HashiCorp

Network as Code
自動化のこれから

Shogo Katsurada
Partner Solutions Engineer at HashiCorp
He/Him

@shogokatsurada

Shogo Katsurada / 桂田 祥吾

Partner Solutions Engineer at HashiCorp

HashiCorp Japan にて CSP,SI,戦略アライアンスパートナー技術担当

前職は、シスコシステムズにて、新卒入社後プリセールスエンジニア、ビジネス開発を担当し、
AppDynamics 事業部 チャネルディレクター兼グロースイニシアチブ担当部長
L1（物理）からL7（アプリ）まで色々やってきました

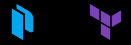
Twitter: @shogokatsurada

Agenda

本日のアジェンダ

1. **クラウド移行フェーズと Infrastructure as Code**
2. **アプリケーションの進化**
   デモ
3. **まとめ**
   自動化のこれから

HashiCorp はマルチクラウド時代のインフラ自動化を支援します

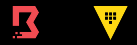


プロビジョニング

インフラストラクチャasコード
コンプライアンス＆ガバナンス
セルフサービスインフラ

ネットワーキング

サービスレジストリ＆ディスカバリ
セキュアネットワーキング
サービスメッシュ
自動化ネットワーク



セキュリティ

シークレット管理
暗号化
アドバンスドデータプロテクション

アプリケーション

ワークロードオーケストレーション
アプリケーションネットワーク
開発者中心のアプリケーション展開

PLATFORM

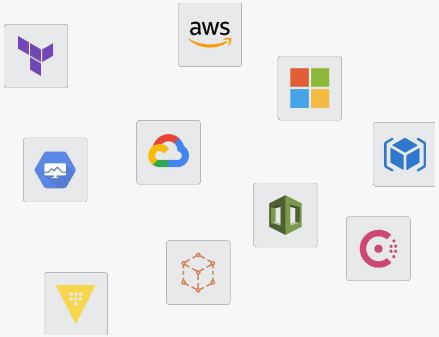
Cloud or **Self-managed**

Identity | Audits & Logging | Billing | Upgrades | Patching | Autoscaling

01

クラウド移行フェーズと Infrastructure as Code

クラウド移行フェーズ
限定的な利用からIndustrialized(工業化)へ

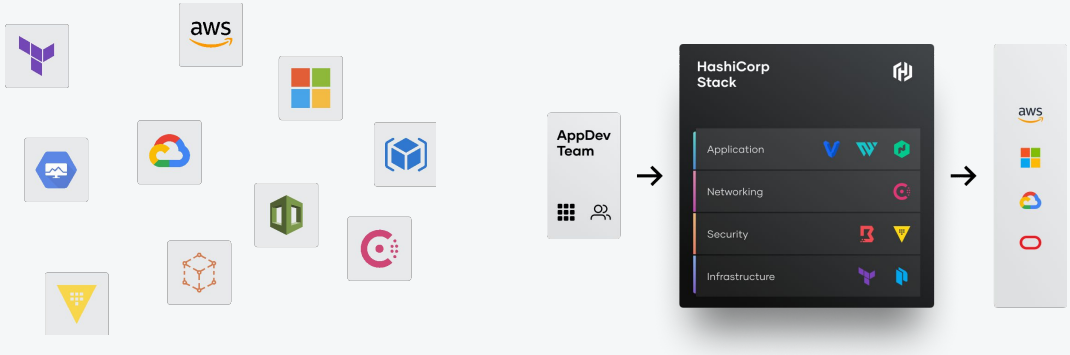


STAGE 1

Tactical cloud

エンジニアリングチームがクラウドサービスを活用し始める

クラウド移行フェーズ
限定的な利用からIndustrialized(工業化)へ



STAGE 1

Tactical cloud

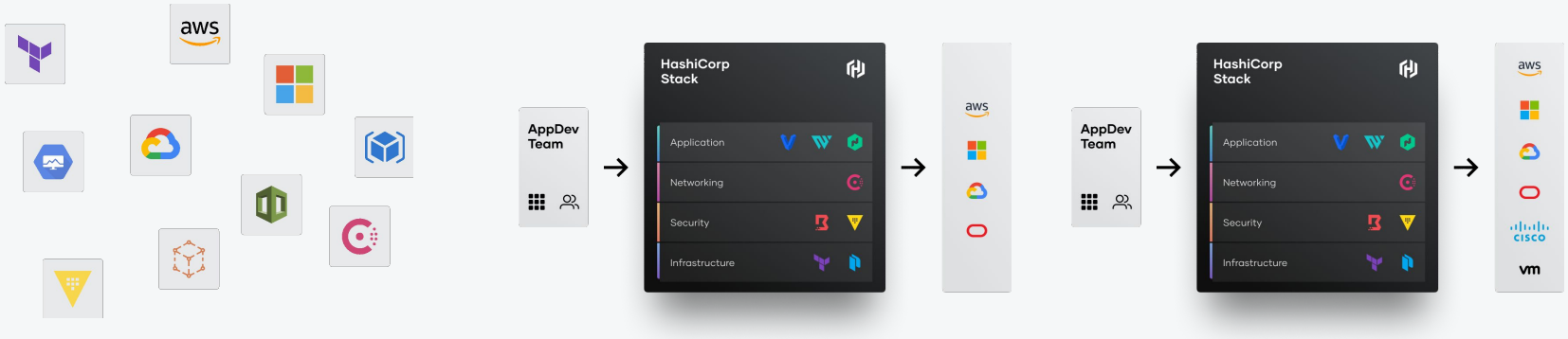
エンジニアリングチームがクラウドサービスを活用し始める

STAGE 2

Cloud program

運用,セキュリティ,ネットワークの各チームは共通のインフラ基盤を採用

クラウド移行フェーズ
限定的な利用からIndustrialized(工業化)へ



STAGE 1 | STAGE 2 | STAGE 3

Tactical cloud

エンジニアリングチームがクラウドサービスを活用し始める

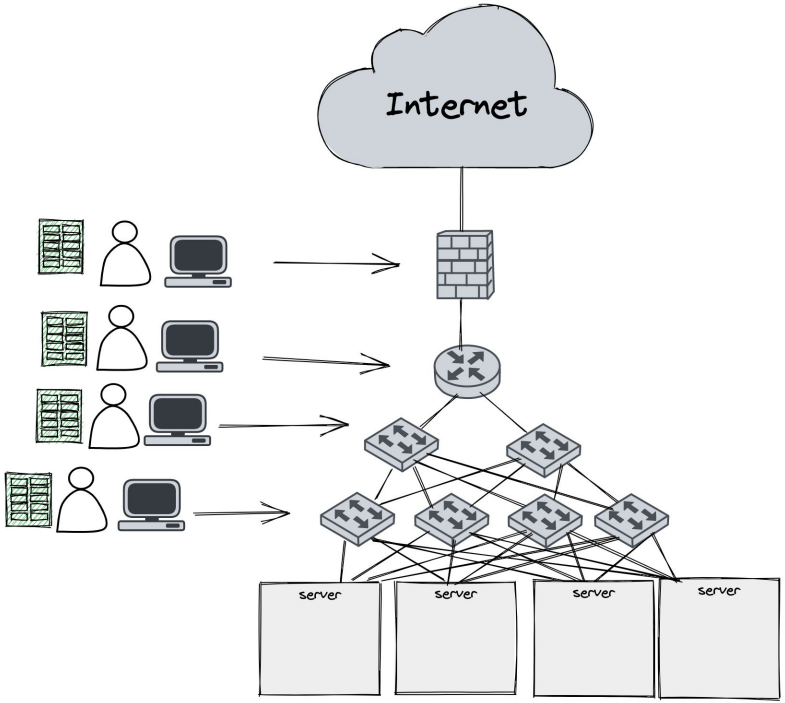
Cloud program

運用,セキュリティ,ネットワークの各チームは共通のインフラ基盤を採用

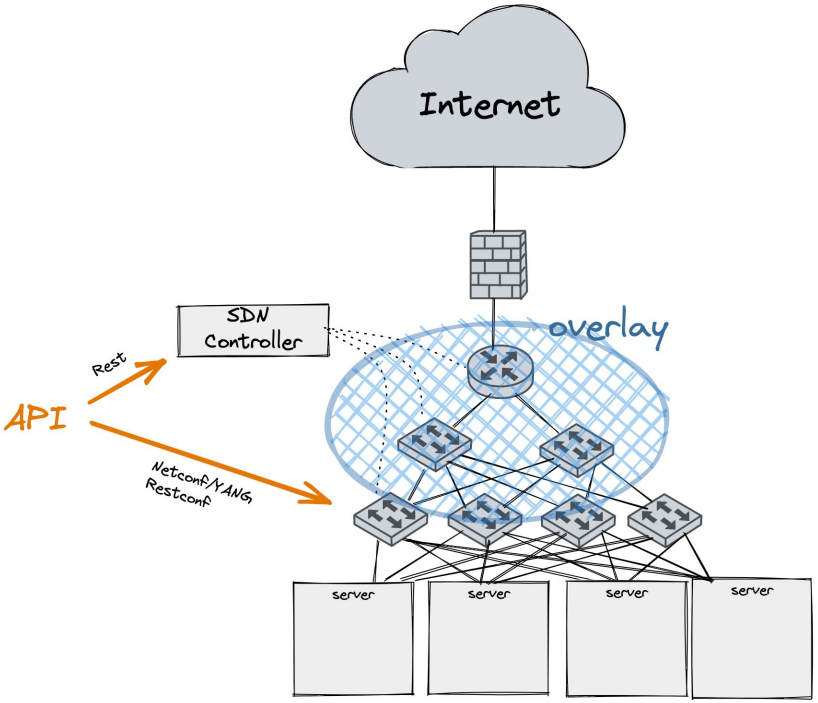
Private estate

プライベートクラウドやオンプレミスなど、幅広く適用されるクラウドオペレーティングモデル

運用から見るプログラマビリティ/ SDN



専用コマンド / CLIによる機器の設定・運用

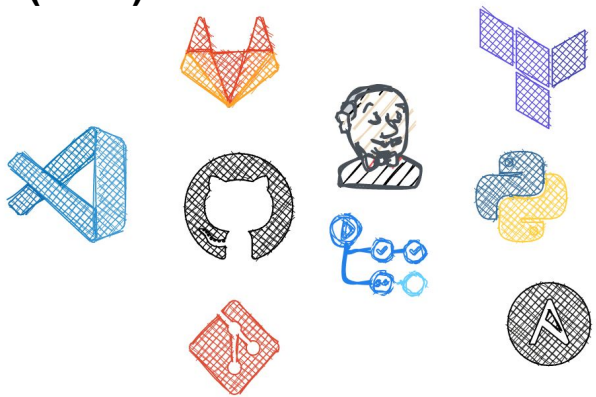


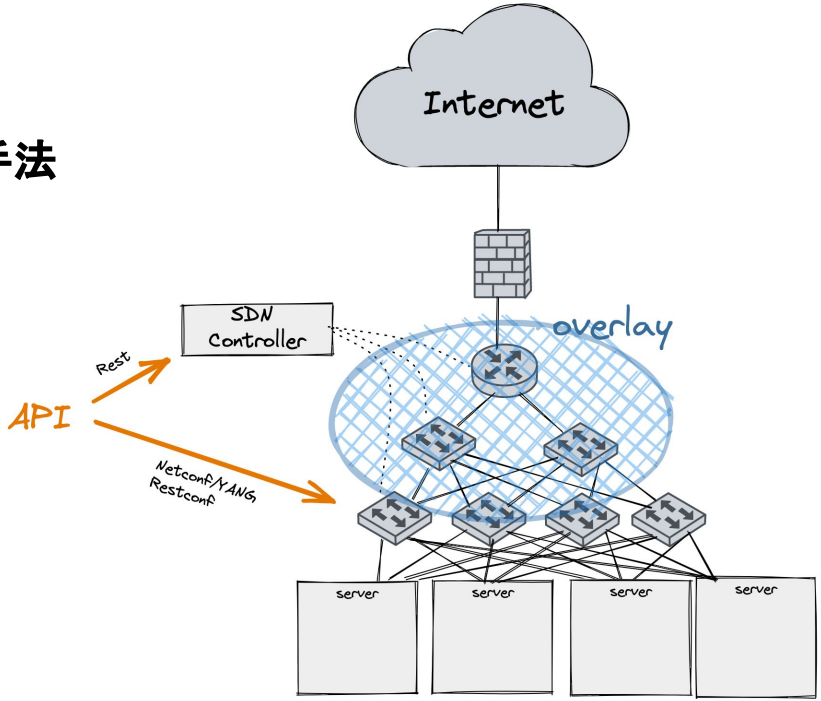
プログラマブルインフラ・ SDN環境における運用

運用から見るプログラマビリティ/ SDN

DXを加速するSDNの価値:

- アプリケーションによるNWの操作
- モダンなアプリ開発で培われたツールセット、手法を活用することができる
- ExcelからGit (VCS)へ
- NetDevOps



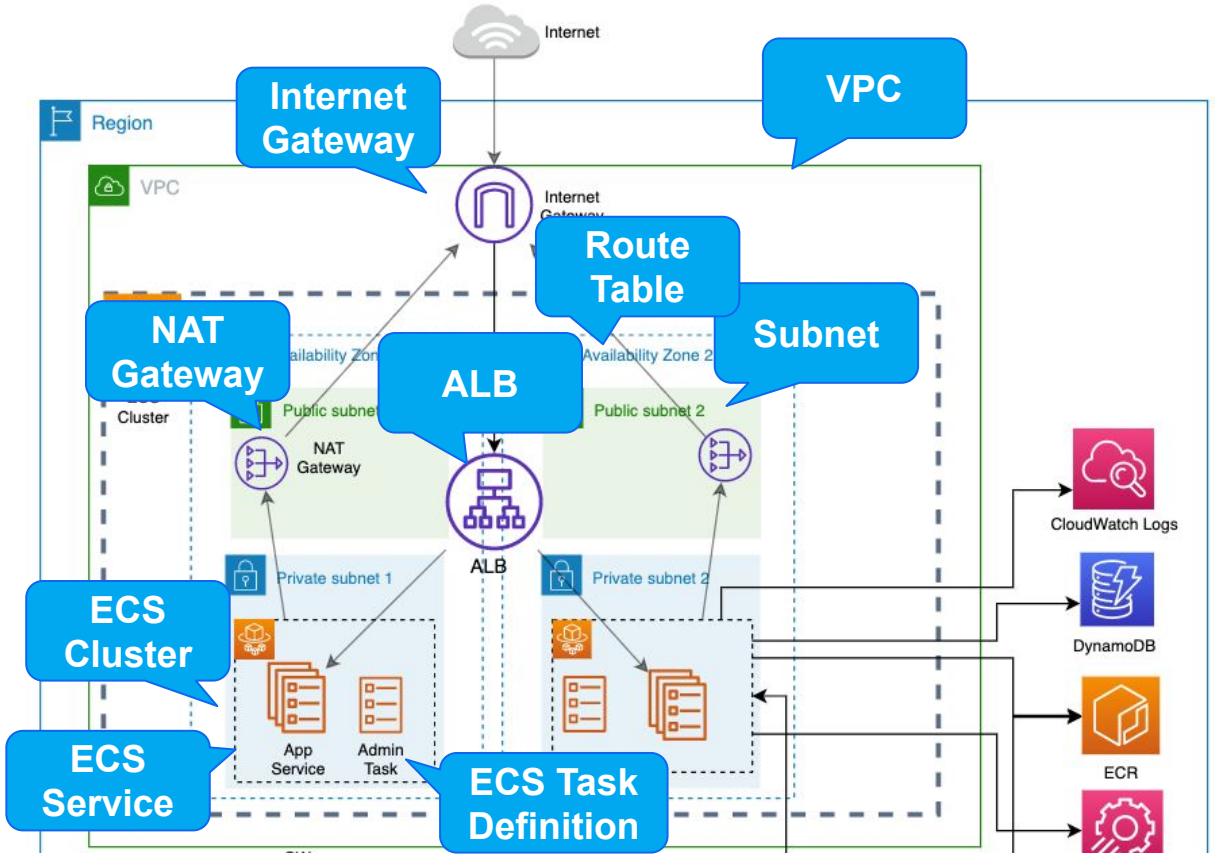


プログラマブルインフラ・SDN環境における運用

Infrastructure as Code

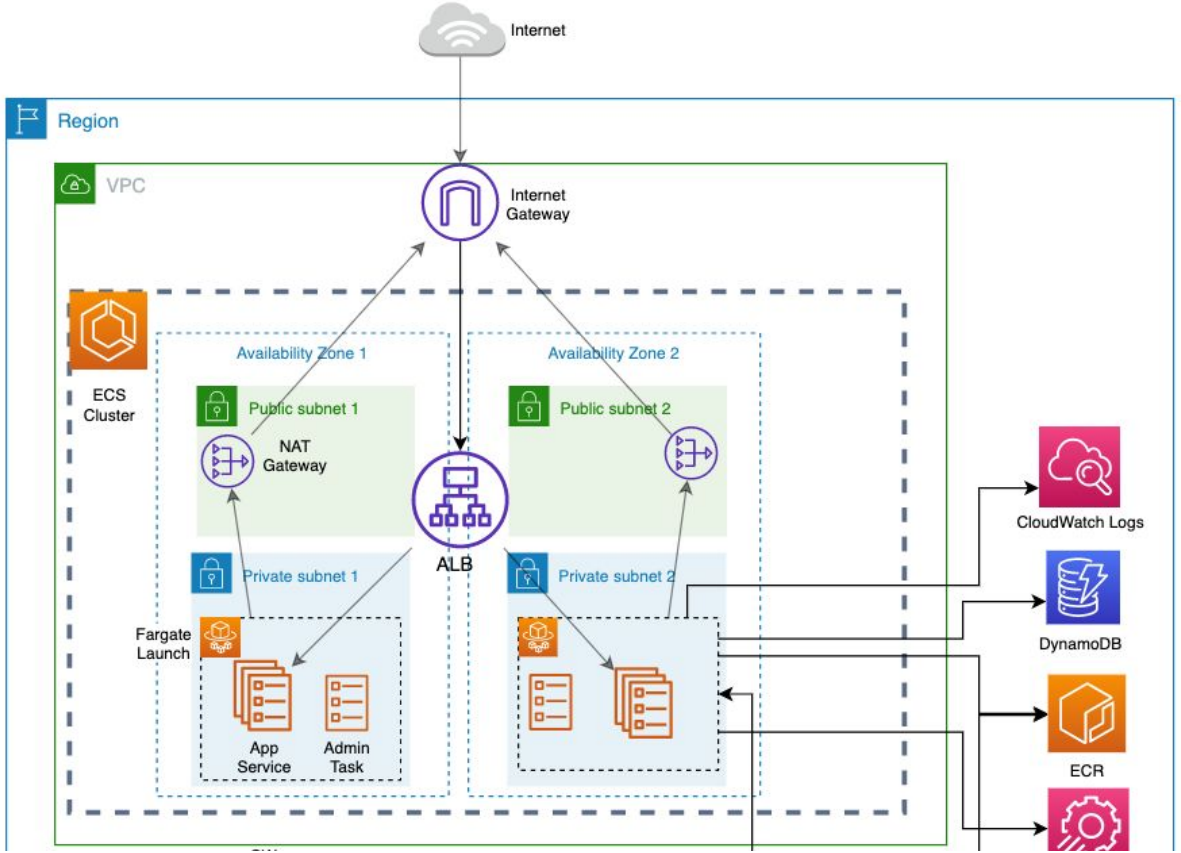
IaC のおさらい

クラウド上で動くアプリの構成例



クラウド上で動くアプリの構成例

- VPC
- Internet Gateway
- EIP
- Subnet
- Route
- Security Group
- IAM Group
- IAM Policy
- IAM User
- CloudWatch
- RDS
- DynamoDB
- ECR
- ECS Cluster
- ECS Task Definition
- ECS Service



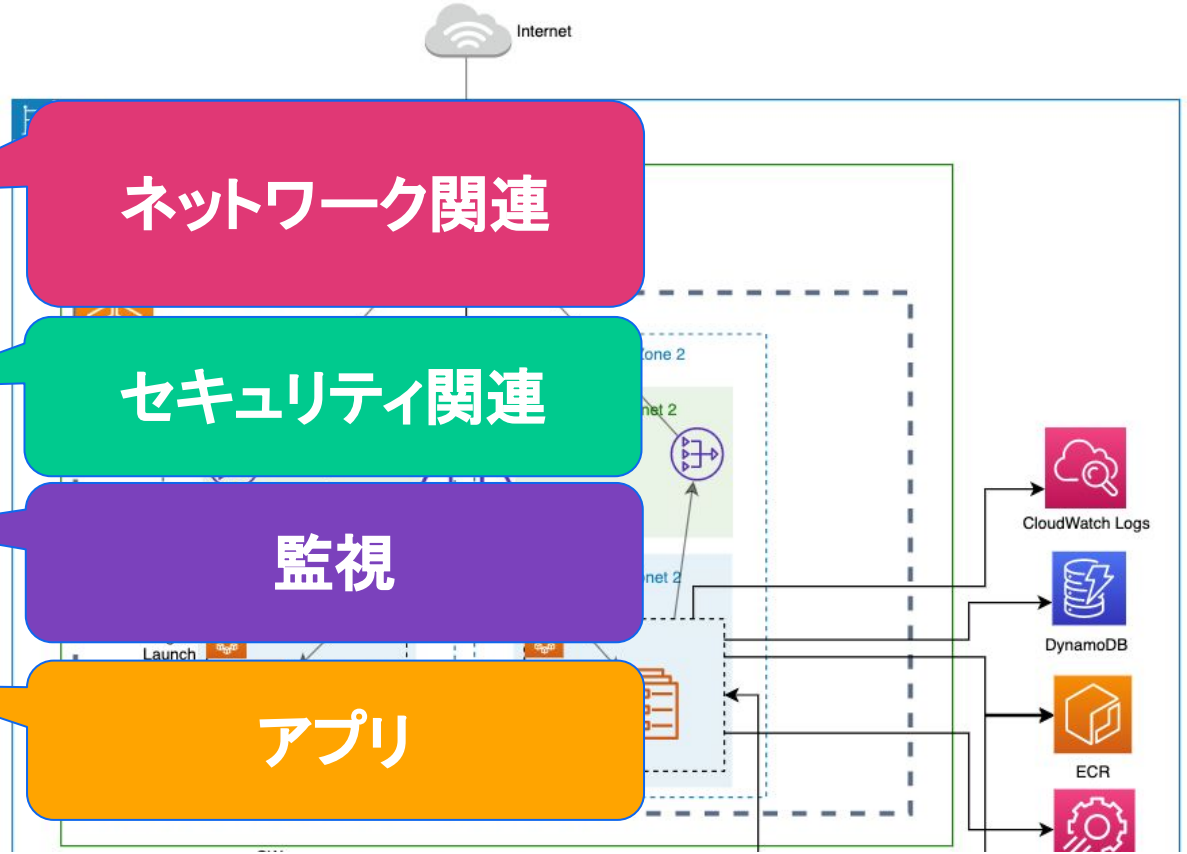
クラウド上で動くアプリの構成例

- VPC
- Internet Gateway
- EIP
- Subnet
- Route

- Security Group
- IAM Group
- IAM Policy
- IAM User

- CloudWatch

- RDS
- DynamoDB
- ECR
- ECS Cluster
- ECS Task Definition
- ECS Service



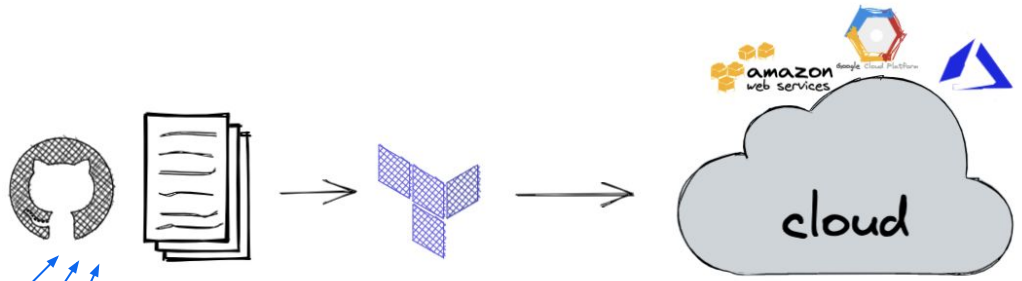
クラウド上で動くアプリの構成例

- VPC
- Internet Gateway
- EIP
- Subnet
- Route

- Security Group
- IAM Group
- IAM Policy
- IAM User

- CloudWatch

- RDS
- DynamoDB
- ECR
- ECS Cluster
- ECS Task Definition
- ECS Service

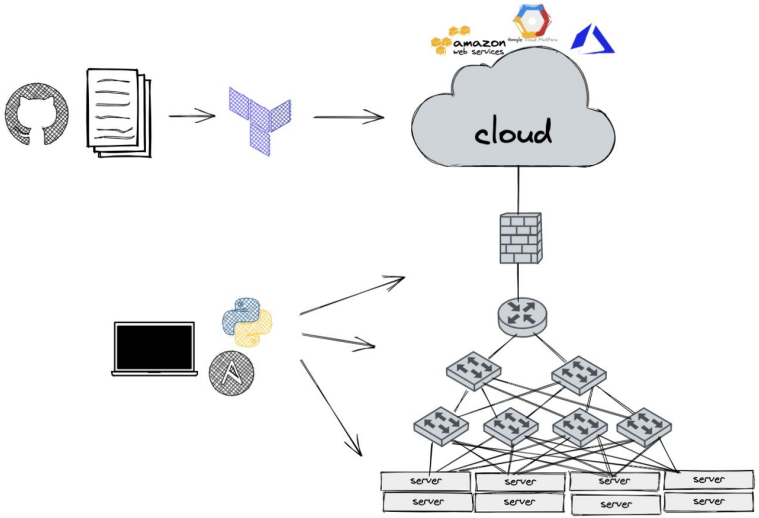


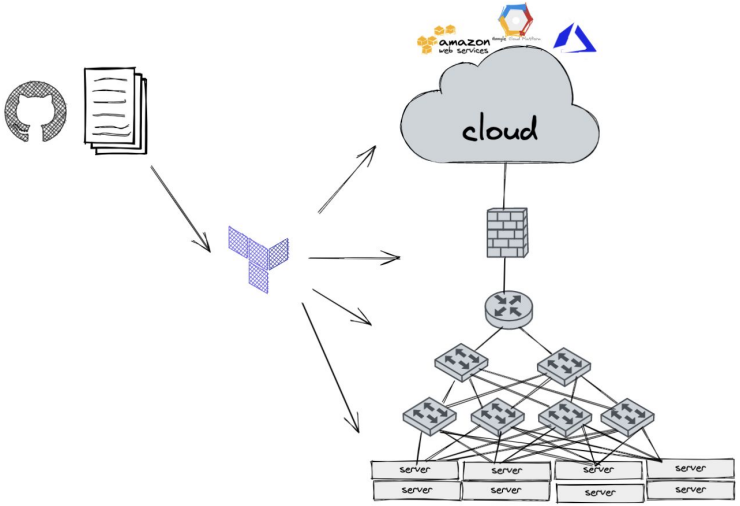
クラウドの自動化では、

- ネットワーク
- セキュリティ
- 監視設定
- アプリケーション

ほとんどのコンポーネントが、一貫したワークフローで運用されている (DevOps)

クラウド運用と同様の
運用パイプラインに組み込むことができるか？





ハイブリッド環境を、
一貫したワークフローで運用

NW関連 Terraform プロバイダの対応状況

2022年1月14日 (Janog49) での傾向 → 10月27日 現在

- プロバイダ数
  - Networking カテゴリ: 103 → **166**
  - Security & Authentication カテゴリ: 155 → **224**

約9ヶ月のうちに
対応プロバイダの数が 50% 程増加
多くの主要ベンダーの機器は対応してきている

ユースケースの例

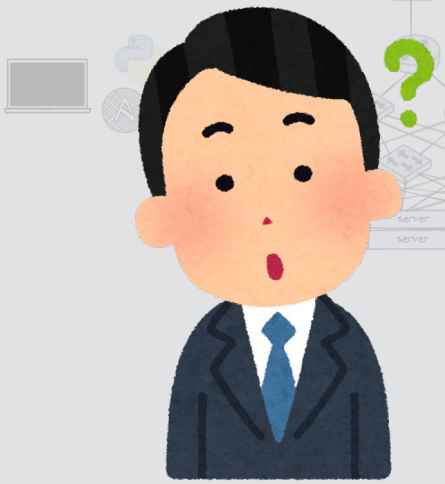
1. Azure/AWS上で 複数の拠点への仮想インスタンス (Fortinet / zScaler / Palo alto / Cisco など) のデプロイをして、クラウド内の VPC/VNETの設定、オンプレミスのFWの設定変更、VPN、URLフィルタリング等の設定を、一つの Infrastructure as Code のワークフローで完了

2. クラウドアプリのデプロイから、監視設定 datadog, ThousandEyes などの設定を一連の流れとして IaCで管理

クラウド・オンプレミス全て IaC で運用を統一することで、ヴァージョン管理、レビューのしやすさ、コードの再利用、手順書の簡略化など
IaC のメリットが享受

クラウド運用と同様の
運用パ　　能に

これだけで良いだろうか？



- **Day0, コードの再利用は、IaC できるだけでも、運用メリットはありそうではあるが..**
  Day2の運用はどうなる？

- **より自動化を進めるためには？**
  ハイブリッド環境を、一貫したワークフローで運用

DevOps の理想

Continuous Integration

Build

Test

Merge

Continuous Delivery

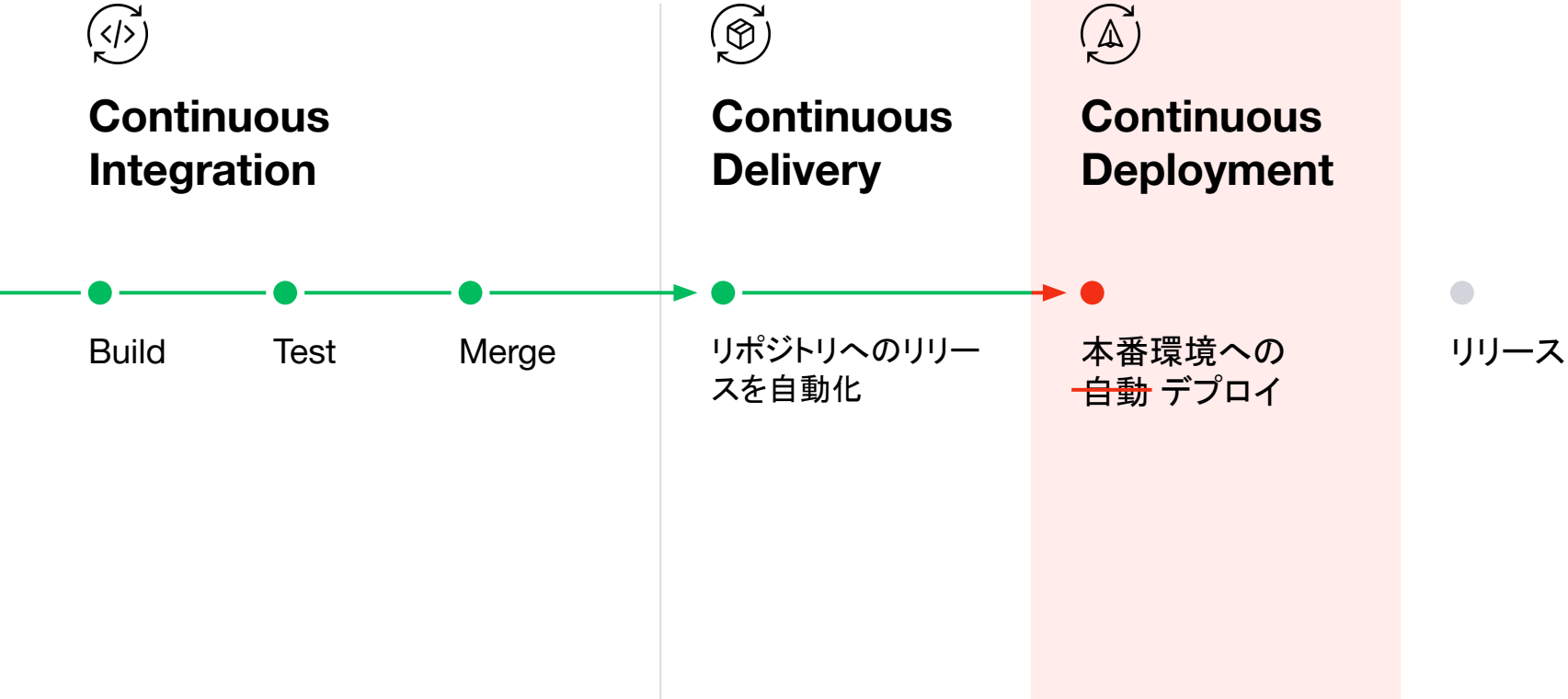
リポジトリへのリリースを自動化

Continuous Deployment

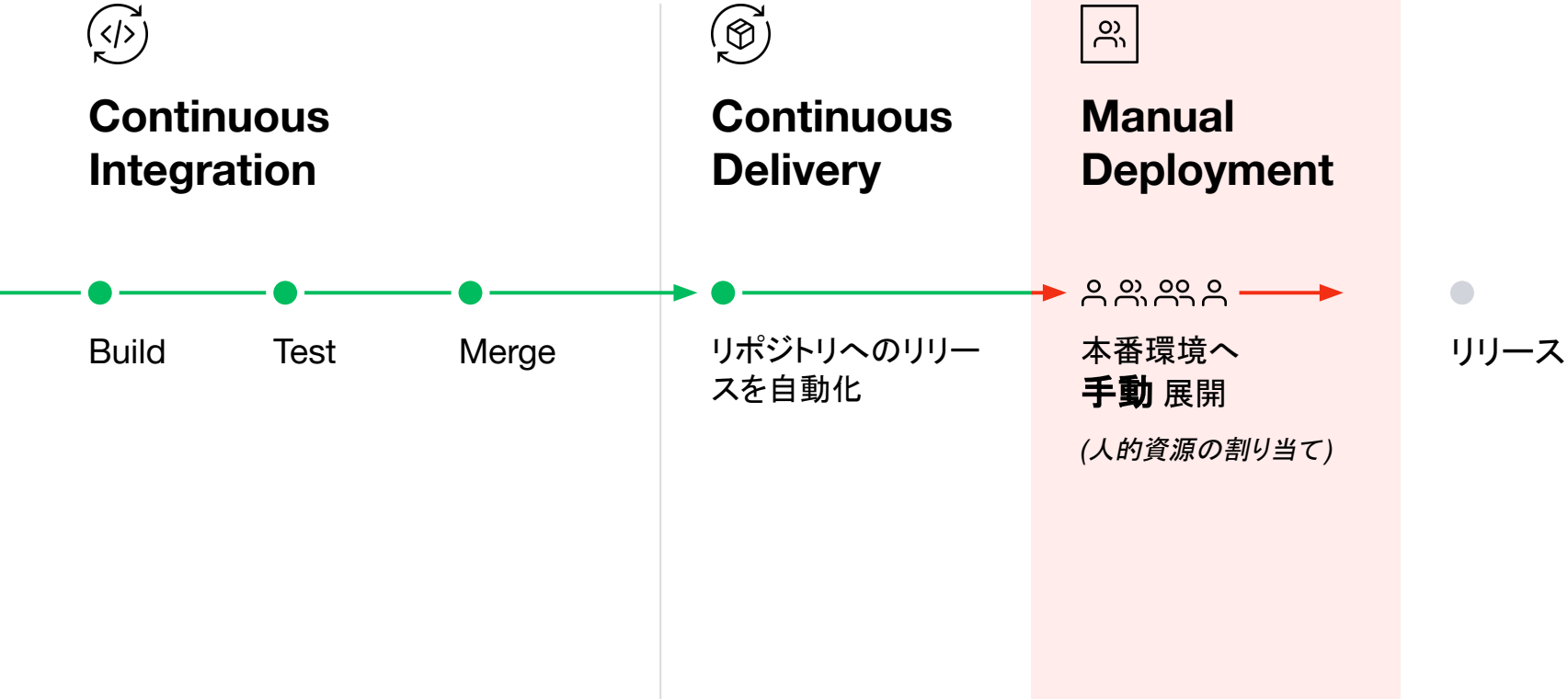
本番環境への自動デプロイ

リリース

DevOps の現実


Continuous Integration
Continuous Delivery
Continuous Deployment
Build
Test
Merge
リポジトリへのリリースを自動化
本番環境への自動デプロイ
リリース

~~解決方法~~ 取り急ぎ対応


Continuous Integration
Continuous Delivery
Manual Deployment
Build
Test
Merge
リポジトリへのリリースを自動化
本番環境へ
手動 展開
(人的資源の割り当て)
リリース

手動のチケットベースのワークフロー

アプリ開発者

アプリのワークロードを拡張の必要がある

チケットに起票したほうが良いですか？

サーバー管理者

新規サーバー/ワークロード用にチケット起票

ネットワーク管理者

ワークロードに新しいIPアドレスを割り当て

アドレスプールのメンバーを更新

ネットワークスイッチの設定を更新

ネットワークスイッチのACLを更新

セキュリティチーム

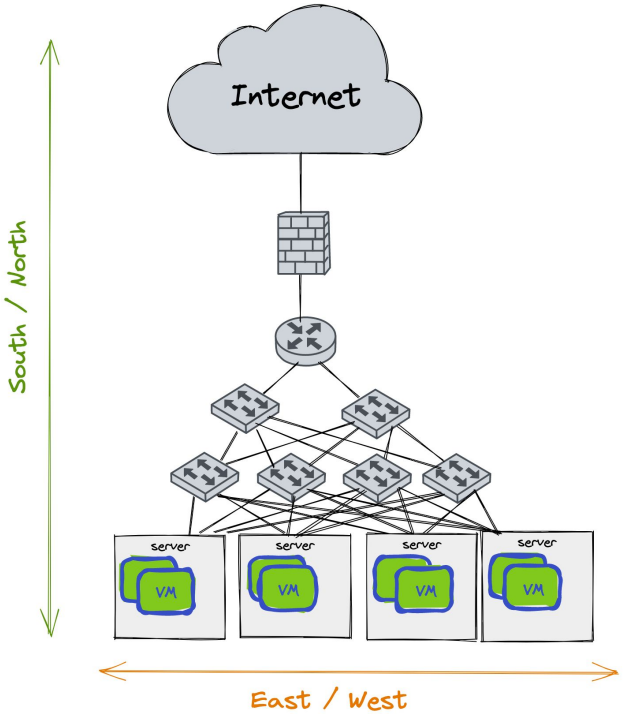
FWのアドレスグループポリシーを更新

FWのNATポリシーを更新

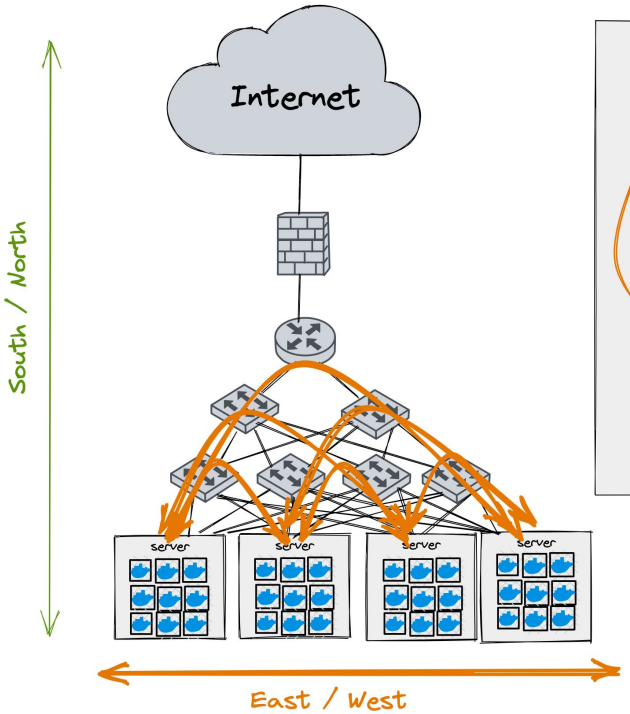
02

アプリケーションの進化

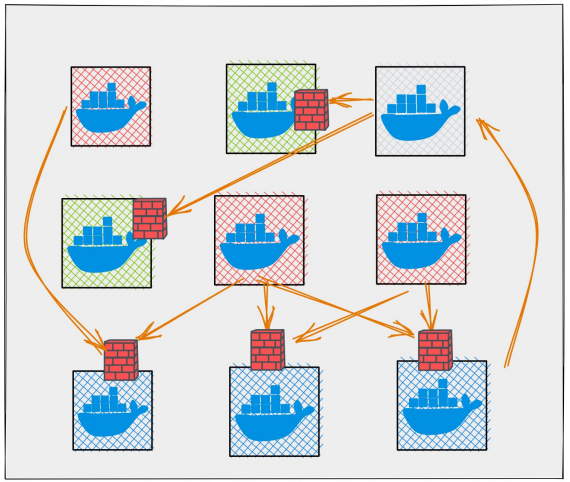
アプリケーションのモダナイゼーション



モノリシックなアプリ



マイクロサービス化により East / West トラフィックが増加



動的にスケールするアプリケーション間のアクセス制御を、旧来の運用だけで行うのは不可能

アプリケーションのモダナイゼーション

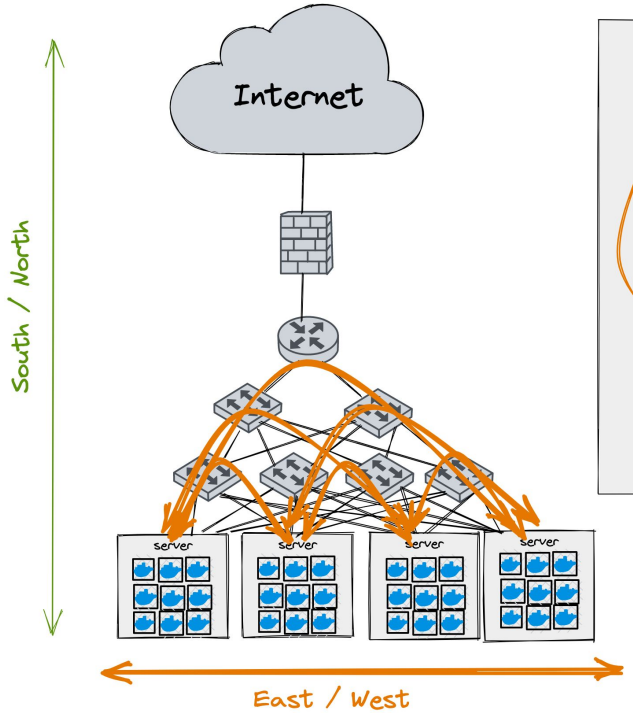
アプリの要求：
動的にスケールするアプリの

- ディスカバリ
- ヘルスチェック
- アクセス制御
- トラフィック制御

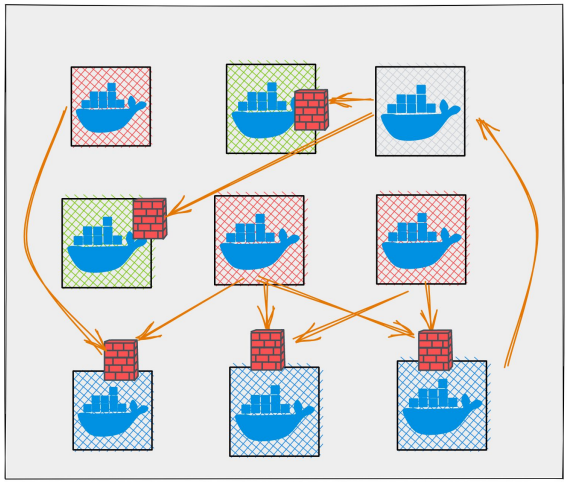
が求められる

アプリケーションレイヤ
ネットワーク

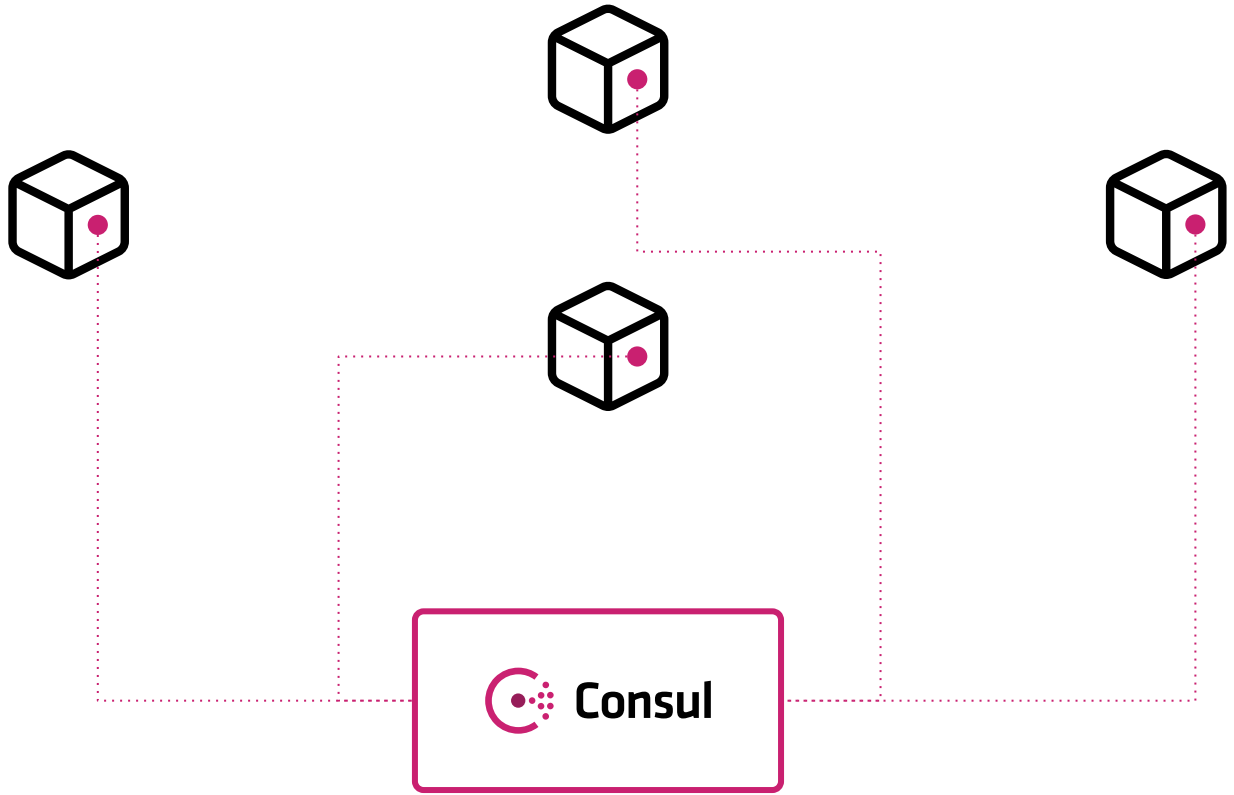
- サービスディスカバリ
- サービスメッシュ



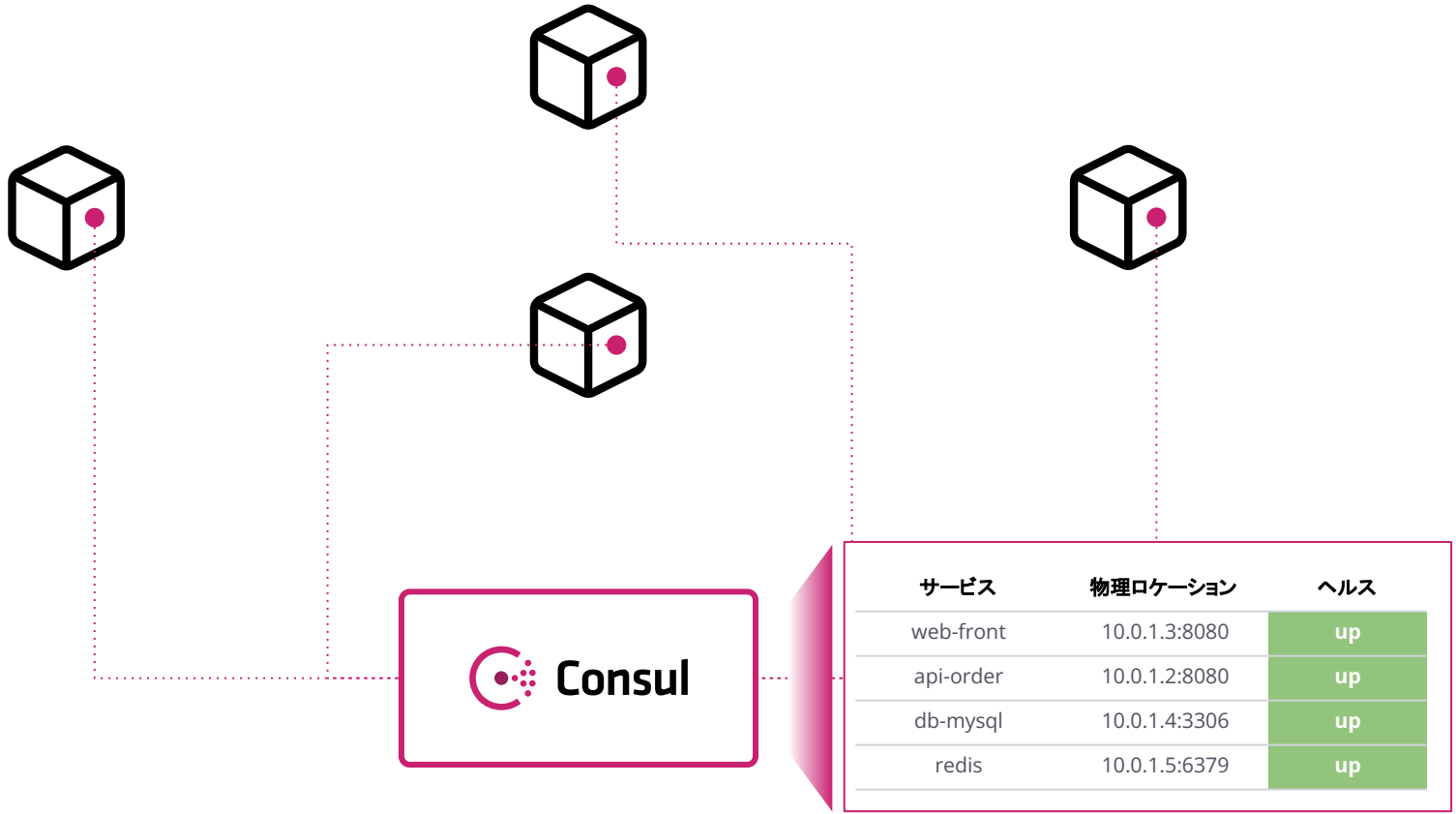
マイクロサービス化



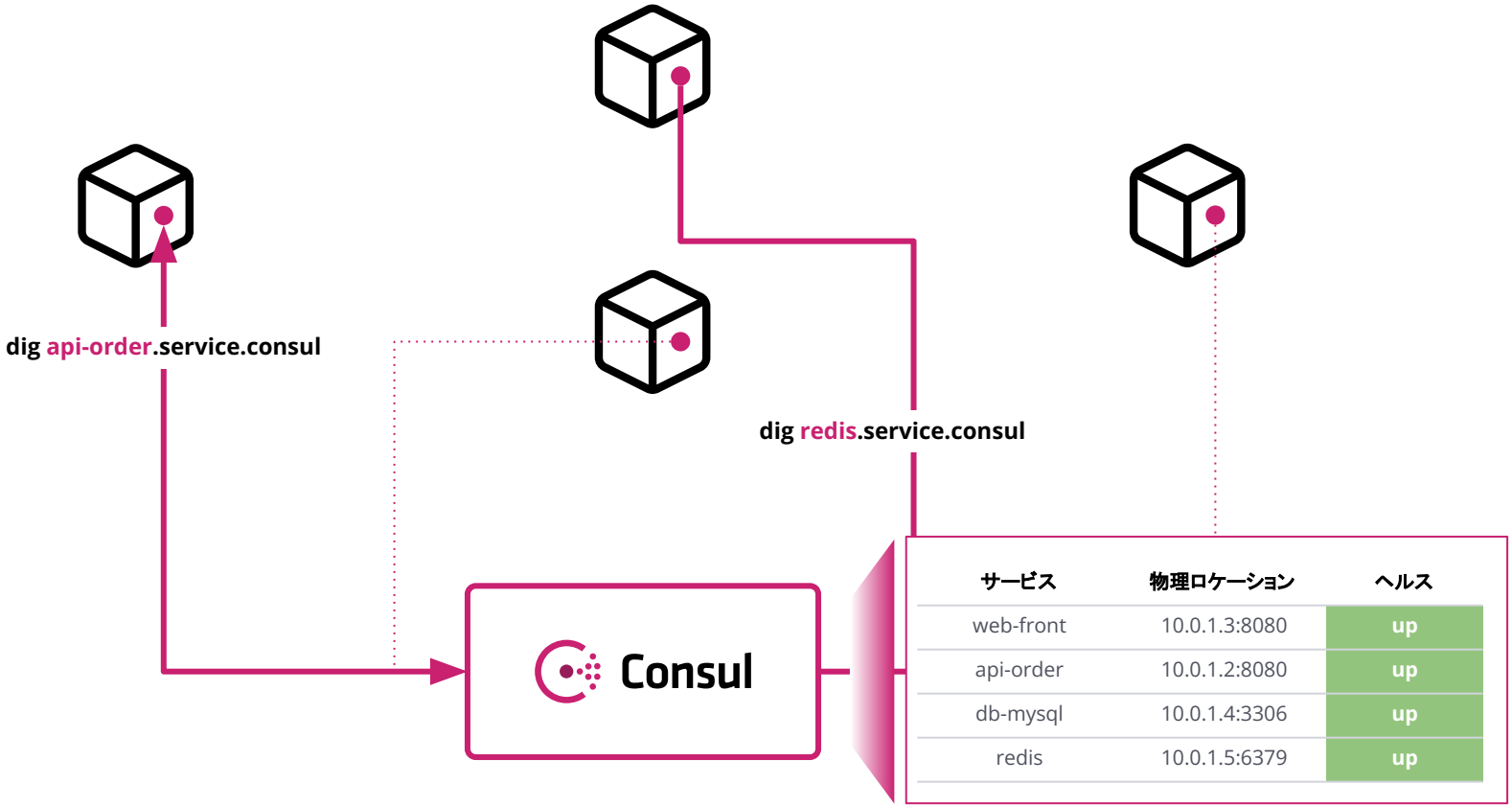
サービスディスカバリ



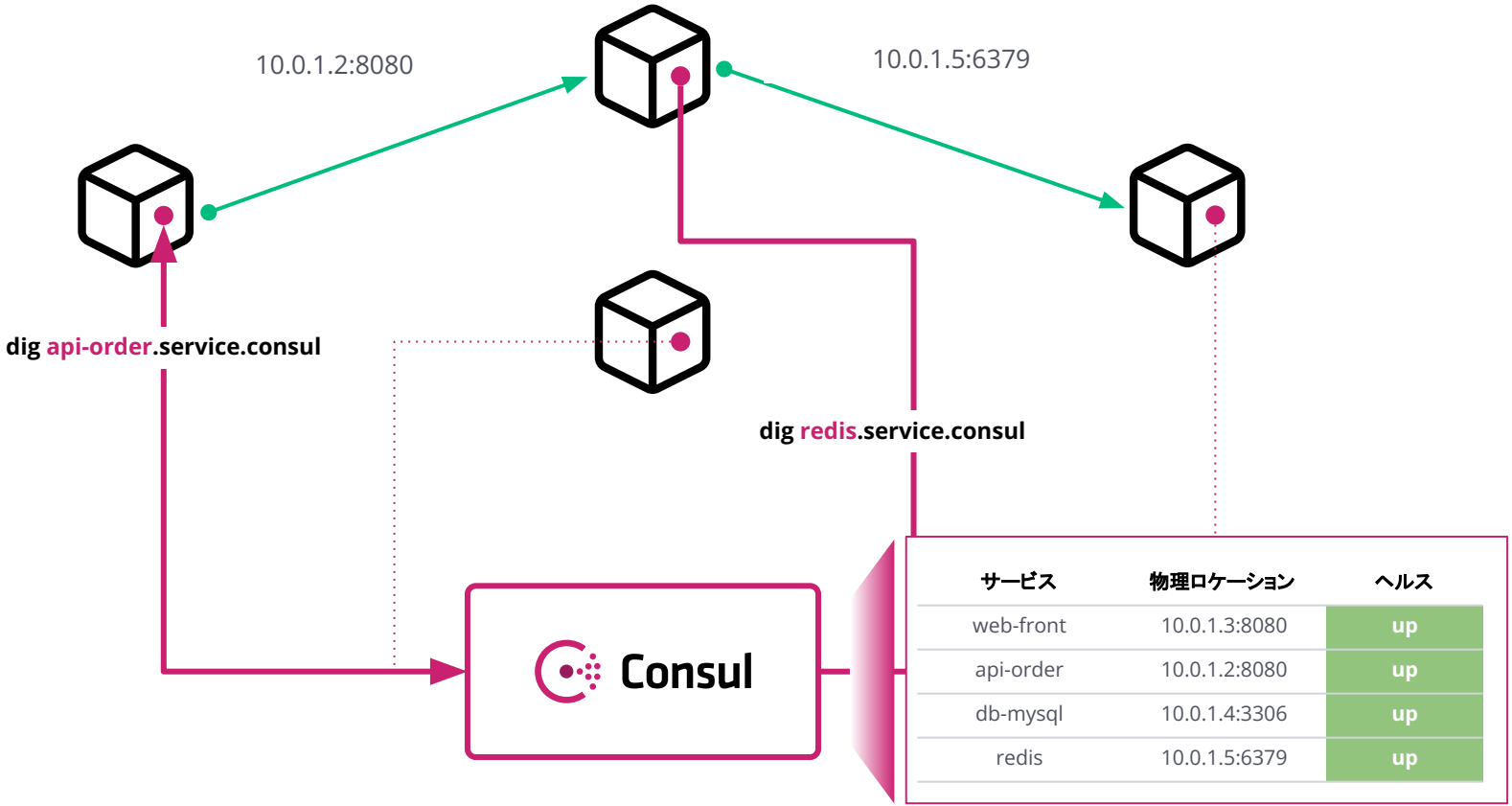
サービスディスカバリ


Consul
サービス
物理ロケーション
ヘルス
web-front
10.0.1.3:8080
up
api-order
10.0.1.2:8080
up
db-mysql
10.0.1.4:3306
up
redis
10.0.1.5:6379
up

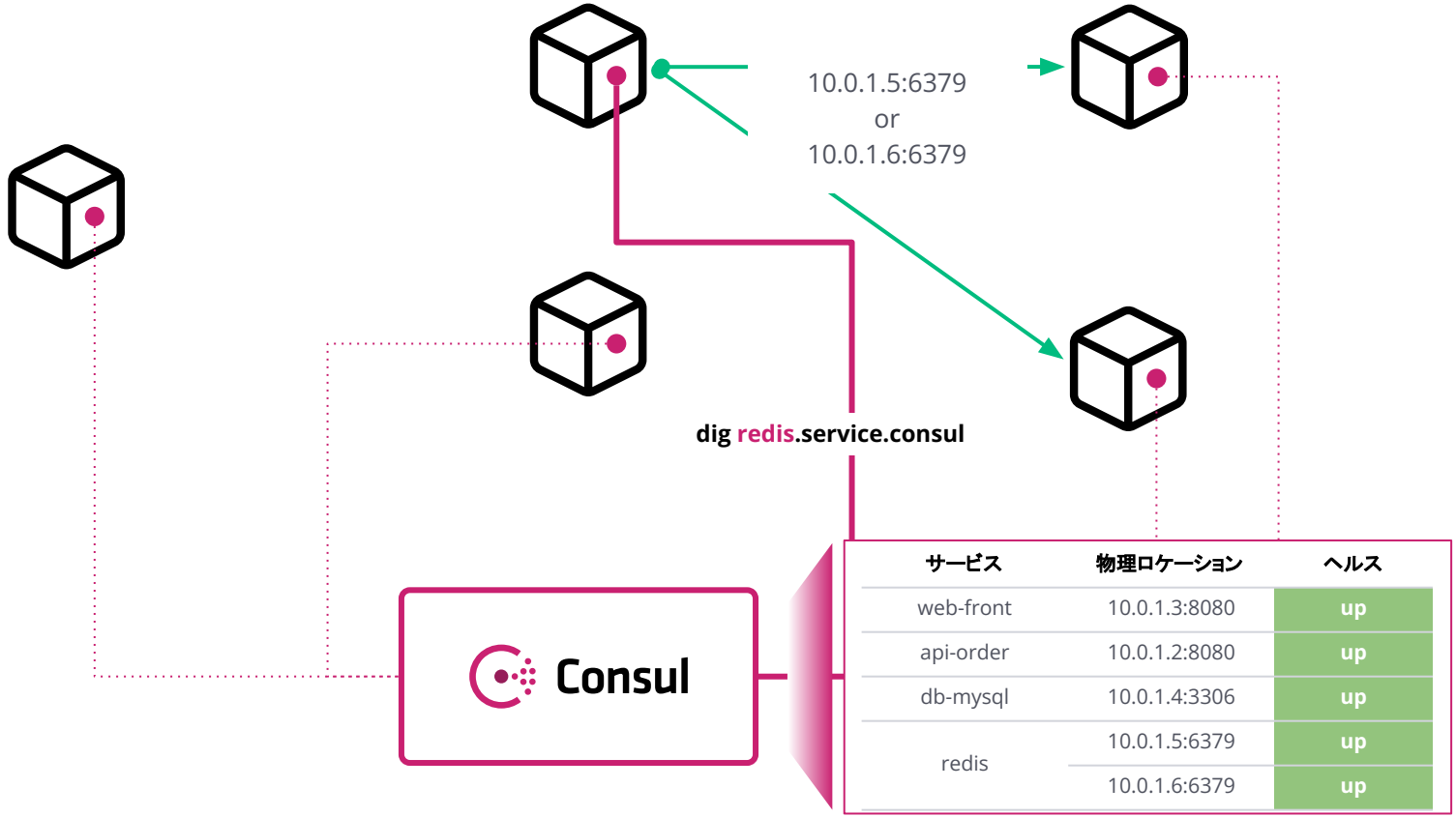
サービスディスカバリ


dig api-order.service.consul
dig redis.service.consul
Consul
サービス
物理ロケーション
ヘルス
web-front
10.0.1.3:8080
up
api-order
10.0.1.2:8080
up
db-mysql
10.0.1.4:3306
up
redis
10.0.1.5:6379
up

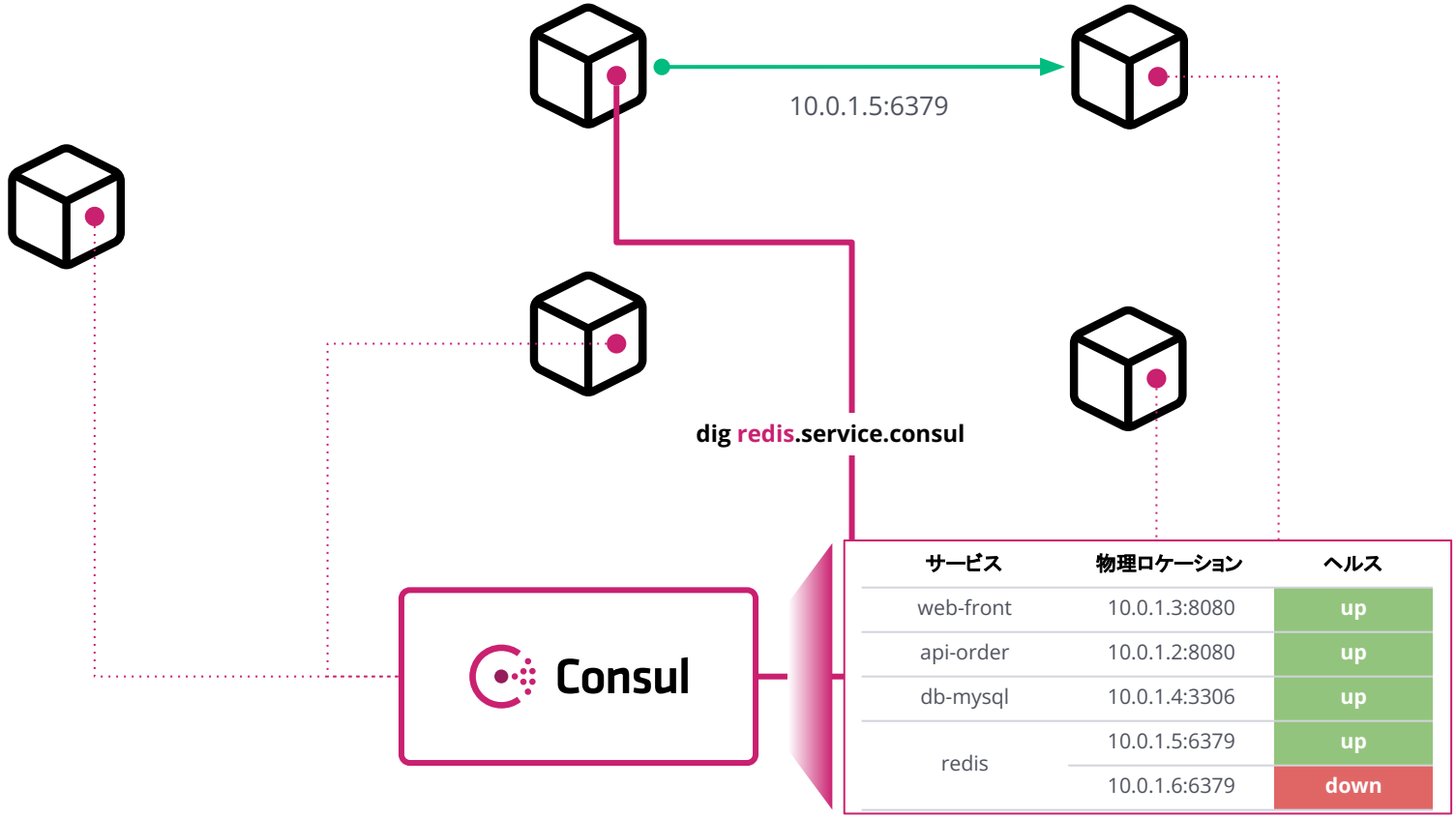
サービスディスカバリ


10.0.1.2:8080
10.0.1.5:6379
dig api-order.service.consul
dig redis.service.consul
Consul
サービス
物理ロケーション
ヘルス
web-front
10.0.1.3:8080
up
api-order
10.0.1.2:8080
up
db-mysql
10.0.1.4:3306
up
redis
10.0.1.5:6379
up

サービスディスカバリ
構成変更時

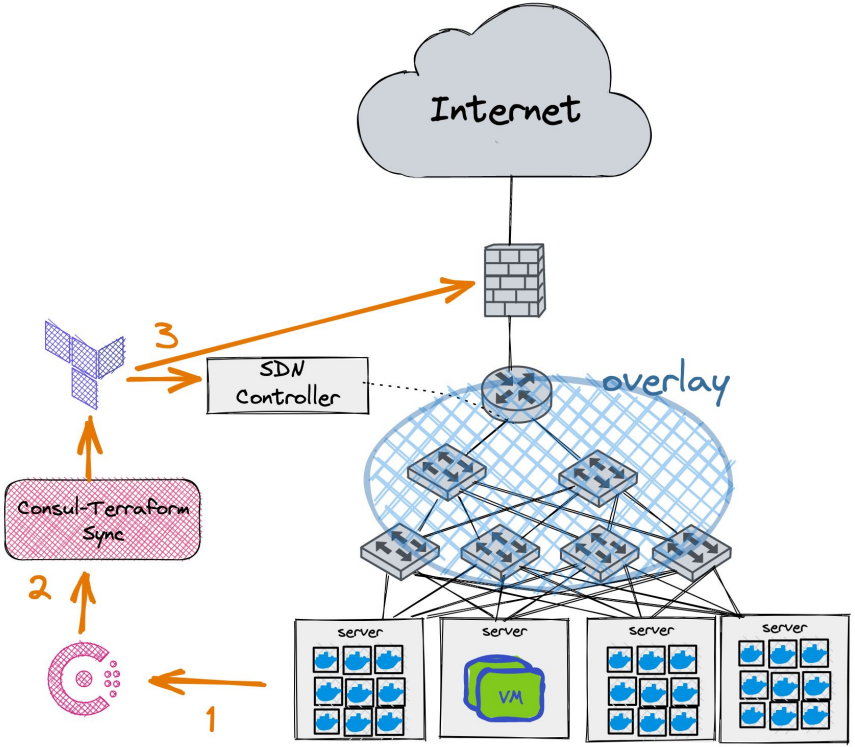


サービスディスカバリ
障害発生時



サービスディスカバリと連携した ネットワークインフラ自動化の例

1. 新しいサービスの登録や変更によるサービスカタログのアップデート
2. Consul Terraform Syncがサービスの情報をConsulのカタログからプル
3. Consul Terraform SyncがTerraformのコードを生成しネットワーク機器の設定変更を適用



Demo

Outcomes

Before
手動・チケットベース

- 複数のエンジニアが巻き込まれる
- 3-5 日かかる作業
- 遊休リソースの無駄

After
サービスディスカバリとの組み合わせ
(Consul-Terraform-Sync)

- 人を介さない運用
- ~150 秒程度のデプロイ時間
- 自動的なクリーンナップ

03

まとめ

自動化のこれから

重要なのは ワークフロー

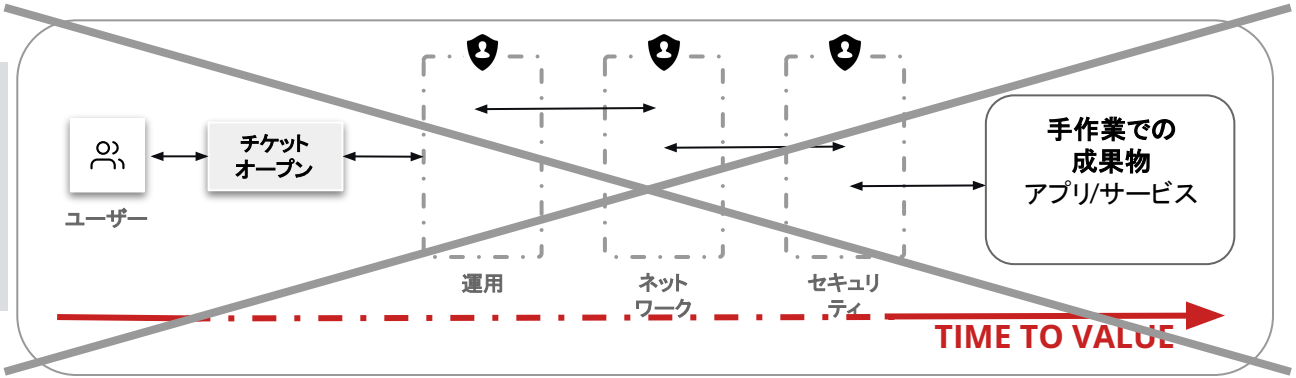
- 最新の要素技術を取り入れるだけでは、効率的な運用に落とせない
- Day0だけでなく、Day1, Day2 を含む 全体のワークフローを意識して自動化のワークフローを考える必要がある
  - 宣言型の自動化
  - Infrastructure as Code

目指すべきワークフローは？

従来のワークフロー:

運用者がインフラに直接設定投入する

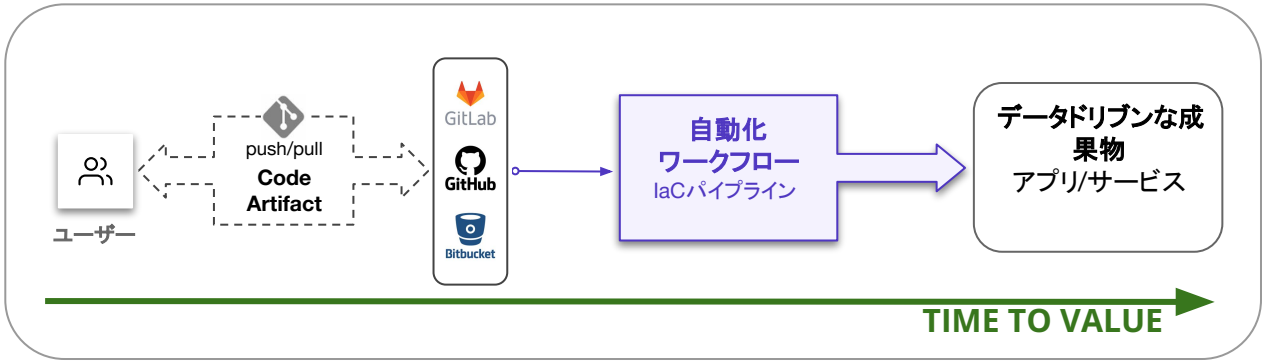
- Manufacturing Snowflakes
- Mutable インフラ
- 変更に時間とコストがかかる



目指すべきワークフロー:

運用者が、IaCパイプラインの権限を持つ

- データドリブン -> IaC -> 自動化
- Immutable インフラ
- 変更は反復的、迅速かつ容易に



まとめ

自動化の方向性

クラウド・SDN・インフラの革新

全てがソフトウェア定義された事により、クラウド・ネットワーク関わらず共通のツールセットで一貫したワークフローを実現

Infrastructure as Code / DevOps / CICD

アプリケーションのモダナイズ

マイクロサービス化、アーキテクチャの近代化によるトラフィックの変化、運用の変化

アプリケーションレイヤネットワーク / サービスディスカバリ、サービスメッシュ

自動化のこれから

アプリケーションの動作、アプリ開発者の要求に合わせた自動化

理想的なワークフローに向けて、全体ワークフローの効率化

Thank You

hello@hashicorp.com
www.hashicorp.com