

# IEC Type1&2 Installation Guide for R2

- [Introduction](#)
- [How to use this document](#)
- [Deployment Architecture](#)
- [Pre-Installation Requirements](#)
  - [Hardware Requirements](#)
    - [Minimum Hardware Requirements](#)
    - [Recommended Hardware Requirements](#)
  - [Software Prerequisites](#)
  - [Database Prerequisites](#)
    - [Schema scripts](#)
  - [Other Installation Requirements](#)
    - [Jump Host Requirements](#)
    - [Network Requirements](#)
    - [Bare Metal Node Requirements](#)
    - [Execution Requirements \(Bare Metal Only\)](#)
- [Installation High-Level Overview](#)
  - [Bare Metal Deployment Guide](#)
    - [Install Bare Metal Jump Host](#)
    - [Creating a Node Inventory File](#)
    - [Creating the Settings Files](#)
    - [Running](#)
  - [Virtual Deployment Guide](#)
    - [Standard Deployment Overview](#)
    - [Snapshot Deployment Overview](#)
    - [Special Requirements for Virtual Deployments](#)
    - [Install Jump Host](#)
    - [Verifying the Setup - VMs](#)
    - [Upstream Deployment Guide](#)
    - [Upstream Deployment Key Features](#)
    - [Special Requirements for Upstream Deployments](#)
    - [Scenarios and Deploy Settings for Upstream Deployments](#)
    - [Including Upstream Patches with Deployment](#)
    - [Running](#)
    - [Interacting with Containerized Overcloud](#)
- [Verifying the Setup](#)
- [OpenStack Verification](#)
- [Developer Guide and Troubleshooting](#)
  - [Utilization of Images](#)
  - [Post-deployment Configuration](#)
  - [OpenDaylight Integration](#)
  - [Debugging Failures](#)
  - [Reporting a Bug](#)
- [Uninstall Guide](#)
- [Troubleshooting](#)
  - [Error Message Guide](#)
- [Maintenance](#)
- [Frequently Asked Questions](#)
- [License](#)
- [References](#)

## Introduction

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

This document provides guidelines on how to manually install the Akraino IEC Release 2, including required software and hardware configurations. The steps described below are automatized in CI using Fuel@OPNFV or Compass. For details on this procedure, check the [IEC Type1&2 Test Document for R2](#)

The audience of this document is assumed to have good knowledge of networking and Unix/Linux administration.

Currently, the chosen operating system (OS) is Ubuntu 16.04 and/or 18.04.

The infrastructure orchestration of IEC is based on Kubernetes, which is a production-grade container orchestration with a rich running eco-system. There are some options for Container Network Interface (CNI) solutions for IEC, e.g Calico, Contiv-vpp and Flannel. But the default container network interface (CNI) solution chosen for Kubernetes is Calico, which is a high performance, scalable, policy enabled and widely used container networking solution with rather easy installation and arm64 support.

Currently MACCHIATObin board is used as a typical Type1 hardware platform, and we provided a guide on [how to setup hardware](#). There is no explicit difference between Type1 and Type2 on the installation method of IEC Release 2.

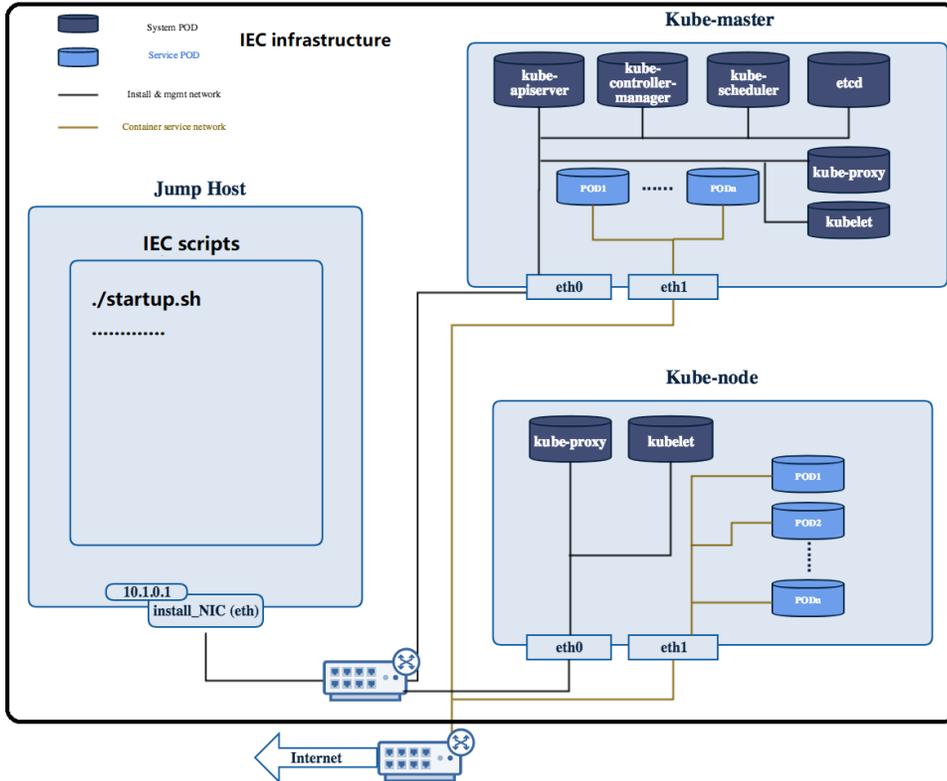
The installation guide is mostly inherited from [that of R1](#).

# How to use this document

The following sections describe the prerequisites for planning an IEC deployment. Once these are met, installation steps provided should be followed in order to obtain an IEC compliant Kubernetes cluster.

## Deployment Architecture

The reference cluster platform consists of 3 nodes, baremetal or virtual machines:



- the first node will have the role of Kubernetes Master;
- all other nodes will have the role of Kubernetes Slave;
- Calico/Flannel/Contiv will be used as container network interface (CNI);

One additional management/orchestration node (which will be referred to as **jumpserver** or **orchestration node**) is necessary for running the installation steps.

If all nodes are virtual machines on the same machine which is also used as the **jumpserver**, the deployment type will be referred to as **virtual** - useful mostly for development and/or testing and not production grade.

**i** The default number of Kubernetes slaves is 2; although less or more slaves can be used as well.

**!** Currently, we assume all the cluster nodes have the same architecture (**aarch64** or **x86\_64**).

All machines (including the **jumpserver**) should be part of at least one common network segment.

## Pre-Installation Requirements

### Hardware Requirements



Hardware requirements depend on the deployment type. If more cluster nodes are used, the requirements for a single node can be lowered, provided that the sum of available resources is enough.

Depending on the intended usecase(s), more memory/storage might be required for running/storing the containers.

## Minimum Hardware Requirements

| HW Aspect    | Requirement   |
|--------------|---|
| 1 Jumpserver | A physical or virtualized machine that has direct network connectivity to the cluster nodes.<br><div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">  For <b>virtual</b> deployments, CPU/RAM/disk requirements of cluster nodes should be satisfiable as virtual machine resources when using the <b>jumpserver</b> as a hypervisor. </div> |
| CPU          | Minimum 1 socket (each cluster node)  |
| RAM          | Minimum 2GB/server (Depending on usecase work load)   |
| Disk         | Minimum 20GB (each cluster node)  |
| Networks     | Minimum 1   |

## Recommended Hardware Requirements

| HW Aspect    | Requirement   |
|--------------|---|
| 1 Jumpserver | A physical or virtualized machine that has direct network connectivity to the cluster nodes.<br><div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">  For <b>virtual</b> deployments, CPU/RAM/disk requirements of cluster nodes should be satisfiable as virtual machine resources when using the <b>jumpserver</b> as a hypervisor. </div> |
| CPU          | 1 socket (each cluster node)  |
| RAM          | 16GB/server (Depending on usecase work load)  |
| Disk         | 100GB (each cluster node)   |
| Networks     | 2/3 (management and public, optionally separate PXE)  |

## Software Prerequisites

- Ubuntu 16.04/18.04 is installed on each node;
- SSH server running on each node, allowing password-based logins;
- a user (by default named IEC, but can be customized via config later) is present on each node;
- **IEC** user has passwordless sudo rights;
- **IEC** user is allowed password-based SSH login;

## Database Prerequisites

### Schema scripts

N/A

## Other Installation Requirements

### Jump Host Requirements

N/A

## Network Requirements

- at least one common network segment across all nodes;
- internet connectivity;

## Bare Metal Node Requirements

N/A

## Execution Requirements (Bare Metal Only)

N/A

# Installation High-Level Overview

## Bare Metal Deployment Guide

### Install Bare Metal Jump Host

The jump host (**jumpserver**) operating system should be pre-provisioned. No special software requirements apply apart from package prerequisites:

- git
- sshpass

### Creating a Node Inventory File

N/A

### Creating the Settings Files

Clone the IEC git repo and edit the configuration file by setting:

- user name for SSH-ing into cluster nodes (default: **iec**);
- user password for SSH-ing into cluster nodes;
- Kubernetes master node IP address (should be reachable from **jumpserver** and accept SSH connections);
- Kubernetes slave node(s) IP address(es) and passwords for SSH access;

```
jenkins@jumpserver:~$ git clone https://gerrit.akraino.org/r/iec.git
jenkins@jumpserver:~$ cd iec/src/foundation/scripts
jenkins@jumpserver:~/iec/src/foundation/scripts$ vim config
```

## Running

Simply start the installation script with default parameters in the same directory:

```
jenkins@jumpserver:~/iec/src/foundation/scripts$ ./startup.sh
```

If you want to deploy K8s with other options, please refer to following commands:

```
jenkins@jumpserver:~/iec/src/foundation/scripts$ ./startup.sh -C flannel -k 1.15.2 -c 0.7.5 #Deploy 1.15.2 K8s
with Flannel CNI
jenkins@jumpserver:~/iec/src/foundation/scripts$ ./startup.sh -C contivpp -k 1.15.2 -c 0.7.5 # Deploy 1.15.2
K8s with Contiv-vpp CNI
```

There are some different options for startup.sh scripts. Please refer to following information:

```
-k|--kube: The version of k8s
-c|--cni-ver: ---- Kubernetes-cni version
-C|--cni: ---- CNI type: calico/flannel/contivpp
```



If you want to deploy the K8s with Contiv-vpp, you must specify 1 NIC which will be used in Contiv-vpp. Then modify the configuration file.

## Virtual Deployment Guide

### Standard Deployment Overview

From the installer script's perspective, virtual deployments are identical to baremetal ones. Preprovision some virtual machines on the **jumpserver** node as hypervisor, using Ubuntu 16.04/18.04, then continue the installation similar to the baremetal deployment process described above.

### Snapshot Deployment Overview

N/A

### Special Requirements for Virtual Deployments

N/A

### Install Jump Host

Similar to baremetal deployments. Additionally, one hypervisor solution should be available for creating the cluster nodes virtual machines (e.g. KVM).

### Verifying the Setup - VMs

N/A

### Upstream Deployment Guide

N/A

### 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

Similar to virtual deployments, edit the configuration file, then launch the installation script:

```
jenkins@jumpserver:~$ git clone https://gerrit.akraino.org/r/iec.git
jenkins@jumpserver:~$ cd iec/src/foundation/scripts
jenkins@jumpserver:~/iec/src/foundation/scripts$ vim config
jenkins@jumpserver:~/iec/src/foundation/scripts$ ./startup.sh
```

### Interacting with Containerized Overcloud

N/A

## Verifying the Setup

IEC installation automatically performs one simple test of the Kubernetes cluster installation by spawning an **nginx** container and fetching a sample file via HTTP.

## OpenStack Verification

N/A

## Developer Guide and Troubleshooting

### Utilization of Images

N/A

### Post-deployment Configuration

N/A

### OpenDaylight Integration

N/A

### Debugging Failures

N/A

### Reporting a Bug

All issues should be reported via [IEC JIRA](#) page. When submitting reports, please provide as much relevant information as possible, e.g.:

- output logs;
- IEC git repository commit used;
- jumpserver info (operating system, versions of involved software components et al.);
- command history (when relevant);

## Uninstall Guide

N/A

## Troubleshooting

### Error Message Guide

N/A

## Maintenance

N/A

## Frequently Asked Questions

N/A

## License

Any software developed by the "Akraino IEC" Project is licenced under the Apache License, Version 2.0 (the "License"); you may not use the content of this software bundle except in compliance with the License. You may obtain a copy of the License at <<https://www.apache.org/licenses/LICENSE-2.0>>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## References

For more information on the Akraino Release 1, please see:

- Apache License 2.0: <https://www.apache.org/licenses/LICENSE-2.0>
- Akraino Home Page: <https://wiki.akraino.org/pages/viewpage.action?pageId=327703>
- IEC Wiki: <https://wiki.akraino.org/display/AK/Integrated+Edge+Cloud+%28IEC%29+Blueprint+Family>
- IEC JIRA: <https://jira.akraino.org/projects/IEC/issues/>
- Akraino Blueprint Validation: <https://wiki.akraino.org/display/AK/Akraino+Blueprint+Validation+Framework>
- Sonobuoy: <https://github.com/heptio/sonobuoy>