

IEC Test Document

Introduction

This document covers both Integrated Edge Cloud Type 1 & 2.

Integrated Edge Cloud (IEC) is an important part of Akraino project. It focuses on the multi-architecture based solution on the Edge and Cloud computing areas. In this document, we will give a brief introduction about IEC test framework.

This document is divided into 3 parts. At first, we will introduce the basic infrastructure about our test framework. Secondly, we will introduce some test cases about IEC environments and user cases. Finally, we will give our testing result about it.

Akarino Test Group Information

Currently, 2 or 3 virtual hosts are deployed on arm64 platform by fuel (compass) tools. For all the baremetal servers, those must meet those hardware and software requirements. In addition, the Internet is essential for the CI platform.

Hardware Information:

CPU Architecture	Memory	Hard disk	Network
Arm64	At least 16G	500G	1Gbps (Internet essential)

Software Information:

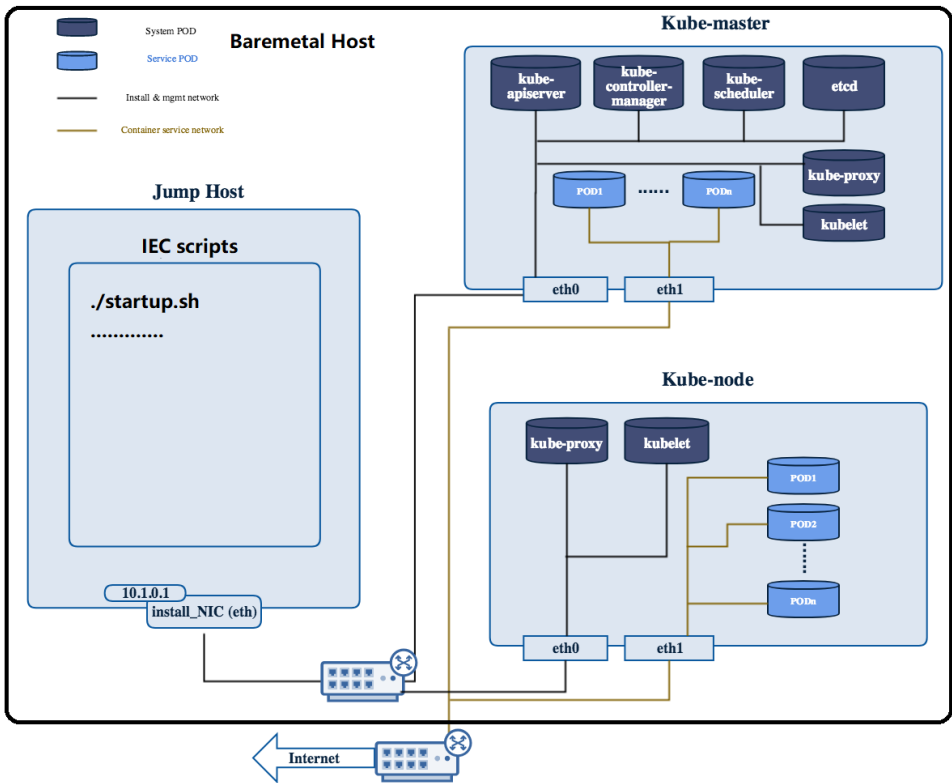
kubectl	kubeadm	Kubelet	Kubernetes-cni	Docker-ce	Calico	OS
1.13.0	1.13.0	1.13.0	0.6.0	18.06.1~ce	V3.3	Ubuntu 16.04

Overall Test Architecture

The following picture describes the overall testing environment deployed by Compass/Fuel tools. For the whole environment, there are several virtual hosts which will be deployed on baremetal host. One is jumper host which is used for running the k8s deployment scripts. Others are K8s nodes which will be deployed K8s with Calico. For each of virtual host, there are 2 NICs with 1Gbps, one is for Internet, the other is for internal connection. The specific information please reference:

<https://jenkins.akraino.org/view/iec/>

Attention: Only the Arm64 platform was deployed in community CI platform so far.



Test API description

The IEC project tests can be divided into 2 parts. The first part is about IEC platform tests which is used for testing the basic function of Kubernetes and Container Networking interface (CNI). And the second part is Feature tests which is about IEC functional test for verifying IEC user cases, just like SEBA and other user cases.

Platform tests

In IEC project, there are 2 cases in Platform test. One is about service deployment test for verifying the basic K8s network function and deployment function end to end. The other is the K8s smoke check for checking K8s environments.

Deployment case

This test will verify the K8s basic function by deploying a simple Nginx server. When Nginx server was deployed, the test will verify the reply which is getted from Nginx server for checking the basic function of K8s. And it can be started by shell scripts nginx.sh.

The Test inputs

There should be a nginx.yaml configuration files which is used for deploying the Nginx pods.

Test Procedure

The test is completed by a script which located at iec project. At first, the program will start a Nginx server based on nginx.yaml file. After the status of Nginx pods are OK, it will send a "get" request to the Nginx server port for getting the reply. If there is a reply, it will be checked and restored in local database. At last, all the resources about Nginx will be deleted by "kubectl delete" command.

Expected output

The script will get the information from the Nginx service IP. Just as follows:

```
wget -O /dev/null "http://serviceIP"
```

Test Results

If correct, it will return OK, otherwise is Error.

```
root@net-orm-thunderx2-02:/home/jingzhao/coding/iec/src/foundation/scripts# wget -O /dev/null "http://172.16.1.142"
--2019-05-16 11:23:20-- http://172.16.1.142/
Connecting to 172.16.1.142:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 612 [text/html]
Saving to: '/dev/null'

/dev/null 100%[=====] 612 --.-KB/s in 0s

2019-05-16 11:23:20 (4.39 MB/s) - '/dev/null' saved [612/612]
```

K8s Healthcheck case

The second case is K8s healthcheck which is used for checking the Kubernetes environments. It creates a Guestbook application that contains redis server, 2 instances of redis slave, frontend application, frontend service and redis master service and redis slave service. The test will write an entry into the guestbook application which will store the entry into the backend redis database. Application flow must work as expected and the data written must be available to read. In the end, you can directly run it from iec/src/foundation/scripts/functest.sh scripts shell.

Feature Tests

The feature tests are still going on developing now.

Test Step:

All the test cases has been integrated into IEC projects. You can start those tests by simple scripts on K8s-master node just as follows.

```
# source src/foundation/scripts/functest.sh <master-ip> #Functest case
# source src/foundation/scripts/nginx.sh #Nginx case
```

Test Dashboards

Single pane view of how the test score looks like for the Blue print.

Total Tests	Test Executed	Pass	Fail	In Progress
2	2	2	0	1

Bottlenecks/Errata