ELIOT R4 - 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 K3s Cluster.

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.

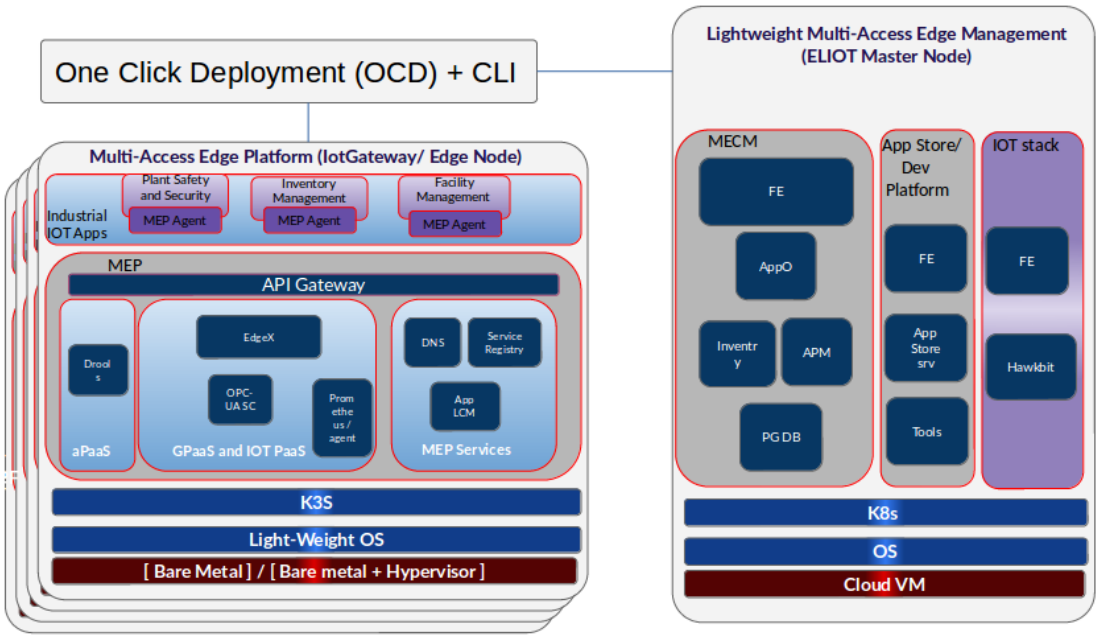

Figure: ELIOT Deployment Architecture

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 R4 - 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 three. (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 16.04 /18.04 for MECM Node.
- Virtual Machines preinstalled with Ubuntu 16.04 / 18.04 or CentOS 7.X 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 > 2.5 installed in One Click Deployment Node (Jump Host)
- git installed in Jump Host.
- Kubespray code is downloaded (https://github.com/kubernetes-sigs/kubespray.git)
- GO Lang (version greater than 1.14.2) installed in Jump Host, required for CLI.

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 **three** Virtual machines, following are the virtual machines 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 > 2.9.6 [https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html]
2. Install git
3. Install GoLang > 1.14+
4. Git clone the Kubespray repo. [https://github.com/kubernetes-sigs/kubespray.git]
5. Install python3 and pip3
6. cd kubespray && pip install -r requirements.txt

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

2. Copy the ssh key to all the nodes in the MECM Cluster, using ssh-copy-id. (example : #ssh-copy-id root@159.178.17.16)

3. Kubespray configuration

```
cd kubespray && cp -rfp inventory/sample inventory/mycluster
```

Update ansible inventory file provided in kubesrpay repo with MECM Cluster node IP's

4. Review and Change Parameters under inventory/mycluster/group_vars

```
cat inventory/mycluster/group_vars/all/all.yml
cat inventory/mycluster/group_vars/k8s-cluster/k8s-cluster.yml
```

5. Alter the config.yml and change parameters under path eliot/blueprints/iotgateway/playbooks/config.yml

```yaml
######### Common parameters #########
######### Mandatory #########
private_repo_ip:
  name:

######### Optional #########
eg_image_tag:
  name: 0.9
# User can either give common pwd or can opt to provide individual pwds
common_pwd:
  name:

########## Edge config #############
######### Mandatory #########
mep_kong_pg_pwd:
  name:
######### Optional #########
edge_management_interface:
  name: eth0
edge_dataplane_interface:
  name: eth1
eg-management-address:
  name: 100.1.1.2/24
eg-dataplane-address:
  name: 200.1.1.2/24
```

```yaml
# All Edge related password which needs to be specified if user doesn't
# need common password for security purpose
mep_pg_admin_pwd:
  name:
mep_cert_pwd:
  name:
generate_cert_pass:
  name:
mecm_mepm_postgresPassword:
  name:
mecm_mepm_postgresLcmCntlrPassword:
  name:
mecm_mepm_postgresk8sPluginPassword:
  name:

########### Center config ###########
######### Mandatory #########
mecm_meo_edgeRepoUserName:
  name:
mecm_meo_edgeRepoPassword:
  name:
```

```
######### Optional #########
usermgmt_port:
  name: 30067
appstore_port:
  name: 30091
developer_port:
  name: 30092
mecm_port:
  name: 30093
docker_registry_port:
  name: 5000
prometheus_node_port:
  name: 30009

# All Center related password which needs to be specified if user
# doesn't need common password for security purpose
user_mgmt_encryptPassword:
  name:
mecm_meo_keystorePassword:
  name:
mecm_meo_truststorePassword:
  name:
mecm_meo_postgresPassword:
  name:
mecm_meo_postgresApmPassword:
  name:
mecm_meo_postgresAppoPassword:
  name:
mecm_meo_postgresInventoryPassword:
  name:
```

Installing Mode : ELIOT using Ansible-Playbooks

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

root@akraino-mec-0002:~# git clone "https://gerrit.akraino.org/r/eliot"

2. go to the below directory

root@akraino-mec-0002:~# 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. Execute the below command
Setup environment -

root@akraino-mec-0002:~# 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"

Other Options:

To Install only Edge node
root@akraino-mec-0002:~# ansible-playbook eliot-all.yml -i eliot-inventory.ini --tags "edge" --extra-vars "operation=install"

Snapshot Deployment Overview

Not Applicable

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. PostgresSQL
    b. AppLCM
    c. Appo
    d. Inventory
    e. Apm
    f. MECM - FrontEnd
    g. Appstore
    h. Developer Portal
    i. Service Center
    j. User Management
    k. Hawkbit
3. Login to Edge Host and check K3S is installed.

Components and Services running in ELIOT Master Node

```
root@[redacted]:~# kubectl get pods
NAME                                      READY   STATUS    RESTARTS   AGE
appstore-be-0                             1/1     Running   0          20m
appstore-be-postgres-0                    1/1     Running   0          20m
appstore-fe-5854976dcf-zc6d4              1/1     Running   0          20m
developer-be-0                            1/1     Running   0          20m
developer-be-postgres-0                   1/1     Running   0          20m
developer-fe-7bb8b865bf-gtmgk             1/1     Running   0          20m
kubernets-deployment-77698bff7d-5qcch     1/1     Running   0          17m
mec-grafana-5fc44fc96d-qt4gz              1/1     Running   0          20m
mecm-apm-6fddb675b-fjr45                  1/1     Running   0          20m
mecm-appo-7cf4d8745f-sjqk6                1/1     Running   1          20m
mecm-fe-64f4786f4f-qj69z                  1/1     Running   1          20m
mecm-inventory-75876d8f45-tb4sf           1/1     Running   1          20m
mecm-postgres-0                           1/1     Running   0          20m
service-center-5cc4dcb6f7-7xhhb           1/1     Running   0          20m
user-mgmt-6df4db56b8-9zstk                1/1     Running   0          20m
user-mgmt-postgres-0                      1/1     Running   0          20m
user-mgmt-redis-0                         1/1     Running   0          20m
root@[redacted]:~# kubectl get svc
NAME                        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
appstore-be-postgres-svc    ClusterIP   10.233.30.3     <none>        5432/TCP         20m
appstore-be-svc             ClusterIP   10.233.6.69     <none>        8099/TCP         20m
appstore-fe-svc             NodePort    10.233.0.5      <none>        8443:30091/TCP   20m
developer-be-postgres-svc   ClusterIP   10.233.25.16    <none>        5432/TCP         20m
developer-fe-svc            NodePort    10.233.39.242   <none>        8443:30092/TCP   20m
kubernetes                  ClusterIP   10.233.0.1      <none>        443/TCP          28m
mec-grafana                 NodePort    10.233.38.81    <none>        80:30000/TCP     20m
mecm-apm                    NodePort    10.233.47.191   <none>        8092:30202/TCP   20m
mecm-appo                   NodePort    10.233.56.131   <none>        8091:30201/TCP   20m
mecm-fe-svc                 NodePort    10.233.61.62    <none>        8443:30093/TCP   20m
mecm-inventory              NodePort    10.233.4.18     <none>        8093:30203/TCP   20m
mecm-postgres               ClusterIP   10.233.19.83    <none>        5432/TCP         20m
service-center              ClusterIP   10.233.49.162   <none>        30100/TCP        20m
user-mgmt-postgres-svc      ClusterIP   10.233.22.178   <none>        5432/TCP         20m
user-mgmt-redis-svc         ClusterIP   10.233.54.183   <none>        6379/TCP         20m
user-mgmt-svc               NodePort    10.233.16.0     <none>        8067:30067/TCP   20m
```

Components and Services running ELIOT IotGateway/ Edge Node

```
root@[redacted]3:~# kubectl get pods
NAME                                                READY   STATUS    RESTARTS   AGE
mep-prometheus-kube-state-metrics-664f799f67-nlxp8  1/1     Running   0          5h17m
mep-prometheus-node-exporter-j7gcm                  1/1     Running   0          5h17m
cadvisor                                            1/1     Running   0          5h17m
mep-prometheus-pushgateway-974bb5cb7-fbnxs          1/1     Running   0          5h17m
mepm-postgres-0                                     1/1     Running   0          5h16m
rabbitmq-0                                          1/1     Running   0          5h17m
mecm-mepm-lcmcontroller-767986fc6f-fll19            1/1     Running   0          5h16m
mecm-mepm-k8splugin-6cd6994685-lqddz                1/1     Running   0          5h16m
mep-prometheus-server-9996ddd7f-c29vd               2/2     Running   0          5h17m
mep-prometheus-alertmanager-b49844896-hqhgq         2/2     Running   0          5h17m
rabbitmq-1                                          1/1     Running   0          5h16m
rabbitmq-2                                          1/1     Running   0          5h15m
kubernets-deployment-7494ddbf6c-tfh4v               1/1     Running   0          5h13m
root@[redacted]:~# kubectl get svc
NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                          AGE
kubernetes                          ClusterIP   10.43.0.1       <none>        443/TCP                          5h17m
rabbitmq                            NodePort    10.43.62.195    <none>        15672:31672/TCP,5672:30672/TCP   5h17m
mep-prometheus-node-exporter        ClusterIP   None            <none>        9100/TCP                         5h17m
mep-prometheus-kube-state-metrics   ClusterIP   None            <none>        80/TCP                           5h17m
mep-prometheus-server               ClusterIP   10.43.36.63     <none>        80/TCP                           5h17m
mep-prometheus-alertmanager         ClusterIP   10.43.57.199    <none>        80/TCP                           5h17m
mep-prometheus-pushgateway          ClusterIP   10.43.184.131   <none>        9091/TCP                         5h17m
mecm-mepm-lcmcontroller             NodePort    10.43.103.162   <none>        8094:30204/TCP                   5h16m
mepm-postgres                       ClusterIP   10.43.154.253   <none>        5432/TCP                         5h16m
mecm-mepm-k8splugin                 NodePort    10.43.14.158    <none>        8095:30205/TCP                   5h16m
```

Developer Guide and Troubleshooting

Uninstall Guide

Using Ansible Playbooks

root@akraino-mec-0002:~#ansible-playbook eliot-all-uninstall.yml -i eliot-inventory.ini --extra-vars "operation=uninstall"
root@akraino-mec-0002:~#ansible-playbook eliot-all-uninstall.yml -i eliot-inventory.ini --tags "edge" --extra-vars "operation=uninstall"

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

OCD Host

S. No	Software	Type	Version	License	Remarks
1.	Kubespray	K8S Tool	2.14.2	Apache 2.0 *license*	No code modifications done
2.	Helm	Application Package Manager	3.0.2	Apache 2.0 license	No code modifications done

ELIOT Master Node

S. No	Software	Type	Version	License	Remarks
1.	Docker	CRI	19.03+	Apache 2.0 *license*	No code modifications done
2.	Kubernetes	Orchestration	v1.17.2	Apache 2.0 *license*	No code modifications done
3.	Helm	Application Package Manager	3.0.2	Apache 2.0 *license*	No code modifications done
4.	Grafana	Monitoring MEC-Graphana	7.1.1	Apache 2.0 *license*	Code part of Edge Gallery
5.	Calico	CNI Plugin	3.16.5	Apache 2.0 *license*	No code modifications done
6.	PostgresSQL	MECM-Service	12.3	*PostgreSQL License*	Code part of Edge Gallery
7.	AppLCM	MECM-Service	0.9	Apache 2.0 *license*	Code part of Edge Gallery
8.	Appo	MECM-Service (MECM-Appo)	0.9	Apache 2.0 *license*	Code part of Edge Gallery
9	Inventory	MECM-Service	0.9	Apache 2.0 *license*	Code part of Edge Gallery
10	Apm	MECM-Service	0.9	Apache 2.0 *license*	Code part of Edge Gallery
11	User Management	Part of Center Node	0.9	Apache 2.0 *license*	Code part of Edge Gallery
12	MECM - FrontEnd	MECM-Service	0.9	Apache 2.0 *license*	Code part of Edge Gallery
13.	Appstore	Service (Part of Center Node)	0.9	Apache 2.0 *license*	Code part of Edge Gallery
14.	Developer Portal	Service (Part of Center Node)	0.9	Apache 2.0 *license*	Code part of Edge Gallery
15	Service Center	Service (Part of Center Node)	0.9	Apache 2.0 *license*	Code part of Edge Gallery

16	Hawkbit	Kubernetes Pod	latest container 0.3.0M6	Apache 2.0 *license*	

EDGE / IotGateway Node

S. No	Software	Type	Version	License Information	Remarks
1.	Docker	CRI	19.03+	Apache 2.0 *license*	No code modifications done
2.	K3S	Orchestration	1.19.4+	Apache 2.0 *license*	No code modifications done
3.	Helm	Application Package Manager	3.0.2	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	mepm-postgres	Service Database	12.3	*PostgreSQL License*	Code part of Edge Gallery
8	MEP	Pod	0.9	Apache 2.0 *license*	Code part of Edge Gallery
9	MECM-MEPM	MEPM-Service	0.9	Apache 2.0 *license*	Code part of Edge Gallery
10	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.