GKEとEKS

Kubernetes meetup ～オンプレ?クラウド?事例共有会～

2020/10/08

株式会社ZOZOテクノロジーズ

SRE部

MLOps、プラットフォームSRE リーダー、CSIRT、SRE スペシャリスト

瀬尾 直利

ZOZO Technologies

瀬尾 直利（そのっつ）

株式会社ZOZOテクノロジーズ

SRE スペシャリスト

CSIRT

MLOps、プラットフォームSREリーダー

Twitter/GitHub: @sonots

CRuby、Fluentd コミッタ

ZOZOTOWN

https://zozo.jp/

- 日本最大級のファッション通販サイト
- 1,200以上のショップ、7,300以上のブランドの取り扱い（ともに2019年6月末時点）
- 常時73万点以上の商品アイテム数と毎日平均3,200点以上の新着 商品を掲載
- 即日配送サービス
- ギフトラッピングサービス
- ツケ払い など

Download on the App Store　GET IT ON Google Play

MLOps (ML-SRE) チーム

役割: ML機能のSRE

- 2019年4月にチームを発足
- 画像検索、髪型検索、検索パーソナライズ、推薦, etc.

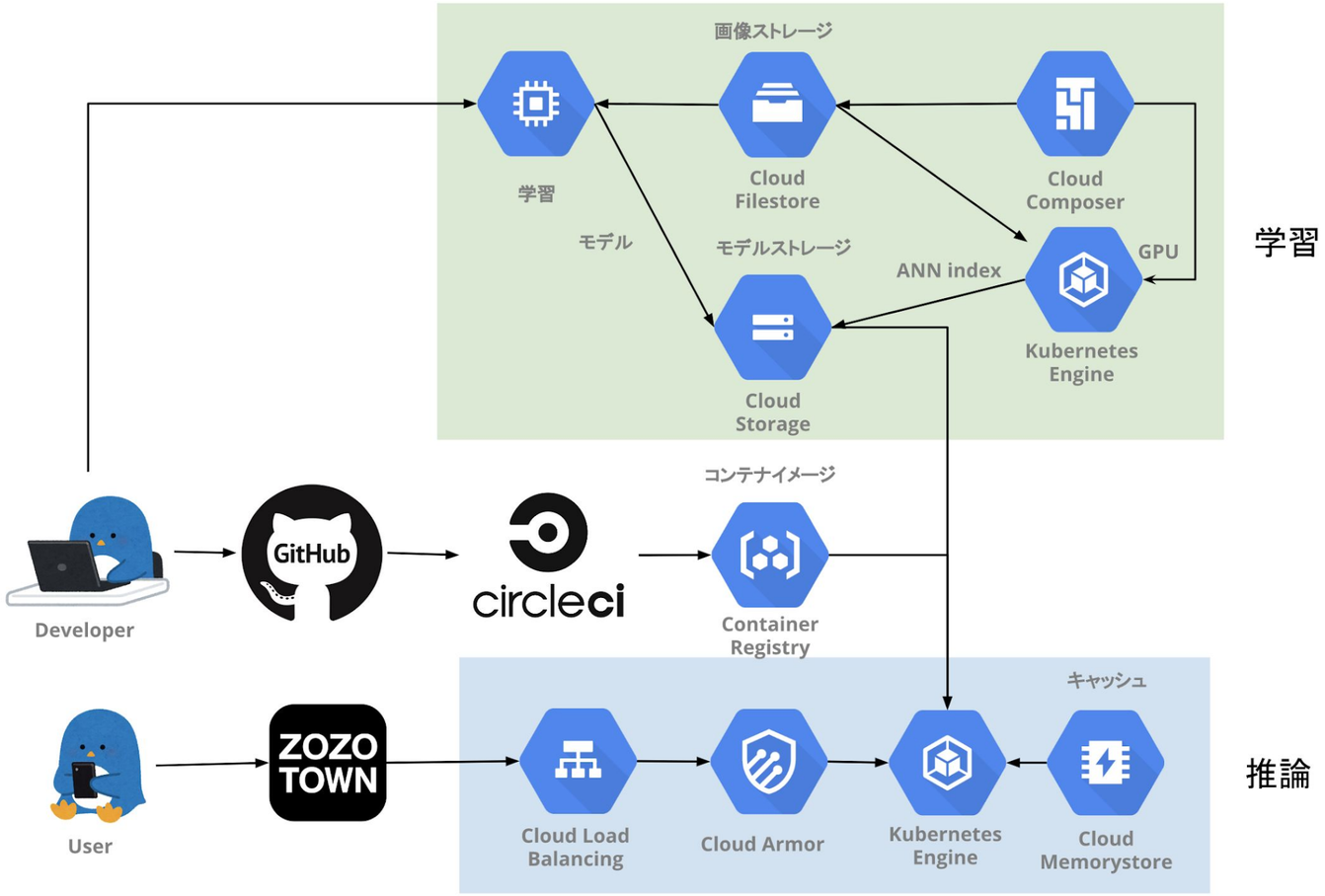
技術スタック : GCP

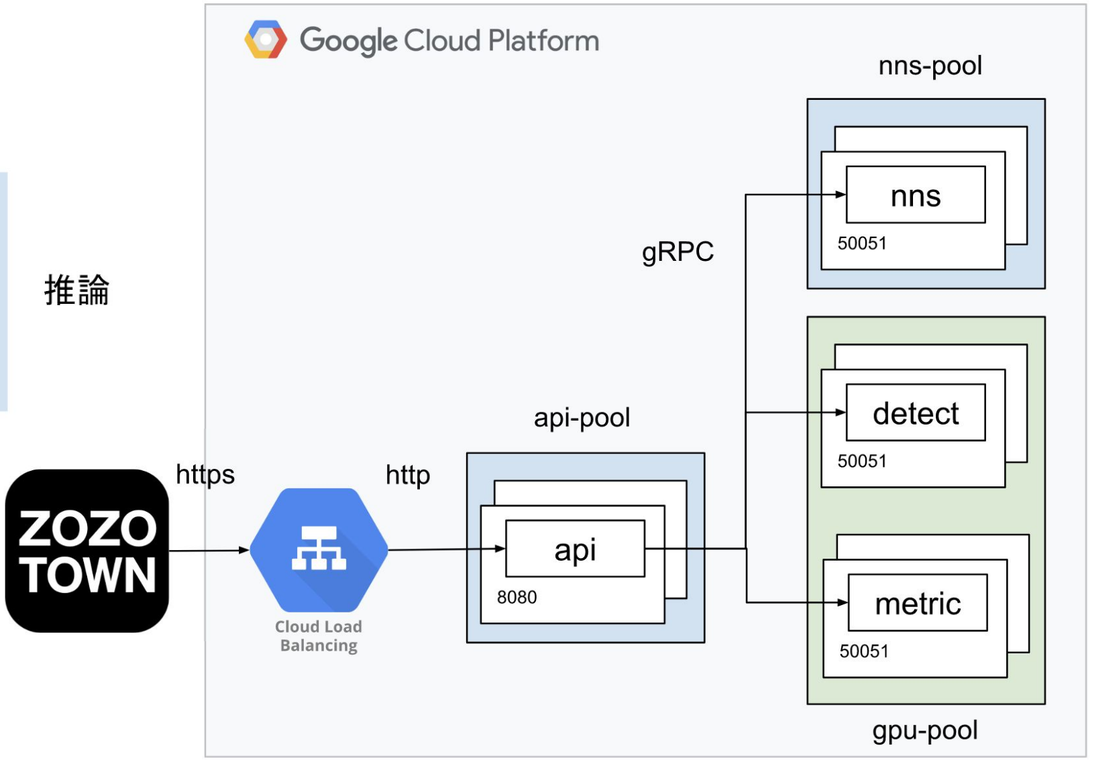
- ML API を GCP 上に構築。GKE を主に利用

Google Cloud Platform

Google
Kubernetes Engine

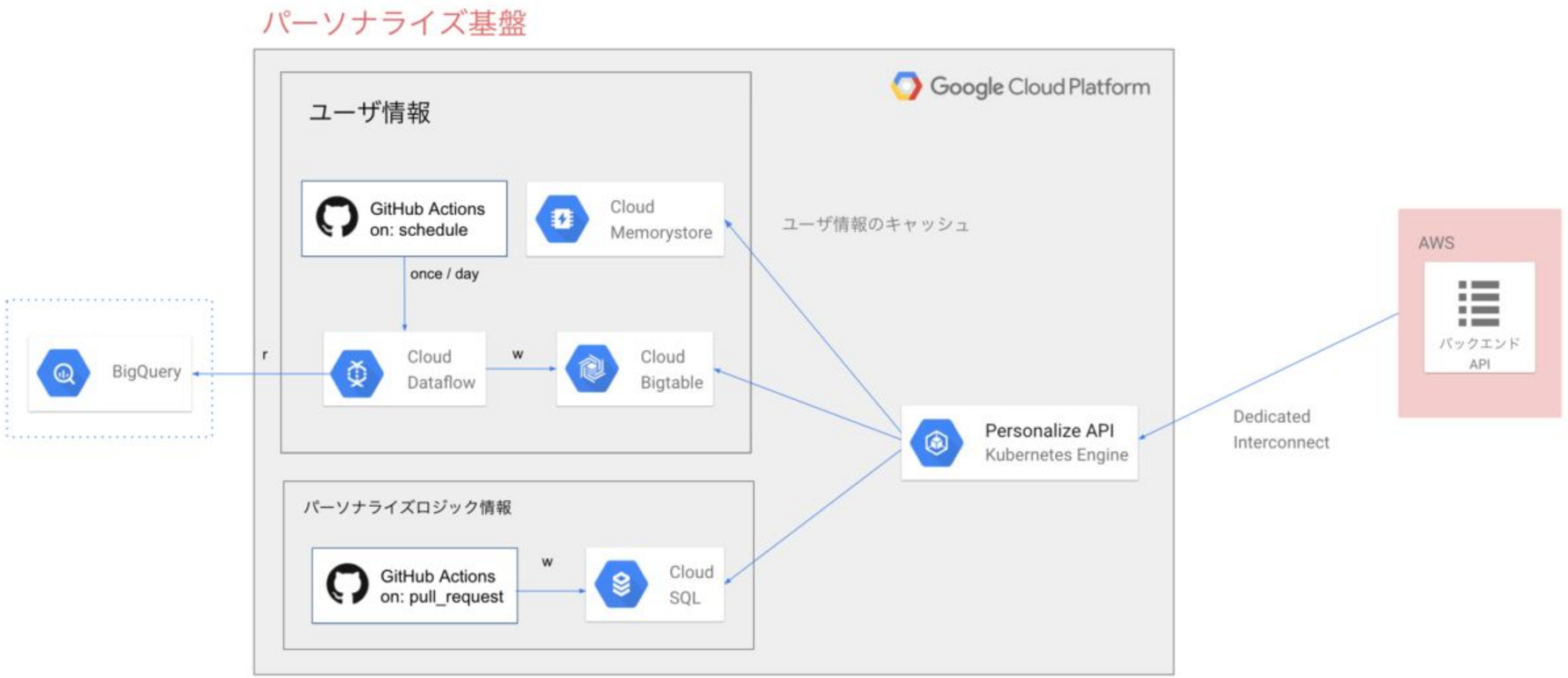
ZOZO 画像検索アーキテクチャ全体





https://techblog.zozo.com/entry/cloudnext19tokyo-imagesearch

検索パーソナライズ



プラットフォームSREチーム

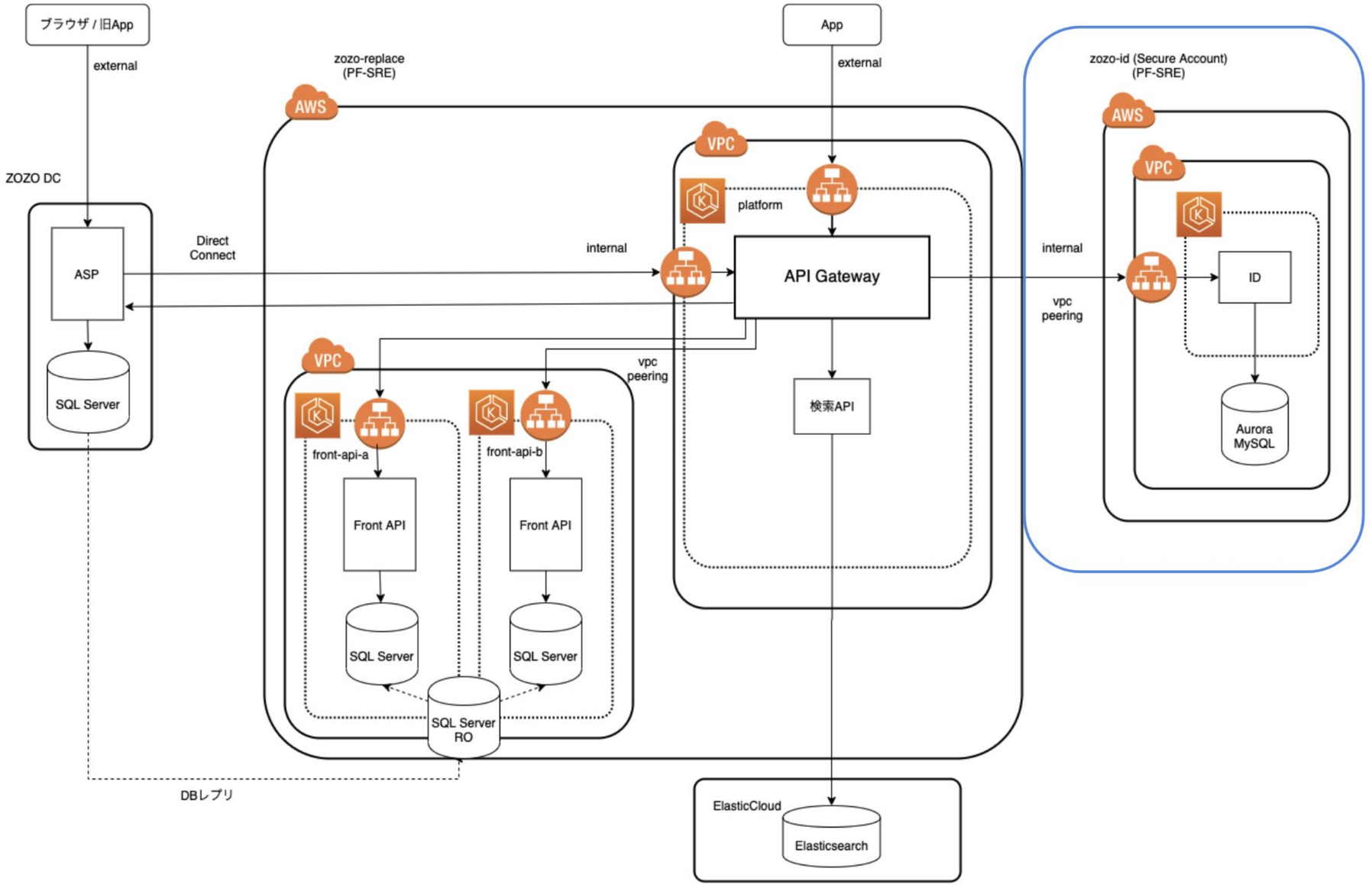
役割: ZOZOTOWNのマイクロサービス化リプレイス

- 2020年4月にチームを発足
- ZOZOTOWNのリプレイスをマイクロサービス化しながら進める
- ex) ZOZOにおけるID基盤のk8sへのリプレイスとセキュリティへの取組み

技術スタック : AWS

- ZOZOTOWN本体のリプレイス先はAWS
- マイクロサービスAPI群をEKS上に構築
  - マイクロサービスなりの工夫や苦労もあるが今回は割愛

ZOZOTOWNプラットフォーム (進行中)



https://event.cloudnativedays.jp/cndt2020/talks/20

GKEとEKS

本発表の内容

- GKEとEKSを本番で運用してきた人間が、
- それぞれの所感とハマりどころ、機能制約について語ります。

Disclaimer

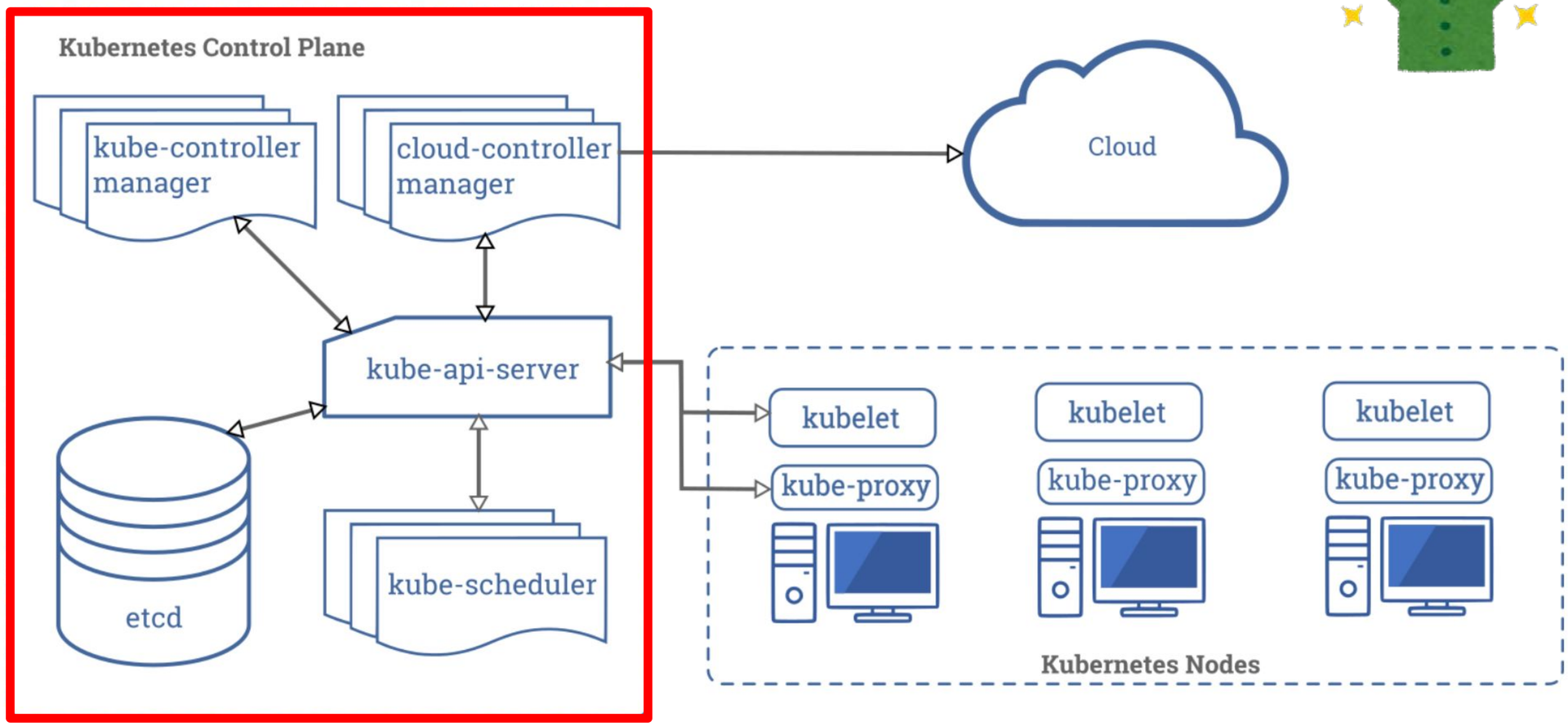
- ソリューションアーキテクトではないので全機能の網羅はしません
- どちらが良い、という結論を出すつもりはありません（両方使ってます）
- 「今はできない」と述べている機能が数ヶ月も経つとできるようになっている可能性があります
- 「それもうできるよ」とかあったらツッコミ歓迎

目次

- コントロールプレーン
  - データプレーン
  - マルチゾーン対応
- 初期構築
- ロードバランサー / 内部ロードバランサー
  - コンテナネイティブ負荷分散
  - TLS/SSLサポート
- DNS / Local DNSキャッシュ
- ノード自動プロビジョニング
- ネットワークとセキュリティ / IAMロール
- その他の制約

コントロールプレーン

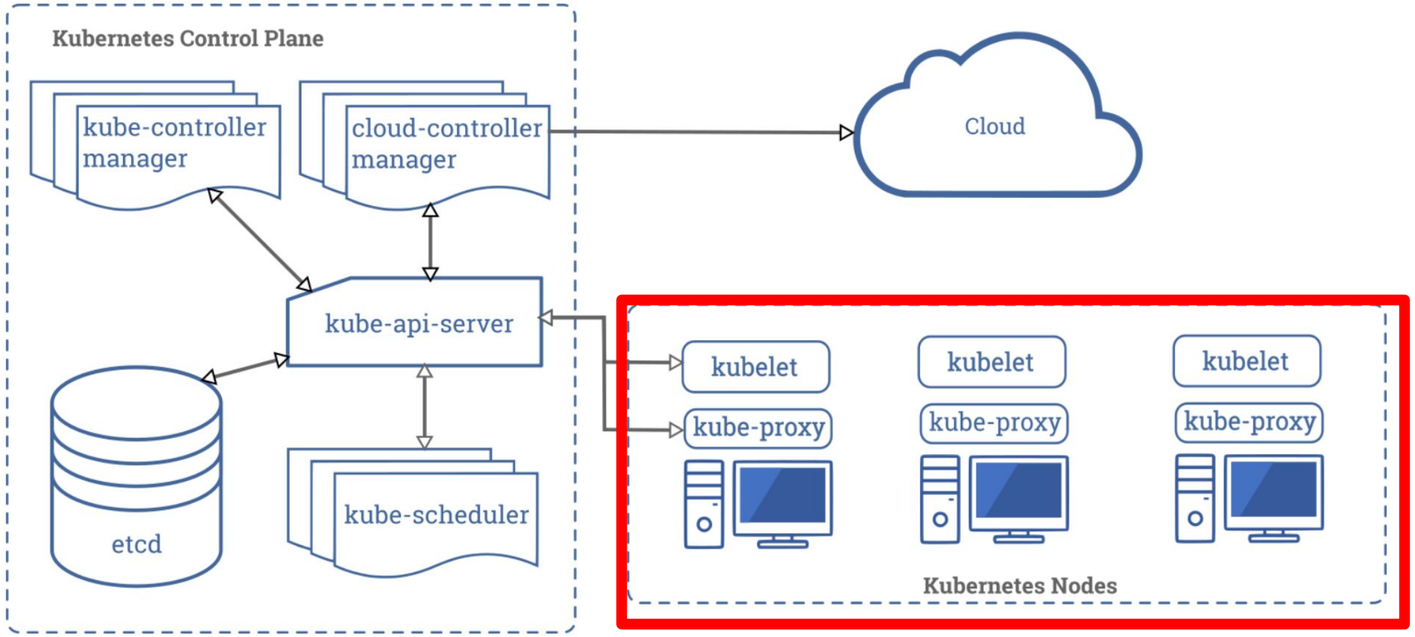
どちらもマネージド



https://kubernetes.io/ja/docs/concepts/overview/components

EKSのデータプレーン

- Self-managed (Unmanaged) Node Group
- Managed Node Group
- Fargate



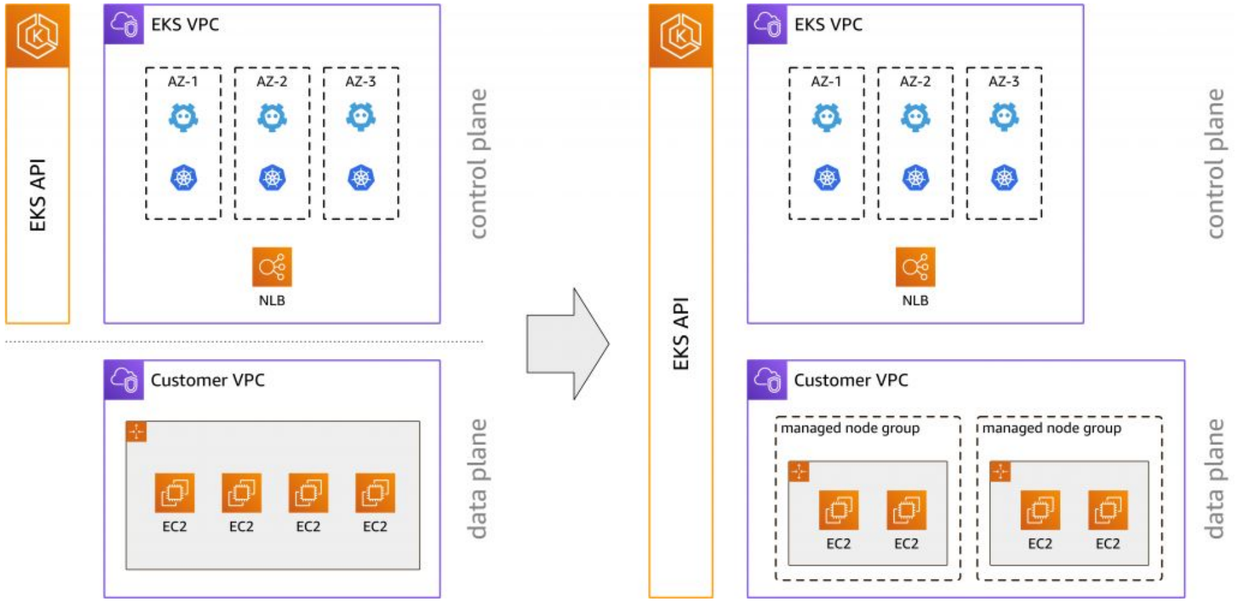
Self-Managed (Unmanaged) Node Group

- 少し前までEKSサービスで用意できるのはコントロールプレーンのみだった
  - 提供される CloudFormation テンプレートないしは eksctl を利用して Auto Scaling Group を作成し EKS に組み込んでいた
  - これが Self-Managed Node Group

https://dev.classmethod.jp/articles/eks-support-provisioning-and-managing-worker-nodes/

Managed Node Group

- 2019/11: Managed Node Group 提供開始
- コンソール/CloudFormationから(普通に)ノードグループを作成できるように
- ノードグループの安全な削除に対応
  - in-place な version upgrade にも対応



https://aws.amazon.com/jp/blogs/containers/eks-managed-node-groups/

EKS on Fargate

- 2019/12: ECSだけでなくEKSでもFargateが使えるようになった
- Fargateは「仮想マシンの管理なしにコンテナをデプロイできるサービス」
  - つまり、ノードがない
  - データプレーンの k8s version upgrade しなくて良い
- 制約: DaemonSet が作れない (ノードがないので) [1]
- 制約: ~~永続ボリュームなど Stateful ワークロードには使えない~~
  - 2020/08: EFSがサポートされ永続ボリュームが使えるようになった
- 制約: ポッドごとに最大 4 vCPU 30 Gb Memory の制約 [2]

[1] https://docs.aws.amazon.com/eks/latest/userguide/fargate.html
[2] https://docs.aws.amazon.com/ja_jp/eks/latest/userguide/fargate-pod-configuration.html

GKEのデータプレーン

- GKE ではノードプールと呼びます
- 当初から EKS の Managed Node Group に相当するものを提供
  - in-place な version upgrade に対応
- Fargate のようなものはない

マルチゾーン対応

EKS

- コントロールプレーンは最初から 3AZ
- データプレーンの Multi-AZ 化はユーザ選択 (複数サブネットを指定)

GKE

- コントロールプレーンの可用性を高めるには Regional Cluster にする
  - Single Zone や Multi Zone クラスタではコントロールプレーンは 1 Zone
- 制約: 以前は Regional Cluster ではノード自動プロビジョニングが使えなかったりと制約があったが最近は解消されている
- 制約: GCPではZoneによって使えない instance type があったりするので注意

初期構築

EKS

- AWSオフィシャルで eksctl の利用を推奨 [1]
  - weaveworks が独自に作成したツールだったがオフィシャルになった
- eksctl を利用すると CloudFormation (CFn) スタックが作られ生成される
  - 弊社ではここから CFn テンプレートを取り込み、CFn から変更する運用フローにした
  - 弊社では EKS に限らず CFn を適用するインフラ CI/CD を構築しているため [2]
- Managed Node Group が使えるようになってからは、eksctl に頼らずとも自前で CFn を書ける程度のボリュームにはなった (個人の感想)

GKE

- 自前で Terraform を書ける程度のボリューム (クラスタとノードプールの作成)

[1] https://docs.aws.amazon.com/ja_jp/eks/latest/userguide/getting-started-eksctl.html
[2] https://medium.com/@sonots/effects-of-infrastructure-as-code-on-full-remote-8cbcb416e3a3

初期構築 - クラスタ認証

EKS

- aws-iam-authenticator を利用してEKSクラスタ認証する必要があった
  - 2019/05: awscli に aws eks get-token が追加され awscli のみで ok に
- EKSクラスタを作成したユーザにのみ kubectl を叩く権限が付く仕様
  - 追加で割り当てたい場合 aws-auth.yaml を編集して kubectl apply

GKE

- 初期から gcloud container clusters get-credentials でクラスタ認証
- IAMアカウントで権限があれば kubectl を叩ける (特に困難なし)

初期構築 - kube-system

EKS

- 標準では kube-system レベルのものが色々入っていないので自分で入れる
- ドキュメントはあるので、ドキュメント通り以下を追加でデプロイした
  - metrics-server
  - cluster-autoscaler
  - alb ingress controller
  - container-insights (fluentd-cloudwatch logs)
- また k8s events をログ出力する event-exporter もないので 3rd party OSS をデプロイした

GKE

- 上記相当が標準装備

目次

- コントロールプレーン
  - データプレーン
  - マルチゾーン対応
- 初期構築
- ロードバランサー / 内部ロードバランサー
  - コンテナネイティブ負荷分散
  - TLS/SSLサポート
- DNS / Local DNSキャッシュ
- ノード自動プロビジョニング
- ネットワークとセキュリティ / IAMロール
- その他の制約

ロードバランサー / 内部ロードバランサー

EKS

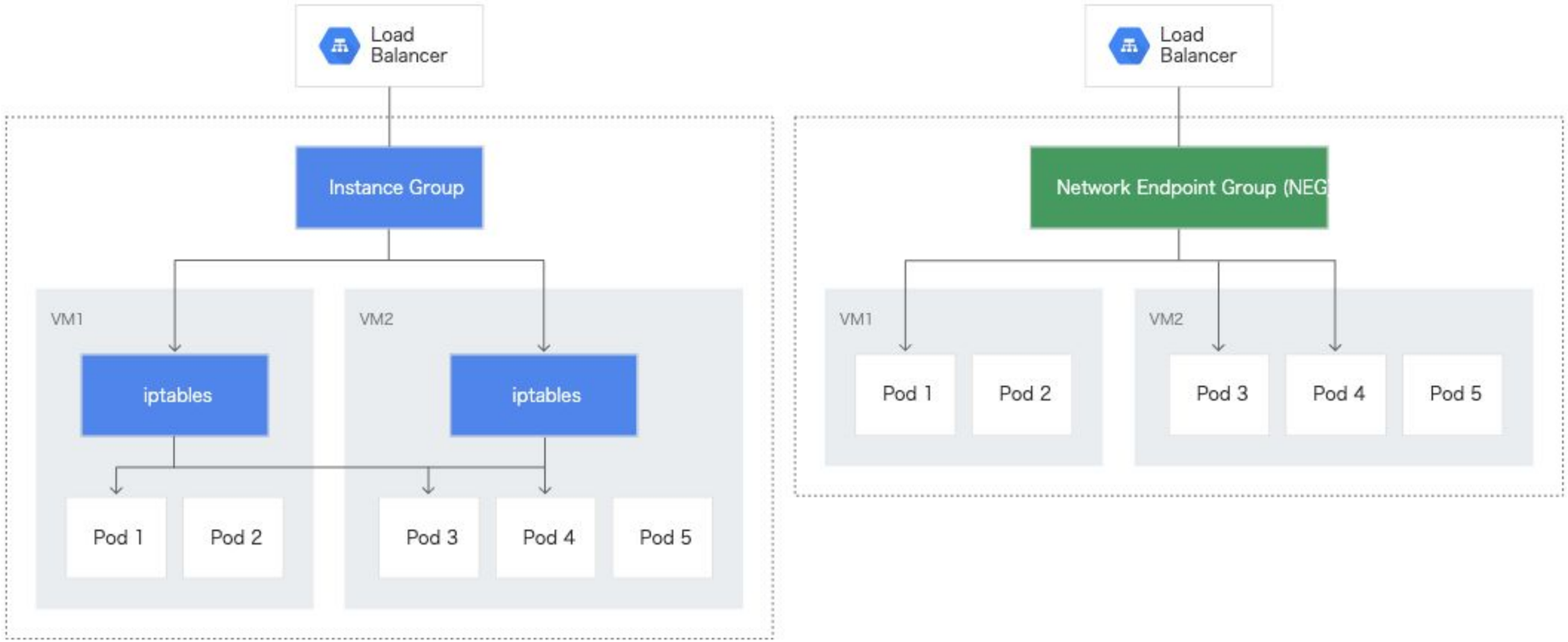
- Ingress ALB、LoadBalancer NLB
- 内部LBも苦労なく使える

GKE

- Ingress GCLB、LoadBalancer GCLB
- 2019/12: 内部LBをサポート
  - 内部的には Envoyベースとのことで全く別の新しい仕組み
- 当初は Ingress が内部LBに対応しておらず、gcloud で立てて紐付けたりしておりました、、、今は使えますがまだβ

コンテナネイティブ負荷分散

kube-proxy (iptables) を経由しないので速い



デフォルトの動作（左）とコンテナ ネイティブ（右）のロードバランサーの動作の比較

https://cloud.google.com/kubernetes-engine/docs/concepts/container-native-load-balancing?hl=ja

コンテナネイティブ負荷分散

EKS (Ingress ALB)

- instance mode: NodePort サービスを介す (kube-proxy を介す) [1]
- ip mode: PodのIPアドレスをALBに直接登録する (kube-proxyを介さない)

GKE (Ingress GCLB)

- Network Endpoint Group (NEG) なるものを利用してサポート [2]

注意: どちらも kube-proxy を介さなくなるので k8s のサービスディスカバリではなく LB の health check に頼るようになります。要調整。

[1] https://labs.gree.jp/blog/2020/01/20271/
[2] https://cloud.google.com/kubernetes-engine/docs/concepts/container-native-load-balancing?hl=ja

TLS/SSLサポート

EKS

- Ingress に Amazon Certificate Manager (ACM) のリソースを指定する

GKE

- Ingress に Managed SSL Certificate のリソースを指定する

DNS / Local DNSキャッシュ

背景: k8s はサービスディスカバリをDNSに頼っている。DNSは通常UDP通信をするためパケロスして 5sec タイムアウトしがち(最近はTCPサポートしているがクライアントが対応していなかったり)

ローカルDNSキャッシュの高い需要

EKS

- DNSサーバに CoreDNS を利用
- ローカルDNSキャッシュは各自頑張る (という認識)

GKE

- DNS サーバに (いまだに) kube-dns を利用 (こちらの方が性能が良いらしい)
- 2020/07: NodeLocal DNSCache が GA に。クラスタ作成で addon を入れる

ノード自動プロビジョニング

resources.requests に応じて適切なノードグループ(プール)がなければ自動で作成してノードを立ち上げる(GKE独自の？)機能

EKS

- 対応していない

GKE

- 対応している

ネットワークとセキュリティ

EKS

- Managed Node Group には ~~Security Group をつけられない~~
  - 2020/08: Launch Template のサポートによりSGを追加できるように
- Pod レベル Security Group
  - 2020/09: Pod レベルで Security Group をつけられるように

GKE

- Node Pool に Firewall Rule を適用できる (ネットワークタグをつける)
- Pod レベルはまだできない？

IAMロール

EKS

- Node Group に IAMロールを割り当てられる
- 2019/09: IAM Roles for (k8s) Service Accounts
  - Pod 単位でIAMロールを割り当てられるように

GKE

- Node Pool にGCPサービスアカウントを割り当てられる
- 2020/03: Workload Identity
  - PodレベルでGCPサービスへの認可制御ができるように

その他の制約

EKS

- インスタンスタイプに応じて割り当てできる最大IPアドレス数制限がある
  - 例えば、m3.medium の場合 11 pods で頭打ち
    - k8s events に「insufficent pods」というワーニングが出る

GKE

- kube-system のサービスは編集できない(戻される)
  - 例えば、kube-dns を落として coredns に入れ替えようとしたが難しい

まとめ

- どちらが良い、という結論を出すつもりはありません(再掲)
- どちらも本番のワークロードで使えます
- GKEかEKSかというよりは、GCPかAWSかという全体を見て選択することになるかと思います
- それぞれ多少の差異はありますが、切磋琢磨している印象
- 差分で混乱した時に本資料が役に立てば幸いです