**TO CREATE DATA LAKE FROM DATA IN DATABASES**

1. **First Step: Extract Data from Databases, like, SQL Server, PostGres, MySQL**

**Using Python program**

1. **Select BackBlaze B2 as Cloud Storage:**

**S3-compatible B2 Cloud Storage at 1/5 the price**

**To install Python libraries for B2: pip install b2sdk**

**\*\* PYTHON CODE -- UPLOAD Files into B2 Cloud Storage using Python:**

***from b2sdk.v2 import B2Api, InMemoryAccountInfo***

***# Replace with your actual account ID and application key***

***ACCOUNT\_ID = 'Our\_ACCOUNT\_ID'***

***APPLICATION\_KEY = 'Our\_APPLICATION\_KEY'***

***# Create an InMemoryAccountInfo object***

***info = InMemoryAccountInfo()***

***# Create a B2Api instance***

***b2\_api = B2Api(info)***

***# Authorize the account***

***b2\_api.authorize\_account("production", ACCOUNT\_ID, APPLICATION\_KEY)***

***# Get the bucket***

***bucket\_name = "bucket-name-from-B2"***

***bucket = b2\_api.get\_bucket\_by\_name(bucket\_name)***

***# To Upload single file***

***file\_path = '/path/to/your/file.txt' ## path can be local***

***file\_name = 'our\_file\_in\_bucket.txt'***

***uploaded\_file = bucket.upload\_local\_file(local\_file=file\_path, file\_name=file\_name)***

***print(f"File uploaded successfully: {uploaded\_file}")***

***# Upload a file***

***file\_path = '/path/to/your/file.txt'***

***bucket\_name = 'our-bucket-name'***

***file\_name = 'the\_file\_in\_bucket.txt'***

***with open(file\_path, 'rb') as f:***

***file\_data = f.read()***

***upload\_url, upload\_auth\_token = b2\_api.get\_upload\_url(bucket\_name, file\_name)***

***response = b2\_api.upload\_file(upload\_url, upload\_auth\_token, file\_name, file\_data)***

***print(f"File uploaded successfully: {response}")***

***## To Upload multiple files***

***## Define a list of files to upload***

***file\_paths = [ "/path/to/file1.txt", "/path/to/file2.csv",***

***"/path/to/folder/subfolder/file3.pdf" ]***

***# Upload each file***

***for file\_path in file\_paths:***

***file\_name = os.path.basename(file\_path) ## Extract filename from path***

***try:***

***uploaded\_file = bucket.upload\_local\_file(local\_file=file\_path, file\_name=file\_name)***

***print(f"File uploaded successfully: {file\_name}")***

***except Exception as e:***

***print(f"Error uploading file {file\_path}: {e}")***