How do I use AWS SDK for JavaScript V3 with Wasabi?

AWS SDK for JavaScript V3 at https://github.com/aws/aws-sdk-js-v3 is live from AWS and offers some significant improvement from V2 however the setup and code is a bit obscure to get working for something like Wasabi. Here is a solution to get a setup and working code using a "trailblazing through it" approach as a user with the following experimental setup and am happy to report it is running. Will be attempting to use in own application using Wasabi. Happy if this helps speed the official certification process via Wasabi.

You can create some sample running code by doing the following. These instructions are for people familiar with JavaScript and npm, but not necessarily that familiar with Wasabi or S3

1) install SDK for Javascript V3 via npm

$ **npm init** (only if your directory has not initialized npm already)
$ **npm install @aws-sdk/client-s3**
$ **npm install @aws-sdk/s3-request-presigner** (if using this, my code does)

2) Make sure you have created a Wasabi account which will have your keys in them, you will need them for the config file.

3) create a config.json file with your keys in them as follows:

$ more **config.json**

```
{
   "accessKeyId": "PasteYourWasabiAccessKeyHere",
   "secretAccessKey": "PasteYourWasabiSecretAccessKeyHere"
}
```

4) create an example file to upload:

$ more upload_me.txt

This is a file we want to upload, hopefully it will be in wasabi without any problems!

5) go get the example code an run it (note if your setup can use import instead of require, great, mine is tied to a node server that uses require) at https://github.com/annabelle

6) Make a unique for your bucket name in the code. If your bucket does not have a unique name it will not be created. Modify this line with your bucket name:

```
// Set the bucket parameters. Make sure to use your own unique bucket name or else the bucket will not be created in your account
const bucketParams = { Bucket: "exampleProject-UseYourUNIQUEBucketNameHere" };
```

7) run the code examples by uncommenting the various functionality, here are the functions you can selectively uncomment:

```
makeBucket(bucketParams);
//listBuckets();
//putObject(uploadStreamingObjectParams);
//putObject(uploadInternalObjectParams); //change to inline
//listObjectsCommand(bucketParams);
//getObjectCommand(getObjectParams);
//deleteObjectCommand(getObjectParams);
//getSignedURL(getObjectParams);
//if want to do an upload and directly after get the signed params then need to make sure signing is done after object is put
//putObject(uploadInternalObjectParams).then(() => getSignedURL(getObjectParams));
```

for example: the uncommented function makeBucket(bucketParams); creates a bucket, run and check it in your wasabi account. Running it should look something like this :

$ node wasabi_S3_JDK_V3_examples.js

```
$ node wasabi_S3_JDK_V3_examples.js
Success {
  '$metadata': {
    httpStatusCode: 200,
    requestId: undefined,
    extendedRequestId:
'59G3i9OgmFGjeD21lmDMt7Y7msrQjvR++mUoGfmN+qN8MmzDyfcP20oeF45RTq89pcrARmZEutlS',
    cfId: undefined,
    attempts: 1,
    totalRetryDelay: 0
  },
  Location:
}
```

8) Here's an example where only listBuckets; is uncommented. This shows the buckets you have created (you can create buckets via this demo code or via the wasabi console)

$ node wasabi_S3_JDK_V3_examples.js

```
$ node wasabi_S3_JDK_V3_examples.js
Success {
  '$metadata': {
    httpStatusCode: 200,
    requestId: undefined,
    extendedRequestId:
'OCq2xF4FlqSuptqJR6DEVHzvlFbTD9KNvebjTF+1T8bP2x6gUCLvwoCXpICYxz35TAKPztNohx3z',
    cfId: undefined,
    attempts: 1,
    totalRetryDelay: 0
  },
  Buckets: [

    {
      Name: 'example-ihasabucket',
      CreationDate: 2021-09-07T22:13:02.000Z
    }
  ],
  Owner: {
    DisplayName: 'yourdisplayname',
    ID: ''
  }
}
```