AWS S3 Storage Integration and Continuous Data Loading to Snowflake

1. In the AWS Console home, search for **S3** (**Simple Storage Service**) and click on it. Create an S3 bucket with a suitable name (e.g., **retaildata.raw**), in which raw data is loaded. Create folders inside the bucket for each data table.
2. Once the S3 bucket and folders are created, search for and select the **IAM** (**Identity and Access Management**) service from the AWS console.
3. Click on **Policies** from the IAM Dashboard and create an IAM policy for the bucket by clicking the "Create Policy" button.
4. Click on the JSON tab and replace the existing text with the provided policy text, ensuring to substitute <bucket> and <prefix> with our actual bucket name and folder path (bucket ARN) in the 'Resource' section of the JSON text. Permissions are defined in this policy document to specify which actions are allowed or denied.
5. Click Next, then skip the Add Tags section. Enter a name (e.g., **retailpolicy**) and description for the policy, and click Create Policy to complete the process.
6. Now, navigate to the IAM Dashboard and click on **Roles**. Click on 'Create Role'.
7. Choose **AWS Account** as the Trusted Entity Type; our account number should be selected by default. If required, check the 'External ID' (note that it is not chosen in this case), then click 'Next'**.**
8. Select the IAM policy (**retailpolicy**) that was created in previous steps to assign the policy to our new role. To define permissions for an IAM identity (user, user group, or role), we need to attach a policy to it.
9. Enter a unique name for your role (e.g., **retailrole**). The description is optional. Then click on 'Create Role' (you may skip the 'Add Tags' section).
10. Click on the newly created role to view its summary page. Note down the **Role ARN** as it will be required when creating the 'Storage Integration' in Snowflake.
11. Now, Log in to your Snowflake account. Create a Cloud Storage Integration (e.g., **S3_RETAILDATA_RAW_INT**) in Snowflake and map the S3 user (Role) to it.
12. In the Snowflake worksheet, run the 'DESC INTEGRATION' command and note down the **STORAGE_AWS_IAM_USER_ARN** and ('STORAGE_AWS_EXTERNAL_ID' if "External ID" was selected while creating the role) from the result set.
13. Then, go to the AWS Console, navigate to IAM Role, select the role you created, and click **Trust Relationships** -> **Edit trust policy**.
14. Replace the value of "**AWS**": with the **STORAGE_AWS_IAM_USER_ARN** string obtained from the 'DESC INTEGRATION' command in Snowflake, and replace the value of 'sts:ExternalId' with the 'AWS_EXTERNAL_ID' string (if "External ID" was selected while creating the role).
15. Then click on 'Update Policy'.
16. Now, create the Snowflake **file format** (e.g., **RETAIL_CSV**) in the Snowflake worksheet, which will be used during the stage creation.
17. Create an external (S3) **stage** (e.g., **RETAIL_STAGE**) that references the storage integration created previously.

18. Upon successful creation of the stage, list the stage. When we list a stage, we are essentially viewing the directory of the stage (i.e. S3 bucket) to see all the files that have been uploaded to it. This can help verify that the stage has been created successfully and that it contains the expected files.
19. Now, create an auto-ingest pipe, i.e., **Snowpipe**, which recognizes CSV files ingested from the external stage and copies the data into the existing table in Snowflake.
20. Ensure that the number of Snowpipes created matches the number of folders (tables) in the S3 bucket. This guarantees that each pipe is attached to its respective folder in the bucket.
21. The 'AUTO_INGEST = TRUE' parameter in each Snowpipe specifies to read event notifications sent from an S3 bucket to an SQS queue when new data is ready to load.
22. After creating Snowpipe, obtain the '**Notification Channel**' value by running the command 'SHOW PIPES'. Alternatively, you can go to the respective **Database** -> **Pipes** in Snowflake to find the notification channel value.
23. Navigate to the S3 bucket you created. Click on the 'Properties' tab, scroll down to 'Event Notifications', and click on 'Create Event Notification'.
24. Enter a name for the notification (e.g., **retaildata_event_notification**). Check the box for 'All Object Create Events'. Scroll down to 'Destination', select 'SQS Queue', choose 'Enter SQS Queue ARN', and paste the 'Notification Channel' under SQS Queue.
25. Then save the changes and now, we are all set to load the files into the S3 bucket.
26. Finally, we can check if any new data is updated in our S3 bucket by using this command in Snowflake "ALTER PIPE <pipe_name> REFRESH;". It is not mandatory to refresh the pipe every time; it is only for the user's purpose to ensure that all the pipe connections are successful and data is flowing from the S3 bucket to the Snowflake table.