

OSS のコンテナオーケストレーションツール Kubernetes のご紹介

オープンソースカンファレンス 2020 Osaka
2020/01/24

SRA OSS, Inc. 日本支社
赤松 俊弘

自己紹介

- ✓ 赤松 俊弘 (Toshihiro Akamatsu)
- ✓ SRA OSS, Inc. 日本支社
OSS 事業本部 基盤技術グループ



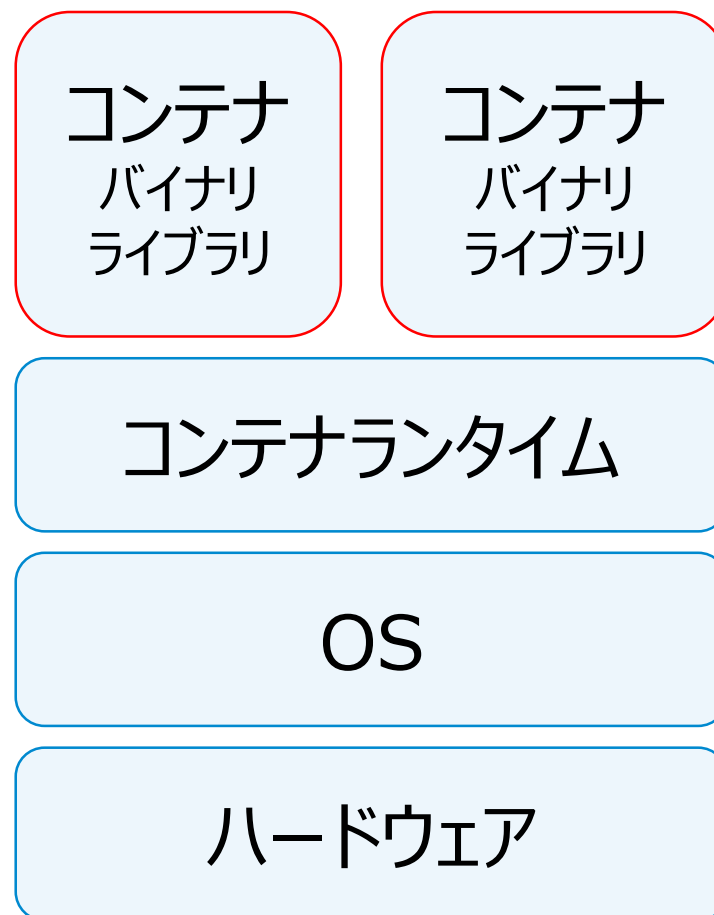
職務

- ✓ PostgreSQL 以外の OSS 全般の技術サポート、構築
- ✓ 主にインフラ構築・監視システム構築を担当

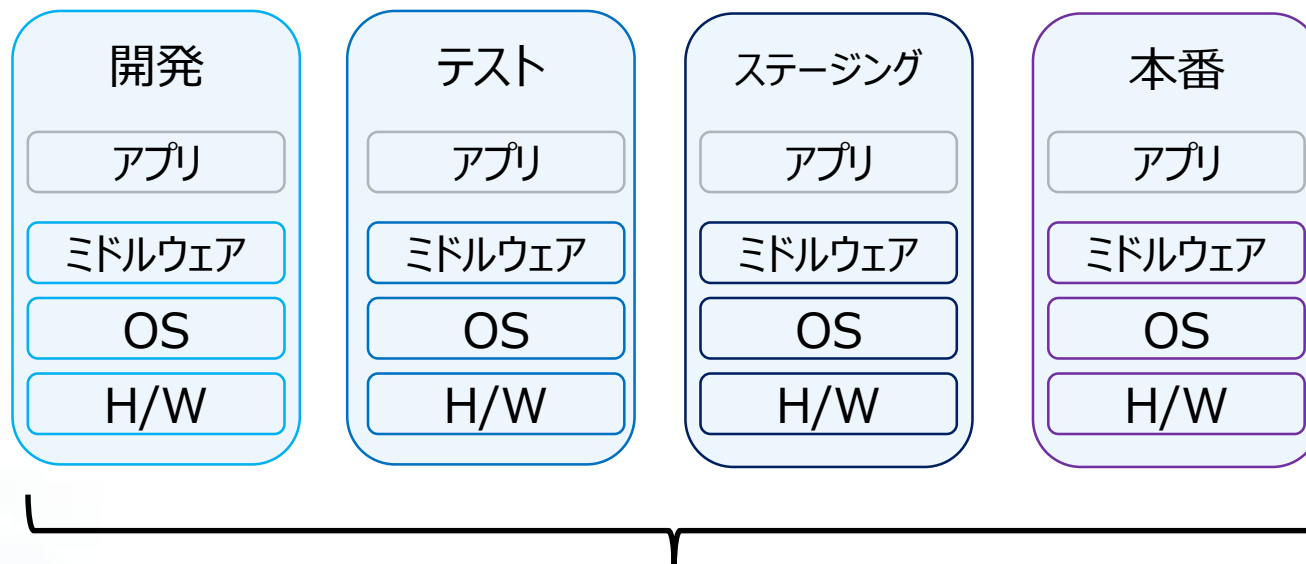


はじめに

- ホスト OS 上に論理的な区画（コンテナ）を作ってアプリケーションを動作させるために必要なプログラムなどをまとめたもの
- ホスト OS 上のリソースをコンテナで共有して利用
- サーバ仮想化に対してオーバヘッドが少ない
- 複数のコンテナを組み合わせると一つのアプリケーションを作るマイクロサービスとの相性がよい

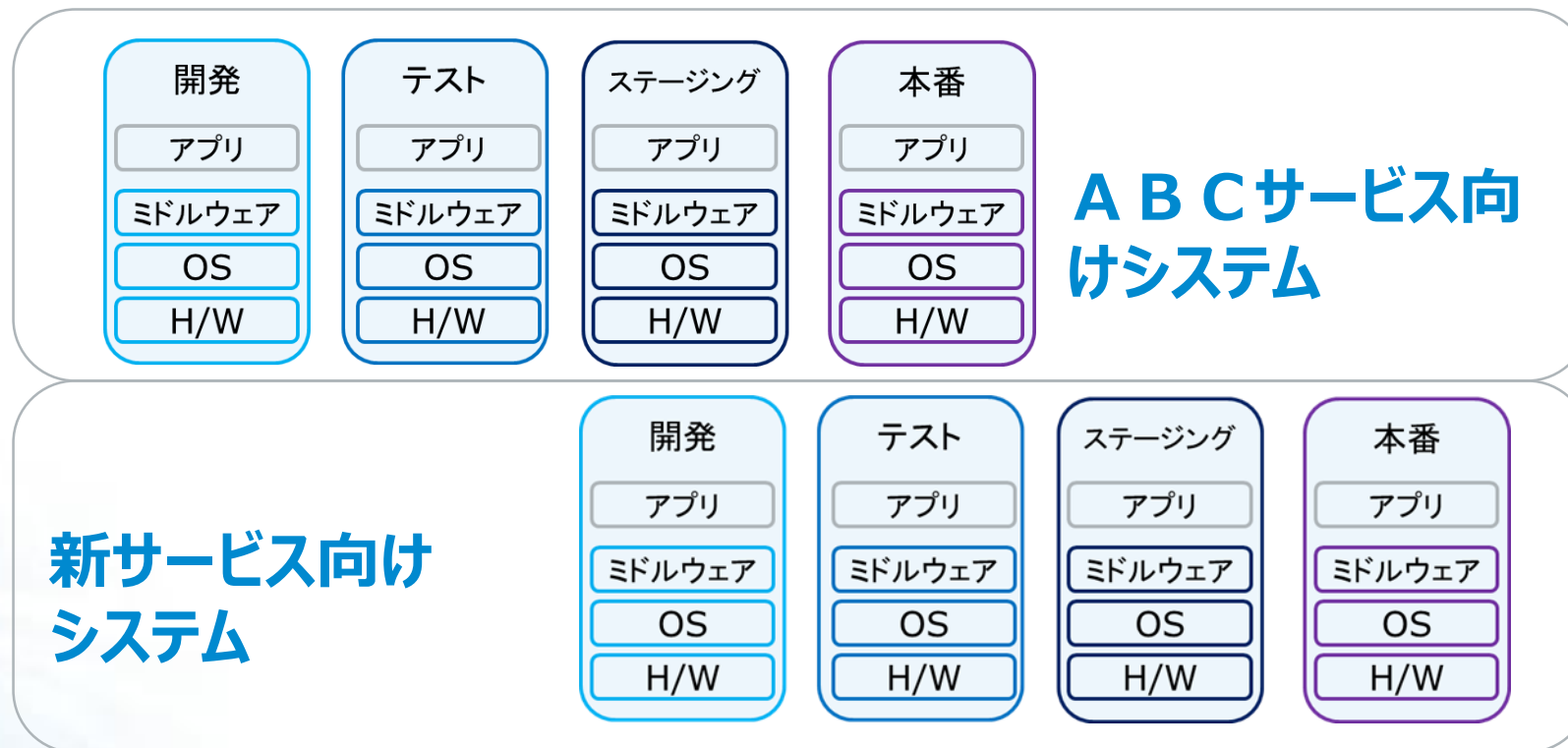


- 開発環境・テスト環境・ステージング環境・本番環境など、複数の環境でインフラ・ミドルウェアの違いに依存する問題が発生しやすい



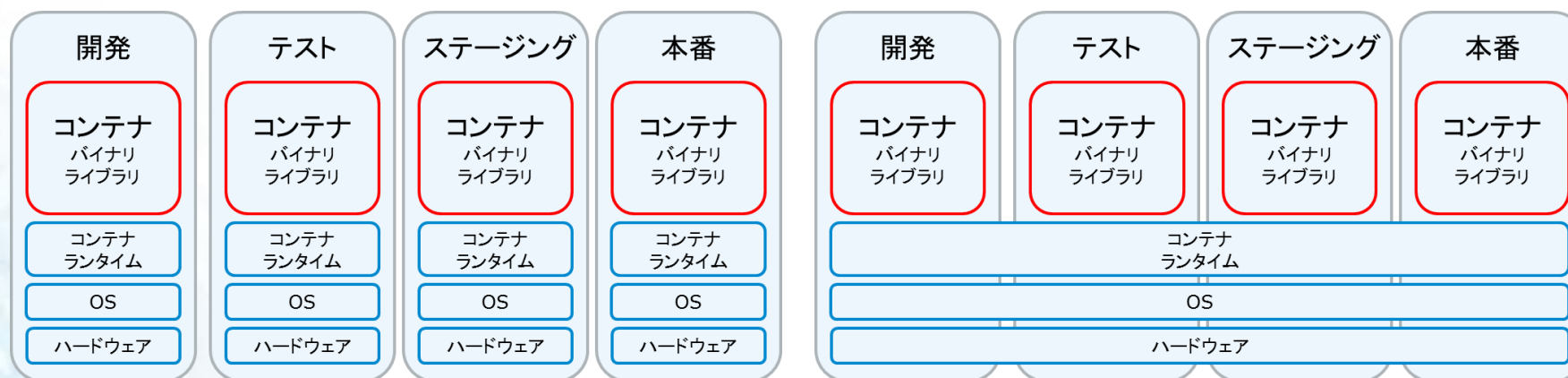
同じ環境??

- サービスが増えると開発環境・テスト環境・ステージング環境・本番環境も増える・・



**インフラの準備時間とテスト実施のための時間がかかって
しまって迅速な開発・サービス提供ができなくなってしまう**

- コンテナランタイム上で同じイメージを動かすことで環境の違いのリスクを減らすことができる
- コンテナ間は独立しているので、同じホスト上でも動作させることができる
- サーバ仮想化に対してオーバヘッドが少ないため、サーバの利用効率が上昇する



- コンテナの構成ファイルをGitリポジトリで管理・共有できる
- コンテナのイメージをリポジトリで管理・共有できる
- 継続的インティグレーション／継続的デリバリー（CI・CD）を実施しやすく、アプリケーション開発が効率的にできる

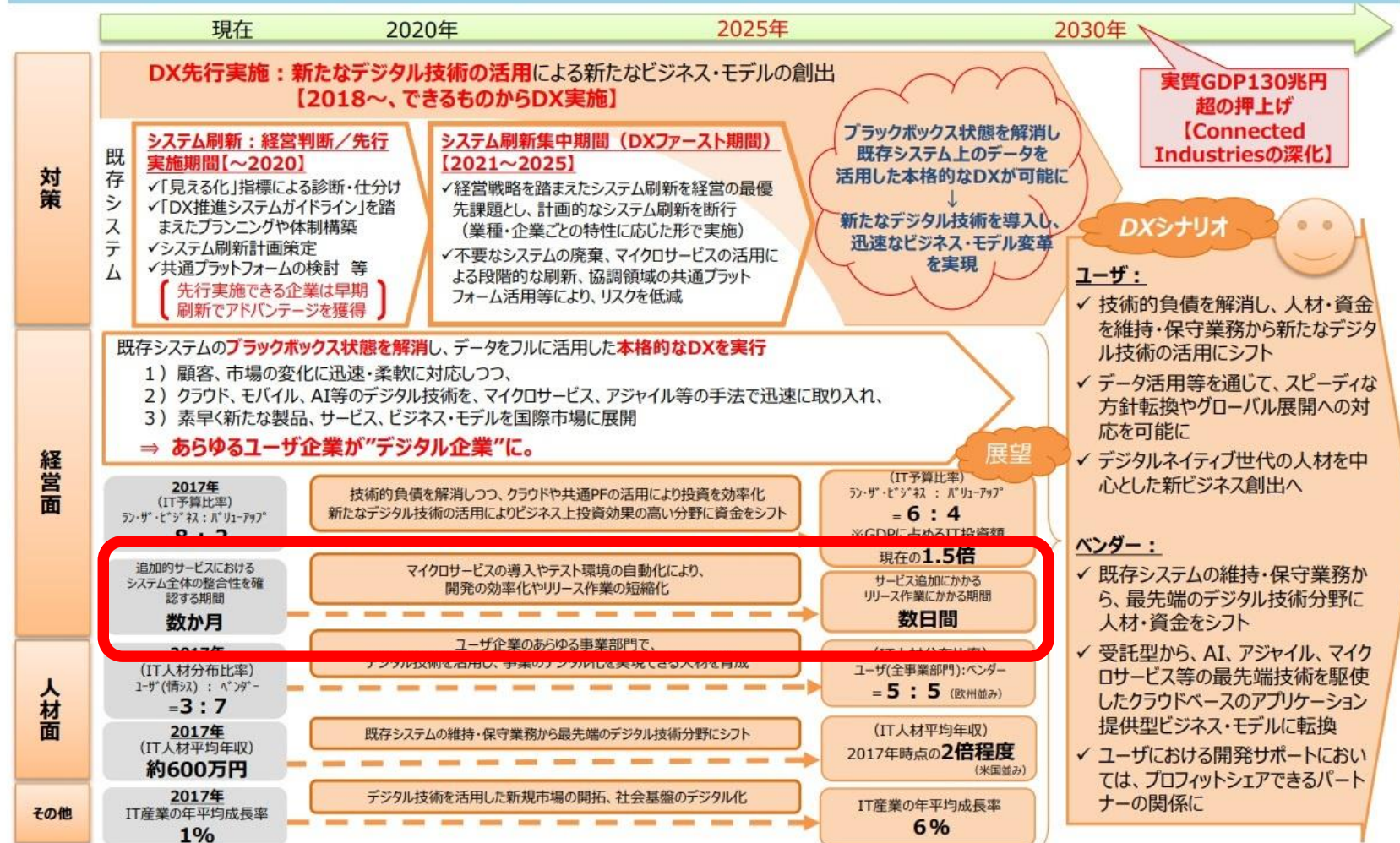
CI/CD

アプリケーションソースコードの変更がされたら即座にコンテナを作成して、テスト環境でテストして本番環境にリリース可能な状況にしておく

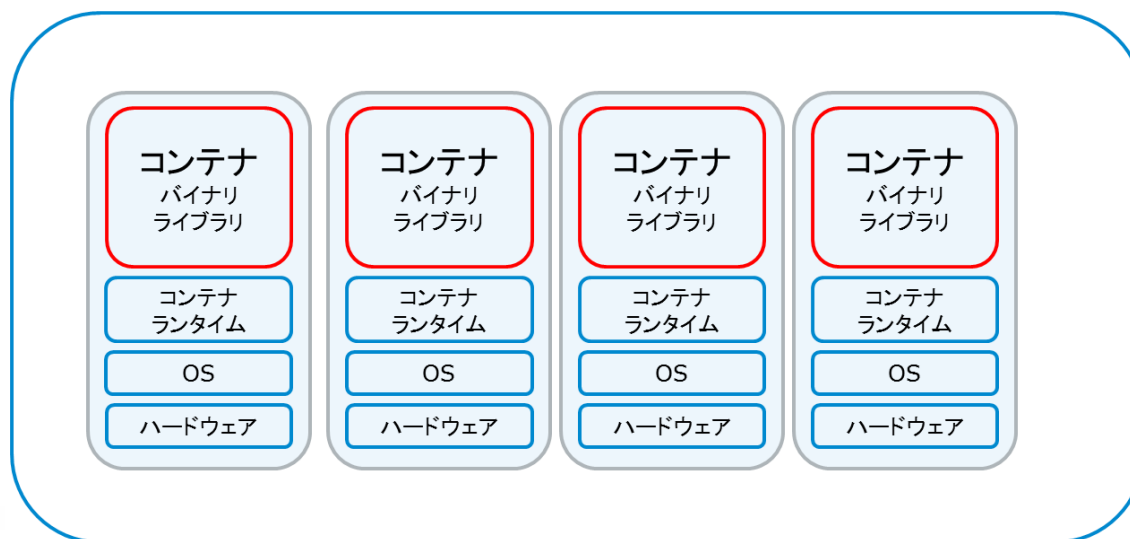
生産性の向上や環境管理の負荷軽減が期待できる

DX実現シナリオ

【DXシナリオ】2025年までの間に、**複雑化・ブラックボックス化した既存システムについて、廃棄や塩漬けにするもの等を仕分けしながら、必要なものについて刷新しつつ、DXを実現することにより、2030年実質GDP130兆円超の押し上げを実現。**



- コンテナを動かす際には、可用性・性能を考慮して複数のホストマシンから構成されるクラスタが必要



コンテナを自律制御・管理するための クラスタプラットフォーム



Kubernetes とは

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

What is Kubernetes.

Production-Grade Container Orchestration – Kubernetes.

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Kubernetes は、**宣言的な構成管理と自動化**を促進し、**コンテナ化**されたワークロードやサービスを管理するための、**ポータブルで拡張性のある**オープンソースプラットフォーム

Kubernetesとは何か？

Production-Grade Container Orchestration – Kubernetes.

<https://kubernetes.io/ja/docs/concepts/overview/what-is-kubernetes/>

- Google での 10 年以上の運用経験をもとに設計されたシステムをオープンソース化 (2014~)
- Cloud Native Computing Foundation (CNCF) により開発・メンテナンス
 - ベンダーフリー・認定システム間で互換性が高い
 - Kubernetes を中心としたエコシステムの整備
 - Prometheus - fluentd
- 大規模クラスタをサポート



<https://www.cncf.io/certification/software-conformance/>

<https://kubernetes.io/docs/setup/best-practices/cluster-large/>

項目	サポート対象
ノード数	5000台以下
Pod数	150,000以下
コンテナ数	300,000以下
1ノードあたりのpod数	100以下

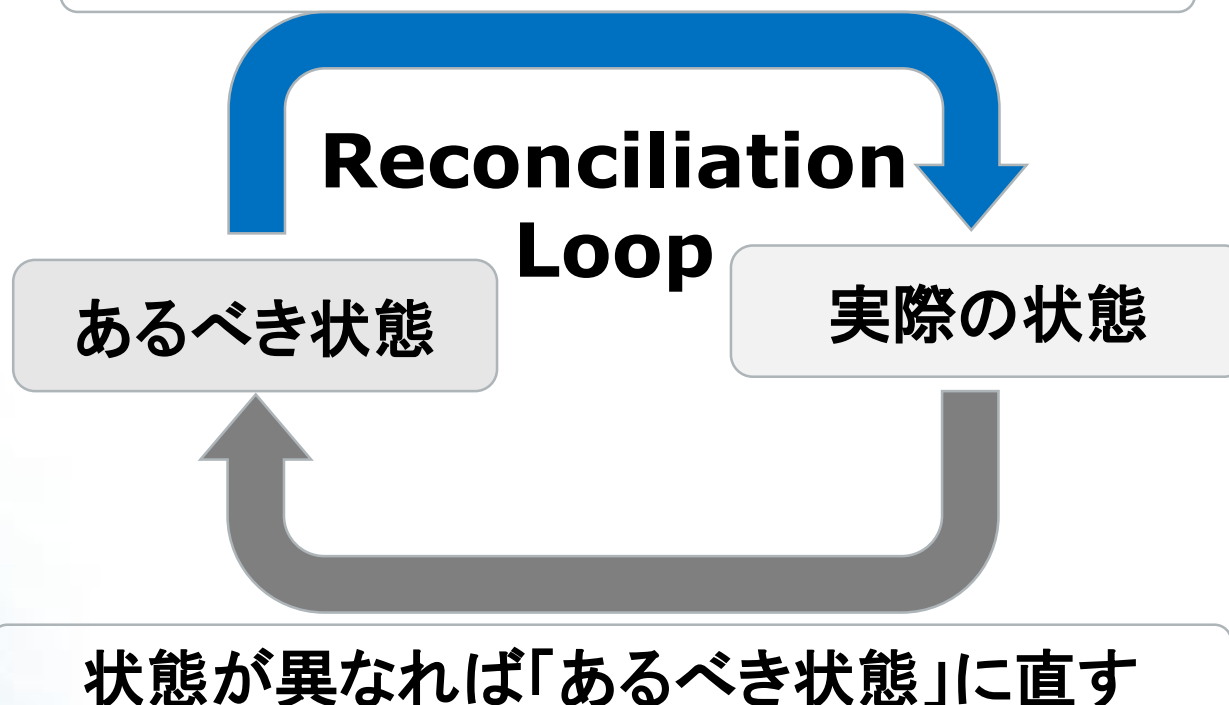
- Kubernetes X.Y.Z
 - X メジャーバージョン
 - Y マイナーバージョン
 - Z パッチバージョン
 - 最新 1.17.2 (2020/01/24 時点)
- マイナーバージョンのリリースサイクル 約 3 カ月
- 3 つのマイナーバージョンをメンテナンス

**一つのマイナーバージョンは
約 9 か月間のみメンテナンスされる**

<https://kubernetes.io/docs/setup/release/version-skew-policy/>

- ユーザが宣言的に「あるべき状態」を定義
- Kubernetes は「あるべき状態」になるように「自律的」に働く

CHECK あるべき状態 = 実際の状態



- 宣言 1 : Nginx を 3 つ動かす
 - 1 つの Nginx に障害が発生して、合計 2 つしか動いていない場合には新規で 1 つ起動
 - ⇒ 障害の自動修復
- 宣言2: Nginx の平均 CPU 利用率が 50% になるように Nginx を自動的に増やす
 - ⇒ 水平オートスケール
- 宣言3: ノードのリソースが不足したらノードを追加
 - ⇒ ノードの水平オートスケール


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-master
  labels:
    app: redis
spec:
  selector:
    matchLabels:
      app: redis
      role: master
      tier: backend
```

replicas: 1

```
template:
  metadata:
    labels:
      app: redis
      role: master
      tier: backend
```

```
spec:
  containers:
    - name: master
```

image: redis:5.0.4

```
resources:
  requests:
```

cpu: 100m

memory: 100Mi

```
ports:
  - containerPort: 6379
```

構成をファイルで定義できる
Github などで管理しやすい

複製数の指定

コンテナイメージの指定

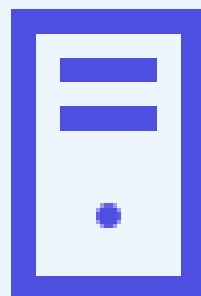
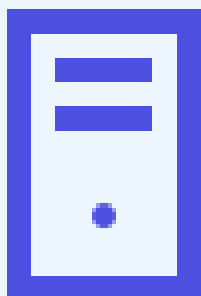
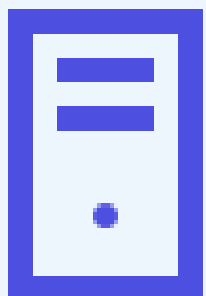
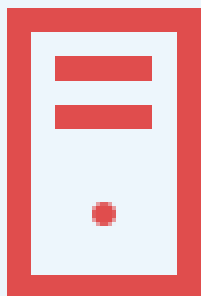
H/Wリソースの設定



Kubernetes アーキテクチャ

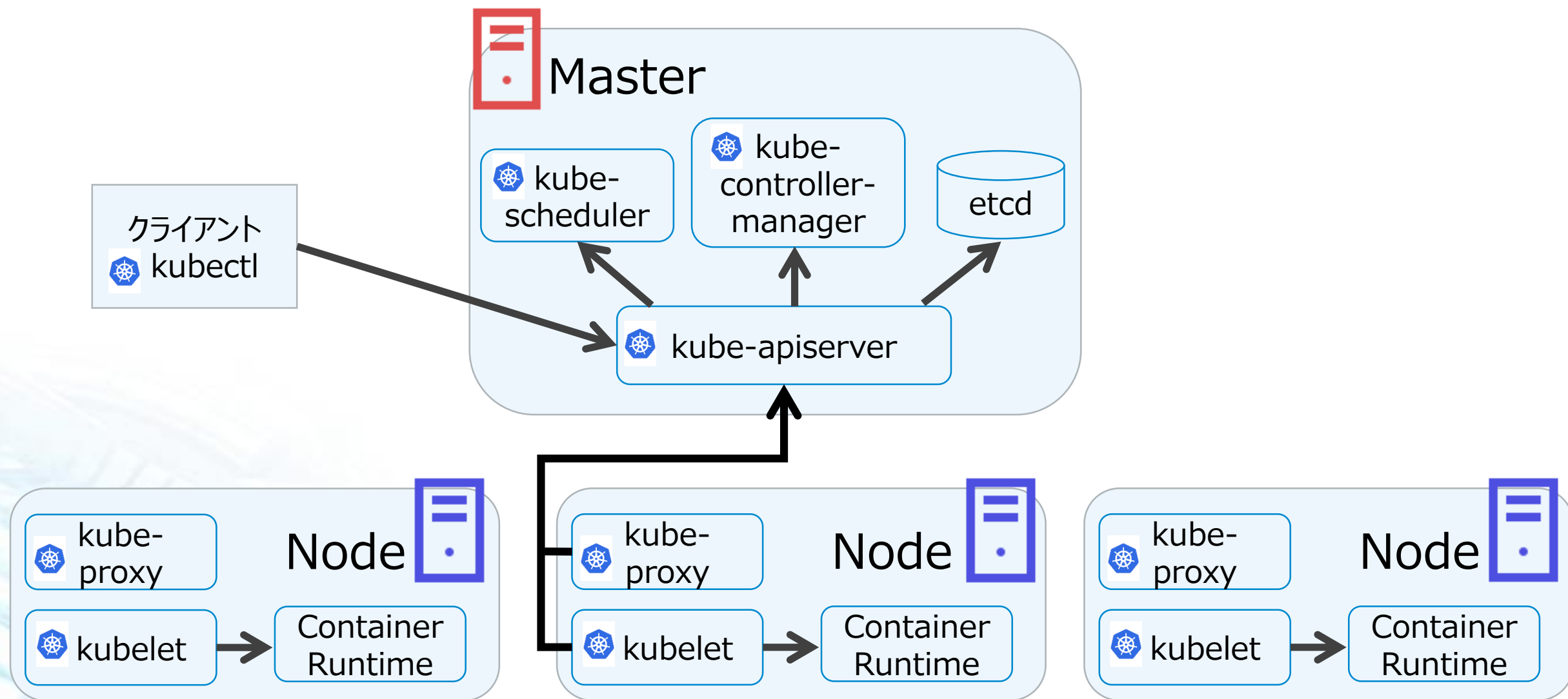


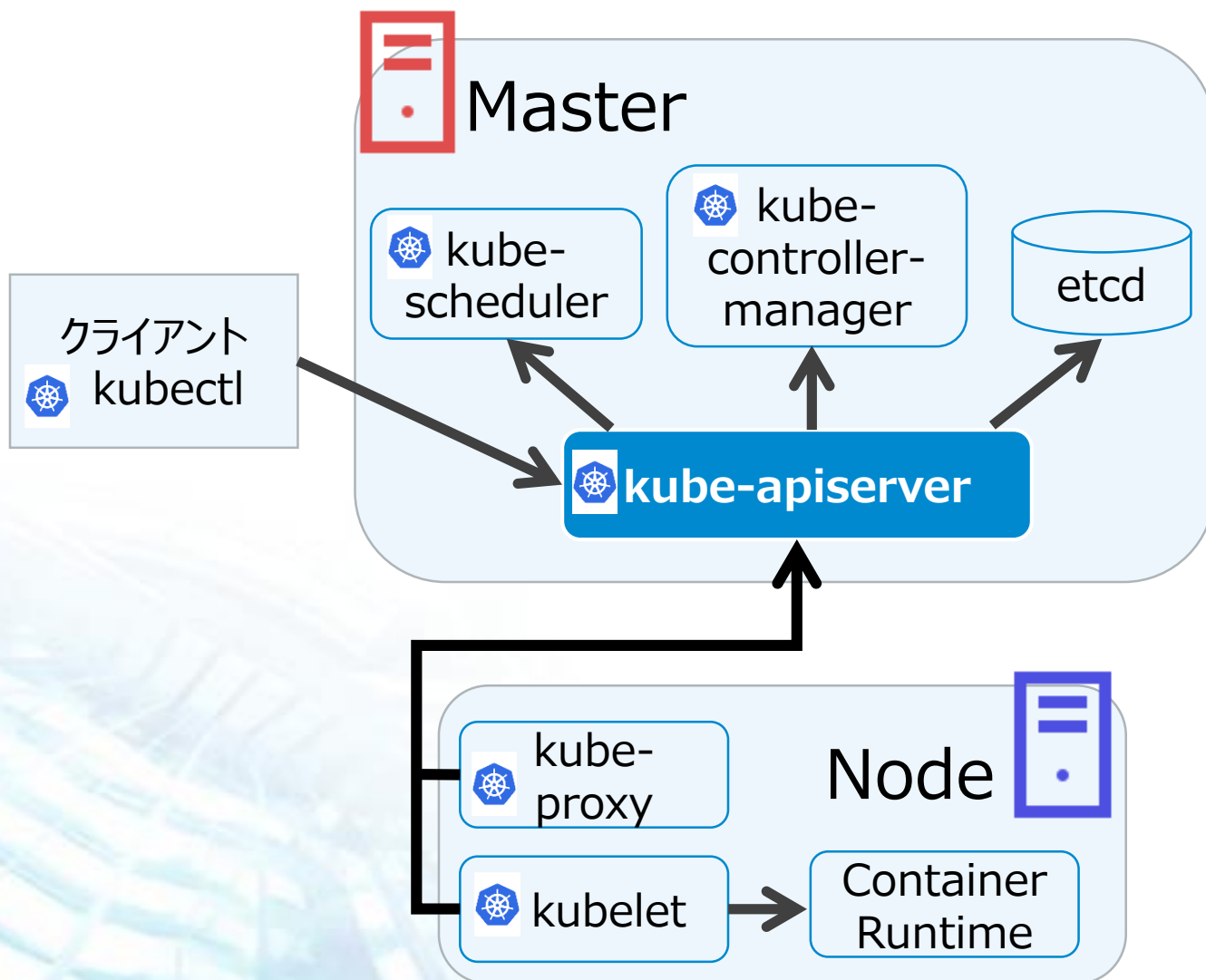
Kubernetes クラスタ



Master
クラスタの管理

Node
アプリケーションのホスト



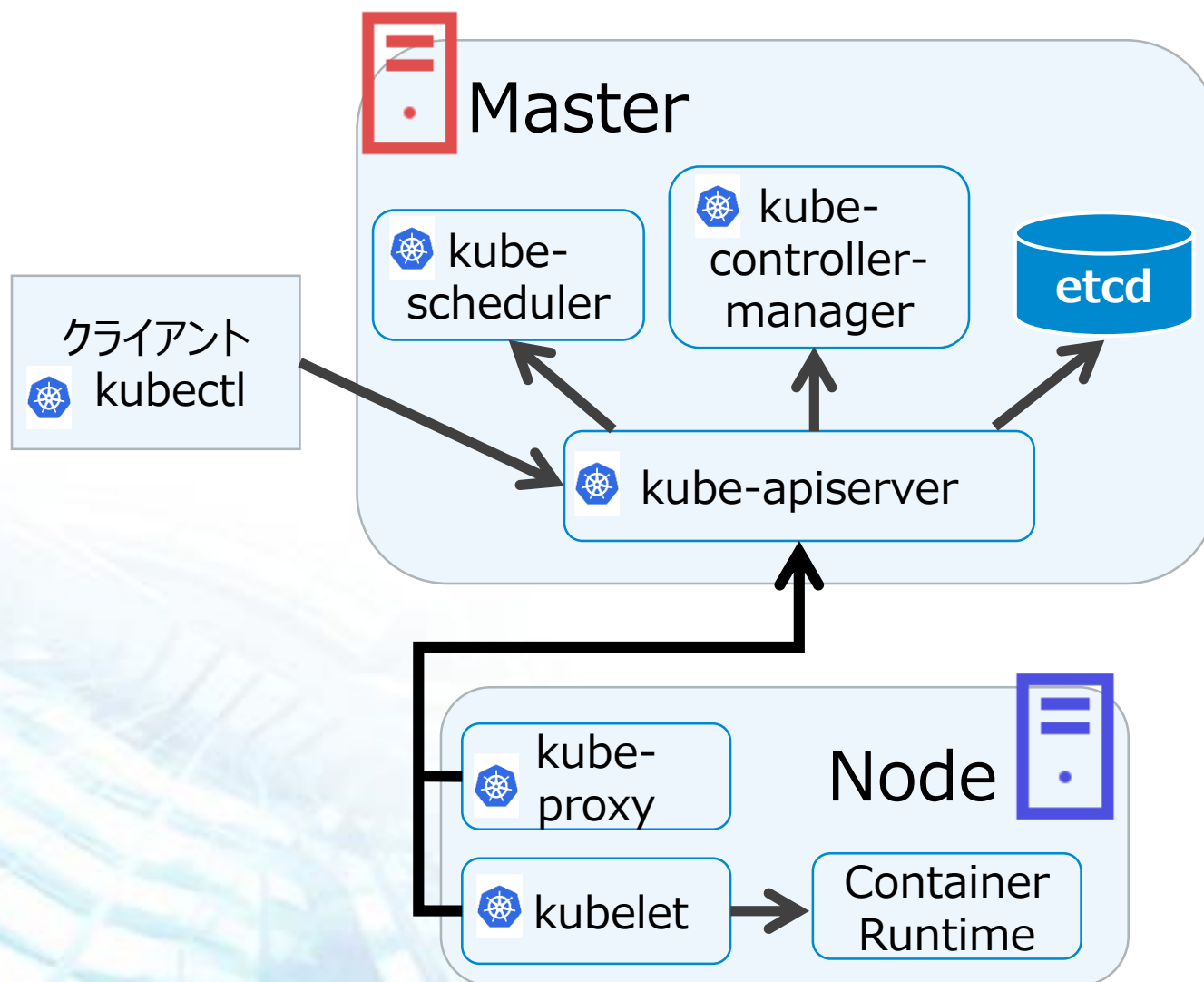


クラスタの中心かつ全体の
フロントエンド

全てのコンポーネントが
Kube-apiserver を通して動作

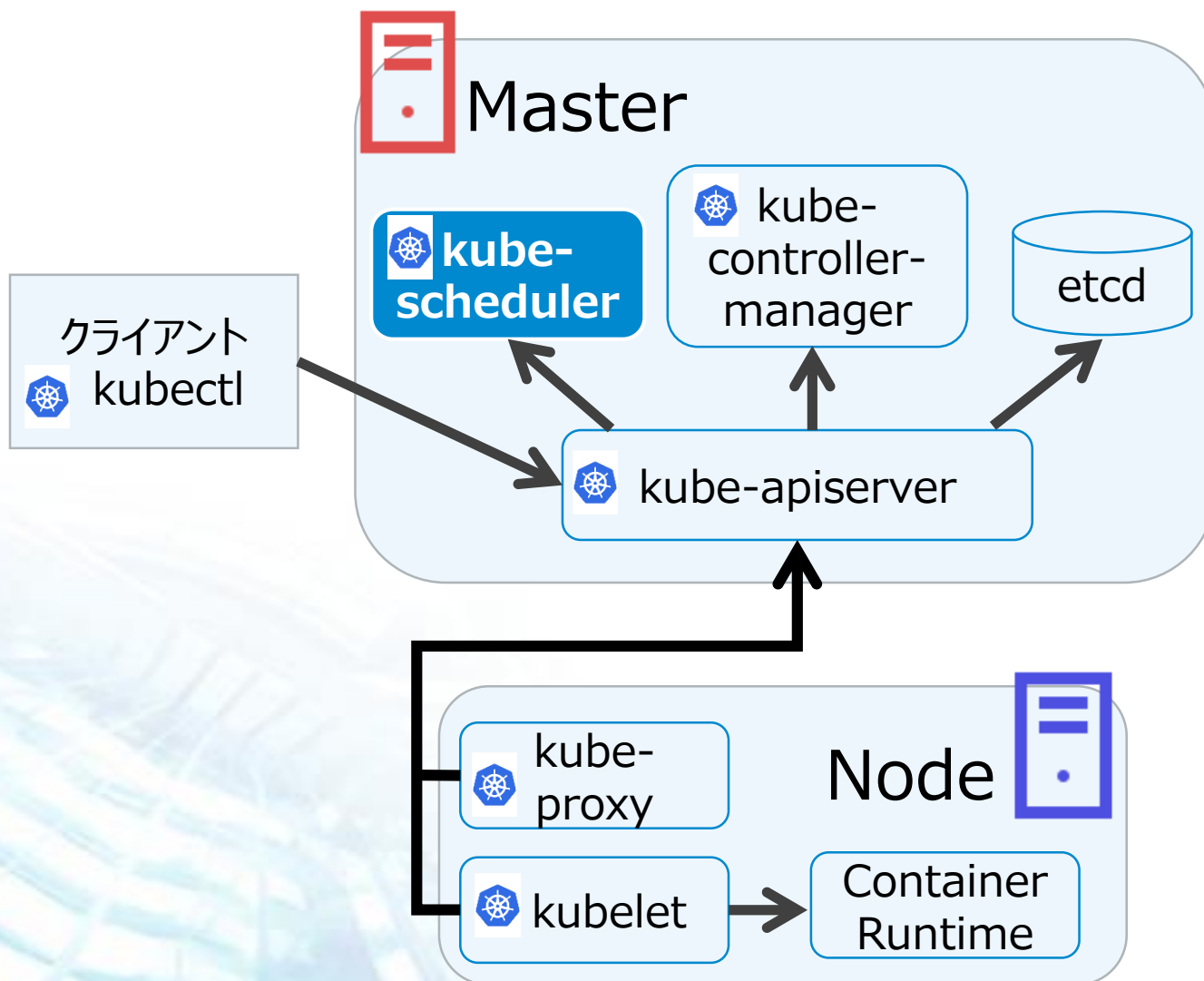
Kubernetes API を外部に提供

etcd へは kube-apiserver
からのみ通信可能



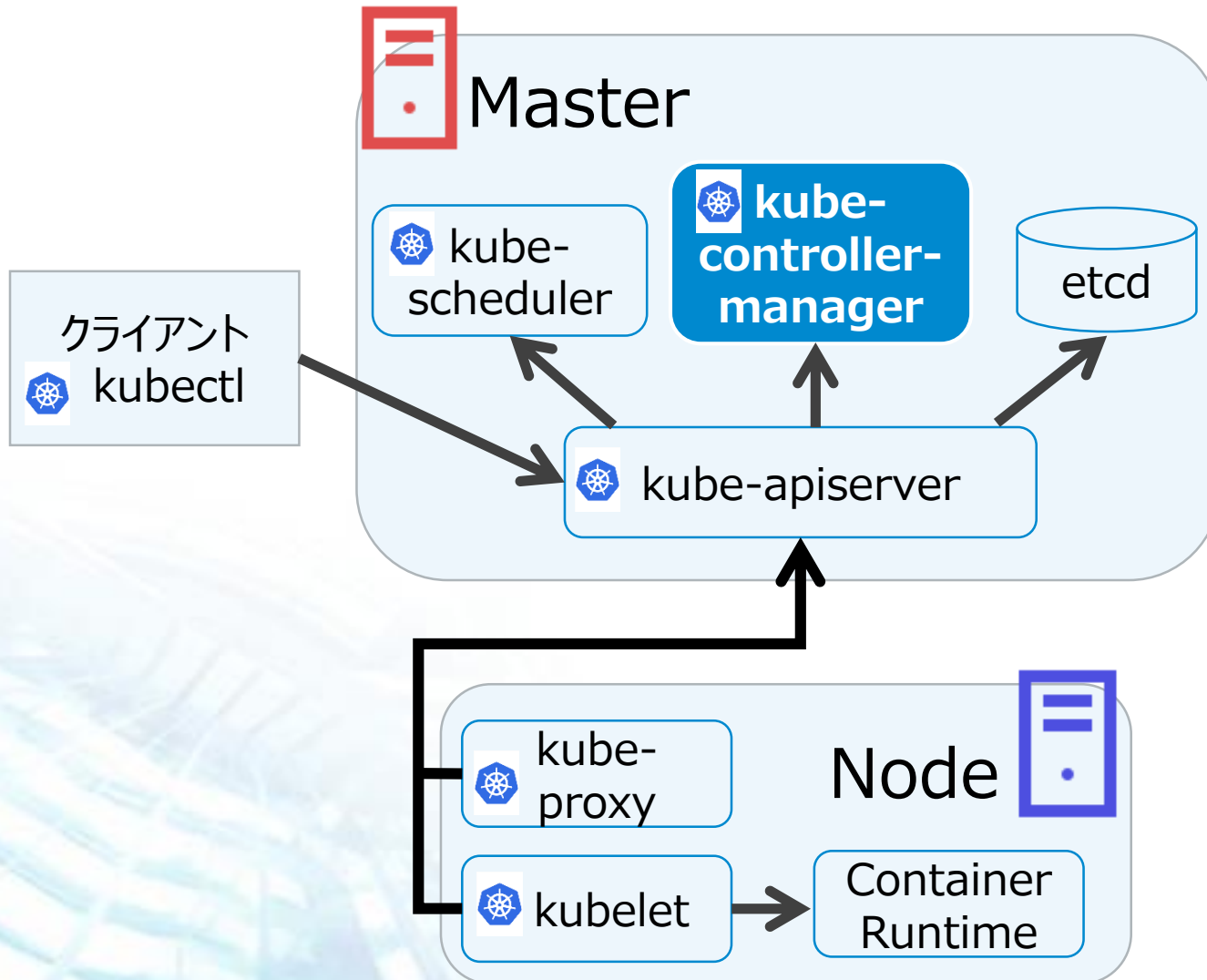
一貫性、高可用性をもつ
キーバリューストア

Kubernetes の全ての情報の
保存先



Pod を配置する Node を決定

配置に関する制約や条件を確認

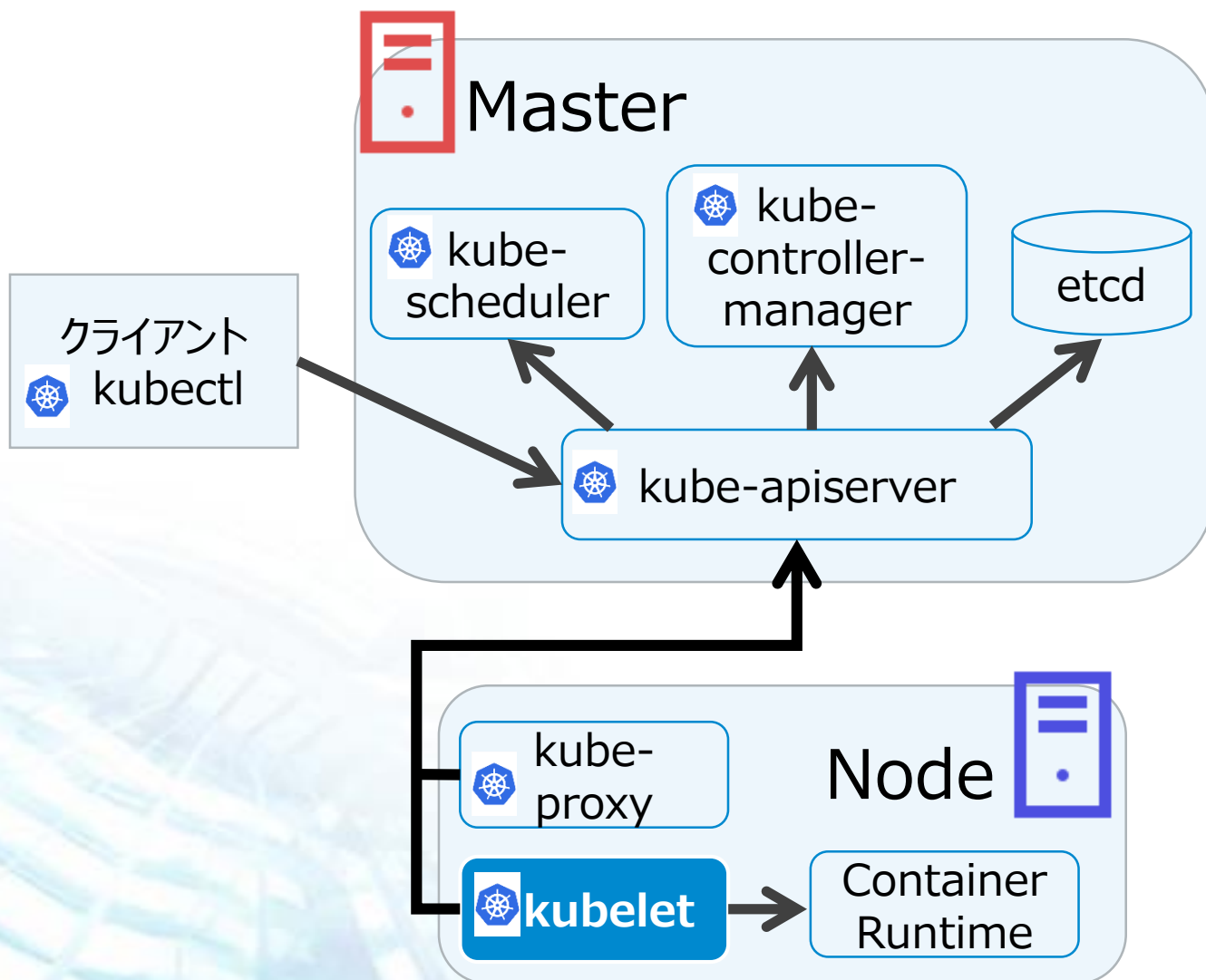


現在の状態が「あるべき状態」に合致するように動作

ノードコントローラ
ノードダウンの場合の通知と対応

レプリケーションコントローラ
Pod の起動数を正常に保つ

エンドポイントコントローラ
エンドポイントの注入

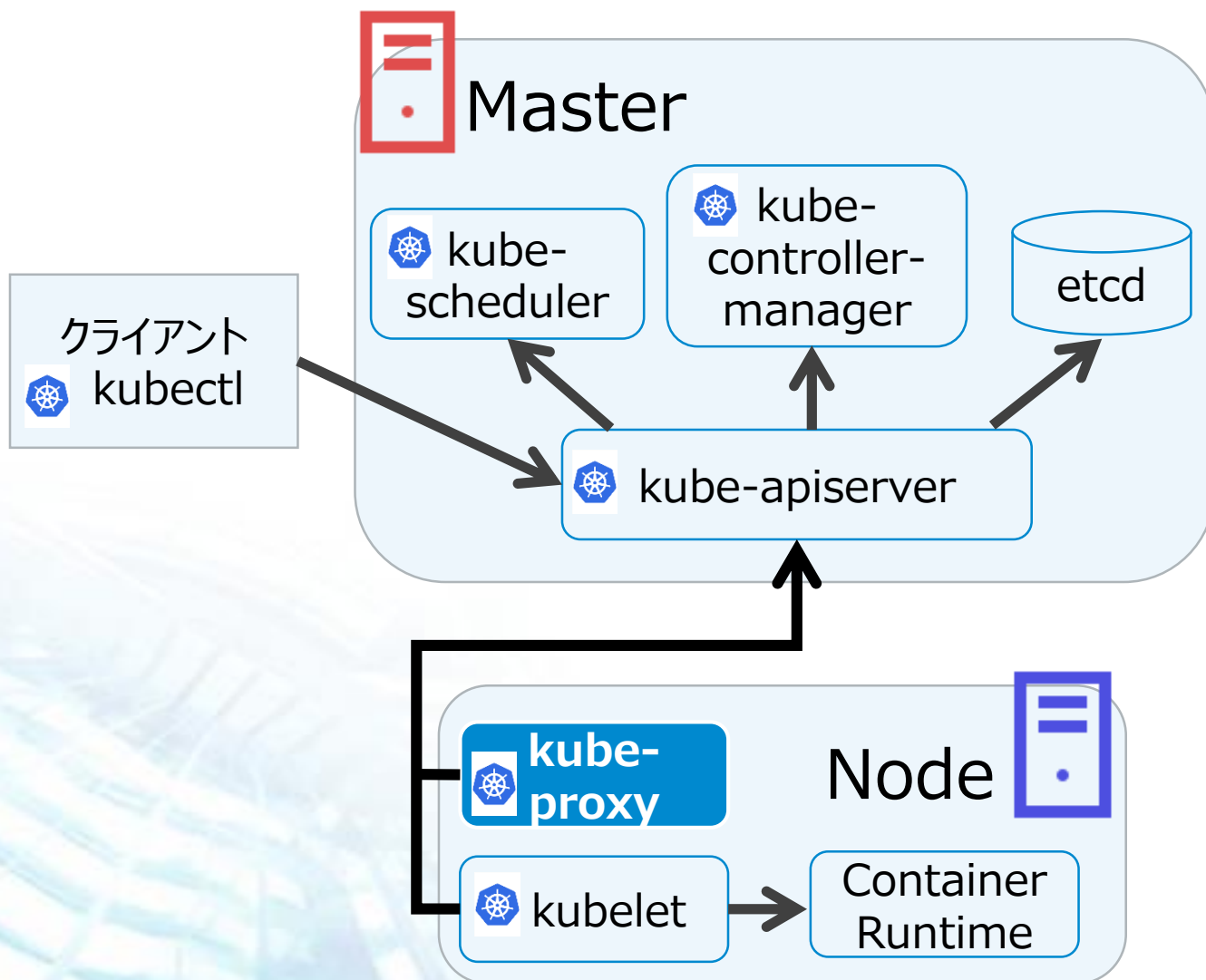


各 Node で実行されるエージェント

Pod の定義を取得し、コンテナが正常に実行されていることを保証

コンテナエンジンにリクエストを受け渡す

クラスタに Pod と Node の状態を報告

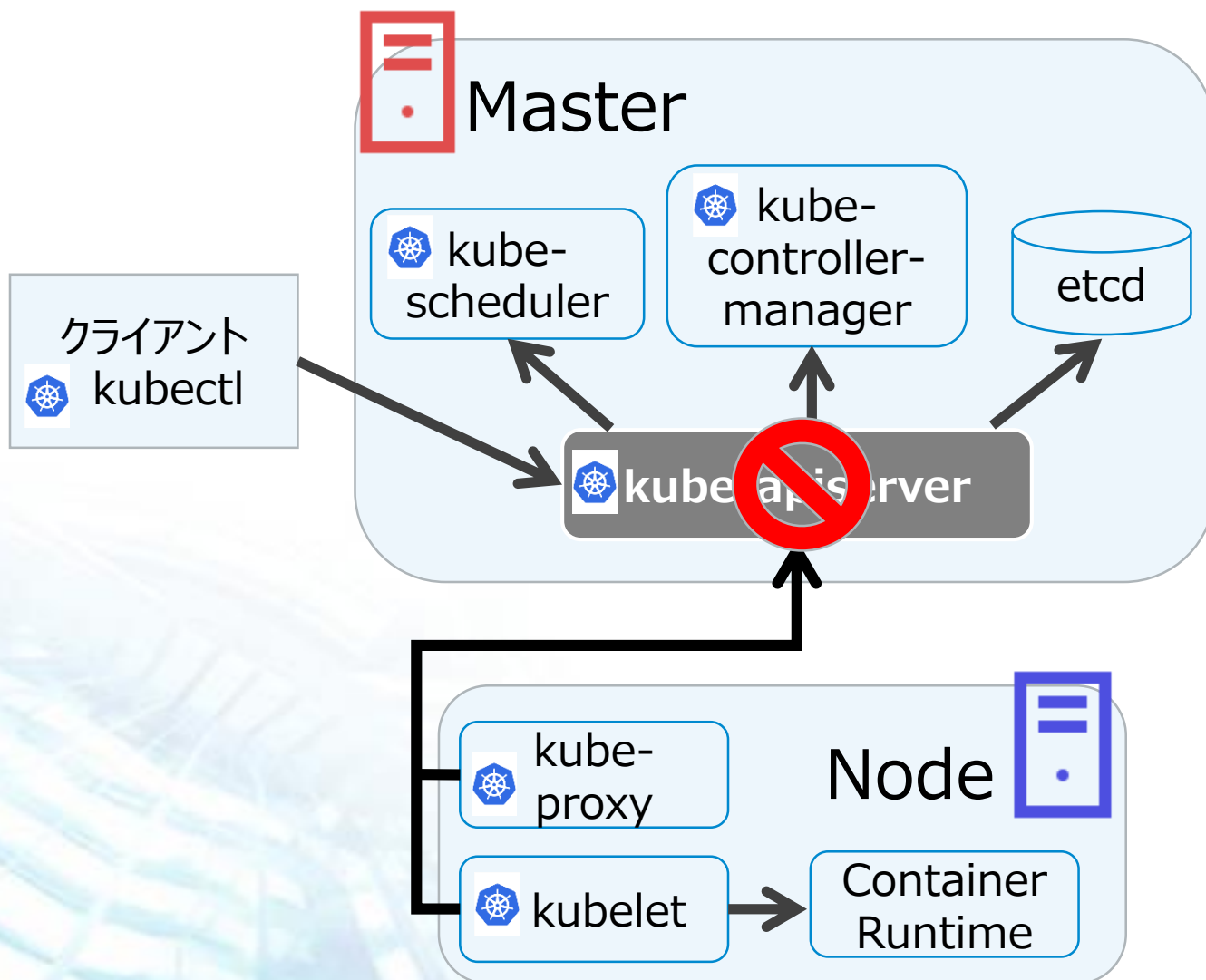


各 Node で動作する
ネットワークプロキシ

Node のネットワークルールを
メンテナンス

クラスタ内外から Pod への
通信を可能に

ネットワークルールは
iptables などを利用



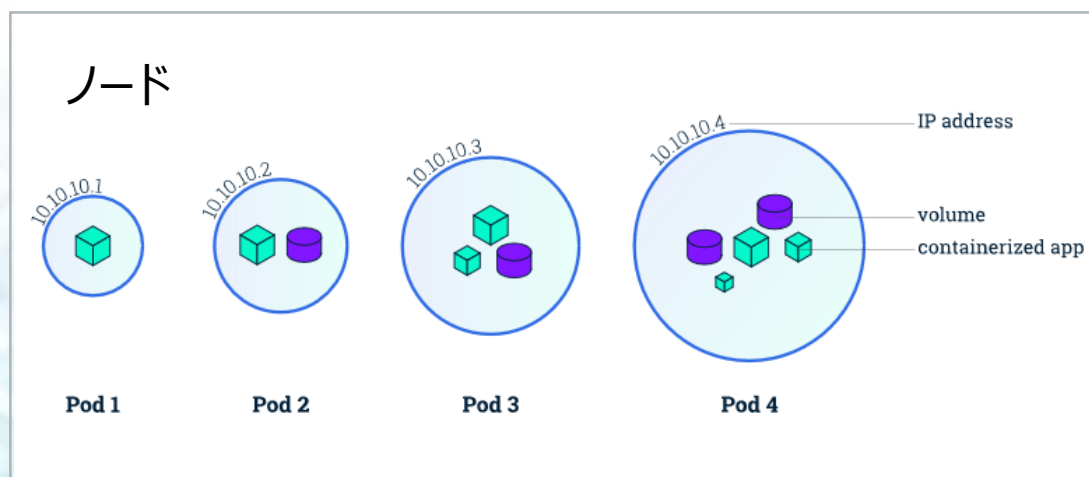
部分的な障害がサービス停止にならないようなアーキテクチャ

他のコンポーネントに接続ができなくても、各コンポーネントは直近の内容をもとに動作を続ける

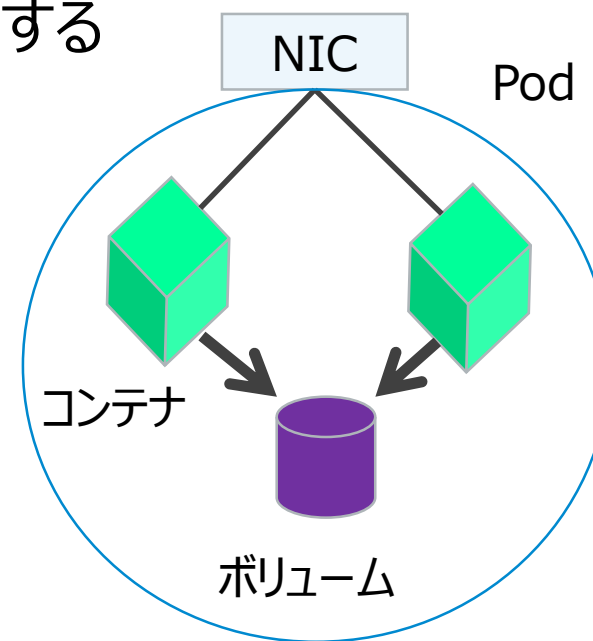


Kubernetes リソース

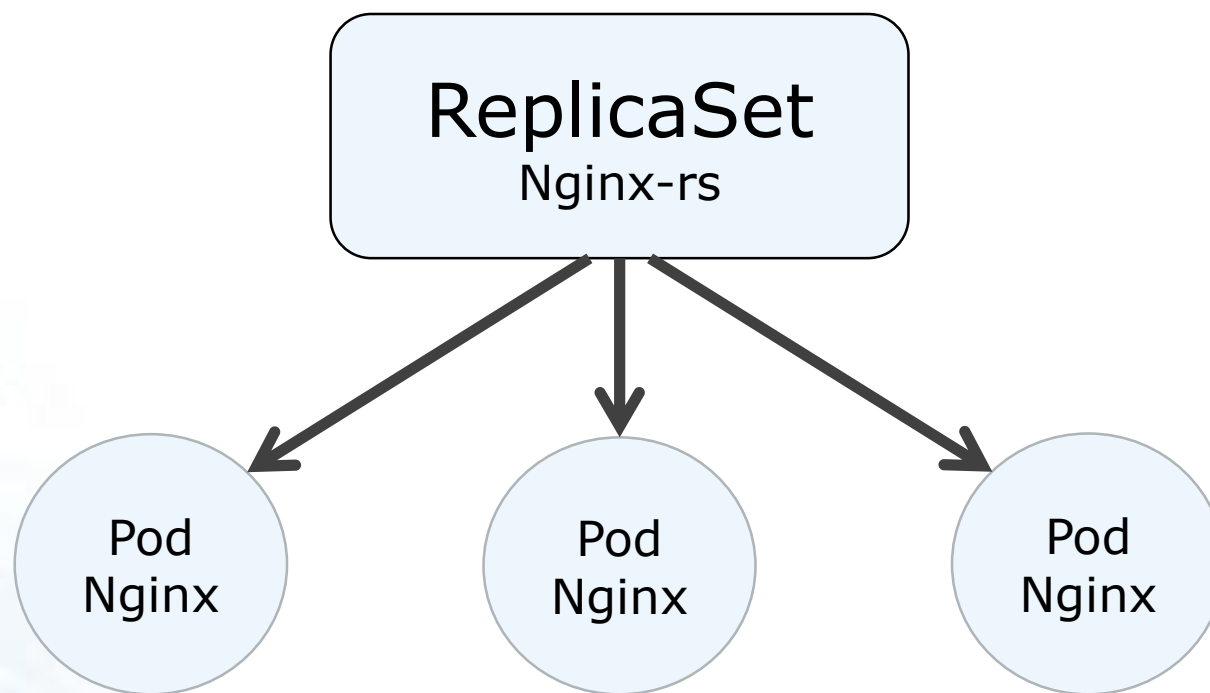
- Kubernetes にデプロイする「論理ホスト」
 - デプロイ可能な最小単位のオブジェクト
 - 1つ以上のコンテナ
 - 共有ネットワーク・ボリューム
 - 同じ Pod 内のコンテナは同一 Node にデプロイされる
 - Pod はいずれかの Namespace に属する

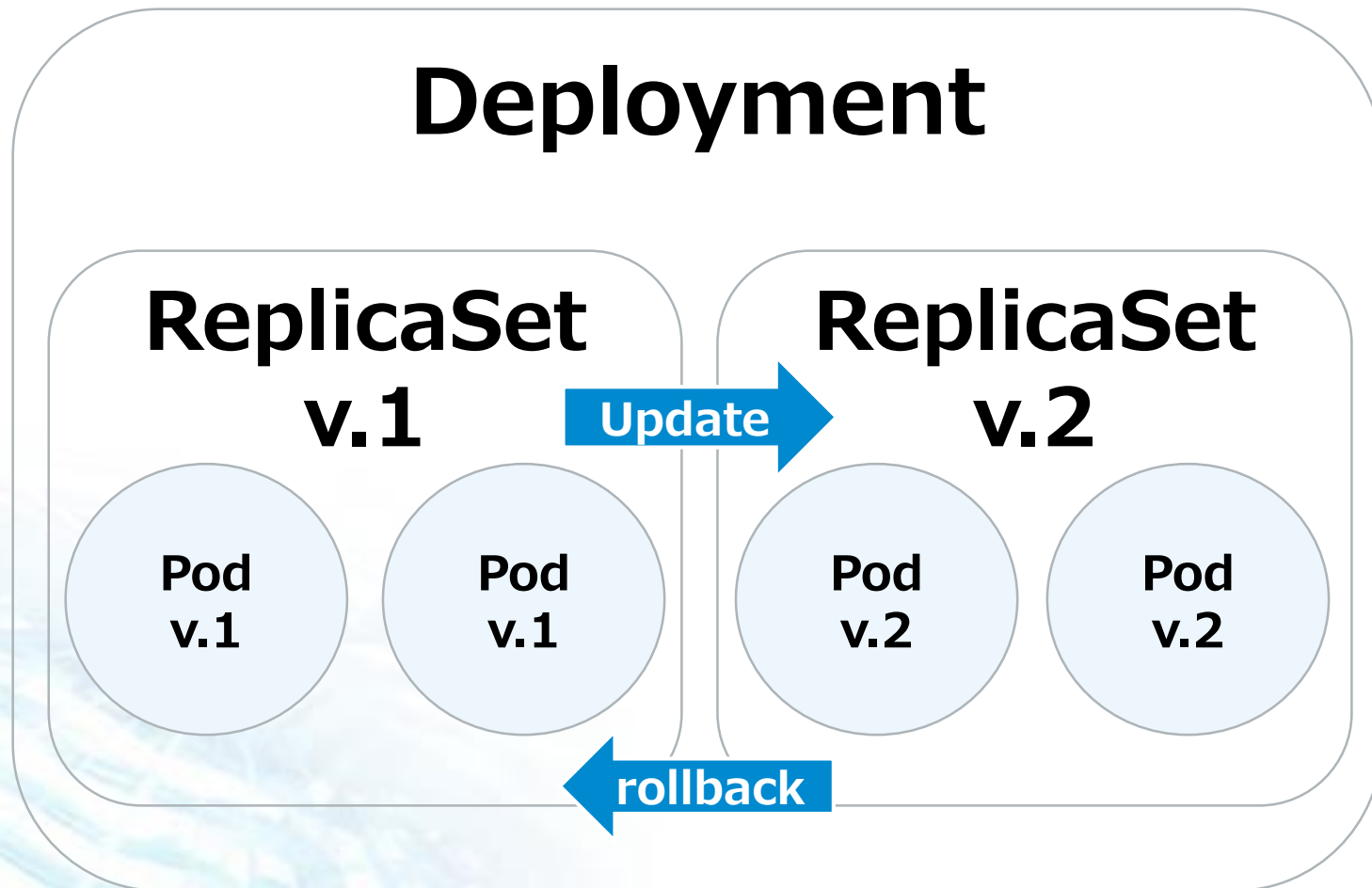


<https://kubernetes.io/docs/tutorials/kubernetes-basics/explore/explore-intro/>



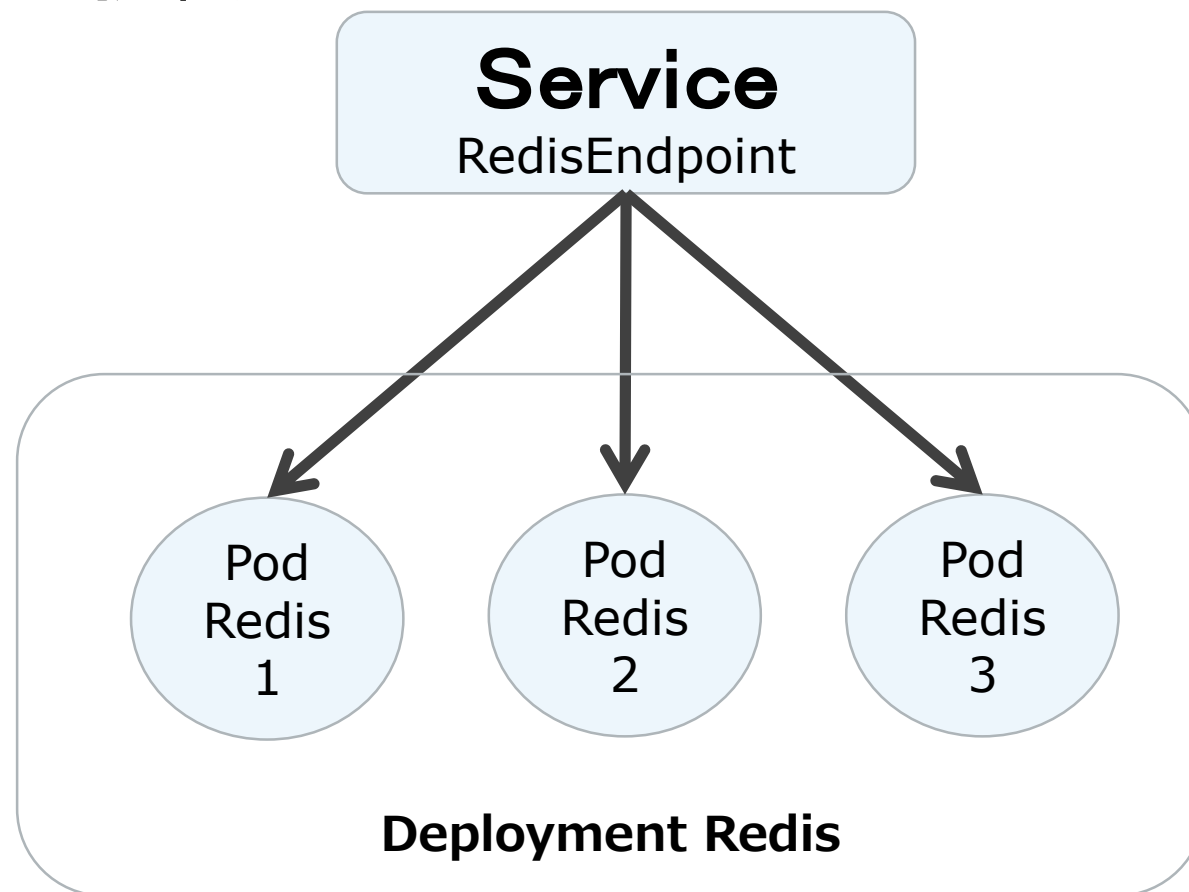
- Pod の上位リソース
- 同じ Pod を指定数だけ起動して管理するリソース
(コントローラが指定数動いているか確認する)





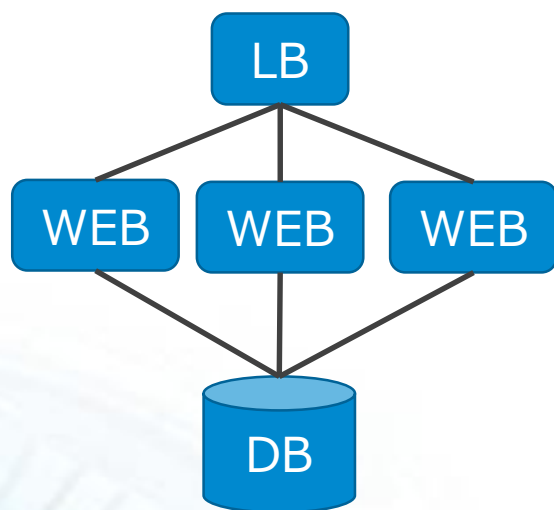
- ReplicaSet の上位リソース
- バージョン管理
特定のバージョンに
アップデート/ロールバック
- アップデート戦略
 - Recreate
Pod をすべて停止、新しい
バージョンで一斉に起動
 - RollingUpdate
少数の Pod を
順次アップデート

- Pod の IP アドレスは不定
- Service は Pod へのエンドポイントを提供
 - クラスタ内部向け
 - 外向け
 - クラスタ内外の負荷分散
- Kubernetes 内部でもっている DNS を利用して名前解決
- アプリからはサービス名でアクセス

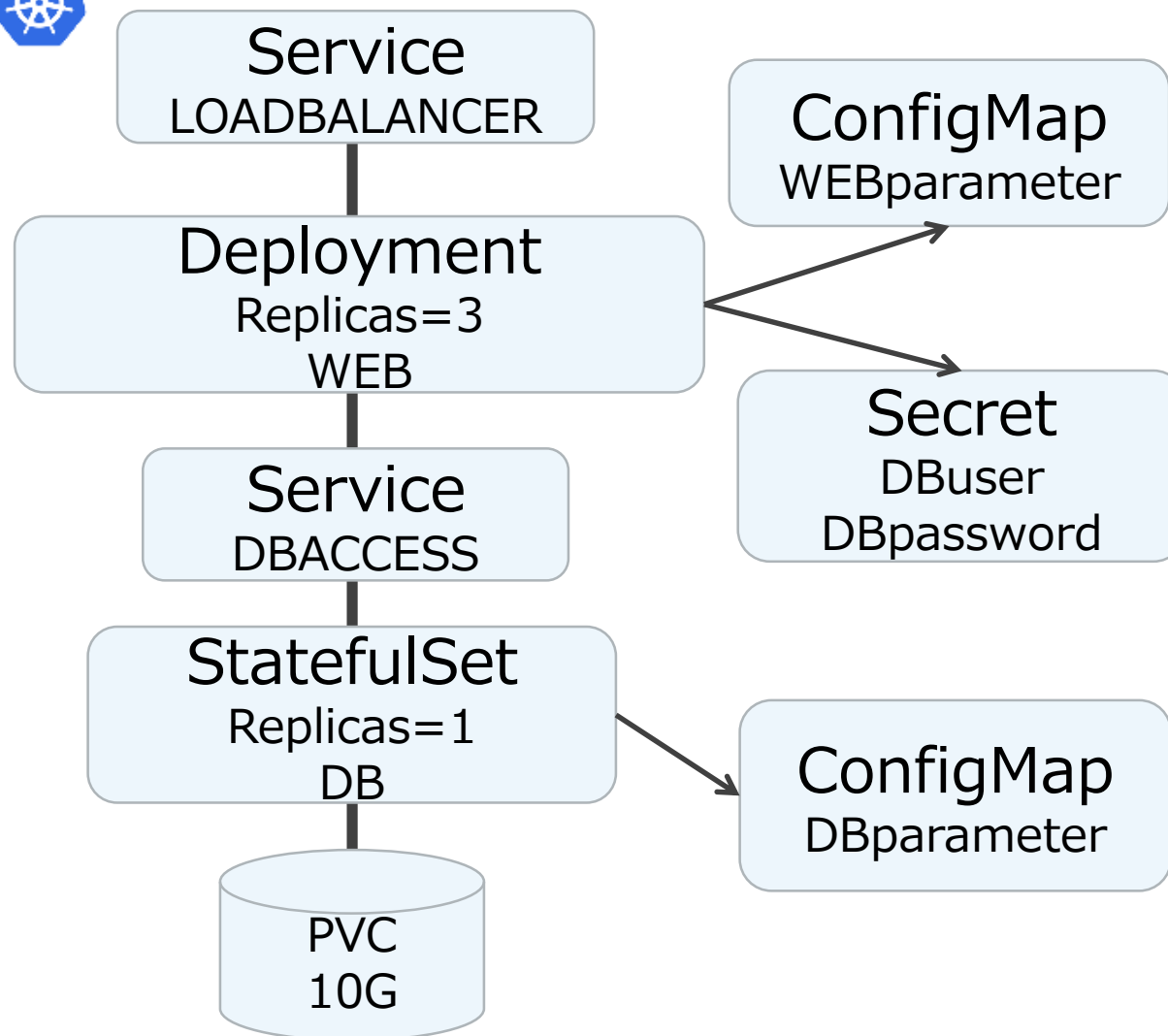


- DaemonSet
 - 各 Node で必ず一つだけ動く Pod
- Statefulset
 - 状態を保持するアプリケーション
 - ネットワーク識別子、永続的なストレージを利用できる
- Job/CronJob
 - 処理が終わったら終了する Pod
- PersistentVolume/PersistentVolumeClaim
 - 永続的なボリューム
- ConfigMap/Secret
 - 設定・パスワード

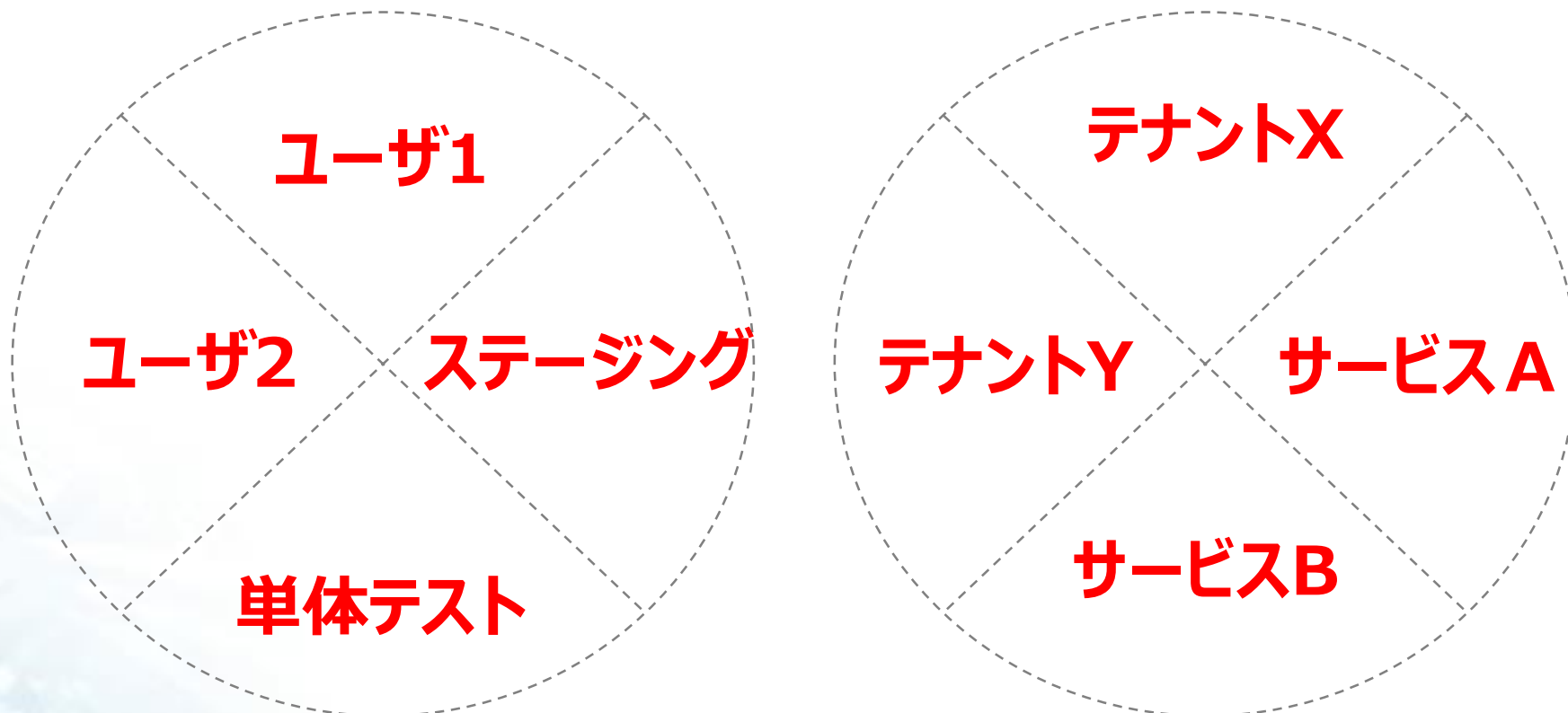
構成図



リソースを組み合わせてYAMLファイルで構成定義



- 一つの kubernetes クラスタ上で複数の仮想クラスタの動作をサポート



開発・テスト環境クラスタ



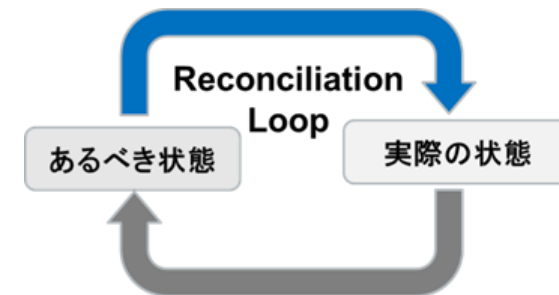
本番環境クラスタ

- リソースを自分で作ることもできる
⇒ 独自の Reconciliation Loopの開発
- オペレータの運用ノウハウをコードで表現できる
- <https://operatorhub.io/>で公開されている

例: Zalando社 : Postgres-Operator

PostgreSQL SR構成を管理してくれるリソース

Primary-Standby(マスター-スレーブ)の状態管理



Postgres-Operator
provided by Zalando SE

Postgres operator creates and
manages PostgreSQL clusters
running in Kubernetes.



まとめ



- コンテナ開発による生産性の向上
- コンテナによる環境管理の負荷軽減
- コンテナによる効率的なサーバ運用
- 自動スケーリング
- 障害の自動修復
- ファイルベースによる構成、バージョン管理
- 特定のベンダに支配されない標準技術



- 学習コストが高い
- Kubernetesのリリースが早くて追従が大変
- エコシステム(関連ソフトウェア)も進歩が速く追従が大変

VMによる運用

- システム管理者
 - VM の払い出し
 - ネットワークの定義
 - ホストの監視・管理
 - ミドルウェアの冗長化対策
- 開発者
 - ssh など で VM にアクセスして環境の構築

コンテナによる運用

- システム管理
 - Kubernetes クラスターの監視・管理
- 開発者
 - ファイルベースで構成を定義
 - Git など でアプリケーションと構成を管理



Appendix

- オンラインコースの活用
 - ブラウザでkubernetesクラスタを操作できる

The screenshot displays the Udemy website interface. At the top, a teal banner reads '今すぐ課題の解決に向けてスタートしよう | 30日間返金保証だから安心。今すぐ登録しよう。' (Start now to solve your problems |安心 because of the 30-day money-back guarantee. Register now.). Below this is the Udemy logo and a search bar with the text '何でも検索' (Search anything). Navigation links for 'ログイン' (Login) and '新規登録' (Sign Up) are visible.

The main content area features a course titled 'Kubernetes for the Absolute Beginners - Hands-on'. The description states: 'Learn Kubernetes in simple, easy and fun way with hands-on coding exercises. For beginners in DevOps.' The course has a rating of 4.6 stars from 6020 reviews and 28531 students. The instructor is Mumshad Mannambeth, and the course was last updated on 7/2019. It is available in English and has subtitles in English, Italian, and others.

Below the course title, the '学習内容' (Course Content) section lists topics such as 'Gain basic understanding of Kubernetes Fundamentals', 'Deploy Kubernetes Cluster on local systems', 'Deploy Applications on Kubernetes', 'Develop Kubernetes Configuration Files in YAML', 'Deploy Kubernetes on Google Cloud Platform', and 'Setup ReplicaSets, Service Deployments on Kubernetes'.

On the right side, there is a section titled 'あなたのペースで学ぼう' (Learn at your own pace), which encourages users to learn at their own pace with expert guidance. It includes a search bar for '学びたいコースを検索' (Search for the course you want to learn).

At the bottom, a red banner highlights three key features: '100000のオンラインコース' (100,000 online courses), '専門的指導' (Expert guidance), and '学習期限なし' (No learning deadline).

- Certified Kubernetes Administrator (CKA) 試験
 - Kubernetesの管理者に必要なスキル、知識、および能力が問われる試験
 - ハンズオン形式の試験 試験時間3時間
- Certified Kubernetes Application Developer (CKAD) 試験
 - Kubernetes上でアプリケーションを設計・構築・設定・展開できることを証明する資格試験
 - ハンズオン形式の試験 試験時間 2時間



- **kubernetes slack** <https://slack.k8s.io/>
#jp-users #jp-events チャンネル
- **Kubernetes Meetup Tokyo**
<https://k8sjp.connpass.com/>
毎月開催 19:00~
200~300人規模のイベント
youtube配信あり



CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape landscape.cncf.io has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider cncf.io/cscsp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

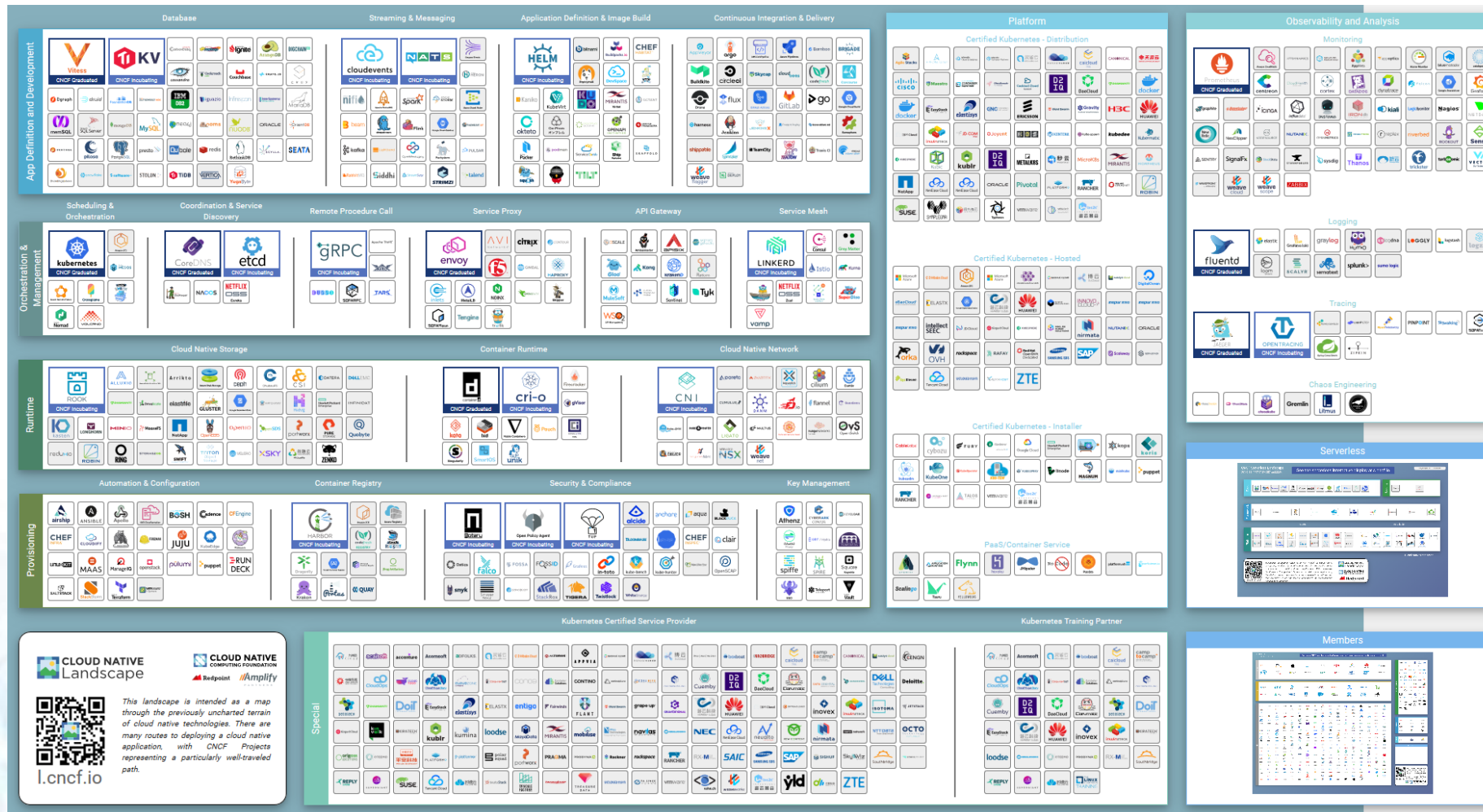
The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

landscape.cncf.io

v20181211



<https://github.com/cncf/landscape/blob/master/README.md#trail-map>



ご清聴ありがとうございました。