

# Azure + クラウド型電子カルテにおける、 リソース利用効率の課題と改善への道すじ



きりんカルテ

# マイクロソフト、きりんカルテシステム 協業の取り組み

## 井上 章

2008 年、日本マイクロソフト株式会社入社。  
Global Black Belt (GBB) というマイクロソフト  
コーポレーションの技術専門組織で、主に .NET と  
Microsoft Azure を中心としたアプリケーション開  
発技術と Azure DevOps の訴求活動に従事。

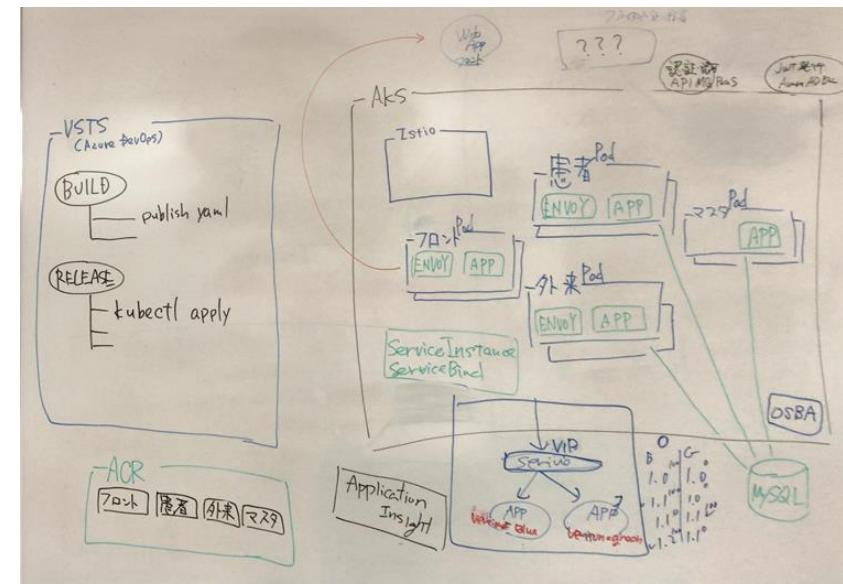
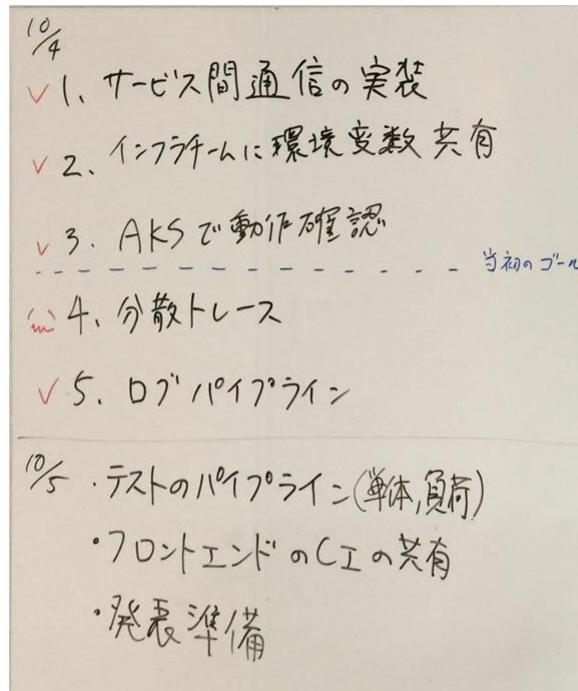
マイクロソフト コーポレーション  
グローバルブラックベルト  
テクノロジー ソリューションプロフェッショナル



# きりんカルテ & マイクロソフトの協業とクラウドネイティブへの取り組み

数回にわたる Hack Fest やアーキテクチャ デザイン セッションの実施

- Azure Database for MySQL の早期導入 (2018 年 1 月)
- AKS による外来受付サービスのマイクロサービス化と CI/CD 化 (2018 年 10 月)
- AKS を利用したレセコン (Orca) のコンテナ化



## 安部 勇輝

Ops畑 を歩んでもう20年。ネットワーク、セキュリティを中心に、クラウドのアーキテクチャー選定、設計を担当しています。

今回のチャレンジでは、プロジェクトの方向性やスコープ等、要件決めフェーズに主に参画しています。

きりんカルテシステム株式会社  
プラットフォーム・スペシャリスト



# 木下 真哉

エンジニア歴16年のフルスタックエンジニア。  
(Dev歴：16年、Ops歴：8年)

きりんカルテシステムでは、開発環境の整備、アプリケーション基盤の開発、新技術の検証及び導入、インフラ構築のコード化を中心としたDevOpsの推進活動に従事しています。

今回のチャレンジでは、Kubernetes(AKS)の技術検証及びレセコン(ORCA)のコンテナ運用の設計をSREの観点で取り組んでいます。

きりんカルテシステム株式会社  
SREエンジニア



# 当社 提供サービス 概要

## — ビジョン —

日本の電子カルテ普及を推進し、  
社会のより良い医療体験づくりに貢献する

---

社名	きりんカルテシステム株式会社
代表	山口 太一（やまぐち たいいち）
設立	平成25年12月3日
資本金	261,480,000円
事業内容	電子カルテ開発・運用保守事業 スマホアプリ・Webシステム開発事業

■ 東京本社  
〒107-0062 東京都港区南青山 3-1-31  
NBF南青山ビル 9F  
Tel: 03-6447-0963  
Fax: 03-6447-0965

■ 福岡開発本部  
〒812-0011 福岡県福岡市博多区博多駅前 4-2-20  
博多駅前 C-9ビル 7F  
Tel: 092-409-1033  
Fax: 092-402-8660

全国に約10万ある「一般診療（クリニック）」では、35%に留まっている。主にコストの問題

	一般病院	病床別規模			一般診療
		400床以上	200～399床	200床未満	
平成20年	14.2%	38.8%	22.7%	8.9%	14.7%
平成23年	21.9%	57.3%	33.4%	14.4%	21.2%
平成26年	<b>34.2%</b>	<b>77.5%</b>	<b>50.9%</b>	<b>24.4%</b>	<b>35.0%</b>

出典：医療施設調査(厚生労働省)

クリニックのIT投資を  
劇的に下げる

電子カルテを  
患者のものにする



・・・なぜ「きりん」なんですか

よく尋ねられますので



しまうまプリント  
システム 株式会社

きりんカルテ  
システム 株式会社

## 提供システムの全体イメージ



予約

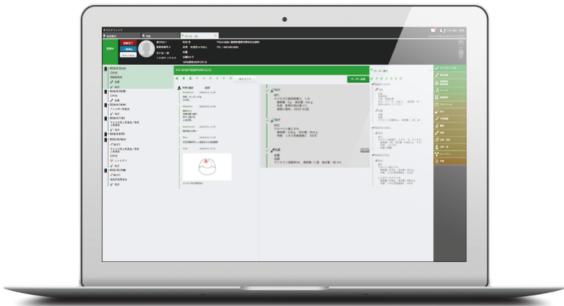
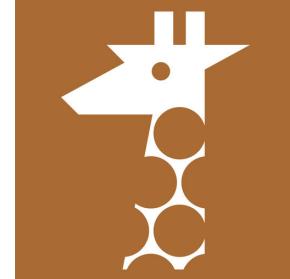


外部  
機器  
検査  
結果

連携

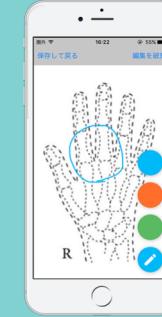


# カルテZERO



日医標準レセプト  
ORCA(オルカ)

患部撮影



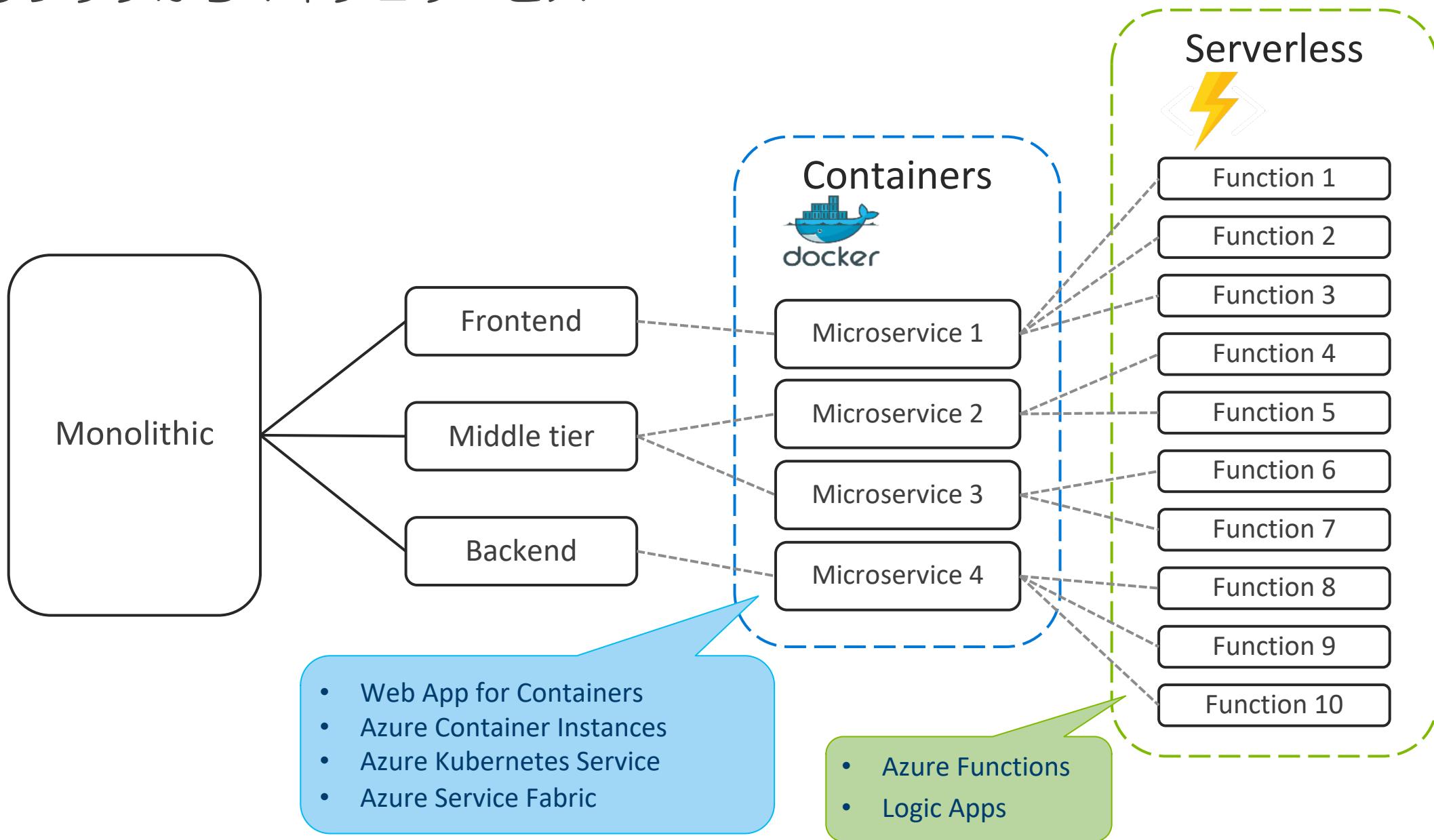
オフライン  
カルテ



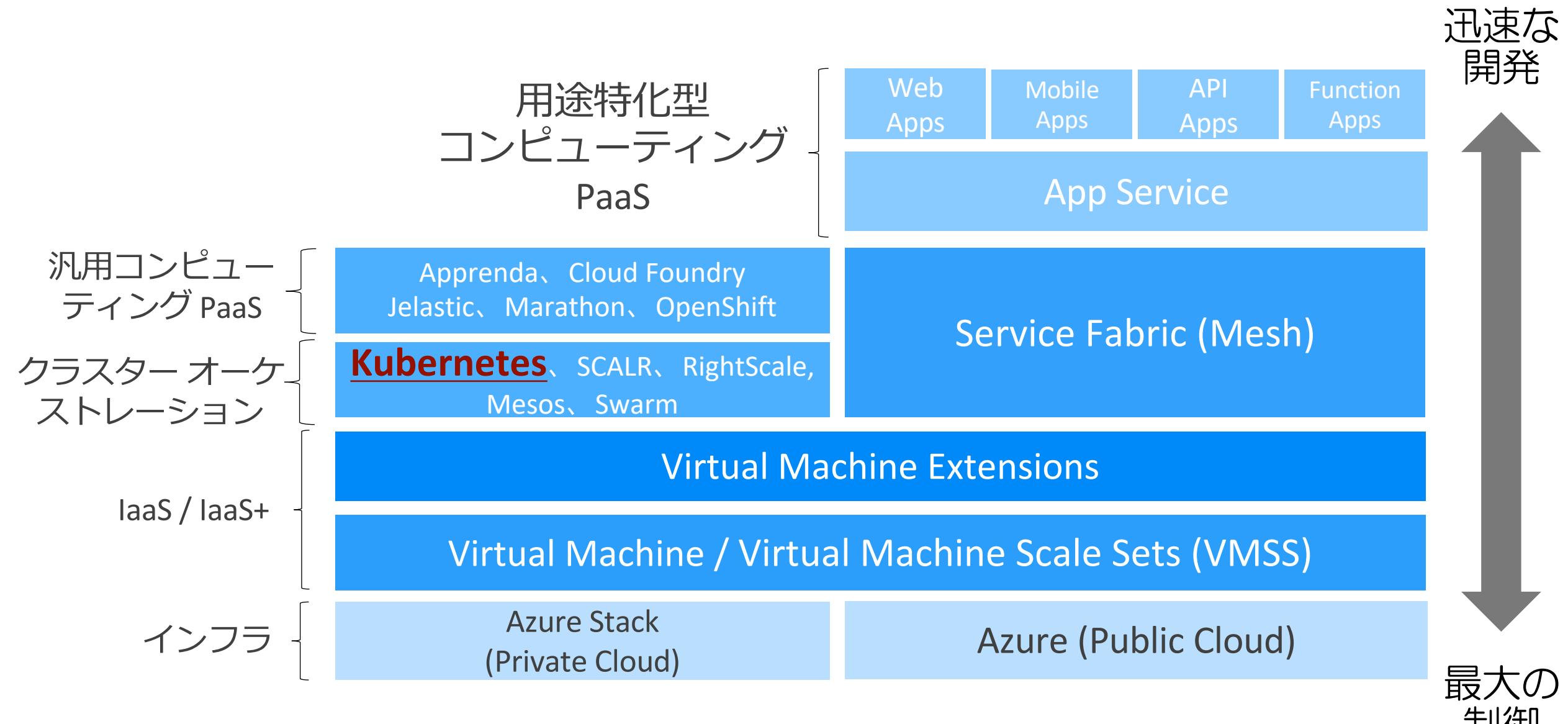
きりんカルテ

# 技術選定やアーキテクチャについて

# モノリシックからマイクロサービスへ



# Azure コンピューティング サービス の選択肢



概要  ソリューション製品 ドキュメント 價格 トレーニング Marketplace  パートナー  サポート  ブログ その他  無料アカウント  おすすめ

データベース

AI + 機械学習

ネットワーク

DevOps

メディア

ID

モノのインターネット (IoT)

Microsoft Azure Stack

モバイル

Storage

移行

Web

開発者ツール

コンテナー

管理

コンピューティング

統合

セキュリティ

分析

すべて表示  
(100+)

すべての製品を検索

## コンピューティング

クラウドのコンピューティング キャパシティ、必要に応じたスケーリングを手に入れましょう。お支払いは使用したリソース分だけ

### Virtual Machines

Windows と Linux の仮想マシンを数秒でプロビジョニング

### Azure Kubernetes Service (AKS)

Kubernetes のデプロイ、管理、操作を簡略化する

### Service Fabric

Windows または Linux でのマイクロサービスの開発とコンテナーのオーケストレーション

### Container Instances

サーバーを管理することなく Azure でコンテナーを簡単に実行

### Azure Batch AI

容易に実験を行い、ディープ ラーニングおよび AI モデルを大きな規模で並行してトレーニングする

### Virtual Machine Scale Sets

数千個の Linux および Windows 仮想マシンを管理およびスケールアップ可能

### Functions

サーバーレス コードを使用してイベントを処理

### App Service

Web およびモバイル向けのパワフルなクラウド アプリを短期間で作成

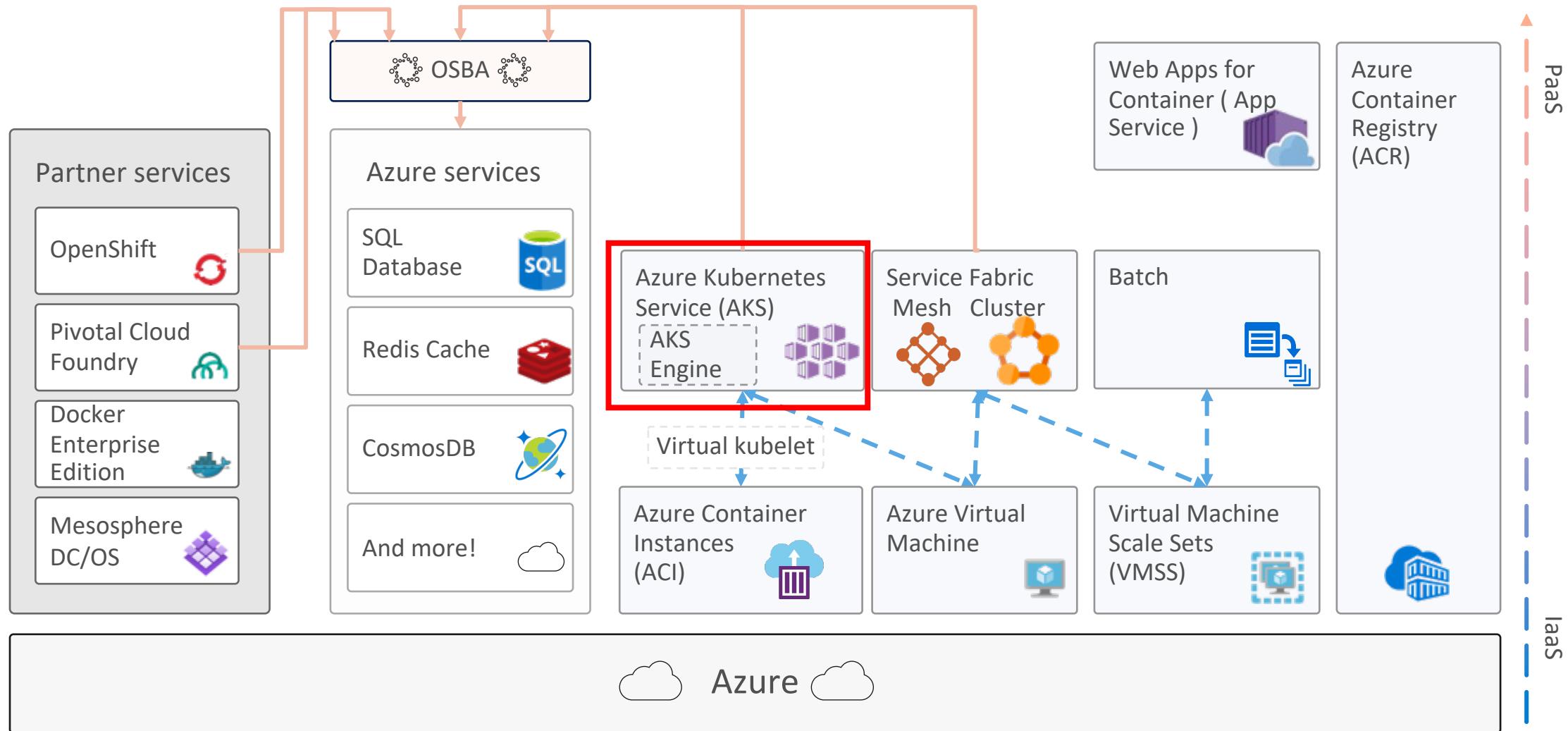
### Batch

クラウド規模のジョブ スケジュール設定とコンピューティング管理

[詳細を表示 >](#)

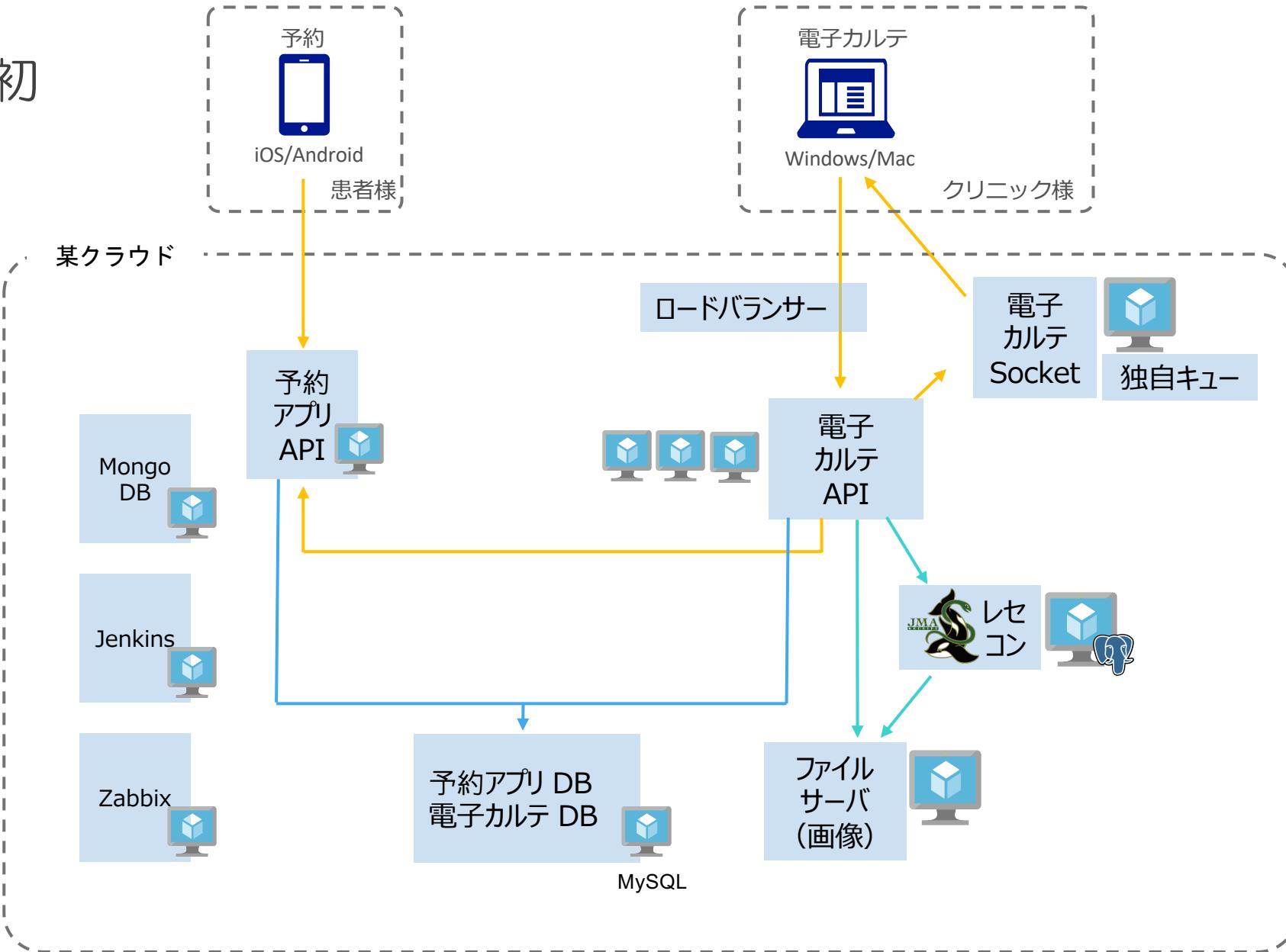


# Azureのコンテナー エコシステム

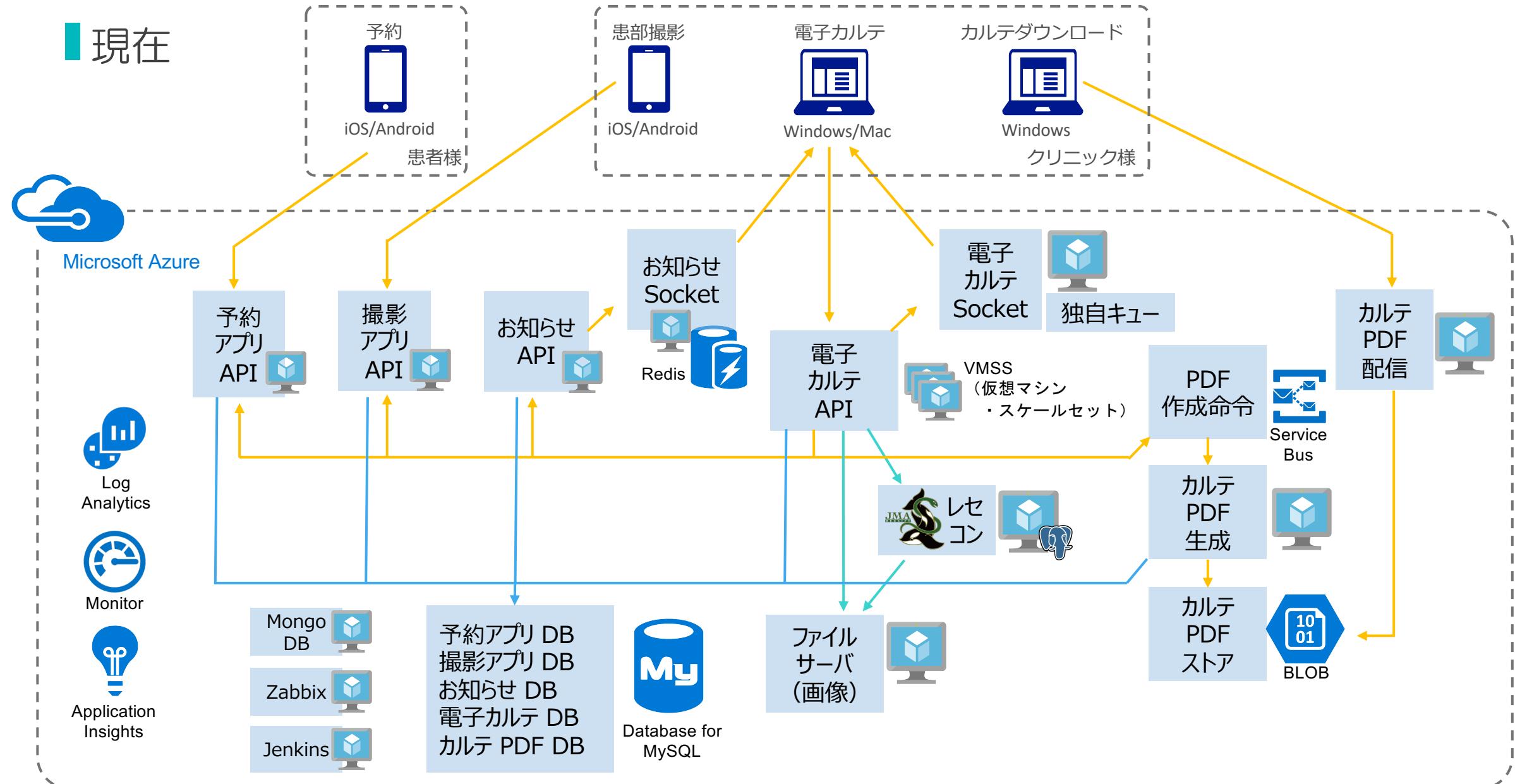


# Cloud Native の取り組み

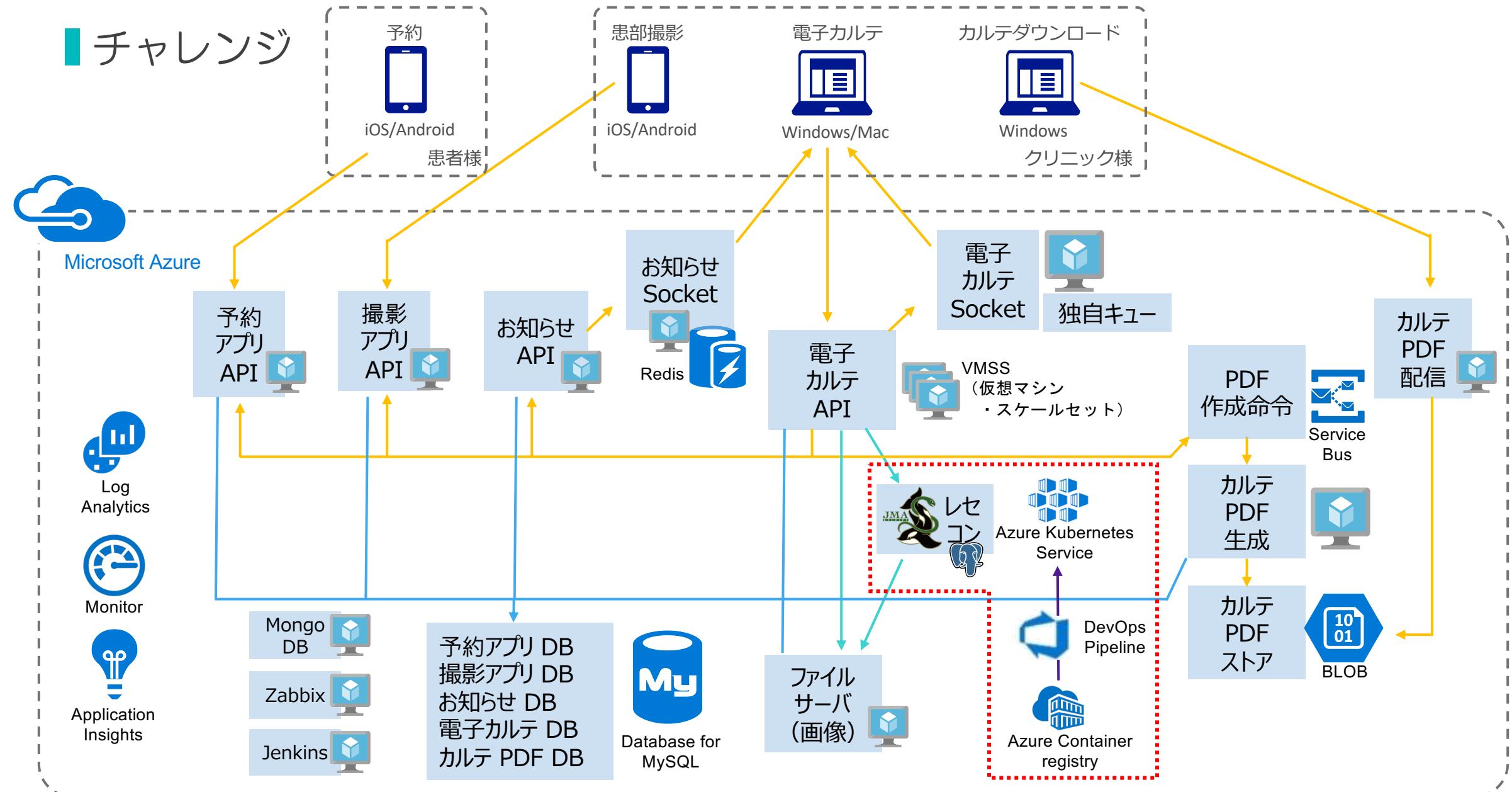
# サービス 開始当初



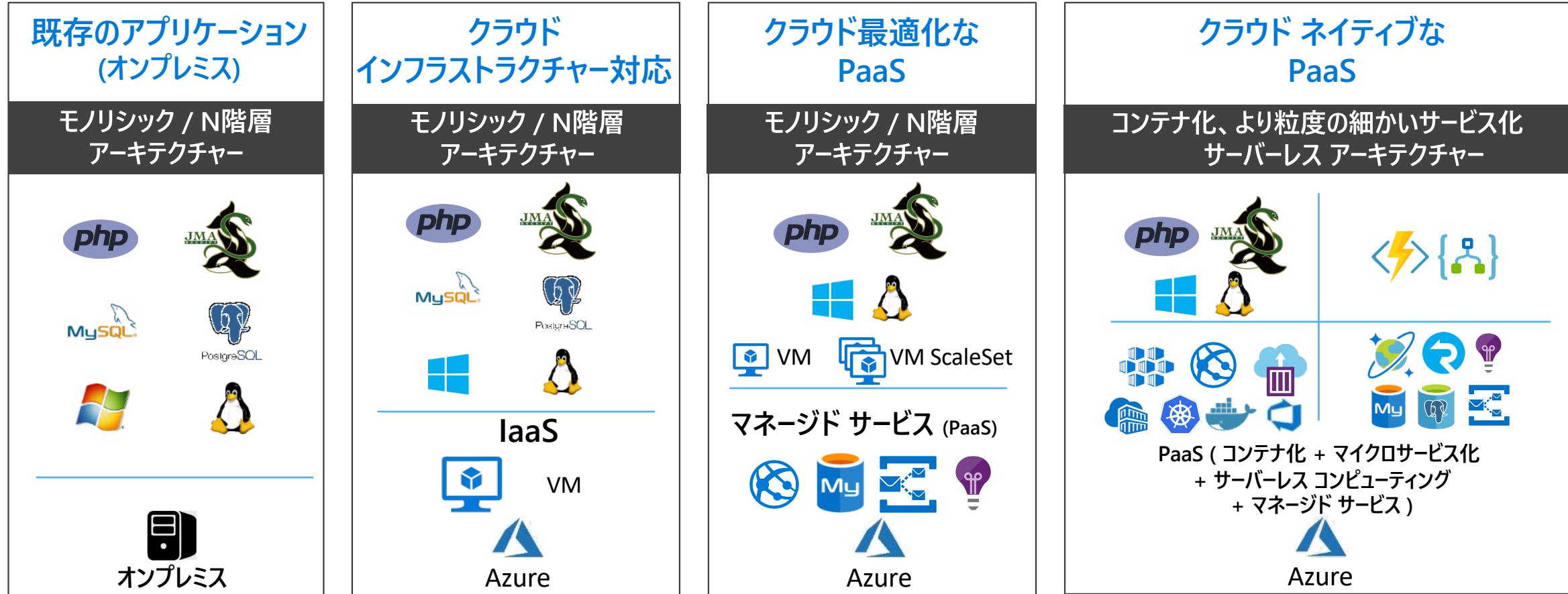
# 現在



# チャレンジ



# アプリケーション最適化モデル



クラウド環境が基本。検討事項: ネットワーク、ハイブリッド クラウド、IDと認証、コストコントロールと運用モデル

移行 / 再ホスト

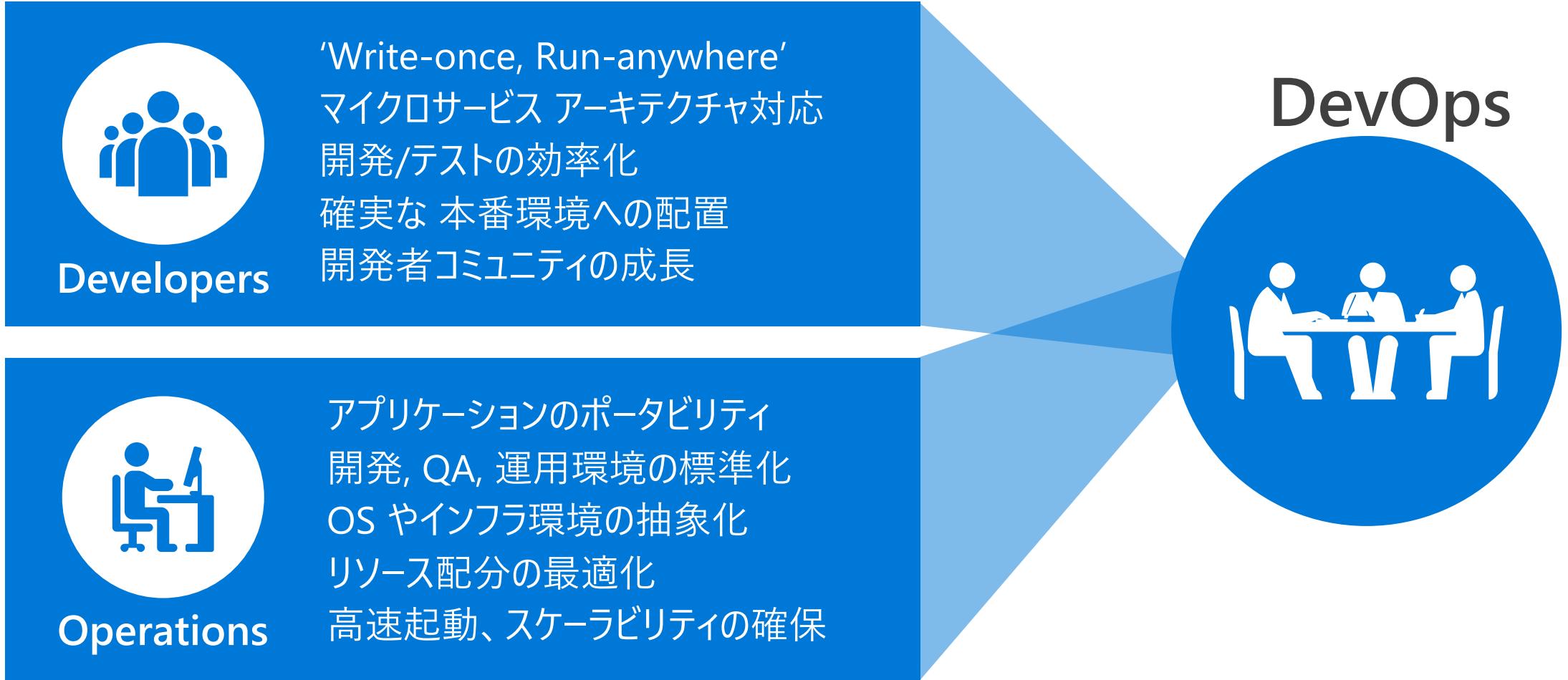
モダナイゼーション

コードの変更は最小限

クラウド向けに設計された新規のコード

▲ 今回のお話

# なぜ、コンテナーか？ – Write-once, Run-anywhere



# アーキテクチャーの選定にいたった経緯

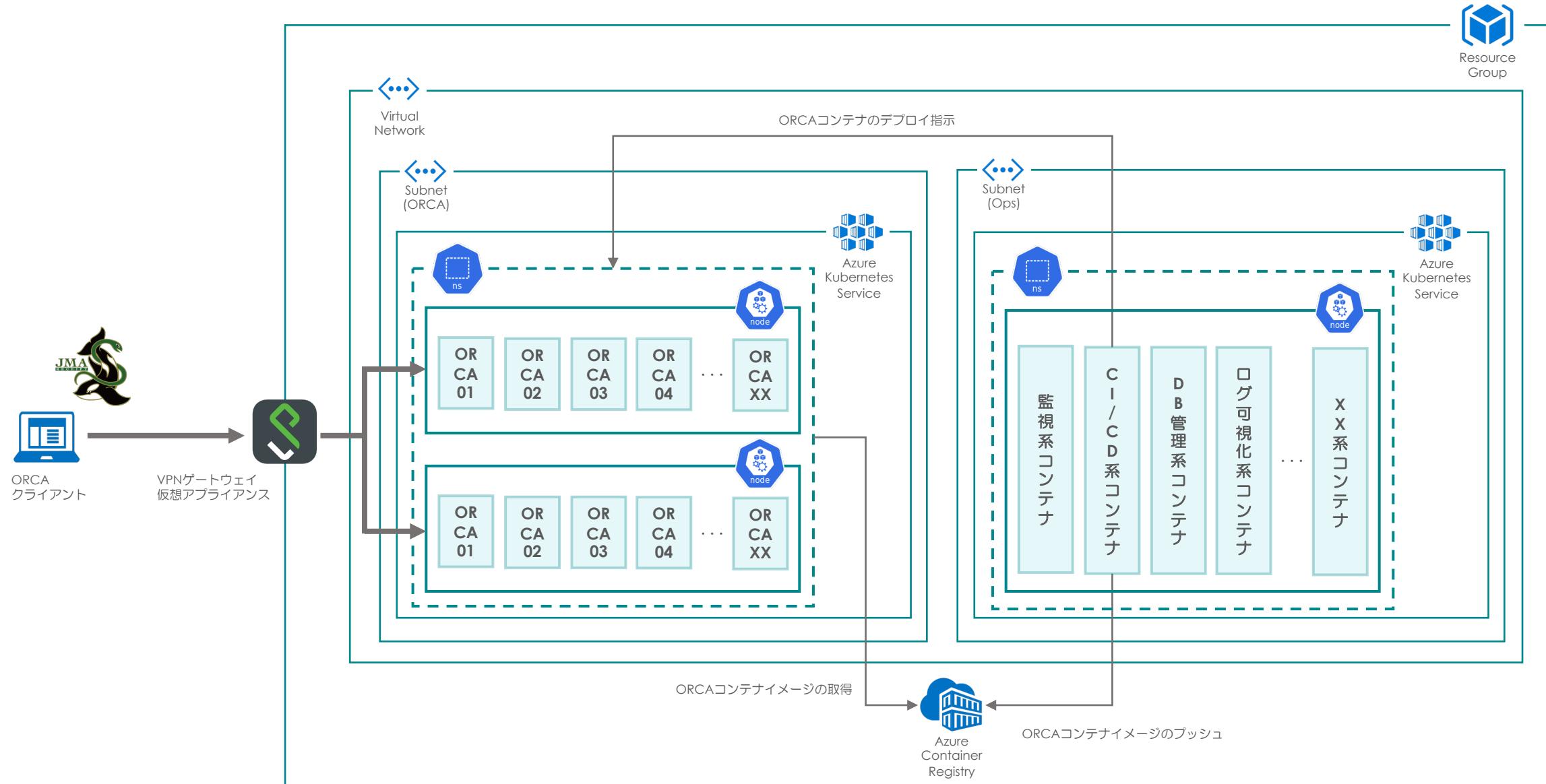
- クラウドネイティブ化
  - ✓ 大規模にスケールさせることができ、同時に OS・DB等の運用にかかる工数が削減される。
  - ✓ 障害対策やバージョンアップなど、インフラ固有のスキルを、クラウド事業者にアウトソースすることで、開発者主体でも運用ができる仕組みが実現できる。
- コンテナ化
  - ✓ すでに、開発者の環境ではコンテナを利用していたので、結合テストやステージング、本番環境にも採用すれば、デプロイの品質があがる。
  - ✓ PHP等のミドルや、ライブラリーなどの、メジャー・バージョンアップを、低いリスクで円滑に行える

# AKSを利用したレセコン（Orca）の コンテナ運用 チャレンジ

## コンテナ運用における必須要件

- Azure Kubernetes Serviceのクラスター内のノード1台辺り、20台程度のORCAコンテナが稼働できること
- コンテナの起動・廃棄が繰り返されても、ORCAで管理しているデータ(患者、レセプト等)が消失しないこと
- VPN経由でレセコン(ORCA)に接続すること

# 目指そうとしているアーキテクチャ (全体図)



## 今回使った技術的な要素 (1/2)

- Ansible … オープンソースの構成管理ツール。ORCAのコンテナイメージをビルドする際に行われるミドルウェアのインストール・設定をする際に使用。
- Packer … オープンソースのサーバーイメージ作成ツール。ORCAのコンテナイメージを作成する際に使用。
- Azure CLI … マイクロソフト社から提供されているAzureリソースを管理するためのコマンドラインツール。Azureリソース (リソースグループ、仮想ネットワーク、コンテナレジストリ(ACR)、Kubernetes(AKS)) を作成する際に使用。

## 今回使った技術的な要素 (2/2)

- Kubernetes … オープンソースのコンテナオーケストレーションシステム。コンテナ化したORCAのデプロイや管理をするために使用。
- Helm … オープンソースのKubernetes用パッケージ管理システム。Kubernetes上に特定のリソースをインストールするために使用。

## 今回使った Kubernetes (AKS) のリソース (1/3)

- Persistent Volume (pv) … 永続化ボリュームを提供するオブジェクト。  
ORCAで使用しているPostgreSQLのデータストレージとして使用。
- Persistent Volume Claim (pvc) … 永続化ボリュームの利用要求をするためのオブジェクト。Pod作成時に永続化ボリュームをORCAコンテナに割り当てる際に使用。
- Storage Class (sc) … ストレージの種類を示すオブジェクト。Azure Filesを永続化ボリュームとして使用できるようにするために使用。

## 今回使った Kubernetes (AKS) のリソース (2/3)

- Pod (pod) … Kubernetesにデプロイできるリソースの最小単位となるオブジェクト。ORCAコンテナを稼働させるために使用。
- Replica Set (rs) … 指定されたPodのだけ、Podの稼働を保証するためのオブジェクト。ORCAコンテナを含んだPodが常に 1 つ稼働している状態にするための信頼性向上のために使用。
- Deployment (deploy) … Replica SetをKubernetesにデプロイする際の制御を行うオブジェクト。

## 今回使った Kubernetes (AKS) のリソース (3/3)

- Service (svc) … コンテナを外部に公開するためのエンドポイント(IPアドレス)を提供するオブジェクト。
- Ingress (ing) … HTTP/HTTPSベースでコンテナを外部に公開するためのエンドポイント(FQDN)を提供するオブジェクト。レセコン (ORCA) のクライアントからサーバーに接続する際のエンドポイントを提供するために使用。

# レセコン（ORCA）仕様

- OS：  
Ubuntu 16.04 LTS
- データベース：  
PostgreSQL 9.5
- パッケージ管理システム(apt)  
からインストール  
(ORCAインストールの際に  
PostgreSQLも一緒に  
インストールされる)



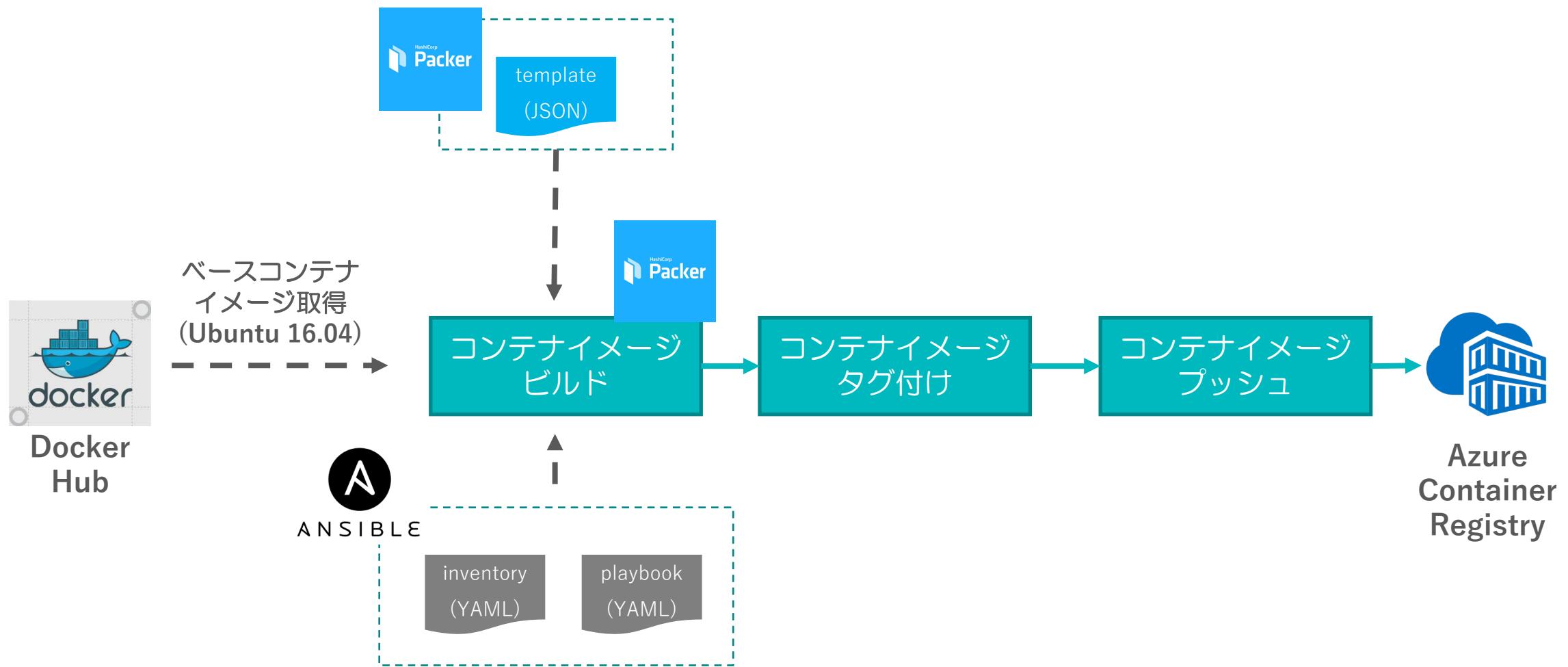
## コンテナ化する際の制約事項

データベースのような、永続化が必要なデータストアは、マネージドサービスに配置するのが定石であるが、今回は、より単体ORCAに近い構成で、スピード感を持って、コンテナ化をすることが必要。

## レセコン（ORCA）：コンテナイメージ作成

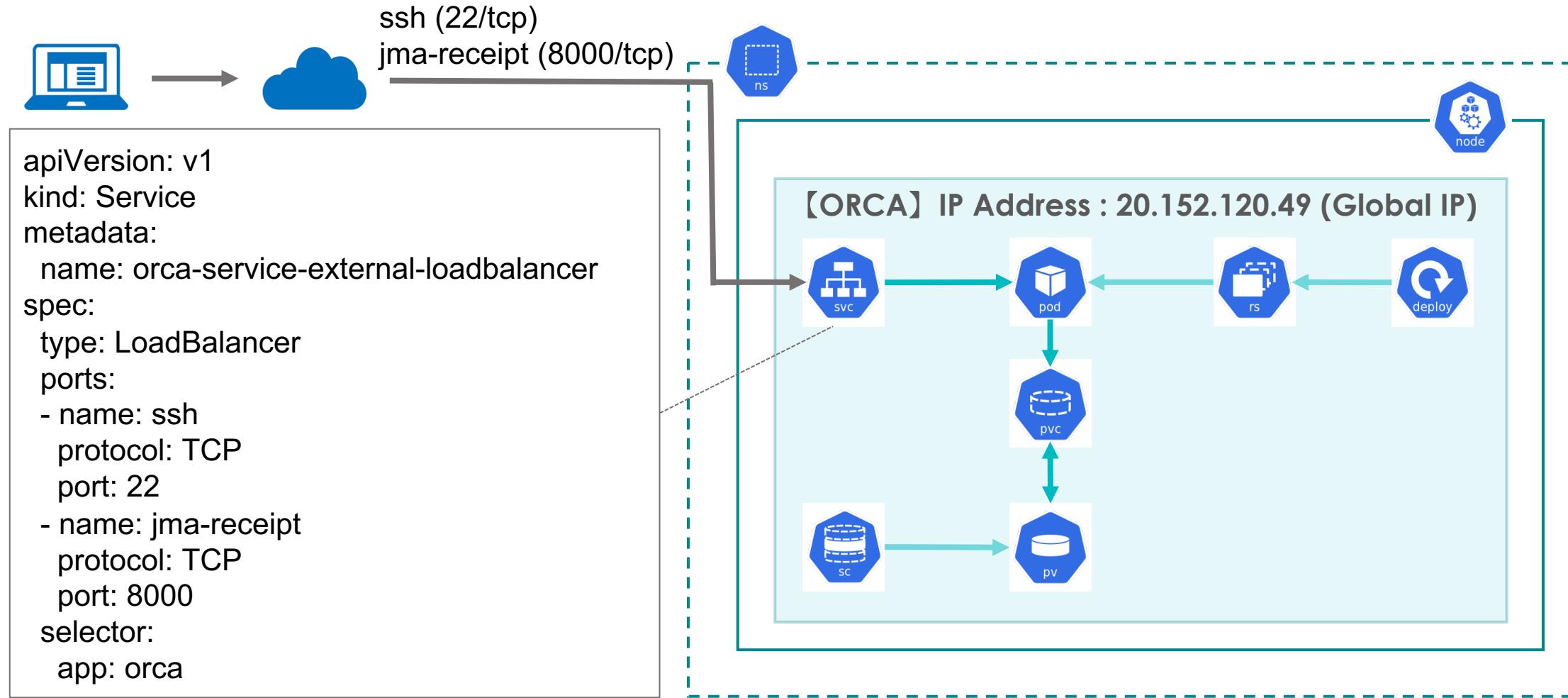
- Dockerのコンテナイメージを作成する場合、Dockerfileを使うのが定石。
- 今回はPacker + AnsibleでDockerのコンテナイメージを作成。  
→ きりんカルテシステムでは**DevOps推進**の一つとしてインフラ構築のコード化を行っている。 レセコン（ORCA）を構築するためのAnsibleの定義ファイル(yaml)があるため、これを有効活用するために Packer + Ansibleを使用。

# レセコン（ORCA）：コンテナイメージ作成



# レセコン（ORCA）：アーキテクチャ検証（ステップ1）

- グローバルなIPアドレスを使って、レセコン（ORCA）のクライアントからサーバーに接続する。

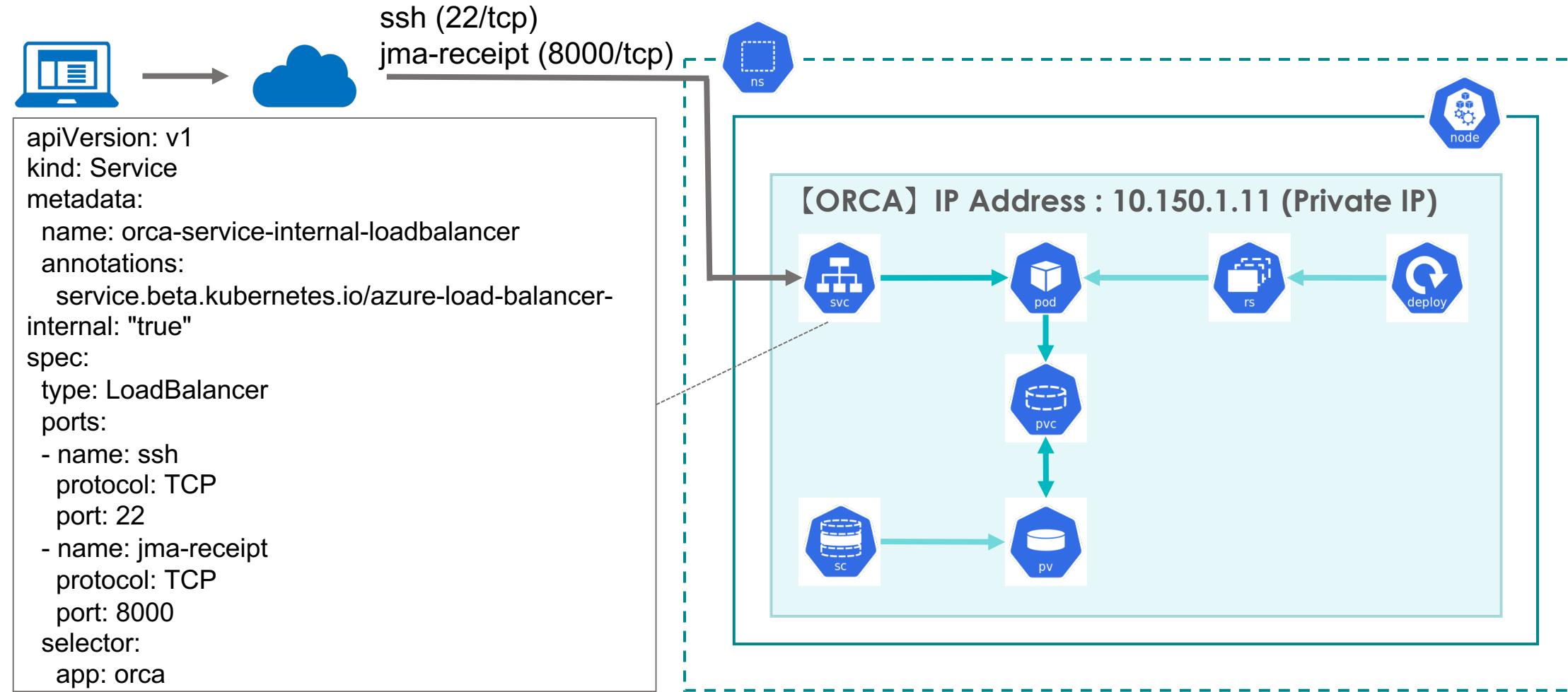


## レセコン（ORCA）：アーキテクチャ検証（ステップ1）

- グローバルIPでORCAのコンテナ(Pod)に接続するために、**外部ポート**バランサー型のServiceを導入。
- ORCAのコンテナ(Pod)が消失・再生成されてもこれまでに登録したデータが消失しないよう、**Persistent Volume**を導入。

# レセコン (ORCA) : アーキテクチャ検証 (ステップ2)

- プライベートなIPアドレスを使ってVPN経由でレセコン (ORCA) のクライアントからサーバに接続する。

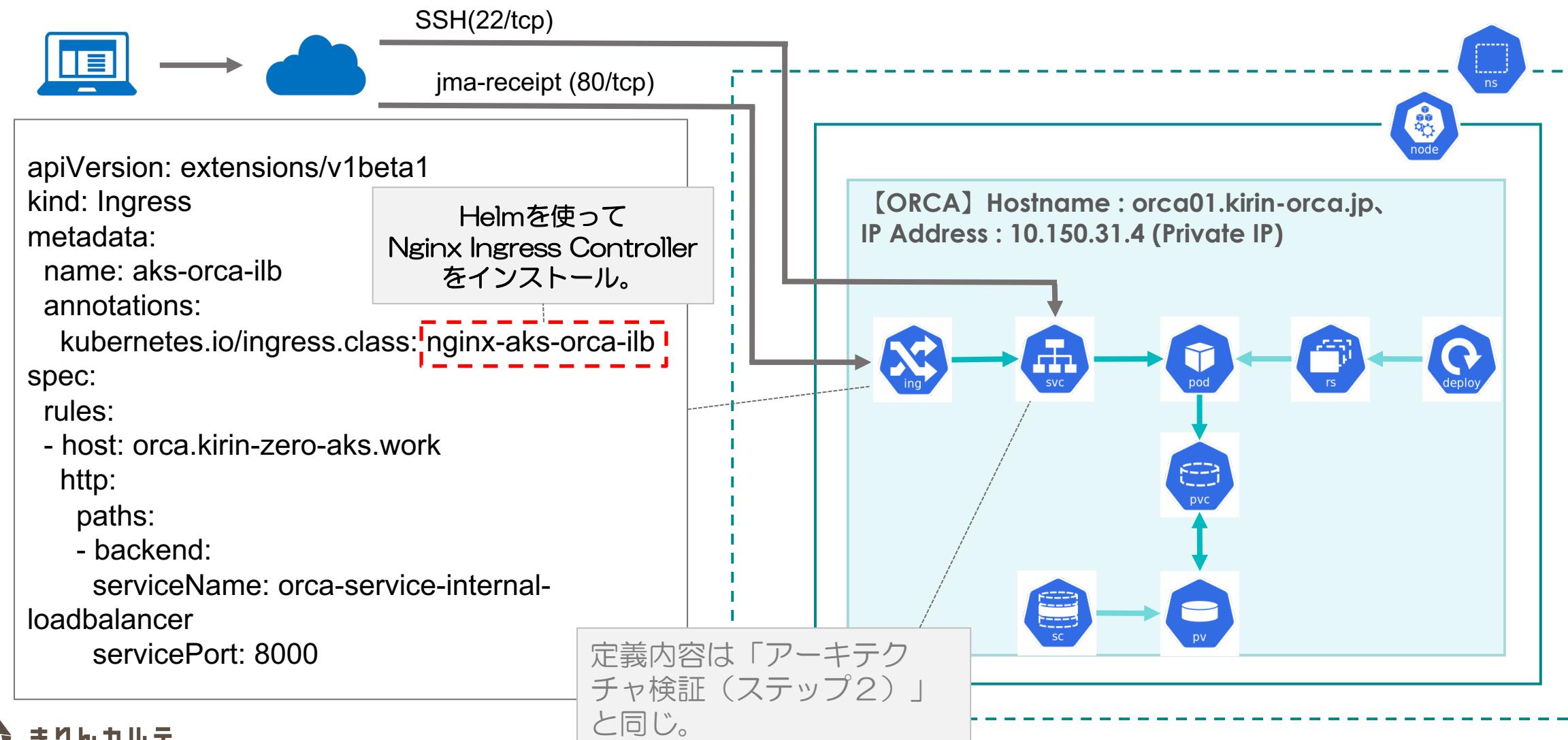


## レセコン（ORCA）：アーキテクチャ検証（ステップ2）

- プライベートIPでORCAのコンテナ(Pod)に接続するために、Azure Internal Load Balancerを使った内部ロードバランサー型のServiceを導入。
- ORCAのコンテナ(Pod)が消失・再生成されてもこれまでに登録したデータが消失しないよう、Persistent Volumeを導入。

# レセコン (ORCA) : アーキテクチャ検証 (ステップ3)

- FQDNを使ってVPN経由でレセコン (ORCA) のクライアントからサーバーに接続する。



## レセコン（ORCA）：アーキテクチャ検証（ステップ3）

- FQDNでORCAのコンテナ(Pod)に接続するために、**Ingress**を導入。
- Ingress Controllerに **Nginx Ingress Controller**を採用。
- Ingressの接続先となるServiceはAzure Internal Load Balancerを使って導入。
- ORCAのコンテナ(Pod)が消失・再生成されてもこれまでに登録したデータが消失しないよう、Persistent Volumeを導入。

## Azure データストアの種類



### Managed Disk

- ・直接ノードへ接続されるディスク  
(iSCSIで接続されたディスクのようなもの)
- ・仮想マシンのサイズによって、vCPUコア数、メモリー容量と同じく、接続できるディスク数に上限がある
- ・IOPS保証のある 高速なSSD を選択可能



### Azure Files

- ・SMBプロトコルで接続できる共有フォルダー  
(巨大なファイルサーバーに、ユーザー毎に、複数の共有フォルダーをサービスしているようなもの)
  - ・一つの共有フォルダー毎に、60MB/sの速度上限、5TBの容量上限があり。
  - ・smb3.0をサポートするkernelが必要（例：CentOS7, Ubuntu16.04）



## Azure Managed Disk (ORCAコンテナ起動成功)

- ✓ Storage Classの定義が不要のため、導入がしやすい。
- ✓ AKSクラスター内の1ノードに構築可能なORCAコンテナの数がそのノードに割り当て可能なデータディスクの数になってしまう。



## Azure Files (ORCAコンテナ起動失敗)

- ✓ Storage Classの定義が必要。
- ✓ Storage Classの定義時にmountOptionsの設定が必要。(ディレクトリ・ファイルのパーミッション、uid・gid)
- ✓ レセコン（ORCA）で使用しているPostgreSQLが起動できない。(起動時にPostgreSQLのデータディレクトリ内でシンボリックリンクを作成しようとしているため。)

Q&A time...

ご清聴ありがとうございました。

お困りのことがあればお気軽にご連絡ください。

<https://xirapha.jp>



きりんカルテ