

目的から考えるIaCツール選定の考え方

富士ソフト株式会社
安齋 寛之



- Amazon Connect Delivery
- IoT Services Competency
- Managed Service Provider
- Govt. Services Competency
- Oracle Services Competency
- Migration Services Competency
- Well-Architected Partner Program

自己紹介







Anzai Hiroyuki

安齋 寛之

AWS Ambassador

富士ソフト株式会社
ソリューション事業本部インフラ事業部クラウドソリューション部
リーダー

～主な業務内容～
インフラエンジニアとしてヘルプデスク・サポートデスク業務から開始し、データセンターでのサーバ設計・構築・運用経験を経て、現在はパブリッククラウドの設計・構築・運用まで幅広く担当。過去の経験から、お客様の希望に沿ったうえで効率的なシステム構築を行う事を得意としている。
得意分野は自動化。



FUJISOFT
INCORPORATED

富士ソフト 株式会社

AWSプレミアティアサービスパートナー

AWSの最新技術に精通したアプリケーションとインフラ技術者が、安心安全スピーディーなAWSへの移行から、AWSを活用してビジネス変革をご支援するDX（デジタルトランスフォーメーション）を実現しています。

国内のAWSのパートナーで技術認定（コンピテンシー）の取得数で2番目の企業です。（2023年3月現在）



- Amazon Connect Delivery
- IoT Services Competency
- Managed Service Provider
- Govt. Services Competency
- Oracle Services Competency
- Migration Services Competency
- Well-Architected Partner Program



アジェンダ

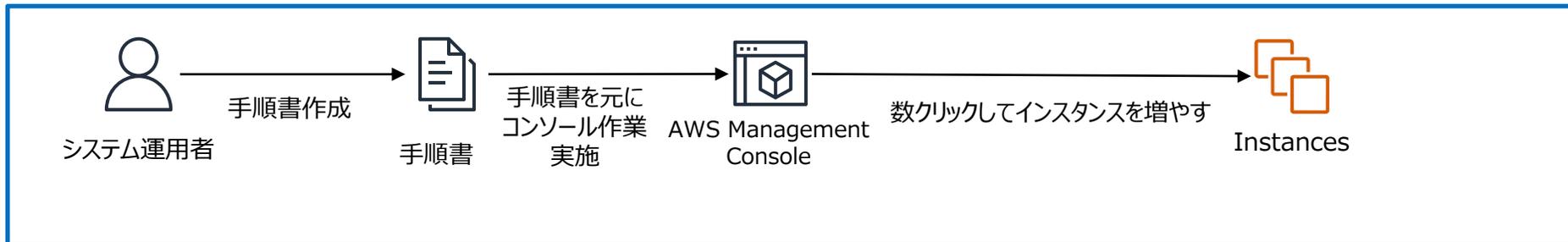
- 01 | IaCとは
- 02 | クラウドにおけるIaC
- 03 | 世の中でのIaC導入率について
- 04 | IaCを導入する前に考える必要があること
- 05 | ツール選定に関するノウハウ

IaC (Infrastructure as Code) とは？

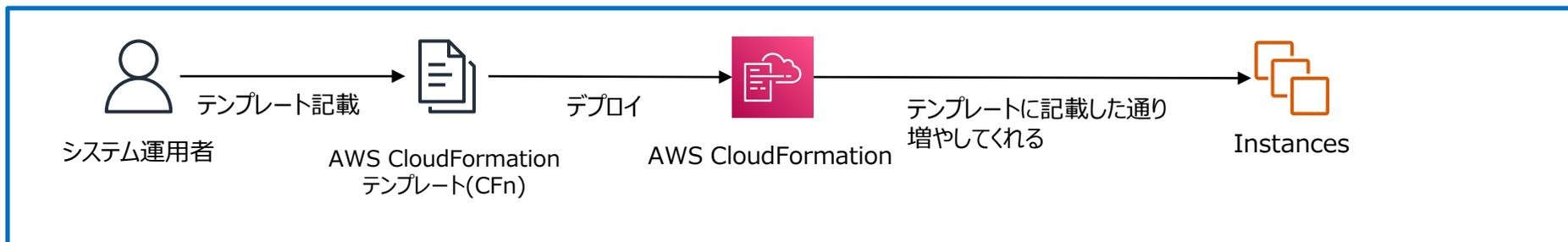
サーバーやネットワークをはじめとしたインフラの構成をプログラムのようにコード化し、その構築や管理を自動化する手法のこと

例) サーバを追加する場合

手
動
構
築



I
a
C



IaC (Infrastructure as Code) 利用するメリット

多くの場合一つのリソースを追加に伴って設定しなければならない項目が複数ありますが、IaCではそれを**モジュール（部品）として用意**しておく事で**設定が簡略化**でき、**設定漏れを解消**する事ができます。

例) サーバ追加に伴って大体実施する作業項目

監視設定追加したり...



バックアップ追加したり...



DISK追加したり...



ロードバランサー
の設定追加したり...



クラウドにおけるIaC (Infrastructure as Code)

クラウドを効率的に利活用する為にはIaCが欠かせない条件となります。
しかしながらIaCにはメリットとデメリットが存在するのも現実です。

メリット



- ・インフラの再利用が可能
- ・オペレーションミスの削減
- ・コードレビューとメンテナンスが容易に
- ・デプロイが簡単になる
- ・静的コード解析が可能になる

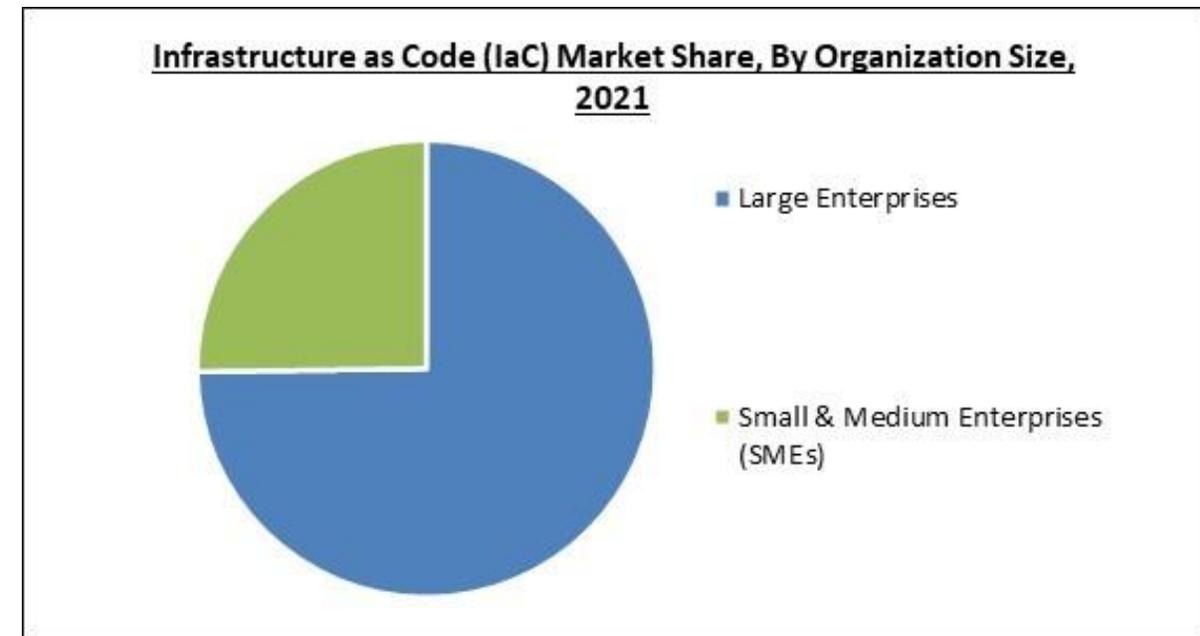
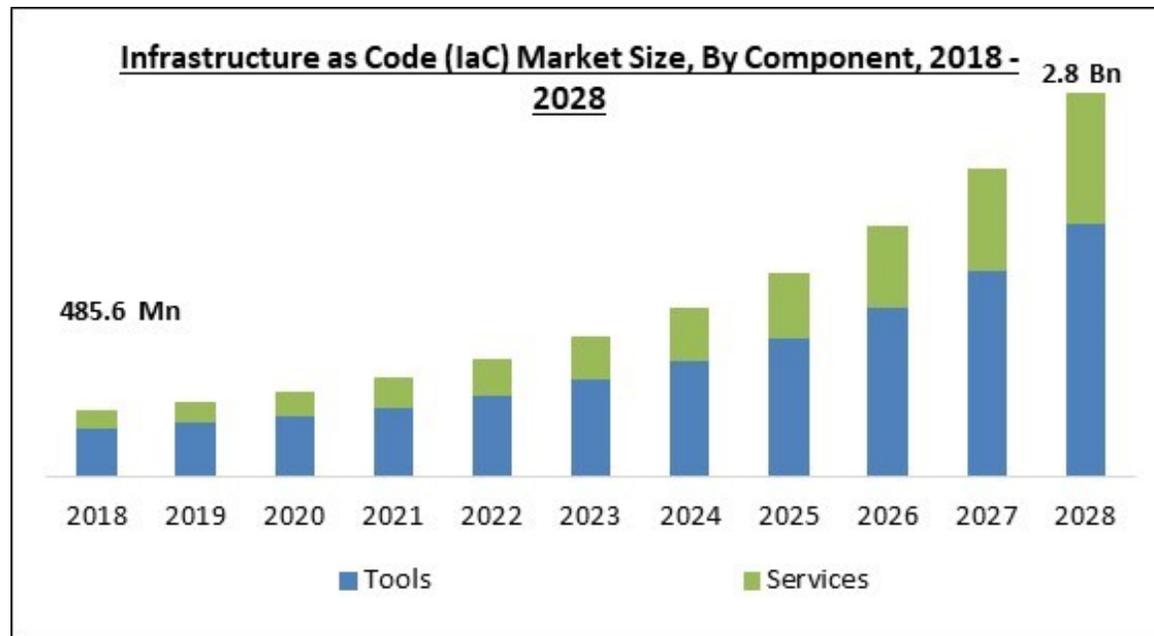
デメリット



- ・初期導入に時間がかかる
- ・学習コストがかかる
- ・手動変更した場合、実機とテンプレートが乖離してしまう

IaC (Infrastructure as Code) の市場

RESEARCH AND MARKETSの調査・分析によると2028年までにIaCの市場規模は28億ドルに達すると予想されており、年平均成長率は21.9%の割合で成長することも予想されています。
IaCを導入している企業規模の割合では大企業の方が中小企業よりも導入率が高い傾向があります。



IaC (Infrastructure as Code) を導入する際に考える事

Point

1

知識やスキルの習得

どのツールを利用するにしても、ツールの使用方法や記述する構文（YAML、JSON等）の習得が必要になります。

Point

2

ガイドラインの整備

IaCによって管理ができない例外が発生した場合の対応方針や、管理対象の基準、使うサービスの選定等を決めます。

Point

3

バージョン管理の必要性

一度作ったら作りっぱなしではなく、その後も継続して更新していく際にバージョン管理は必ず必要になります。



GitHub



GitLab

IaC (Infrastructure as Code) のツールにおける役割

IaCにおいてはツールで管理できる対象と範囲を理解する事が大切です。



ANSIBLE



CHEF

HashiCorp
Packer

puppet

Application

data

MW

OS

Server

Storage

Network

HashiCorp
Terraform

Pulumi



AWS CloudFormation

OS・MW管理ツール比較

	Ansible	CHEF	puppet	Packer
ライセンス	OSS ※有償版もあり	有償ライセンス	OSS	OSS
対応範囲	OS, MW	OS, MW	OS, MW	OS, MW
対応プラットフォーム	依存なし	依存なし	依存なし	依存なし
エージェント	必要なし(push型)	必要あり(pull型)	必要あり(pull型)	イメージビルド
構文	YAML	Ruby base DSL	Ruby base DSL	Amazon EC2 Image Builder
AWSマネージドサービス使うなら	AWS Systems Manager Run Command ※Windows非対応	AWS OpsWorks	AWS OpsWorks	Amazon EC2 Image Builder

プラットフォーム管理ツール比較

	Terraform	Pulumi	AWS CloudFormation
ライセンス	OSS ※有償版もあり	OSS ※個人利用に限る	無料
対応範囲	マネージドサービス	マネージドサービス	マネージドサービス
対応プラットフォーム	パブリッククラウド全般	パブリッククラウド全般	AWSのみ
構文	HCL (HashiCorp Configuration Language)	YAML, JavaScript, Python, Go, .NET, Java, TypeScript ...	YAMLまたはJSON
その他	Terraform Enterprise (有償版) もあるが無償版でも充分機能が充実している	個人利用ではオープンソースライセンスで料金がかからない。複数名で利用する場合は有償プランの購入が必要となる	AWSが提供するツール AWS CDKとも連携しているので開発者も使いやすい

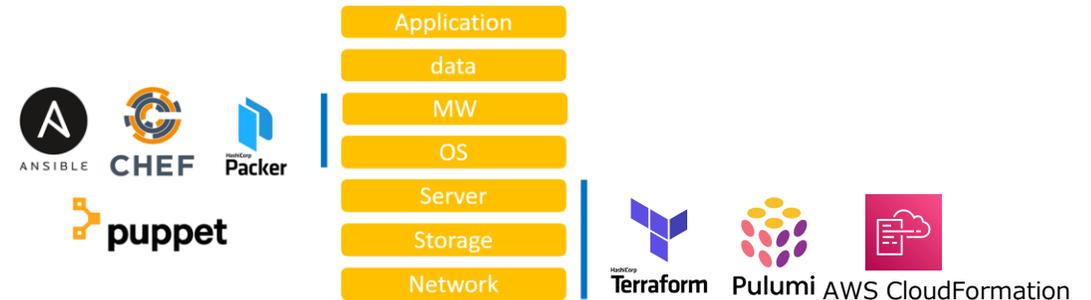


まとめ

IaCのツール選定においては、以下の項目を考慮検討する事がポイントです

管理対象・項目

- クラウドサービス全般なのか
- OSやアプリケーションまでなのか



管理範囲

- ハイブリットクラウドやマルチクラウドを管理したいのか
- シングルクラウドのみなのか

使い慣れた構文

- ツールを導入しても使う人がいないと意味がないので
親しみやすさや使った人がいる事が多いものを利用しましょう



FUJISOFT

ご清聴ありがとうございました。

OS・MWレイヤー管理ツール (Ansible)

■ カバー範囲

OS (Windows, Linux) 、 MW (Apache, Nginx etc...)

■ 対応プラットフォーム

プラットフォーム依存なし、エージェントなし(Push型)

■ 構文

YAML

■ AWSマネージドサービス使うなら

AWS Systems Manager Run Command

・AWS-ApplyAnsiblePlaybooks (**Windows対応してない。**)



ANSIBLE

OS・MWレイヤー管理ツール (CHEF)

■ カバー範囲

OS (Windows, Linux) 、MW (Apache, Nginx etc…)

■ 対応プラットフォーム

プラットフォーム依存なし、エージェントあり (Pull型)

■ 構文

Ruby

■ AWSマネージドサービス使うなら

AWS OpsWorks

・AWS OpsWorks for Chef Automate



OS・MWレイヤー管理ツール (puppet)

■ カバー範囲

OS (Windows, Linux) 、 MW (Apache, Nginx etc…)

■ 対応プラットフォーム

プラットフォーム依存なし、エージェントあり (Pull型)

■ 構文

Ruby

■ AWSマネージドサービス使うなら

AWS OpsWorks

・AWS OpsWorks for Puppet Enterprise



OS・MWレイヤー管理ツール (Packer)

■ カバー範囲

OS (Windows, Linux) 、MW (Apache, Nginx etc...)

■ 対応プラットフォーム

プラットフォーム依存なし、イメージビルドツール

■ 構文

json

■ AWSマネージドサービス使うなら

Amazon EC2 Image Builder

※Packer互換ではない



クラウドリソース管理ツール（Terraform）

■ カバー範囲

クラウドのマネージドサービス全般

■ 対応プラットフォーム

パブリッククラウド全般

■ 構文

HCL（HashiCorp Configuration Language）

■ その他

有償版もあるが無償版でも充分機能が充実している



クラウドリソース管理ツール (Pulumi)

■ カバー範囲

クラウドのマネージドサービス全般

■ 対応プラットフォーム

パブリッククラウド全般

<https://www.pulumi.com/docs/reference/pkg/>

■ 構文

YAML, JavaScript, Python, Go, .NET, Java, TypeScript..

■ その他

オープンソースなので個人利用では料金はかからない
複数名で利用する場合は有償プランの購入を推奨



クラウドリソース管理ツール (AWS CloudFormation)

■ カバー範囲

AWSのマネージドサービス全般

■ 対応プラットフォーム

AWSのみ対応

■ 構文

YAML,JSON

■ その他

AWSが提供するツール

AWS CDKとも連携しているので開発者も使いやすい



AWS CloudFormation