# Build O-DU docker image

Please follow the build procedures on baremetal host to build FAPI/WLS/FH and copy the code/binaries to the oran\_release folder. The docker build will just copy the pre-built binaries to docker image.

oran\_release/phy

oran\_release/FlexRAN-FEC-SDK-21.03

oran\_release/FlexRAN

Set the network proxy and ORAN\_RELEASE environment variable before build

export http\_proxy=<your\_proxy>

export https\_proxy=<your\_proxy>

export no\_proxy="localhost,127.0.0.1"

export ORAN\_RELEASE=<TOP\_FOLDER>/oran\_release

Update ${ ORAN\_RELEASE}/phy/setupenv.sh, change DIR\_ROOT to /opt/oran\_release.

Run below script to trigger the docker build:

sh ${ORAN\_RELEASE}/FlexRAN/docker/build-oran-l1-image.sh

After the docker build finishes, below docker image will be generated

…

Successfully tagged oran-release-du:2.0

# Run O-DU docker container in Kubernetes cluster

You can push the O-DU docker image to your local docker registry where Kubernetes cluster can pull the image. Or you can copy the image locally to your Kubernetes worker node.

In your worker node, make sure you have configured the system/tools correctly following previous sections for running L1 in baremetal host.

Assume intel system studio is installed in /opt/intel/system\_studio\_2019/

Make sure you have enough 1G hugepages

Update the Kubernetes pod yaml configuration for your system configuration

cat ${ORAN\_RELEASE}/FlexRAN/docker/oran-release-du.yaml

apiVersion: v1

kind: Pod

metadata:

labels:

app: oran-release-du-pod

name: oran-release-du-pod

spec:

containers:

- securityContext:

privileged: false

capabilities:

add:

- SYS\_ADMIN

- IPC\_LOCK

- SYS\_NICE

command:

- sleep

- infinity

tty: true

stdin: true

image: oran-release-du:2.0

name: l1app

resources:

requests:

memory: "16Gi"

hugepages-1Gi: 8Gi

limits:

memory: "16Gi"

hugepages-1Gi: 8Gi

volumeMounts:

- name: hugepage

mountPath: /hugepages

- name: varrun

mountPath: /var/run/dpdk

readOnly: false

- name: iss-path

mountPath: /opt/intel/system\_studio\_2019/

- securityContext:

privileged: false

capabilities:

add:

- SYS\_ADMIN

- IPC\_LOCK

- SYS\_NICE

command:

- sleep

- infinity

tty: true

stdin: true

image: oran-release-du:2.0

name: oran-fapi

resources:

requests:

hugepages-1Gi: 4Gi

memory: "4Gi"

limits:

hugepages-1Gi: 4Gi

memory: "4Gi"

volumeMounts:

- name: hugepage

mountPath: /hugepages

- name: iss-path

mountPath: /opt/intel/system\_studio\_2019/

- name: varrun

mountPath: /var/run/dpdk

readOnly: false

- securityContext:

privileged: false

capabilities:

add:

- SYS\_ADMIN

- IPC\_LOCK

- SYS\_NICE

command:

- sleep

- infinity

tty: true

stdin: true

image: oran-release-du:2.0

name: testmac

resources:

requests:

memory: "4Gi"

limits:

memory: "4Gi"

volumeMounts:

- name: hugepage

mountPath: /hugepages

- name: iss-path

mountPath: /opt/intel/system\_studio\_2019/

- name: varrun

mountPath: /var/run/dpdk

readOnly: false

volumes:

- name: hugepage

emptyDir:

medium: HugePages

- name: varrun

emptyDir: {}

- name: iss-path

hostPath:

path: /opt/intel/system\_studio\_2019/

Create the pod using kubectl

kubectl create –f oran-release-du.yaml

After the pod/containers are running, login different containers (l1app, fapi, testmac) and start the applications.

1. Open first terminal and execute l1app container:

kubectl exec oran-release-du-pod -c l1app -it bash

Setup environment variables by

source /opt/oran\_release/phy/setupenv.sh

Start l1app by

cd /opt/oran\_release/FlexRAN/l1/bin/nr5g/gnb/l1/

./l1.sh -e

1. After l1app is started, open second terminal and execute oran-fapi container:

kubectl exec oran-release-du-pod -c oran-fapi -it bash

Setup environment variables by

source /opt/oran\_release/phy/setupenv.sh

Start oran\_5g\_fapi by

cd /opt/oran\_release/phy/fapi\_5g/bin

./oran\_5g\_fapi.sh --cfg oran\_5g\_fapi.cfg

1. Open third terminal and execute testmac container:

kubectl exec oran-release-du-pod -c testmac -it bash

Setup environment variables by

source /opt/oran\_release/phy/setupenv.sh

Start testmac by

cd /opt/oran\_release/FlexRAN/l1/bin/nr5g/gnb/testmac

./l2.sh --testfile oran\_bronze\_rel\_fec\_sw.cfg

After testmac container is started and running, you can see the test is running. You can check the terminal of each container to verify the test result.