AWS

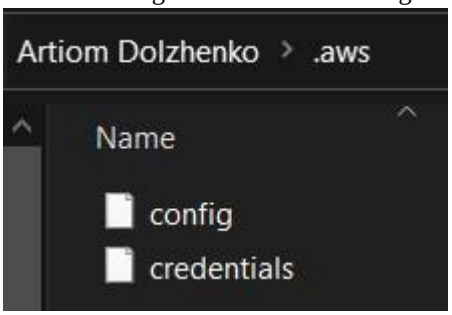
1 REPORT

CONTENTS

TASK 1 ESTABLISH CONNECTION TO AWS ACCOUNT: 2

TASK 1 ESTABLISH CONNECTION TO AWS ACCOUNT:

a. All settings were set before realizing that it's part of a home task. So, you have to believe that all is set correctly, otherwise it wouldn't work.
b. Keys were generated.
c. Config and credentials were generated automatically using "aws configure" in CMD.



d. As shown below i have a python script that is generating tokens based on my key and mfa code.

```python
import boto3
import configparser
import sys

prm_profile = 'default'_# profile with your user credentials (they are used to retrieve temp creds)
prm_tempprofile = 'epam_lab_mfa'_# profile where to record temp credentials to
prm_credfile = 'C:/Users/Artiom_Dolzhenko/.aws/credentials'_# filename where profile creds are stored
prm_user_arn = 'arn:aws:iam::260586643565:mfa/artiom_dolzhenko@epam.com'_# your user arn
prm_token_code = sys.argv[1]_# token_code from your MFA device

# uses credentials to get temporary token info
def get_token(profile_name:str, user_arn:str, token_code:str):
    try:
        session = boto3.session.Session(profile_name=profile_name)
        client = session.client('sts')
        response = client.get_session_token(SerialNumber=user_arn,TokenCode=token_code)
```

But it's annoying to write "python <script_name> <mfa_code> in cmd every single time, so I had to implement a ".bat" file that accepts mfa code and starts a script using that code.

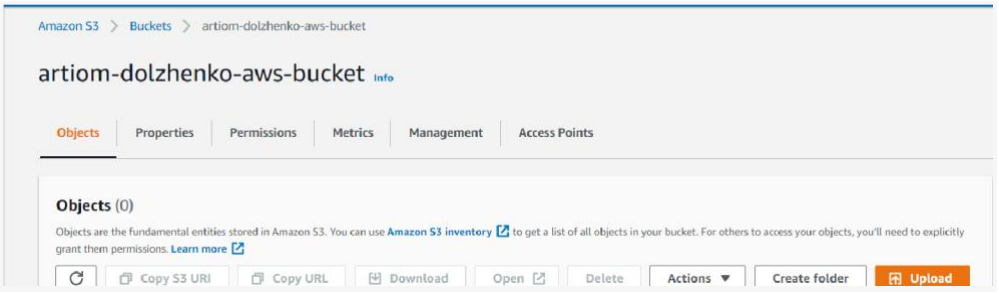
```
@echo off
set PGCLIENTENCODING=UTF8
set RSPORT=5439
set /p token="Give me a token for Epam Lab account host: "

python prepare-mfa-epam-lab.py %token%

pause
```

```
echo "Updating temporary security credentials for profile $AWS_TEMP_PROFILE .."

output=`aws --profile "$AWS_PROFILE" sts get-session-token --serial-number $ARN_OF_MFA --token-code $MFA_TOKEN_CODE --output text`

array=( $output )
AWS_ACCESS_KEY_ID=${array[1]}
AWS_SECRET_ACCESS_KEY=${array[3]}
AWS_SESSION_TOKEN=${array[4]}

aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profile=$AWS_TEMP_PROFILE
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --profile=$AWS_TEMP_PROFILE
aws configure set aws_session_token $AWS_SESSION_TOKEN --profile=$AWS_TEMP_PROFILE
aws configure set region $AWS_REGION --profile=$AWS_PROFILE
aws configure set output $AWS_OUTPUT --profile=$AWS_PROFILE

echo "Testing"

TEST=`aws s3 ls --profile=$AWS_TEMP_PROFILE`
if [ $? -ne 0 ]; then
  echo "Failed"
  exit 1
else
  echo "Successful!"
fi
```

After that, all is running

The next step is to create bucket in s3
The task was done using this command

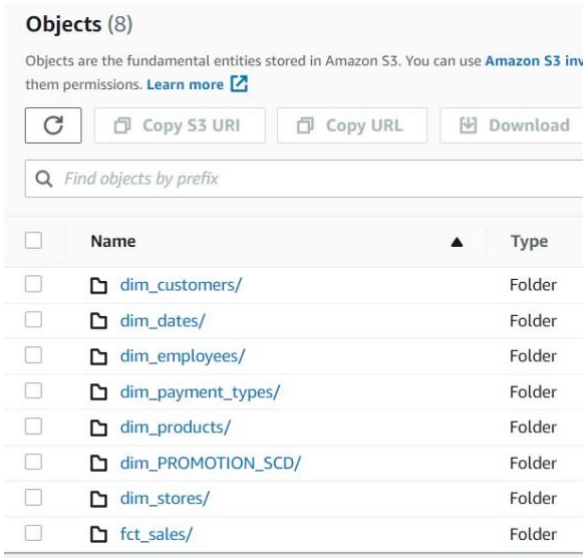
```
Create bucket;
aws s3api create-bucket --bucket artiom-dolzhenko-aws-bucket --region eu-central-1 --profile epam_lab_mfa
```



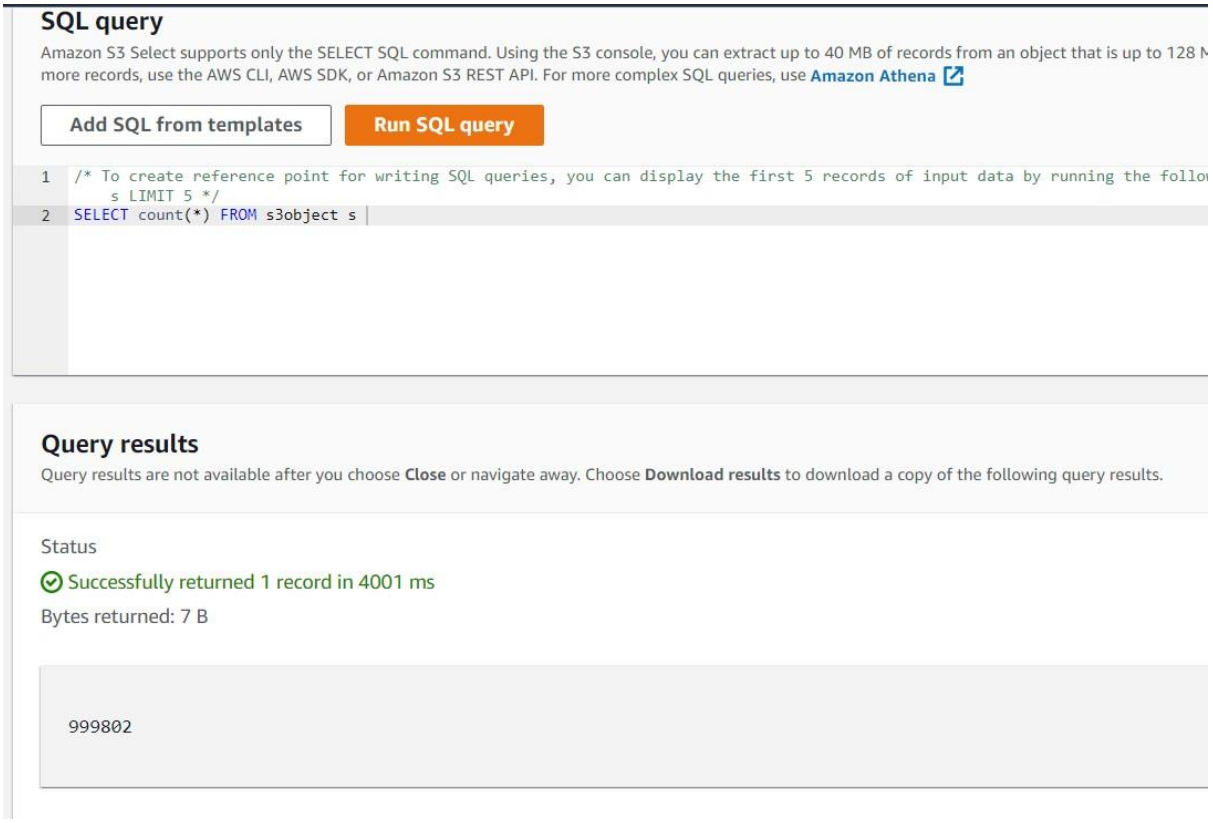
Files were uploaded to bucket via cmd

```
C:\Users\Artiom_Dolzhenko>aws s3api put-object --bucket artiom-dolzhenko-aws-bucket --key stores/bl_dm/dim_customers/
dim_customers.scv --body C:\Users\Artiom_Dolzhenko\TABLES_FOR_AWS/dim_customers.csv --profile epam_lab_mfa
{
    "ETag": "\"cb941404b3a077b87e43d11f3ad031f1\""
}

C:\Users\Artiom_Dolzhenko>aws s3api put-object --bucket artiom-dolzhenko-aws-bucket --key stores/bl_dm/dim_employees/
dim_employees.scv --body C:\Users\Artiom_Dolzhenko\TABLES_FOR_AWS/dim_employees.csv --profile epam_lab_mfa
{
    "ETag": "\"e7cfc4bfcb4d574f47ed0ecbe74e08c7\""
}
```

Now it’s all uploaded to cloud

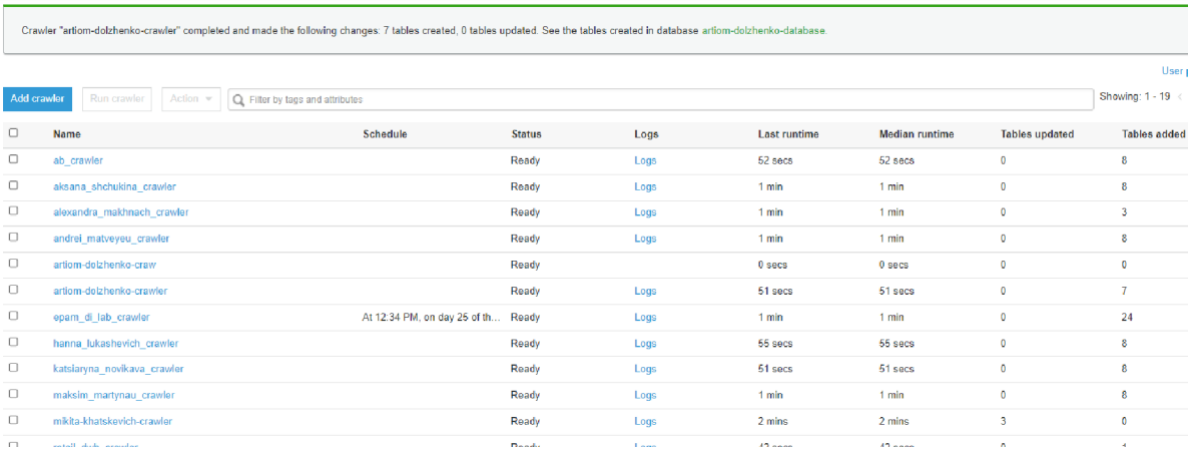


After uploading files I counted rows to see if the amount data in fct_sales was correct.
It was made in S3



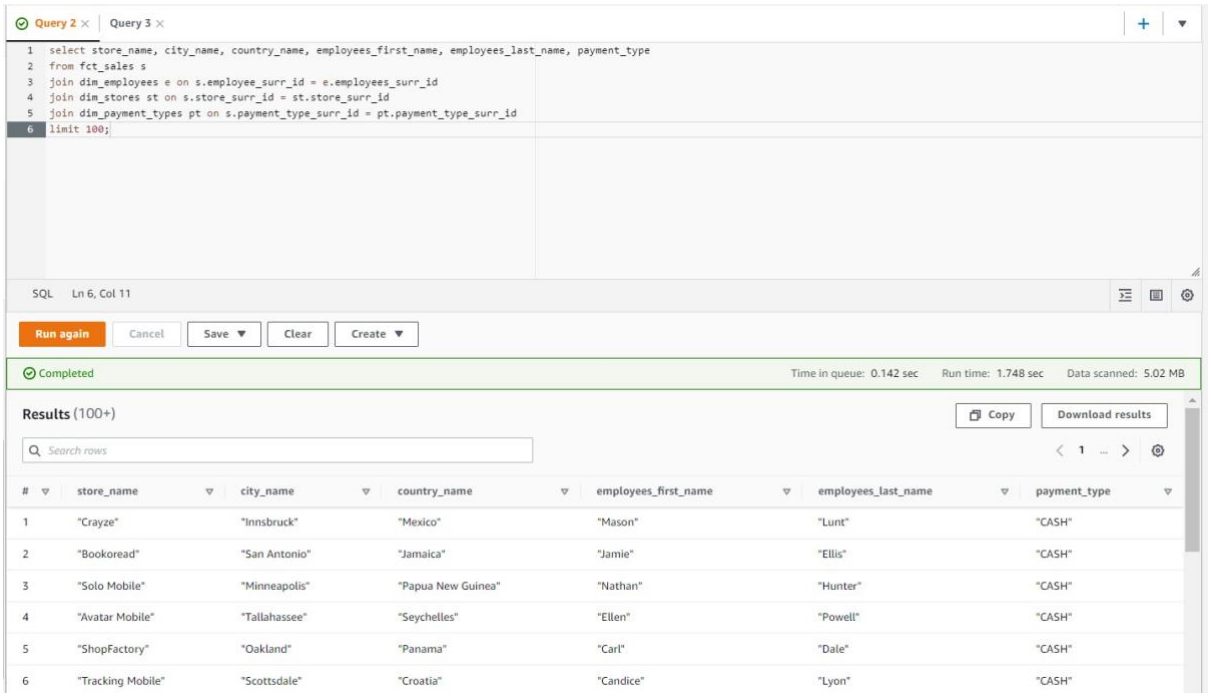
Crawler:

In my case crawler settings were default, because of clean csv files.



I faced a problem with query running. The reason was that I forgot to rerun crawler after adding missing tables.

Finally i wrote a query in athena as you can see it actually works.


Query 2
Query 3
select store_name, city_name, country_name, employees_first_name, employees_last_name, payment_type
from fct_sales s
join dim_employees e on s.employee_surr_id = e.employees_surr_id
join dim_stores st on s.store_surr_id = st.store_surr_id
join dim_payment_types pt on s.payment_type_surr_id = pt.payment_type_surr_id
limit 100;
SQL Ln 6, Col 11
Run again
Cancel
Save
Clear
Create
Completed
Time in queue: 0.142 sec
Run time: 1.748 sec
Data scanned: 5.02 MB
Results (100+)
Copy
Download results
Search rows
1 ...
store_name city_name country_name employees_first_name employees_last_name payment_type
1 "Crayze" "Innsbruck" "Mexico" "Mason" "Lunt" "CASH"
2 "Bookoread" "San Antonio" "Jamaica" "Jamie" "Ellis" "CASH"
3 "Solo Mobile" "Minneapolis" "Papua New Guinea" "Nathan" "Hunter" "CASH"
4 "Avatar Mobile" "Tallahassee" "Seychelles" "Ellen" "Powell" "CASH"
5 "ShopFactory" "Oakland" "Panama" "Carl" "Dale" "CASH"
6 "Tracking Mobile" "Scottsdale" "Croatia" "Candice" "Lyon" "CASH"