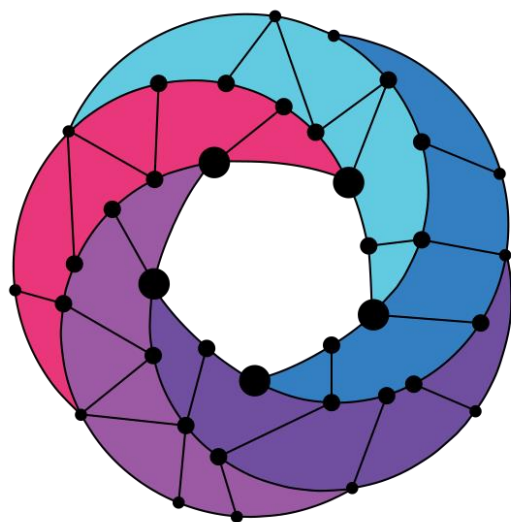




Open Source
MANO

Development Guidelines for OSM Modules

K Sai Kiran
Zero Touch Specialist – TATA ELXSI



Goals of the session

- General workflow in OSM development.
- Efficient development environment setup demo for
 - NBI (Pycharm)
 - RO (Pycharm)
 - NG-UI (VsCode, Angular)
- Multiple development environment setup.
- Tricks for productivity.

OSM Developer Resources

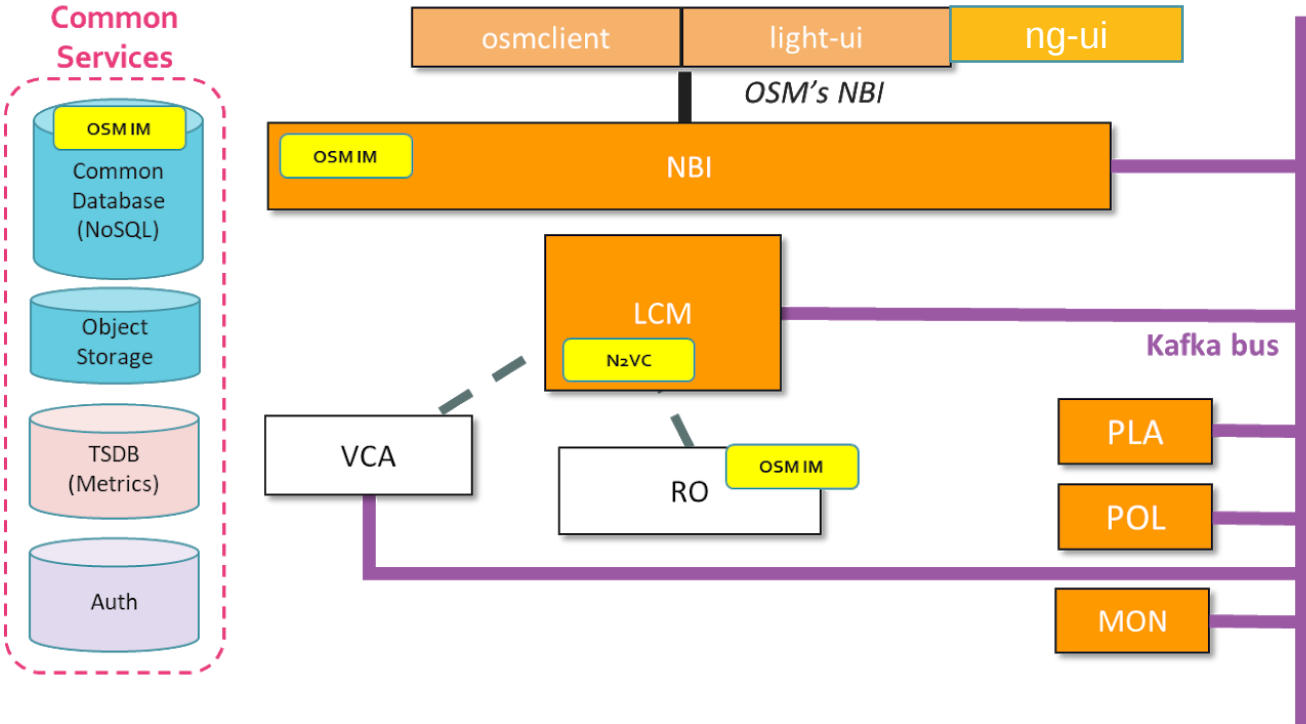
- Developer guide
 - <https://osm.etsi.org/docs/developer-guide/>
- Developer HowTo
 - https://osm.etsi.org/wikipub/index.php/Developer_HowTo
- Gitweb
 - <https://osm.etsi.org/gitweb/>
- Gerrit
 - <https://osm.etsi.org/gerrit/>
- Bugzilla
 - <https://osm.etsi.org/bugzilla/>
- Gitlab
 - <https://osm.etsi.org/gitlab/>
- More Information
 - Hackfest #9 presentation
 - <http://osm-download.etsi.org/ftp/osm-8.0-eight/OSM-MR9-hackfest/presentations/OSM-MR%239%20Hackfest%20-%20HD5.3%20-%20%20How%20to%20contribute%20to%20OSM.pdf>



Open Source
MANO

OSM Architecture Docker Swarm and Kubernetes

OSM Architecture



Docker Swarm/Kubernetes CLI Commands

Sl. No	Objective	Docker Swarm CLI Command	Kubernetes CLI Command
1	List services	\$ docker service ls	\$ kubectl get deployments -n osm
2	List containers/pods	\$ docker ps	\$ kubectl get pods -n osm
3	Enter container/pod	\$ docker exec -it <container-id> bash	\$ kubectl exec -it <pod-id> bash -n osm
4	Restart a service	\$ docker service scale <service-name>=0 \$ docker service scale <service-name>=1	\$ kubectl delete deployment <deployment-id> -n osm
5	Service logs	\$ docker service logs osm_nbi	\$ kubectl logs <pod-id> -n osm
6	Running service logs	\$ docker service logs -f osm_nbi	\$ kubectl logs <pod-id> -f -n osm
7	Update service image	\$ docker service update <service-name> --force --image <image-name>	\$ kubectl set image <deployment/statefulset>/<deployment/statefulset-name> <container-name>=<image-name>
8	Deploying stack	\$ docker stack deploy -c /etc/osm/docker/docker-compose.yaml osm	\$ kubectl apply -f /etc/osm/docker/osm_pods -n osm
9	Removing stack/ns	\$ docker stack rm osm	\$ kubectl delete ns osm
10	Building docker image	\$ docker build NBI -f NBI/Dockerfile.local -t opensourcemano/nbi:develop --no-cache	\$ docker build NBI -f NBI/Dockerfile.local -t opensourcemano/nbi:develop --no-cache

Bash Alias – Save your precious time

Tired of typing the same long commands over and over?
Do you feel inefficient working on the command line?
Bash aliases can make a world of difference.



Use ~/.bashrc file
to source
automatically

Step 1

- Prepare alias command
- Ex: Entering NBI container
 - `alias enbi="docker exec -it $(docker ps | grep nbi | awk '{print $1}') /bin/bash"`

Step 2

- Create a bash aliases file for osm dev
- `$ touch ./bash_aliases_osm_dev`

Step 3

- Add the prepared alias commands from step1 to file created in step2.

Step 4

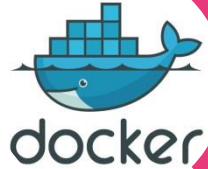
- Load them into your current bash session
- `$ source ./bash_aliases_osm_dev`



Open Source
MANO

Developing OSM NBI and OSM RO

Development styles for OSM NBI



Inside running container

- Enter running container of OSM NBI and edit the source files.
- For tracing of small code blocks.
- Quick hands-on.
- Will loose all changes once container restart!



Minimal development setup (on server)

- Clone the repo and follow initial repo setup from <https://osm.etsi.org/docs/developer-guide/>
- Code changes (tracing/bug fix/feature development)
- Run code and verify. (1: Using Python interpreter on host, 2 As Docker image)



Pro OSM Developer (on workstation)

- Install/Setup/Configure PyCharm/VsCode for your OS.
- Clone the repo and follow initial repo setup from <https://osm.etsi.org/docs/developer-guide/>
- Code changes (tracing/bug fix/feature development)
- Run code and verify. (1: Using Python interpreter on host, 2 As Docker image)

Dockerfile for OSM NBI

Official Dockerfile.local file

- Uses ubuntu 18.04 as base image
- Installs all the dependent packages
- Sets up ENV variables and configurations.
- Consumes lot of bandwidth and time.
- Good to use when codebase is stable.

Optimized Dockerfile for quick builds

- Uses nbi:8.0.4 as base image.
- Doesn't install any packages.
- Sets up ENV variables and configurations.
- Consumes very less bandwidth and time.
- Good to use when codebase is unstable or during testing logics.

Development styles for OSM RO



Minimal development setup (on server)

- Clone the repo and follow initial repo setup from <https://osm.etsi.org/docs/developer-guide/>
- Code changes (tracing/bug fix/feature development)
- Run code and verify. (1: Using Python interpreter on host, 2 As Docker image)



Pro OSM Developer (on workstation)

- Install/Setup/Configure PyCharm/VsCode for your OS.
- Clone the repo and follow initial repo setup from <https://osm.etsi.org/docs/developer-guide/>
- Code changes (tracing/bug fix/feature development)
- Run code and verify. (1: Using Python interpreter on host, 2 As Docker image)



Open Source
MANO

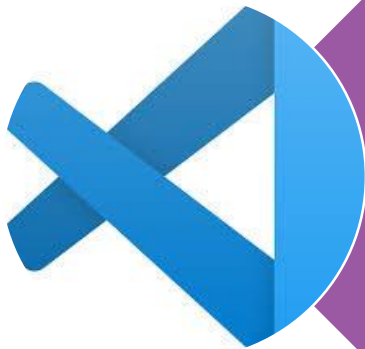
Developing OSM NG UI

Development styles for OSM NG-UI



Minimal development setup (on server)

- Clone the repo and follow initial repo setup from <https://osm.etsi.org/docs/developer-guide/>
- Code changes (tracing/bug fix/feature development)
- Run code and verify. (1: Using npm start or ng server on host, 2 As Docker image)



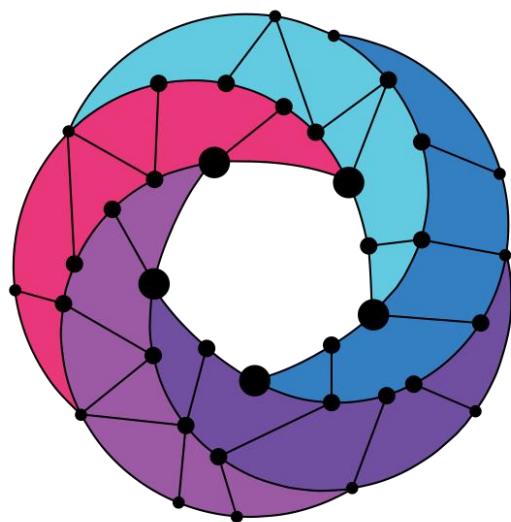
Pro OSM Developer (on workstation)

- Install/Setup/Configure VsCode for your OS.
- Clone the repo and follow initial repo setup from <https://osm.etsi.org/docs/developer-guide/>
- Code changes (tracing/bug fix/feature development)
- Run code and verify. (1: Using npm start or ng server on host, 2 As Docker image)



Open Source
MANO

Q & A



Open Source MANO

Find us at:

osm.etsi.org
osm.etsi.org/wikipub