire les ressources :

```
pulumi destroy
```

```
terraform destroy
```




# The CLI : Une migration facilitée

Une commande :

```
pulumi convert \  
-from terraform \  
-language python \  
-out pulumi
```

Disponibles depuis la  
version **v3.71.0**.

A stylized illustration of a computer monitor. The screen is black and displays a terminal window with a green prompt and a command. The command is 'pulumi convert --from terraform -language typescript --out pulumi'. The monitor has a grey bezel and a stand.

```
[~/demo/vault-infra/terraform/main (master)]$ pulumi convert --from terraform -  
-language typescript --out pulumi
```

# The CLI : Une migration facilitée

Migrer ses ressources :

```
pulumi import --from terraform ./terraform.tfstate
```

Quelques ressources pour aider à la migration :

- **Migration Hub**
  - <https://www.pulumi.com/blog/migration-hub/>
  - <https://www.pulumi.com/migrate/>

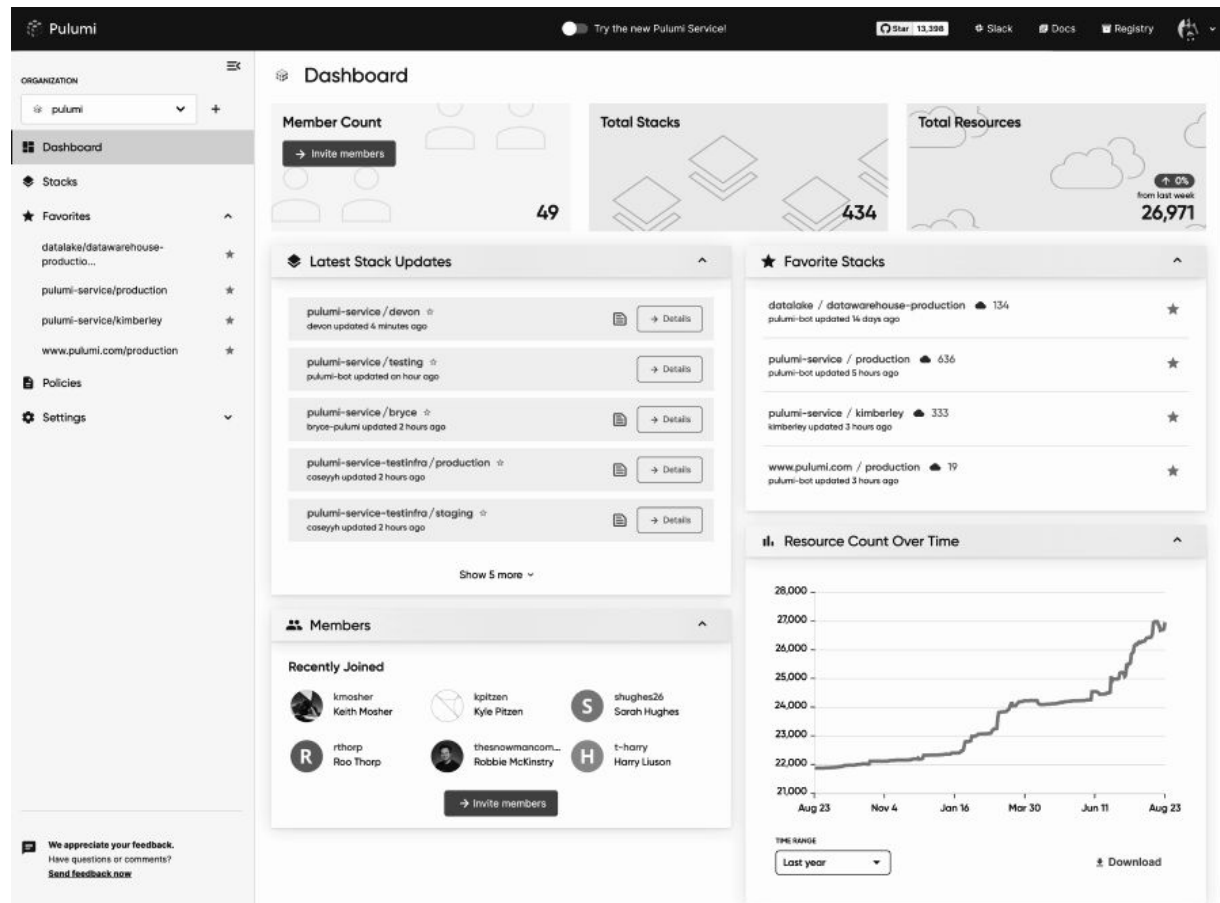
# Pulumi Cloud Console

- Gestion de l'état de déploiement et des secrets dans **Pulumi Cloud** par défaut.
- Permet l'exécution à distance des déploiements.
- S'intègre avec votre CI/CD.
- Gestion des politiques/contrôles d'accès.

**i** Self-hosting is only available with **Pulumi Business Critical**. If you would like to evaluate the Self-Hosted Pulumi Cloud, sign up for the 30 day trial or contact us.

To manage your state with a self-managed backend, such as a cloud storage bucket, see State and Backends.


# Pulumi Cloud Console



@ju\_hnny5



# Pulumi Cloud Console

 Pulumi


☐ Try the new Pulumi Service!

Star 13,398

Slack

Docs

Registry



Stacks

+ Create project





Group By  
Project

Sort By  
Repository name

Search



actions-pulumify

**pulumify**  
Pulumify your docs!



  pulumify-pulumi-docs-cnunciato-pulumify ☆	updated 2 years ago	35	×
  pulumify-pulumi-docs-cnunciato-pulumify-v2 ☆	updated 2 years ago	38	×

autobots



**community-issue-bot**  
A bot to manage issues opened by the community

  community-issue-bot-production ☆	updated 4 months ago	23	✓
--	----------------------	----	---

**hubspotron**  
A bot that fills gaps in HubSpot's CRM functionality.

  hubspotron-production ☆	updated 4 months ago	20	✓
---	----------------------	----	---

**pagerduty-assistant**  
A Slack bot for interacting with Pulumi's pagerduty account.

  production ☆	updated 4 months ago	120	✓
--	----------------------	-----	---

@ju\_hnny5

# Pulumi ESC

(Environment, Secrets and  
Configuration)



@ju\_hnny5

# Pulumi ESC

- Permet aux équipes d'agrèger des secrets et des configurations provenant de nombreuses sources, de gérer des collections hiérarchiques de configurations et de secrets ("environnements"), et de consommer ces configurations et secrets à partir d'une variété de services d'infrastructure et d'application différents.
- Peut fonctionner avec mais aussi sans Pulumi.

A black and white photograph of a sloth climbing a tree trunk. The sloth is positioned on the right side of the frame, with its body angled upwards and its long tail visible. The background is a dense forest with many trees and foliage. The text "Terraform Bridge" is overlaid on the left side of the image in a white font on a purple rectangular background.

# Terraform Bridge

@ju\_hnny5

# Pulumi Terraform Bridge

Adapter n'importe quel provider Terraform (construit à l'aide du Terraform Plugin SDK) en provider provider **Pulumi**.

Les providers Terraform font des opérations **CRUD\***, il est donc “facile” de les transposer en provider Pulumi.

Exemple :

<https://github.com/juhnny5/pulumi-maas-python>

\*Create, Read, Update, Delete

# Pulumi Terraform Bridge

<https://github.com/pulumi/pulumi-tf-provider-boilerplate>

 **README**  Code of conduct  Apache-2.0 license  Security  

## Terraform Bridge Provider Boilerplate

This repository contains boilerplate code for building a new Pulumi provider which wraps an existing Terraform provider.

### Background

This repository is part of the [guide for authoring and publishing a Pulumi Package](#).

Learn about the concepts behind [Pulumi Packages](#).

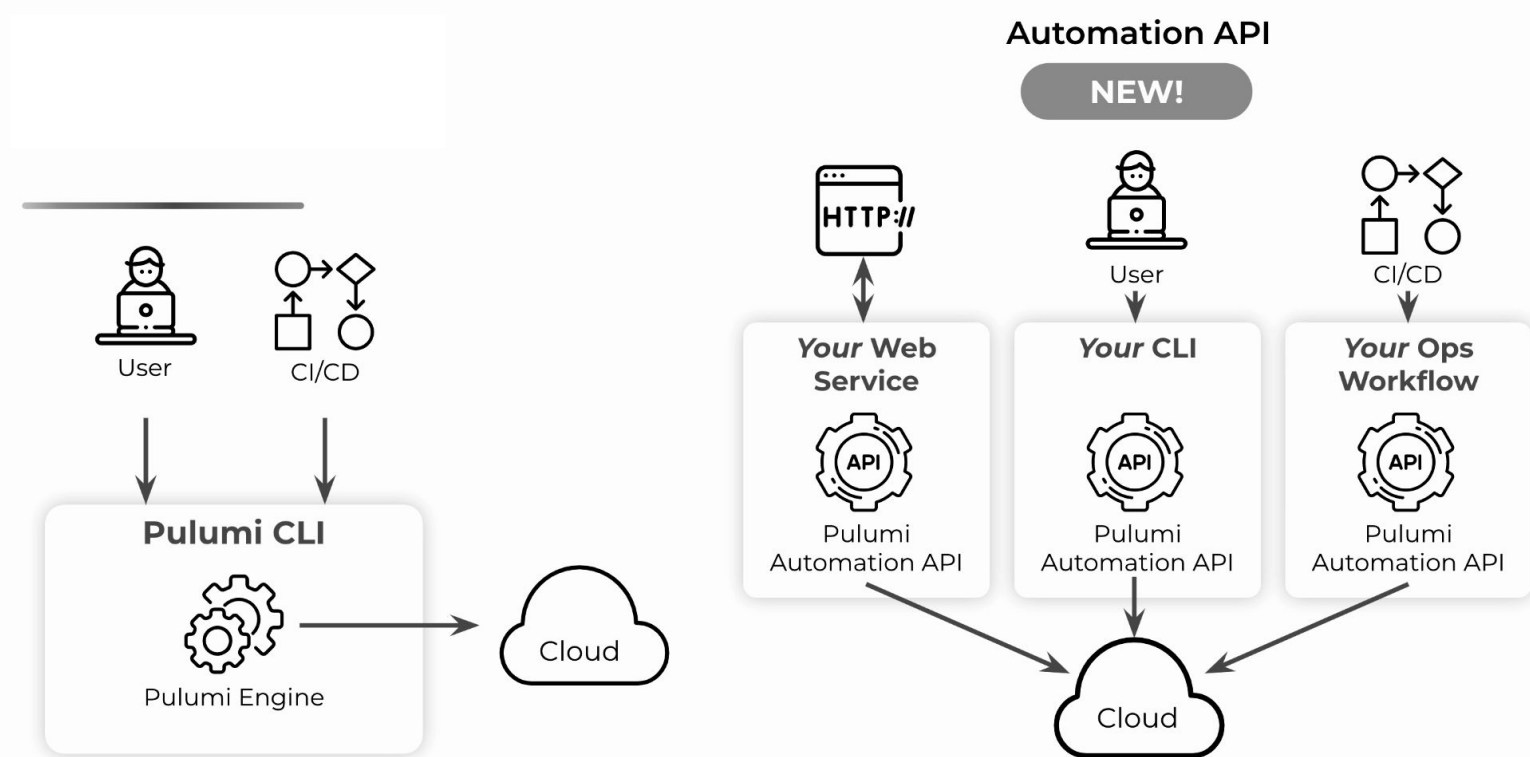
@ju\_hnny5

A black and white photograph of an industrial robotic cell. The cell has a glass front door, and inside, a robotic arm is visible. In the foreground, there are several cylindrical metal parts arranged on a surface. On the left side of the cell, there is a warning symbol (a triangle with an exclamation mark) and a small circular logo. A purple rectangular box with white text is overlaid on the left side of the image.

# Automation API

@ju\_hnny5

# Pulumi Automation API : Intégrations et workflows automatisés





A black and white photograph of two men from the TV show Community. The man on the left is wearing a plaid shirt and looking slightly to the right. The man on the right is wearing a dark shirt and a striped tie, looking towards the left. A woman with long dark hair is partially visible in the background on the left.

Community

@ju\_hnny5

# Community

- <https://slack.pulumi.com/>
- <https://github.com/pulumi>
- <https://twitter.com/PulumiCorp>
- Calendrier des conférences /  
Meetup :  
[https://docs.google.com/spreadsheets/d/1TXphpxOOJN87ptH39JdSv32nzoTpBmhE2HhOGUA\\_E4yw/edit](https://docs.google.com/spreadsheets/d/1TXphpxOOJN87ptH39JdSv32nzoTpBmhE2HhOGUA_E4yw/edit)




A black and white photograph of a man with a beard and mustache, wearing a dark hoodie with white drawstrings. He is sitting and looking down at a laptop, with his hands on the keyboard. The background is out of focus, showing a bright light source that creates a large, soft glow and some lens flare effects. A yellow rectangular box is overlaid on the left side of the image, containing the text "Demo time !".

**Demo time !**

@ju\_hnny5

# Démo 1 : Créer une première instance sur **OpenStack** (avec Python)

A dark, low-key photograph of a man with a beard and a hoodie, smiling while looking at a laptop screen. The image is dimly lit, with the man's face and the laptop screen being the primary light sources.

# Démo 2 : Créer une première instance sur **OpenStack** (avec Python) et comparer avec **Terraform**

# Démo 3 : Créer des ressources en apportant **une couche d'abstraction** (avec Python)



@ju\_hnny5

A dark, low-key photograph of a man with a beard and mustache, wearing a grey hoodie, sitting at a desk and working on a laptop. The image is dimly lit, with the man's face and the laptop screen being the primary light sources. The background is dark and out of focus.

## Démo 4 : Créer des **ressources** (avec YAML)



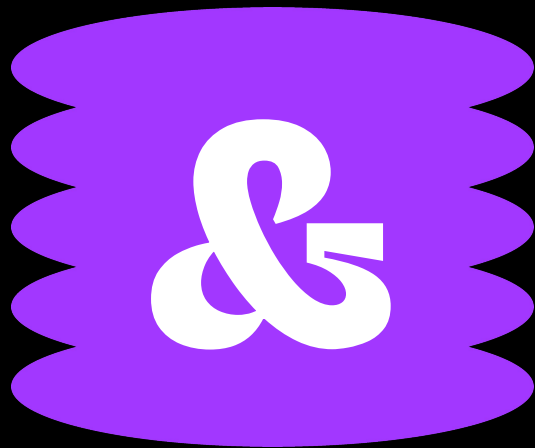
Voila`

@ju\_hnny5



One link : **<https://t.ly/wxK0Q>**

Q



A

# Thank you



DEEZER

@ju\_hnny5