How to Install and Configure your own Identity Manager GE

Álvaro Alonso – Federico Fernández
Security Team
Technical University of Madrid (UPM)
aalonso@dit.upm.es – fefernandez@dit.upm.es
Outline

- Introduction
- KeyRock Architecture
- Installing and Configuring KeyRock
- Demo
Why do I need an Identity Manager?
What is an Identity Manager?
Why should I install FIWARE Identity Manager GE?

KeyRock

SECURITY
KeyRock GE: features

- Users
- Organizations
- Authorization via roles
- Applications and OAuth
- IoT identity management
- OpenStack services
- Admin tools
- SCIM API
KeyRock Architecture
KeyRock Architecture

Horizon

Keystone

DB
KeyRock Architecture: Horizon

- Front-end view
- Based on OpenStack Horizon
- User views
- Contains…
  - Oauth2 Driver
  - reCAPTCHA
  - FIWARE Accounts
  - Admin Tools
  - AuthZForce Driver
- Extra dependencies
  - Python Keystoneclient
  - Django OpenStack Auth
KeyRock Architecture: Keystone

- Back-end component
- Resources management
- Connection to database
- Extensions
  - OAuth2
  - SCIM 2.0
  - User registration
  - Two factor authentication
KeyRock Architecture: Database

- For development:
  - SQLite

- For deployment:
  - Keystone
  - MySQL
  - PostgreSQL
#handsOn
Documentation & Source Code

- **Quick Installation Guide**

- **Detailed Installation Guide**

- **GitHub**
  - [https://github.com/ging/fiware-idm](https://github.com/ging/fiware-idm)
  - [https://github.com/ging/horizon](https://github.com/ging/horizon)
  - [https://github.com/ging/keystone](https://github.com/ging/keystone)

- **API description**
  - [http://docs.keyrock.apiary.io](http://docs.keyrock.apiary.io)
Installing KeyRock
Installing the **back-end**

1. Install Ubuntu dependencies
   1. 14.04 LTS fully supported
   2. 16.04 LTS should work
2. Get the code
3. Install Python dependencies
4. Create a configuration file

```bash
$ sudo apt-get install python python-dev python-virtualenv libxml2-dev libxslt1-dev libtasl2-dev libsqlite3-dev libssl-dev libldap2-dev libffi-dev
$ git clone https://github.com/ging/keystone && cd keystone
$ sudo python tools/install_venv.py
$ cp etc/keystone.conf.sample etc/keystone.conf
```
Installing the back-end

5. Create the tables and populate the database

Creation of the idm user account

$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=oauth2
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=roles
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=user_registration
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=two_factor_auth
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=endpoint_filter
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --populate
5. Create the tables and populate the database

```
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=oauth2
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=roles
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=user_registration
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=two_factor_auth
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --extension=endpoint_filter
$ sudo tools/with_venv.sh bin/keystone-manage -v db_sync --populate
```

Creation of the **idm** user account

6. That’s it!!

```
$ sudo tools/with_venv.sh bin/keystone-all -v
```
Installing the front-end

1. Install Ubuntu dependencies

$ sudo apt-get install python python-dev python-virtualenv libssl-dev libffi-dev libjpeg8-dev

2. Get the code

$ git clone https://github.com/ging/horizon && cd horizon

3. Create a configuration file

$ cp openstack_dashboard/local/local_settings.py.example openstack_dashboard/local/local_settings.py

4. Install Python dependencies

$ sudo python tools/install_venv.py
Installing the **front-end**

1. Install Ubuntu dependencies

   ```shell
   $ sudo apt-get install python python-dev python-virtualenv libssl-dev libffi-dev libjpeg8-dev
   ```

2. Get the code

   ```shell
   $ git clone https://github.com/ging/horizon && cd horizon
   ```

3. Create a configuration file

   ```shell
   $ cp openstack_dashboard/local/local_settings.py.example openstack_dashboard/local/local_settings.py
   ```

4. Install Python dependencies

   ```shell
   $ sudo python tools/install_venv.py
   ```

5. That’s it!

   ```shell
   $ sudo tools/with_venv.sh python manage.py runserver localhost:8000
   ```
Installing Keyrock

Good News

- Installation tools to ease the process

- Bash script
  - Idm user: `idm`
  - Idm psswd: `idm`
  - Keystone port: `5000`
  - Horizon port: `8000`

- Docker image

- Chef cookbook
Configuring KeyRock
Configuring the **back-end**

- Admin token
- Admin port
- Public port
- Configure authorization, roles…

```bash
#admin_token=ADMIN
#admin_port=35357
#public_port=5000
```

```
/etc/keystone.conf
```

```
/etc/policy.json
```
Configuring the front-end

- Credentials for idm user
- reCAPTCHA
- Account expiration
Configuring the **front-end**

- AJAX pagination
- Connection with Access Control GE

```python
# Table Pagination
PAGE_SIZE = 5

# access control GE
ACCESS_CONTROL_URL = 'http://azf_host:6019'
ACCESS_CONTROL_MAGIC_KEY = 'azf_pep_key'
```
Considerations for production environments

- **Do not** run Horizon from the dev server
- **Do not** run KeyRock without having enabled reCAPTCHA
- **Do not** use SQLite
- **Do not** forget about the emails!
- **Do not** run Keystone in dev mode
- **Do** run Horizon under Apache+mod_wsgi
- **Do** enable reCAPTCHA
- **Do** use some production-ready DB engine (MySQL)
- **Do** set up an SMTP server to send mails (POSTFIX)
- **Do** set up Keystone as a service
Production env: MySQL

- Configure the new SQL backend in Keystone
- Grant privileges to database

```bash
[database]

# The SQLAlchemy connection string used to connect to the database
connection =
    mysql://keystone:KEystone_DBPASS@MYSQL_ADDRESS/keystone

# mysql -u root -p
mysql> CREATE DATABASE keystone;
mysql> GRANT ALL PRIVILEGES ON keystone.*
    TO 'keystone'@'localhost' IDENTIFIED BY 'KEystone_DBPASS';
mysql> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'KEystone_DBPASS';
```
Production env: email

This will get the settings from the default SMTP server in your host.

```python
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'

# Configure these for your outgoing email host
EMAIL_HOST = 'smtp.my-company.com'
EMAIL_PORT = 25
EMAIL_HOST_USER = 'djangomail'
EMAIL_HOST_PASSWORD = 'top-secret'
EMAIL_URL = 'your-host.com'
DEFAULT_FROM_EMAIL = 'your-no-reply-address'
EMAIL_SUBJECT_PREFIX = '[Prefix for emails subject]'```
Production env: setting up Keystone as a service

- It works like any other Linux service

Create a `/etc/init/keystone_idm.conf` file

To run the service...

```
# keystone_idm - keystone_idm job file
description "Service conf file for the IdM backend based in Keystone"
start on (local-filesystems and net-device-up IFACE1=lo)
stop on runlevel [016]
# Automatically restart process if crashed
resrawn
setuid root
script
 cd $absolute_keystone_path
#activate the venv
 . .venv/bin/activate
#run keystone
 bin/keystone-all
end script

$ sudo service keystone_idm start
```
Production env: CORS

- Whitelist to restrict access to all the endpoints in the front-end
- Django signal to allow everyone access only some of the endpoints

```python
# CORS configuration
CORS_ALLOW_CREDENTIALS = True
CORS_ORIGIN_WHITELIST = (  
    'cloud.lab.fiware.org',  
    'store.lab.fiware.org',  
    'mashup.lab.fiware.org',  
    'data.lab.fiware.org',  
    'help.lab.fiware.org',
)

def cors_allow_api_to_everyone(sender, request, **kwargs):
    # return request.path.startswith('/api/')
    return False
check_request_enabled.connect(cors_allow_api_to_everyone)
```
Administrating KeyRock
Administrating KeyRock

$ git clone https://github.com/ging/fiware-idm
 imd-admin && cd imd-admin

$ sudo pip install -r requirements.txt

$ sudo python setup.py install

$ idm-admin --help
#handsOn
Achievements

✔ What is an IdM and why should I install one?
✔ What is the architecture of FIWARE IdM GE?
✔ Installing KeyRock
  • Step-by-step
  • Installation tools
✔ Configuring KeyRock
  • Development environment
  • Production environment
✔ Administrating KeyRock
Contact us!

Open an Issue in GitHub:
https://github.com/ging/fiware-idm

E-mail & Help Desk

Here at the Summit!!
Thank you!

http://fiware.org
Follow @FIWARE on Twitter