



Deploy Intel® Smart Edge Open Developer Experience Kit without Edge Software Provisioner (ESP)

Configuration Guide

April 2023



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit www.intel.com/design/literature.htm.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No product or component can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

Contents

1.0	Overview	5
2.0	Requirements.....	6
3.0	Prerequisites	7
4.0	Downloading.....	8
4.1	Hotfix for PRC Only.....	8
5.0	Deployment.....	9
6.0	Verify Result.....	11
7.0	Known Issues	12
7.1	Failed to Install pip3.....	12
7.2	Failed to Initialize Cluster.....	12
8.0	See Also.....	14

Revision History

Date	Revision	Description
April 2023	1.2	Added PyPI mirror suggestion. Added new Known Issue: pipenv install timeout.
December 2022	1.1	Updated code throughout the document. Added Hotfix for PRC. Added new Known Issues.
August 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 single node cluster without using the ESP.

2.0 *Requirements*

One physical machine or VM is required, minimum recommendations are outlined here:

- CPU: 4
- MEM: 8 GB
- SSD: 50 GB
- OS: Ubuntu* 20.04 LTS

3.0 Prerequisites

Setup NTP and proxy **if necessary**:

Then, complete the preparation for the target machine:

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

# install necessary packages
sudo apt update
sudo apt install openssh-server git

# setup SSH access with authorized key
# DO leave passphrase as empty
ssh-keygen
ssh-copy-id $USER@$host

# 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
```

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
```

Intel suggests using a stable PyPI mirror in China, e.g.:

```
pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
pip config set global.trusted-host pypi.tuna.tsinghua.edu.cn
```


5.0 Deployment

Create a inventory.yml for the intended deployment:

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

cat <<EOF > inventory.yml
---
all:
  vars:
    cluster_name: demo
    deployment: dek
    single_node_deployment: true
    limit:
controller_group:
  hosts:
    controller:
      ansible_host: $host
      ansible_user: $USER
edgenode_group:
  hosts:
    node01:
      ansible_host: $host
      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](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-password}' | base64 -d && echo
```

Access [Harbor](#) with URL [https://\\$host:30003](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-password}' | base64 -d && echo
```

7.0 Known Issues

7.1 Failed to Install pip3

```
task path: /home/box/se0/dek/roles/infrastructure/docker/tasks/install_
pip_dep.yml:16
...
File "/home/box/.local/lib/python3.8/site-packages/pip/_internal/locati
ons/__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 Pipenv Install Timeout

```
An error occurred while installing ansible==2.9.27! Will try again.
An error occurred while installing cffi==1.15.0! Will try again.
An error occurred while installing cryptography==3.3.2! Will try again.

ERROR: Exception:
...
File "/usr/lib/python3.8/ssl.py", line 1099, in read
    return self._sslobj.read(len, buffer)
socket.timeout: The read operation timed out
```

Maybe <https://pypi.python.org/simple> is too busy, use a mirror instead:

```
pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simpl
e
pip config set global.trusted-host pypi.tuna.tsinghua.edu.cn

pipenv run pip install ansible==2.9.27 cffi==1.15.0 cryptography==3.3.2
```

7.3 Failed to Initialize Cluster

error execution phase wait-control-plane: couldn't initialize a Kubernetes cluster

Check kubelet status, it works fine, but critical Pods, such as kube-apiserver, are not running:

```
sudo systemctl status kubelet
sudo docker ps -a
```

Check the logs of kubelet:

```
journalctl -r -u kubelet | grep apiserver
... "Failed to admit pod, unexpected error while attempting to recover
from admission failure" pod="kube-system/kube-apiserver-joez-hce-ub20-v
m-seo-test" err="preemption: error finding a set of pods to preempt: no
set of running pods found to reclaim resources: [(res: cpu, q: 250), ]"
```

Which means there is not enough CPU resources for new Pods.

```
sudo grep reserved /tmp/config.yaml
reservedSystemCPUs: 0,1

nproc
2
```

We reserved 2 CPUs for system daemons, no CPUs left for the Pods, more CPUs are required.

8.0 *See Also*

- [Intel® Smart Edge Open](#)
- [Intel® Smart Edge Open Developer Experience Kit at GitHub](#)