Tested on Ubuntu 20.04, with ONOS 3.0.0, Mininet 2.2.2, OVS 2.9.2, IntelliJ 2022.3.3

Tested on Debian 5.10.179 (2023), with ONOS 3.0.0, Mininet 2.3.0, OVS 2.9.2, IntelliJ 2022.3.3

ATTENTION: with ONOS 3.0.0 it is required to use bazel 6.0.0-pre.20220421.3 that is not supported by the last version of IntelliJ bazel plugin. Please use 2022.11.07.0.1-api-version-223

--- INSTALL JAVA and BAZEL

env
It is a framework for java project management and building (required by ONOS). It is required to build the ONOS controller.

Procedure to install BAZEL is based on the documentation at the following link:

- https://docs.bazel.build/versions/master/install-ubuntu.html

Procedure:

```
> sudo apt install openjdk-11-jdk
```

Current version of ONOS 3.0.0 requires 6.0.0-pre.20220421.3. Using the binary installer that can be downloaded.

https://github.com/bazelbuild/bazel/releases

After download:

```
> chmod +x 6.0.0-pre.20220421.3-installer-linux-x86_64.sh
```

To allow the execution of the file…

```
> ./6.0.0-pre.20220421.3-installer-linux-x86_64.sh --user
```

With the --user option bazel is installed in the user home/bin directory, then execute the following command to add the “bazel” command in the .bashrc file so that you can use it.

```
> export PATH="$PATH:$HOME/bin"
```

--- INSTALL RIGHT VERSION BAZEL

(NOT REQUIRED IF YOU ALREADY INSTALLED 6.0.0-pre.20220421.3.)

Using the following commands the bazellisk tool will be installed.

Such tools read the required version of bazel into the file .bazelversion and install it.

> wget https://github.com/bazelbuild/bazelisk/releases/download/v1.8.1/bazelisk-linux-amd64

> export BAZELISK_BASE_URL="https://releases.bazel.build/6.0.0/rolling"

> chmod +x bazelisk-linux-amd64

> sudo mv bazelisk-linux-amd64 /usr/local/bin/bazel

> cd ~/onos

> bazel version

--- INSTALL ONOS

It is an open source project implementing an SDN controller (our reference SDN controller).

Links:

- https://wiki.onosproject.org/display/test/Building+ONOS
- https://wiki.onosproject.org/display/ONOS/ONOS+from+Scratch
- https://wiki.onosproject.org/display/ONOS/Developer+Quick+Start

Procedure:

```
> sudo apt install git

> git clone https://gerrit.onosproject.org/onos
```

This step may require some minutes depending on your Internet connection.

After this step you have the onos folder in your home directory.

Add the string “. ~/onos/tools/dev/bash_profile" at the end of file .bashrc in your home directory.

```
> echo ". ~/onos/tools/dev/bash_profile" >> ../.bashrc
```

Then open a new linux terminal, to properly load environment variables.

Check the environment:

```
> env | grep ONOS
```

Fig: ONOS environment variables.

Specifically, the ONOS_ROOT variable should point to the folder where you have installed it. The default folder is /home/user/onos, but it could be different depending on your preferences (e.g., if you forked the project to a private github repository).

--- BUILD and RUN for the first time

```
> cd onos
```

You can use the script ok

```
> ok clean
```

or

```
> ok -- clean debug
```

or you can directly use the command

```
> bazel run onos-local -- clean debug
```

The first time you execute this command, it will take some time, up to 30-60 minutes.

The terminal will show a long textual log, when the text stop (without errors) it means that the ONOS controller is regularly running, and you can continue with the following commands.

--- In case of ERRORS during the building process, this is likely due to a package that is missing in your system… check that the following packages are installed:

```
> sudo apt install bzip2
> sudo apt install tar
> sudo apt install perl
> sudo apt install curl
> sudo apt install zip
```

--- In case of the following ERROR when you start ONOS: this appears when a wrong version of bazel is installed, so ONOS tries to dynamically download the correct version… but something goes wrong in the download. You can solve installing the right version, or even with the command reported below, that fixes the url for the download.

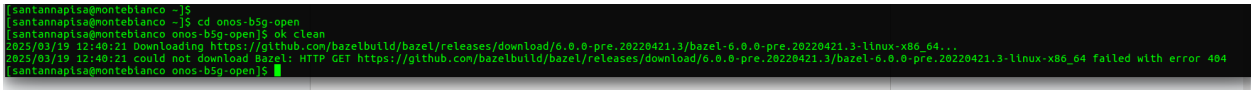


Fig: BAZEL error.

Update the URL to download the proper Bazel version:

```
> export BAZELISK_BASE_URL="https://releases.bazel.build/6.0.0/rolling"
```

To attach a shell to the ONOS CLI

```
> onos localhost
```

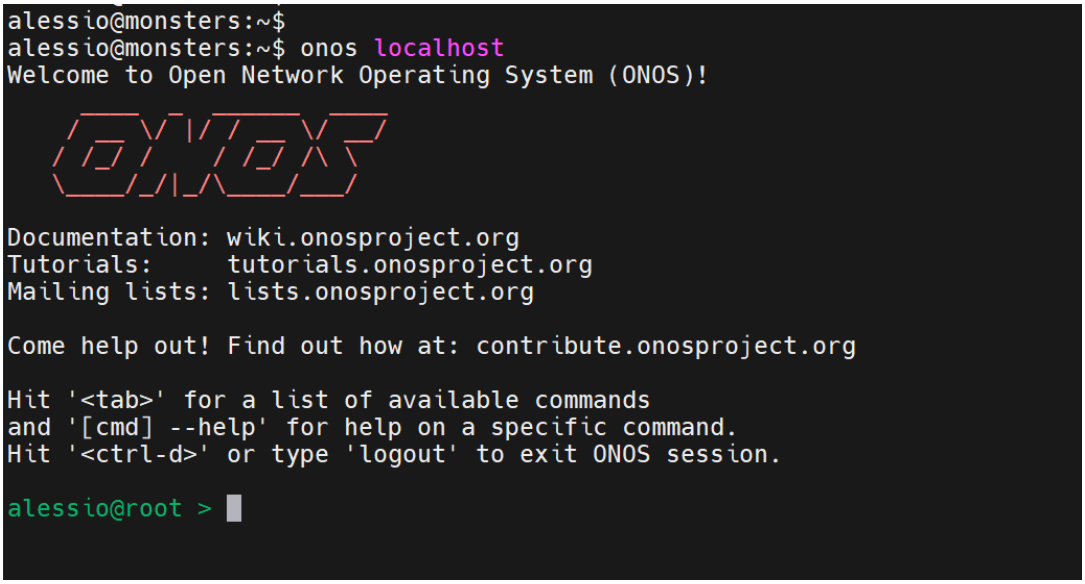


Fig: Example of ONOS CLI.

To connect to web GUI:

- http://localhost:8181/onos/ui/
- user: karaf
- pass: karaf

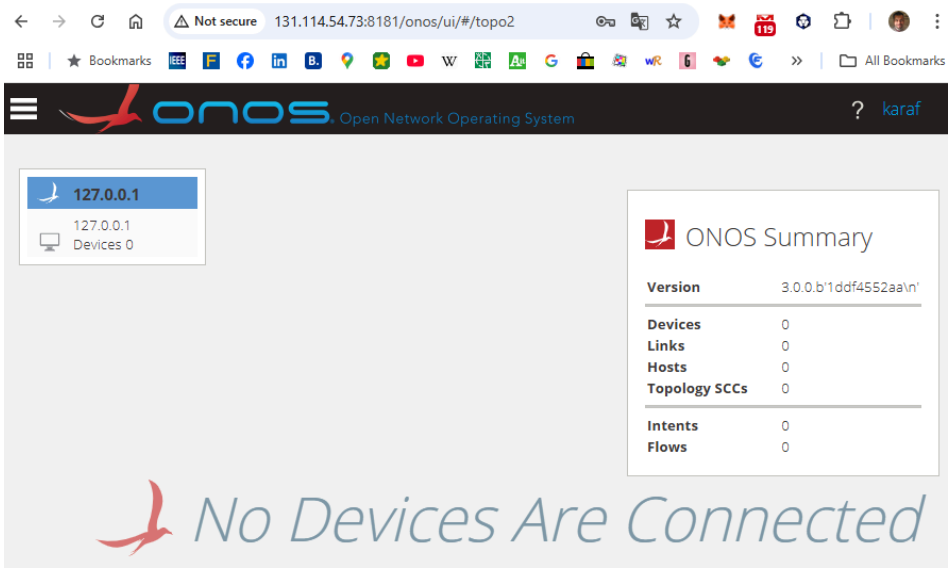


Fig: Example of ONOS web-based GUI

To connect to the ONOS REST interfaces:

- http://localhost::8181/onos/v1/docs/
- user: karaf
- pass: karaf

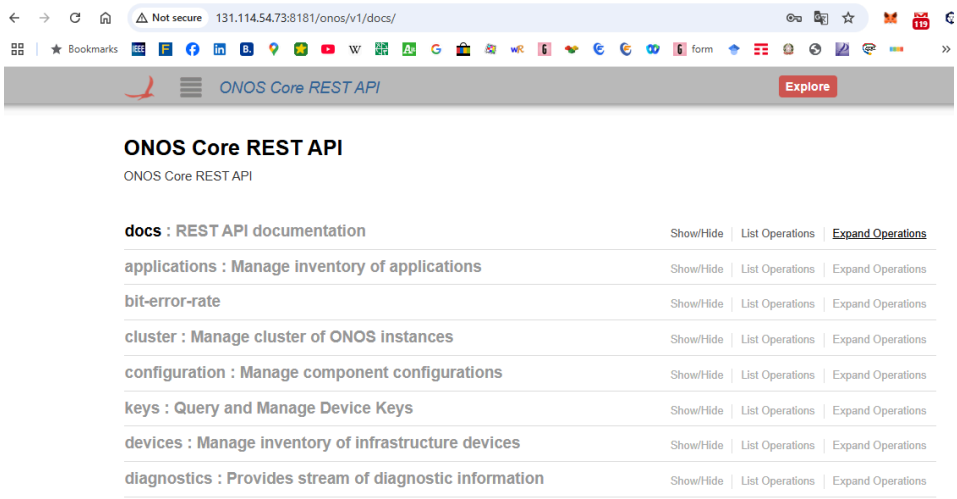


Fig: Example of ONOS REST APIs.

--- QUIT ONOS

In the ONOS CLI you can use the following command to reset the whole network vision. This

command is typically used when you stopped mininet and you want to remove memories of previously connected devices from the controller:

```
onos> wipe-out please
```

To stop ONOS you should use the script oh at the linux prompt

```
> oh
```

It is suggested to check if all ONOS processes are off.

```
> ps -aux | grep onos
```

If some processes are active you should manually kill before restarting the controller.

```
> kill process_id
```

Finally you can type CNTRL+C in the terminal where you launched ONOS.

bazel--- RUN ONOS

```
> ok clean
```

This command automatically checks if there is some modified code to build. If yes, it builds the new code then starts the controller. If not, it only starts the controller.

Run it with debugging logger:

```
> ok -- clean debug
```

If you need only to build the controller

```
> bazel build onos-local
```

If you need to start without building

```
> bazel run onos-local
```

--- USING ONOS CLI AND LOGGER

To enter the ONOS CLI

```
> onos localhost
```

Inside the ONOS CLI you can use a huge variety of commands to control the network. All commands support the --h option describing its use.

From the ONOS CLI if you want to enter the logger

```
onos> log:tail
```

To exit the logger you can type CNTRL+C, to exit the ONOS CLI you can type CNTRL+D.

```
12:59:19.934 INFO [ReactiveForwarding] Configured. Matching TCP/UDP fields is disabled
12:59:19.935 INFO [ReactiveForwarding] Configured. Matching ICMP (v4 and v6) fields is disabled
12:59:19.935 INFO [ReactiveForwarding] Configured. Ignore IPv4 multicast packets is disabled
12:59:19.936 INFO [ReactiveForwarding] Configured. record metrics  is disabled
12:59:19.937 INFO [ReactiveForwarding] Configured. Flow Timeout is configured to 10 seconds
12:59:19.937 INFO [ReactiveForwarding] Configured. Flow Priority is configured to 10
12:59:19.938 INFO [ReactiveForwarding] Configured. Inherit flow treatment is disabled
12:59:19.953 INFO [ReactiveForwarding] Started
12:59:19.954 INFO [FeaturesServiceImpl] Done.
12:59:19.956 INFO [ApplicationManager] Application org.onosproject.fwd has been activated
12:59:20.182 INFO [ReactiveForwarding] Configured. Packet-out only forwarding is disabled
12:59:20.183 INFO [ReactiveForwarding] Configured. Forwarding using OFPP_TABLE port is disabled
12:59:20.184 INFO [ReactiveForwarding] Configured. IPv6 forwarding is disabled
12:59:20.185 INFO [ReactiveForwarding] Configured. Match Dst MAC Only is disabled
12:59:20.185 INFO [ReactiveForwarding] Configured. Matching Vlan ID is disabled
12:59:20.186 INFO [ReactiveForwarding] Configured. Matching IPv4 Addresses is disabled
12:59:20.186 INFO [ReactiveForwarding] Configured. Matching IPv4 DSCP and ECN is disabled
12:59:20.187 INFO [ReactiveForwarding] Configured. Matching IPv6 Addresses is disabled
12:59:20.188 INFO [ReactiveForwarding] Configured. Matching IPv6 FlowLabel is disabled
12:59:20.188 INFO [ReactiveForwarding] Configured. Matching TCP/UDP fields is disabled
12:59:20.189 INFO [ReactiveForwarding] Configured. Matching ICMP (v4 and v6) fields is disabled
12:59:20.189 INFO [ReactiveForwarding] Configured. Ignore IPv4 multicast packets is disabled
12:59:20.190 INFO [ReactiveForwarding] Configured. record metrics  is disabled
12:59:20.191 INFO [ReactiveForwarding] Configured. Flow Timeout is configured to 10 seconds
12:59:20.191 INFO [ReactiveForwarding] Configured. Flow Priority is configured to 10
12:59:20.192 INFO [ReactiveForwarding] Configured. Inherit flow treatment is disabled
12:59:31.849 WARN [UiWebSocket] No GUI message handler for type alarmTopovDisplayStart
```

Fig: Example of ONOS events logger.

--- APPs STARTED AT ONOS BOOT

The file ~/onos/tools/dev/bash_profile contains the ONOS environment setup. Add the following line at the end of this file to select which applications have to be started with ONOS.

```
> export ONOS_APPS=drivers,openflow,gui,fwd
```

- Remember to open a new terminal, to reload the updated environment variables.

You can check which applications are currently loaded in your environment.

```
> env | grep ONOS_APPS
```

If you want to modify temporarily the list of applications that will be started with ONOS you can type the command in the terminal from which you are going to start ONOS:

```
> ONOS_APPS=drivers,openflow,gui ...
```

--- MININET

It is a software tool to emulate networks.

Links:

- http://mininet.org/walkthrough/
- https://wiki.onosproject.org/display/ONOS/Running+ONOS+with+onos.py

Procedure

```
> sudo apt install mininet bridge-utils pip
```

When the mininet software is installed, and the ONOS controller is active you can continue with the following commands to generate a network topology and connects all the devices to the controller.

Generate simple tree topology and connect to onos controller (ONOS controller must be active)

```
> sudo mn --controller=remote,ip=127.0.0.1 --topo tree,2,2
```

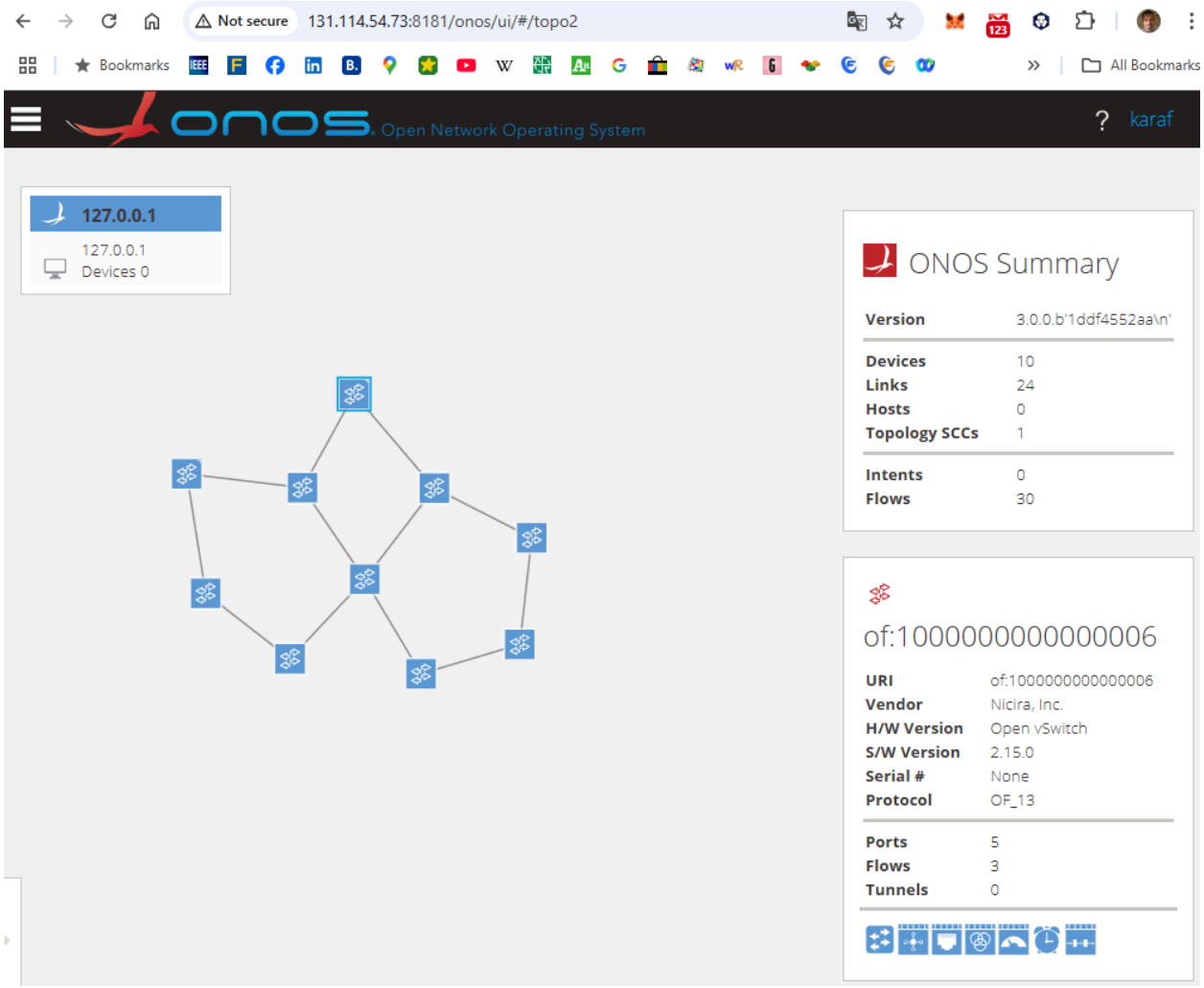
More complex topology

```
> sudo mn --controller=remote,ip=127.0.0.1 --topo torus,3,3
```

Topologies can be manually defined using a python script. Please have a look to the topology.py

file that is included in this same google drive. Download that file and save it inside the onos folder. Then you can generate the topology defined in the script with the following command, that essentially asks to mininet “please generate the topology here described”.

```
> sudo python3 ./topology.py
```



--- QUIT MININET

In the mininet prompt

```
mininet> quit
```

If mininet is not closed properly, the topology remains running, and emulated devices appears to be still connected to the ONOS controller (are visible in the GUI). To cancel the topology you can use this mininet command command at the linux prompt.

```
> sudo mn --clean
```

--- INTENLLIJ IDEA

It is a Java editor

Links

- https://wiki.onosproject.org/pages/viewpage.action?pageId=28836246

Procedure:

It is suggested to use version 2019.3.4 for Linux with bundled JBR 8 (tar.gz).

You can find and download IntelliJ 2019.3.4 version here:
https://www.jetbrains.com/idea/download/previous.html

Download the Community version 2019.3.4 for Linux tar.gz package (with JDK). After download execute the following commands.

```
> cd Downloads

> tar xfz ideaIC-2019.3.4.tar.gz

> cd idea-IC-193.6911.18/bin

> ./idea.sh
```

--- INSTALL BAZEL PLUGIN for IDEA

When Bazel is installed, after some options, at the Welcome panel of IntelliJ opens where you can decide to open/import/create a project. In this panel, on the bottom-right corner select

Configure->Plugins

Search for BAZEL, and install the BAZEL plugin, then restart IntelliJ.

Now in the Welcome panel of IntelliJ you find the option Import Bazel Project. Select this option, and configure the ONOS root folder as Workspace, then select the option

Create from scratch

This will take some tens of minutes to synchronize the whole project.

ATTENTION: with ONOS 3.0.0 it is required to use bazel 6.0.0-pre.20220421.3 that is not supported by the recent versions of bazel plugin. Therefore, you must install bazel plugin version: 2022.11.07.0.1-api-version-223.

--- INSTALL AN EXTERNAL ONOS APP

To install an external ONOS app you can use the we-based GUI, or you can type the following commands in the Linux terminal.

```
> onos-app localhost install ./target/packagename.oar
```

```
> onos-app localhost activate ./target/appname.app
```

To execute both previous commands in a single shot you can use the following:

```
> onos-app localhost install! packagename.oar
```

Is the app is already installed but you need to re-install is after modification:

```
> onos-app localhost reinstall! packagename.oar
```

--- EDIT EXTERNAL ONOS APP

Links:

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

To import and edit the Java code of an external ONOS application the following steps are required.

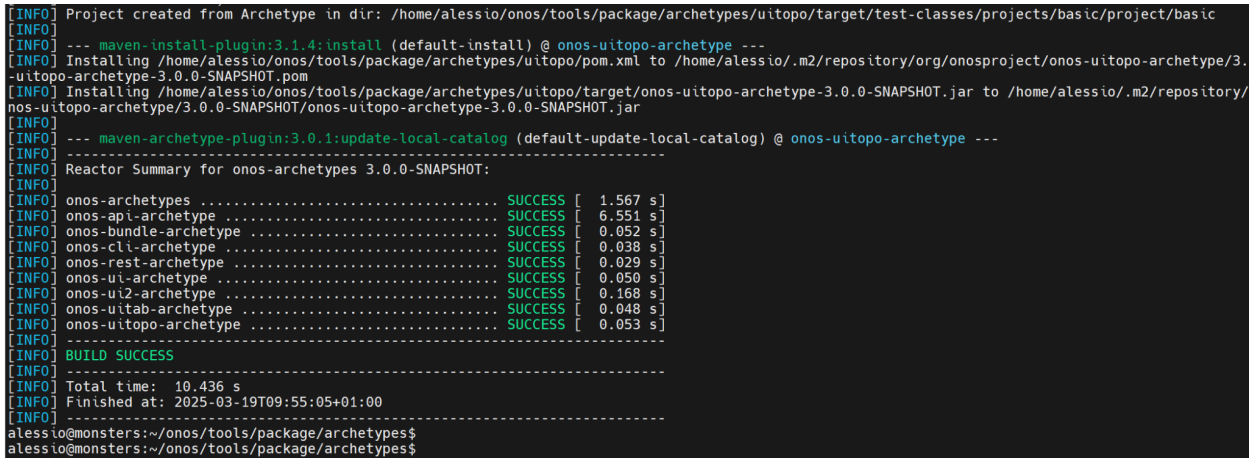
First you have to install some software in your system with the following commands. Maven is a tool utilized by ONOS to build the apps.

```
> sudo apt install maven
> sudo apt install pyhton-pip

> sudo pip install requests
```

Then you have to execute the following commands in the ONOS_ROOT folder. Those commands are required to allow external ONOS application to see and utilize all the services provided by the ONOS core.

```
> cd /home/alessio/onos/tools/package/archetypes
> mvn clean install
```




```
> cd /home/alessio/onos
> onos-publish -l
```

```
publishing: /onos-protocols-gnmi-stub/3.0.0-SNAPSHOT/onos-protocols-gnmi-stub-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-p4runtime-api/3.0.0-SNAPSHOT/onos-protocols-p4runtime-api-3.0.0-SNAPSHOT.pom
publishing: /onos-core-common/3.0.0-SNAPSHOT/onos-core-common-3.0.0-SNAPSHOT.pom
publishing: /onos-drivers-default/3.0.0-SNAPSHOT/onos-drivers-default-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-grpc-api/3.0.0-SNAPSHOT/onos-protocols-grpc-api-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-p4runtime-model/3.0.0-SNAPSHOT/onos-protocols-p4runtime-model-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-grpc-utils/3.0.0-SNAPSHOT/onos-protocols-grpc-utils-3.0.0-SNAPSHOT.pom
publishing: /onos-drivers-gnmi/3.0.0-SNAPSHOT/onos-drivers-gnmi-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-bgp-bgpio/3.0.0-SNAPSHOT/onos-protocols-bgp-bgpio-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-bgp-api/3.0.0-SNAPSHOT/onos-protocols-bgp-api-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-lisp-msg/3.0.0-SNAPSHOT/onos-protocols-lisp-msg-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-openflow-api/3.0.0-SNAPSHOT/onos-protocols-openflow-api-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-gnoi-api/3.0.0-SNAPSHOT/onos-protocols-gnoi-api-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-gnmi-api/3.0.0-SNAPSHOT/onos-protocols-gnmi-api-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-lisp-api/3.0.0-SNAPSHOT/onos-protocols-lisp-api-3.0.0-SNAPSHOT.pom
publishing: /onos-protocols-netconf-api/3.0.0-SNAPSHOT/onos-protocols-netconf-api-3.0.0-SNAPSHOT.pom
publishing: /onos-dependencies/3.0.0-SNAPSHOT/onos-dependencies-3.0.0-SNAPSHOT.pom
alessio@monsters:~/onos$
alessio@monsters:~/onos$
alessio@monsters:~/onos$
alessio@monsters:~/onos$
```

The execution of the two above commands may require some minutes (e.g., 10 minutes).

If the core software is modified, before using the updated core software from an external ONOS app, it is necessary to perform the above steps, i.e., artifact compilation and publication.

Then you can create and enter a folder for your external ONOS applications.

```
> cd
> mkdir onos-apps
> cd onos-apps
```

Then export the following environment variable.

```
> export ONOS_POM_VERSION=2.0.0
```

At this point you can download the pcklog-app folder from the OneDrive. Copy it in your local folder onos-apps. Then you can open the application using IntelliJ.

--- CREATE YOUR OWN ONOS APP

Links:

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

First you have to execute all the steps listed in the previous section.

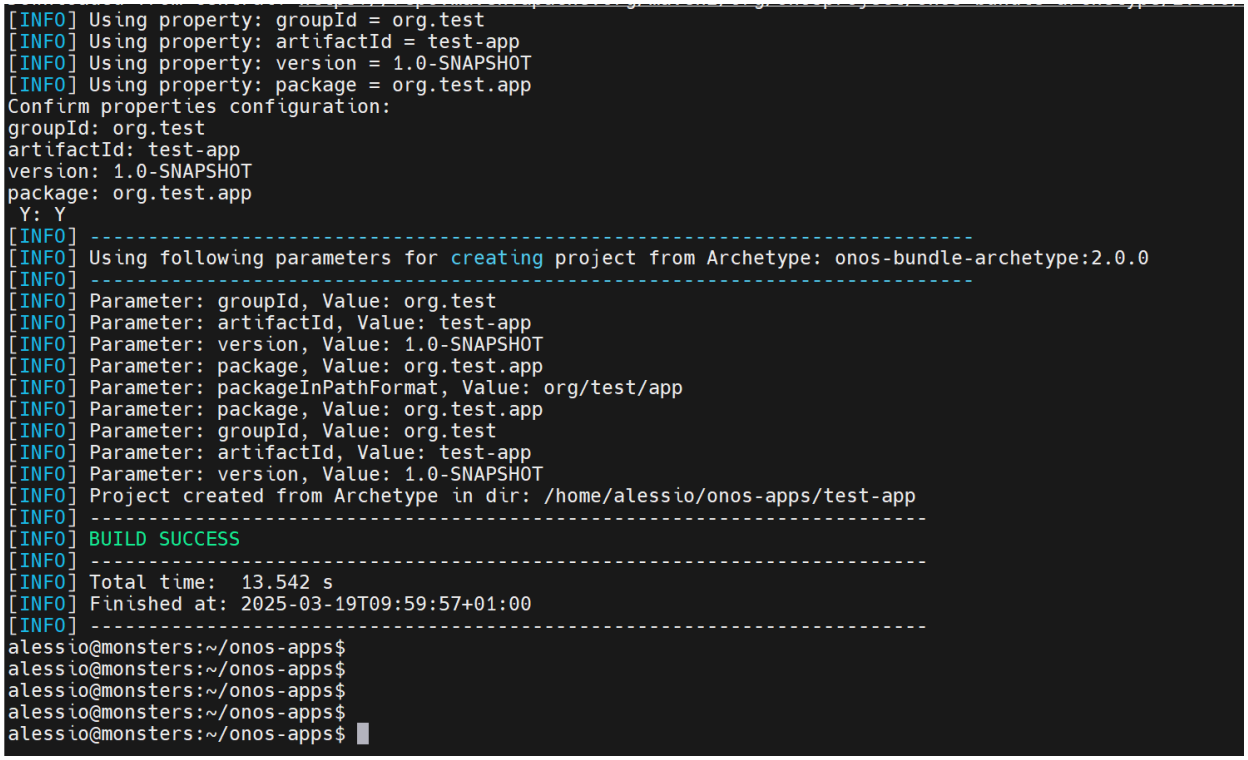
Then you can create and enter a folder for your own applications.

```
> cd
```

```
> mkdir onos-apps
> cd onos-apps
> export ONOS_POM_VERSION=2.0.0
```

Execute the onos script for creating an application with name “name”.

```
> onos-create-app app org.name name-app 1.0-SNAPSHOT org.name.app
```

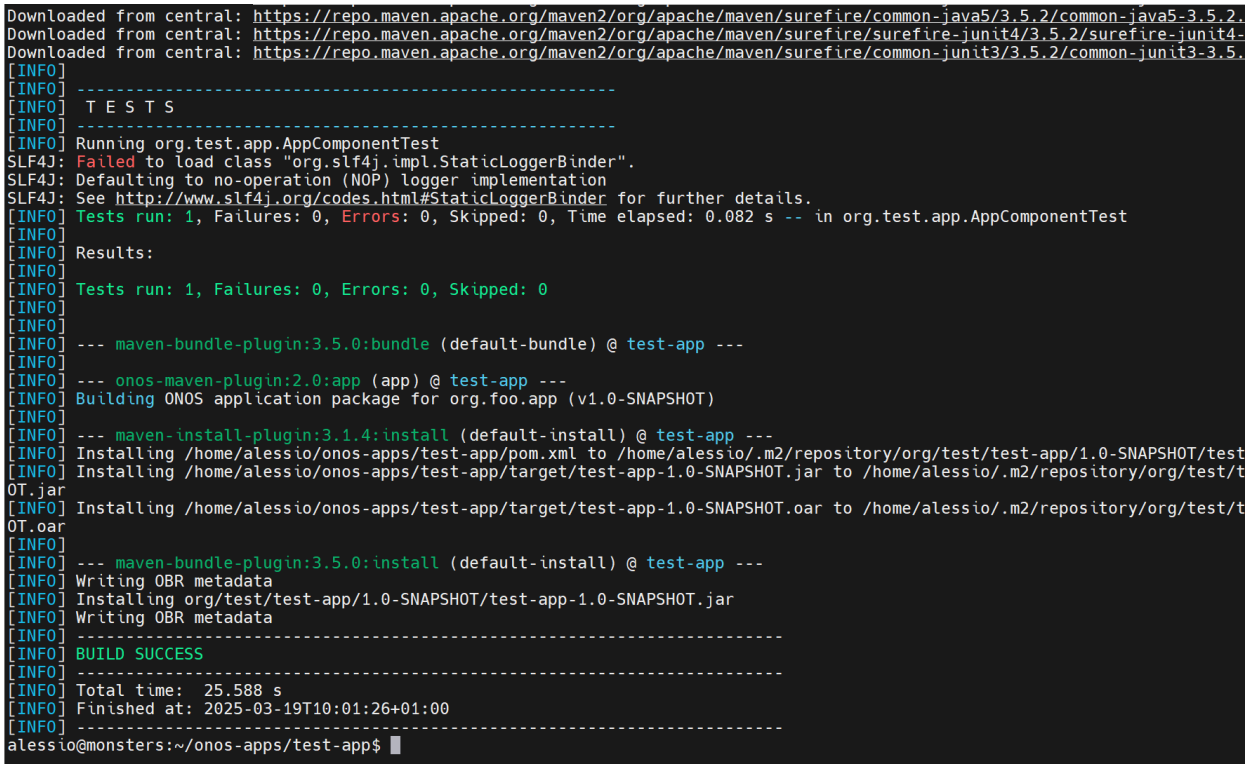


This command generates the app within the folder name-app. At this point it is needed to edit the properties section of the pom.xml file that you find in the created folder.

```
> cd name-app
> nano pom.xml
```

Specifically, in the file pom.xml, where you see the word “foo”, you have to write the name of your app. You can now enter the created folder and build the application.

```
> cd name-app
> mvn clean install
```



This step generates an .oar archive in the target folder, then you need to install the application. To install the application, the ONOS controller should be running. Then you can type the following commands to install and then activate the app, respectively.

```
> onos-app localhost install target/foo-app-1.0-SNAPSHOT.oar

> onos-app localhost activate org.foo.app
```

The Java code of your app can be edited, after modification you need to rebuild, reinstall and reactivate the application.

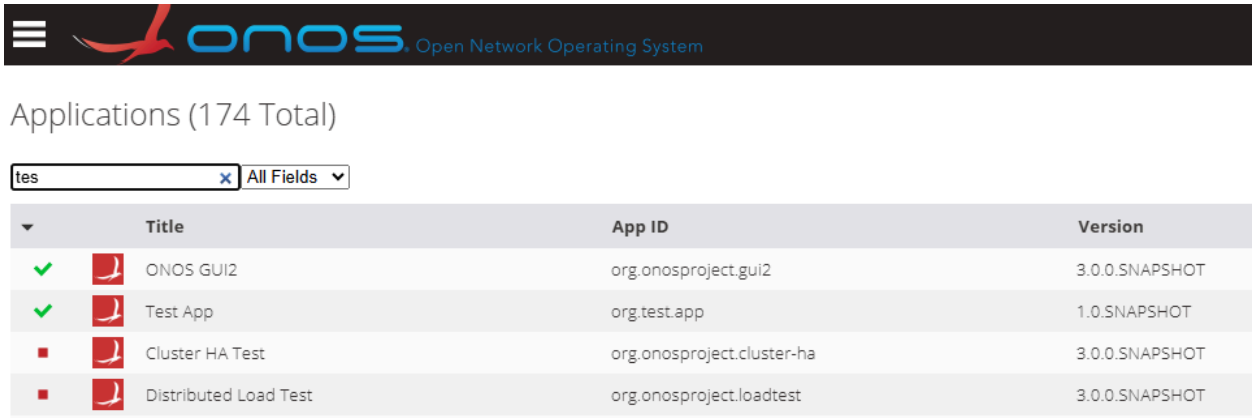
```
> mvn clean install

> onos-app localhost reinstall target/foo-app-1.0-SNAPSHOT.oar

> onos-app localhost activate org.foo.app
```

To execute contemporarily the two previous commands you can type.

```
> onos-app localhost reinstall! target/foo-app-1.0-SNAPSHOT.oar
```



Similarly it is possible to create an app that provides an example CLI command, REST interface or utilization of the web based GUI.

```
> onos-create-app cli     org.foo foo-app 1.0-SNAPSHOT org.foo.app
> onos-create-app rest    org.foo foo-app 1.0-SNAPSHOT org.foo.app
> onos-create-app ui      org.foo foo-app 1.0-SNAPSHOT org.foo.app
> onos-create-app uitab   org.foo foo-app 1.0-SNAPSHOT org.foo.app
> onos-create-app uitopo  org.foo foo-app 1.0-SNAPSHOT org.foo.app
```

As an example, the first command add to the app folder a file AppCommand.java where you can implement a command to be executed in the ONOS cli.

--- OPENVSWITCH OVS

OpenvSwitch is the tool for emulating virtual switches used by mininet, it is automatically installed by mininet. If the PC is switched off without explicit turn off of mininet, the ovs emulated switches will remain in the system.

You can visualize the list of currently emulated switches with the following command:

```
> sudo ovs-vsctl show
```

Links:

- https://www.openvswitch.org/
- http://www.openvswitch.org/support/dist-docs/ovs-vsctl.8.txt
- http://www.openvswitch.org//support/dist-docs/ovs-ofctl.8.txt

--- BUILD ONOS BEHIND a PROXY

If you are below an HTTP proxy configure an environment variable HTTP_PROXY and run the bazel build command with the following option:

> bazel build onos-local --action_env=HTTP_PROXY=$HTTP_PROXY