# ELIOT Installation Guide

## Introduction

This guide covers both ELIOT IOT Gateway and ELIOT SD-WAN Use cases.

This guide provides the information for setting up the ELIOT Environment which includes ELIOT Manager and ELIOT Edge Node(s) with details of software and hardware specifications.

For Release-1.0 the scope of the guide covers two ELIOT deployments one is using Kubernetes and the other one using Kubeedge (Lightweight Kubernetes).

The ELIOT Manager can be any Cloud Virtual Machine having Ubuntu OS 16.04 and should be able to ping and SSH to the Eliot Edge Node.  The ELIOT Edge Node will be a Virtual Machine having Ubuntu 0S 16.04.

For Kubernetes setup Kubeadm tool is being used and for Container Networking Interface Calico plugin is being used.

For KubeEdge the connection between the ELIOT Manager and ELIOT Edge Node is being established via Web Socket. Currently KubeEdge does not support Container Networking Interface.

# 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.

For Installation, there are two mode Auto Deployment installation script steps and manual installation.

# Deployment Architecture

For Release 1.0 the Deployment Architecture consists of two nodes  ELIOT Manager and a single ELIOT Edge Node.

ELIOT Manager acts as the Controller Node managing the ELIOT Edge Node(s) and the containerized applications running on the various ELIOT Edge Nodes.

# Pre-Installation Requirements

## Hardware Requirements

> ⚠ Number of Hardware requirements depends mainly on the Use Case Scenario. A use case can have one to multiple ELIOT Edge Nodes
>
> Note: For ELIOT Release 1.0 ELIOT Manager is acting a dual role Jump server (to execute the setup script) and as the Kubernetes Master. Also currently the below hardware capability are mentioned for the Virtual Machines.

ELIOT Manager - One VM - 8 vCPU - 32 GB RAM (can be hosted in any cloud provider) with Ubuntu OS having internet and network connectivity with ELIOT Edge Node.

ELIOT Edge Node - Ubuntu OS 16.04 / 18.04 installed with networking capability and internet capability.

### Minimum Hardware Requirements

| **ELIOT Manager** | |
|---|---|
| HW Aspect | Requirements |
| # of Node(s) | A virtualized machine hosted in any Cloud Provider having internet connectivity. |
| CPU | Minimum 1 socket x86_AMD64 or ARM64 with Virtualization support. |
| RAM | 4 GB, it depends on the number of ELIOT Edge nodes which will be managed by ELIOT Manager.<br>For Deployment in Development environment 8 GB is sufficient. |
| Disk | 120 GB ~ 512GB (Depending on the overall ELIOT Topology) |
| Networks | 1 |

| **ELIOT Edge Node(s)** | |
|---|---|
| HW Aspect | Requirements |
| # of Node(s) | 1 Eliot Edge Node |
| CPU | Minimum 1 socket x86_AMD64 or ARM64. |
| RAM | 1 GB, it depends on the number of ELIOT Edge nodes which will be managed by ELIOT Manager.<br>For Deployment in Development environment 4 GB is sufficient. |
| Disk | 20 GB ~ 256 GB |
| Network | 1 |

### Recommended Hardware Requirements

| ELIOT Manager | |
|---|---|
| HW Aspect | Requirements |
| cc | A virtualized machine hosted in any Cloud Provider having internet connectivity. |
| CPU | Minimum 1 socket x86_AMD64 or ARM64 with Virtualization support. |
| RAM | 8 GB ~ 64 GB it depends on the number of ELIOT Edge nodes which will be managed by ELIOT Manager.<br>For Deployment in Development environment 4 GB is sufficient. |
| Disk | 120 GB ~ 512GB (Depending on the overall ELIOT Topology) |
| Networks | 2 to 4 Network Interfaces with public IP. |

| ELIOT Edge Node(s) | |
|---|---|
| HW Aspect | |
| #of Node(s) | 1 Eliot Edge Node |
| CPU | Minimum 1 socket x86_AMD64 or ARM64 with Virtualization support. |
| RAM | 2 GB ~ 16 GB it depends on the number of ELIOT Edge nodes which will be managed by ELIOT Manager.<br>For Deployment in Development environment 4 GB is sufficient |
| Disk | 20 GB ~ 256GB (Depending on the overall ELIOT Topology) |
| Networks | 2 Network Interfaces . |

## Software Prerequisites

- Virtual Machines preinstalled with Ubuntu 16.04
- root user created in the ELIOT Manager and ELIOT Edge Node
- SSH Server running in both the Nodes.
- sshpass installed in ELIOT Manager
- scp installed in ELIOT Manager and ELIOT Edge Node
- GO Lang installed in ELIOT Manager and ELIOT Edge Node (Note : This is required when we are setting up the ELIOT using KubeEdge.)
- git should be installed in ELIOT Manager and ELIOT Edge Node.

## Database Prerequisites

### Schema scripts

N/A

## Other Installation Requirements

### Jump Host Requirements

N/A

### Network Requirements

- Internet Connectivity in both ELIOT Manager and ELIOT Edge Node
- ELIOT Manager and ELIOT Node should be able to ping each other. Same network

### Bare Metal Node Requirements

N/A

### Execution Requirements (Bare Metal Only)

N/A

# Installation High-Level Overview

In ELIOT Release 1.0 for deployment 2 Ubuntu 16.04 Virtual Machines are used.

The steps includes to setup the following two environments :

- ELIOT Deployment using Kubernetes.
- ELIOT Deployment using KubeEdge

## Bare Metal Deployment Guide

### Install Bare Metal Jump Host

Currently ELIOT Deployment have been done in Virtual Machines only but the installation steps and scripts will hold good for execution in Bare Metal also if the 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 two Ubuntu VM's are required one for ELIOT Manager and ELIOT Edge Node both the node should have internet connectivity , network interface and network connectivity between the two VM's.

### Standard Deployment Overview

#### ELIOT Deployment using Kubernetes

Clone the ELIOT git repo in **ELIOT Manager** node. It will create eliot folder, for executing the script go to $HOME/eliot/scripts folder.

Edit the nodelist file with ELIOT Edge Node details.  Replace the below contents with the actual value with '|' separated.

The below details are required in the **setup.sh** for sshpass command. The details are required by sshpass to login to ELIOT Edge Node from ELIOT Manager and execute the deployment scripts in the ELIOT Edge Node.

<eliotedgenodeusername>|<eliotedgenodeip>|<eliotedgenodepassword>

- eliotedgenodeusername – ELIOT Edge Node User Name.
- eliotedgenodeip – ELIOT Edge Node IP
- eliotedgenodepassword - ELIOT Edge Node Password used in setup.sh script to login to ELIOT Edge node from ELIOT Manager to execute scripts in ELIOT Edge node.

The ELIOT Edge Node should be reachable from ELIOT Manager and ssh should work from ELIOT Manager to ELIOT Edge node as a prerequisite.

```
root@akraino-slave-001:#git clone "https://gerrit.akraino.org/r/eliot"
root@akraino-slave-001:#cd eliot/scripts/
root@akraino-slave-001:~/eliot/scripts#vim nodelist
```

After editing the nodelist file , execute the setup.sh script.  The setup.sh script internally installs Docker and Kubernetes with Calico CNI plugin and RBAC for authorization.

```
root@akraino-slave-001:~/eliot/scripts#./setup.sh
```

### ELIOT Deployment using KubeEdge

Clone the ELIOT git repo in ELIOT Manager node. It will create eliot folder, for executing the script go to $HOME/eliot/scripts/src folder.

Edit the **config_kubeedge** file with ELIOT Manager and ELIOT Edge Node

The format of config_kubeedge file is as follows :

edgenodeusr=""

edgenodeip=""

edgenodepassword=""

masternodeusr=""

masternodeip=""

masternodepassword=""

```
root@akraino-slave-001:#git clone "https://gerrit.akraino.org/r/eliot"
root@akraino-slave-001:#cd eliot/scripts/
root@akraino-slave-001:~/eliot/scripts#vim config_kubeedge
root@akraino-slave-001:~/eliot/scripts#./kubeedge_setup.sh
```

## Snapshot Deployment Overview

There are two images created for Manager and Node machines. These are .qcow2 files. After deploying the VMs using these images, your setup for ELIOT is ready in minutes. Following is the link to the document to use these images:

**Using Kubernetes:**

https://docs.google.com/document/d/1TwtDRvxJ_0IFzc5KvO7aaOh3If-jhpCSrccjttQA4LE/edit?usp=sharing

**Using KubeEdge:**

https://docs.google.com/document/d/1kmSJtX3jZRLJEcHlBSSiwNGDdQ9XVB3xIfEUaeHnWGo/edit?usp=sharing

## 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 Kubernetes Deployment

The setup.sh script will install the ELIOT Manager and one ELIOT Edge Node. After script execution is completed the kubernetes cluster setup can be checked by executing kubectl get nodes. It will show both the ELIOT Manager and the ELIOT Edge Node as ready.

The setup script also deploys NGINX server on the ELIOT Edge Node and verifies whether its deployed successful or not.

# Developer  Guide and Troubleshooting

# Uninstall Guide

### For Installation via Kubernetes:

Following two files can be used to revert the changes made by ELIOT deployment through kubernetes.

1. **kubernetes_reset.sh:**
   **Purpose**: There were certain issues occurring when the setup.sh file was executed again on the same setup. it threw error saying 'the file/port in already in use' when setup.sh file was executed again.
   **Solution:** A list was prepared for errors occurring on re-execution of setup.sh file. To fix them, some commands were executed manually to reset the changes. Once the list was finalized, these were consolidated in a .sh file. The file is in eliot/scripts/ folder.
   **What it does:**
   -- Resets kubeadm changes on both Manager and Edge Node machine.
   -- Installs iptables, ipvsadm required to restart kubectl service on both Manager and Edge Node machine.
   -- Restarts kubelet service on both Manager and Edge Node machine.
   -- Releases the port 10250 on both Manager and Edge Node machine.
   -- Deletes kubernetes file on Edge Node machine.

2. **kubernetes_cleanup.sh**
   **Purpose**: There was no file to rollback the changes done by setup.sh.
   **Solution:** A list of changes done by setup.sh file was prepared and then commands were written to rollback those installations and changes. The file is in eliot/scripts/ folder.
   **What it does:**
   -- Uninstalls Docker, Kubernetes, Kubectl, Kubeadm, Kubelet on both Manager and Edge Node machine.
   -- Resets kubeadm changes on both Manager and Edge Node machine.
   -- Installs iptables, ipvsadm required to restart kubectl service on both Manager and Edge Node machine.
   -- Restarts kubelet service on both Manager and Edge Node machine.
   -- Releases the port 10250 on both Manager and Edge Node machine.

### For Installation via KubeEdge:

Following two files can be used to revert the changes made by ELIOT deployment through kubernetes.

1. **kubeedge_reset.sh:**
   **Purpose**: There were certain issues occurring when the setup.sh file was executed again on the same setup. it threw error saying 'the file/port in already in use' when setup.sh file was executed again.
   **Solution:** A list was prepared for errors occurring on re-execution of setup.sh file. To fix them, some commands were executed manually to reset the changes. Once the list was finalized, these were consolidated in a .sh file. The file is in eliot/scripts/ folder.
   **What it does:**
   -- Resets kubeadm changes on both Manager and Edge Node machine.
   -- Installs iptables, ipvsadm required to restart kubectl service on both Manager and Edge Node machine.
   -- Restarts kubelet service on both Manager and Edge Node machine.
   -- Releases the port 10250 on both Manager and Edge Node machine.
   --

2. **kubeedge_cleanup.sh**
   **Purpose**: There was no file to rollback the changes done by setup.sh.
   **Solution:** A list of changes done by setup.sh file was prepared and then commands were written to rollback those installations and changes. The

file is in eliot/scripts/ folder.
**What it does:**
-- Uninstalls Docker on both Manager and Edge Node machine.
-- Resets kubeadm changes on both Manager and Edge Node machine.
-- Installs iptables, ipvsadm required to restart kubectl service on both Manager and Edge Node machine.
-- Restarts kubelet service on both Manager and Edge Node machine.
-- Releases the port 10250 on both Manager and Edge Node machine.

# Troubleshooting

N/A

## 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

# References

# Definitions, acronyms and abbreviations