Setup process

Steps done

1. Created ssh key and added it in the FIT IoT-Lab platform

   ```
   ssh-keygen -t rsa
   ```

2. login to the console and authenticate using iotlab-auth

   ```
   ssh <login>@grenoble.iot-lab.info
   iotlab-auth -u <login>
   ```

▼ Setup the environment

   a. Clone repo

   ```
   git clone https://github.com/iot-lab/iot-lab.git
   cd iot-lab
   make
   =======================Output=======================================
   Welcome to the IoT-LAB development environment setup.

   targets:
           setup-aggregation-tools
           setup-cli-tools
           setup-contiki
           setup-iot-lab-contiki-ng
           setup-iot-lab.wiki
           setup-oml-plot-tools
           setup-openlab
           setup-riot
           setup-wsn430
           setup-zephyr

           pull
   ===================================================================
   ```

   b. Setup Contiki target

   ```
   make setup-contiki
   =======================Output=======================================
   Welcome to Contiki for IoT-LAB !
   ================================

   This repository is a fork of the official Contiki repository, bringing support for the IoT-LAB platforms.

   You may retrieve the last changes from the official repository with these commands:

       git remote add contiki https://github.com/contiki-os/contiki
       git fetch contiki
       git merge contiki/master

   Supported platforms:
   - iotlab-m3
   - iotlab-a8-m3

   Requirements:
   - gcc-toolchain: https://launchpad.net/gcc-arm-embedded
   - openlab (already checked-out if you used iot-lab)

   See this [tutorial](https://www.iot-lab.info/tutorials/contiki-compilation/) for explanations on how to setup your environmen

   Basic setup:
   - ``$ make TARGET=iotlab-m3     savetarget ``  # for m3 nodes
   - ``$ make TARGET=iotlab-a8-m3  savetarget ``  # for a8 nodes

   Further doc:
   - README-BUILDING.md
   - README-EXAMPLES.md
   ========================================================
   cd parts/contiki
   ```

c. run example

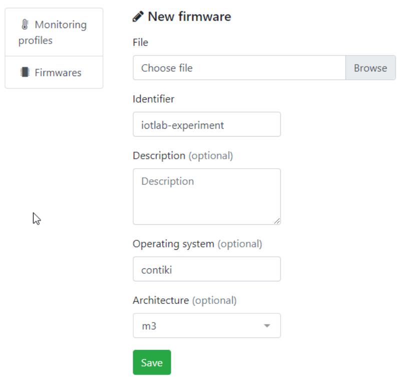
```
cd examples/iotlab/03-sensors-collecting/
cat README.md
=======================Output======================================
IoT-LAB Sensors Collecting
==========================

Prints all the available sensors once every second

Node M3
-------

    gyros: -8 -490 96 xyz m°/s
    light: 2.5552368E2 lux
    press: 9.8917236E2 mbar
    accel: -66 0 1080 xyz mg
    magne: -11 143 -438 xyz mgauss
    gyros: 1338 -385 -78 xyz m°/s
    light: 2.5463867E2 lux
    press: 9.891687E2 mbar
    magne: -15 148 -433 xyz mgauss
    accel: -61 -3 1070 xyz mg
    gyros: 840 -577 -367 xyz m°/s
    light: 2.542572E2 lux
    press: 9.891116E2 mbar
    magne: -9 150 -436 xyz mgauss
    gyros: 358 -332 -376 xyz m°/s
    accel: -60 -1 1076 xyz mg
    light: 2.538147E2 lux
    press: 9.8897876E2 mbar
    magne: -10 149 -435 xyz mgauss
    gyros: 105 236 -288 xyz m°/s
    light: 2.5352478E2 lux
    press: 9.890857E2 mbar
    accel: -62 -2 1072 xyz mg
    magne: -10 146 -439 xyz mgauss
    gyros: 778 -1015 201 xyz m°/s
    light: 2.565155E2 lux
    press: 9.8914575E2 mbar
    accel: -62 -3 1069 xyz mg
    magne: -9 148 -430 xyz mgauss
    gyros: 437 35 -87 xyz m°/s
    light: 2.4443054E2 lux
=============================END==================================
cat sensors-collecting.c
```

d. Compile the example

```
make TARGET=iotlab-m3
```

e. Submit an expirement with this firmware

```
iotlab-experiment submit -n contiki -d 15 -l 1,archi=m3:at86rf231+site=grenoble+mobile=0,sensors-collecting.iotlab-m3
```



New firmware

File: Choose file

Identifier: iotlab-experiment

Description (optional): Description

Operating system (optional): contiki

Architecture (optional): m3

f. Went back to

```
cd ~/iot-lab/parts/contiki/examples/iotlab/03-sensors-collectin
```

g. Compile the example

```
make TARGET=iotlab-m3
```

3. Compile the firmwares on SSH frontend

   a. Border router

```
cd ~/iot-lab/parts/contiki/examples/ipv6/rpl-border-router
make TARGET
ls
cp er-example-server.iotlab-m3 ~/
```

   b. **IoT-LAB version of CoAP Erbium server**

```
wget --no-check-certificate https://raw.githubusercontent.com/wiki/iot-lab/iot-lab/firmwares/contiki/border-router.iotlab-m3 -O bo
```

```
cd ~/iot-lab/parts/contiki/examples/iotlab/04-er-rest-example
make TARGET=iotlab-m3
ls
cp er-example-server.iotlab-m3 ~/
```

▼ **Public IPv6 (6LoWPAN/RPL) network with M3 nodes**

- Download firmware binaries on SSH
  - Border Router
  - HTTP server

```
wget --no-check-certificate https://raw.githubusercontent.com/wiki/iot-lab/iot-lab/firmwares/contiki/http-server.iotlab-m3 -O
```

- Choose an available IPv6 prefix for the site you are experimenting on. For example in Grenoble testbed :
  - we choose **2001:660:5307:3100::/64**
- Choose one node in your nodes list to implement the Border Router (BR) node
  - we chose node **m3-1**

4. Choose an available IPv6 prefix for the site you are experimenting on. For example in Grenoble testbed:

```
<login>@grenoble:~$ ip -6 route
===============================Output===============================
::1 dev lo proto kernel metric 256 pref medium
2001:660:5307:30ff::/64 dev ens3 proto kernel metric 256 pref medium
fe80::/64 dev ens3 proto kernel metric 256 pref medium
fe80::/64 dev ens7 proto kernel metric 256 pref medium
default via 2001:660:5307:30ff:ff:: dev ens3 metric 1024 onlink pref medium
```

- we choose **2001:660:5307:3100::/64**

5. Choose one node in your nodes list to implement the Border Router (BR) node

- we choose node m3-100.grenoble.iot-lab.info

```
sudo tunslip6.py -v2 -L -a m3-1 -p 20000 2001:660:5307:3100::1/64
```

6. Start tunslip6

```
sudo tunslip6.py -v2 -L -a m3-100.grenoble.iot-lab.info -p 20000 2001:660:5307:3100::1/64
```

7. Deploy the Border Router(BR)

```
iotlab-node --update ~/iot-lab/parts/contiki/examples/ipv6/rpl-border-router/border-router.iotlab-m3 -l grenoble,m3,100
```

8. Border Router IPv6

```
0356.796 *** Address:2001:660:5307:3100::1 => 2001:0660:5307:3100
0356.812  Starting 'Border router process' 'Web server'Got configuration message of type P
0356.812 Setting prefix 2001:660:5307:3100::
0357.804 Server IPv6 addresses:
0357.804  2001:660:5307:3100::b080
0357.804  fe80::b080
```

9. Flash CoAP server Firmware

```
iotlab-node --update ~/iot-lab/parts/contiki/examples/iotlab/04-er-rest-example/er-example-server.iotlab-m3 -e grenoble,m3,100

sys:1: DeprecationWarning: update command is deprecated and will be removed in next release. Please use flash instead.


{
    "0": [
        "m3-101.grenoble.iot-lab.info",
        "m3-102.grenoble.iot-lab.info",
        "m3-103.grenoble.iot-lab.info",
        "m3-104.grenoble.iot-lab.info",
        "m3-105.grenoble.iot-lab.info",
        "m3-106.grenoble.iot-lab.info",
        "m3-108.grenoble.iot-lab.info",
        "m3-109.grenoble.iot-lab.info"
    ],
    "1": [
        "m3-107.grenoble.iot-lab.info"
    ]
```

10. Grab the CoAP server node’s IPv6 address from the BR’s web interface

```
lynx -dump http://[2001:660:5307:3100::b080]
```

```

Neighbors
fe80::9881
fe80::a775
fe80::b576
fe80::8477
fe80::9382
fe80::1062
fe80::a881
fe80::9181

   Routes
2001:660:5307:3100::9181/128 (via fe80::9181) 1714s
2001:660:5307:3100::8477/128 (via fe80::8477) 1713s
2001:660:5307:3100::a881/128 (via fe80::a881) 1712s
2001:660:5307:3100::9382/128 (via fe80::9382) 1711s
2001:660:5307:3100::9881/128 (via fe80::9881) 1711s
2001:660:5307:3100::b576/128 (via fe80::b576) 1711s
2001:660:5307:3100::a775/128 (via fe80::a775) 1711s
2001:660:5307:3100::1062/128 (via fe80::1062) 1710s
```

11. ping CoAP server

```
ping6 2001:660:5307:3100::9181
```

12. designate a node as a sniffer in this case the node m3-100 and put it in a pcap file

```
sniffer_aggregator -l grenoble,m3,100 -o m3-100.pcap
```

13. For resource discovery, CoAP use the **/.well-known/core** path to provide resource descriptions in its CoRE Link Format. Send CoAP request with the aiocoap-client installed on the SSH frontend:

```
aiocoap-client coap://[2001:660:5307:3100::9181]:5683/.well-known/core
```

14. Test different GET requests to the different sensors resources

```
<login>@grenoble:~$ aiocoap-client coap://[2001:660:5307:3100::9982]:5683/sensors/light
0
<login>@grenoble:~$ aiocoap-client coap://[2001:660:5307:3100::9982]:5683/sensors/pressure
992
<login>@grenoble:~$ aiocoap-client coap://[2001:660:5307:3100::9982]:5683/sensors/gyros
-96;1557;-52
<login>@grenoble:~$ aiocoap-client coap://[2001:660:5307:3100::9982]:5683/sensors/accel
-107;-12;-1025
<login>@grenoble:~$ aiocoap-client coap://[2001:660:5307:3100::9982]:5683/sensors/magne
-75;-261;422
```

15. Once finished download the pcap file

```
scp <login>@grenoble.iot-lab.info:m3-100.pcap m3-100.pcap
```