

precase_week4

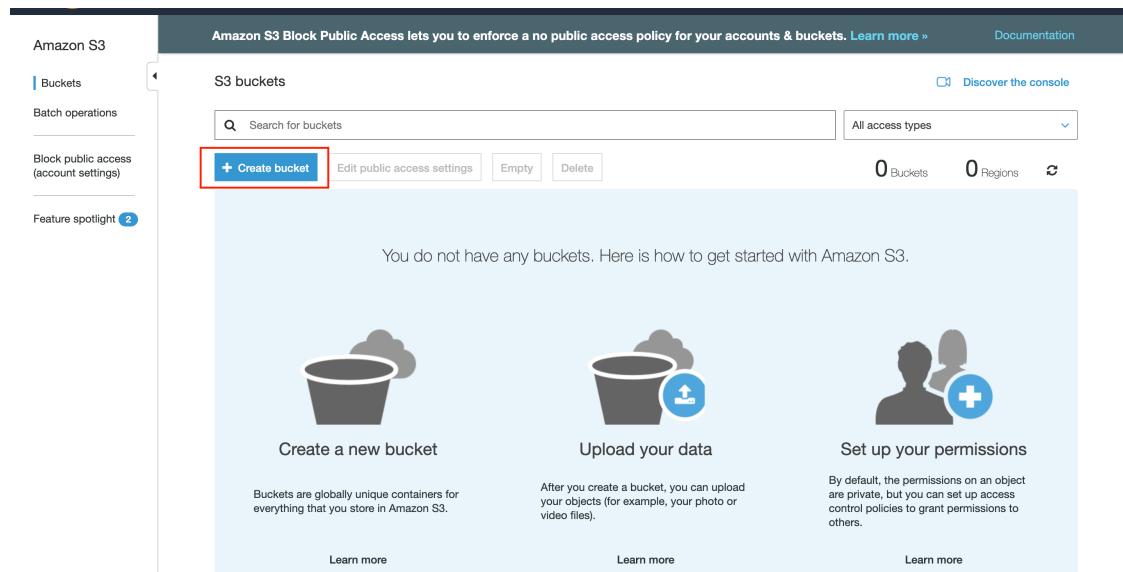
May 13, 2020

0.1 Uploading Files to AWS S3 and setting up API Keys

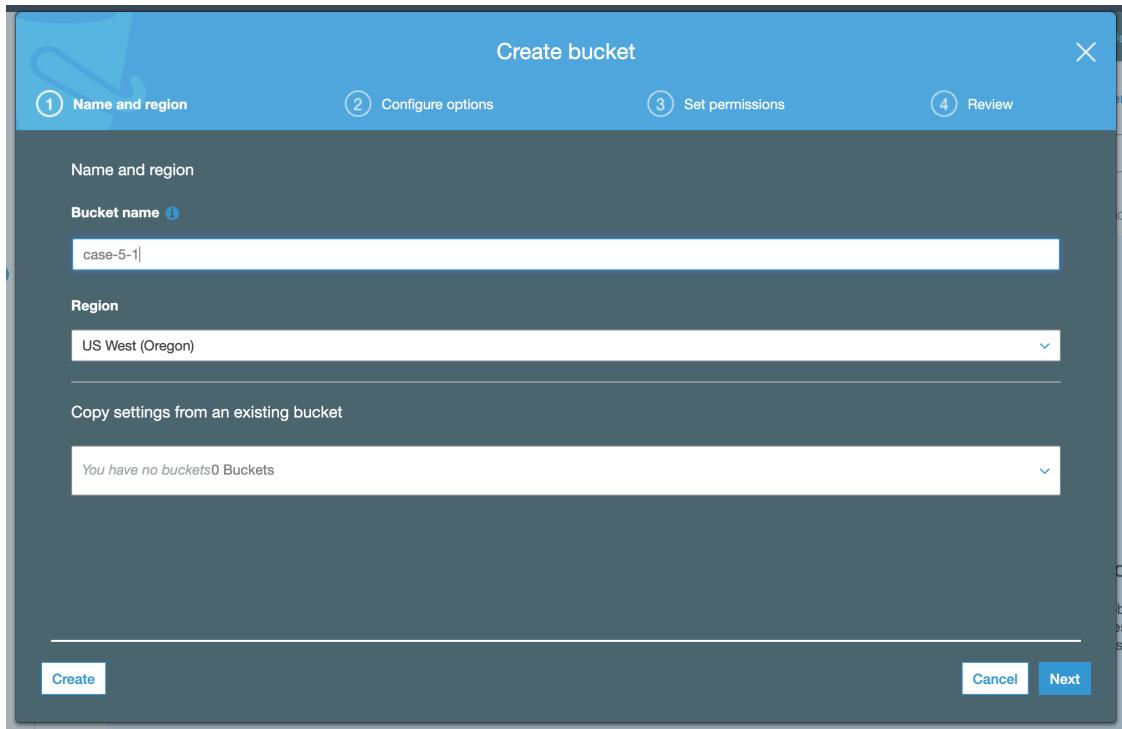
In this section, our focus is to upload the data available in the zip to AWS S3, and then generate the API Keys (Access Key ID and Secret Access Key) to be used in the case 5.1

0.2 Create S3 Bucket

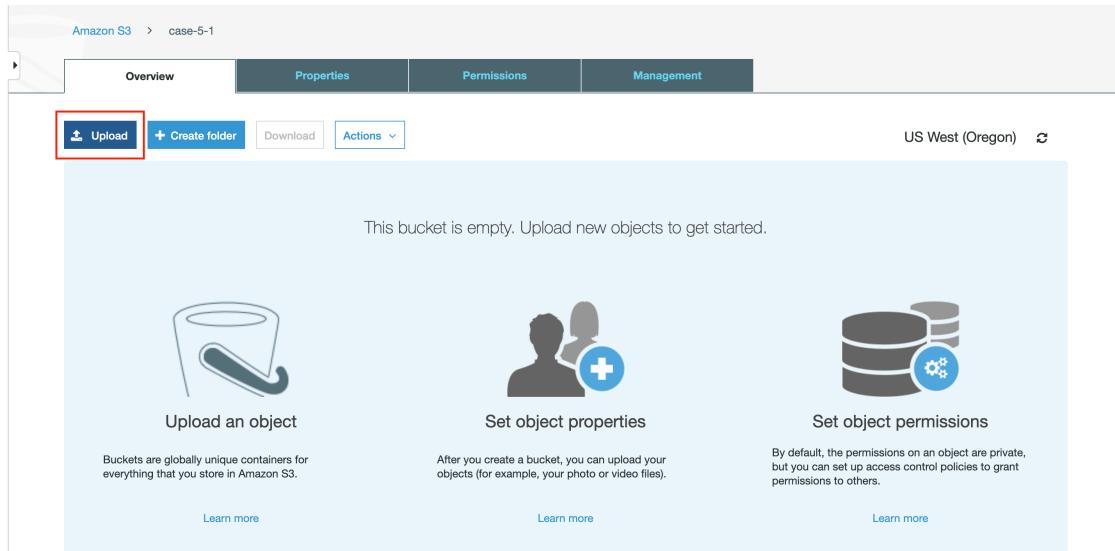
Login to your AWS portal and then to S3. In order to upload the file, you will need to create a bucket to hold the data files. In your S3 portal, click on the **Create Bucket** button.



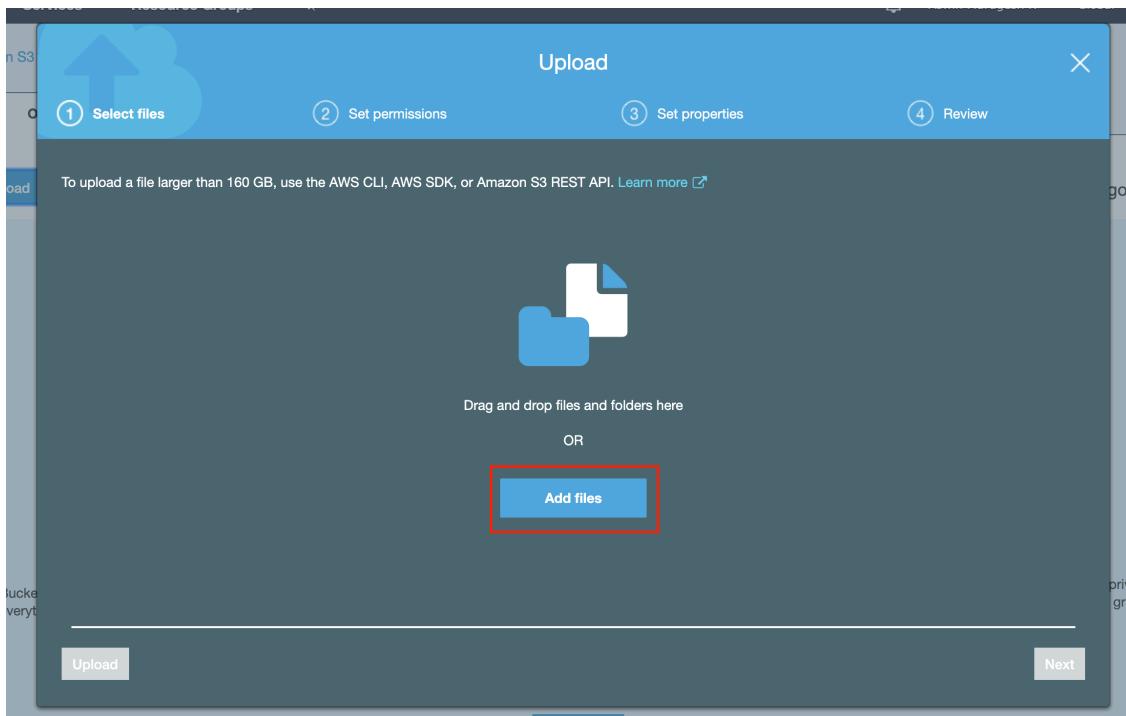
Once the Create bucket form opens up, Provide a name for your bucket and click on the create button.



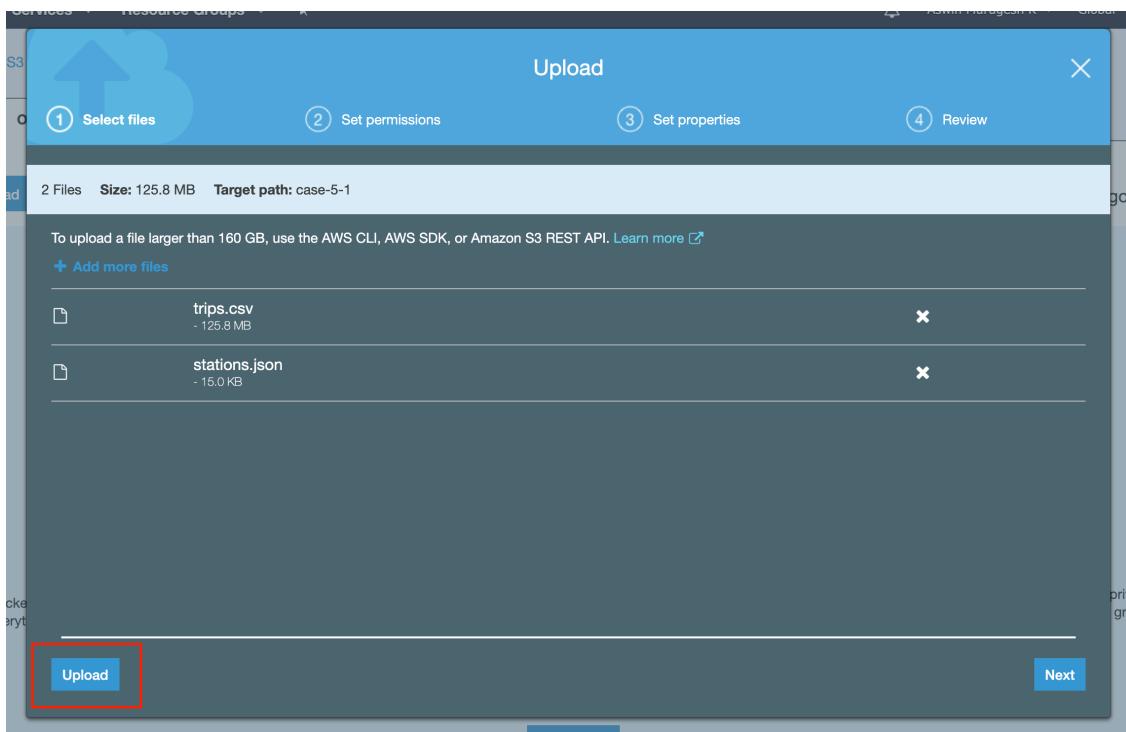
Once the bucket is created, click on the bucket name to go to the bucket detail page. Once in the detail page, click on Upload button present in the page.



When the add form shows up, click on the Add Files button



When the dialog box opens up, select the `trips.csv` and `stations.json` file provided to you. You will then see the below page

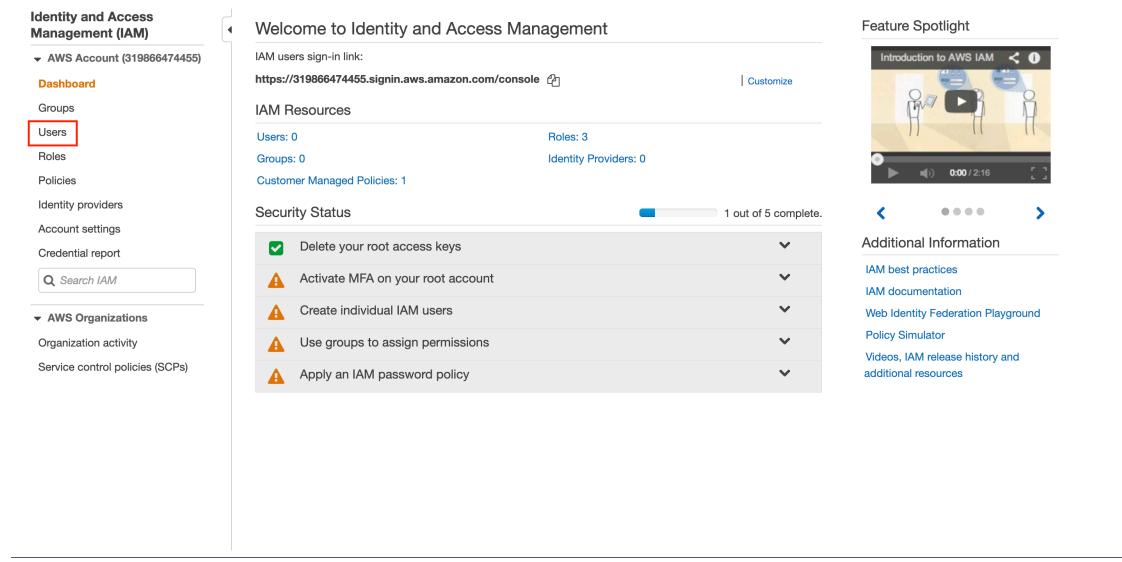


Once you click on the **Upload** button, you can see the **Operation in Progress** button at the bottom of the screen. The data will be uploaded and in a while you will see the two files in the list

of files in the bucket, as below:

0.3 Setup Security Credentials in IAM

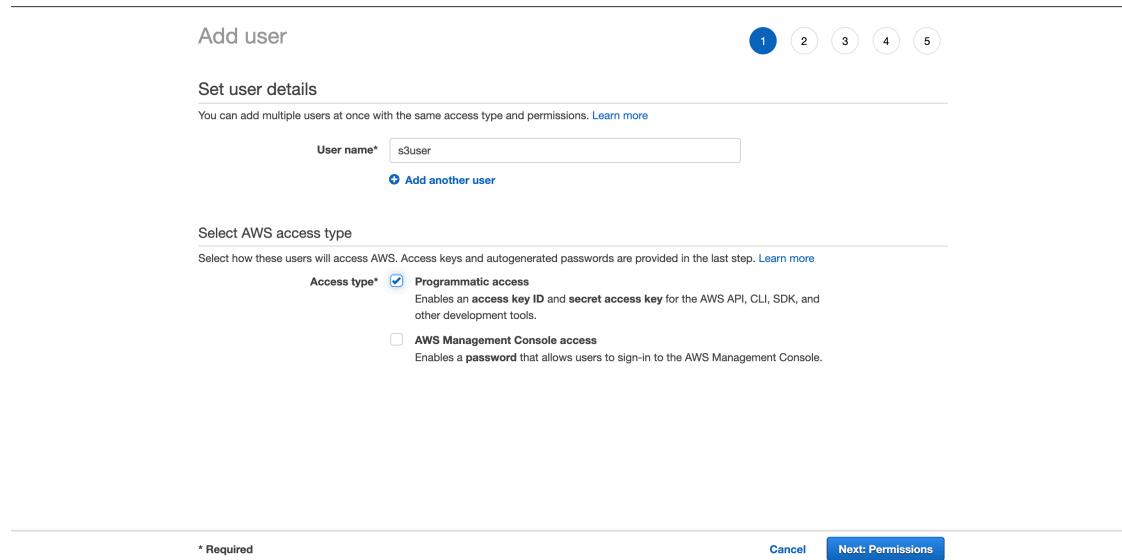
Once the files are uploaded, we will need a way to access the uploaded file programmatically. In order to access the files via any programmatic manner (e.g. Python program), you will need to create Security credentials. For that, you will need to go to the IAM service of AWS. Go to **Services** -> **IAM** to see the below page



The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left, there's a sidebar with options like AWS Account, Groups, Users (which is selected and highlighted with a red box), Roles, Policies, Identity providers, Account settings, and Credential report. Below the sidebar is a search bar labeled 'Search IAM'. The main content area has a title 'Welcome to Identity and Access Management' and a sub-section 'IAM Resources' showing 'Users: 0', 'Groups: 0', 'Roles: 3', and 'Identity Providers: 0'. To the right of the resources is a 'Feature Spotlight' section with a video thumbnail titled 'Introduction to AWS IAM' and a progress bar showing '0:00 / 2:16'. Below the spotlight is an 'Additional Information' section with links to 'IAM best practices', 'IAM documentation', 'Web Identity Federation Playground', 'Policy Simulator', and 'Videos, IAM release history and additional resources'.

Click on the **Users** section to see the list of users in your account. Click on **Add Users** button at the top of the page to create a new user.

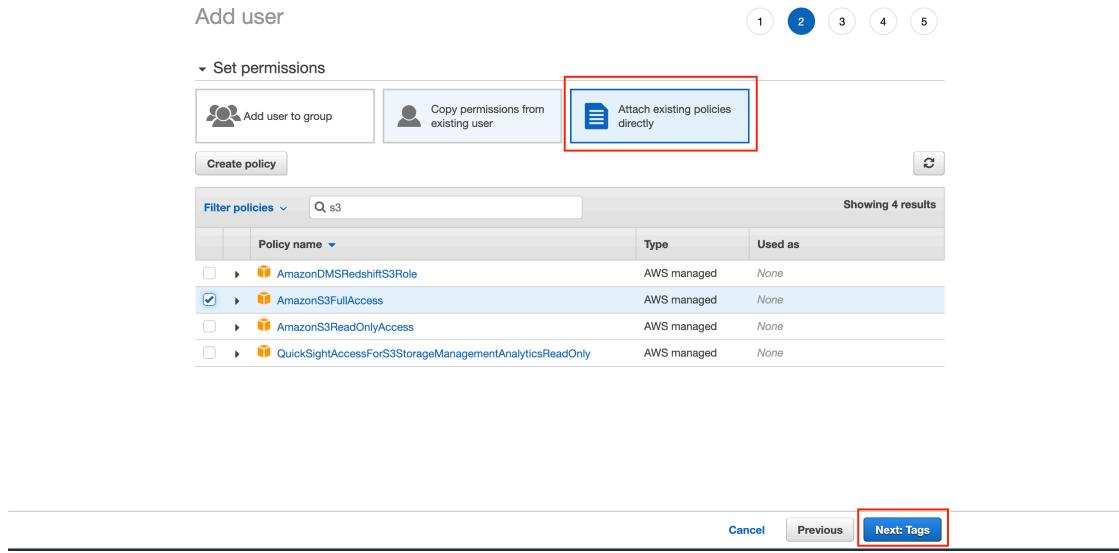
In the create user page, provide a username. Also, enable only programmatic access for the user, as we don't want the new user to login to the AWS portal.



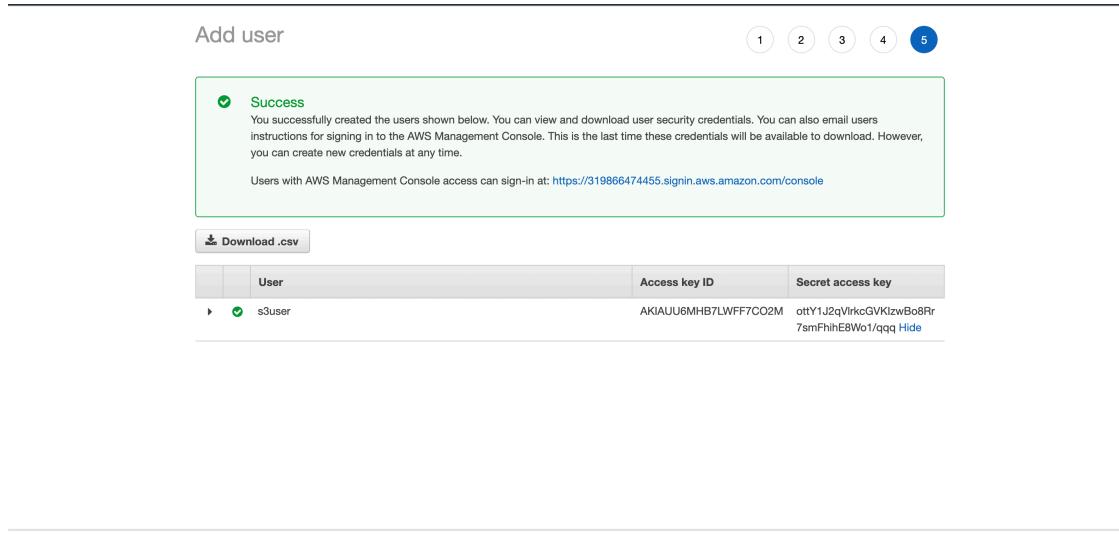
The screenshot shows the 'Add user' wizard, step 1: Set user details. At the top, there's a navigation bar with steps 1 through 5. Step 1 is highlighted with a blue circle. The main form has a title 'Set user details' and a note: 'You can add multiple users at once with the same access type and permissions.' Below this is a 'User name*' field containing 's3user' and a link to 'Add another user'. The next section is 'Select AWS access type' with a note: 'Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step.' It shows two options: 'Programmatic access' (selected with a checked checkbox) and 'AWS Management Console access' (unchecked). The 'Programmatic access' option is described as enabling an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools. The 'AWS Management Console access' option is described as enabling a password that allows users to sign-in to the AWS Management Console. At the bottom of the form are buttons for '* Required', 'Cancel', and 'Next: Permissions'.

Click on **Next:Permissions** to go to the next step.

Since the user is going to be reading and writing files via S3, let's give the user full access to S3. In the permissions page, Select **Attach existing policies directly**, and search for S3. You should see a policy named **AmazonS3FullAccess**. Select that option to allow the user complete access to S3. Once selected, click on **Next:Tags** to go to the next section.



Keep clicking on next and at the end you will see a **Create User**. Click on that to create the user. Once the user is created, AWS will display your Access Key and Secret on the page. Please copy them and make a note of them, as you will not be able to see them again.



0.4 Setup Mapbox

Mapbox is a service that allows you to add interactive maps in your website. You can sign up to Mapbox in this link - <https://account.mapbox.com/auth/signup/>. Once you signup and login, you

should see the dashboard page as seen below. Click on the **Tokens** link on the top of the page

The screenshot shows the Mapbox Account Dashboard. At the top, there's a navigation bar with links for Dashboard, **Tokens** (which is highlighted with a red box), Statistics, Invoices, and Settings. Below the navigation, a welcome message "Welcome, aswinmurugesh!" is displayed. A central call-to-action button says "Start by designing a map >" with an illustration of a map and a building. To the right of this, there's information about Mapbox plans ("Pay-as-you-go") and current billing cycle usage ("No usage information to display"). Below this, there's a section titled "Tools & resources" with links to "Integrate Mapbox", "Design in Mapbox Studio", and "Documentation". On the left, there's a sidebar with a "Access tokens" section containing a "Create a token" button. The URL https://account.mapbox.com is visible at the bottom of the sidebar.

In the tokens list, you will see an existing token with the name `Default public token`; copy that token and add it to your code, to authenticate yourself.

The screenshot shows the "Access tokens" list on the Mapbox Account Dashboard. The tokens table has columns for Name, Token, Last modified, and URLs. One row is shown for "Default public token", which was created 7 minutes ago. The "Token" column contains a redacted string starting with "tUgBg". There is a "Create a token" button at the top right of the table area. The URL https://account.mapbox.com is visible at the bottom of the sidebar.

Name	Token	Last modified	URLs
Default public token	tUgBg	7 minutes ago	N/A

0.5 Setting up Jupyter notebook in EC2

First, create a new EC2 instance (as taught in Week 3).

In the next couple of cases, you will be executing data wrangling operations in this newly created EC2 instance, using jupyter notebook. Let's setup the Jupyter notebook and dependencies for the case in this section.

1. Create virtualenv

```

sudo -H pip3 install --upgrade pip
sudo -H pip3 install virtualenv
virtualenv jupyter_env

2. Activate virtualenv

source jupyter_env/bin/activate

3. Install Jupyter inside the environment

pip install jupyter pandas numpy geopy boto3 shapely scikit-learn==0.21.3

4. Configure notebook to accept remote connections

jupyter notebook --generate-config

```

With the above step, there will be a configuration file generated in the `.jupyter` folder in your home folder. We need to update the config to allow remote connections. Next:

1. Add the below line at the end of the config file

```
c.NotebookApp.ip = '*'
```

2. Set a password for this Jupyter instance

```
jupyter notebook password
```

3. Start Jupyter

```
jupyter notebook --no-browser --port=8889
```

You can now access the remote instance via the URL `server_ip:8889`

0.6 Conclusion

1. We have now successfully uploaded the data files to S3 for case 5.1, and created an access key and secret that can be used to access the files from your Python code.
2. We have created and got access to a Mapbox account and an API key for the account that will be used in case 4.1 and 4.2
3. We have setup a new EC2 instance and configured a virtualenv so that we can do data wrangling operations on the full datasets in the cloud, in case 5.1 and 5.2