R5 End-to-End Validation Guide

Demo of KubeEdge Federation & EdgeMesh functionality

1. Deploy KubeEdge Federation, open source Karmada project
2. Deploy KubeEdge open source EdgeMesh project

The demo consists of two distinct parts:

- The first demo shows Karmada's multi-cluster management capabilities. Karmada (Kubernetes Armada) is a Kubernetes management system that enables cloud-native applications to run across multiple Kubernetes clusters and clouds with no changes to the underlying applications. By using Kubernetes-native APIs and providing advanced scheduling capabilities, Karmada truly enables multi-cloud Kubernetes environment.
  It aims to provide turnkey automation for multi-cluster application management in multi-cloud and hybrid cloud scenarios with key features such as centralized multi-cloud management, high availability, failure recovery, and traffic scheduling.

  The demo specifically demonstrates how federation level policies seamlessly manage deployment of workloads (Pods) across two underlying independent cloud clusters. Two sets of Kubernetes clusters are built in this demo environment: cluster1 and cluster2. Create an Nginx application. Create a PropagationPolicy. The policy defines which clusters the Resource Template needs to be scheduled in. This concept is the same as Kubefed-v2. Here the policy strategy will schedule the application to two clusters at the same time, the corresponding application specified by resource Selectors.

  Let's take a look at the corresponding environment, first switch use-context to cluster01, we can see that the corresponding Nginx application has been deployed under the cluster01, switch use-context to cluster02, we can find another Nginx application in cluster02 Has been created. Let's modify the PropagationPolicy to only allow deployment to the Cluster01, switch to the karmada-apiserver environment to modify the policy, and delete the cluster02 assignment. The result is as expected.

- Second part of the demo demonstrates the EdgeMesh functionality. The demo covers how EdgeMesh provides support for a simple network solution for the inter-communications between services at edge scenarios (east-west communication). EdgeMesh is a part of KubeEdge, and provides a simple network solution for the inter-communications between services at edge scenarios. EdgeMesh supports cross-edge communication and cross-cloud edge communication. KubeEdge is built on Kubernetes, extending cloud-native containerized application orchestration capabilities to the edge. However, in the edge computing scenario, the network topology is more complex, the edge nodes in different areas are often not interconnected, and the intercommunication of traffic between applications is the primary requirement of the business, and EdgeMesh provides a set of solutions for this.
  Next we show a use case using EdgeMesh.

  Demo Background: Map drawing on the sea, used in marine transportation, marine environment exploration and other scenes. Due to ocean parks, ocean communication restrictions and cloud access obstacles. Generally, nodes are composed of multiple oceans, and each cruise carries ship a certain number of ocean maps for sea shooting. Multiple cruise ships jointly build a maritime plan. We built two nodes as edge node. ROS-related applications, EdgeMesh and EdgeCore were deployed on each node, and CloudCore and Kubernetes master were deployed in the cloud.

  Related applications and deployment instructions have been uploaded to the docker hub. In this application, each node will perform slam simulation, and the simulated pictures of each node are different, and the pictures of different nodes will eventually be merged. The point source graph generated here is a picture taken by a car in two rooms. Both nodes A and B have generated the original point graph, and then the slave node will transmit the picture information to the master A node. At this time, the Merge is performed on the A node to generate pictures. Here node B transfers the picture to node A to use EdgeMesh for service discovery. The ROS master address configured on node B is a domain name. This is a fixed access domain name generated in EdgeMesh, and the domain name will be diverted to the corresponding Ros application on node A.

Demonstration Video

https://wiki.akraino.org/download/attachments/28968608/Federated%20Multi-Access%20Edge%20Cloud%20Platform%20R-5%20Demo.mp4?api=v2