

次世代構成管理ツール Pulumi 試してみた 👍

whoami 🤔

- Name: 森 早人
 - すべて小1で習得する漢字です
- Birth: 1993/10/24
 - 国際連合デー
- Team: 基盤開発T
 - インフラエンジニア
- Resent Events:
 - キーボード買いました、スコスコ感がだいぶいい感じ
 - [FILCO Majestouch Stingray](#)
 - シーシャにハマっています
 - 週1で、近所の感染対策しっかりしてるお店に

Pulumi とは 🛠️

好きな言語でIaCできる！

- CloudFormationやTerraformのような、IaCツール。
- 複数言語（JavaScript、Python、TypeScript、Go、C#）で、AWS、Azure、GCP、K8s上にリソースを構築できる。
-
- [公式ページ](#)



インストールとプロジェクトの作成 🎉

macOSにて

```
## CLIをインストール
$ brew install pulumi

## プロジェクトを作ると、ログインが求められる。
$ pulumi new aws-python
Manage your Pulumi stacks by logging in.
Run `pulumi login --help` for alternative login options.
Enter your access token from https://app.pulumi.com/account/tokens
    or hit <ENTER> to log in using your browser
We've launched your web browser to complete the login process.

Waiting for login to complete...
```

プロジェクトの情報を入力する

新しく作るプロジェクトの入力が求められる

```
project name: (quickstart)
project description: (A minimal AWS Python Pulumi program)
Created project 'quickstart'
```

```
Please enter your desired stack name.
To create a stack in an organization, use the format <org-name>/<stack-name> (e.g. `acmecorp/dev`).
stack name: (dev)
Created stack 'dev'
```

```
aws:region: The AWS region to deploy into: (us-east-1) ap-northeast-1
Saved config
```

作られるプロジェクトを確認してみる 👁👁

S3を作るファイル

```
#__main.py__

"""An AWS Python Pulumi program"""

import pulumi
from pulumi_aws import s3

# Create an AWS resource (S3 Bucket)
bucket = s3.Bucket('my-bucket')

# Export the name of the bucket
pulumi.export('bucket_name', bucket.id)
```

Cloudformationで書くと

前スライドと同じことを実行する

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  S3Buckt:
    Type: "AWS::S3::Bucket"
    Properties:
      BucketName: "my-bucket"
```

変更を適用する

```
$ pulumi up
Previewing update (dev):
   Type                                Name                Plan
+   pulumi:pulumi:Stack                quickstart-dev      create
+   └─ aws:s3:Bucket                  my-bucket           create

Resources:
+ 2 to create

Do you want to perform this update? [Use arrows to move, enter to select, type to filter]
> yes
no
details

Updating (dev):
   Type                                Name                Status
+   pulumi:pulumi:Stack                quickstart-dev      created
+   └─ aws:s3:Bucket                  my-bucket-edited    created

Outputs:
  bucket_name: "my-bucket-edited-7705bd4"

Resources:
+ 2 created

Duration: 11s

# S3バケットが生成されたことを確認する
$ aws s3 ls
2020-08-22 17:55:01 my-bucket-7705bd4
```


いいところ

- リソースの状態をPulumi側で保持してくれる
 - 差分だけを自動で更新してくれる
- コンソール画面でリソースの状態、デプロイ履歴などを確認できる
- 静的型付け言語で書いた場合は、リソースに必要なパラメータをサジェストしてくれたたり型情報のサポートをしてくれる

リソース生成方法の比較 🤔

- Cloudformation
 - 専用の記法を勉強する必要がある
 - 宣言的に記載するため、コードの量が冗長になりがち
- SDK
 - APIを叩くだけなので、プロビジョニング失敗時のリカバリ用の実装が必要

料金 💰

- 1人なら無料
- 3人までは \$50/mo per team
- 3人以上は、\$75/mo per user

まとめ

- マルチクラウドでリソース運用する場合は便利
- 普通のコードに近いので、テストやCIしやすい
- アプリケーション用のコードとインフラ用のコードを分けなくて済む