

# Mini-projets

November 16, 2019

Pour les mini-projets vous allez soit développer (ou modifier) une application SDN soit analyser la réaction d'un contrôleur SDN à des événements particuliers. **Si cela n'est pas expressément indiqué, vous pouvez choisir librement entre ONOS ou OpenDayLight.**

Pour le développement des applications pour ODL, veuillez consulter le guide officiel disponible ici :

<https://docs.opendaylight.org/en/stable-sodium/developer-guide/developing-apps-on-the-openshift-controller.html>

Pour le développement des applications pour ONOS, veuillez consulter le guide officiel disponible ici :

<https://wiki.onosproject.org/display/ONOS/Template+Application+Tutorial>

Pour les autres projets, voici des références utiles

- **CPMan** : <https://wiki.onosproject.org/display/ONOS/Control+Plane+Management+Application>
- **VPLS Onos** <https://wiki.onosproject.org/display/ONOS/VPLS+User+Guide>
- **SFC ODL** : <https://docs.opendaylight.org/en/stable-sodium/user-guide/service-function-chaining.html>
- **SFC ONOS** : <https://wiki.onosproject.org/pages/viewpage.action?pageId=4163192>
- **Intent ONOS** : <https://wiki.onosproject.org/display/ONOS/Intent+Framework>
- **Intent ODL** : [https://docs.opendaylight.org/en/stable-fluorine/user-guide/network-intent-composition-\(nic\)-user-guide.html](https://docs.opendaylight.org/en/stable-fluorine/user-guide/network-intent-composition-(nic)-user-guide.html)

## Topologies

Afin tester vos applications avec une topologie, vous pouvez la générer à partir de **mininet**. Pour créer une topologie, vous pouvez créer un script python et coder la topologie à l'aide de Mininet (<http://mininet.org/walkthrough/>). Inspirez vous de l'exemple disponible sur github :

`https://raw.githubusercontent.com/mininet/mininet/master/custom/topo-2sw-2host.py`

Si vous le préférez, vous pouvez aussi vous appuyer sur les scripts utilisées pendant le TP SDN.

Pour les projets LB ONOS et LB ODL utilisez le topologie suivante :

