|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Reference Tool | Operating System | Platform | Database | Documentation | Remarks |
| Zato | Ubuntu | Python 2.7 | SQlite | [Documentation](#zato_documentation) |  |
| OpenFn | Windows/Linux | NodeJS | N/A | [Documentation](#openfn_documentation) |  |
| OpenHIM |  | NodeJS |  | [Documentation](#openhim_documentation) |  |

# **Documentation for Implementers**

# **Zato Documentation**

## **Requirements**

* Ubuntu 16.04 LTS, 18.04 LTS (64-bit)
* Sudo rights

## **Installation steps**

* Install helper programs

sudo apt-get install apt-transport-https curl

sudo apt-get install software-properties-common

* Install an additional package on Ubuntu 16.04 LTS

sudo apt-get install python-software-properties

* Additional dependencies on Ubuntu 18.04 LTS

sudo add-apt-repository universe

sudo apt-get install tzdata

* Add the package signing key

curl -s https://zato.io/repo/zato-3.1-C9B13DF28CFE287D.pgp.txt | sudo apt-key add -

* Add Zato repository

sudo add-apt-repository \

"deb [arch=amd64] https://zato.io/repo/stable/3.1/py27/ubuntu $(lsb\_release -cs) main"

sudo apt-get update

* Install Zato

sudo apt-get install zato

* Install latest additions

sudo su - zato

cd /opt/zato/current

./update.sh

* Confirm the installation

ubuntu$ zato --version

Zato 3.1+rev.nnnnnnn-py2.7.n

ubuntu$

## **Creating a quick-start cluster**

Prepare as a zato user:

* An empty directory ($path), such as /opt/zato/zato\_cluster1
* Host, port and a password for connecting to Redis ($kvdb\_host, $kvdb\_port and $kvdb\_password)

Run the command below:

$ zato quickstart create $path sqlite $kvdb\_host $kvdb\_port \

--kvdb\_password $kvdb\_password --verbose

For example, to create a cluster with sqlite database:

$ zato quickstart create /opt/zato/zato\_cluster1 sqlite localhost 6379 \

--kvdb\_password '' --verbose

Quickstart cluster quickstart-962637 created

Web admin user: [admin], password: [Fc8-43IRUVeyWVgKtUR8YqXJSQriJt6x]

Note the highlighted line. This provides the zato web-admin username and password.

The above command will create the following:

* A development CA
* 2 servers
* An HA load-balancer in front of the two
* A web-admin panel instance and an admin user
* ODB structure in SQLite and Redis
* Scripts to (re-)start the environment and stop it

The password for web-admin is randomly generated which can be changed with the following command:

$ zato update password path username

An example in above quick start cluster:

$ zato update password /opt/zato\_cluster1/web-admin/ admin

Changing password for user 'admin'

Password:

Password (again):

The cluster can be started by issuing the command:

$ sudo su – zato

$ service zato start

$ /opt/zato/zato\_cluster1/zato-qs-start.sh

**Important TCP ports that will be used are:**

| **Port** | **Notes** |
| --- | --- |
| 11223 | Load-balancer’s HTTP port (this is what external applications use to invoke services you develop) |
| 17010 | server1’s HTTP port |
| 17011 | server2’s HTTP port |
| 8183 | Web admin’s HTTP port (this is where you point your browser to and log in as an admin user) |

References:

<https://zato.io/docs/admin/guide/install/py27/ubuntu.html>

<https://zato.io/docs/index.html>

<https://zato.io/docs/tutorial/01.html>

<https://zato.io/docs/tutorial/02.html>

------------------------------ \*\*\*\* -------------------------------

# **OpenFn Documentation**

## **OpenFn Devtools**

A set of tools for writing & testing expressions, managing OpenFn projects, and developing new adaptors (language-packages).

## **Pre-Requisites**

Git

$ sudo apt-get install git

Node.js - Version 6.11 LTS

$ sudo apt-get install curl python-software-properties

$ curl -sL https://deb.nodesource.com/setup\_10.x | sudo -E bash –

$ sudo apt-get install nodejs

$ node –v

$ npm -v

## **Installation**

Clone openfn-devtools from git repository

$ git clone https://github.com/OpenFn/openfn-devtools.git

$ cd openfn-devtools

$ ./install.sh

If you get a "permission denied" message when running ./install.sh, run run chmod +x ./install.sh then retry the install command.

## **Install dependencies**

$ cd core

$ npm install

## **Install language-packages**

$ install.sh language-${name}

$ cd language-${name}

$ npm install

## **Usage**

Execute takes:

1. -l [language-package].Adaptor: The language-package.
2. -e [expression.js]: The expression being tested.
3. -s [state.json]: The message data: {...} and credential configuration: {...}.
4. -o [output.json]: The file to which the output will be written.

## **Bash usage**

path/core/lib/cli.js execute -l path/language-[XXX].Adaptor -s path/tmp/state.json -o path/tmp/output.json -e path/tmp/expression.js

## **Install openFn UI**

$ sudo apt-get install php7.0

$ git clone [repository to be added]

# 

## **Testing**

$ cd /path/of/application

$ php –S localhost:port

## **Production**

Any webserver with php support may be used.

# **OpenHIM Documentation**