**1.Create a AWS Lambda function to read from any open source API such https://rapidapi.com/ and store it into s3 bucket?**

1. Ans:- Create an S3 bucket in your AWS account to store the data.
2. Create an API key on RapidAPI and note down the API endpoint URL.
3. Create a new Lambda function in your AWS account for Api data access by using python boto3 library and import request.
4. Set the following environment variables for the Lambda function:
   * **API\_URL**: The URL of the API endpoint to retrieve data from.
   * **API\_KEY**: Your RapidAPI key.
   * **S3\_BUCKET**: The name of the S3 bucket to store the data in.
5. Configure the Lambda function to run on a schedule (e.g. every hour) or trigger it manually to retrieve and store data.

**2.Create a redshift table to store record of s3 files data?**

Ans:-

CREATE TABLE s3\_data (

column\_1 data\_type,

column\_2 data\_type,

column\_3 data\_type,

...

)

DISTSTYLE KEY

DISTKEY (column\_1)

SORTKEY (column\_2);

COPY s3\_data

FROM 's3://your-bucket/your-path/'

IAM\_ROLE 'your-iam-role'

FORMAT AS FORMAT AS DELIMITER ','

IGNOREHEADER 1;

**3. Crate a AWS Glue Job to move data from s3 bucket to Amazon Redshift?**

Ans:-

1. First Create an IAM Role with permissions to access the S3 bucket and Amazon Redshift. To do this, go to the IAM console and create a new role with the AmazonS3FullAccess and AmazonRedshiftFullAccess policies attached.
2. Create a new Glue job in the AWS Glue console. Select "Add Job" and provide a name for the job.
3. In the "Job properties" section, set the "Type" to "Spark" and the "Glue version" to "Spark 2.4, Python 3". You can leave the other properties at their default values.
4. In the "Security configuration, script libraries, and job parameters" section, choose the IAM role you created earlier.
5. In the "Data source" section, select "S3" as the data source type. Enter the S3 bucket name and the path of the data files..
6. In the "Data target" section, select "Amazon Redshift" as the data target type. Enter the Redshift database connection information, including the JDBC URL, username, password, and database name.

**4. Schedule AWS Lambda function to download data automatically on specific interval?**

Ans:-

1. Create a Lambda function that contains required code for downloading data. We can write function code in any of the supported languages (Python, Node.js, Java, etc.).
2. Open the AWS CloudWatch console and navigate to the "Events" section.
3. Click on the "Create rule" button to create a new rule.
4. In the "Event Source" section, choose "Schedule" as the event source and set the desired interval for your function to run (e.g. every 5 minutes, every hour, etc.).
5. In the "Targets" section, choose "Lambda function" as the target and select the Lambda function you created in step 1.
6. Click on the "Create rule" button to save the rule.

Once you have created the CloudWatch Event rule, it will trigger your Lambda function at the specified interval, and your Lambda function will download the data as per your code. Note that you may need to set up any necessary permissions or authentication mechanisms for your Lambda function to access the data you want to download.

**5. Schedule AWS Glue Job to move data from s3 to Redshift?**

1. Create an IAM Role: Create an AWS Identity and Access Management (IAM) role with permissions to access the S3 bucket and Redshift cluster. The role should include the necessary permissions to read data from S3 and write to Redshift.
2. Create a Glue Crawler: Create a Glue Crawler to discover the data stored in your S3 bucket. The Crawler will create a metadata catalog for the data.
3. Create a Glue Job: Create an AWS Glue job that reads data from the metadata catalog created by the Crawler and writes it to Redshift. The job should include the necessary database and table mappings to ensure the data is loaded correctly into Redshift.
4. Configure Job Schedule: Set up a schedule for the job using the AWS Glue console or AWS Command Line Interface (CLI). You can schedule the job to run at specific times, intervals, or based on a trigger event.
5. Monitor Job Status: Monitor the status of the Glue job to ensure that it runs as expected. You can check the job logs in the AWS Glue console or AWS CloudWatch Logs.
6. Validate Data: Validate the data that was loaded into Redshift to ensure that it is accurate and complete.

By following these steps, you can create an AWS Glue job to move data from S3 to Redshift and schedule it to run on a regular basis. This will enable you to keep your Redshift database up-to-date with the latest data from your S3 bucket.