Remote Python Script

The Remote Python Script Snap is a **Transform type Snap** that enables you to execute a Python script on a Remote Python Executor (RPE)

***Prerequisites:***

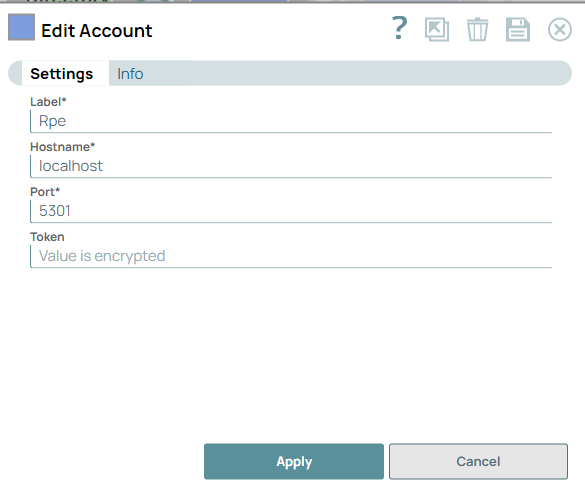
*Docker image* ***ds***: the data science version for CPU instances containing recommended libraries for data science.

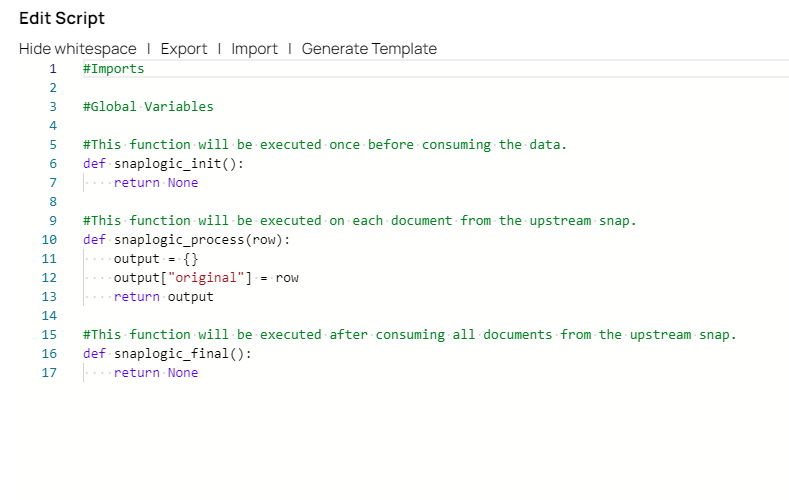
The steps to install RPE in docker

sudo docker pull snaplogic/rpe:ds

sudo docker run --memory-swap="-1" --restart=always -dti -p 5301:5301 -e "REMOTE\_PYTHON\_EXECUTOR\_TOKEN=" -v /opt/remote\_python\_executor\_log/:/opt/remote\_python\_executor\_log/ --name=rpe snaplogic/rpe:ds

Add Rpe account



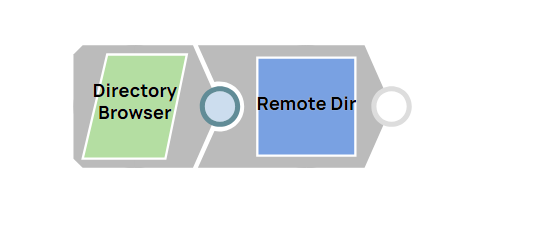
Editing Script in a snap

Code should be written side the snaplogic\_process() function code gets executed.  
row is a parameter those are the inputs to be transformed in a code.

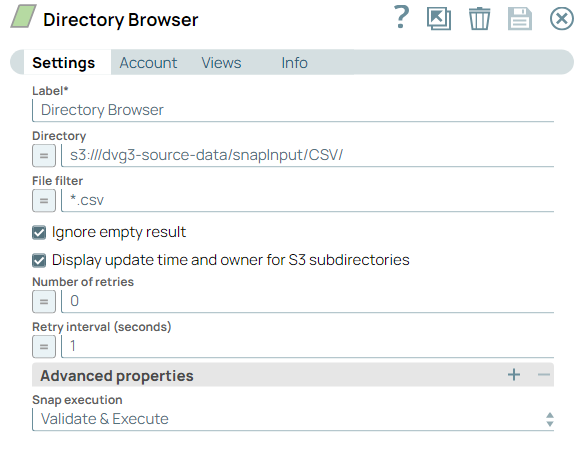
When working with data (csv data), row represents the row of csv data

We can use python libraries by simply pulling the libraries in the container.

**Here is the use case to convert csv data files from the directory convert it to parquet file and loaded into S3 bucket using the script.**



Configuring Directory Browser

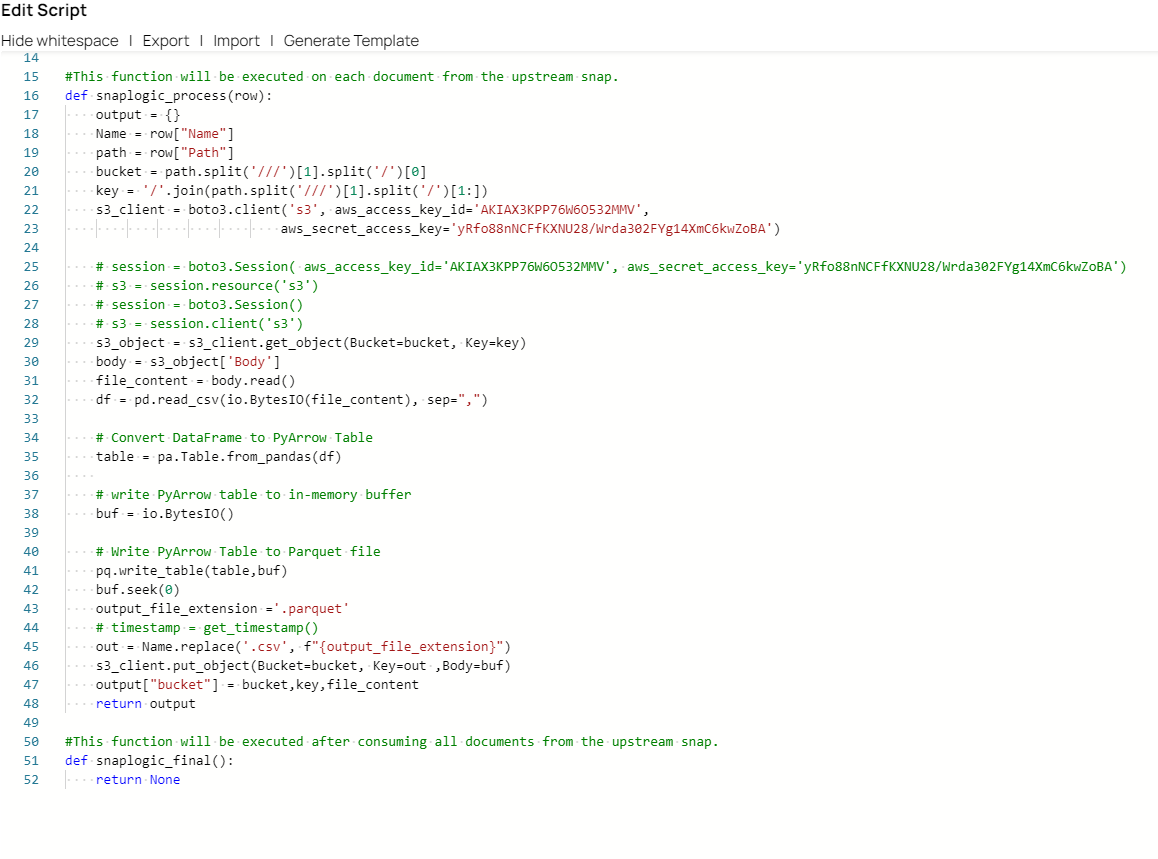


The directory is path where we have a source files

Filter: to get only csv data

**Remote python script**

The below script will read csv file from the path (that is coming from the argument -row) in S3 and convert those into parquet file and loaded into S3 bucket



**The code wriiten in process()**

output = {}

Name = row["Name"]

path = row["Path"]

bucket = path.split('///')[1].split('/')[0]

key = '/'.join(path.split('///')[1].split('/')[1:])

s3\_client = boto3.client('s3', aws\_access\_key\_id='yur access id',

aws\_secret\_access\_key='yur access key’)

# session = boto3.Session( aws\_access\_key\_id='AKIAX3KPP76W6O532MMV', aws\_secret\_access\_key='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')

# s3 = session.resource('s3')

# session = boto3.Session()

# s3 = session.client('s3')

s3\_object = s3\_client.get\_object(Bucket=bucket, Key=key)

body = s3\_object['Body']

file\_content = body.read()

output["bucket"] = bucket,key,file\_content

df = pd.read\_csv(io.BytesIO(file\_content), sep=",")

# Convert DataFrame to PyArrow Table

table = pa.Table.from\_pandas(df)

# write PyArrow table to in-memory buffer

buf = io.BytesIO()

# Write PyArrow Table to Parquet file

pq.write\_table(table,buf)

buf.seek(0)

output\_file\_extension ='.parquet'

# timestamp = get\_timestamp()

outKey = 'snapInput/parquet/'

out = outKey + Name.replace('.csv', f"{output\_file\_extension}")

s3\_client.put\_object(Bucket=bucket, Key=out ,Body=buf)

output["bucket"] = bucket,key,file\_content