

# Case Study

## Kubernetes (K8s) on R-Car R-CarH3への最適実装例のご紹介

---

Osaka NDS Embedded Cross Online Forum #13

山口 洋樹



株式会社大阪エヌデーエス

# プロフィール

---

## □ 経歴

- 2011 ~
  - 大阪エヌデーエス入社
  - 組み込み開発 (ITS車載器, 空調システム, 空気清浄機 etc.)
- 2015 ~ 現在
  - Linux 組み込み開発
    - » 無線LANルータ開発
    - » パブリッククラウド 連携 コネクテッドカー向け PoC
    - » 次世代車載IVIシステム向け開発  
etc.

## □ 趣味

- Tennis (Two-Handed Backhand)

# 目次

---

## ❖ Kubernetes(K8s) とは

- K8s の概要、導入方法を解説

## ❖ K8s on R-Car 実装例

- R-Car H3 (arm64) 上で K8s を構築時、苦労した点や課題について共有

# Kubernetes(K8s) とは

# K8s 概要

## ❖ 意味/読み方

Apple SE



francesc @francesc · 2014年7月11日

So kubernetes is pronounced /koo-ber-nay'-tace/ and means "sailing master"

Ok, that's a cool name!  
[biblestudytools.com/lexicons/greek...](http://biblestudytools.com/lexicons/greek...)

2

59

67



Kubernetes = 船長



Kubernetes共同創設者/  
Microsoft Azure SE



brendandburns  
@brendandburns

返信先: @imdsmさん

@imdsm @gregde @francesc @jbeda @developerluke  
most people on the team say: koo-ber-net-ees, or just  
'k8s' or k-eights ケーエイツ/クーバネティス

午前1:29 · 2015年4月8日 · TweetDeck

8件のリツイート 10件の引用ツイート 31件のいいね



Kubernetes  
8文字

<https://twitter.com/brendandburns/status/585479466648018944>

# K8s 概要

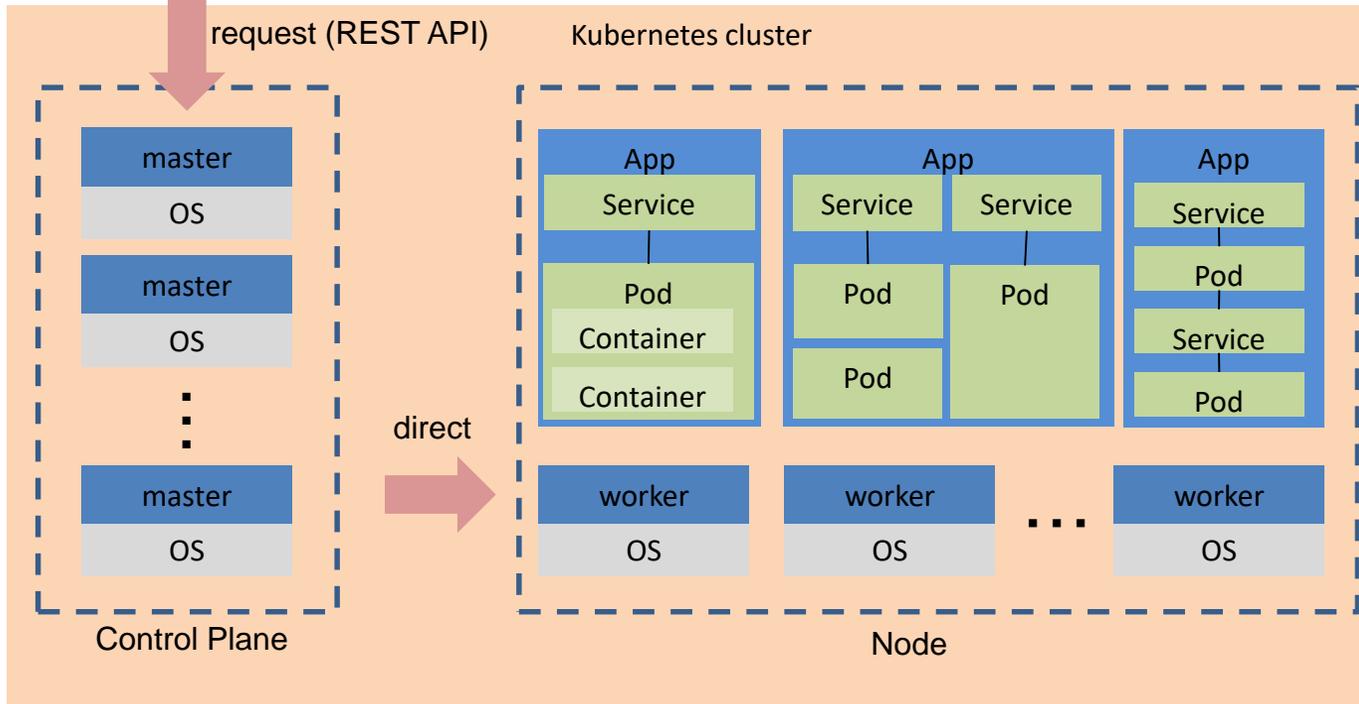
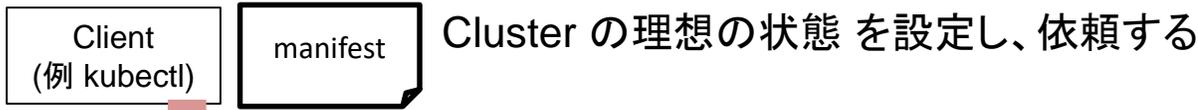
---

## ❖ K8s とは

- ❑ 本番環境に耐えられるように、  
コンテナ化されたアプリを自律的に管理するオープンソースプラットフォーム
- ❑ 何が嬉しいか
  - コストがかかるアプリの管理(フェイルオーバー、スケーリング等)をK8sが面倒見してくれる
  - 管理コストが削減され、開発者はアプリ開発に集中することができる。
- ❑ コンテナ・オーケストレーション ツール
  - オーケストレーション:  
複数のシステムにまたがるワークフローを自動化する
  - クラウド・オーケストレーション 例:
    - AWS CloudFormation / OpenStack Heat / Ansible Automation Platform
  - コンテナ・オーケストレーション 例:
    - Kubernetes (コンテナ・オーケストレーションのデファクトスタンダード)
    - Docker Swarm / Apache Mesos

# K8s 概要

## ❖ アーキテクチャ



### Control Plane

master とも呼ばれる。  
コンテナ・オーケストレーションを指揮する。  
複数の Control Plane で構成される。(高可用性)

### Node

worker とも呼ばれる。  
App を実行するコンポーネント。  
複数の Node で構成される。(高可用性)

### Cluster

複数 Node の集合。

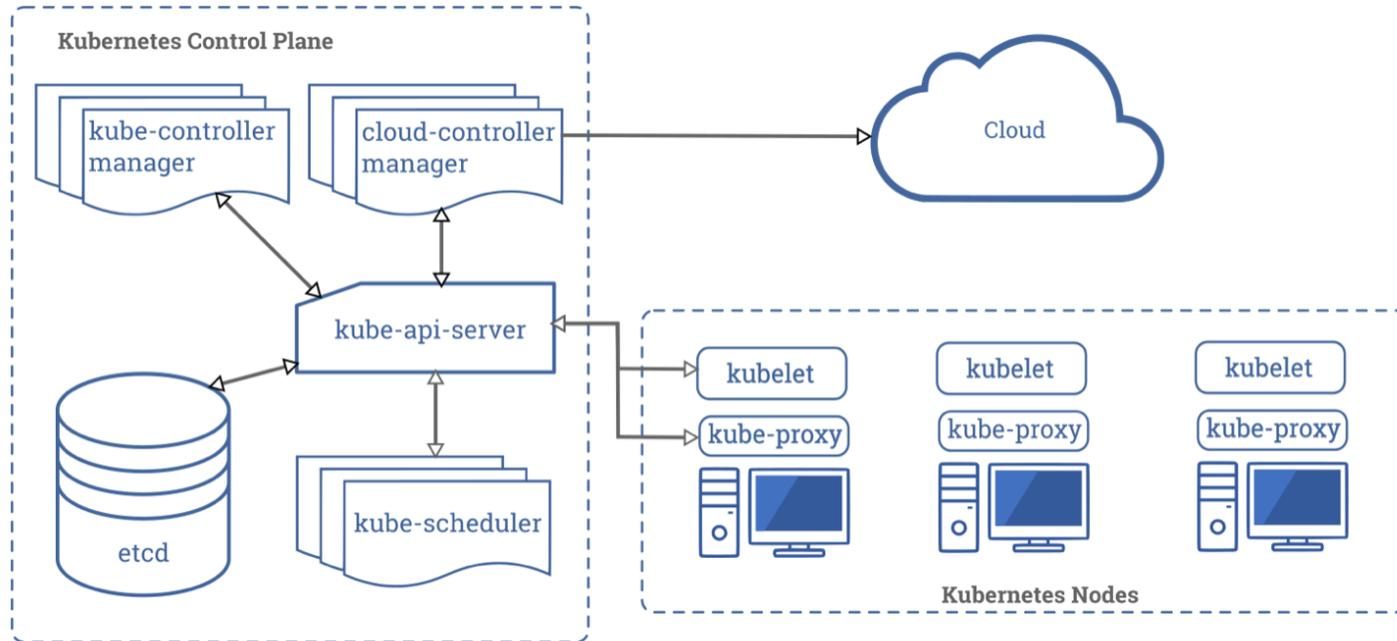
### App

Pod (複数のコンテナのセット) と  
Service (どう接続するかの定義) 等の  
オブジェクトで構成される。

# K8s 概要

## ❖ アーキテクチャ

### ❑ K8sクラスタを構成するコンポーネント



### 「Control Plane」

#### kube-api-server

K8sのAPI。(REST API)

#### kube-controller manager

クラスタを「理想の状態」にする。

#### kube-scheduler

Podを適切なノードへスケジューリング。

#### etcd

クラスタのすべての情報はここに保存。

### 「Node」

#### kubelet

コンテナの操作/ノードの監視。

#### kube-proxy

定義されたルールに従いパケット転送を行う。

<https://v1-18.docs.kubernetes.io/ja/docs/concepts/overview/components/>

❑ 1つ1つ設定するのは超大変 => 構築ツールを使用して簡単に構築できる (後述)

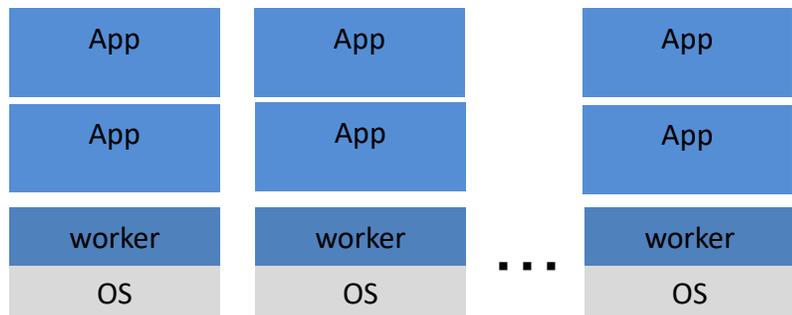
# K8s 概要

## ❖ 主な機能 (1/3)

### □ ノードの抽象化

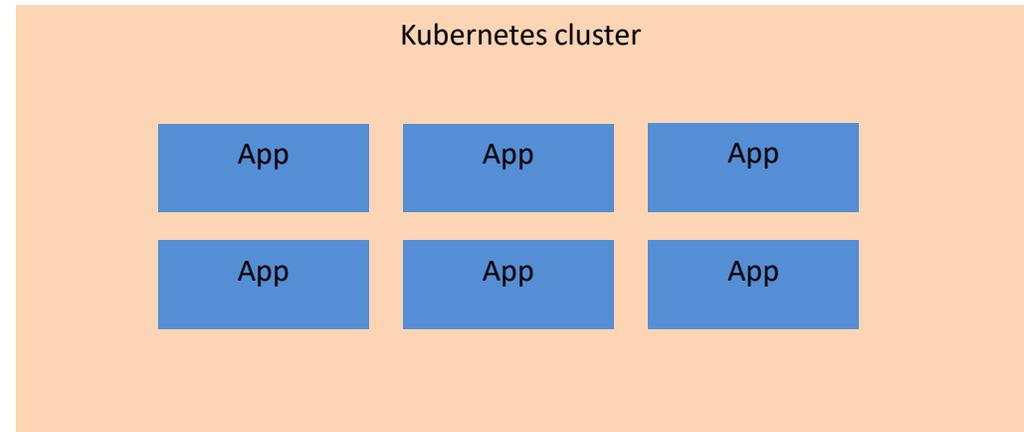
- 複数のノードを1つの大きなリソースプール (クラスタ) として抽象化する。
  - 開発者はどのノードでAppが実行されるかを意識せずに開発できる。
    - » K8sは、コンテナが適切なノードで実行できるようにスケジューリングする。
    - » ホストを跨いだコンテナ間通信の仕組みを提供している。

実際にはAppはいずれかのNode上で実行される



Node

抽象化



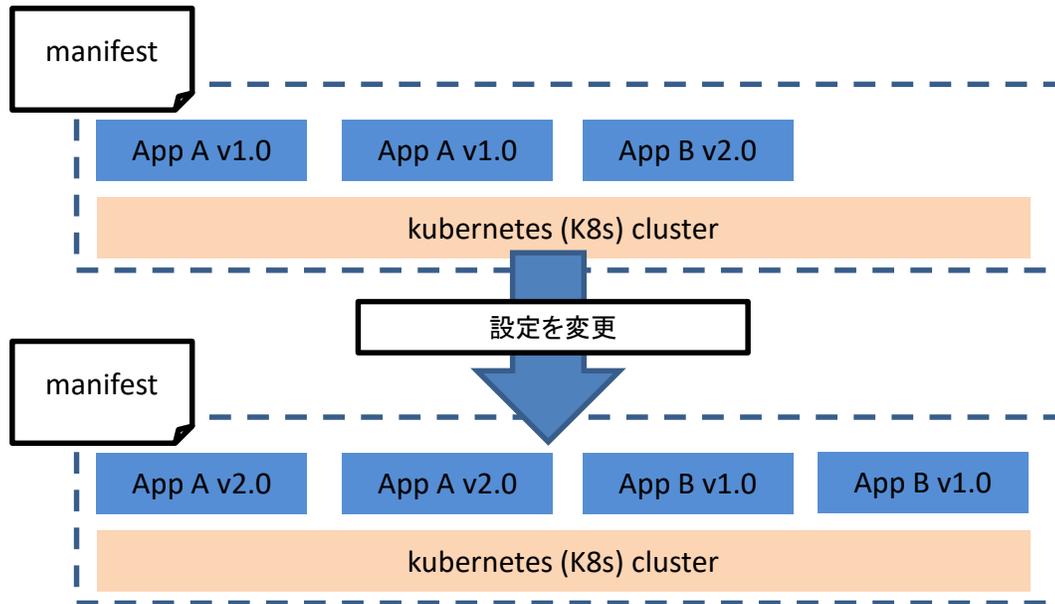
1つの大きなクラスタとして見せる

# K8s 概要

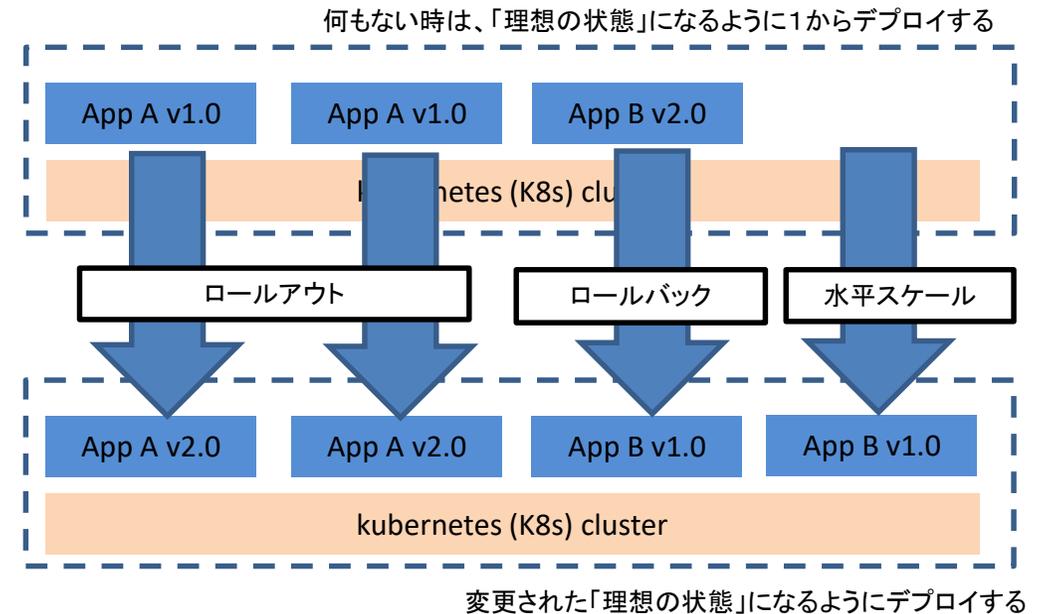
## ❖ 主な機能 (2/3)

### □ デプロイの自動化

- 「理想の状態」を設定ファイル(manifest)に記述することで、常にその状態になるように自動デプロイする。



「理想の状態」を宣言



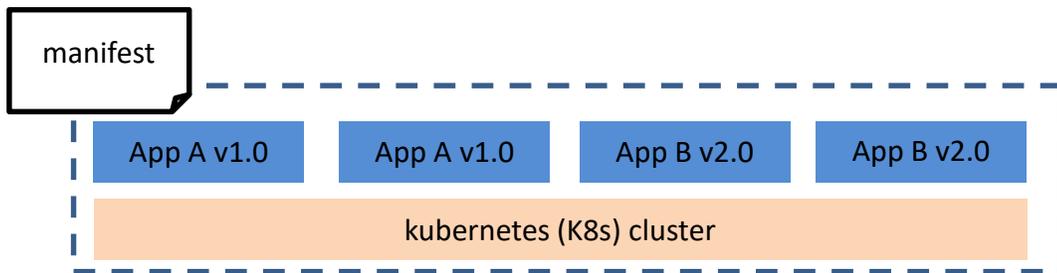
デプロイの自動化

# K8s 概要

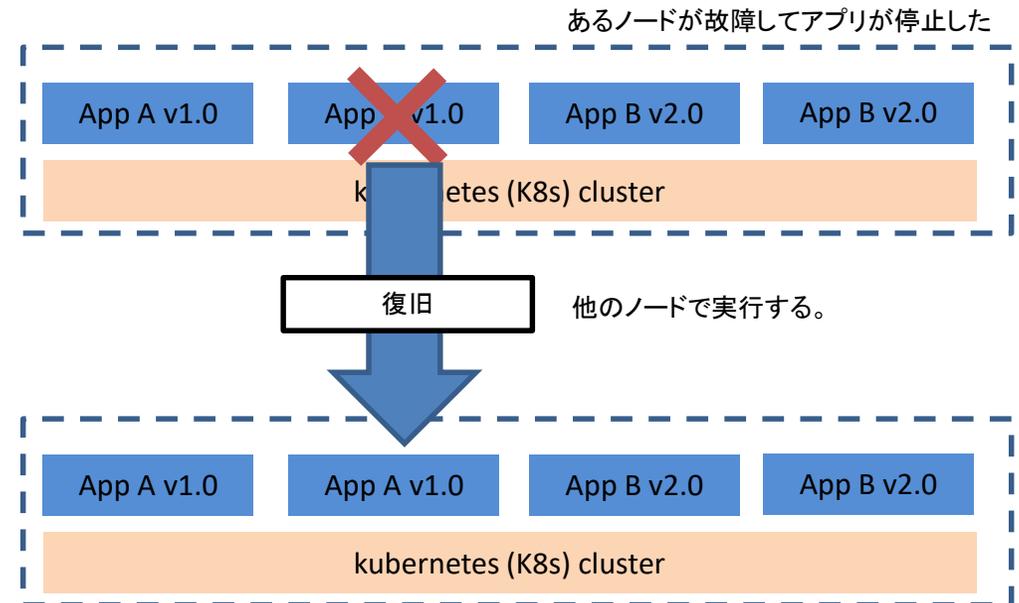
## ❖ 主な機能 (3/3)

### □ 自己回復

- App の一部が停止しても、「理想の状態」になるようにデプロイし、復旧してくれる。
- 高信頼性を持つシステムを構築できる。



「理想の状態」を宣言



自己回復

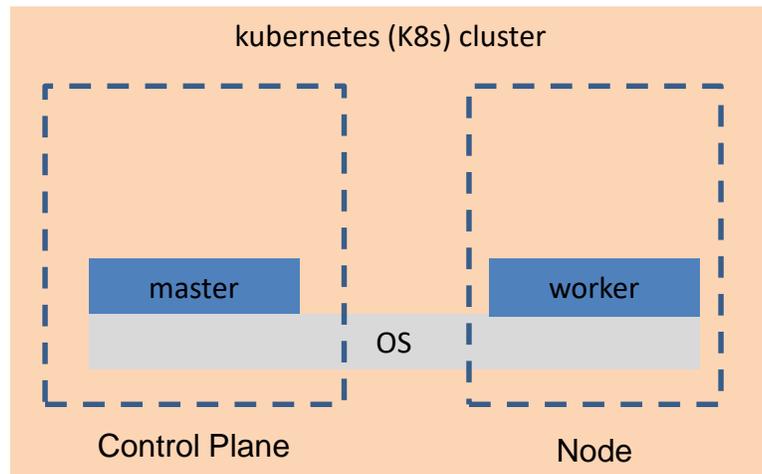
# K8s 導入

## ❖ K8s クラスタ構築ツールの紹介 (1/2)

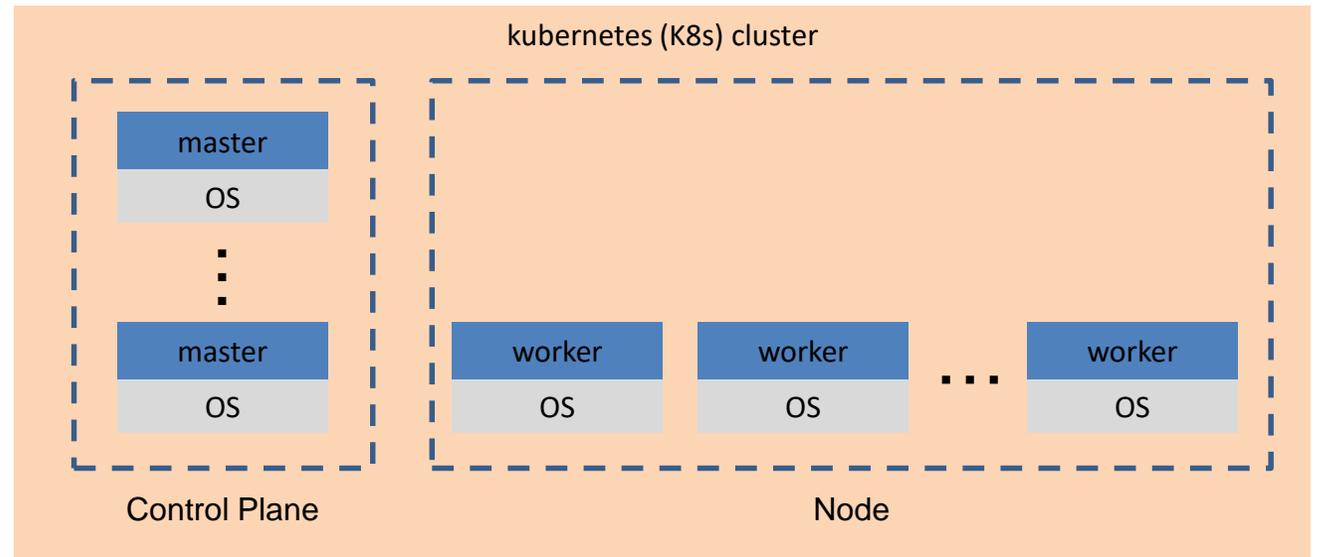
### □ オンプレミス環境に構築

- Kubeadm
- Minikube (Kubeadm を Wrap)
- Kubespray (Kubeadm を Wrap / Ansible)

} K8sの機能を理解しながら構築/検証できる (オススメ)  
} まずは簡単にPC1つで動かしたい人向け



Minikube で構築した例



Kubeadm で構築した例

# K8s 導入

## ❖ K8s クラスタ構築ツールの紹介 (1/2)

### □ Kubeadm でクラスタを構築する手順概要

- ① ロードバランサーを設置
- ② 1個目の Control Plane を作成

```
$ kubeadm init
```

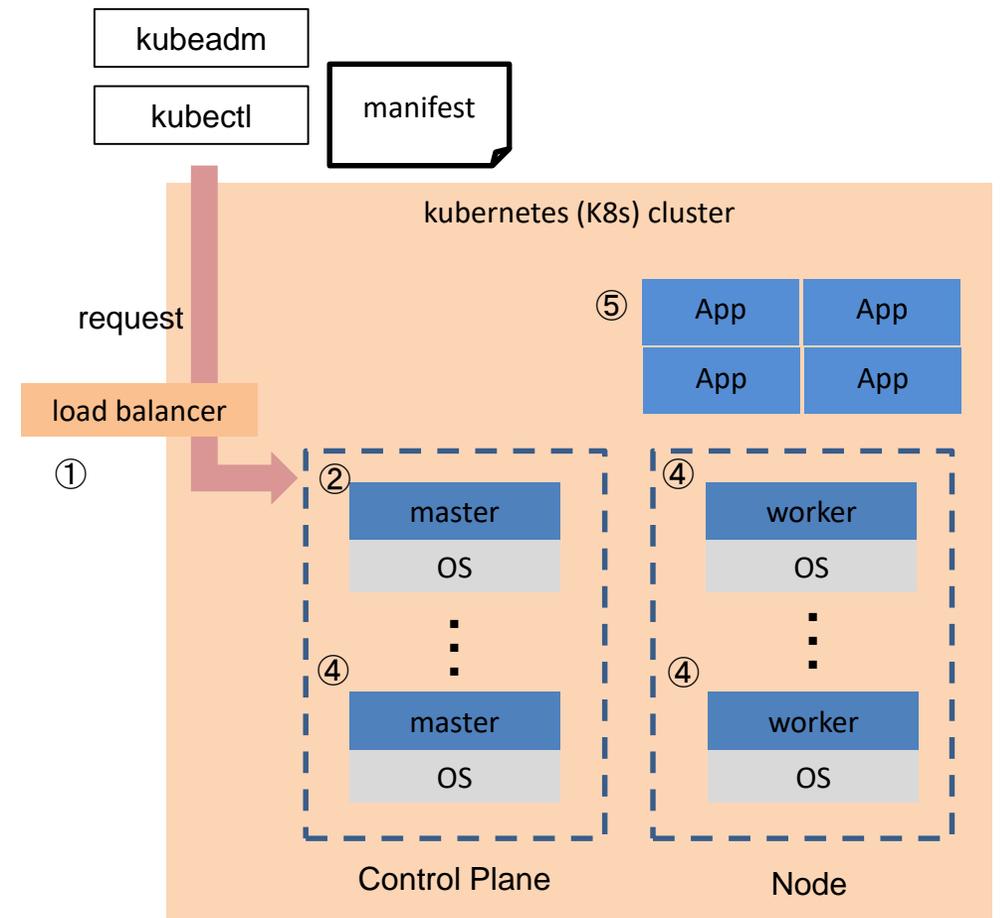
- ③ ネットワークアドオンをインストール
  - ホストを跨いだコンテナ間通信をする為に、CNI 準拠のネットワークアドオンをインストールする

- ④ Control Plane / Node を登録

```
$ kubeadm join # ホスト毎に実行
```

- ⑤ manifest を記述し、App のデプロイ実施

```
$ kubectl apply
```



手順詳細: <https://kubernetes.io/ja/docs/setup/production-environment/tools/kubeadm/>

# K8s 導入

## ❖ K8s クラスタ構築ツールの紹介 (2/2)

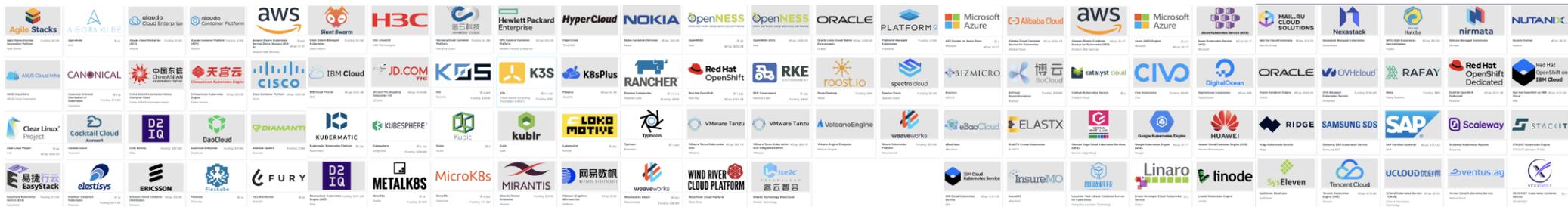
### □ マネージドサービス を使って簡単に構築 (本番環境向け)

- Google Kubernetes Engine (GKE)
- Amazon Elastic Kubernetes Service (Amazon EKS)

## ❖ CNCF認定ツール

### □ 紹介したツール含む、Cloud Native Computing Foundation に認定されている 安心のディストリビューション/ツール一覧

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



# K8s on R-Car 実装例

# K8s on R-Car 実装例

---

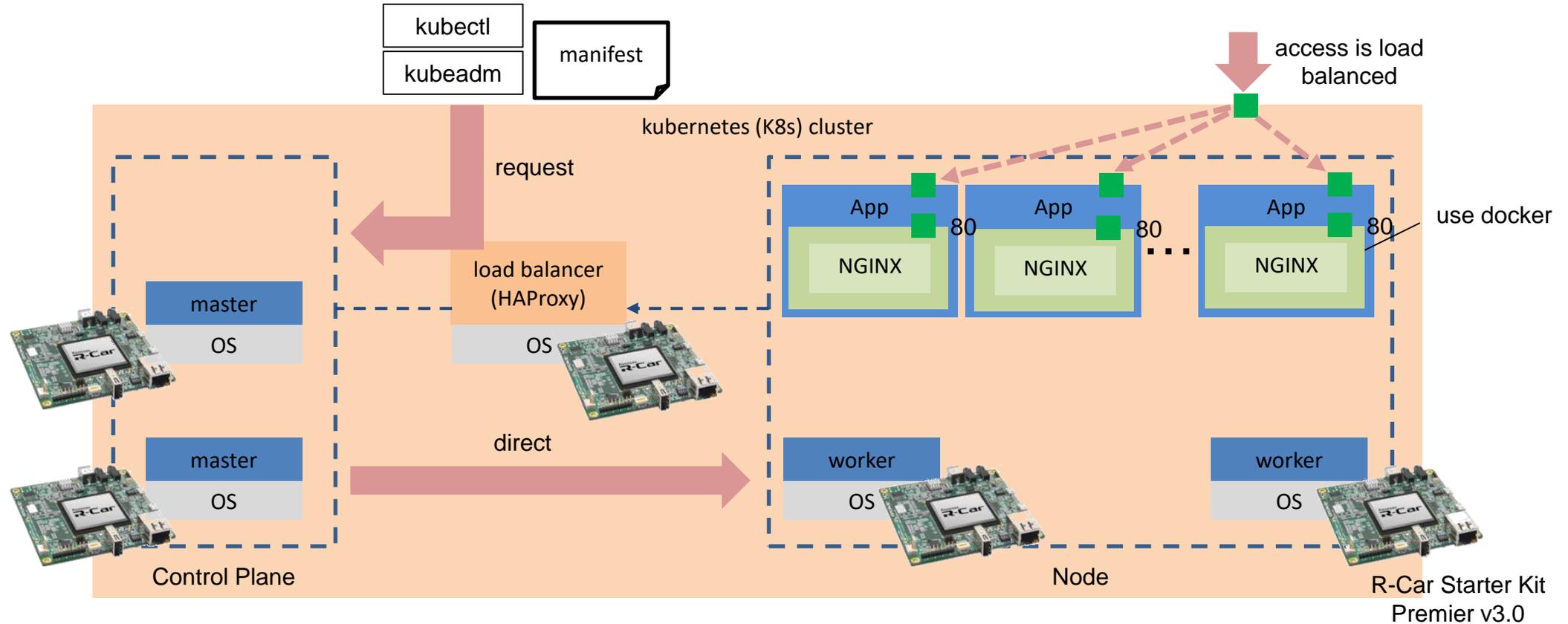
## ❖ 概要

- ❑ 複数台の R-Car を使用した K8s 環境の実装例の紹介
- ❑ 構築にあたり苦労した点や課題を共有
  - K8s クラスタ構成ツールの選定に苦労
  - etcd への Request Timeout 問題
  - ネットワークアドオンのインストールエラー 問題

# K8s on R-Car 実装例

## ❖ システム構成

□ Kubeadm を使用して以下の K8s クラスタを構成。



手順詳細: [https://elinux.org/R-Car/k8s\(J\)](https://elinux.org/R-Car/k8s(J))

# K8s クラスタ構成ツールの選定に苦勞

---

## ❖ Linux ディストリビューションの選定 (2020/9 時点)

### × Yocto (poky)

- meta-virtualization に必要なpackage (docker, kubeadm etc...) は提供されているが docker / K8s のバージョン互換性がなく実行できない。

### ○ Ubuntu 18.04 (arm64)

## ❖ K8s クラスタ構築ツールの選定 (2020/9 時点)

### × Kubespray, Minikube

- Arm64 未サポート の為実行できない。

### ○ Kubeadm

# etcd への Request Timeout 問題

---

## ❖ 問題と原因

- ❑ etcd への Request Timeout エラーが発生し、Control Plane / Node を追加できない。
- ❑ eMMC上の etcd へのアクセスに時間がかかっていることが原因。

## ❖ 対策(暫定)

- ❑ etcd を eMMC -> RAM にマウントして対策 (永続性が無くなる)

## ❖ 備考

- ❑ タイムアウト時間の設定変更を試みた
  - kubeadm コマンドのオプションで設定ファイルを指定できるが、他のオプションと併用できず断念。
- ❑ etcd を別のホストに分離し、アクセス速度の改善ができそう。
  - 高可用性etcdクラスターの作成: <https://kubernetes.io/ja/docs/setup/production-environment/tools/kubeadm/setup-ha-etcd-with-kubeadm/>

# ネットワークアドオンのインストールエラー 問題

## ❖ 問題と原因

- ❑ ネットワークアドオンのインストールに失敗する
- ❑ 原因は (主に) ネットワーク関連の Linux Kernel 機能不足

## ❖ 対策

- ❑ ネットワークアドオン、ネットワーク通信を担う Kube-proxy の エラーログを確認し、不足している Kernel Option を調査して追加。(地道な作業)

## ❖ 備考

- ❑ Docker のように、必要な Kernel Option 一覧を出してくれるツールがなく苦勞した (2020/9 時点)

```
CONFIG_CGROUP_FREEZER=y
CONFIG_NETFILTER_XT_MATCH_IPVS=m
CONFIG_IP_VS=m
CONFIG_IP_VS_TAB_BITS=12
CONFIG_IP_VS_SH_TAB_BITS=8
```

```
CONFIG_BLK_DEV_THROTTLING=y
CONFIG_CFQ_GROUP_IOSCHED=y
CONFIG_NET_SCHED=y
CONFIG_NET_CLS=y
CONFIG_NET_CLS_CGROUP=m
CONFIG_NET_SCH_FIFO=y
CONFIG_CGROUP_NET_CLASSID=y
CONFIG_RT_GROUP_SCHED=y
CONFIG_CGROUP_NET_PRIO=y
CONFIG_CFS_BANDWIDTH=y
CONFIG_IP_VS_NFCT=y
CONFIG_IP_VS_PROTO_TCP=y
CONFIG_IP_VS_PROTO_UDP=y
CONFIG_IP_VS_RR=m
CONFIG_EXT3_FS_POSIX_ACL=y
CONFIG_EXT3_FS_SECURITY=y
CONFIG_EXT4_FS_SECURITY=y
```

```
CONFIG_XFRM_USER=m
CONFIG_XFRM_ALGO=m
CONFIG_INET_ESP=m
CONFIG_NET_L3_MASTER_DEV=y
CONFIG_IPVLAN=m
CONFIG_DUMMY=m
CONFIG_NF_NAT_FTP=m
CONFIG_NF_CONNTRACK_FTP=m
CONFIG_NF_NAT_TFTP=m
CONFIG_NF_CONNTRACK_TFTP=m
```

```
CONFIG_DAX=m
CONFIG_DM_THIN_PROVISIONING=m
CONFIG_SQUASHFS_XZ=y
CONFIG_BRIDGE_NF_EBTABLES=m
CONFIG_IP_NF_TARGET_REDIRECT=m
CONFIG_NETFILTER_XT_MATCH_COMMENT=m
CONFIG_IP_NF_RAW=m
CONFIG_IP_VS=m
CONFIG_IP_VS_WRR=m
CONFIG_IP_VS_SH=m
CONFIG_IP_VS_CT_NETLINK=m
CONFIG_NF_CONNTRACK_IPV4=m
CONFIG_NETFILTER_XT_SET=m
CONFIG_NETFILTER_XT_MATCH_MULTIPORT=m
CONFIG_NETFILTER_XT_MATCH_PHYSDEV=m
CONFIG_NETFILTER_XT_MATCH_RECENT=m
CONFIG_NETFILTER_XT_TARGET_REDIRECT=m
CONFIG_IP_SET=m
CONFIG_IP_SET_HASH_IP=m
CONFIG_IP_SET_HASH_NET=m
CONFIG_NETFILTER_XT_MARK=m
CONFIG_NETFILTER_XT_MATCH_STATISTIC=m
```

# 課題

---

## ❖ K8s on R-Car 構築 課題

- ❑ データベース(etcd)のアクセス速度の改善
- ❑ デプロイ時間の改善
  - R-Car: K8sクラスタ構成～アプリのデプロイまで約30分 要する
  - Ubuntu(x86): 約 2～3分 で終わる

# Thank you

---

## ❖ まとめ

### □ Kubernetes の概要と導入方法について紹介

- Kubeadm 等のクラスタ構築ツールを利用して簡単にスタートできます。

### □ K8s on R-Car 実装例 について紹介

- エッジ側にK8sを構築する場合の課題について共有。

近年、クラウドを活用したシステム連携が進み、  
クラウドを支えるK8sを含むコンテナ技術の理解が求められています。  
本内容がお役に立てれば幸いです。