Amazon EKS と KubeVela で プラットフォームエンジニアリングに入門しよう！

2023/09/28

Kenta Goto

Amazon Web Services Japan G.K.
Solutions Architect

自己紹介

- 名前
  - 後藤 健汰（ Kenta Goto ）
- 所属
  - ISV/SaaS Solutions Architect
- 好きな AWS サービス
  - Amazon EKS
- 趣味
  - サウナ

X@kennygt51

Agenda

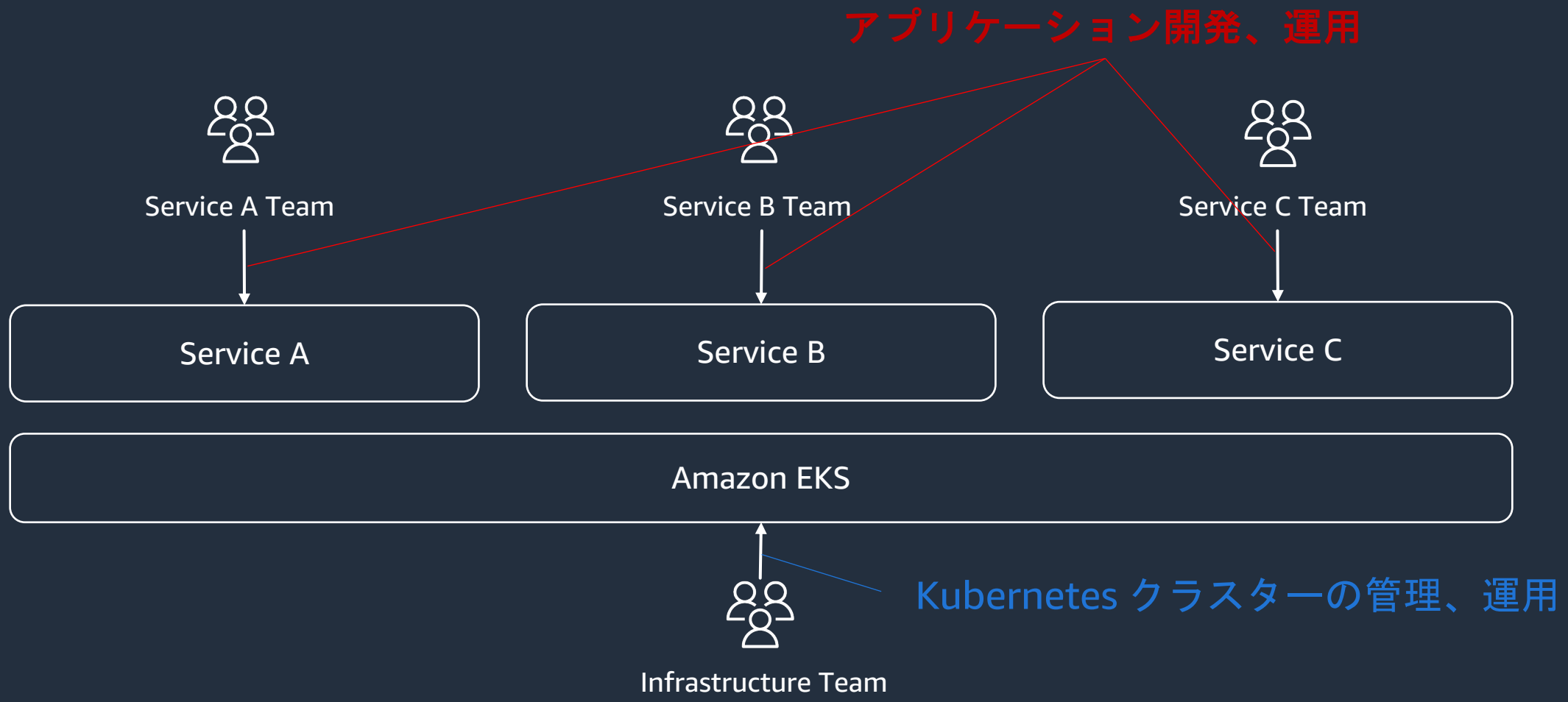
- Amazon EKS 運用における課題
- KubeVela とは
- デモ
- まとめ

Agenda

- Amazon EKS 運用における課題
- KubeVela とは
- デモ
- まとめ

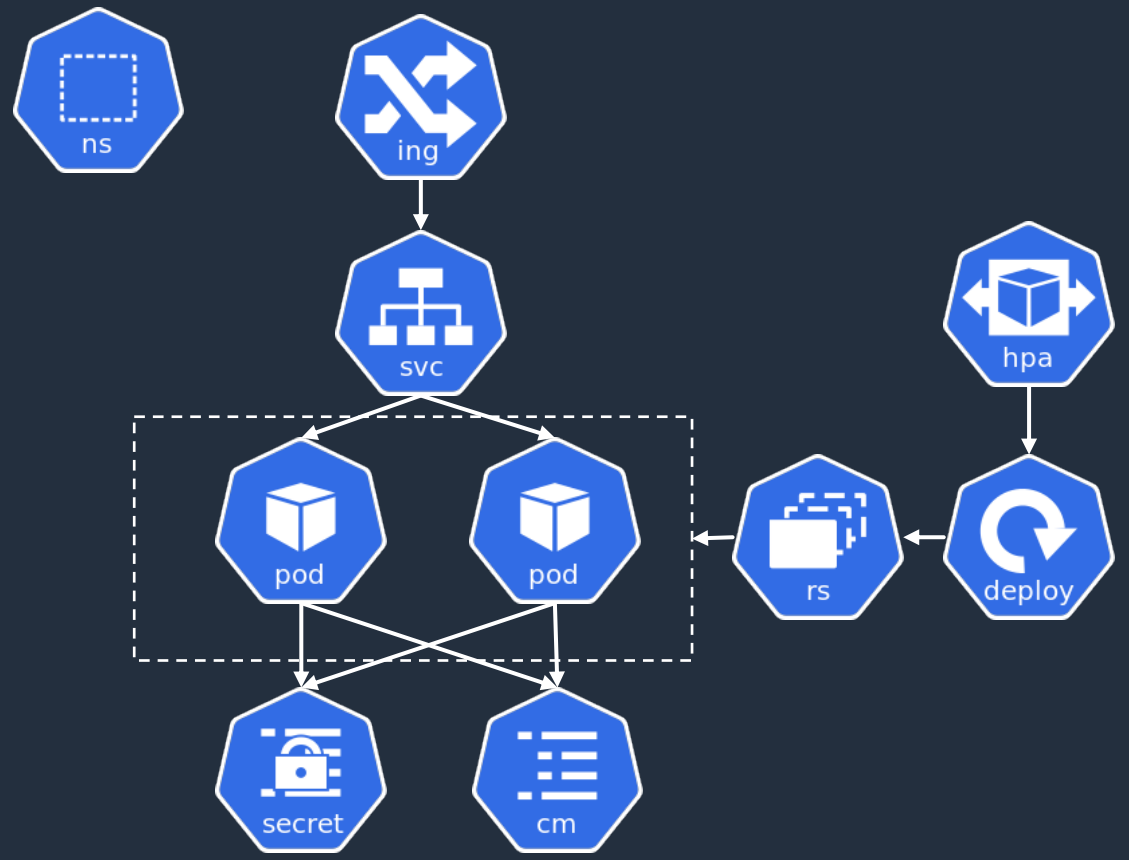
Amazon EKS 運用における課題

Amazon EKS クラスターを活用する開発組織の一例



Amazon EKS 運用における課題

開発チームの認知負荷の拡大

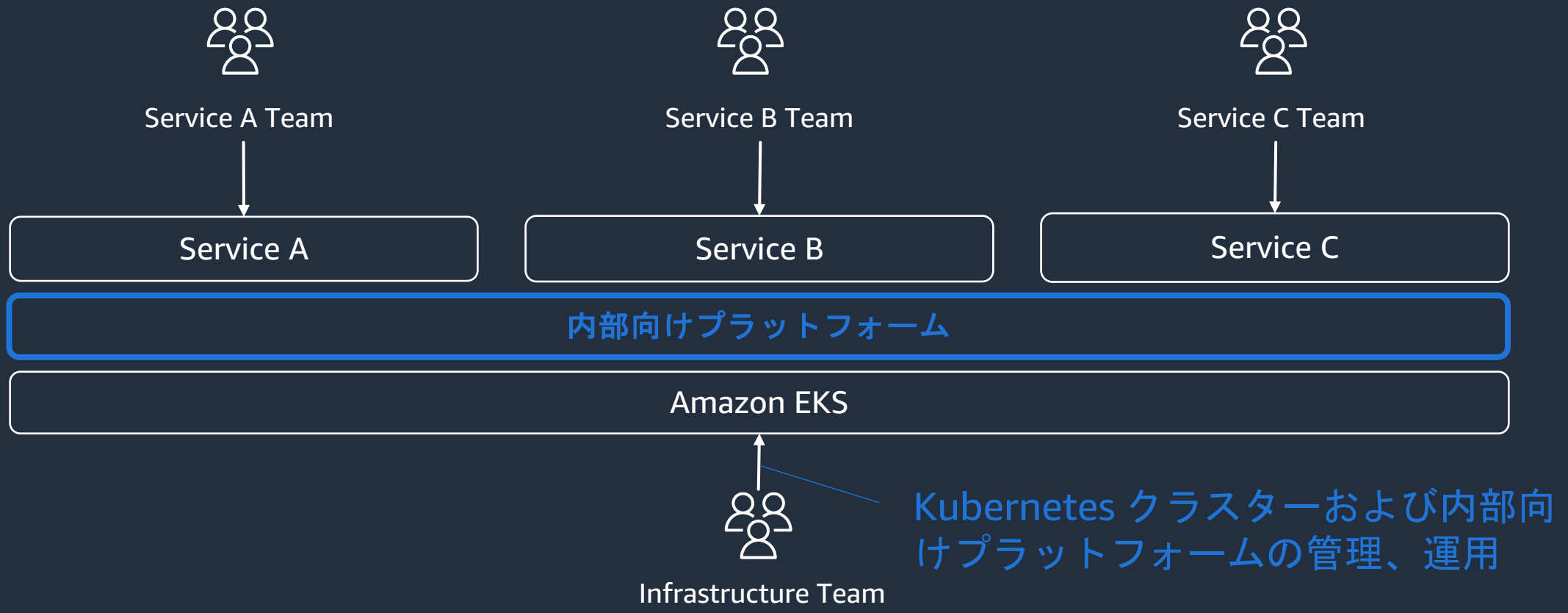


- Kubernetes を活用するには Pod や Service など、Kubernetes の各種 API についての知見が必要
- 開発チームに Kubernetes の知見がない場合、開発者に対するナレッジシェアが課題になることがある
- 運用の中で課題や不明点がある場合、クラスターを管理するインフラチームにエスカレーションすることがある
- 結果としてインフラチームに負荷が集中し、定期的な Kubernetes のアップグレードなど、本来おこなうべき作業に支障をきたす可能性がある

Amazon EKS 運用における課題

独自の抽象化レイヤーによる認知負荷の削減

→ プラットフォームエンジニアリング

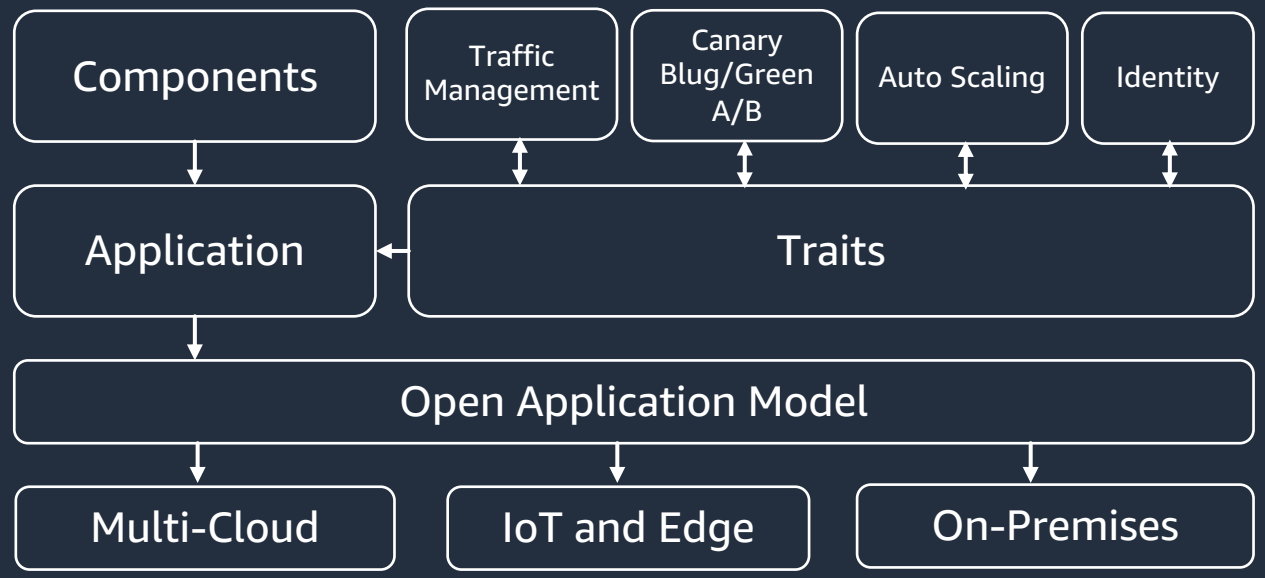


Agenda

- Amazon EKS 運用における課題
- KubeVela とは
- デモ
- まとめ

OAM (Open Application Model)

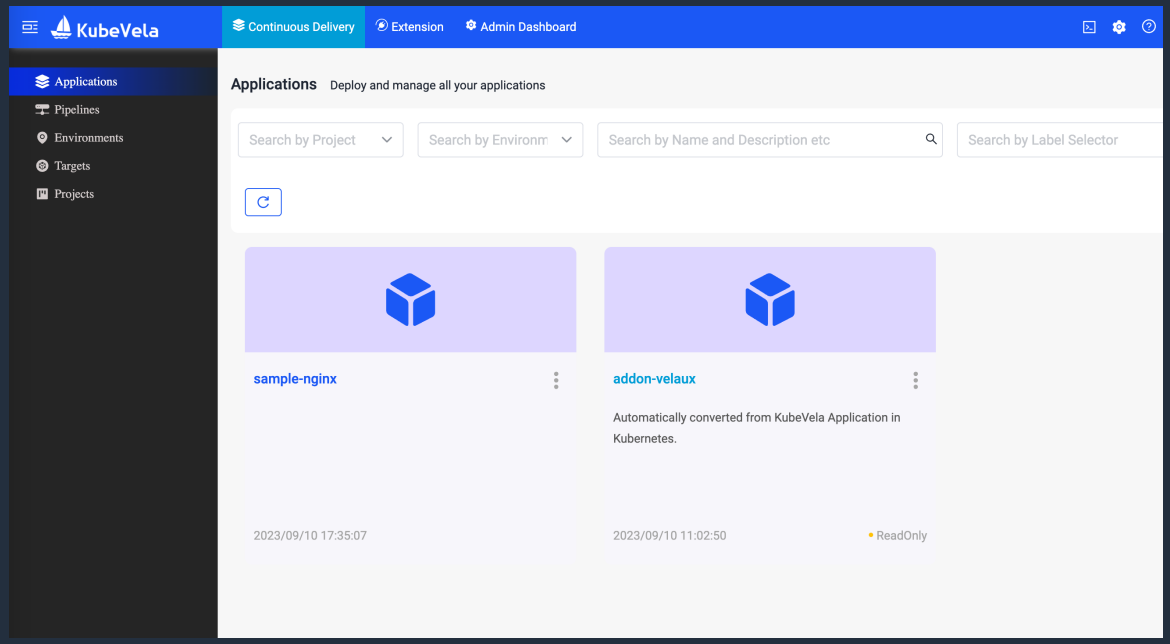
- 「プラットフォームを意識しないアプリケーションを記述する」ための仕様
- アプリケーションの定義を、インフラストラクチャへのデプロイや管理の定義から分離する（関心事の分離）
- OAM はあくまで仕様であり、実装が存在する
- KubeVela は OAM を実装する OSS



KubeVela とは

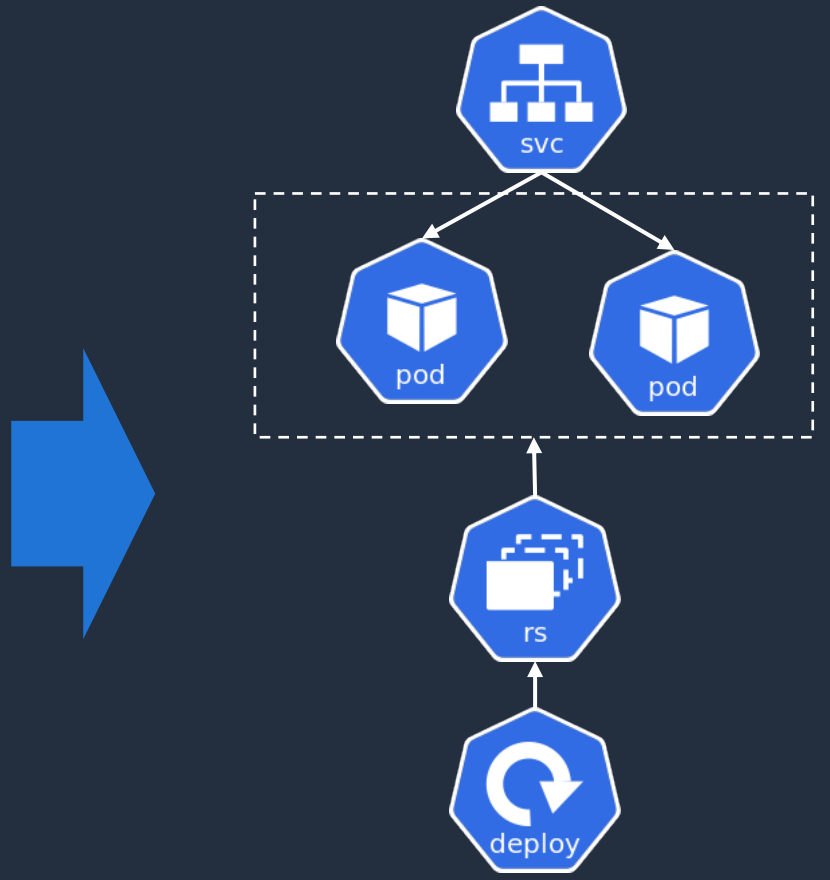
アプリケーションのデプロイや運用をより安全かつ迅速におこなう
モダンソフトウェアデリバリープラットフォーム

- OAM を Kubernetes で実装する
- 開発者は事前定義された「パターン」と「固有の指定」を Application カスタムリソースとして定義する
- VelaUX というアドオンで Web UI を提供
- アプリケーションデプロイのセルフサービス化を実現



KubeVela で実現可能な抽象化

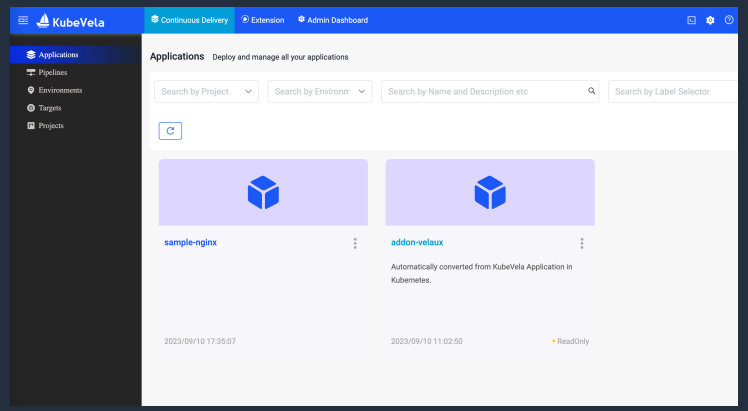
```yaml
apiVersion: core.oam.dev/v1beta1
kind: Application
metadata:
  name: web-app
spec:
  components:
  - name: backend
    type: webservice
    properties:
      image: "public.ecr.aws/aws-containers/ecsdemo-nodejs:latest"
      ports:
      - port: 8080
        expose: true
    traits:
    - type: scaler
      properties:
        replicas: 2
```



KubeVela でのセルフサービス

組織のポリシーやベストプラクティスに沿ったテンプレートを登録

Infrastrucutre Team



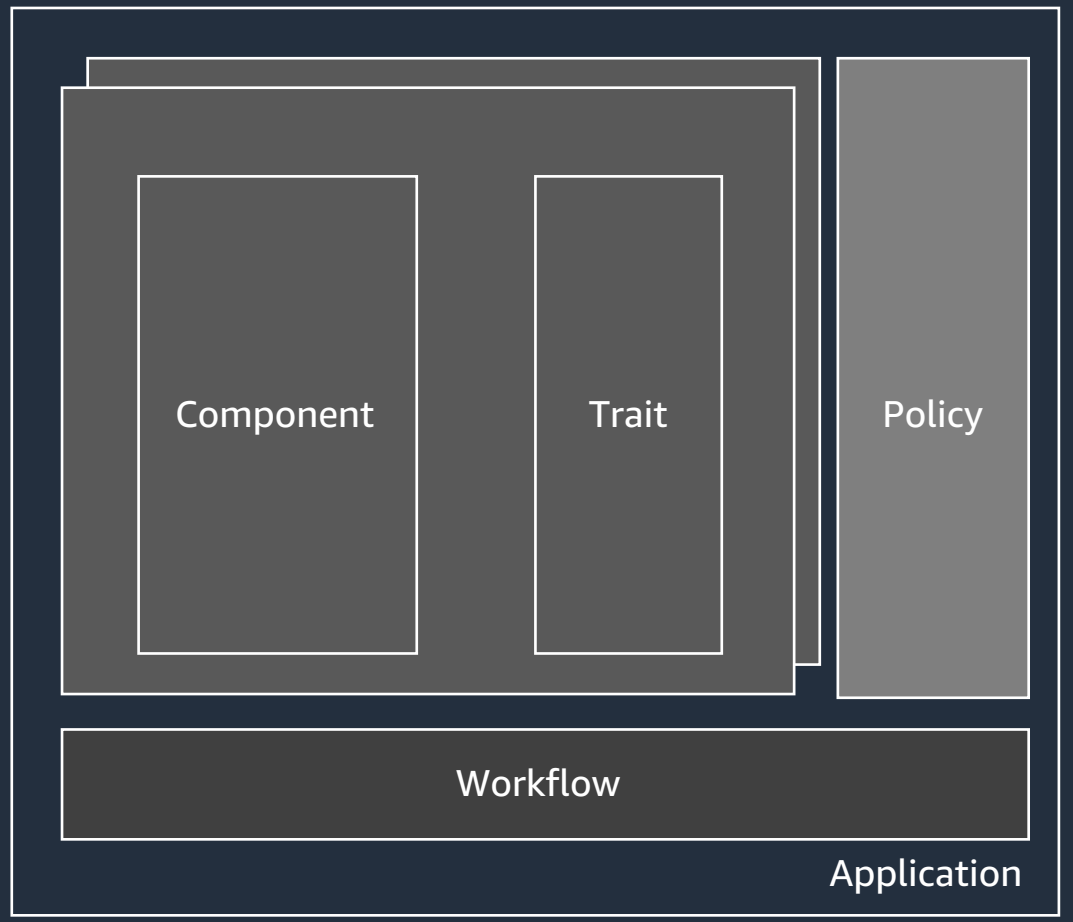
KubeVela

Developer Team

テンプレートを選択、最低限のアプリケーション固有の設定を指定し、セルフサービスでデプロイ

KubeVela Application における重要な概念

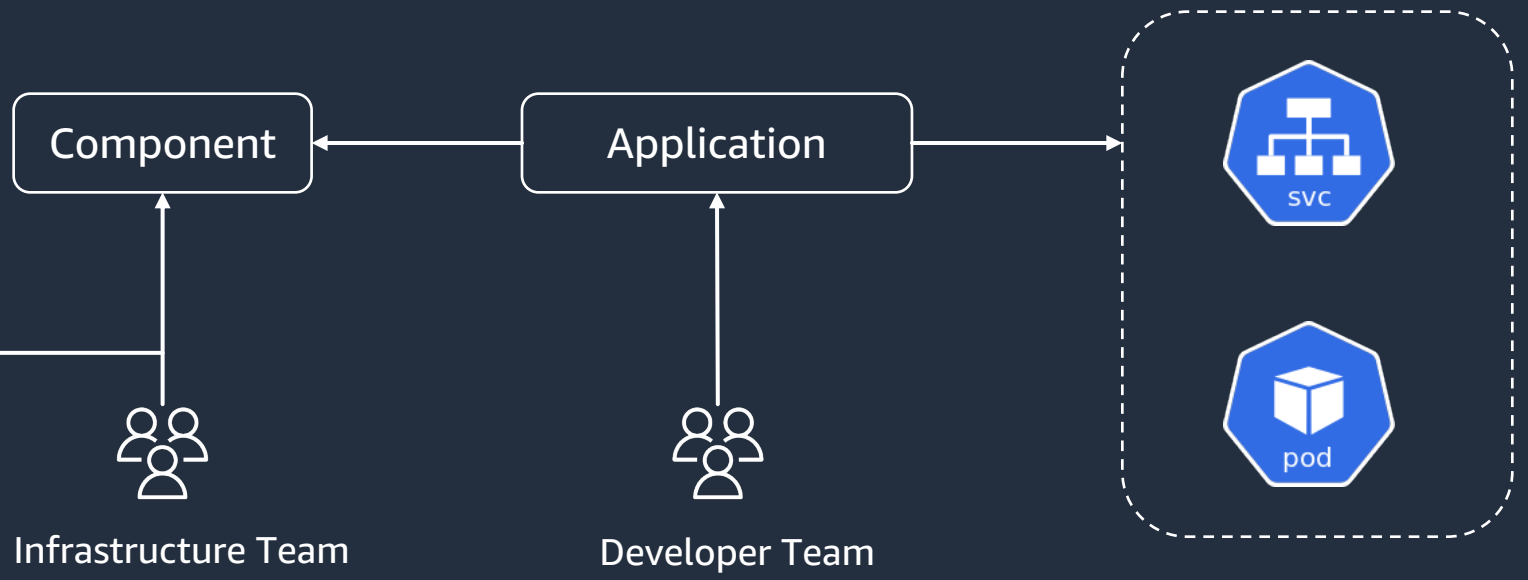
- Component: Application に含まれる個々のコンポーネント
  - バックエンド API アプリケーションや、フロントエンドなど
- Trait: Component に関連付ける追加の情報
  - スケーリング戦略など
- Policy: Application 全体に影響する追加の情報
  - セキュリティやファイヤウォールなど
- Workflow: デプロイプロセスのステップを定義する
  - 手動承認や通知など



Definition

- Built-in に加えて独自の Component や Trait を Definition として作成できる
- 組織のポリシーに従った Definition を作成
- 独自 Definition を作成する際には CUE 言語で定義する

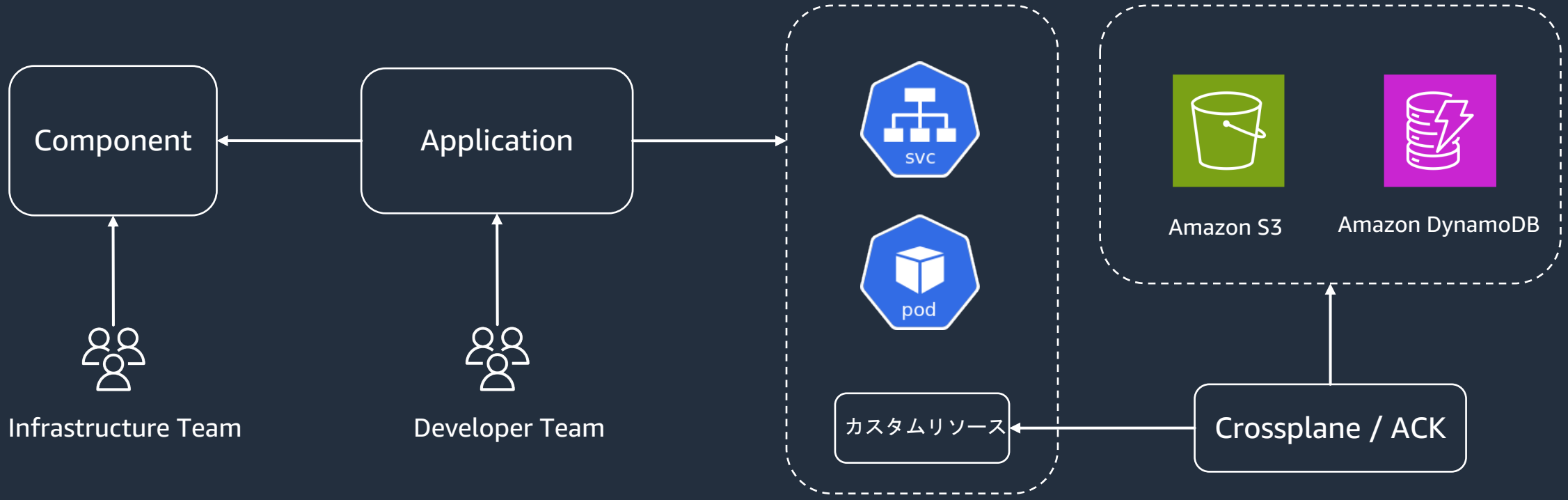
```
"my-comp": {
	alias: ""
	annotations: {}
	attributes: workload: definition: {
		apiVersion: "apps/v1"
		kind:       "Deployment"
	}
	description: "My component."
	labels: {}
	type: "component"
}

template: {
	output: {
		apiVersion: "apps/v1"
		kind:       "Deployment"
		metadata: name: "hello-world"
		spec: {
			replicas: 1
			selector: matchLabels: "app.kubernetes.io/name": "hello-world"
			template: {
				metadata: labels: "app.kubernetes.io/name": "hello-world"
				spec: containers: [{
					image: "somefive/hello-world"
					name:  "hello-world"
					ports: [{
						containerPort: 80
						name:          "http"
						protocol:      "TCP"
```



CUE 言語で作成した Definition

Crossplane や ACK との連携

- Crossplane や ACK で扱うカスタムリソースを含む Definition から Component を作成することで、Crossplane や ACK と連携



Agenda

- Amazon EKS 運用における課題
- KubeVela とは
- デモ
- まとめ

Agenda

- Amazon EKS 運用における課題
- KubeVela とは
- デモ
- まとめ

本セッションのまとめ

- Amazon EKS の運用において、開発チームへのナレッジシェアが課題になることがある

- プラットフォームエンジニアリングのアプローチを用いた、抽象化のレイヤーを用意することが解決策になりうる

- KubeVela は OAM の実装で、複雑な Kubernetes リソースを抽象化しセルフサービス化を実現する

Thank you!