Deploy Intel® Smart Edge Open Developer Experience Kit in a Multi-node Cluster without Edge Software Provisioner (ESP)

Configuration Guide

December 2022

Document Number: 764706-1.0

Contents

December 2022
Document Number: 764706-1.0

Revision History

Date	Revision	Description
December 2022	1.0	Initial release.

1.0 Overview

Intel® Smart Edge Open Developer Experience Kit enables you to install and instantiate an Intel® Smart Edge Open edge cluster.

The official way to deploy the Intel® Smart Edge Open Developer Experience Kit is based on Edge Software Provisioner (ESP). ESP automates provisioning the operating system and software stack as a whole. But in some cases, you may want to deploy the Intel® Smart Edge Open Developer Experience Kit on an existing OS. Follows are the instructions to deploy a multi-node cluster without the ESP.

December 2022
Document Number: 764706-1.0

2.0 Requirements

For simplicity, here we use one control-plane and one worker node, you can add more worker nodes accordingly.

Two physical machines or VMs are required, minimum recommendations are outlined here:

- CPU: 4
- MEM: 8 GB
- SSD: 50 GB (HDD is acceptable for worker nodes)
- OS: Ubuntu* 20.04 LTS
- Can access via SSH with the same username

3.0 Prerequisites

Setup NTP and proxy for all nodes **if necessary**.

Then, complete the preparation for **all nodes**:

```
# set variable for later use
export host=$(hostname)

# install and enable sshd
sudo apt update
sudo apt install openssh-server

# SUDO without password
echo "$USER ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/$USER

# add hostname into `/etc/hosts`
sudo sed -i "s/localhost.*\$/& $host/" /etc/hosts
```

Unless otherwise specified, the following operations are performed on the **control-plane node**.

Set up SSH access to all nodes with authorized key for the control-plane node.

- Assume the hostname of the control-plane node is `$host`.
- Assume the hostname of the worker node is `$worker`.

```
# set variable for later use
export worker="TBD" # replace with actual value

# DO leave passphrase as empty
ssh-keygen

# to self
ssh-copy-id $USER@$host
# to worker
ssh-copy-id $USER@$worker
```

Set up Python environment for Ansible:

```
# install python packages
sudo apt install python3-pip pipenv
```

4.0 Downloading

Clone the Developer Experience Kit (DEK) repository:

```
git clone https://github.com/smart-edge-open/open-developer-experience-kits.git --branch=smart-edge-open-22.03.03 dek
cd dek

# prepare pipenv
pipenv install
```

4.1 Hotfix for PRC Only

Apply PR #21 for the restricted network environment in China:

CAUTION: The hotfix depends on the available open mirror, proxy and CDN, we cannot guarantee that they will work 100%, use at your own risk.

```
curl -sL https://github.com/IntelSmartEdge/open-developer-experience-kits/pull/21.patch | git apply
```

5.0 Deployment

Create a inventory.yml for the intended deployment:

```
# double check the required variables
# should not be empty
echo $host
echo $worker

cat <<EOF > inventory.yml
---
all:
  vars:
    cluster_name: demo
    deployment: dek
    single_node_deployment: false
    limit:
controller_group:
  hosts:
    controller:
      ansible_host: $host
      ansible_user: $USER
edgenode_group:
  hosts:
    node01:
      ansible_host: $worker
      ansible_user: $USER
EOF

# check the result
git diff
```

Customize features as needed:

```
cat <<EOF >> inventory/default/group_vars/all/10-default.yml

# Customized vars
iommu_enabled: false
sriov_network_operator_enable: false
sriov_network_operator_configure_enable: false
e810_driver_enable: false
platform_attestation_node: false
sgx_enabled: false
install_hwe_kernel_enable: false
docker_registry_mirrors:
  - https://hub-mirror.c.163.com
proxy_env:
  http_proxy: "$http_proxy"
  https_proxy: "$https_proxy"
```

```
  ftp_proxy: "$ftp_proxy"
  no_proxy: "$no_proxy"
  all_proxy: "$all_proxy"
EOF

# check the result
git diff
```

Deploy and wait for about half an hour to finish:

```
pipenv run ./deploy.py
```

The host will reboot during deployment. **Run the same command again to continue**.

Find all the logs under logs/

6.0 Verify Result

Access Grafana with URL https://$host:32000, login with user admin. You can get the default password with command:

```
kubectl get secret -n telemetry grafana -o jsonpath='{.data.admin-passw
ord}' | base64 -d && echo
```

Access Harbor with URL https://$host:30003, login with user admin. You can get the default password with command:

```
kubectl get secret -n harbor harbor-admin -o jsonpath='{.data.admin-pas
sword}' | base64 -d && echo
```

7.0 Known Issues

7.1 Failed to Install pip3

```
task path: /home/box/seo/dek/roles/infrastructure/docker/tasks/install_
pip_dep.yml:16
...
  File "/home/box/.local/lib/python3.8/site-packages/pip/_internal/loca
tions/__init__.py", line 244, in get_scheme
    from . import _distutils
ImportError: cannot import name '_distutils' from 'pip._internal.locati
ons' (/home/box/.local/lib/python3.8/site-packages/pip/_internal/locati
ons/__init__.py)
```

There is something wrong in the pip3 v22.2, which is installed by the latest pipenv.

See also #11103

The solution is to use the Ubuntu maintained one instead:

```
rm -rf ~/.local
sudo apt install pipenv
pipenv install
```

7.2 The Role for PTP Was Not Found

```
ERROR! the role 'infrastructure/ptp/controlplane' was not found in /hom
e/box/dek/playbooks/roles:/home/box/dek/roles:/home/box/dek/roles/basel
ine_ansible:/home/box/dek/playbooks

The error appears to be in '/home/box/dek/playbooks/infrastructure.yml'
: line 59, column 7, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

  roles:
    - role: infrastructure/ptp/controlplane
      ^ here
```

The roles in roles/infrastructure/ptp/ no longer exists, but are still referenced by the playbook, create empty folders as workaround.

```
mkdir -p infrastructure/ptp/{controlplane,node}
```

8.0 See Also

- Intel® Smart Edge Open
- Intel® Smart Edge Open Developer Experience Kit at GitHub

December 2022
Document Number: 764706-1.0