Fily A cloud based file sharing service

Project 17
Monday, December 7th, 2015

Mathieu Devaud (mathieu.devaud@master.hes-so.ch)
Frederic Montet (frederic.montet@master.hes-so.ch)

Project Context

This project is about using the power of the cloud to create a file sharing system on invitation with Amazon AWS S3. A main user with an AWS account can use his credentials to create URLs to give to secondary users. These URLs can be used by the secondary users to upload any file on the S3 of the main user. The main user can then monitor the links he created and download files uploaded by the secondary users.

Work done

All the system is web based with Django on the server side and and give to the users an elegant interface built in Javascript on the client side. Thanks to Amazon Elastic Beanstalk, we have been able to deploy our application from command line with very little configuration and it has been really easy to redeploy when we updated the application. Also, Amazon Elastic Beanstalk has allowed us to embrace the autoscalling capability of a cloud environment such as Amazon AWS.

From the main user perspective, an admin interface allows him to generate links via a simple button. The link is generated via the javascript SDK of Amazon S3 and create a temporary presigned URL containing the needed permissions to upload a file. When a link is created, it is stored in a local SQLite3 database. The main user can watch all the links and interact with them via buttons for deleting the links, download the file uploaded by the invited user and get the presigned url from an alert box.

From the invited user's perspective, he can navigate to the link received from the main user and get to a simple form to choose a file on his system to upload it to the S3 without Amazon account. The file is uploaded via a PUT method in an AJAX request to the pre-signed URL.

Using the app

It is easy to get started with our app, nevertheless, a few things are needed:

- An Amazon AWS Access Key ID
- An Amazon Secret Access Key
- A configured Amazon S3 Bucket

Configure your S3 Bucket

Configuring the bucket is about setting the right CORS policy to allow cross-origin uploads. The steps to do so are the following:

1. Create a new Bucket on Amazon S3
2. Go to Permissions in the Properties menu
3. Click on Edit CORS configuration
4. Add the following CORS Policy from figure 2
5. The configuration is finished

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration
xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <CORSRule>
        <AllowedOrigin>*</AllowedOrigin>
        <AllowedMethod>GET</AllowedMethod>
        <AllowedMethod>PUT</AllowedMethod>
        <AllowedMethod>POST</AllowedMethod>
        <AllowedMethod>DELETE</AllowedMethod>
        <AllowedHeader>*</AllowedHeader>
    </CORSRule>
</CORSConfiguration>
```

Figure 1 : S3 Bucket CORS Policy

Get started

The Fily file sharing service is available at: http://filyproject2-dev.elasticbeanstalk.com/
Once on the login page, insert your AWS Credentials as indicated in the placeholders and write the Bucket name you just created.

Once on the admin page, write the expiration time of your link and generate it.

Project structure

The project is separated in three main directories:

- **`filyApp` :** The client side code
- **`filyRest` :** The server side code
- **`filyProject` :** The Django project files

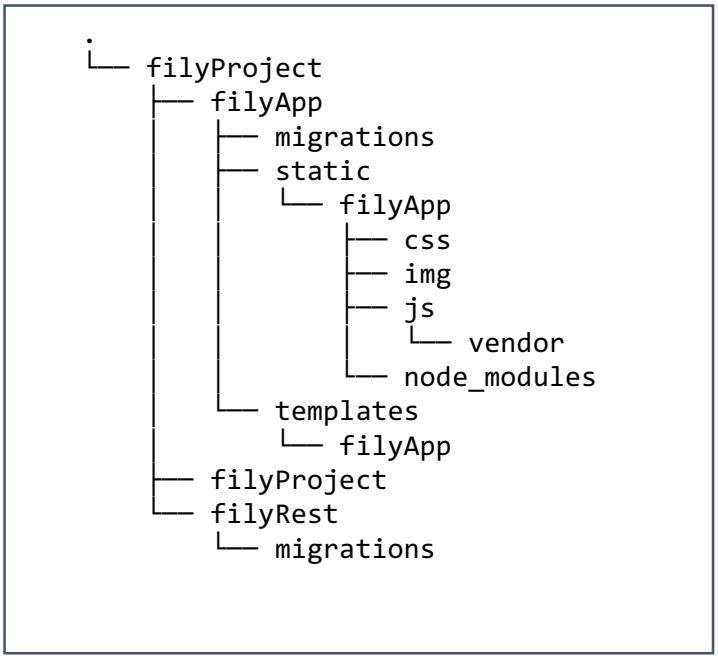

Figure *2* : FilyProject directory structure

`FilyApp`

The `filyApp` module contains all the client side code. This module has been mainly used for distributing the static pages. The Fily pages are being build with the different `.html` templates in the `filyApp/templates` directory and their dynamic is done with the `main.js` file that you can find in `filyApp/static/filyApp/js`.

When the server is launched, the Fily App has two pages:

`http://127.0.0.1:8000/`	The main page with the bucket list
`http://127.0.0.1:8000/bucket?uuid=<uuid>`	The upload page for invited user

FilyProject

The second module is `filyProject`, it contains all the Django project files. In this folder, we set the url patterns for FilyApp and FilyRest in `urls.py` and in `settings.py` you are able to find the main Django settings.

FilyRest

The last module is `filyRest`, it contains the server side code. As we used the Django Rest Framwork package, the structure of this Django package has been given by the aforementioned framework. The main files in this module are `models.py` where you can find the `Bucket` class that we used to store our informations in the SQLite3 database and the `views.py` which tells which `http` methods are callable for each views of the rest service.

The rest service is available at the following urls:

`http://127.0.0.1:8000/api/buckets/`	The list of all the buckets
`http://127.0.0.1:8000/api/buckets/<bucket_pk>/`	The detail of a single bucket

Run the application locally

As the user authentication is not taken into account for this project, we hardcoded the Amazon AWS Access key ID and AWS Secret key to authentify ourselves during the development. Therefore, for security reason we can't distribute a proper production version of our code on an online server.

Prior to any command, please put your Amazon AWS Access key ID and Secret key in `filyApp/static/filyApp/js/main.js` at line 7.

```
AWS.config.update({
      accessKeyId: '<myAccessKeyID>',
      secretAccessKey: '<mySecretAccess>'
      });
```

Fily installation

A few command are needed to run Fily locally:

Go to the parent directory of the given archive
```
cd ../filyProject
```

Install a virtualenv
```
virtualenv -p /usr/bin/python2.7 filyEnv
```

Activate the filyEnv
```
source filyEnv/bin/activate
```

Install the Python package on the filyEnv
```
pip install -r requirements.txt
```

Run the server
```
python manage.py runserver
```

Access the Fily application on `http://127.0.0.1:8000/`

Run the application on Amazon Elastic Beanstalk

On the command-line, go to the top-level of your project (`filyProject`) and create a new directory, called `.ebextensions`:

```
mkdir .ebextensions
```

Using a text editor, create a new file called 01-filyProject.config in the .ebextensions directory, and add the following lines to it:

```
option_settings:
  "aws:elasticbeanstalk:application:environment":
    DJANGO_SETTINGS_MODULE: "filyProject .settings"
    PYTHONPATH:
"/opt/python/current/app/filyProject:$PYTHONPATH"
  "aws:elasticbeanstalk:container:python":
    WSGIPath: "filyProject/wsgi.py"
```

From within your application's top-level directory (`filyProject`), deploy your application using the command:

```
eb create
```

The application is now launched at the URL you set in the interactive console.