

# Clochette and co

- [Les machines](#)
- [Config réseau](#)
- [Samba](#)
- [Hass](#)
  - [Api de hass](#)
  - [Maintenance](#)
  - [Links](#)
  - [vm](#)
- [Rhasspy 2 \(v1\)](#)
  - [Maintenance / update/install](#)
  - [Commande rasp](#)
  - [Rhasspy 3 \(v2\)](#)
- [Zigbee](#)
- [wall-e v4](#)

# Les machines

- **Wall-e**, le presque-routeur/nas sur 10.10.10.10, wall-e.home
  - dns avec "unbound"
  - dhcp avec "dhcpd"
  - serveur samba
  - Home assistant : port 8123
  - serveur MQTT
  - mopidy (musique.home)
  - serveur samba
- **Eve**, la raspberry sur 10.10.10.9
  - commande vocal: <https://rhasspy.readthedocs.io> : port 12101
  - capteur de temperature
- **Cri-kee**, le A5 sous android
- **Picsou**, la tour sur 10.10.10.11
- **Winnie**, le pc de l'école sur 10.10.10.12
- **Maitre yoda**, la commande vocal
- **La box**, 10.10.10.1
- **Chihiro**, la lampe de la cuisine - 10.10.10.211
- **Totoro**, la lampe de la chambre (zigbee)
- **Hauru**, la prise intelligente qui commande la guirlande (zigbee)
- **Sheeta**, la lampe du salon (zigbee) (le chateau dans le ciel)
- **Ponyo**, la lampe du bureau (zigbee)
- **luna (lovegood)** a5 sous linux

# Libre

- **Tigrou**, pour un ordinateur ou un serveur
- **Marko** pour un petit objet

# Config réseau

## DHCP

```
#
# DHCP Server Configuration file.
# see /usr/share/doc/dhcp-server/dhcpd.conf.example
# see dhcpd.conf(5) man page
#
# OpenNIC DNS

option domain-name-servers 10.10.10.8, 1.1.1.1;
option subnet-mask 255.255.255.0;
option routers 10.10.10.1;
subnet 10.10.10.0 netmask 255.255.255.0 {
    range 10.10.10.20 10.10.10.250;

    host livebox{
        hardware ethernet a4:3e:51:3e:62:15;
        fixed-address 10.10.10.1;
    }

    host picsou{
        hardware ethernet 4c:cc:6a:fc:0f:28;
        fixed-address 10.10.10.11;
    }

    host winnie{
        hardware ethernet e4:b3:18:8c:c8:96;
        fixed-address 10.10.10.12;
    }

    host cri-kee{
        hardware ethernet 14:9f:3c:6b:9e:2b;
```

```

        fixed-address 10.10.10.13;
    }

    host ewewifi{
        hardware ethernet b8:27:eb:3f:8b:54;
        fixed-address 10.10.10.9;
    }

    host totoro{
        hardware ethernet 08:be:ac:03:29:09;
        fixed-address 10.10.10.200;
    }

    host chihiro{
        hardware ethernet 24:62:AB:07:1B:6E;
        fixed-address 10.10.10.211;
    }
}

```

# DNS

/etc/unbound/conf.d/external-dns.conf :

```

forward-zone:
  name: "."
  forward-addr: 208.67.222.222
  forward-addr: 208.67.220.220

```

/etc/unbound/unbound.conf :

```

view:
  name: "home"
  include: /etc/unbound/local.d/home.conf

```

/etc/unbound/local.d/home.conf :

```

local-data: "wall-e.home.      3600 IN A 10.10.10.10"
local-data: "musique.home.    3600 IN A 10.10.10.10"
local-data: "photo.home.      3600 IN A 10.10.10.10"

```

```
local-data: "music.home.      3600 IN A 10.10.10.10"
```

```
local-data: "electric.home.   3600 IN A 10.10.10.8"
```

```
local-data: "money.home.      3600 IN A 10.10.10.8"
```

```
local-data: "eve.home.        3600 IN A 10.10.10.9"
```

```
local-data: "voice.home.      3600 IN A 10.10.10.9"
```

```
local-data: "chihiro.home.    3600 IN A 10.10.10.211"
```

```
local-data: "totoro.home.     3600 IN A 10.10.10.200"
```

/etc/unbound/local.d/adServers.conf

<https://pgl.yoyo.org/adservers/serverlist.php?hostformat=unbound&showintro=0&mimetype=plain text>

/etc/unbound/local.d/interface.conf

```
interface: 10.10.10.8
```

```
access-control: 10.10.10.1/24 allow
```

# Network Manager

/etc/NetworkManager/system-connections/Clochette\ 1.nmconnection

```
[connection]
```

```
id=Clochette 1
```

```
uuid=79af5700-b779-46b6-a8db-698531d0e9b7
```

```
type=wifi
```

```
interface-name=wlp3s0f3u2u2
```

```
timestamp=1682974157
```

```
[wifi]
```

```
mode=infrastructure
```

```
ssid=Clochette
```

```
[wifi-security]
```

```
auth-alg=open
```

```
key-mgmt=wpa-psk
```

```
psk=foretdesrevesbleus
```

[ipv4]

address1=10.10.10.8/24,10.10.10.1

dns=1.1.1.1;

method=manual

[ipv6]

addr-gen-mode=default

method=auto

[proxy]

```
nmcli connection mod "Clochette 1" connection.autoconnect yes
```

# Samba

/etc/samba/smb.conf:

## [global]

```
workgroup = Pixar
server string = Wall-e truck
security = user
hosts allow = 10.10.10.

passdb backend = tdbsam

guest account = hal
#printing = cups
#printcap name = cups
#load printers = yes
#cups options = raw
```

## [Truck]

```
path = /mnt/truck
browsable = yes
writable = yes
read only = no
public = yes
guest ok = yes
create mask = 0664
directory mask = 0775
force user = hal
```

# Hass



Hass

# Api de hass

**Doc :** <https://developers.home-assistant.io/docs/api/rest/>

`/api/states:`

switch.prise\_bureau\_switch / 47 / hauru

weather.toulouse / 49 / meteo france

7398c1c0495011eb9672a3e91d842fb7 / 21 / meteo france pluie dans l'heure

device\_tracker.linux\_travail / 58 / pc travail internet

device\_tracker.picsou\_1 / 59 / picsou wifi

device\_tracker.totoro / 56 / totoro wifi

device\_tracker.chihiro / 55 / chihiro wifi

device\_tracker.winnie / 54 / winnie ifi

device\_tracker.cri\_kee / 53 / cri-kee wifi

device\_tracker.picsou / 62 / picsou cable

device\_tracker.eve\_1 / 63 / eve cable

sensor.chihiro\_status / 67 / chihiro light

light.chihiro\_chihiro / 68 / chihiro status

Hass

# Maintenance

- `sudo -i -u hass`
- <https://www.home-assistant.io/installation/raspberrypi#install-home-assistant-core>

# Update

```
pip3 install --upgrade homeassistant
```

Hass

# Links

**Detecteur d'objets:** <https://github.com/snowzach/doods>

**Un exemple super stylé:** <https://imgur.com/a/8BHxBVN#XQsYIGS>

**Son lien reddit:**

[https://www.reddit.com/r/homeassistant/comments/lqo7wr/ha\\_2\\_years\\_later\\_and\\_what\\_ive\\_built\\_so\\_far/](https://www.reddit.com/r/homeassistant/comments/lqo7wr/ha_2_years_later_and_what_ive_built_so_far/)

Hass

# vm

```
mount -t cifs -o username=homeassistant,password=toto01,domain=WORKGROUP //192.168.122.36/share  
/mnt/hass/
```

## Autostart

```
systemctl enable virtnetworkd.service
```

[https://docs.redhat.com/fr/documentation/red\\_hat\\_enterprise\\_linux/9/html/configuring\\_and\\_managing\\_virtualization/starting-virtual-machines-automatically-when-the-host-starts\\_assembly\\_starting-virtual-machines#starting-virtual-machines-automatically-when-the-host-starts\\_assembly\\_starting-virtual-machines](https://docs.redhat.com/fr/documentation/red_hat_enterprise_linux/9/html/configuring_and_managing_virtualization/starting-virtual-machines-automatically-when-the-host-starts_assembly_starting-virtual-machines#starting-virtual-machines-automatically-when-the-host-starts_assembly_starting-virtual-machines)

# Rhasspy 2 (v1)

# Maintenance / update/install

```
sudo systemctl start/stop/status voice
```

<https://rhasspy.readthedocs.io/en/latest/installation/#virtual-environment>

## Installation

- (clone le repo)
- `export PIP_VERSION="pip<=20.2.4"`
- `./configure RHASSPY_LANGUAGE=fr --enable-in-place`
- `make`

## With Gruut:

- `make install-init`
- comment or delete all install-init in the makefile
- `source .venv/bin/activate`
- install `onnxruntime` with pip from `https://github.com/synesthesiam/prebuilt-apps/releases`
- install `phonetisaurus` with pip from `https://github.com/rhasspy/phonetisaurus-pypi/releases`
- install `gruut` `https://github.com/rhasspy/gruut/releases/tag/v0.9.0`
- `make install` in rhasspy dir

## Without Gruut:


- Edit `setup.py` and answer "no" to gruut installation
- `make install` in rhasspy dir

## Microphone

- `sudo apt install pulseaudio`
- `pactl list sinks`
- `pactl set-default-sink 0`
- `vim ~/.asoundrc`

```
pcm.!default {  
    type asym playback.pcm { type plug slave.pcm "hw:0" }  
    capture.pcm { type plug slave.pcm "hw:1" }  
}
```

- reboot

 Audio Recording arecord

Starts an arecord process locally and reads audio data from its standard out ([Documentation](#))

Device:

Refresh Test

UDP Audio (Output)   (outside ASR listening)

Output siteId

Rhasspy 2 (v1)

# Commande rasp

- <https://memodugeek.info/augmenter-la-taille-du-swap-sur-le-raspberry-pi/>



Rhasspy 2 (v1)

# Rhasspy 3 (v2)

## ALSA

Lister les cartes audio:

```
arecord -l
```

```
aplay -l
```

Mettre par défaut une carte:

```
/etc/asound.conf
```

```
pcm.!default {  
    type plug  
    slave.pcm {  
        type hw  
        card 2  
    }  
}
```

## Config

```
programs:  
mic:  
arecord:  
command: |  
    arecord -q -r 16000 -c 1 -f S16_LE -t raw -  
adapter: |  
    mic_adapter_raw.py --rate 16000 --width 2 --channels 1
```

snd:

  mplayer:

    command: |

      mplayer - -cache 1024 -volume 7 -rawaudio channels=1:rate=22050:samplesize=2 -demuxer rawaudio

    adapter: |

      snd\_adapter\_raw.py --rate 22050 --width 2 --channels 1

wake:

  porcupine1:

    command: |

      .venv/bin/python3 bin/porcupine\_stream.py --model "\${model}" "\${sensitivities}" --lang\_model

  "\${lang\_model}"

    template\_args:

      model: "\${data\_dir}/framboise\_linux.ppn"

      lang\_model: "\${data\_dir}/porcupine\_params\_fr.pv"

      sensitivities: "0.6"

asr:

  whisper-cpp.client:

    command: |

      client\_unix\_socket.py var/run/whisper-cpp.socket

  larynx2:

    command: |

      client\_unix\_socket.py var/run/vosk.socket

handle:

  date\_time:

    command: |

      bin/date\_time.py

    adapter: |

      handle\_adapter\_text.py

  ratatouille:

    command: |

      bin/converse.py "\${url}"

    adapter: |

      handle\_adapter\_text.py

    template\_args:

      url: "http://127.0.0.1:5555"

tts:

  piper.client:

    command: |

      client\_unix\_socket.py var/run/piper.socket

servers:

```

asr:
  whisper-cpp:
    command: |
      script/server "${data_dir}/ggml-small.bin"
  vosk:
    command: |
      script/server "${data_dir}/vosk-model-fr-0.22"
tts:
  piper:
    command: |
      script/server "${model}"
    template_args:
      model: "${data_dir}/fr-siwis-low.onnx"

pipelines:
  default:
    mic:
      name: arecord
    wake:
      name: porcupine1
    asr:
      name: vosk.client
    handle:
      name: ratatouille
    tts:
      name: piper.client
    snd:
      name: mplayer

```

## Customisation wake word

config/programs/wake/porcupine1/bin/porcupine\_stream.py

```

if keyword_index >= 0:
    os.system("/usr/bin/mplayer -volume 10 /srv/rhasspy/media/long-oiseau.mp3 &>/dev/null &")
write_event(
    ...

```

# Tester tts

```
script/run bin/tts_speak.py 'Bonjour'
```

# Tester ratatouille

```
echo "est-ce qu'il va pleuvoir de demain" | config/programs/handle/ratatouille/bin/converse.py  
http://10.10.10.11:5555
```

# Zigbee

<https://www.zigbee2mqtt.io/>

La config est là: `/home/tjiho/zigbee/zigbee2mqtt/data` sur wall-e

Pour ajouter un nouvel objet, mettre `permit_join:true` puis reset l'objet.

Il faudra redemarrer le service `systemctl restart zigbee.service`

- pour la prise (hauru), appui long suivie d'un clignotement orange de la led jusqu'a ce que la led arrête de clignoter
- l'ampoule (sheeta) je sais pas

## Unit systemd

```
# cat /etc/systemd/system/zigbee.service

[Unit]
Description=zigbee2mqtt
After=mosquitto.service

[Service]
ExecStart=/bin/bash -c 'source /home/tjiho/zigbee/env/bin/activate; /home/tjiho/zigbee/env/bin/npm start'
WorkingDirectory=/home/tjiho/zigbee/zigbee2mqtt
StandardOutput=inherit
StandardError=inherit
Restart=always
User=tjiho

[Install]
WantedBy=multi-user.target
```

# wall-e v4

## publish-external-temp.sh

```
URL="https://data.toulouse-metropole.fr/api/explore/v2.1/catalog/datasets/03-station-meteo-toulouse-busca/records?select=temperature_en_degre_c&order_by=heure_de_paris%20DESC&limit=1"
```

```
mosquitto_pub -h 10.10.10.8 -t sensors/jardin-des-plantes/temperature -m $(curl $URL | jq  
.results[0].temperature_en_degre_c) -r
```

## run\_sattelite.sh

```
cd /home/tjiho/repos/wyoming-satellite  
/home/tjiho/repos/wyoming-satellite/script/run \  
--name 'walle' \  
--uri 'tcp://0.0.0.0:10700' \  
--mic-command 'arecord -r 16000 -c 1 -f S16_LE -t raw' \  
--snd-command 'mplayer - -cache 1024 -volume 70 -rawaudio  
channels=1:rate=22050:samplesize=2 -demuxer rawaudio -really-quiet' \  
--awake-wav /home/tjiho/repos/wyoming-satellite/sounds/beer2.wav \  
--done-wav /home/tjiho/repos/wyoming-satellite/sounds/done.wav \  
--vad \  
--wake-uri unix:///home/tjiho/sockets/porcupine.sock \  
--wake-word-name 'framboise' \  
--mic-no-mute-during-awake-wav \  
--debug \  
--mic-auto-gain 0 \  
--mic-noise-suppression 0
```

## run\_vosk.sh

```
cd /home/tjiho/repos/wyoming-vosk  
/home/tjiho/repos/wyoming-vosk/script/run --uri 'tcp://0.0.0.0:10701' --data-dir data/ --model-for-language fr  
vosk-model-fr-0.22 --sentences-dir sentences --correct-sentences --limit-sentences --debug
```

# run\_wake.sh

```
cd /home/tjiho/repos/wyoming-porcupine  
#/home/tjiho/repos/wyoming-porcupine1/script/run --uri unix:///home/tjiho/sockets/porcupine.sock --system linux  
--sensitivity 0.5 --debug  
  
/home/tjiho/repos/wyoming-porcupine/script/run --uri unix:///home/tjiho/sockets/porcupine.sock --system linux --  
sensitivity 0.5 --debug --custom-keyword-dir /home/tjiho/ --access-key  
'b5c4/0xAxSS2OvhQXyF6e4cLSedpsmvvef+19HXDI9ho9G4Na0jhjA=='  
  
#/home/tjiho/repos/wyoming-porcupine1/script/run --uri tcp://0.0.0.0:10702 --system linux --sensitivity 0.5 --  
debug
```

# Firewall

```
firewall-cmd --permanent --add-port 10702/tcp --zone libvirt  
firewall-cmd --permanent --add-port 10702/udp --zone libvirt  
firewall-cmd --permanent --add-port 10701/udp --zone libvirt  
firewall-cmd --permanent --add-port 10701/tcp --zone libvirt  
firewall-cmd --permanent --add-port 10700/udp --zone libvirt  
firewall-cmd --permanent --add-port 10700/tcp --zone libvirt
```