

KNI PAE Installation Guide

- [KNI PAE specific workloads](#)
 - [All-platforms workloads](#)
 - [Baremetal workloads](#)

This blueprint is part of the Kubernetes-Native Infrastructure for Edge family. All blueprints in this family follow the same installation guide, so please see the KNI family's [User Documentation](#).

KNI PAE specific workloads

The KNI blueprints share same installation procedure, until the cluster has been successfully deployed. However, they have different workloads applied on it (knictl apply_workloads step). Following is a detail of the applied workloads and how to adapt for your site.

The workloads applied to KNI PAE are following the base/profiles/site pattern. Please see [KNI PAE Architecture document](#) for reference.

All-platforms workloads

- **Kubevirt (<https://kubevirt.io>):**
It allows to spin up VMs using Kubernetes platform.
To customize, you could patch the manifests from https://github.com/akraino-edge-stack/kni-blueprint-pae/tree/master/base/02_cluster-addons/00_kubevirt
- **CNI macvlan (<https://intel.github.io/multus-cni/doc/how-to-use.html>):**
It adds a NetworkAttachmentDefinition configuration for Multus, creating a macvlan. It will allow pods to use the network interface created there.
To customize, you could patch the manifests from https://github.com/akraino-edge-stack/kni-blueprint-pae/tree/master/base/02_cluster-addons/01_cni-macvlan
You may need to update the configuration, replacing the full *config* string
- **CNI ipvlan:**
Similar as previous one, but with ipvlan configuration. You can patch manifests on https://github.com/akraino-edge-stack/kni-blueprint-pae/tree/master/base/02_cluster-addons/02_cni-ipvlan
- **Node feature discovery (<https://github.com/kubernetes-sigs/node-feature-discovery>):**
It adds NodeFeatureDiscovery component to the Kubernetes cluster. It performs a set of checks in nodes, and adds annotations with the information it can find. It will report hardware, software, network facts, etc...
To customize, you could patch the manifests from https://github.com/akraino-edge-stack/kni-blueprint-pae/tree/master/base/02_cluster-addons/03_nfd

Baremetal workloads

These workloads will only be applied when the site is using the baremetal profile:

- **Performance Profile**

The PerformanceProfile CRD is the API of the openshift-performance-addon operator (<https://github.com/openshift-kni/performance-addon-operators>) that applies various performance tunings to cluster nodes to achieve lower latency.

The first step would be to install the operator. The operator manifest has the following bits -

Target Namespace - Namespace in which the operator will be installed - https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/05_performance-operator/01_namespace.yaml

Operator Group - Create an OperatorGroup CR in the target namespace - https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/05_performance-operator/02_perf-operatorgroup.yaml

Subscription - Create a subscription CR to subscribe the target namespace to the operator by tracking a channel, like so - https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/05_performance-operator/03_perf-sub.yaml

The next step would be to create and apply the [PerformanceProfile CRD](#).

An example can be found here - https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/05_performance-operator/04_perfprofile-conf.yaml.

This will automatically update the kernel by setting the kernel arguments as given in the yaml file with other parameters like the enablement of real time kernel, setting huge pages to 1G, reserving CPUs that will not be affected by any container workloads.

- **sriov-network-operator:**

It adds the SRIOV network operator, that will add support for managing SRIOV interfaces inside Kubernetes cluster (<https://github.com/openshift/sriov-network-operator>). The following manifest can be patched at site level to reflect the settings needed for the environment: https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/01_sriov-network-operator/03_sriovnetwork_v1_sriovnetworknodepolicy_crd.yaml

- **ptp-daemonset:**

It adds components to enable PTP (precision time protocol). It has the following components:

- https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/02_ptp-daemonset/01_ptp-machineconfig.yaml : enables PTP kernel module on nodes labelled as worker-ran
- https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/02_ptp-daemonset/05_configmap.yaml: configmap used to configure PTP. The configmap has two settings (PTP4.OPTIONS, PHC2CSYS.OPTIONS), that need to be configured properly per site. This manifest should be patched at site level, to change the NIC and the desired parameters.
- **storage:**
Adds Ceph storage to the Kubernetes cluster. It deploys and configures the Rook Ceph operator (<https://github.com/rook/rook/blob/master/Documentation/ceph-quickstart.md>), relying on directories on nodes to setup the storage space. Following manifests can be patched:
 - https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/03_storage/02_ceph_cluster.yaml: adds specific settings for the ceph cluster
 - https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/03_storage/03_ceph_storage_class.yaml: it defines an CephBlockPool storage class to be used on pods.
 - https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/03_storage/04_ceph_storage_filesystem.yaml: it defines a CephFilesystem storage class to be used on pods.
 - https://github.com/akraino-edge-stack/kni-blueprint-pae/blob/master/profiles/production.baremetal/02_cluster-addons/03_storage/05_ceph_image_registry_pvc.yaml: using the defined rook-filestorage class, creates a PersistentVolumeClaim, to be used as storage for Image Registry.
- **nodes:**
It adds custom labels to specific worker nodes. Allows to give worker-rt, worker-ran and cpumanager-enabled labels at node level. It needs to be patched per site, as the node names will change.