

A. Configuring Secure Access to AWS S3

1. Create a S3 bucket where you want to stage the files for load

- Log into AWS Console
- Launch S3. Create a S3 Bucket <Name: **snfl-sp-demo**. Bucket name should be unique. Choose a unique name.>
- Optional step: Create a folder to place the files. The files can be placed in the root directory too.

2. Configure S3 Bucket Access Permissions

- Launch IAM.
- Confirming AWS Region is active
- Left pane --> Account Settings --> Expand the Security Token Service Regions list, find the AWS region corresponding to the region where your account is located, and choose Activate if the status is Inactive.
- Create a Policy that allows read and write permission to AWS S3 bucket
- Left pane --> Policies --> Create Policy --> Click JSON tab -- Add the below policy document that allows access to the S3 Bucket - <Name: snfl-sp-policy>

--<bucket>**snfl-sp-demo**. Bucket name should be unique. Choose a unique name.

--<prefix>**stage**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": "arn:aws:s3:::<bucket>/<prefix>/*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::<bucket>",
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            ""
          ]
        }
      }
    }
  ]
}
```

```
}  
  ]  
    }  
      }  
        ]  
          }  
            ]  
              }
```

3. Create Role

- The Role will grant permission to the stage using the policy that was created. AWS Console—> IAM—>Left pane --> Role --> Create Role --> Click Another AWS Account -->
- --Get the account ID from Top Right by clicking the name -- Account Name and get the Account ID (AWS Account ID)
- Select "Require external ID" and enter some dummy number '0000' --> Click "Next:Permissions" --> Click "Next:Tags" --> Click "Review" <Name: snfl-sp-role> --> Attach the policy that was created during the previous step snfl-sp-policy --> Create Role <Name: snfl-sp-role>
- Copy the Role ARN "arn:aws:iam::<AWS AccountID>:role/snfl-sp-role"

B. Snowpipe Configuration to AWS File system

1. Create External Stage

- Launch Snowflake Worksheets
 - Create Stage using the Role ARN that was copied

--set the context of the worksheet to meetup warehouse and schema

USE WAREHOUSE SNFL_ATL_MEETUP_WH;

USE SCHEMA SNFL_ATL_MEETUP_DB.SNFL_ATL_MEETUP_SCHEMA;

-- Create a file format.

CREATE OR REPLACE FILE FORMAT

SNFL_ATL_MEETUP_DB.SNFL_ATL_MEETUP_SCHEMA.SNFL_SP_JSON_FORMAT
TYPE = 'json';

--Please use the appropriate Bucket URL and role arn credentials

--Create an External Stage

CREATE OR REPLACE STAGE

SNFL_ATL_MEETUP_DB.SNFL_ATL_MEETUP_SCHEMA.SNFL_SP_DEMO_STAGE
_trial

url='s3://s3_bucketURL'

CREDENTIALS = (aws_role = 'role ARN')

FILE_FORMAT = SNFL_SP_JSON_FORMAT

encryption=(type='AWS_SSE_KMS' kms_key_id = 'aws/key');

Data Streaming into Snowflake using Snowpipe - Step by Step Instructions

- Execute the describe statement on the stage to get the property_value of Snowflake IAM User and AWS External ID from the Stage:

```
DESC STAGE
SNFL_ATL_MEETUP_DB.SNFL_ATL_MEETUP_SCHEMA.SNFL_SP_DEMO_STAGE
_trial;
```

```
SNOWFLAKE_IAM_USER: arn:aws:iam::<SNOWFLAKE ACCOUNT ID>:user/xxxxxxx
AWS_EXTERNAL_ID- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

2. Configure the IAM Role to Allow Access to the Stage

- Enable snowflake to access Stage through the role that was created (snfl-sp-role).
 - Launch IAM and click Role. Search for the role name we created in step 3.
 - Click the role and click on the Trust relationships tab --> click Edit Trust Relationship
 - Replace the AWS Account ID with the Snowflake user_ID (arn:aws:iam::<SNOWFLAKE ACCOUNT ID>:user/xxxxxxx)
 - Replace the External ID '0000' with the Snowflake External ID and Update the Trust Policy
 - Confirm that the files are accessible.

```
LIST
@SNFL_ATL_MEETUP_DB.SNFL_ATL_MEETUP_SCHEMA.SNFL_SP_DEMO
_STAGE_trial;
```