**Uploading and Downloading Files in S3 with Ruby**

To date, the latest Ruby AWS SDK is version 3. In this version, each resources has its own module while the version 2 had the one with everything, aws-sdk. To interact with S3 with the v3 SDK, let’s use the aws-sdk-s3 module.

Ruby has its own way to upload to S3. First of all, you need to create the s3 object and then call methods on that object. With Node.js, we first read the file and convert to the binary format and upload the data into the object (see [here](https://www.mydatahack.com/uploading-and-downloading-files-in-s3-with-node-js)). With Python, we can simply pass the bucket name, key, and local file path in the upload function on the S3 object (see [here](https://www.mydatahack.com/moving-data-in-s3-and-redshift-with-python/)).

Downloading is the same. We first need to create the S3 object that you want to retrieve and call the get() method.

The authentication is handled by configuring AWS CLI on your local machine or server. To configure AWS CLI, you can check the link [here](https://docs.aws.amazon.com/lambda/latest/dg/setup-awscli.html).

First of all, you need to install the module.

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

gem install aws-sdk-s3

[/cc]

**Uploading File**

The function takes 3 arguments, bucket name, prefix and local file path. From the File object, you can extract the file name in the path with basename(). Once the target S3 object is created, you can all the upload\_file() function.

<pre class=" toolbar:2 lang:ruby decode:true" >

</pre>

**Downloading File**

Downloading takes the same flow as uploading. You first need to create the object you want to retrieve with bucket name and object key. Then, call the get() method.

<pre class=" toolbar:2 lang:ruby decode:true" >

</pre>