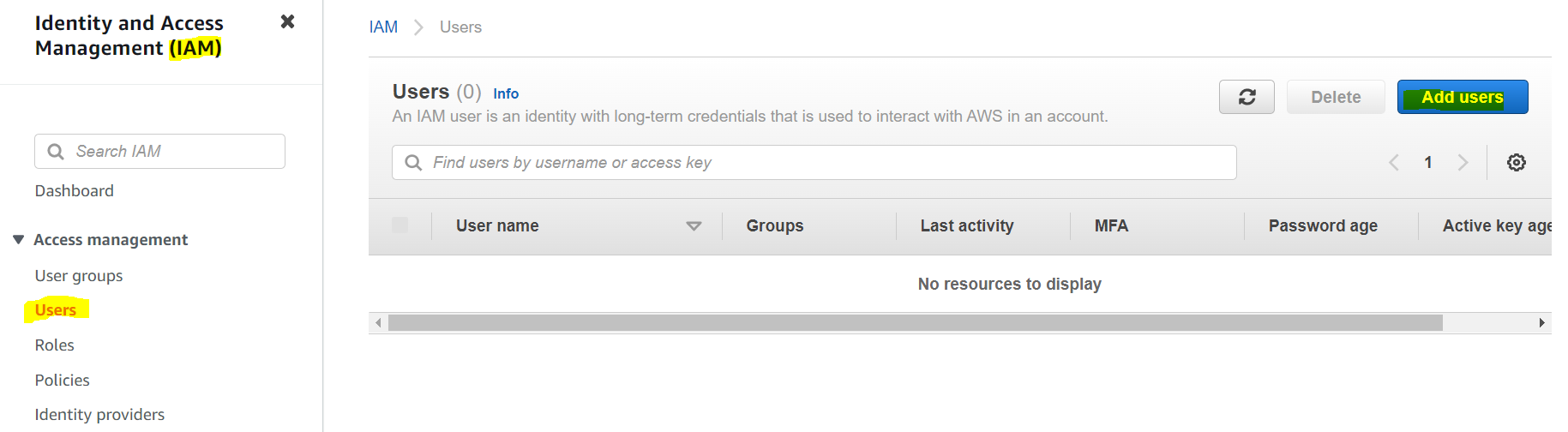
**Workshop step by step guide**

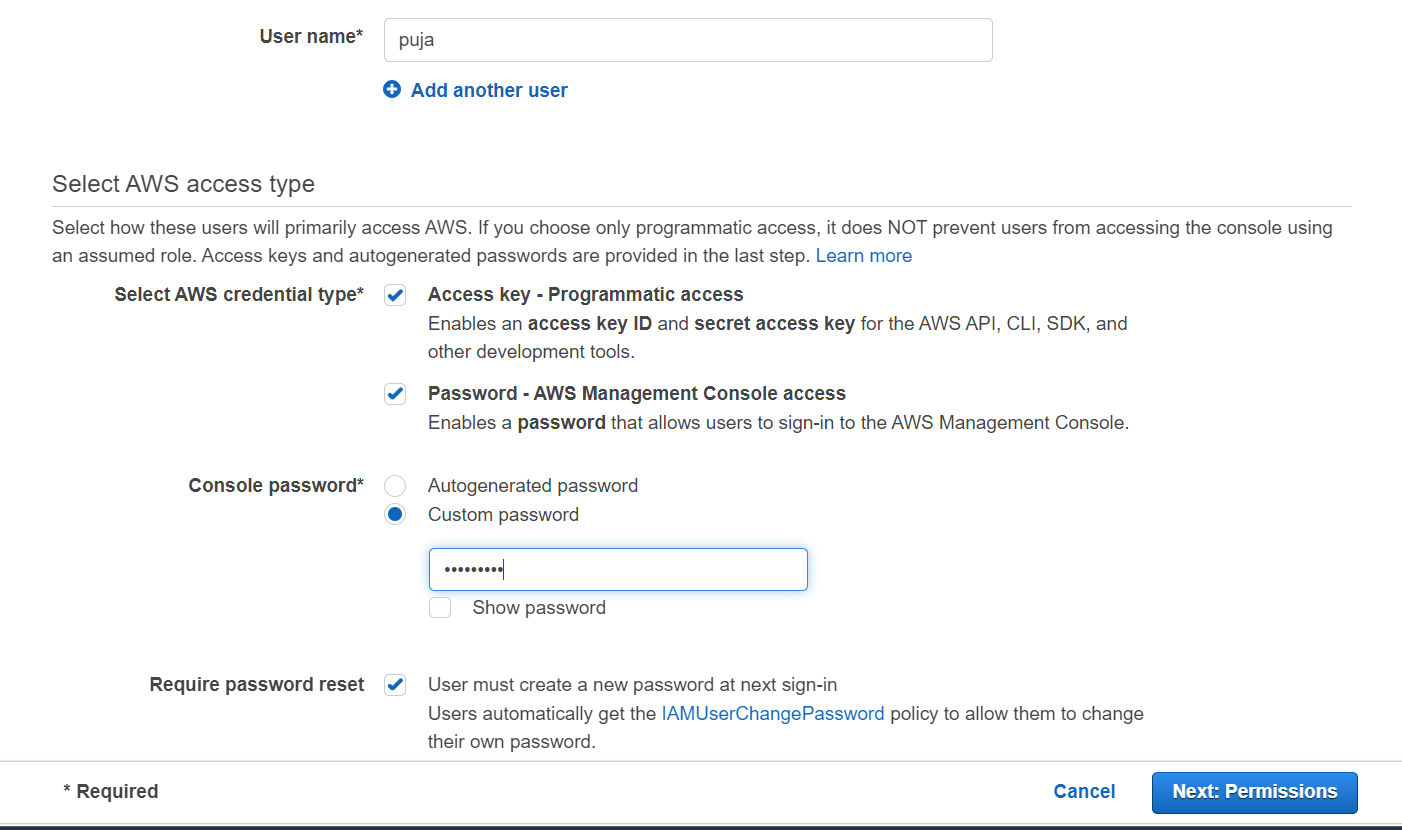
**Setup**

Sagemaker setup:

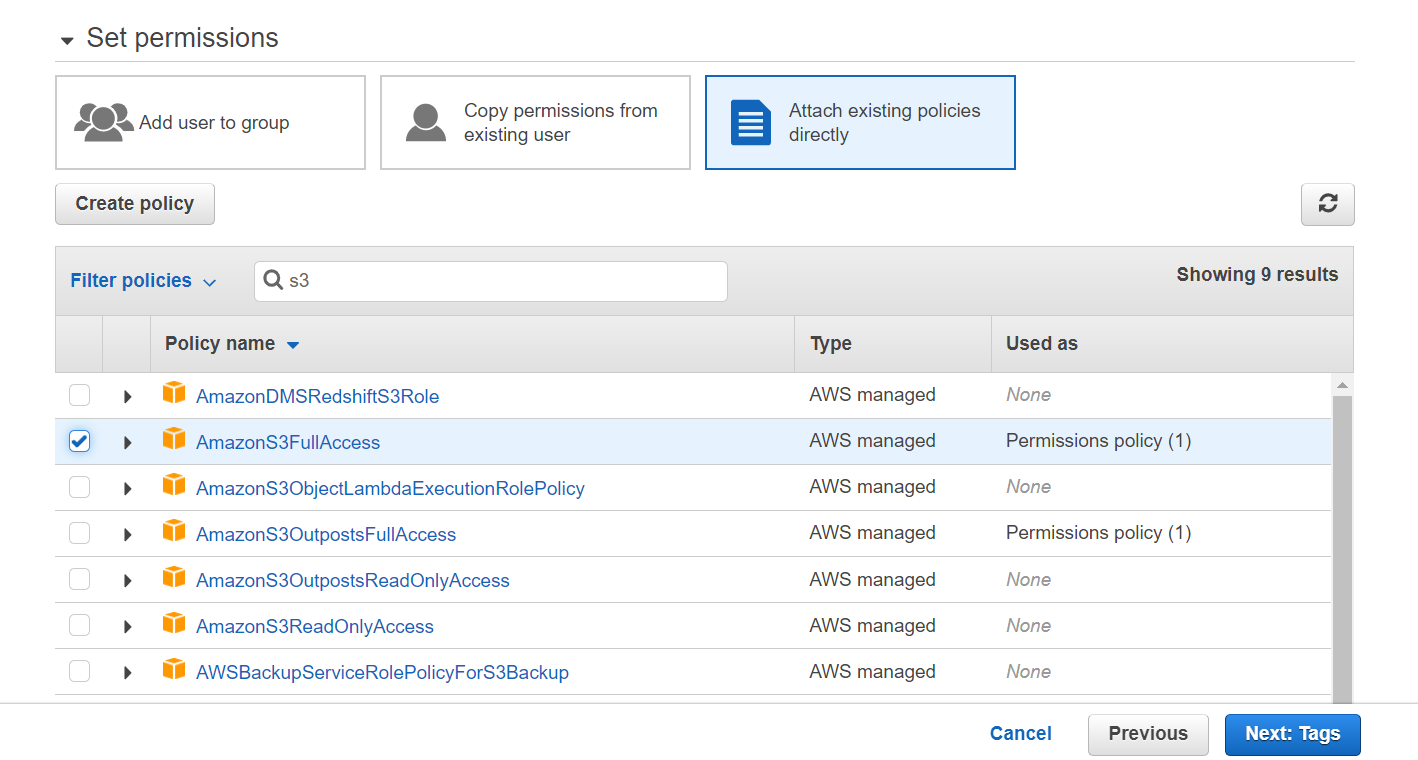
1. Clone github repo: <https://github.com/PujaVerma-06/sagemaker-snowpark-workshop.git>
2. Download census-income.csv data and upload to the sagemaker s3 bucket already created for you.
3. Create new s3 bucket (e.g. snowflake-stage-area)
4. Upload the downloaded census-income.csv in the new s3 stage bucket.
5. Download credit-g.csv file from the repo and upload this also in the new s3 stage bucket created.
6. Create aws key\_id and secret\_key (Go to IAM Role 🡪 Users🡪 Add users)



1. Add name in username, check both access key – programmatic and console, create password then click on next



1. Select Attach existing policy, search for s3, select se full access and click next



1. Click next (i.e. skip tags) and create user
2. Copy aws\_key\_id and aws\_secret\_key\_id and save it somewhere.

Snowflake setup:

1. Create Snowflake free trial account with cloud as AWS
2. In Snowflake worksheet, copy the following command and replace with your s3 stage bucket name and aws\_key\_id and aws\_secret\_key\_id

USE ROLE ACCOUNTADMIN;

CREATE WAREHOUSE IF NOT EXISTS WORKSHOPWH

WITH WAREHOUSE\_SIZE = 'XSMALL'

WAREHOUSE\_TYPE = 'STANDARD'

AUTO\_SUSPEND = 300

AUTO\_RESUME = TRUE

MIN\_CLUSTER\_COUNT = 1

MAX\_CLUSTER\_COUNT = 2

SCALING\_POLICY = 'STANDARD'

COMMENT = 'Created wh for workshop';

USE WAREHOUSE WORKSHOPWH;

CREATE DATABASE IF NOT EXISTS WORKSHOPDB;

CREATE SCHEMA IF NOT EXISTS WORKSHOPSCH;

USE DATABASE WORKSHOPDB;

USE SCHEMA WORKSHOPSCH;

CREATE FILE FORMAT IF NOT EXISTS PUBLIC.CSV\_FILE\_FORMAT

TYPE = 'CSV' COMPRESSION = 'AUTO' FIELD\_DELIMITER = ','

RECORD\_DELIMITER = '\n' SKIP\_HEADER = 1

FIELD\_OPTIONALLY\_ENCLOSED\_BY = '\042'

TRIM\_SPACE = FALSE ERROR\_ON\_COLUMN\_COUNT\_MISMATCH = TRUE

ESCAPE = 'NONE' ESCAPE\_UNENCLOSED\_FIELD = '\134'

DATE\_FORMAT = 'AUTO' TIMESTAMP\_FORMAT = 'AUTO' NULL\_IF = ('\\N');

CREATE STAGE IF NOT EXISTS SNOWFLAKE\_S3\_STAGE

url = 's3://{snowflake-stage-area-bucket}/'

credentials = (aws\_key\_id = ''

aws\_secret\_key = '')

;

create or replace table sagemaker\_table (

"age" number,

"class of worker" varchar2,

"education" varchar2,

"major industry code" varchar2,

"capital gains" number,

"capital losses" number,

"dividends from stocks" number,

"num persons worked for employer" number,

"income" number

);

copy into sagemaker\_table

from

@snowflake\_s3\_stage/census-income.csv

file\_format = (format\_name='public.csv\_file\_format');

create or replace table snowpark\_dataset (

checking\_status varchar,

duration number,

credit\_history varchar,

purpose varchar,

credit\_amount number,

savings\_status varchar,

employment varchar,

installment\_commitment number,

personal\_status varchar,

ther\_parties varchar,

residence\_since number,

property\_magnitude varchar,

age varchar,

other\_payment\_plans varchar,

housing varchar,

existing\_credits number,

job varchar,

num\_dependents number,

own\_telephone varchar,

foreign\_worker varchar,

class varchar

);

copy into snowpark\_dataset

from

@snowflake\_s3\_stage/credit-g.csv

file\_format = (format\_name='public.csv\_file\_format');

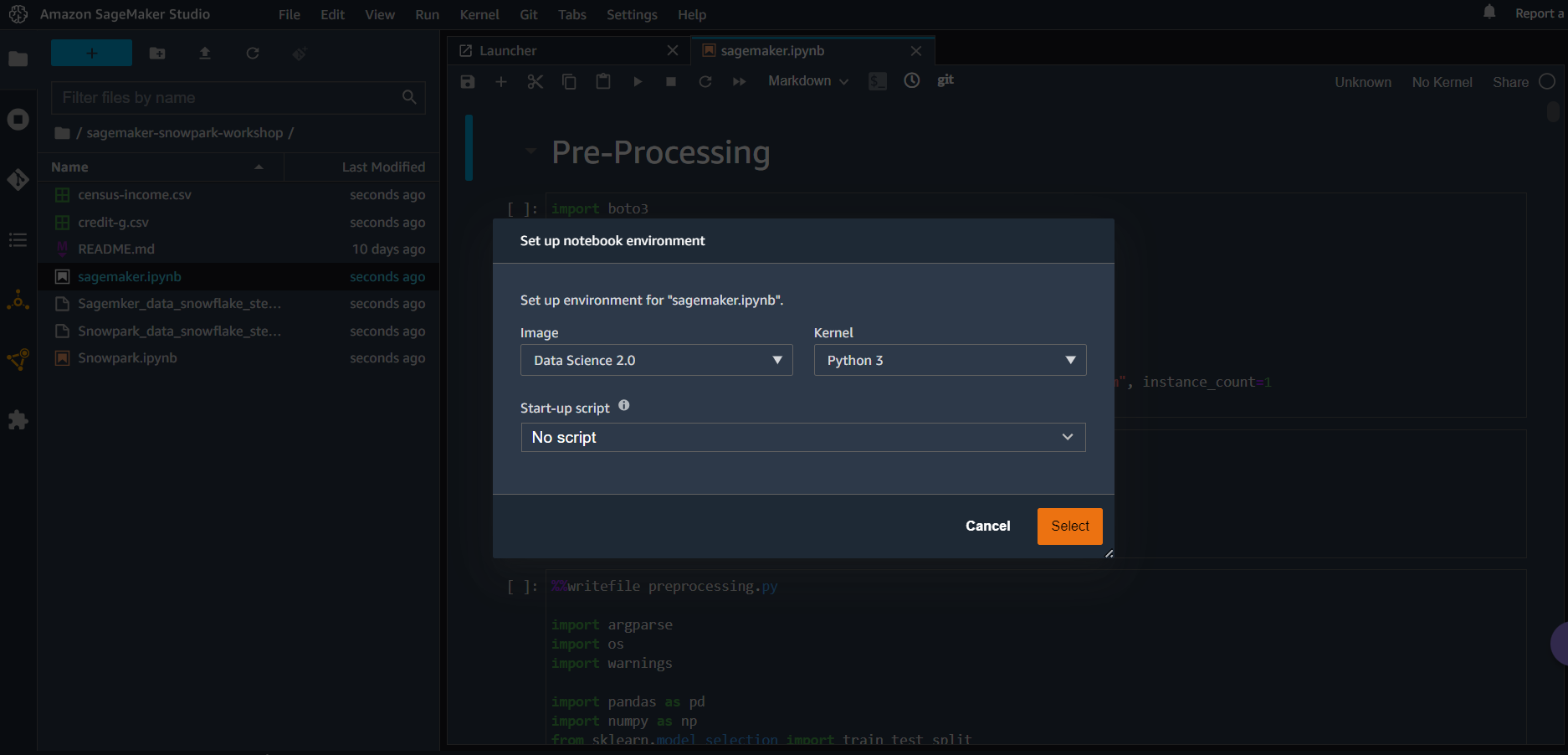
CREATE STAGE IF NOT EXISTS SNOWFLAKESTAGE

COMMENT = 'Workshop Stage in Snowflake';

In the notebook, in one of the cells update these snowflake connection parameters

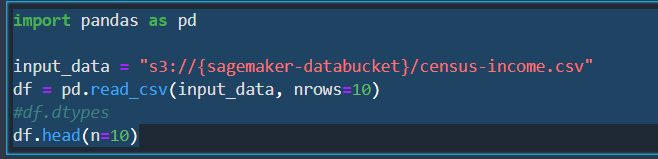
2 tables sagemaker\_table and snowpark\_dataset should be generated

**Workshop 1: PART A : Using s3 as input and output source**

1. Go inside the folder🡪 Open sagemaker.ipynb file   
   A pop-up window will open to set environment, select Image as Data Science 2.0, Kernel Python 3 and start-up script as No script as shown below and click on select  
   

The notebook kernel will take some time to start

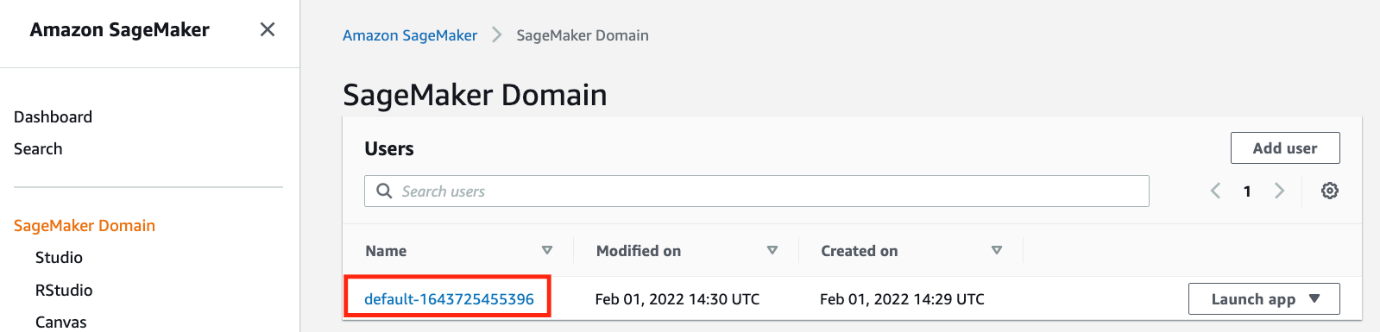
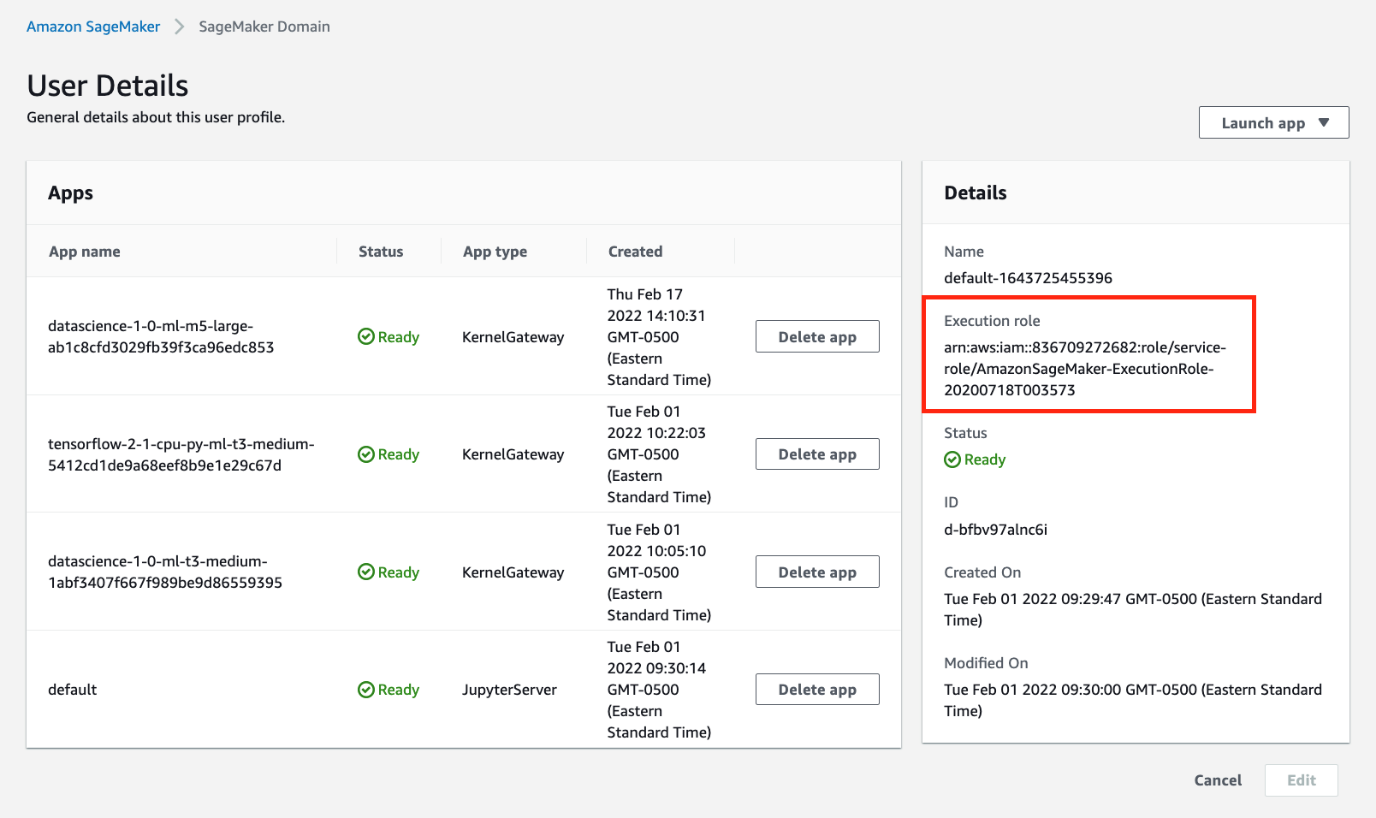
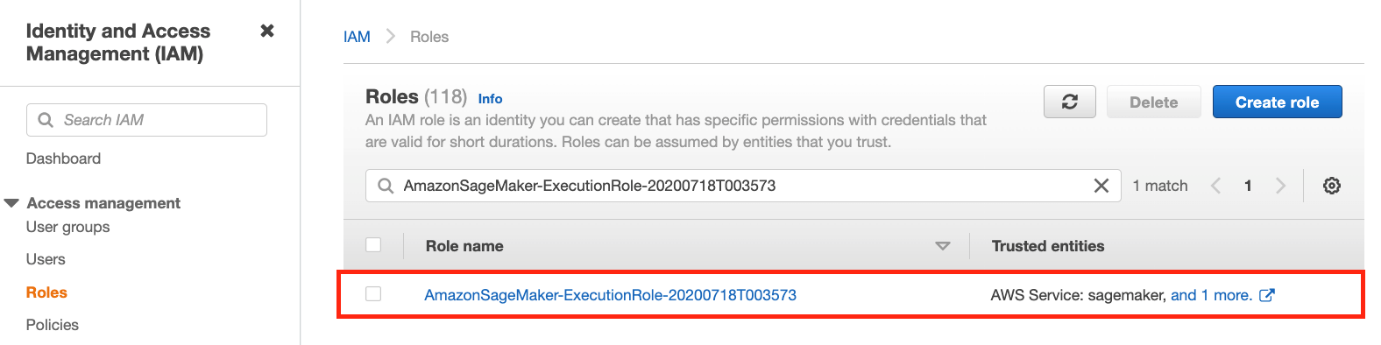
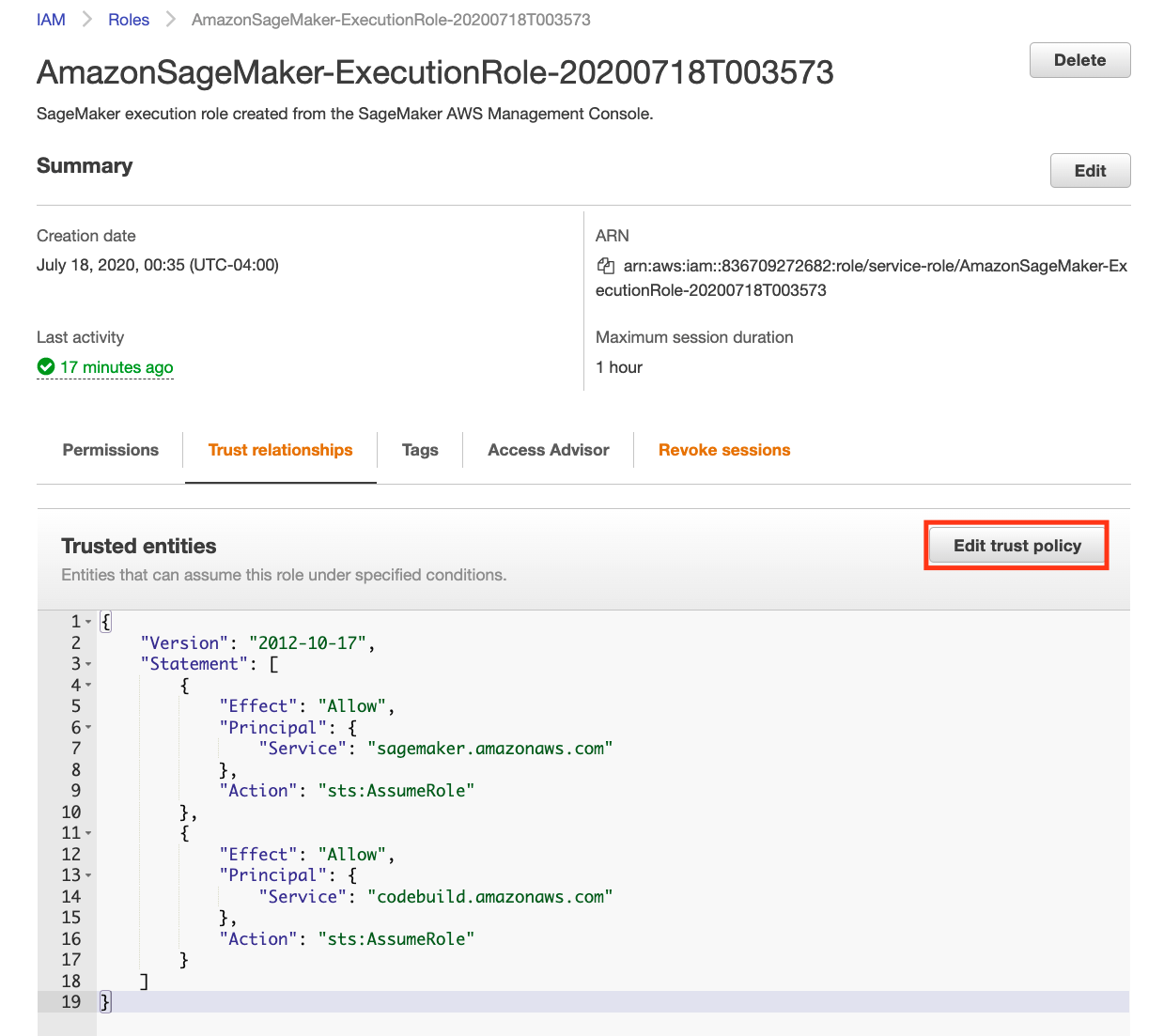
1. In the notebook replace the bucket name to your accounts sagemaker s3 bucket name



1. Start executing cells under **s3 as input and output source**

i.e. Execute all of the cells under Pre-processing, Training, model evaluation and batch inference.

**Workshop 1: PART B : Using Snowflake as input and output source**

1. Open AWS Console
2. Search for sagemaker
3. go to SageMaker Studio, then click on the SageMaker Studio "User name”name" 
4. copy the name of the SageMaker Studio execution role (not the full ARN)ARN). 
5. go to IAM, then click on "Roles", then search roles by the name of the execution role you copied above
6. click on the "Trust relationships" tab, then click on "Edit trust relationship"r
7. in the "Edit trust relationship" text area, paste the json iam policy from the codeblock below then click on "Update Trust Policy"

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"Service": "sagemaker.amazonaws.com"

},

"Action": "sts:AssumeRole"

},

{

"Effect": "Allow",

"Principal": {

"Service": "codebuild.amazonaws.com"

},

"Action": "sts:AssumeRole"

}

]

}

1. click on the "Permissions" tab in IAM roles, then click on "Create inline policy"
2. in the "Create policy" page click on the "JSON" tab, then copy the below and paste it in the text area replacing the default text, then click on "Review policy". In Name\*, enter SageMakerCodeBuildPolicy, select "Create policy"

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"codebuild:DeleteProject",

"codebuild:CreateProject",

"codebuild:BatchGetBuilds",

"codebuild:StartBuild"

],

"Resource": "arn:aws:codebuild:\*:\*:project/sagemaker-studio\*"

},

{

"Effect": "Allow",

"Action": "logs:CreateLogStream",

"Resource": "arn:aws:logs:\*:\*:log-group:/aws/codebuild/sagemaker-studio\*"

},

{

"Effect": "Allow",

"Action": [

"logs:GetLogEvents",

"logs:PutLogEvents"

],

"Resource": "arn:aws:logs:\*:\*:log-group:/aws/codebuild/sagemaker-studio\*:log-stream:\*"

},

{

"Effect": "Allow",

"Action": "logs:CreateLogGroup",

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"ecr:CreateRepository",

"ecr:BatchGetImage",

"ecr:CompleteLayerUpload",

"ecr:DescribeImages",

"ecr:DescribeRepositories",

"ecr:UploadLayerPart",

"ecr:ListImages",

"ecr:InitiateLayerUpload",

"ecr:BatchCheckLayerAvailability",

"ecr:PutImage"

],

"Resource": "arn:aws:ecr:\*:\*:repository/sagemaker-studio\*"

},

{

"Effect": "Allow",

"Action": "ecr:GetAuthorizationToken",

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"s3:GetObject",

"s3:DeleteObject",

"s3:PutObject"

],

"Resource": "arn:aws:s3:::sagemaker-\*/\*"

},

{

"Effect": "Allow",

"Action": [

"s3:CreateBucket"

],

"Resource": "arn:aws:s3:::sagemaker\*"

},

{

"Effect": "Allow",

"Action": [

"iam:GetRole",

"iam:ListRoles"

],

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": "iam:PassRole",

"Resource": "arn:aws:iam::\*:role/\*",

"Condition": {

"StringLikeIfExists": {

"iam:PassedToService": "codebuild.amazonaws.com"

}

}

}

]

}

1. Run cells under build your custom container
2. Replace below values of the connection\_parameters in one of the notebook cell under pre-processing using custom container with your snowflake connection values

CONNECTION\_PARAMETERS = {

"account": "",

"user": "",

"password": "",

"role": "",

"warehouse": "",

"database": "",

"schema": "",

}

1. And run the remaining cells of the notebook (i.e. pre-processing, training, inference).

**Workshop 2: ML using SnowPark**

1. Open Snowpark.ipynb
2. On environment setup window, again select image as Data Science 2.0, Kernel as Python3
3. Give access to anaconda in Snowflake by following below steps

Log into Snowsight, the Snowflake web interface.

Click the dropdown menu next to your login name, then click Switch Role » ORGADMIN to change to the organization administrator role.

Click Admin » Billing » Terms & Billing.

Scroll to the Anaconda section and click the Enable button. The Anaconda Packages (Preview Feature) dialog opens.

Click the link to review the Snowflake Third Party Terms.

If you agree to the terms, click the Acknowledge & Continue button.

1. Start executing notebook cells.