OpenHAB2

- https://mysmarthomeweb.wordpress.com

Install

/etc/apt/sources.list.d/openhab2.list

```
cat > /etc/apt/sources.list.d/openhab2.list <<EOF
deb  [trusted=yes]
https://openhab.ci.cloudbees.com/job/openHAB-Distribution/ws/distributi
ons/openhab-offline/target/apt-repo/ /
deb  [trusted=yes]
https://openhab.ci.cloudbees.com/job/openHAB-Distribution/ws/distributi
ons/openhab-online/target/apt-repo/ /
EOF
```

```
wget -qO - 'http://www.openhab.org/keys/public-key-snapshots.asc' | sudo
apt-key add -
sudo apt update
apt install openjdk-8-jre
```

```
apt install openhab2-online

sudo systemctl enable openhab2.service
sudo systemctl start openhab2.service
sudo systemctl status openhab2.service
```

WEB GUI

- http://localhost:8080

CLI

```
/usr/share/openhab2/runtime/karaf/bin/client
```

Install features

show features

```
feature:list | grep network
```

interesting features

```
feature:install openhab-runtime-compat1x

feature:install openhab-transformation-regex
feature:install openhab-binding-exec
feature:install openhab-persistence-rrd4j

feature:install openhab-ui-habmin
feature:install openhab-ui-hapanel

feature:install openhab-action-telegram

feature:install openhab-binding-network
# feature:install openhab-binding-lgtv
```

Introspection

```
openhab> things list
yahooweather:weather:158197f6007 (Type=Thing, Status=ONLINE, Label=Weather
Information, Bridge=null)

openhab> items list
yahooweather_weather_158197f6007_temperature (Type=NumberItem, State=2,
Label=Temperature, Category=Temperature)
yahooweather_weather_158197f6007_humidity (Type=NumberItem, State=77,
Label=Humidity, Category=Humidity)
yahooweather_weather_158197f6007_pressure (Type=NumberItem, State=30308.19,
Label=Pressure, Category=Pressure)

openhab> links list
yahooweather_weather_158197f6007_temperature ->
yahooweather:weather:158197f6007:temperature
yahooweather_weather_158197f6007_humidity ->
yahooweather:weather:158197f6007:humidity
yahooweather_weather_158197f6007_pressure ->
yahooweather:weather:158197f6007:pressure
```

Log

console

```
# change log level
log:set DEBUG

# clear
log:clear
```

```
# show continuos tail
log:tail
```

Persistence

Enable rrd4j binding

/etc/openhab2/services/rrd4j.cfg

```
ctr5min.def=COUNTER,900,0,U,300
ctr5min.archives=AVERAGE,0.5,1,365:AVERAGE,0.5,7,300
ctr5min.items=systeminfo_computer_15819c96492_cpu_load,yahooweather_wea
ther_158197f6007_temperature,yahooweather_weather_158197f6007_humidity,
yahooweather_weather_158197f6007_pressure
```

/etc/openhab2/persistence/rrd4j.persist

```
// persistence strategies have a name and a definition and are referred
to in the "Items" section
Strategies {
    everyHour : "0 0 * * * ?"
    everyDay  : "0 0 0 * * ?"

    // if no strategy is specified for an item entry below, the default
list will be used
    default = everyChange
}

/*
 * Each line in this section defines for which item(s) which
strategy(ies) should be applied.
 * You can list single items, use "*" for all items or "groupitem*" for
all members of a group
 * item (excl. the group item itself).
 */
Items {
    // persist all items once a day and on every change and restore
them from the db at startup
    * : strategy = everyChange, everyDay, restoreOnStartup

    // additionally, persist all temperature and weather values every
hour
    Temperature*, Weather* : strategy = everyHour
}
```

demo mode

/ect/openhab2/services/addons.cfg

```
package = demo
```

Programming

items

items/cpu.items

```
// create an item from another channel item "items list"
Number Cpu "Cpu [%.0f]" (signal) {
channel="systeminfo:computer:15819c96492:cpu#load" }

// fetch data from script every 2 seconds (install exec binding and
regex transformation)
Number TestLoad "Test load [%.0f]" (signal) {
exec="<[/etc/openhab2/test.sh:1000:REGEX((.*?))]"
```

rules/cpu.rules

```
import org.openhab.core.library.types.*
import org.openhab.core.persistence.*
import org.openhab.model.script.actions.*

val Timer timer = null
val Timer waitTimer = null
val int sleepMinutes = 1

rule "CPU too high"
when
    Item Cpu changed
then
    if (Cpu.state > 40) {
      if (timer == null) {
        // watch cpu for 5 seconds
    timer = createTimer(now.plusSeconds(5)) [|
          val message = "CPU troppo alta: " + Cpu.state.toString() + ".
Prossimo messaggio in "+ sleepMinutes.toString() + " minuti"
          sendTelegram("bot1", message)
      logWarn("home", message)
```

```
      waitTimer = createTimer(now.plusMinutes(sleepMinutes)) [|
         timer = null
         sleepMinutes = sleepMinutes*2
          ]
    ]
      }
    } else {
      if(timer != null) {
         timer.cancel
         timer = null
         sleepMinutes = 1
      }
    }
end
```