IMACS server installation and set-up

Install Ubuntu server 14.04 with default settings for English.

Do not choose any extra packages.

Machine name: imacs

User: imacs

Password: [!m@c5](mailto:!m@c5)

# Initial

Get all software for initial installation up to date, run:

|  |
| --- |
| $ sudo apt-get update  $ sudo apt-get upgrade |

Install Openssh server for easier access

|  |
| --- |
| $ sudo apt-get install openssh-server |

Connect to console with:

|  |
| --- |
| $ ssh imacs@[ip address] |

# Influxdb

Run the following:

|  |
| --- |
| $ curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -  $ source /etc/lsb-release  $ echo "deb https://repos.influxdata.com/ubuntu trusty stable" | sudo tee /etc/apt/sources.list.d/influxdb.list |

Do installation

|  |
| --- |
| $ sudo apt-get update && sudo apt-get install influxdb  $ sudo service influxdb start |

Test if services are running:

Look for funnies in log file:

|  |
| --- |
| $ tail -f /var/log/influxdb/influxd.log |

Check if admin web is running

|  |
| --- |
| http://[IP Adress]:8083/ |

# Chronograf

Download and install packages

|  |
| --- |
| $ wget <https://dl.influxdata.com/chronograf/releases/chronograf_0.13.0_amd64.deb>  $ sudo dpkg -i chronograf\_0.13.0\_amd64.deb |

Change config to acces from other server

|  |
| --- |
| $ sudo nano /opt/chronograf/config.toml |

Uncomment the following line

|  |
| --- |
| Bind = "0.0.0.0:10000" # <- This will expose Chronograf to the public internet. Use with caution! |

Comment the following line:

|  |
| --- |
| # Bind = "127.0.0.1:10000" |

Start/stop services

|  |
| --- |
| $ sudo service chronograf *start/stop* |

Interface URL

|  |
| --- |
| http://79.6.23.3:10000/ |

# Kapacitor

Download and install files

|  |
| --- |
| $ wget <https://dl.influxdata.com/kapacitor/releases/kapacitor_0.13.1_amd64.deb>  $ sudo dpkg -i kapacitor\_0.13.1\_amd64.deb |

Start/stop the service

|  |
| --- |
| $ sudo service kapacitor *start/stop* |

# Grafana

Open APT Repositry

|  |
| --- |
| $ sudo nano /etc/apt/sources.list |

Add the folowing at the bottom

|  |
| --- |
| deb https://packagecloud.io/grafana/stable/debian/ wheezy main |

Add the Cloud key

|  |
| --- |
| curl https://packagecloud.io/gpg.key | sudo apt-key add - |

Update and installation

|  |
| --- |
| $ sudo apt-get update  $ sudo apt-get install grafana |

# Rabbitmq

Install APT repositories

|  |
| --- |
| $ echo 'deb http://www.rabbitmq.com/debian/ testing main' |  sudo tee /etc/apt/sources.list.d/rabbitmq.list |

Add public key for unsigned packages

|  |
| --- |
| $ wget -O- https://www.rabbitmq.com/rabbitmq-release-signing-key.asc |  sudo apt-key add - |

Install

|  |
| --- |
| $ sudo apt-get update  $ sudo apt-get install rabbitmq-server |

Start the service:

|  |
| --- |
| $ sudo invoke-rc.d rabbitmq-server *stop/start/restart/etc*. |

Set up config file

|  |
| --- |
| $ sudo gunzip /usr/share/doc/rabbitmq-server/rabbitmq.config.example.gz  $ sudo cp /usr/share/doc/rabbitmq-server/rabbitmq.config.example /etc/rabbitmq/  $ sudo mv /etc/rabbitmq/rabbitmq.config.example /etc/rabbitmq/rabbitmq.config |

Open the config file and uncomment the following line, remember to remove the comma after the curly brackets if it is the only line:

|  |
| --- |
| $ sudo nano /etc/rabbitmq/rabbitmq.config  {loopback\_users, []} |

Restart the service

Install management plug-ins

|  |
| --- |
| $ sudo rabbitmq-plugins enable rabbitmq\_management |

Management URL:

|  |
| --- |
| http://[IP Address]:15672/ |

User: guest

Password: guest

# Git

$ apt-get install git

$ git config --global user.name "YOUR NAME"

$ git config --global user.email "YOUR EMAIL ADDRESS"

# Nodejs

Install:

|  |
| --- |
| $ curl -sL https://deb.nodesource.com/setup\_4.x | sudo -E bash -  $ sudo apt-get install -y nodejs  $ sudo apt-get install -y build-essential |

Fix npm permissions:

|  |
| --- |
| $ mkdir ~/.npm-global  $ npm config set prefix '~/.npm-global' |

Open profile file:

$ nano ~/.pofile

Add to the file:

export PATH=~/.npm-global/bin:$PATH

Test:

$ npm install -g jshint

## NPM Packages

Install the following packages from NPM:

$ nmp install *package name (e.g. $ npm install winston)*

* *winston*
* *collections*
* *coap*
* *async*
* *coap-packet*
* *restify*

Packages:

# Disona

Install:

$ git clone <https://github.com/ksstech/DiSoNa.git>

To start and stop:

$ forever start -m 5 DiSoNa/server.js

$ forever stop -m 5 DiSoNa/server.js

# Slimjan

Install:

$ git clone <https://github.com/ksstech/slimjan.git>

To start and stop:

$ forever start -m 5 slimjan/server.js

$ forever stop -m 5 slimjan/server.js

# Fluxator

Install:

$ git clone <https://github.com/ksstech/fluxator.git>

To start and stop:

$ forever start -m 5 fluxator/server.js

$ forever stop -m 5 fluxator/server.js

**Actions to get running:**

In Influxdb create database “sense”

**To start all services**

cd ~

./start-services.sh

**To backup the system:**

cd /

sudo tar -cvpzf /media/onegig/backups/yymmdd.tar.gz –exclude/media/onegig –one-file-system /

**Background:**

What I am trying to achieve is to make the basic data visualization work as it was. So, to this end I am trying to understand:

* what has been configured and working?
* Where is the process currently stopping?
* What is the minimum, easiest and most effective way to get the process working again?
* What has to be configured and where?

The modules I have identified are as follows:

* Disona / node.js / used to get coap to amqp comms going??
* slimjan / node.js / used to do the rules & sense lookup based on d\_hw and d\_id
* fluxator / node.js / takes data coming over coap via disona through RabbitMQ, strips the fluff, breaks it up and writes it to InfluxDb
* RabbitMQ, ESB, single connection point for DiSoNa, SlimJan, Fluxator and InfluxDB
* InfluxDb, obviously the tsdb
* Kapacitor, installed but used?
* Chronograf, installed but used?
* Telegraf, not installed. reasonably equivalent to Grafana?
* Grafana, primary graphing tool, installed but configured/connected to InfluxDB?

**Questions:**

Q1. What is the flow of data and modules involved in the ping/register phase?

Q2: What is the flow of data and modules involved in the sense and rules phase(s)?

Q3: What is the flow of data and modules involved for sense data from mote to getting it into InfluxDB?

Q4: Do we actually use components like Kapacitor, Chronograf, and if so how/where?

Q5: What is the flow of data, and modules involved in getting data from InfluxDb through to representing graphs in Grafana? Is Kapacitor involved and if so how?

Q6: are there any config files anywhere that can be reused to speed up the process?