

EXL Hackathon - Cloud Agnostic Solution

By Team Trinity

API Documentation

We have incorporated the following Cloud Providers in our solution:

1. **AWS (Amazon Web Services):** Millions of customers use AWS storage services to transform their business, increase agility, reduce costs, and accelerate innovation. Choose from a broad portfolio of storage solutions with deep functionality for storing, accessing, protecting, and analyzing your data.



2. **Microsoft Azure:** The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers highly available, massively scalable, durable, and secure storage for a variety of data objects in the cloud.



3. **Alibaba Cloud:** Alibaba Cloud Object Storage Service (OSS) provides industry-leading scalability, durability and performance. Customers of all sizes and industries can use it to store and protect any amount of data for use cases, such as backup and restore, content distribution, data lakes, websites, mobile applications, data archive and IoT devices.



Appropriate handling of sensitive keys and information for all Cloud Platforms:

1. To setup **AWS** credentials, run the following commands:
 - a. Firstly, install AWS CLI.
 - i. `curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"`
 - ii. `sudo installer -pkg AWSCLIV2.pkg -target /`
 - b. Run the following command for AWS configuration
 - i. `aws configure`
 - c. Enter your AWS credentials and make sure it has access to AWS S3.
2. To setup **AZURE** credentials, run the following commands:
 - a. Setup environment variables
 - i. `export azure_account_name=<account-name>`
 - ii. `export azure_account_key=<account-key>`
3. To setup **ALIBABA** credentials, run the following commands:
 - a. Setup environment variables
 - i. `export alibaba_access_key_id=<access-key-id>`
 - ii. `export alibaba_access_key_secret=<access-key-secret>`

Functionality of our Cloud-agnostic solution:

1. **Upload Feature:** The authorized user can upload files (such as jpg, pdf, doc, ppt etc) to any desired cloud platform.
2. **Download Feature:** The authorized user can download any file from the cloud platforms directly by providing the filename.
3. **Download Temporary Feature:** The unauthorized user will receive a temporary link which will be valid to download a particular file for a given amount of time. Also, for security purposes we have created short URL using python library "pyshorteners" because the link generated by the cloud SDKs were displaying credentials inside the link.
4. **List the Files Feature:** To make it easy for the authorized user to know which files are available in the cloud storages, this feature will allow authorized users to view all the files in the selected cloud platform.
5. **Delete Feature:** This is again an added functionality of our solution which enhances the user experience by allowing an authorized user to delete a file from the selected cloud storage.

Steps to execute the code:

1. Make sure your working directory is same as the code directory and Python 3 is installed in the system.
2. To install all the packages and dependencies required, run the following command:
 - a. `"pip install -r requirements.txt"`
3. Now to start the application, run the following command:
 - a. `"python3 app.py"`

Detailed Explanation of the API with screenshots:

We have created this Restful API in python using Flask. We have a single endpoint capable of handling all the use cases of the user for all the cloud providers. (**Postman collection is added in the GitHub Repo as well as the Code Zip Folder**)

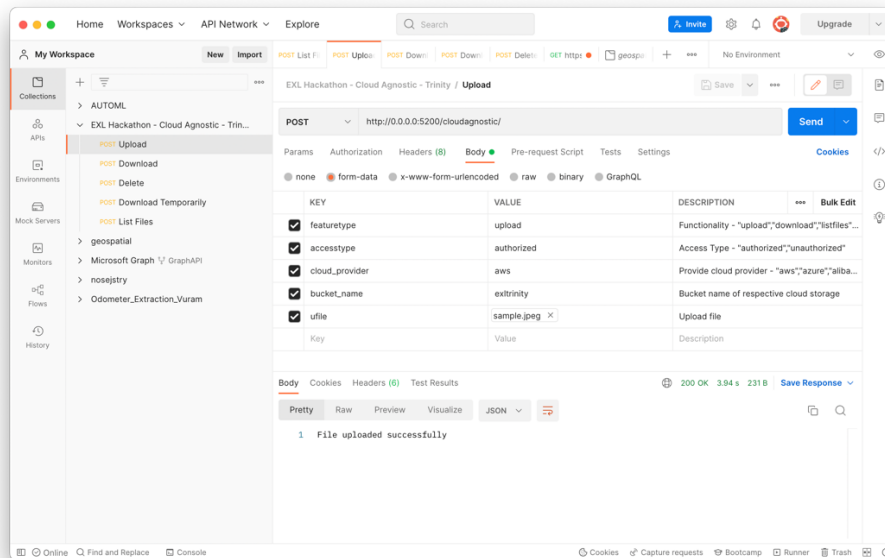
Cloudagnostic Endpoint - POST <http://0.0.0.0:5200/cloudagnostic/>

All key parameters and their value description:

1. featuretype: Functionality - "upload","download","listfiles","delete"
2. accesstype: Access Type - "authorized","unauthorized"
3. cloud_provider: Provide cloud provider - "aws","azure","alibaba"
4. bucket_name: Bucket name of respective cloud storage
5. ufile: Upload File
6. file_name: Filename of the file to be downloaded
7. exptime: Expiration time for temporary download link (in sec)

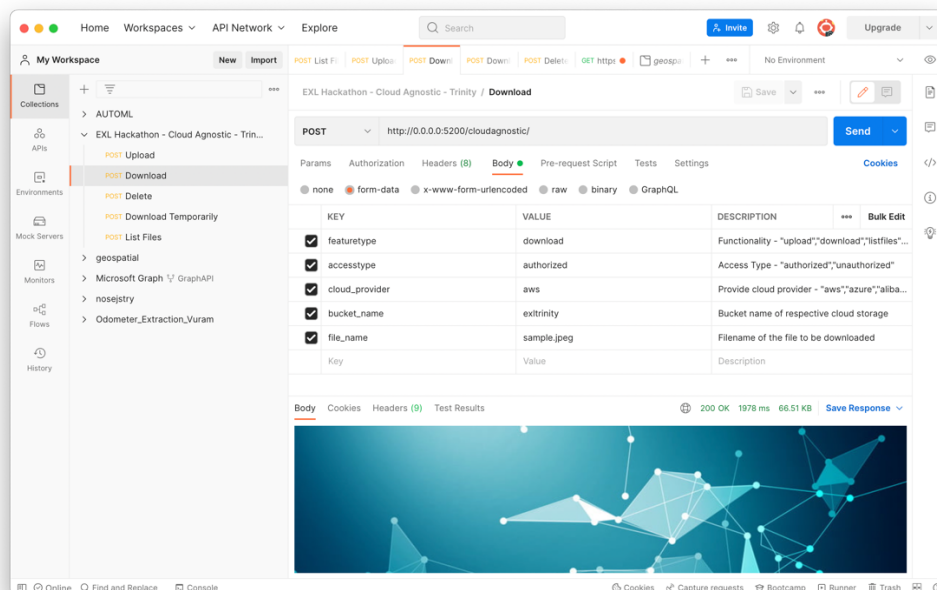
Sample Upload Case –

```
{  
  "featuretype": "upload"  
  "accesstype": "authorized"  
  "cloud_provider": "aws"  
  "bucket_name": "exltrinity"  
  "ufile": "<upload-file>"  
}
```



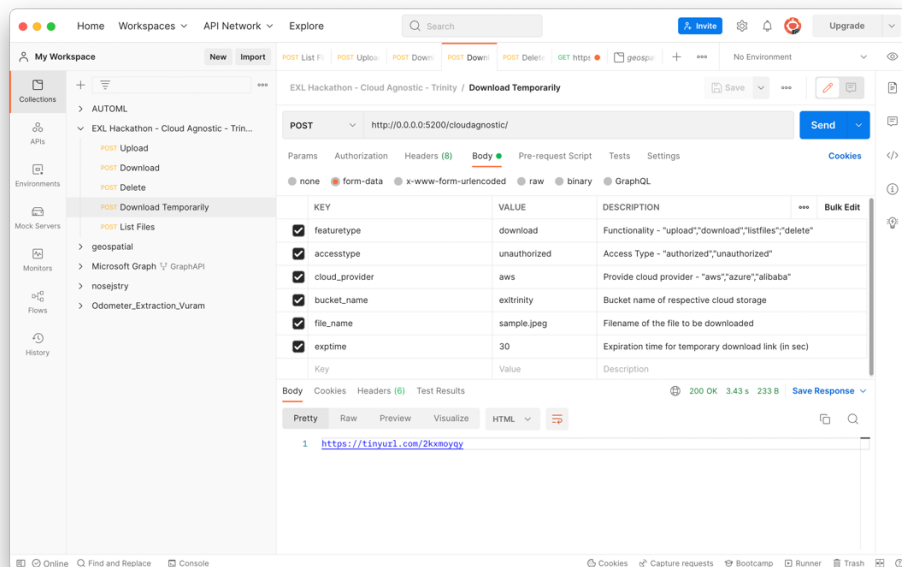
Sample Download Case –

```
{  
  "featuretype": "download"  
  "accesstype": "authorized"  
  "cloud_provider": "aws"  
  "bucket_name": "exltrinity"  
  "file_name": "sample.jpeg"  
}
```



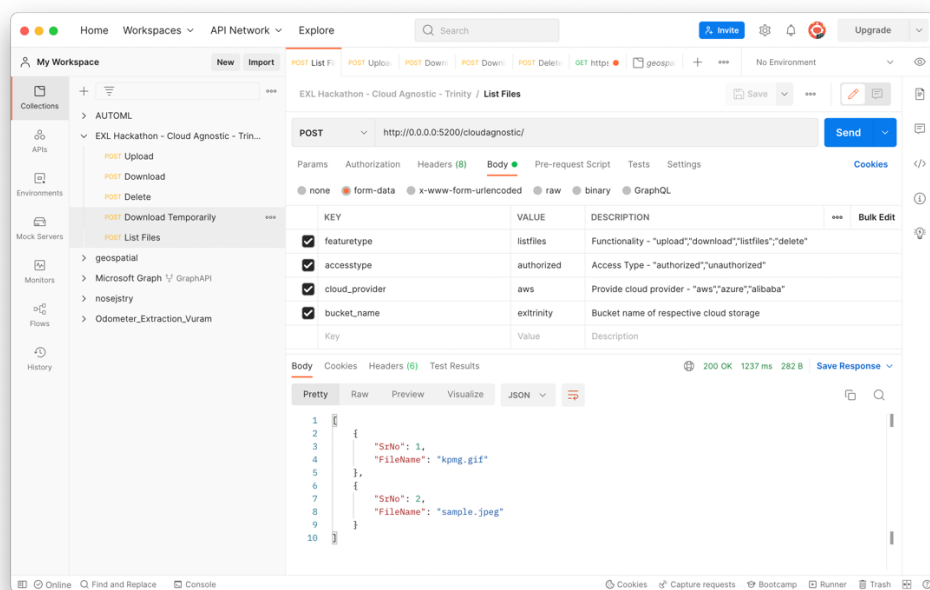
Sample Download Temporarily Case –

```
{  
  "featuretype": "download"  
  "accesstype": "unauthorized"  
  "cloud_provider": "aws"  
  "bucket_name": "exltrinity"  
  "file_name": "sample.jpeg"  
  "exptime": "30"  
}
```



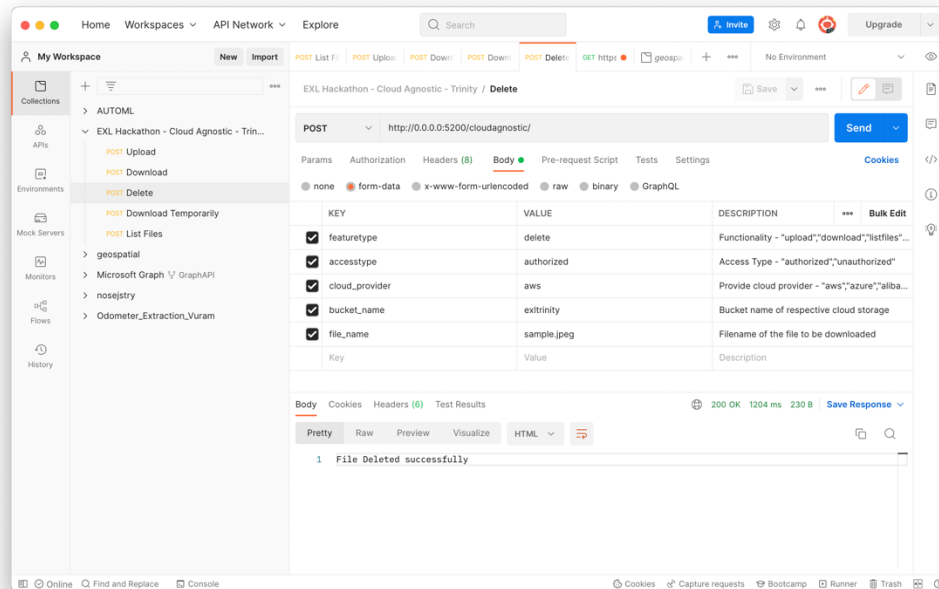
Sample List Files Case –

```
{  
  "featuretype": "listfiles"  
  "accesstype": "authorized"  
  "cloud_provider": "aws"  
  "bucket_name": "exltrinity"  
}
```



Sample Delete Case –

```
{  
  "featuretype": "delete"  
  "accesstype": "authorized"  
  "cloud_provider": "aws"  
  "bucket_name": "exltrinity"  
  "file_name": "sample.jpeg"  
}
```



- Post deleting files, Response for the “listfiles” feature:

