ELIOT R5 - IoT Gateway Installation Guide

Introduction

The guide covers the installation details which are related to ELIOT Iot Gateway Blueprint.

This guide covers detailed information of the various types of deployments, detailed steps and what are the various components it will install. In addition, the guide provides information on hardware requirements, prerequisite software and minimum hardware requirements. On successful deployment, Center and Edge Nodes will be installed. The number of nodes in Center cluster and Edge node in the cluster is configurable.

The CENTER Node is a K8s Cluster and EDGE Node is a K8s Cluster worker node.

How to use this document

The document includes details of prerequisites /pre-installation, installation and uninstalls steps.

The prerequisites and pre-installation software and hardware should be ready before executing the installation steps.

In BP first release Two types of installation mechanisms are provided, as below

1. Ansible-Playbook single command
2. Command Line Interface (CLI)

Deployment Architecture

The Deployment Architecture consists of the following nodes

- One-Click Deployment Node
- ELIOT Master Node
- IotGateway Node

Note: For Development environment two nodes is sufficient, where one node plays a dual role of One-Click Deployment Node and Master Node with other as IotGateway Node.

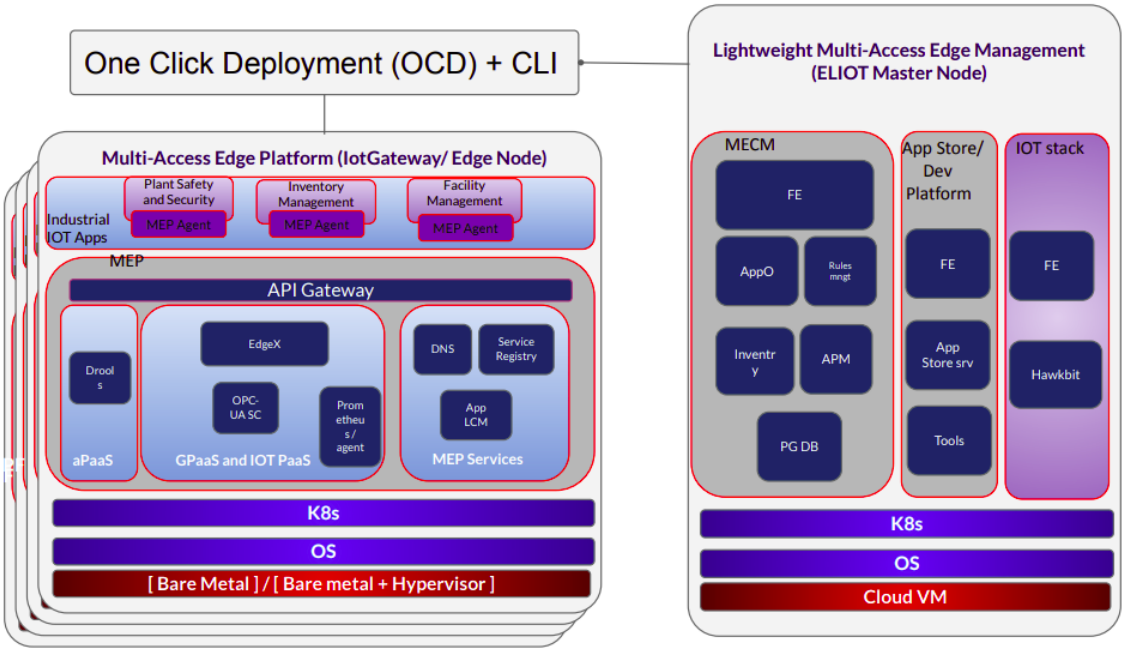


Figure: ELIOT Deployment Architecture

Note: *ELIOT IotGateway Blueprint Deployment has been tested on Cloud VM and is not tested on Bare-Metal Environment. Though, theoretically deployment should work in bare metal, provided hardware and software prerequisites are met. Kindly refer R5 - Test Documentation of Edge Lightweight IotGateway (ELIOT) to get details on the tested deployment.*

Pre-Installation Requirements

Hardware Requirements

The number of Hardware requirements depends mainly on the Use Case Scenario and the enterprise scale. A use case can have one Deployment node, ELIOT Master or controller node with one or multiple IotGateway nodes.

The minimum number of nodes required for a complete ELIOT Topology is 2. (Bare-Metal or Virtual Machines)

1) Deployment Node

2) ELIOT Master

3) ELIOT IotGateway node

Note: The Hardware details provided are of Virtual Machine configurations.

Minimum Hardware Requirements

ELIOT Master Node	
HW Aspect	Requirements
# of Node(s)	A virtual machine hosted in any Cloud Provider having internet connectivity.
# of CPU	8
Architecture	x86_AMD64 or ARM64.
RAM	8 GB
Disk	120 GB ~ 512GB
Networks	1

IOTGateway Node(s)	
HW Aspect	Requirements
# of Node(s)	1 MEC Host
# of CPU	4
Architecture	x86_AMD64 or ARM64.
RAM	4 GB
Disk	20 GB ~ 256 GB
Network	1

Note: The above specifications are given considering the ELIOT CI / CD environment. User can try lower configuration considering lightweight components being used.

Recommended Hardware Requirements

ELIOT Master Node	
HW Aspect	Requirements
# of Node(s)	A virtual machine hosted in any Cloud Provider having internet connectivity.
# of CPU	8
Architecture	x86_AMD64 or ARM64.
RAM	8 GB
Disk	120 GB ~ 512GB
Networks	1

IOTGateway Node(s)	
HW Aspect	Requirements
# of Node(s)	1 MEC Host
# of CPU	4
Architecture	x86_AMD64 or ARM64.
RAM	4 GB
Disk	20 GB ~ 256 GB
Network	1

Software Prerequisites

- Virtual Machines preinstalled with Ubuntu 18.04 for MECM Node.
- Virtual Machines preinstalled with Ubuntu 18.04 for MEC Host Nodes
- root user created in the Deployment Node, MEC Node and MEC Host Node.
- SSH Server running in all the Nodes.
- Ansible > 2.10.7 installed in One Click Deployment Node (Jump Host)
- git installed in Jump Host.

Database Prerequisites

Schema scripts

N/A

Other Installation Requirements

Jump Host Requirements

Network Requirements

- Internet connectivity in OCD Host, ELIOT Master and IOTGateway Nodes.
- The ELIOT Master Node and EDGE/IotGateway Node should be able to ping each other.

Bare Metal Node Requirements

N/A

Execution Requirements (Bare Metal Only)

N/A

Installation High-Level Overview

The blueprint provides one click deployment and command-line interface for installing the ELIOT blueprint components.

Bare Metal Deployment Guide

Install Bare Metal Jump Host

Note: ELIOT Blueprint Deployment has been tested on Huawei Cloud Virtual Machines and is not tested on Bare-Metal Environment.

Though theoretically deployment should run successfully in bare metal too provided hardware and software prerequisites are met.

Creating a Node Inventory File

N/A

Creating the Settings Files

N/A

Running

N/A

Virtual Deployment Guide

For Virtual Deployment minimum **2** Virtual machines, following are the virtual machines(OCD and Master on same node) and their usage

No	Usage
1	One Click Deployment Node
2	ELIOT Master Node
3	IotGateway Node

All the nodes should have internet connectivity , network interface and network connectivity between the VM's.

In this release to install the ELIOT environment.

i) ELIOT Deployment using Ansible-Playbook single command

Standard Deployment Overview

Jump Host Software Installations:

Login to the Jump Host and perform the below steps:

1. Install Ansible > 2.10.7
2. Install git
3. Install python3 and pip3

Jump Host Pre-Configurations for MECM Components Installation

Login to the Jump Host and perform the below configuration steps (Steps : as below-

1. Generate public key : #ssh-keygen -t rsa

2. Setup password-less login from ocd to center and ocd to edge.

sshpass -p <password> ssh-copy-id -p <ssh-port> -o StrictHostKeyChecking=no root@<node_ip>

3. Review and Change Parameters

For EdgeGallery MUNO Mode

eliot/blueprints/iotgateway/playbooks/muno-config/controller/hosts-muno-controller

eliot/blueprints/iotgateway/playbooks/muno-config/controller/var.yml

eliot/blueprints/iotgateway/playbooks/muno-config/edge/hosts-muno-controller

eliot/blueprints/iotgateway/playbooks/muno-config/edge/var.yml

For EdgeGallery AIO Mode

eliot/blueprints/iotgateway/playbooks/hosts-aio

For ELIOT stack:

*eliot/blueprints/iotgateway/*playbooks/eliot-inventory.ini

Installing Mode : ELIOT using Ansible-Playbooks

1. git clone the eliot repo, to download the software to install the ELIOT Environment.

```
root@akraino-mec-0001:~# git clone "https://gerrit.akraino.org/r/eliot"
```

2. go to the below directory

```
root@akraino-mec-0001:~# cd eliot/blueprints/iotgateway/playbooks
```

3. Modify the Configuration File : eliot-inventory.ini with the details of Master and Edge/IotGateway Nodes.

```
root@akraino-mec-0002:~# vi eliot-inventory.ini
```

4. Modify other configuration files
Edit the configuration file so that the IP addresses of the OCD, ELIOT Master Node, and IotGateway Node are <ocp_ip>, <master_ip>, and <edge_ip> respectively.
An example of the description of the configuration file when the admin password of HARBOR is <password> is shown below.

```
$ eliot/blueprints/iotgateway/playbooks/muno-config/controller/hosts-muno-controller
[master]
<master_ip>


$ eliot/blueprints/iotgateway/playbooks/muno-config/controller/var.yml

HARBOR_ADMIN_PASSWORD: <password>

# Could be true or false
# true: Deploy k8s NFS Server to keep the persistence of all pods' data
# false: No need to keep the persistence of all pods' data

# ENABLE_PERSISTENCE: true
ENABLE_PERSISTENCE: false

# ip for portals, will be set to private IP of master node default or
# reset it to be the public IP of master node here
# PORTAL_IP: xxx.xxx.xxx.xxx
PORTAL_IP: <master_ip>


$ eliot/blueprints/iotgateway/playbooks/muno-config/edge/hosts-muno-edge
[master]
<edge_ip>


$ eliot/blueprints/iotgateway/playbooks/muno-config/edge/var.yml

HARBOR_ADMIN_PASSWORD: <password>

# Could be true or false
# true: Deploy k8s NFS Server to keep the persistence of all pods' data
# false: No need to keep the persistence of all pods' data
ENABLE_PERSISTENCE: false

# ip for portals, will be set to private IP of master node default or
# reset it to be the public IP of master node here
# PORTAL_IP: xxx.xxx.xxx.xxx

# NIC name of master node
# If master node is with single NIC, not need to set it here and will get
# the default NIC name during the run time
# If master node is with multiple NICs, should set it here to be
# 2 different NICs
# EG_NODE_EDGE_MP1: eth0
# EG_NODE_EDGE_MM5: eth0

OCD_IP: <ocp_ip>
```

5.Send the git clone file from OCD to ELIOT Master Node and IotGateway Node.

```
$ scp -r ./eliot <node_ip>:~
```

For Edge Gallery installation:

Select one of Multi Node Inventory Mode (MUNO-Mode) or All in one mode (AIO mode) and install it.

MUNO-Mode:

Execute the below command:

cd eliot/blueprints/iotgateway/playbooks

ansible-playbook -i muno-config/controller/hosts-muno-controller eliot-eg-muno-controller.yml --extra-vars "operation=install" -e "ansible_user=root"

ansible-playbook -i muno-config/edge/hosts-muno-edge eliot-eg-muno-edge.yml --extra-vars "operation=install" -e "ansible_user=root"

For AIO mode:

Execute the below command

cd ealt-edge/ocd/infra/playbooks

root@akraino-mec-0001:~#ansible-playbook eliot-eg-aio-latest.yml -i hosts-aio --extra-vars "operation=install" -e ansible_user=root"

FOR ELIOT Stack:

Execute the below command
Setup environment -

root@akraino-mec-0001:~# ansible-playbook eliot-all.yml -i eliot-inventory.ini --extra-vars "operation=install"

Once the execution is completed in console will see prompt "ELIOTEdge Environment Installed , Components Install ELIOT Master and EDGE Nodes Successfully"

Snapshot Deployment Overview

N/A

Special Requirements for Virtual Deployments

N/A

Install Jump Host

N/A

Verifying the Setup - VM's

N/A

Upstream Deployment Guide

Upstream Deployment Key Features

N/A

Special Requirements for Upstream Deployments

N/A

Scenarios and Deploy Settings for Upstream Deployments

N/A

Including Upstream Patches with Deployment

N/A

Running

N/A

Interacting with Containerized Overcloud

N/A

Verifying the Setup

Verifying ELIOT IotGateway Deployment

Currently the verification is manually done.

1. Login to the Master Node and check whether K8S cluster is installed.
2. Check the below mentioned components and services are running as Pods / Services in Kubernetes cluster
    a. Edge Gallery
    b. grafana
    c. rabbitmq
    d. cadvisor
    e. edgex
    f. Hawkbit
    g. opc-ua
3. Login to Edge Host and verify the worker node setup

Components and Services running in ELIOT Master Node

Components and Services running ELIOT IotGateway/ Edge Node

Deploy Application in ELIOT

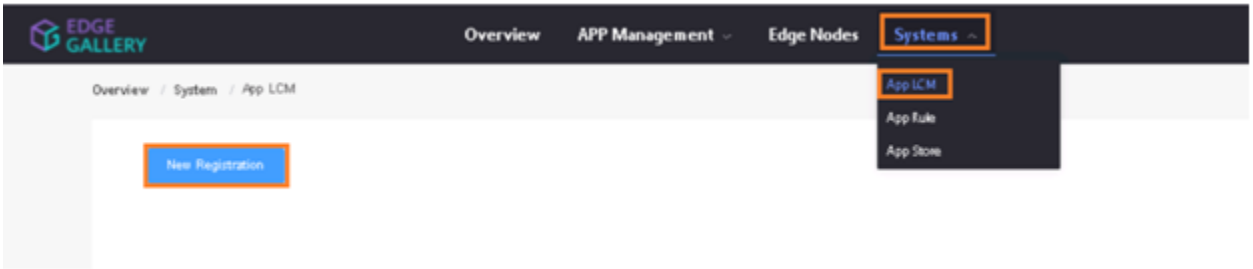
1. Login to MECM Portal https://ip:30093

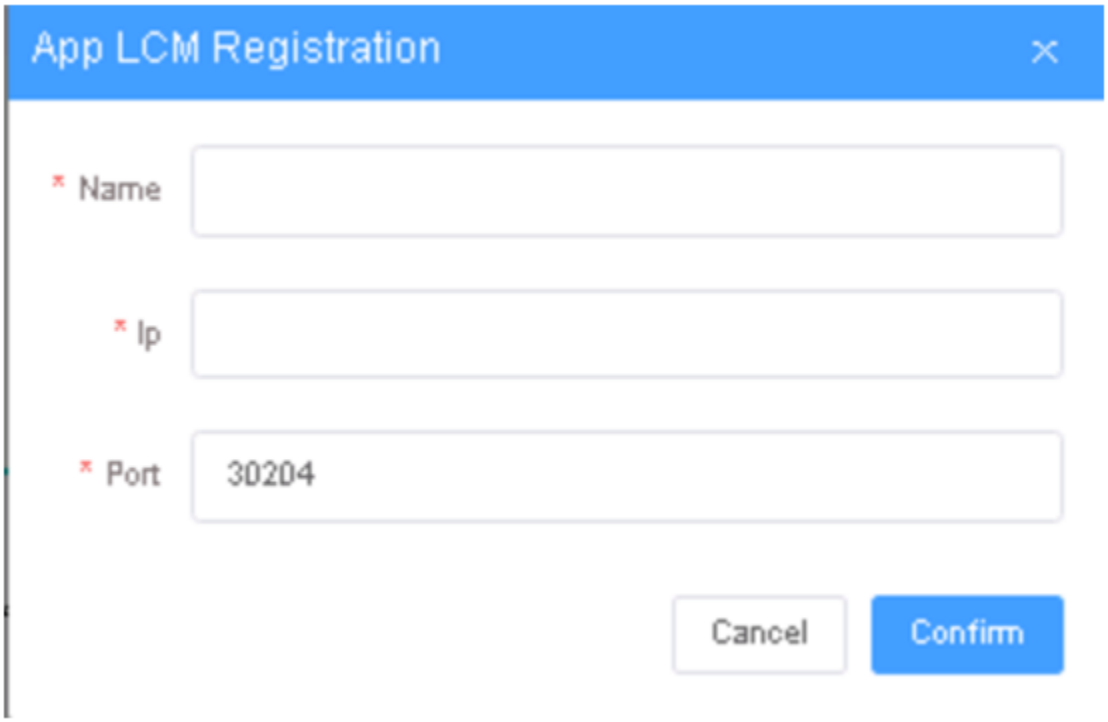
    1.1 click on **Systems ->App LCM ->New Registration**

    Name: Applcm(any general name)

    IP: applcm"public ip"

    Port: 30204



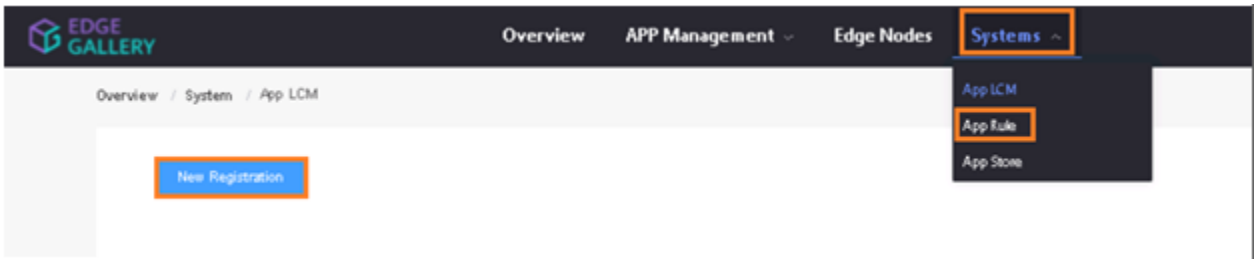


1.2. Click on **Systems ->App Rule -> New Registration**

Name: Apprule(any general name)

IP: applcm"public ip"

Port: 30206



1.3. click on **Systems ->App Store ->New Registration**

App Store Name: appstore(any general name)

IP: Appstore public ip

Port: 30099

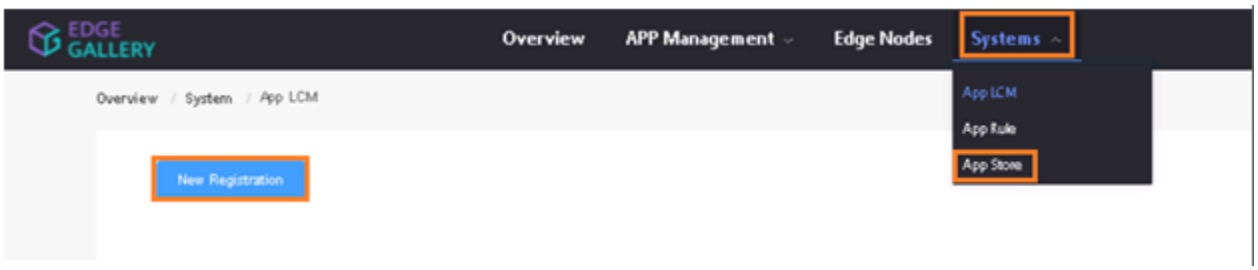
Appstore Repo: {HarborIP:443}(192.168.1.1:443)

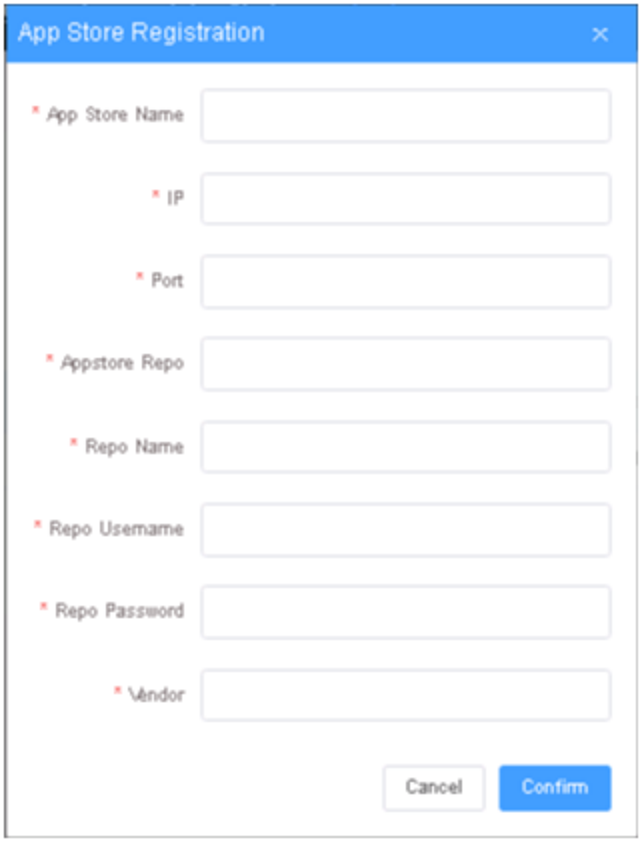
Repo Name: appstore(any general name)

Repo Username: admin(harbor user name)

Repo Password: Harbor12345(harbor password)

Vendor: vendor(any general name)





2. Login to Developer Portal https://ip:30092

2.1. Add sandbox env to deploy application before publish

Click System ->Host Management ->Add Host

Name: general name

System: k8s

Lcmip: sandbox ip(for testing purpose can provide edge ip, if no sandbox env)

mecHost: sandbox ip(for testing purpose can provide edge ip, if no sandbox env)

Port: 31252

Protocol: https

Architecture: X86

Status: Normal

Port Range: leave as it is

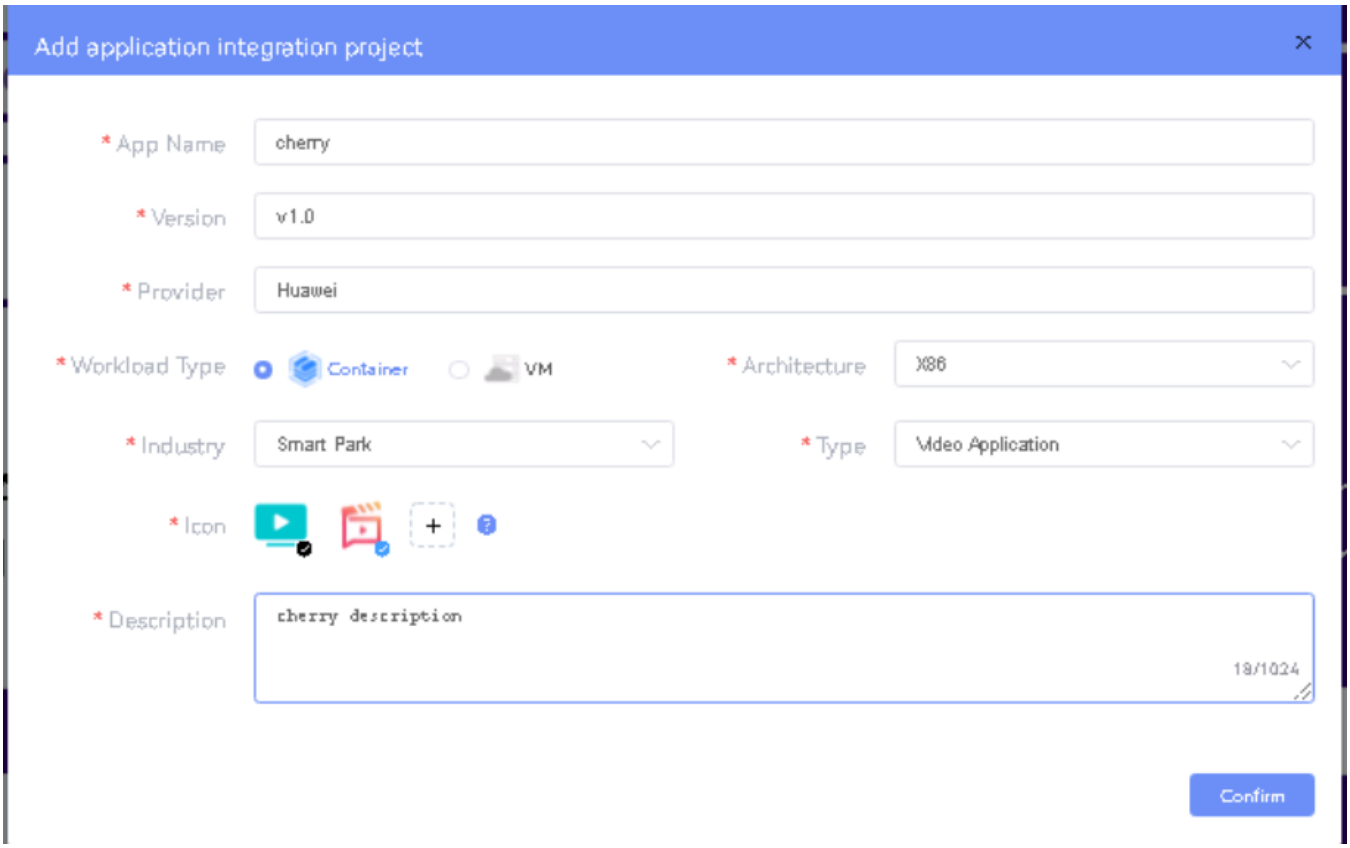
Address: Bangalore

UploadConfig File: upload sandboxenvkubeconfig file

2.2 Click on **Workspace** -> **Create Project -> Application Integration -> Start**

- Provide **App Name, Version, Provider, Workload Type, Architecture, Industry, Type.**

- Upload **Icon,** provide **Description.** And click on **confirm.**



2.3. Now click on **Deployment Test.**

- Upload Docker images directly from portal by clicking on **Upload App Image**

  Or, directly push Docker images to Harbor repo (takes lesser time, preferred)

- Click **next**, upload deployment yaml file now.
- After config upload, click **next** and click **start deployment**
- After Deployment is success, click on Release Recourses

Note:

- While Deployment test if any error happens, open ATP portal (https://ip:30094) in another tab of browser, sing in, come back to developer portal and re run deployment test
- gitee.com/edgegallery/applications repo provides, A lot of applications with their logo, deployment yaml & user guides

2.4. Now click on **Application Release**

Upload file for **Application Description**

Click **save config**, click **Next Step**, click **Start Test,** scroll down to find & click **Start Test** button, click **Next Step,** click **publish** to publish application to appstore.

3. Login to MECM Portal https://ip:30093

3.1. **Add k8s node:**

Click on **Edge Nodes ->New Rgistration**

VM: k8s

Name: edge1

IP: edge public ip

Location: select from drop down

Address: yanta

Coordinates: 116.39,39.90

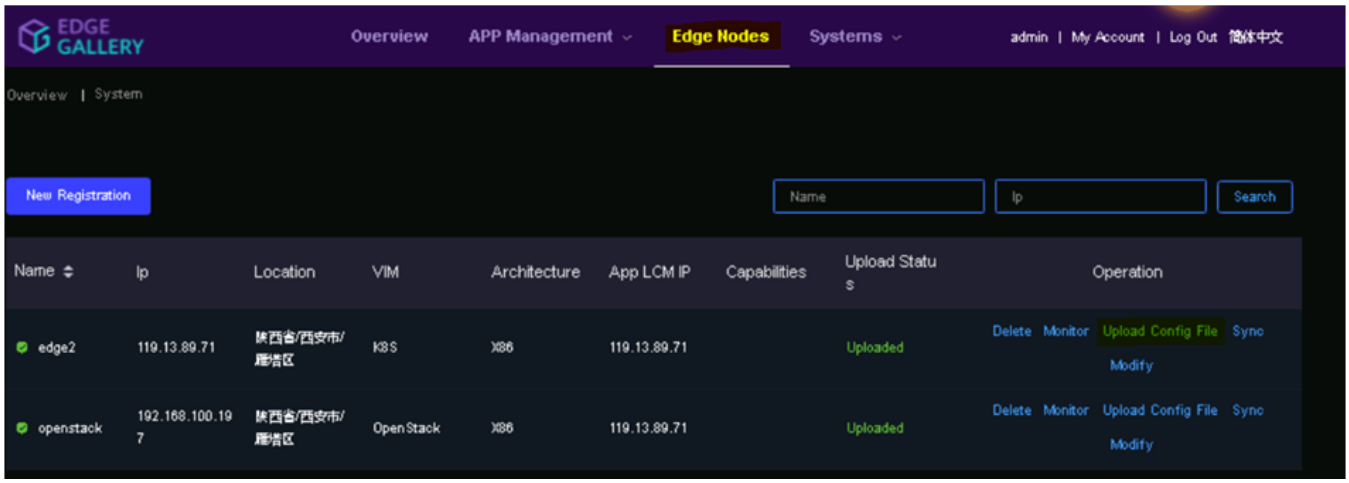
Architecture: x86

Capabilities: select none

MEPM: select applcm node from dropdown

3.2. Download /root/.kube/config file from edge node

And click on **Upload config file** to upload.



Developer Guide and Troubleshooting

Uninstall Guide

Using Ansible Playbooks

root@akraino-mec-0001:~#ansible-playbook eliot-all-uninstall.yml -i eliot-inventory.ini --extra-vars "operation=uninstall"

For MUNO Mode

root@akraino-mec-0001:~#ansible-playbook -i muno-config/controller/hosts-muno-controller eliot-eg-muno-controller.yml --extra-vars "operation=uninstall" -e "ansible_user=root"

root@akraino-mec-0001:~#ansible-playbook -i muno-config/edge/hosts-muno-edge eliot-eg-muno-edge.yml --extra-vars "operation=uninstall" -e "ansible_user=root"

For AIO Mode

root@akraino-mec-0001:~#ansible-playbook -i hosts-aio eliot-eg-aio-latest.yml --extra-vars "operation=uninstall" -e "ansible_user=root"

Troubleshooting

Error Message Guide

N/A

Maintenance

Blueprint Package Maintenance

Software maintenance

N/A

Hardware maintenance

N/A

Blueprint Deployment Maintenance

N/A

Frequently Asked Questions

N/A

License

License information of ELIOT Blueprint Components

ELIOT Master Node

S. No	Software	Type	Version	License	Remarks
1.	Docker	CRI	18.09	Apache 2.0 *license*	No code modifications done
2.	Kubernetes	Orchestration	v1.18.7	Apache 2.0 *license*	No code modifications done
3.	Edge Gallery	Open Source MEC Platform	1.1.1	Apache 2.0 *license*	Open Source MEC Platform
4.	Grafana	Monitoring	7.1.1	Apache 2.0 *license*	

EDGE / IotGateway Node

S. No	Software	Type	Version	License Information	Remarks
1.	Docker	CRI	18.09	Apache 2.0 *license*	No code modifications done
2.	K8s	Orchestration	1.18.7	Apache 2.0 *license*	No code modifications done
3.	Edge Gallery	Open Source MEC Platform	1.1.1	Apache 2.0 *license*	No code modifications done
4.	cAdvisor	Container Metrics	v0.36.0	Apache 2.0 *license*	No code modifications done
5.	RabbitMQ	Message Broker	3.7	Mozilla Public *License*	No code modifications done. RabbitMQ image is deployed as is.
6.	Prometheus	Metrics Collector	9.3.1	Apache 2.0 *license*	Code part of Edge Gallery
7.	OPC-UA	IoT Protocol	Geneva	Apache 2.0 *license*	Upstream
11	EdgeX	Services	Edinburgh	Apache 2.0 *license*	Upstream

References

Definitions, acronyms and abbreviations

Abbreviations

- ELIOT - Edge Lightweight IotGateway
- MECM - Multi Access Edge Computing Manager.
- MEC - Multi Access Edge Computing.
- MEP - Multi Access Edge Platform.