サイオステクノロジー株式会社

次世代OSSデプロイツール Dacraneの開発を始めました！

クラウドインフラからアプリまですべてをコードでデプロイ

2023/12/15

サイオステクノロジー株式会社
プロフェッショナルサービス SL
池田 透

SIOS

アジェンダ

- 解決したい課題
- Dacraneが目指すツール
- Daccraneの構成
- Dacraneの動かし方
- Dacraneのメリットとデメリット
- 既存ツールとの違い

自己紹介

サイオステクノロジー株式会社
プロフェッショナルサービスSL
池田 透

- API Management、Webアプリバックエンド
- Node.js、TypeScript、C#、Go
- Docker、Kubernetes
- Azure、AWS
- Terraform、GitHub Actions

Dacraneが解決したい課題

デプロイ作業の例

みなさんはクラウド上で稼働するWebアプリ開発運用チームの一員です。

今週は開発したアプリとそれに伴うインフラ変更を行います。

プロジェクトではあらかじめ「デプロイ手順書」を用意していました。

予定の日時になったので、手順書を手に取り、デプロイ作業を始めました。

次になにが起こるでしょう？

環境構築は失敗する

「なんかエラーで失敗します。。」

「障害アラートが大量にあがっています。。」

失敗の原因は？

チームメンバーが集まって解決します。メンバー何人かが集まって数時間に渡り調べた結果、次の問題があることが分かりました。

- CLIコマンドを打ち間違えて、意図しない変更をしてしまった。
- クラウドのインフラ変更の内容に伴う、アプリの環境変数変更を飛ばしてしまった。
- テスト環境作成時とクラウドのGUIの仕様変更で、インフラの設定値が変わってしまった。
- 作業者の使っているデプロイで使うツールのバージョンが古かった。

どう解決するか？

今回の問題は解決しましたが、問題の解決に多くの時間を費やしてし、ユーザに大きな影響を与えてしまいました。今後どんな対策が考えられるでしょうか？

- メンバーに注意してミスしないように意識させる？
  - 注意してもミスは起きる。。
- シェルスクリプトを書いて自動化する？
  - コストかかるな。。手続きで途中で失敗時の回復とかも考えると。。。
- IaCを使う？
  - インフラ部分はいいけど、ミドルウェア、アプリはどうする？
  - IaCツールもサブプロセスでCLI呼び出しているので、作業者のツール統一できない。

根本的な失敗の原因は何か？

問題を引き起こした根本的な原因をもう少し整理します。

1. 開発者間の環境の違う
   - ✓ OSの違い、設定の違い、インストール済みミドルウェアの違い
2. 手順が多いすぎると正確に作業できない
   - ✓ 読み飛ばし、 勘違い、typo、多くのツールへの理解不足
3. 手順書は正確でない
   - ✓ 誤植、記載漏れ、ツールの仕様変更、GUIのデザイン変更

Dacraneの解決方法

誰もが同じインフラ定義を使って、一様にデプロイがしたい。

Dacraneは

- ✓ 手順書を作る代わりに、デプロイをコード化する
- ✓ 多くのツールと手順を統合して、デプロイを自動化する
- ✓ 個人のPC上でなく、コンテナ上でデプロイ動かす

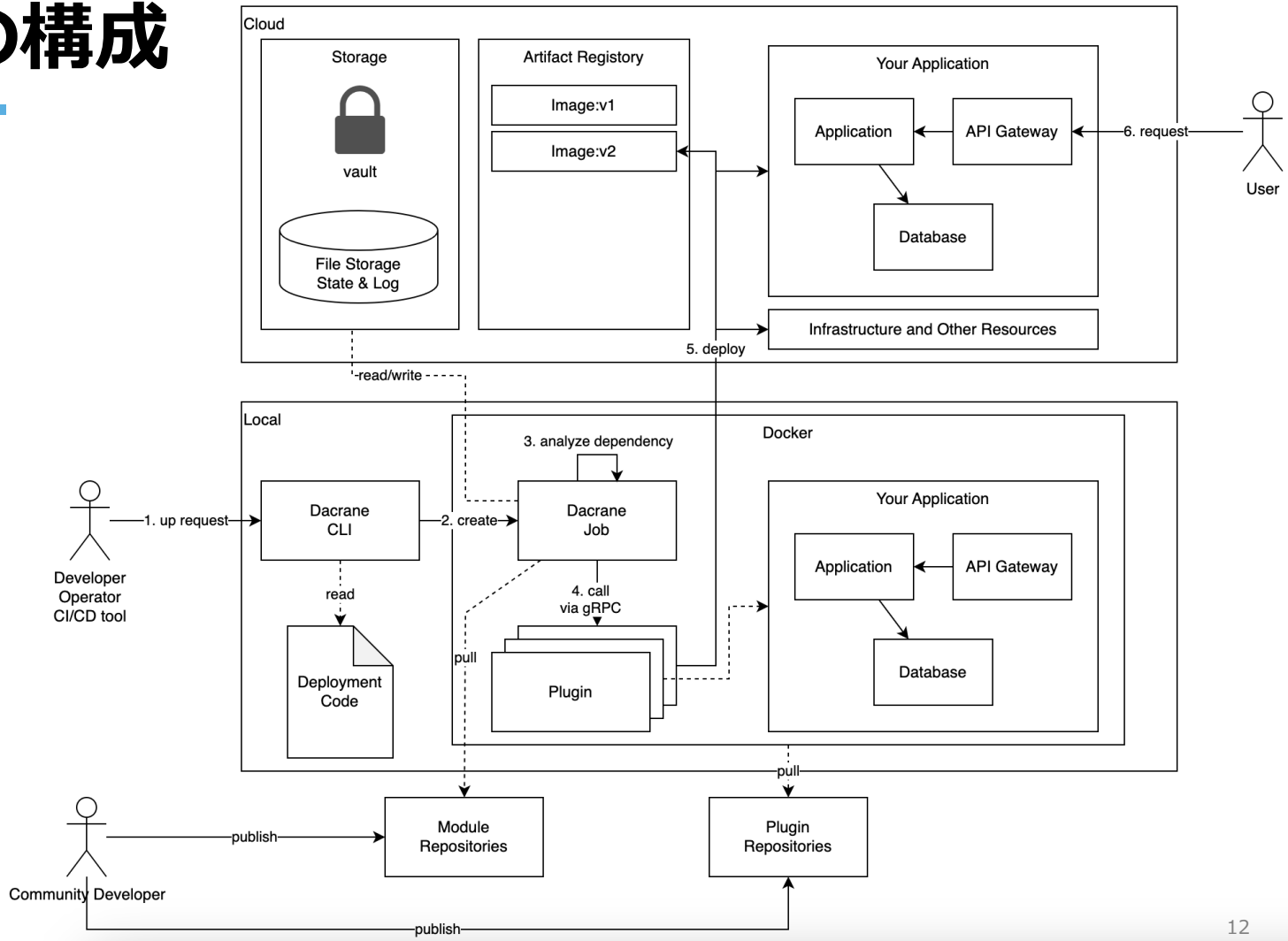
を実現して問題を解決する！

Dacraneが目指すツール

Daccraneをどんなツールとして提供したいか？

- ✓ 手に馴染むシンプルなCLIツール
- ✓ アプリが「動く」までをコード化
- ✓ 宣言的で再利用性の高いコード
- ✓ ローカルでもCI/CDでもどこでも動く
- ✓ 簡単に拡張できるプラガブルな設計
- ✓ コードとしてノウハウを共有できる

Dacraneの構成


Cloud
Storage
vault
File Storage
State & Log
Artifact Registory
Image:v1
Image:v2
Your Application
Application
API Gateway
Database
6. request
User
Infrastructure and Other Resources
5. deploy
read/write
Local
Docker
3. analyze dependency
Developer
Operator
CI/CD tool
1. up request
Dacrane
CLI
2. create
Dacrane
Job
read
Deployment
Code
4. call
via gRPC
pull
Plugin
Your Application
Application
API Gateway
Database
pull
publish
Module
Repositories
Plugin
Repositories
Community Developer
publish

Dacraneの動かし方

```
$ cat dacrane.yaml
  name: resource-group # モジュールの名前。
  parameter: # 再利用性を高めるためモジュールの引数を定義できる
    type: object
    required: ["prefix"]
    properties:
    prefix: { type: string }
  modules:
  - name: rg
    module: terraform/resource/azurerm_resource_group # terraformがインストールされたプラグインを呼び出している。
    argument:
     provider:
       features: {}
       client_id:       ${{ env.ARM_CLIENT_ID }}
       client_secret:   ${{ env.ARM_CLIENT_SECRET }}
       tenant_id:       ${{ env.ARM_TENANT_ID }}
       subscription_id: ${{ env.ARM_SUBSCRIPTION_ID }}
      resource:
        name: ${{ parameter.prefix }}-rg
        location: "Japan East”


$ dacrane apply resource-group my-resource-group -a ‘{ prefix: t-ikeda }’
[resource-group (terraform/resource/azurerm_resource_group)] Evaluating...
[resource-grpup (terraform/resource/azurerm_resource_group)] Evaluated.
[resource-gourp.rg (terraform/resource/azurerm_resource_group)] Creating...
…
```

Dacraneのメリットとデメリット

- メリット
  - インフラからアプリが「動く」までを宣言的にコード化できる。
  - DacraneとDockerだけインストールしておけば、ローカル環境だろうとCI/CD環境だろうサクッと動く。
  - プラグインをコンテナとして配布でき可搬性に優れる。
- デメリット
  - 宣言的なコードであり、「どうやって（How）」の表現は苦手。
  - コンテナ上で動かせる必要があるという制約がつく。
  - コンテナを使うので、ネイティブにツールをインストールより計算機リソースを使う。

既存ツールとの違い

DacraneってIaCツールや既存ツールとどう違うの？

項目	AWS CloudFormation	Azure Developer CLI	HashiCorp Terraform	HashiCorp Waypoint	Dacrane
インフラ	○	○	○	×	○
アプリ	×	○	×	○	○
対応クラウド	AWSのみ	Azureのみ	主要クラウド	主要クラウド	主要クラウド
可搬性	×(AWS上のみ)	×(MS製品群必要)	△(依存ツール別途インストール必要)	△(依存ツール別途インストール必要)	○
言語	YAML/JSON	YAML	HCL	HCL	YAML
成熟	○	△	○	△	×
ライセンス	(SaaS)	MIT	BSL	BSL	Apache 2.0

今後の展望

- 計画
  - 2024年1月 ベータ版リリース0.1.0 (コア機能が出揃う)
  - 2024年1月~ 品質の改善、機能の改善、プラグインの拡充、ドキュメント準備
  - 2024年夏 v1リリース
- 開発予定
  - 主要プログラミング言語ビルドサポート (JavaScript/TypeScript、Python、Java、Go、Rubyなど)
  - Terraform/ARM Template/CloudFormationサポート
  - カスタムプラグイン (自由に拡張できるようになります)
  - モジュール公開機能 (DacraneHub ？)

ぜひStarで応援してください

- GitHub Starで応援してください！！！
  - 今後も鋭意開発を進めていきます！
  - 開発が続けられるかどうかが注目度にかかっています！

「dacrane github」で検索


SIOS