ROS / ROS 2
with
Kubernetes and KubeEdge

Oct.19th, 2023
ROSCon 2023 @ New Orleans, US
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 Multi-Node Deployment

LAN (Physical Network)

VxLAN (CNI plugin)

DNS hostname

Kubernetes API Server

Pod/Service

Label:
{nodetype: master}

Primary node

Worker node

Kubelet (agent)

Pod/Service

Label:
{nodetype: worker}

ROS Multi-Node Deployment

ros-master

ros-listener

ros-talker

rosmaster

rospy listener

gkube-dns

rosccpp talker
ROS 2 Localhost Only

LAN (Physical Network)

Layer 2 Emulation (WeaveNet)

Kubernetes API Server

Primary node

Worker node

ros2 talker

ros2 listener

Pod

ROS_LOCALHOST_ONLY=1

ROS_LOCALHOST_ONLY=1

ROS_LOCALHOST_ONLY=1

ROS_LOCALHOST_ONLY=1

ros2 talker

ros2 listener

Pod

Kubelet (agent)
ROS 2 Logical Partition / Multiple RMW Implementation

LAN (Physical Network)

Bind Host Network Interface

ROS_DOMAIN_ID=5
RMW_IMPLEMENTATION=rmw_fastrtps_cpp

ROS_DOMAIN_ID=10
RMW_IMPLEMENTATION=rmw_cyclonedds_cpp

Kubernetes agnostic

Docker

ROS2

CLI

Pod

ROS2 talker

ROS2 listener

Kubernetes API Server

Primary node

Label:
{nodetype: edgeserver}

Worker node

Label:
{nodetype: edgedevice}

Kubelet (agent)
ROS 2 Deployment Intermediate

LAN (Physical Network)

Layer 2 Emulation (WeaveNet)

Primary node

Worker node

Kubernetes API Server

Kubelet (agent)

Label:

{nodetype: edgeserver}

{nodetype: edgedevice}

Start turtlesim teleop app in this container

Pod

Pod

Pod

Pod

turtlesim

rgt

ros2 shell

<>/

Primary node

Sony

R&D Center U.S. Laboratory
ROS 2 / Micro-ROS with KubeEdge (W.I.P)

LAN (Physical Network)

- **cloudcore**
  - Kubernetes API Server
  - Pod: {nodetype: cloudserver}
  - Kubernetes Primary Node
  - KubeEdge Cloud Core

- **edgecore**
  - Pod: {nodetype: nearedge}
  - KubeEdge Edge Core (Raspi4)

- **XRCE-DDS Client Micro-ROS**
  - Pod: {nodetype: faredge}
  - Far Edge Device (SPRESENSE)

Connections:
- ros2 talker-A to ros2 listener-A
- ros2 talker-B to ros2 listener-B
- ROS 2 Agent to ros2 listener-A
- ros2 talker-B to ros2 listener-B
- XRCE-DDS Client Micro-ROS to ros2 listener-A
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
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](#)