Agenda

• Who are we?
• Background
• Problems
• Goals / Requirements
• Kubernetes
• KubeEdge
• Sample Deployment
• What’s missing? Next-gen proposal
• Community
Who are we?

• Tomoya Fujita (Presenter)
  • Software Engineer, Sony R&D US Laboratory
  • ROS TSC (Technical Steering Committee)
  • KubeEdge SIG Robotics Chair
  • fujitatomoya@github, tomoyafujita@linkedin

• Co-Authors
  • Yin Ding (Engineering Manager, Google)
    o KubeEdge TSC, Co-Founder of KubeEdge Project
    o Leading the Kubernetes Hardening team
  • Kevin Wang (Lead of Cloud Native Open Source Team, Huawei)
    o KubeEdge TSC, Co-Founder of KubeEdge Project
    o CNCF Ambassador, TOC contributor
  • Fei Xu (Senior Engineer, Huawei)
    o KubeEdge TSC, Maintainer
Background

• Broad use cases.

• Distributed and Connected System.

• Collaborative and Orchestrated Application.

• Circulatory Functioning System and Development

• Specific Hardware Acceleration.

• Security. (Device, Data, Network)
What is the pain?

• Platform Dependencies.

• Proprietary hardware support.

• Application Modularity.

• System and Security Integration.

• Application Specific Network Bridge.

• Application Developer Friendly.
Goal / Requirements

• Flexible Application Deployment.

• Zero Trust Security Support.

• Application Agnostic Network Configuration.

• Extend Device Capability.

• System Global Observability.

• Platform Agnostic Device Abstraction.
Kubernetes (Service Mesh)

- Application Deployment and Orchestration.
- Device Capability and Label Control.
- Custom Resource Extension.
- Auto- Scaling and Healing.
- Roll Up/Down, Canary Test.
- Role Based Access Control.
- Device-Plugin / Container Device Interfaces.
- Container Network Interfaces.
- Traffic Management.
- Observability.
- Security Policy.
KubeEdge

is built upon Kubernetes and provides core infrastructure support for networking, application deployment and metadata synchronization between cloud and edge.

- Cloud-Edge Coordination
- Edge Computing
- Edge Autonomy
- Simplified Deployment
- Kubernetes-native Support
- Resource Efficient

https://kubeedge.io/docs/
Sample Deployment

*Theory is good, but please see how it works in the flesh!*

**ROS Kubernetes Tutorials**

*Contribution (Issues/PRs) always welcome!*
ROS 2 Localhost Only

LAN (Physical Network)

Layer 2 Emulation (WeaveNet)

Primary node

Worker node

Kubernetes API Server

Kubelet (agent)

Pod

ros2 talker

ros2 listener

ROS LOCALHOST ONLY=1

ROS LOCALHOST ONLY=1

ROS LOCALHOST ONLY=1

ROS LOCALHOST ONLY=1
ROS 2 Logical Partition / Multiple RMW Implementation

LAN (Physical Network)

Bind Host Network Interface

ROS_DOMAIN_ID=5
RMW_IMPLEMENTATION=rmw_fastdds_cpp

ROS_DOMAIN_ID=10
RMW_IMPLEMENTATION=rmw_cyclonedds_cpp

Primary node

Worker node

Label: {nodetype: edgeserver}

Label: {nodetype: edgedevice}

Docker

Kubernetes API Server

Kubelet (agent)

ROS2 CLI

ROS2 talker Pod

ROS2 listener Pod

ROS2 listener Pod

ROS2 talker Pod

Kubernetes agnostic
ROS 2 Deployment Intermediate

LAN (Physical Network)

Layer 2 Emulation (WeaveNet)

Primary node

Worker node

Kubernetes API Server

Label:
{nodetype: edgeserver}

Kubelet (agent)

Label:
{nodetype: edgedevice}

Start turtlesim teleop app in this container

Pod

Pod

turtlesim

rgt

ros2 shell
ROS 2 / Micro-ROS with KubeEdge (W.I.P)
Device-Abstraction (Device-Plugin, Container Device Interface)

- Kubernetes Custom Resource Definition, that allows us to plugin vendor specific hardware and device to the containers.
- After advertising the custom resource to Kubernetes, Kubernetes controls those resources with workload based on application requirements.
Support SROS 2 security enclaves via ConfigMap

Certificate to Join this entire distributed system. Access permission for each topics and services

Administrator

Primary

Worker

Worker

User

Bind Security Enclaves

kublet

App Container

App

App

App

LAN

Access Control

Registration

Load

ConfigMap & Secrets for Each ROS2 Application
What’s missing? Proposals?

• Device Abstraction Enhancement
  • KEP-3162: Add Deallocate and PostStopContainer to Device Manager API
  • Add CDI devices to device plugin API resolved this issue.

• KubeEdge CNI support (e.g. edgemesh, Cilium)

• More Edge Optimization / Configurable Options for Resource Constrained Device Support

• Cloud-Native Robotics Management Solution
  • `RoboDevOps` through Edge-Cloud Synergy
  • Cloud-Native Digital Twin for testing and data generation training
  • Robotics App Development Friendly
  • Cloud Robotics Custom Resource Definition and Operator Proposal
  • Edgemesh: adaptive cross-edge and edge-cloud data plane support
  • VSLAM algorithm with KubeEdge
  • Building a Robot-Oriented Intelligent Monitoring System
Community

Kubernetes IoT Edge WG

KubeEdge SIG Robotics

ROS Kubernetes Tutorials