User's Guide for Cloud Backup

Captain CyBeard: Neil Before Us

Ryan Breitenfeldt | Noah Farris

Trevor Surface | Kyle Thomas

May 4, 2020

Washington State University Tri-Cities

CptS 423 Software Design Project 2

Contents

List of Figures

Revision History

Revision	Date	Author(s)	Description
1.0	05.03.2020	KT	Document Completion
0.1	04.20.2020	KT	Document Creation

1   Introduction

This document is an instruction manual for users on how to use the *Cloud Backup* software. There will be three parts. The first part will be how user's of application will interact with the application with the second part covering how system administrators will deploy the application. Followed by the third part, which is how developers will expand the application to include more cloud platforms.

2   Users

The users of the application are people who are visiting the URL of the application to import their files into the resiliency platform.

2.1   Invocation

To begin using the application the user will open their web browser and navigate to the URL that contains the application. This will be provided by the administrators hosting the application.

2.2   Stopping

To terminate their interaction with the application users will simply close their browser tab or window containing the application or navigate to another site. This can be done during any step of using the application.

2.3   Selecting a Cloud Provider

On the main page of the application the user will be presented with a pull down menu containing each cloud platform that is supported by the application. Selecting one of these options and clicking the "Submit" button will proceed the user to the next step (authenticating to the cloud platform).

2.4   Authenticating to the Cloud Platform

After selecting a platform the user will redirected to an authentication page. Depending on the platform this page will be different and require different authentication methods. For example, the Amazon Web Services platform will direct the user to a page where they will

enter their AWS client secret and AWS client token into a text box. Other platforms such as Google will redirect the user to Google's authentication platform. From there they will enter their Google username and platform.

Some of these application during authentication will also require that the user grant permissions for the application to interact with their account. The application needs these permissions in order to function properly.

After the user authenticates to their account on the cloud platform they will be redirected back to the application for the next step (Selecting and Downloading Files).

2.5   Selecting and Downloading Files

After the user has authenticated to their cloud account they will be redirected to a screen where they will be able to view the files that are contained on this platform. On this screen, the user will see a list of their files, and depending on the platform their directories. Each file will have a selection box next to it to mark for importing. There is also a "select all" button for the user to select all of their files. Once the user has selected the files they wish to import into the resiliency platform they can select the "Download" button to initiate the import.

Once the "Download" button is selected the user will be redirected to the last screen of the application where they will see a list of the files imported for confirmation.

Both of the screens mentioned in this section also contain a "Start Over" button that will allow the user to return to the main screen where they select a platform. This makes it easier to quickly visit all of their platforms to import their files.

3   Administrators

The administrators or operations team for the application has a few things that need consideration but is otherwise straight forward.

3.1   Starting and Stopping the Application

The number of ways to serve a Python-Django application is almost infinite so please refer to the Django documentation and the documentation of your web server on how to deploy the application.

Whichever deployment method is choses, there are several settings that need to be changed in *project_root/cloud_backup/settings.py*.

The two settings that must be changed is the **DEBUG** value must be set to **False**. This will limit the information sent to potential attackers.

The second setting that must be absolutely changed is the **SECRET_KEY**. The default key shipped is publicly available and should not be used. Please generate a random 50 character string to replace it with.

Other settings in that file that could be changed is the **DATABASES** driver section for your database (if you have one, if not the default SQLite is adequate). As well as the **LANGUAGE_CODE** and **TIME_ZONE** settings.

For the built-in development server one can simply issue the command:

```
$ python manage.py runserver 0:80
```

Note that this is not recommended in a production environment.

3.2   Special Requirements

Some special requirements for deploying the application is some platforms will need this application submitted for verification to become a trusted application. The application will work if it is unverified but users will get several warning screens to click through that may turn them away from the application.

The second special requirement is that the application should be served over HTTPS with a trusted TLS certificate. This will result in less warnings by the platform. Obtaining a certificate and using HTTPS is beyond the scope of this document. Administrators should refer to their web server documentation if they need more information.

4   Developers

Software developers that would like to add additional cloud platforms to the application will need to update several files in addition to the platform module. The modular design and implementation of the application makes this fairly straightforward and quick.

Once a cloud platform has been implemented following the template from the `design document` that module should be added into the *project_root/cloud_download/platforms* directory. Then inside

that directory is a _init_.py file where the module will need to be imported.

The next step is adding the platform to the main screen. In the *project_root/cloud_download/views.py* the *Index* class will need to be modified. There is list variable called **platforms** that will need the name of the platform added (to be seen by the user) and also the **post** method will need to have the platform added with its appropriate redirect path for authentication.

How to integrate authentication will vary depending on the cloud platform. The *project_root/cloud_download/platforms/dr* contains a good example of using an **Oauth2** derived authentication. The *project_root/cloud_download/platforms/aws.py* contains a good example of using a custom page for prompting the user for their credentials. In addition, the *project_root/cloud_download/urls.py* will need to be modified with the correct URL redirect paths. The files are well documented for how to add additional platforms.

The final step in adding a cloud platform is modifying the **Files** class in the *project_root/cloud_download/views.py* file. There are already several examples of cloud platforms and adding additional ones should be straight forward.

The design of the application was focused on making it as trivial as possible to add additional platforms but that is not a guarantee. Special variables may need to be added, extra pages, views or other unforeseen modifications. But special care was taken to make it as easy as possible. The html templates for example, should not need to be modified to add additional platforms.