

# IEC Internal Verification and Validation Lab Setup

## Design Overview

We want to be as flexible as possible when it comes to different lab layouts, architecture and/or location.

To achieve that, *all IEC components will **eventually** run in VMs* (virtual machines), either collocated on the same hypervisor or distributed across a pool of hypervisors in the same region.

This will allow us to treat the following scenarios in an abstract and uniform manner:

- no hypervisor, all IEC components distributed on 3 baremetal nodes (this scenario is not to be used in production, as we want to treat the baremetal nodes more or less like firmware, i.e. a black box we don't want to touch);
- single hypervisor node holding 3 VMs (one K8s master and two slaves);
- multiple hypervisor nodes holding one or more VMs each (e.g. 1 hypervisor with 1 VM for K8s master, 2 hypervisors with 1 VM each for K8s slaves);

## Development Lab Setup

For development purposes, where performance is not crucial, the setup with the smallest hardware footprint is recommended.

To mimic a real production setup, 3 nodes are still desired, in which case a single hypervisor running 3 virtual machines makes the most sense.

Hardware/software requirements:

- 1 x aarch64/x86\_64 baremetal machine (e.g. [Marvell ThunderX2](#));
- A preinstalled operating system (e.g. Ubuntu 16.04/18.04, CentOS7) with KVM support enabled;
- Internet connectivity (or all the artifacts referenced throughout this wiki available offline);

Before you get started with the instructions on the [IEC Blueprints Installation Overview](#):

- create 3 virtual machines on the hypervisor node (networking layout is not very strict, all it matters is that all VMs can access the internet);
- provision Ubuntu 16.04/18.04 in each of them;

These steps can be performed manually or automated using one of the available tools (e.g. OPNFV Fuel installer, OPNFV Compass4NFV installer, MaaS etc.).

## CI Lab Setup

For CI, 3 baremetal nodes, the TOR switch and one additional (installer) node will be grouped together as described below:

- an installer node (also referred to as "the jumpserver" in OPNFV) will be running the Jenkins slave process and orchestrate all the CI operations involving the 3 baremetal nodes;  
This node can be a baremetal node or a virtual machine (or even a container, although persistence of the Jenkins slave configuration should be properly considered in that case).  
Preferably, the architecture of this node and the architecture of the 3 baremetal nodes should be the same (this is a hard requirement if using OPNFV Fuel as an installer).  
The installer node should be pre-provisioned with a supported operating system (e.g. Ubuntu 16.04/18.04, CentOS7).  
For community labs, access to this installer (jumpserver) node should be made available on request, usually via SSH over lab owner controlled VPN access.  
The installer node should have access to the other 3 baremetal nodes BMCs (e.g. to power them on/off via IPMI, set boot options et. al.).  
The installer node should have internet access.  
The installer node should have access to all relevant POD networks as well (e.g. PXE/admin, management networks).
- TOR switch configuration should account for all blueprint requirements of the projects that will be tested in that lab;  
For example, at least the following isolated networks are recommended:
  - PXE/admin for operating system bootstrapping on the baremetal nodes;
  - public network for internet access;
  - management network;  
Apart from PXE/admin, which is recommended to be always untagged, the rest of the networks can be a mix of tagged and/or untagged VLANs.
- 3 baremetal nodes - all should have the same architecture;  
All nodes should be able to PXE boot from the installer node (or an equivalent operating system bootstrapping mechanism should be supported).