Radio Edge Cloud Developer Documentation

Radio Edge Cloud uses a large number of Open Source components. This wiki will provide some "glue" documentation as well as hyperlinks to specific component documentation elsewhere. This page is not an exhaustive list of components, but serves as a location to incrementally accumulate developer wisdom on "good to know" stuff about the various components.

CM-Plugins

Style Guide Note

Certain class names must be all lowercase (as opposed to following PEP8 recommendation of CamelCase) due to an assumption in the framework that builds method names dynamically with the assumption of all lowercase class names as a part of the method name. Refer to https://gerrit.akraino.org/r/gitweb?p=ta/config-manager.git;a=blob;f=cmdatahandlers/src/cmdatahandlers/api/configmanager.py;h=e4aa72d470c8359288e34b50f35ba168e4b03c93;hb=HEAD#l99 for the code where the domain handler function names are built based on the handlers located in https://gerrit.akraino.org/r/gitweb?p=ta/config-manager.git;a=tree;f=cmdatahandlers/src/cmdatahandlers;hb=HEAD which results in the method names having to match the case of the directory names.

CPU Pooler

CPU Pooler for Kubernetes is a solution for Kubernetes to manage predefined, optimized, and isolated CPU pools in Kubernetes nodes.

K8s Pods can ask CPU resources from the default, shared, or exclusive pool of a properly configured and partitioned REC Node.

The project README provides additional detail and examples.

DANM - TelCo grade K8s network manager

In order to provide multiple network interfaces per Kubernetes pod as well as to support a variety of different network interface types, REC uses DANM as the primary K8s network manager component.

The main concept of DANM is providing consistent Kubernetes APIs for network management, which allows both operators and tenant users to describe network topologies in greater detail and variety that typically supported by Kubernetes.

The DANM README provides extensive documentation on this component. There are examples of YAML files including how to provide pods with SR-IOV interfaces and how to manage networks through the various, production grade APIs.

Akraino REC supports provisioning Pod interfaces via IPVLAN, MACVLAN, Flannel, or SR-IOV backends. When a Pod does not explicitly ask network connections, DANM still connects it to the default cluster network. That is, the Flannel provided VxLAN overlay.

By default Pods do not have external network access through this network connection, but from R2 onwards it is possible to configure the underlay to provide external connectivity for Pods only connecting to the default Kubernetes cluster network.

Gerrit Code Repository Overview

The code for Radio Edge Cloud is primarily derived from the Telco Appliance Blueprint Family and is stored in Gerrit under the "ta" prefix. As with most Linux Foundation projects TA and REC follow the typical usage patterns of the Gerrit code review tool for cloning repos and pushing changes with the "git review" command. The information below provides a detailed overview of the structure of the repositories. The function of the Continuous Integration (CI) system (the LF's Jenkins server) is to use the contents of these repos along with a large amount of upstream binaries (including CentOS, Kubernetes, Docker, Ansible, Disk Image Builder, and others) to build an ISO image of Radio Edge Cloud which will be deployed by Continuous Deployment systems to install and test on bare metal test systems. After successful testing, the exact same ISO can be used to deploy a fully tested and fully reproducible image onto production servers. Much of the code in the repos relates to how to build and configure all the intermediate building blocks.

How to Build a REC or Telco Appliance ISO

It is possible to build ISO images on a simple Centos VM (with only 8GB memory) and it should work easily. Please report problems on the mailing list https://lists.akraino.org/g/blueprints/topics with tag #rec

If building for the first time, one needs to install the docker-ce and libguestfs tools (exact commands and list in the ta-ci-build job config).

Workload performance management and elasticity

- Application scaling based on performance metrics
  - Check core metrics in the system
    - Metrics APIs provided by Kubernetes can be gotten by:
    - cAdvisor automatically scrapes metrics from every active pod in the system regarding to CPU and memory usage. The state of core metrics can be requested by the following: