最高の “Kubernetes on AWS” を実現するために

複数クラスタ考察編

Tori Hara / toricls
Sr. Product Developer Advocate
Amazon Web Services

aws SUMMIT ONLINE

Tori Hara / toricls

Sr. Product Developer Advocate
Containers Product, Amazon Web Services

ERP パッケージベンダー R&D チーム SDE
➡ UI 自動テスト SaaS
➡ クラウド利用の SI + MSP にて、コンテナやサーバーレスによる設計・開発・運用
Web 技術利用のゲームやビジネスアプリケーション開発、ML/DL 環境構築運用など
➡ Sr. Containers Specialist SA, AWS Japan
➡ 現ロール

AWS Fargate と AWS Lambda が好きです

想定聴講者とゴール

想定聴講者

- Amazon Elastic Kubernetes Service (Amazon EKS)/Kubernetes を使っているが自信を持って使えていない
- Kubernetes クラスタを作ってみたが長期運用を考えた構成になっていない気がしている
- アップグレードできずに放置している Kubernetes クラスタがあり、先行きに不安がある

ゴール

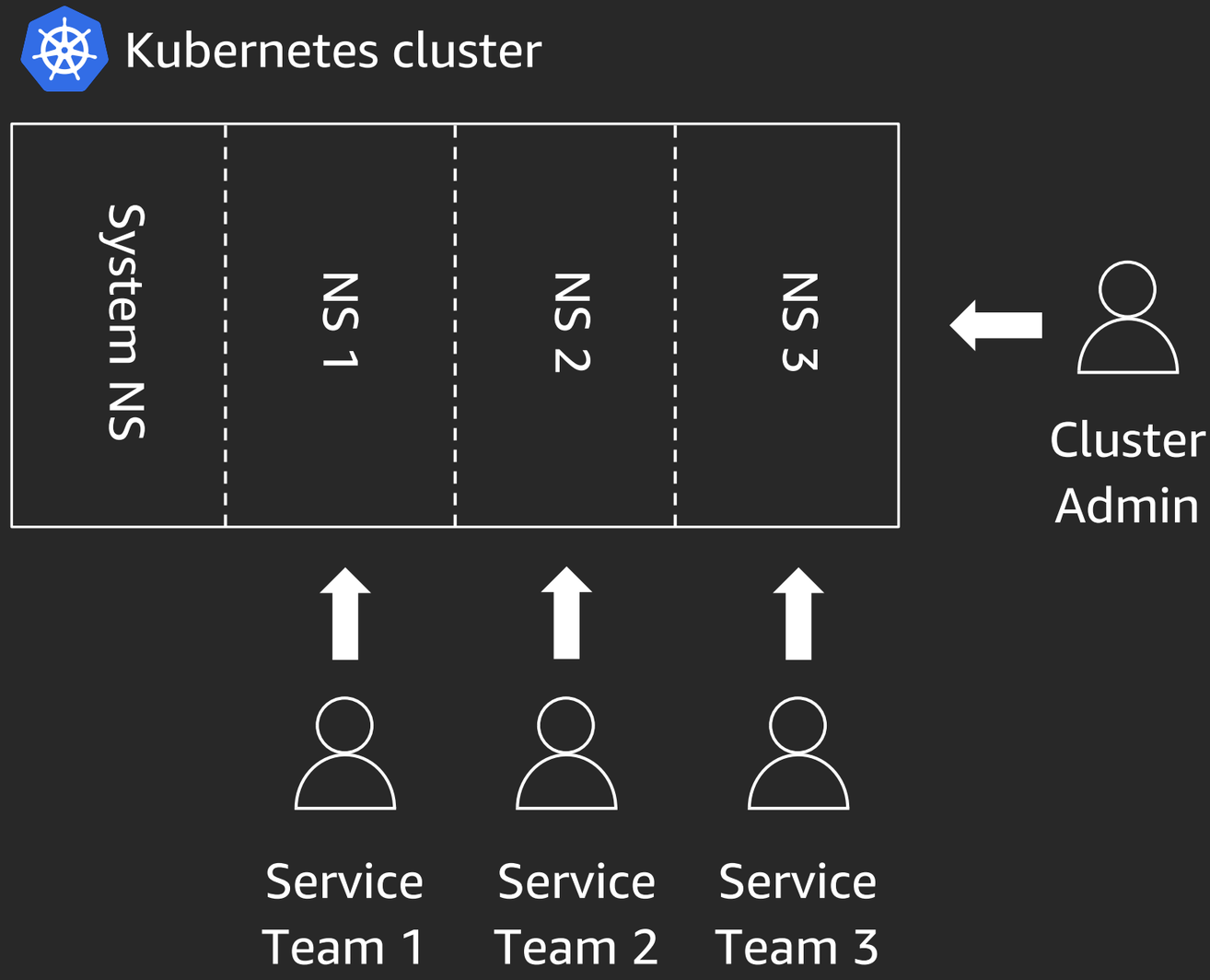
- クラスタ管理者の観点から、状況変化に対応しやすい Kubernetes クラスタについて学び、AWS 上でサステナブルな Kubernetes クラスタを構築する上で押さえておくべきポイントを理解する

※ 本セッションの内容は 2020/07 時点の公開情報に基づいています. 配信時の最新情報とは異なる場合がありますのでご注意ください.

思考停止のマルチテナントを見直そう

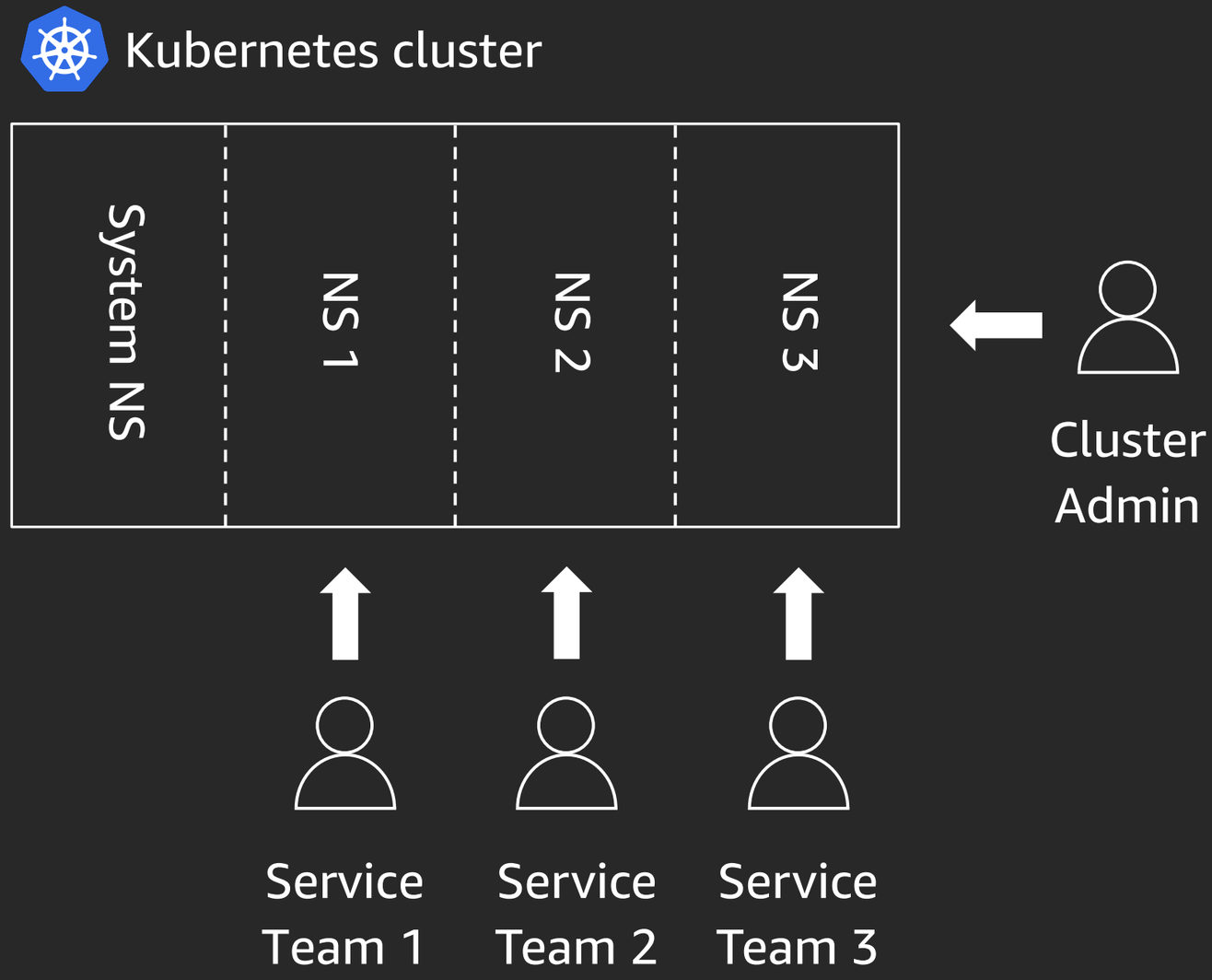
aws SUMMIT ONLINE

マルチテナント Kubernetes クラスタ



- Namespaces や RBAC を活用
- 複数サービスを単一の Kubernetes クラスタで実行
- 各チームが独自のタイミングでデプロイ
- Kubernetes クラスタの代表的利用パターンの1つ
- 意図的に違うパターンにしない限り、この形になりやすい

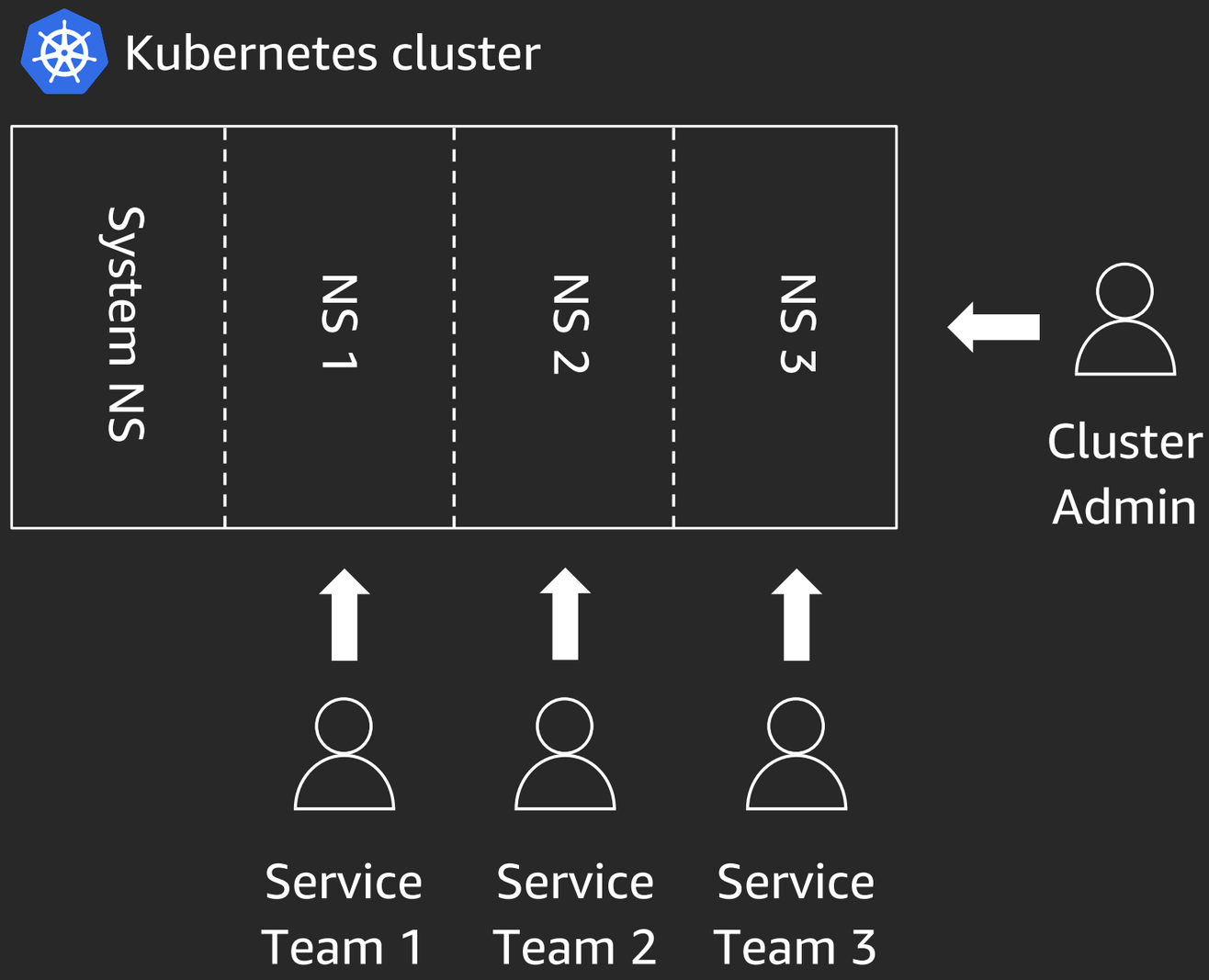
マルチテナント Kubernetes クラスタ



マルチテナント単一クラスタの利点

- 運用クラスタ数が最小限
- クラスタ配下ワーカーノード群のリソース使用効率
- Kubernetes クラスタに閉じた世界での設計が可能
- エコシステム OSS を素直に使える
- オンプレミスでもよく見られる構成であり事例も多い

マルチテナント Kubernetes クラスタ



マルチテナント単一クラスタの課題

- クラスタ責務肥大化
- クラスタ障害の影響範囲
- ステークホルダが多い
- **上記への考慮がなくとも それなりに動くものを容易に 作れてしまう**
  - (Kubernetes の魅力の裏返しとも 言える)

マルチテナント Kubernetes クラスタ

思考停止マルチテナントクラスタとそのリスク

- クラスタを増やすことへの心理的障壁
  - e.g. 作成時は IaC で始めたが、その後の運用・更新作業は手作業
  - e.g. 新チームのオンボード時はクラスタ管理チームに連絡して新しい Namespace 作成や RBAC 設定が完了するのを待つ
- クラスタ運用開始後のサービスや組織の拡大によるクラスタ肥大化
  - e.g. 新サービス・新チームは待ってくれない、しょうがないのでひとまず既存クラスタに同居
- 秘伝のタレ継ぎ足しによるクラスタ構築・運用
  - e.g. クラスタがブラックボックス化し、再現不能
- 仮想マシン時代の「ペット化」と同じことが Kubernetes クラスタでも容易に発生する
  - 横にクラスタを増やせばサービスや組織のスケールに対応できる構成を目指したい

Kubernetes クラスタのペット化を避ける

基本に立ち返る

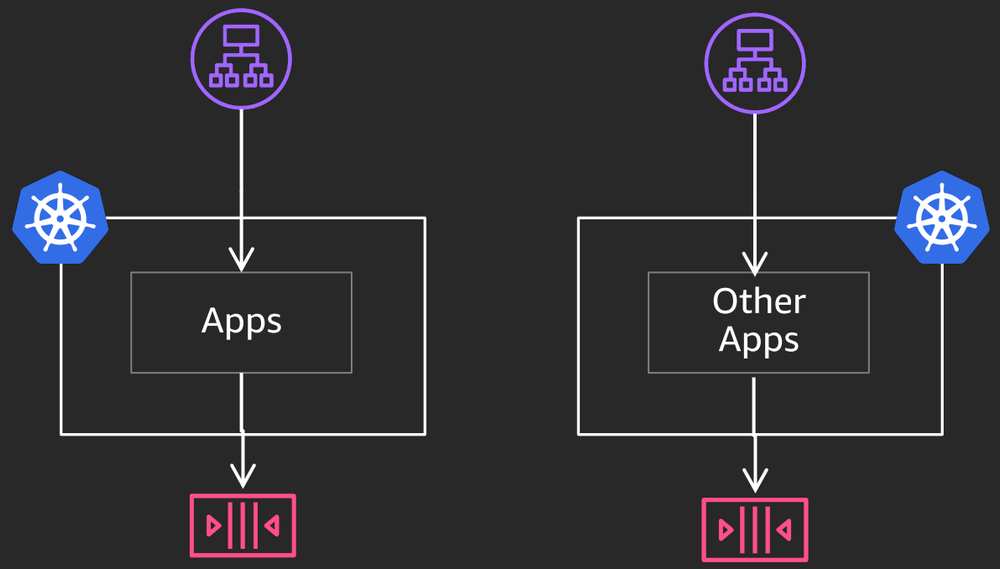
- Infrastructure as Code / Immutable Infrastructure
  - クラスタ上のアプリケーションについては Kubernetes がいい感じにやってくれる
  - 作って動いたから OK ではなく、更新作業を含めた将来のクラスタ運用を考慮する
  - 手動オペレーションでのクラスタ構築・運用を避け、可能な限り自動化する
  - いつでも削除でき、いつでも再作成できる Kubernetes クラスタが実現できれば最も安全に運用できる
- クラスタ移行(切り出し)や分割といった新規クラスタの追加可能性を考慮して構築方法を検討する

Beyond single Kubernetes cluster

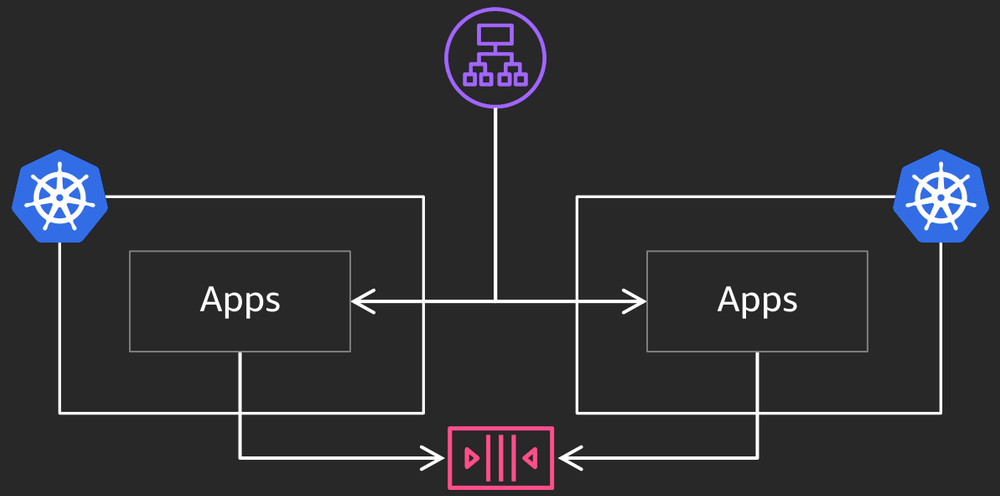
aws SUMMIT ONLINE

複数クラスタの代表的構成パターン

Beyond single Kubernetes cluster



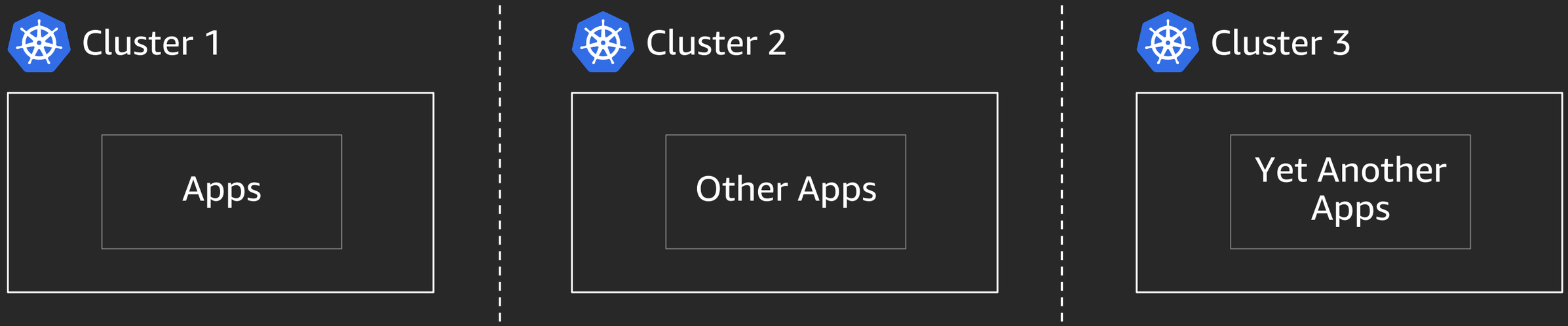
1. AWS リソースを共有しないクラスタ群



2. AWS リソースを共有するクラスタ群

1. AWS リソースを共有しないクラスタ群

よくあるユースケース: 異なる事業ライン別、環境別 (dev/staging/production)、...



Pros

- クラスタ障害が互いに影響を与えない
- クラスタごとに異なるライフサイクルを持てる

Cons

- **クラスタ運用コストの増大**

どのようなシチュエーションで分割するか

あらかじめ想定・意思決定しておくことが重要

1. AWS リソースを共有しないクラスタ群

クラスタ分割の判断基準を持つ

独立した (shared nothing な) **クラスタを持つべき**	**クラスタ分割を検討すべき**
• 環境別 (dev, staging, prod) • 異なる事業ライン (1つの事業内のサービス群はクラスタを共有しうる) • 複数サービスが利用する共通サービス (e.g. 独自 ID 基盤) • 固有の SLA があるサービス • R&D 用クラスタ	• SLO があるサービスとないサービス、あるいは設定した SLO に大きな隔たりがあるサービス群 • マイクロサービスの各サービスチームに強い自律性を持たせたい • Kubernetes クラスタバージョンを上げるための各チームとの調整作業に数週間要してしまっている

1. AWS リソースを共有しないクラスタ群

AWS アカウントはどうする？

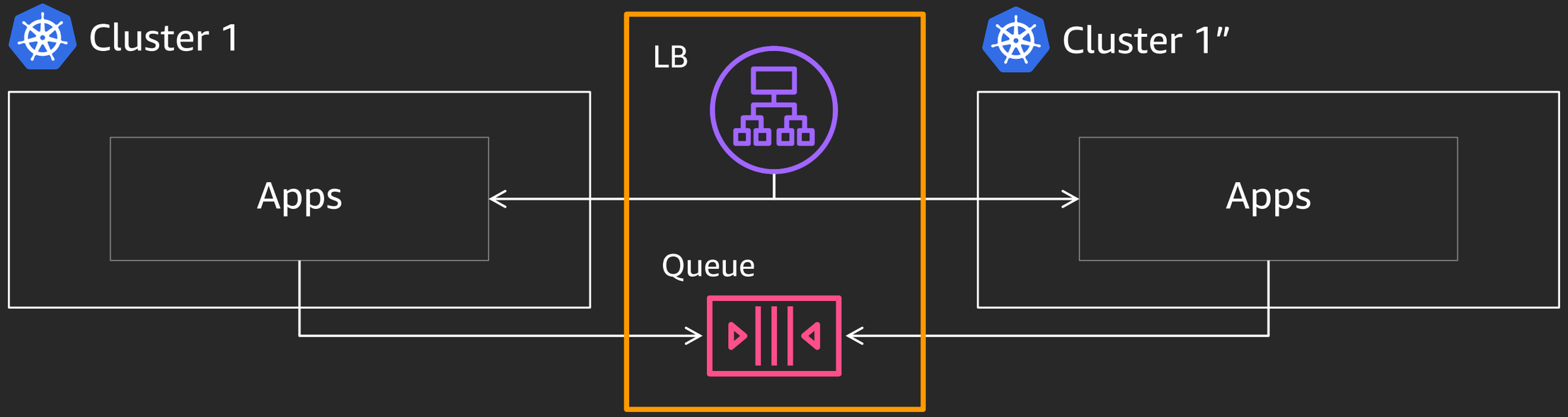
- AWS リソースを共有しないクラスタ群は作ること自体は難しくない(IaC)
- 可能な限り AWS アカウントも分けることを推奨
  - セキュリティ、アカウントレベルのサービス上限値(クォータ)、…
- AWS Organizations や AWS SSO の併用によりマルチ AWS アカウントの管理コストを低減
  - e.g. SAML 連携による AWS IAM ユーザへの Kubernetes API サーバアクセス権付与 [1]
- Kubernetes クラスタコントロールプレーンの Audit ログ [2] を各 AWS アカウントの CloudWatch Logs からクラスタ群管理用 AWS アカウントに集約し、クラスタ群全体に渡る API 呼び出しの監査を実施する (CloudTrail でやってきたことと同様に)

[1] Okta の利用例: https://aws.amazon.com/jp/blogs/news/manage-amazon-eks-with-okta-sso/

[2] Amazon EKS コントロールプレーンのログ記録: https://docs.aws.amazon.com/ja_jp/eks/latest/userguide/control-plane-logs.html

2. AWS リソースを共有するクラスタ群

よくあるユースケース: クラスタと AWS リソースの疎結合性実現、クラスタレベルの B/G デプロイ



Pros

- クラスタと AWS リソースを疎結合にできる
- クラスタ責務範囲を狭くできる

Cons

- **構築時に考えることが増える**
  - (ビルディングブロック的な難しさ)

2. AWS リソースを共有するクラスタ群

構築時に考慮する必要がある事柄の例 – Application Load Balancer (ALB)

課題

- ALB の作成・リスナの作成・ターゲットグループの作成・ターゲットグループへの登録の全てを ALB Ingress Controller が実施するため、別 Kubernetes クラスタ上で動くアプリケーションから同じ ALB を利用することができない
- 1つの ALB を複数の Kubernetes アプリケーションで共有したい、EC2 アプリケーションにもルーティングしたい
- ALB の作成に利用した Kubernetes クラスタを削除すると ALB も一緒に削除されてしまう

2. AWS リソースを共有するクラスタ群

構築時に考慮する必要がある事柄の例 – Application Load Balancer (ALB)

ワークアラウンド

- ALB・リスナ・ターゲットグループの作成には CloudFormation や Terraform などを用いる
- Kubernetes Service を type: NodePort でポート番号固定で用意
- EC2 Auto Scaling グループを ALB ターゲットグループと関連付けて作成し [1]、EC2 のスケールアウトに応じて自動的に EC2 が ALB ターゲットグループに登録されるようにする ※現時点ではセルフマネージド型ワーカーノードのみ対応
- ALB に加えて Nginx Ingress Controller を利用することで、いったん ALB からトラフィックを Kubernetes クラスタ内に通した上で Nginx Ingress を利用して複数の Kubernetes アプリケーションにルーティング可能

[1] : https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/create-asg-launch-template.html

2. AWS リソースを共有するクラスタ群

構築時に考慮する必要がある事柄の例 – Application Load Balancer (ALB)

その他の関連情報

- ALB Ingress Controller の “IngressGroup” [1] を利用することで1つの ALB から複数の Kubernetes アプリケーションにルーティング可能
- ALB Ingress Controller が既存ターゲットグループへの登録だけを受け持てるよう機能追加を予定
- 同様の機能を持ったサードパーティの OSS ツールも存在はしている
- ALB はあくまでも ALB Ingress Controller で作成し、Route 53 や AWS Global Accelerator でクラスタにまたがる加重ルーティングを実施することも1つの選択肢

[1] : (未正式リリース) https://github.com/kubernetes-sigs/aws-alb-ingress-controller/releases/tag/v1.2.0-alpha.1

2. AWS リソースを共有するクラスタ群

構築時に考慮する必要がある事柄の例 – Virtual Private Cloud (VPC)

課題

- 新バージョンの Kubernetes クラスタを横に作成し、RDBMS は新旧クラスタから同じものを参照する構成を取ろうとしたがサブネットの IP アドレスが足りなくなってしまい、新クラスタへのアプリケーションデプロイができない
- VPC やサブネットの再作成は避けたい

2. AWS リソースを共有するクラスタ群

構築時に考慮する必要がある事柄の例 – Virtual Private Cloud (VPC)

ワークアラウンド

- CNI カスタムネットワーキング [1] を利用し、Pod の IP アドレスをノードとは異なる CIDR レンジから割り当てる ※現時点ではセルフマネージド型ワーカーノードのみ対応

本質的な解決策

- クラスタ作成時点で少なくとも2つのクラスタを実行できるように VPC やサブネットをサイジングする
  - ロードバランサや、その他 VPC IP アドレスも消費するリソースについても意識しておく

[1] : https://docs.aws.amazon.com/ja_jp/eks/latest/userguide/cni-custom-network.html

まとめ

aws SUMMIT ONLINE

最高の “Kubernetes on AWS” を実現するために

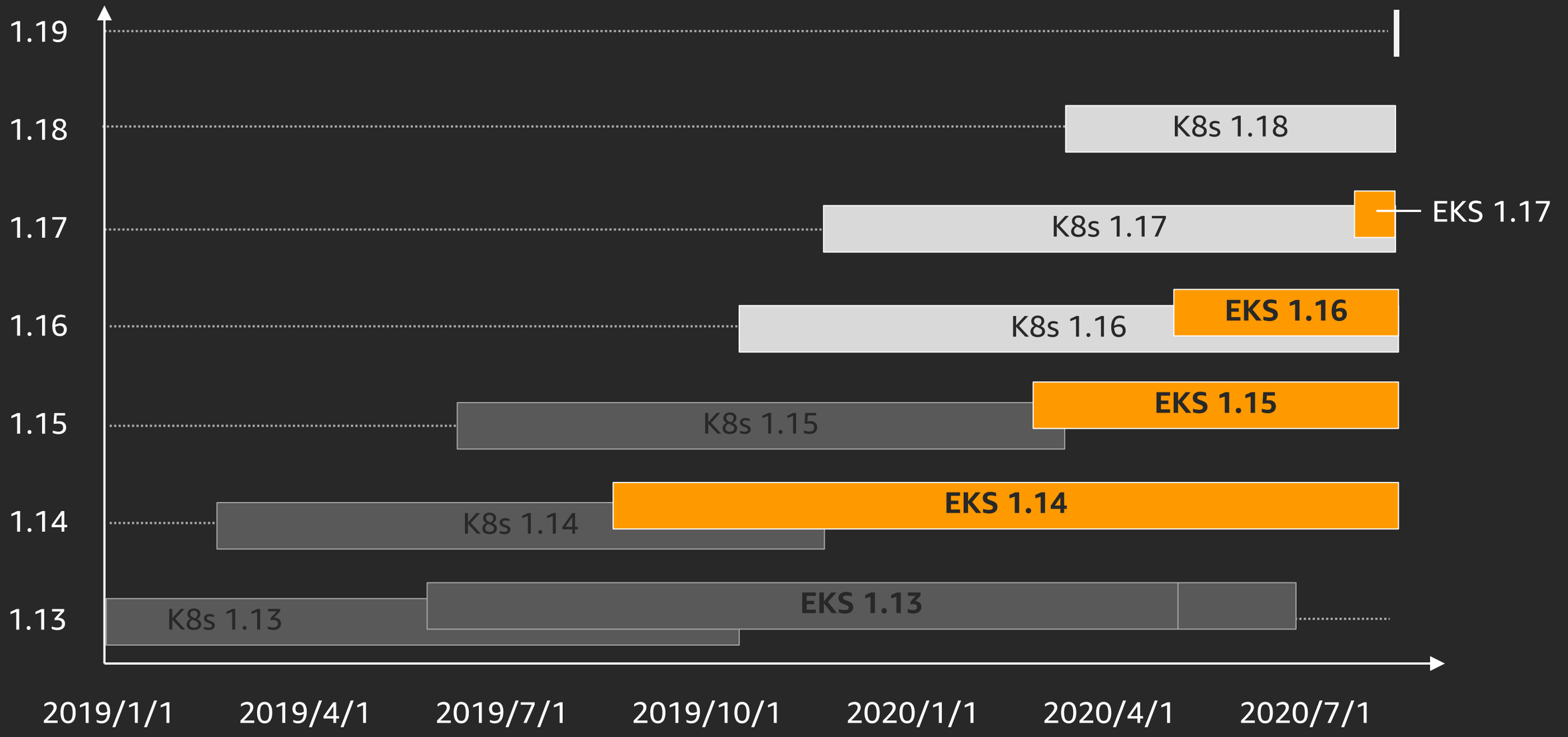
複数クラスタ考察編

- Kubernetes は単一クラスタだけで利用すると使いやすい
  - **思考停止のマルチテナント単一クラスタには注意が必要**
  - なし崩し的なクラスタ肥大化、複雑化、メンテナンス性悪化につながる
  - チーム間の依存関係から「動きにくい」Kubernetes クラスタになりやすい
- クラスタのペット化を避ける
  - IaC, Immutable Infrastructure
  - **肥大化してからクラスタ分割について考えるのではなく、事前検討**
- クラスタ管理者の視点で、どのようなシチュエーションでクラスタを分割すべきか、あるいは分割を検討すべきか事前に把握しておく
- AWS リソースを Kubernetes クラスタから疎結合に作成する方法は構築に手間はかかるが、Kubernetes クラスタそのものの運用容易性が高い

Appendix

※ 2020/07/28 時点

Kubernetes & Amazon EKS サポートバージョン



Amazon EKS on AWS Fargate

Bring existing pods

You don’t need to change your existing pods.

Fargate works with existing workflows and services that run on Kubernetes.

Production ready

Launch pods quickly. Easily run pods across multiple AZs for high availability.

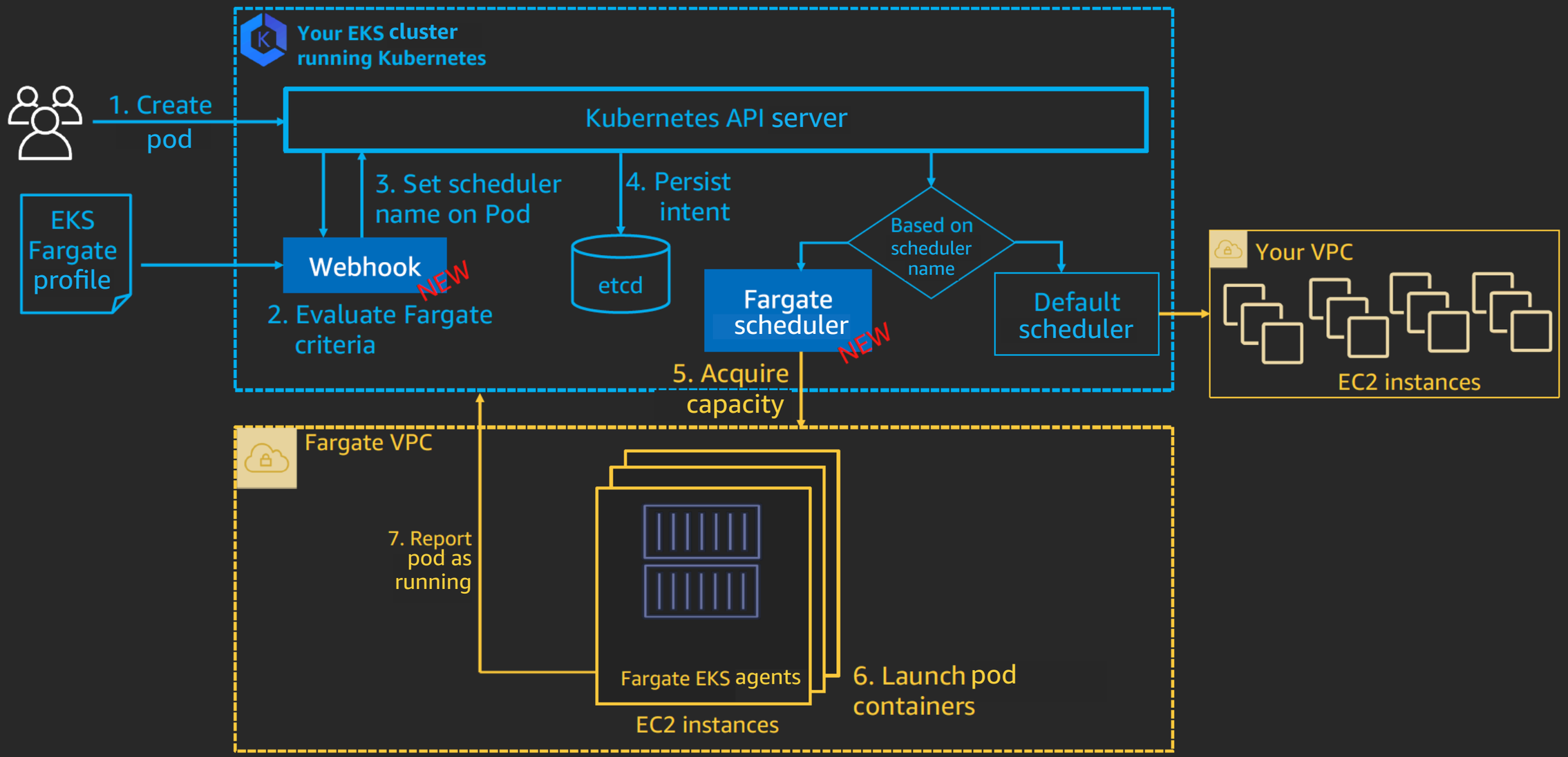
Each pod runs in an isolated compute environment.

Rightsized and integrated

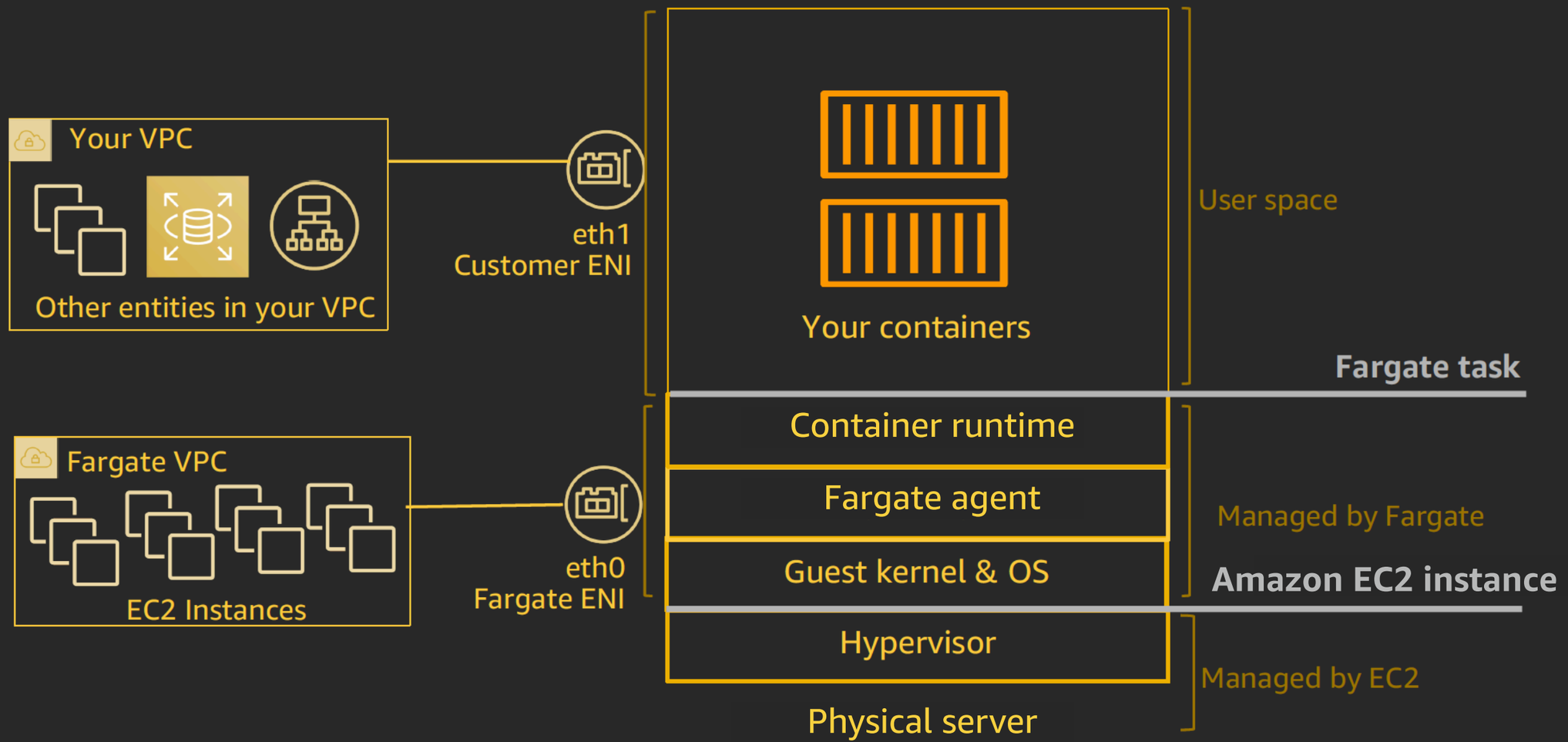
Only pay for the resources you need to run your pods.

Includes native AWS integrations for networking and security.

Amazon EKS on AWS Fargate architecture



AWS Fargate data plane

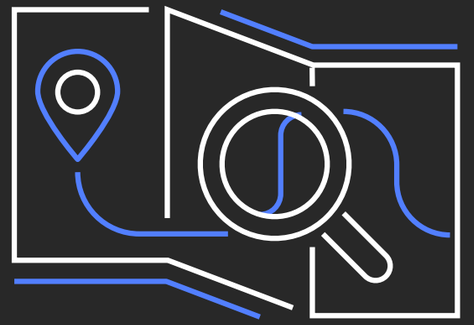


Thank you!

Tori Hara
toricls

aws SUMMIT ONLINE

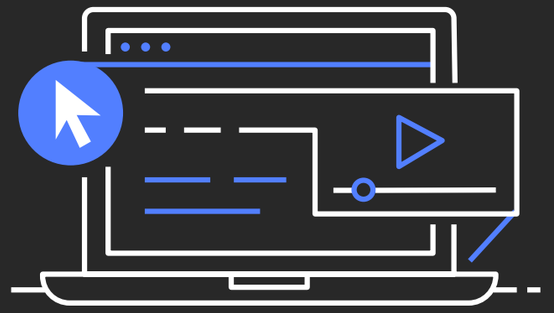
AWS トレーニングと認定



クラウド人材の育成

AWS トレーニングを活用し、ビジネスを牽引する人材の育成と組織作りを促進する

AWS トレーニング活用事例 »



自習コンテンツの活用

ウェビナーやのデジタルトレーニングを受講して、個人のスキルアップを目指す

AWS デジタルトレーニング »

AWS 認定取得を目指す

認定取得を目指して知識を底上げし、AWS の経験とスキルを証明する

AWS 認定の詳細 »

学習パスをお探しの方に

日本語版ランプアップガイドを公開しました。AWS ウェブページ、無料のデジタルトレーニング、クラスルームコース、動画、ホワイトペーパー、認定等を含んだ、9 種の役割別学習ガイドをご覧いただけます。 詳細を見る »

aws.amazon.com/training