IEC Type1&2 Test Document for R5

Introduction

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

Integrated Edge Cloud (IEC) focuses on the multi-architecture based solution on the Edge and Cloud computing areas. In this document, we will give an overview about IEC testing done in CI/CD.

This document is a successor of IEC Test Document R1.

Testing of the IEC platform is done in CI/CD using the validation labs described here.

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, access to Internet is required 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	Kubenetes-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
1.15.2	1.15.2	1.15.2	0.7.5	19.03.4~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.
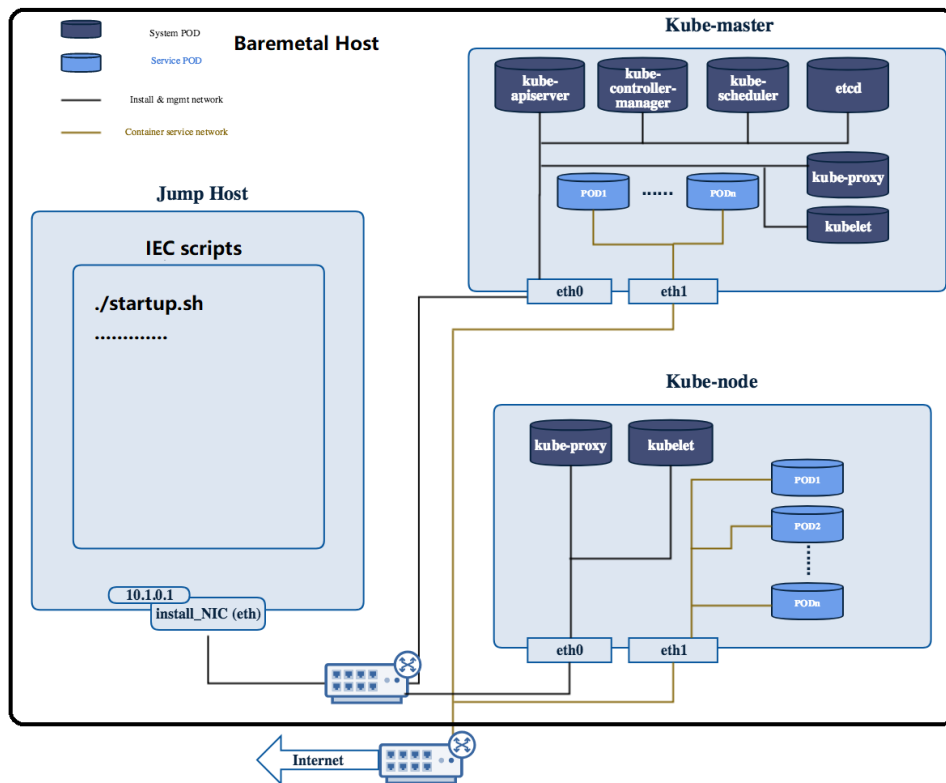
Test description

The validation of the IEC project in CI/CD consists in several steps:

1. Automatically deploy the IEC platform and perform a basic platform test
2. Run the validation project K8S conformance tests

> ⓘ Enabling this step is still work in progress for Release2. Logs for this steps are not yet pushed to Nexus, but the results can be seen in Jenkins at https://jenkins.akraino.org/view/iec/job/validation-enea-daily-master

3. Install SEBA use-case

At the end of each job the the environment is cleaned up by destroying the cluster and the networks that were created.

> ⓘ For the self-release process, only the first step is mandatory

1. Automatic deploy

The deployment is done both using Fuel@OPNFV and Compass installers.

The Installer performs the following step:

- For virtual deploys it creates the cluster VMs (using KVM). For baremetal deploys it provisions the nodes via IPMI using PXE boot
- Will clone the IEC repo and it will install the IEC platform on them as described in the Installation guide
- Will perform the Nginx Deployment test described below

The installation with Fuel@OPNF is done through this installation script.

The installation with Compass is done through this installation script.

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 verifies the basic function of K8S by deploying a simple Nginx server. After the Nginx server is deployed, the test will issue a request to the server and verify the reply got from Nginx. The test can also 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-arm-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 (not included in CI)

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

2. Conformance Test

> ⓘ Enabling this step is still work in progress for Release2. Logs for this steps are not yet pushed to Nexus, but the results can be seen in Jenkins at https://jenkins.akraino.org/view/iec/job/validation-enea-daily-master/)

The K8S conformance tests are ran using the Akraino Blueprint Validation project framework.

The Test inputs

The inputs are given via the IEC Jenkins job parameters, plus validation specific parameters:

- Bluprint name (iec)
- Layer to test (k8s)
- Version of the validation docker images to use
- Flag to run optional tests

Test Procedure

The tests are ran using the Akraino Blueprint Validation project procedure, by calling the Bluval Jenkins job

Expected output

All tests pass

```
Status: Downloaded newer image for akraino/validation:k8s-latest
==============================================================================
Conformance
==============================================================================
Conformance.Conformance :: Run k8s conformance test using sonobuoy
==============================================================================
Run Sonobuoy Conformance Test                                   | PASS |
------------------------------------------------------------------------------
Conformance.Conformance :: Run k8s conformance test using sonobuoy  | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
==============================================================================
Conformance                                                     | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
==============================================================================
Output:  /opt/akraino/results/k8s/conformance/output.xml
Log:     /opt/akraino/results/k8s/conformance/log.html
Report:  /opt/akraino/results/k8s/conformance/report.html
```

3. Install SEBA use-case

The installation of the use-case is done as recommended by the upstream CORD community.

The Test inputs

The inputs are given via the Jenkins job parameters.

Test Procedure

The script found here is used to install SEBA.

Expected output

Installation in Jenkins is successful

Tested configurations

IEC type	Lab	Installer	Installation type	OS	Logs
Type1	Arm-China	compass	baremetal	ubuntu16.04	https://nexus.akraino.org/content/sites/logs/arm-china/jenkins092/iec-type1-deploy-compass-virtual-ubuntu1604-daily-master/
Type2	Arm-China	compass	virtual	ubuntu16.04	https://nexus.akraino.org/content/sites/logs/arm-china/jenkins092/iec-type2-deploy-compass-virtual-ubuntu1604-daily-master/
Type2	Enea	fuel	virtual	ubuntu16.04	https://nexus.akraino.org/content/sites/logs/production/vex-yul-akraino-jenkins-prod-1/iec-type2-deploy-fuel-virtual-ubuntu1604-daily-master/
Type2	Enea	fuel	virtual	ubuntu18.04	https://nexus.akraino.org/content/sites/logs/production/vex-yul-akraino-jenkins-prod-1/iec-type2-deploy-fuel-virtual-ubuntu1804-daily-master/
Type2	Enea	fuel	virtual	centos7	https://nexus.akraino.org/content/sites/logs/production/vex-yul-akraino-jenkins-prod-1/iec-type2-deploy-fuel-virtual-centos7-daily-master/
Type2	Enea	compass	virtual	ubuntu16.04	https://nexus.akraino.org/content/sites/logs/production/vex-yul-akraino-jenkins-prod-1/iec-deploy-compass-virtual-ubuntu1604-daily-master/

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 |