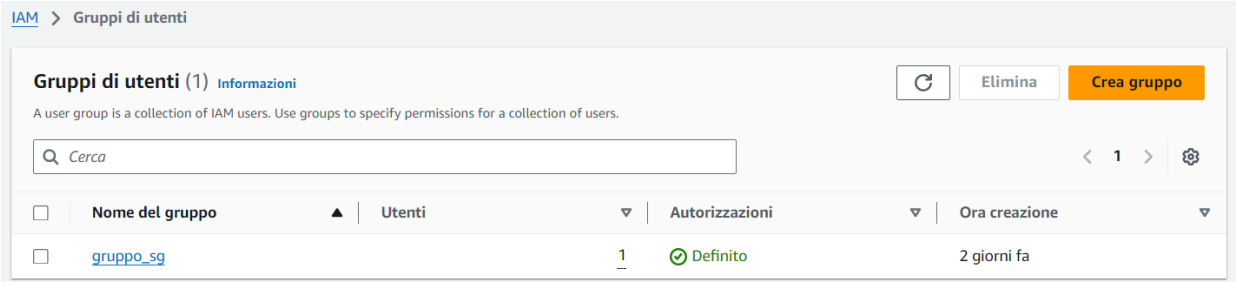
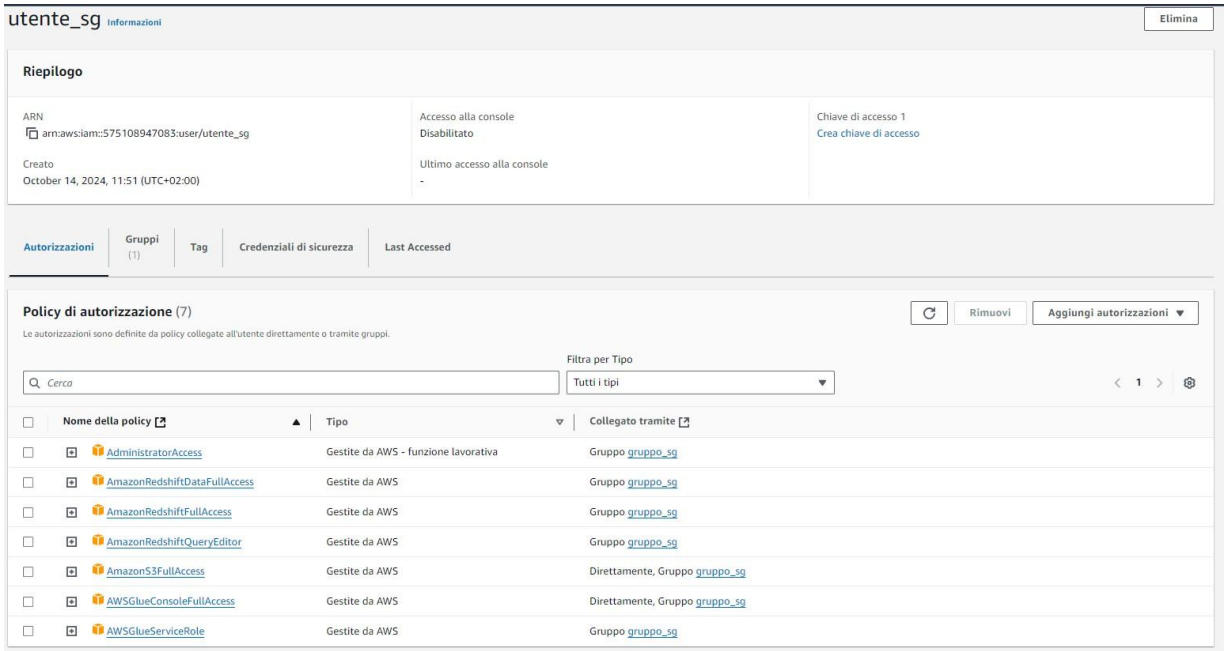
Indication of how the exercise was carried out

1 - Preparation of the Environment

1.1 - Account and Group Creation.

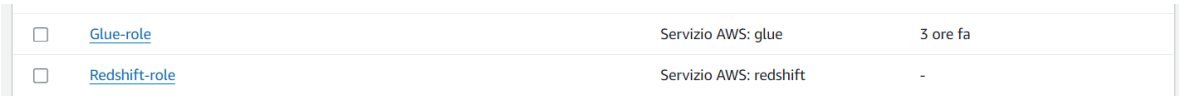
- An AWS account and a specific user group was created to manage the resources needed for the pipeline.
- Users in the group were associated with all necessary administrative policies to facilitate access and management of the AWS services involved.

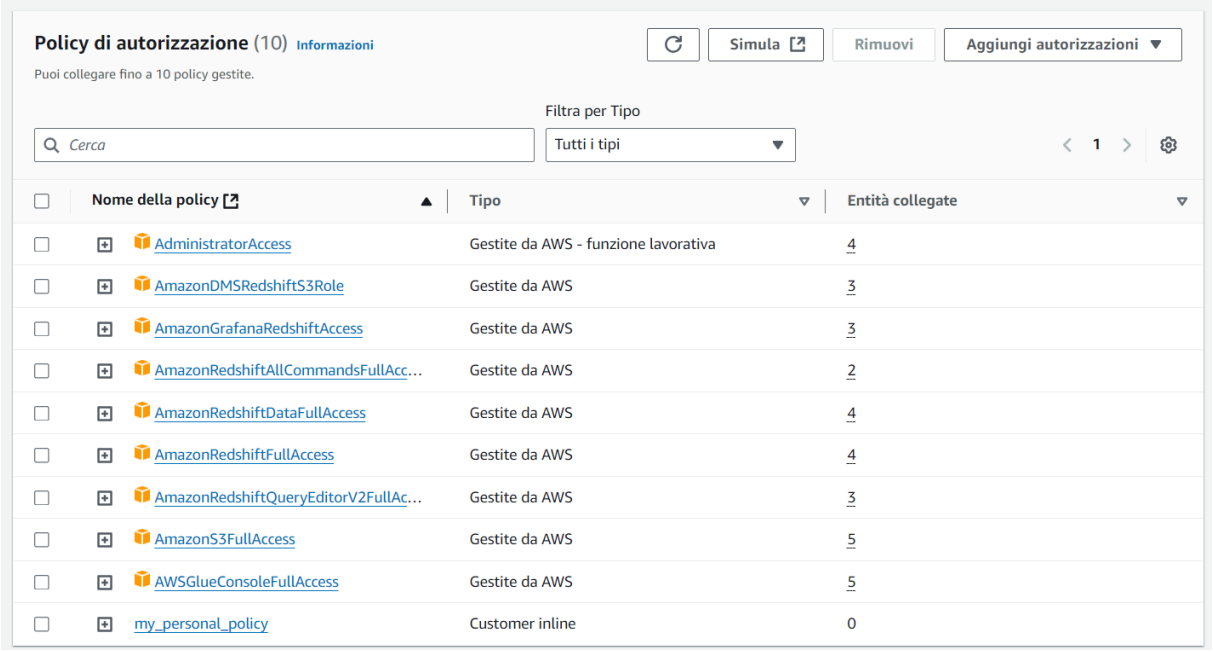




1.2 - Role Configuration.

- Two specific roles were created: one for AWS Glue and one for Amazon Redshift, both with the necessary policies to access S3 and Redshift resources.





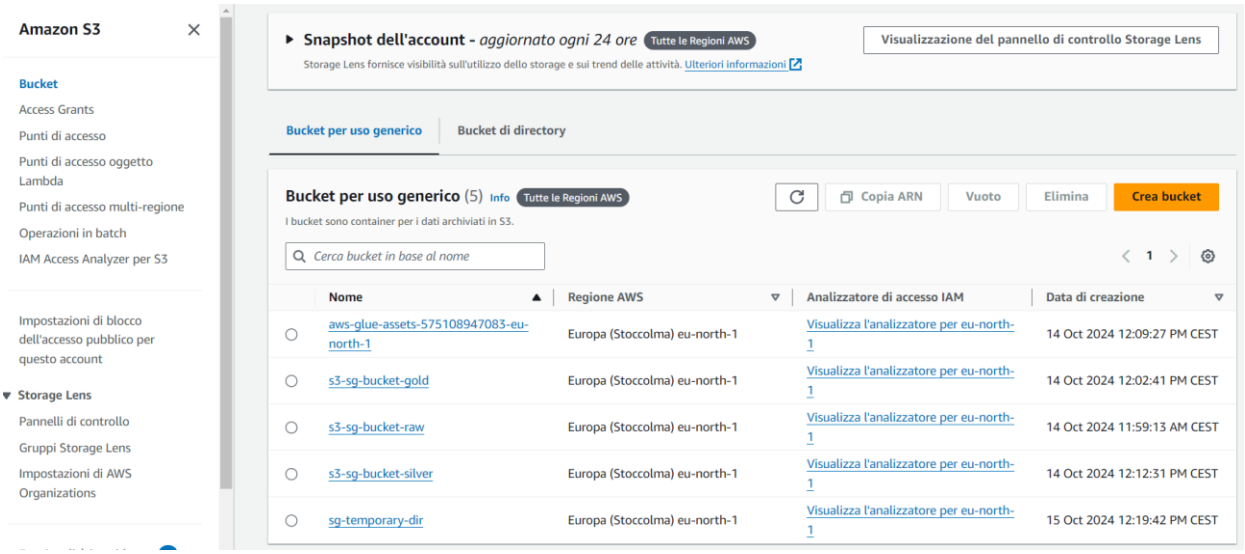
- A custom policy was added to ensure security and appropriate access to resources:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "VisualEditor0",
   "Effect": "Allow",
   "Action": [
    "redshift-serverless:*",
    "s3:*",
    "glue:*"
   ],
   "Resource": "*"
  }
 ]
}
```

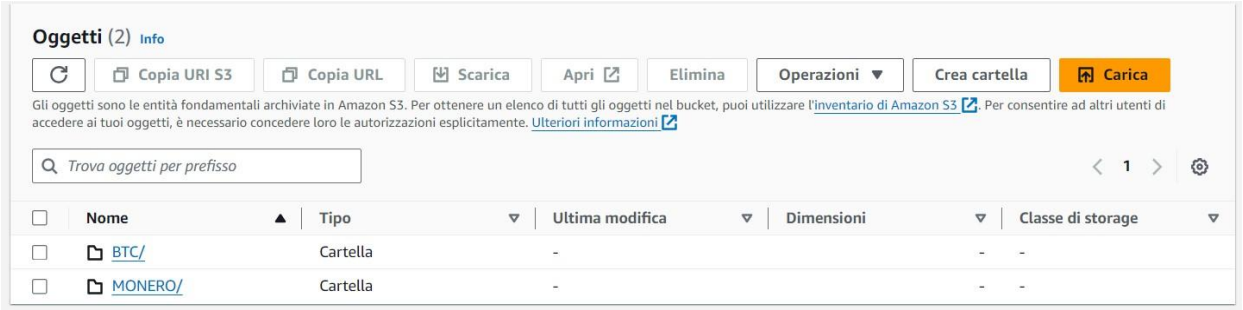
(Redundant but added for safety, as this is my first time interfacing with AWS).

1.3 - Creation of the S3 Buckets.

- Three buckets were created: s3-sg-bucket-raw for raw data, s3-sg-bucket-silver for clean and transformed data, and s3-sg-bucket-gold for finalized data.



- Within each bucket, 'subfolders' were created for BTC and MONERO to organize the data in a structured way.



- Necessary files were manually uploaded into the subfolders of s3-sg-bucket-raw.

2 – Pipeline Implementation with AWS Glue

2.1 - Cleanup and Transformation Scripts.

- Imported and configured the environment with Spark and Glue to process the data.
- Data were read from the S3 bucket in CSV format.
- Converted data to DataFrame to facilitate cleanup and transformation operations, handling missing values by averaging over the last 5 days.
- Saved the cleaned result in Parquet format to s3-sg-bucket-silver.

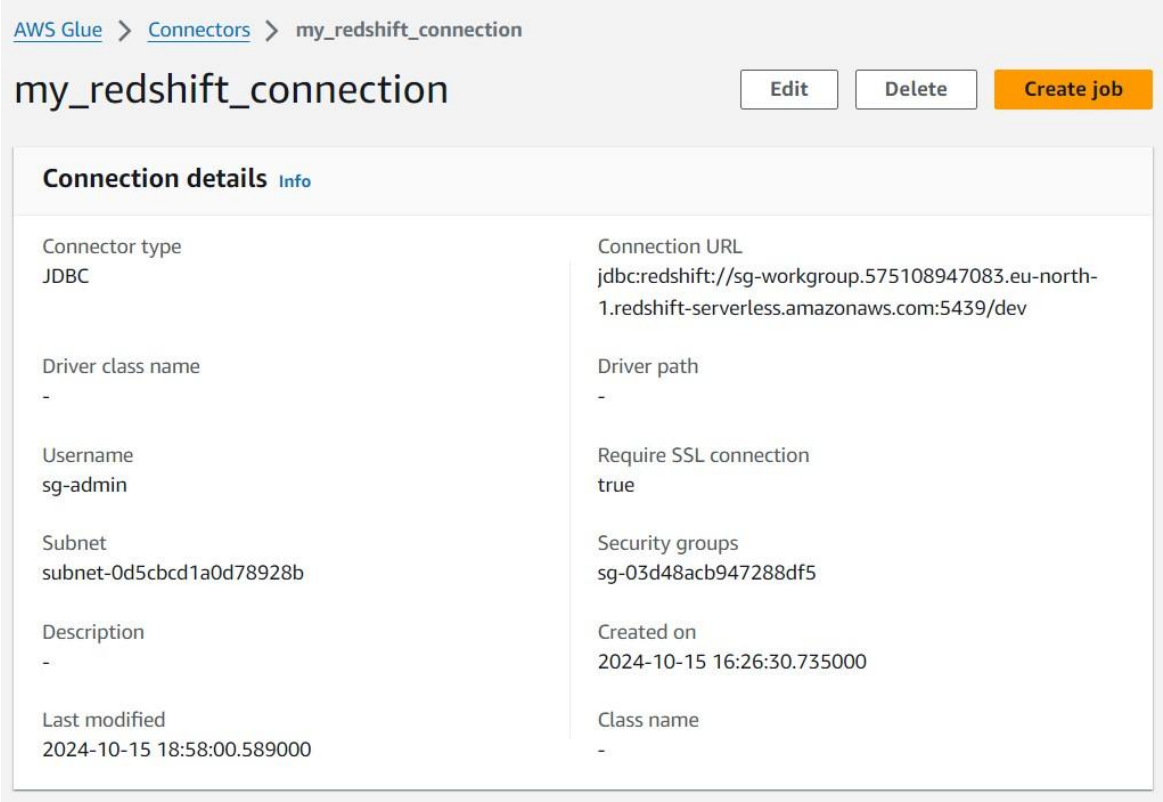
2.2 - Script T2: Calculating Moving Average and Join

- Clean data were read from the s3-sg-bucket-silver bucket.
- The 10-day moving average was calculated to reduce noise in the price data.
- Performed join with Google Trends data to create a unified dataset.
- Saved the final result in Parquet format to s3-sg-bucket-gold.

2.3 - Script L: Loading to Redshift

- Converted and mapped the data for inclusion in the Redshift database, ensuring proper typing.
- Uploaded the final dataset to Redshift Serverless to allow for future analysis.

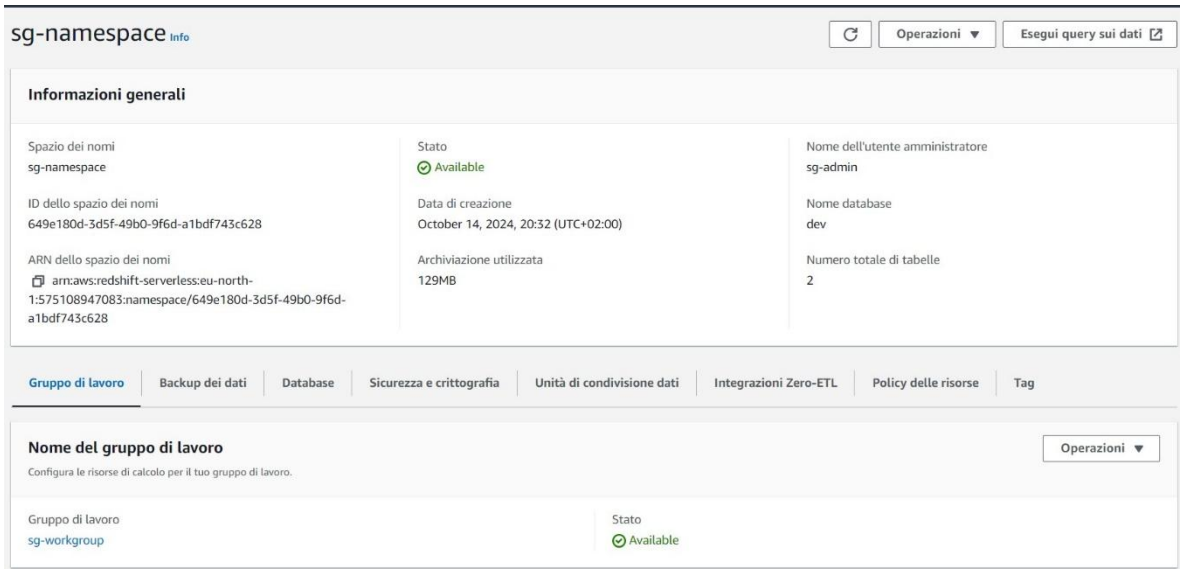
- To make the upload to Redshift, a 'Connection' was created in Glue bringing back Redshift namespace and workgroup configuration data (see on next section).

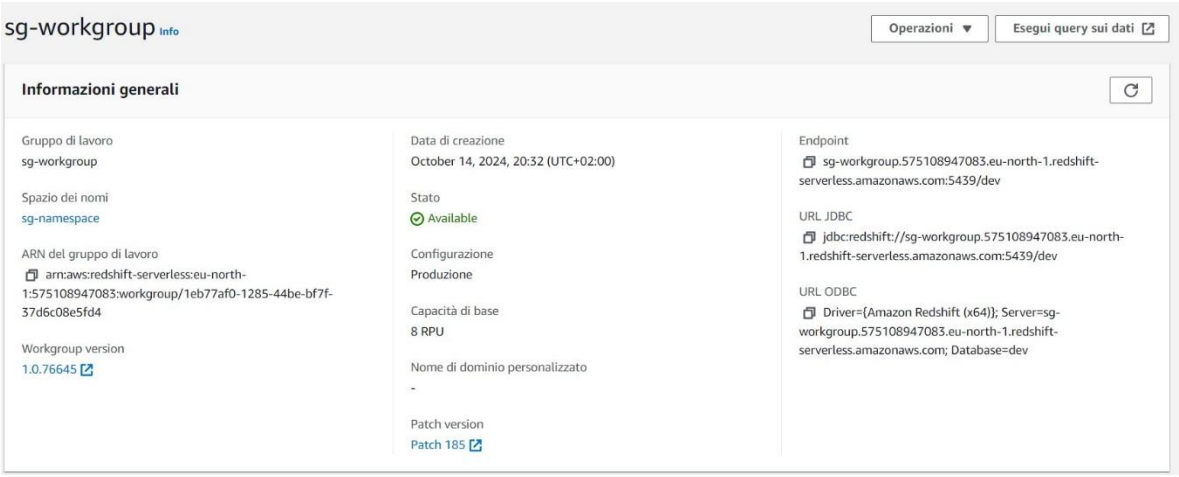


3 - Configuring Amazon Redshift Serverless

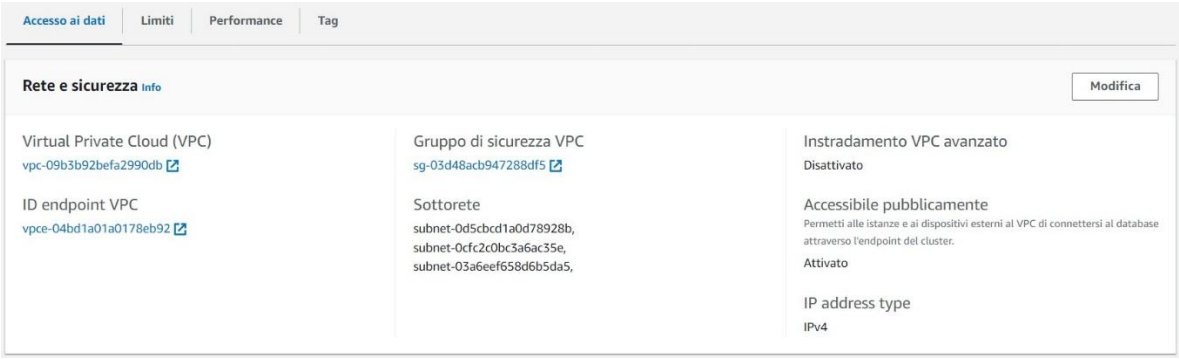
3.1 - Creating the Namespace and Workgroup.

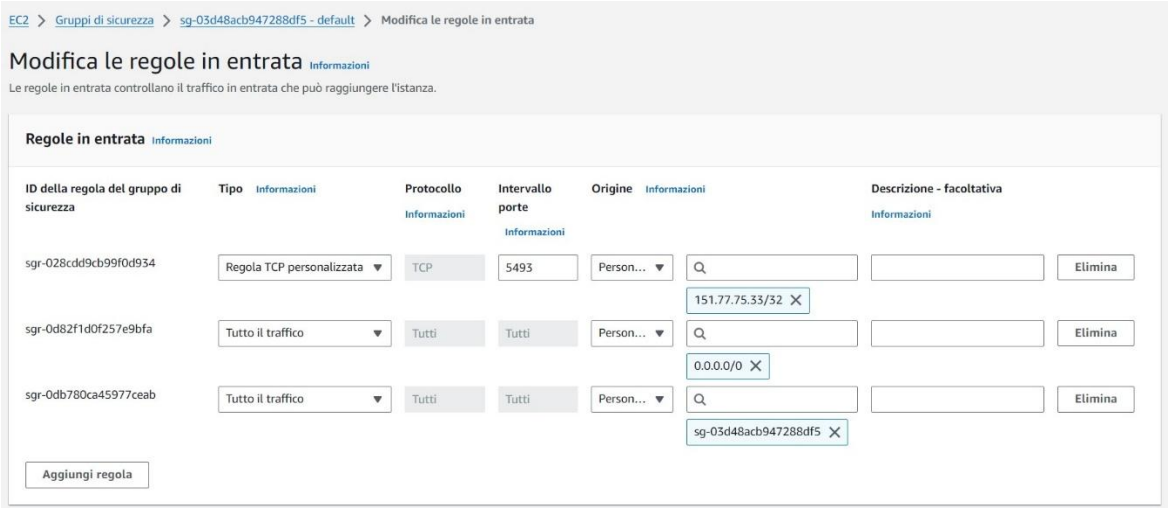
- Configured the namespace and workgroup of Redshift Serverless to manage database resources.





- Security rules have been set up to allow secure access to Redshift resources.





(I added more rules to make sure Glue's 'Connection' worked.)

3.2 - Table Creation

- The tables my_btc and my_monero were created with an optimized structure for analytical queries:

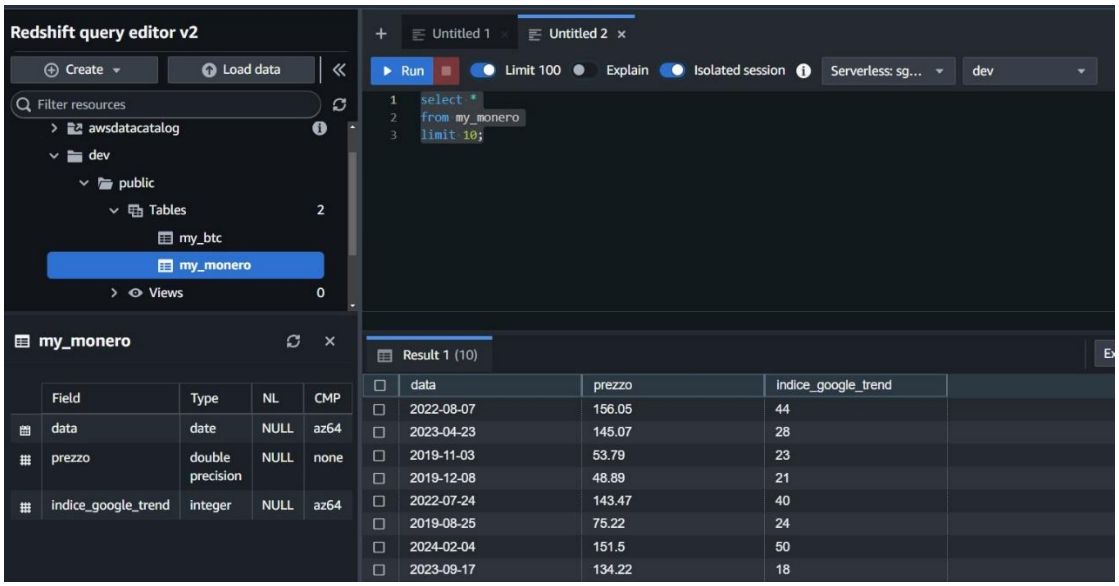
```
CREATE TABLE public.my_btc (
   data date ENCODE az64,
   prezzo double precision ENCODE raw,
   indice_google_trend integer ENCODE az64
) DISTSTYLE AUTO;
```

```
CREATE TABLE public.my_monero (
   data date ENCODE az64,
   prezzo double precision ENCODE raw,
   indice_google_trend integer ENCODE az64
) DISTSTYLE AUTO;
```

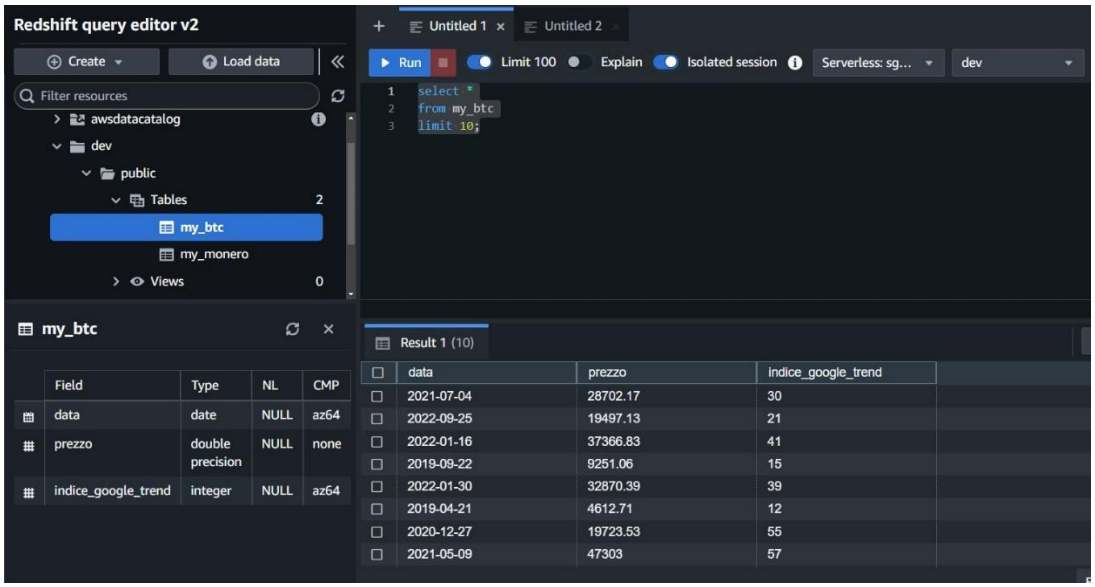
3.3 - Data Verification

- Basic verification queries were performed to ensure that the data were loaded correctly into the Redshift tables.

```
select *
from my_monero
limit 10;
```



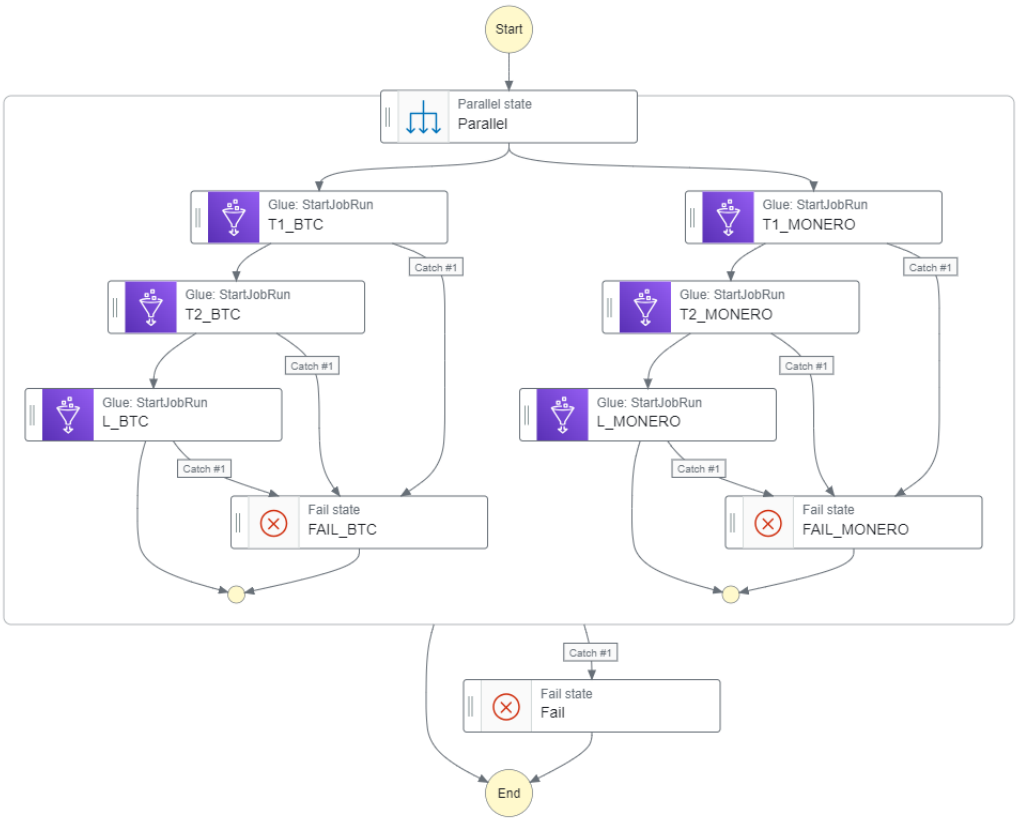

```
select *
from my_btc
limit 10;
```



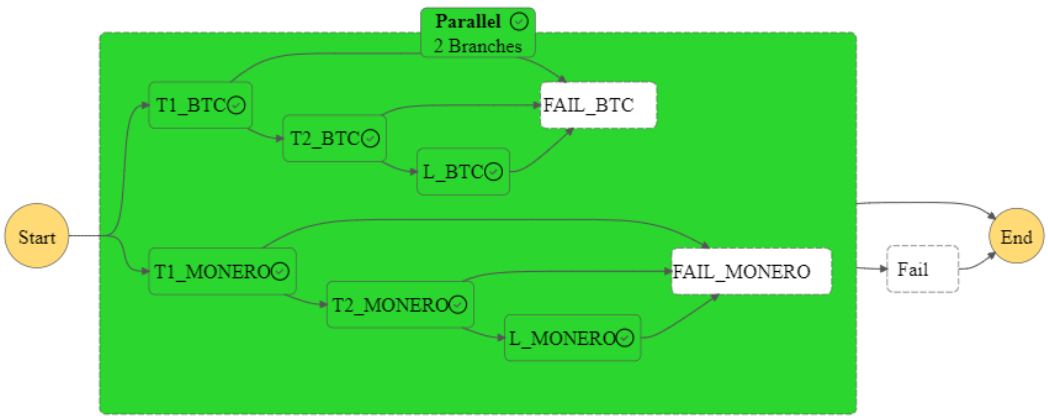
4 – Orchestration Step functions

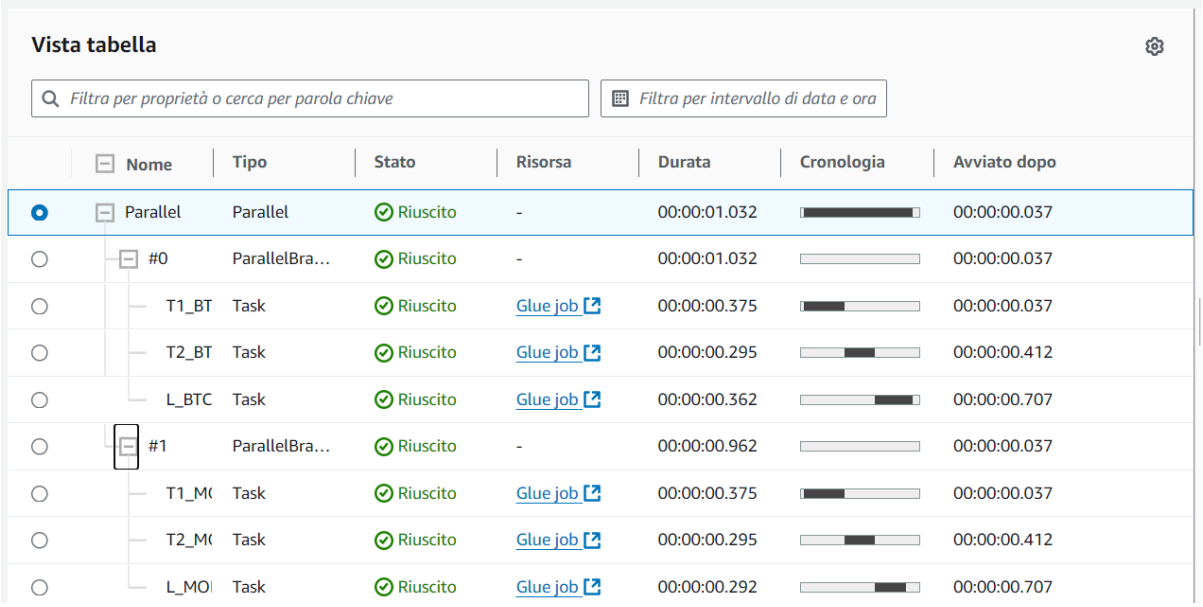
4.1 – Workflow definition.

- A state machine was created using AWS Step Functions to orchestrate the sequential execution of AWS Glue scripts.
- The state definition included steps for the execution of cleanup, transformation and loading scripts.



- Execution has been initiated.





Vista tabella

Filtra per proprietà o cerca per parola chiave | Filtra per intervallo di data e ora

Nome	Tipo	Stato	Risorsa	Durata	Cronologia	Avviato dopo
Parallel	Parallel	Riuscito	-	00:00:01.032		00:00:00.037
#0	ParallelBra...	Riuscito	-	00:00:01.032		00:00:00.037
T1_BT	Task	Riuscito	Glue job	00:00:00.375		00:00:00.037
T2_BT	Task	Riuscito	Glue job	00:00:00.295		00:00:00.412
L_BTC	Task	Riuscito	Glue job	00:00:00.362		00:00:00.707
#1	ParallelBra...	Riuscito	-	00:00:00.962		00:00:00.037
T1_M(	Task	Riuscito	Glue job	00:00:00.375		00:00:00.037
T2_M(	Task	Riuscito	Glue job	00:00:00.295		00:00:00.412
L_MO	Task	Riuscito	Glue job	00:00:00.292		00:00:00.707

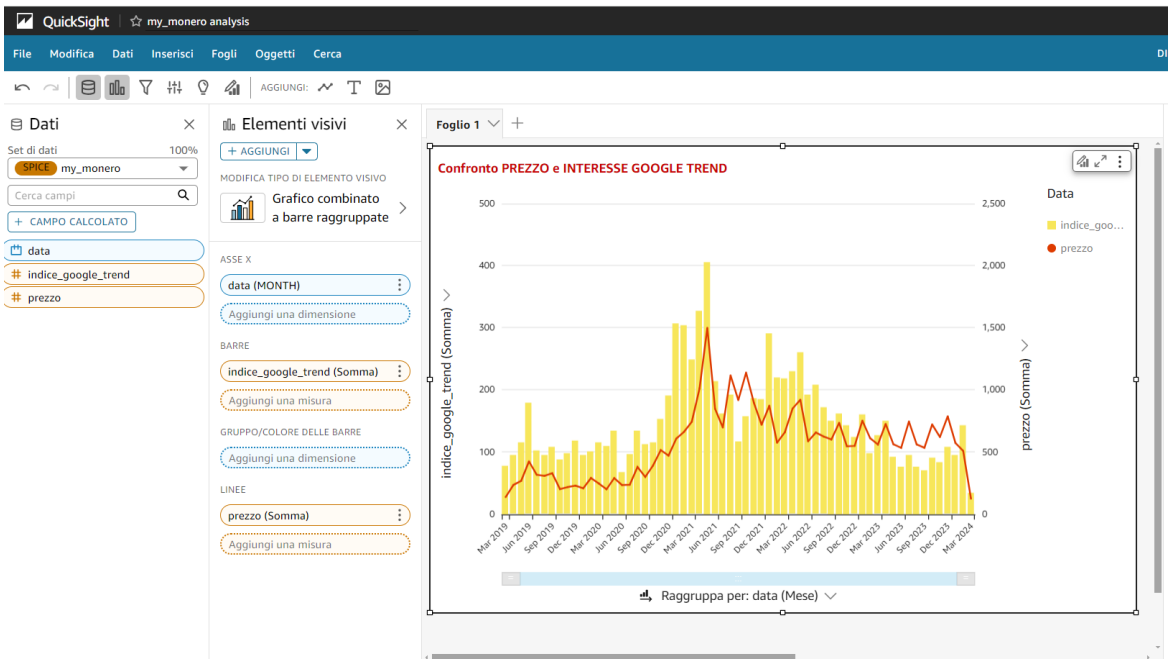
5 - Viewing with Amazon QuickSight

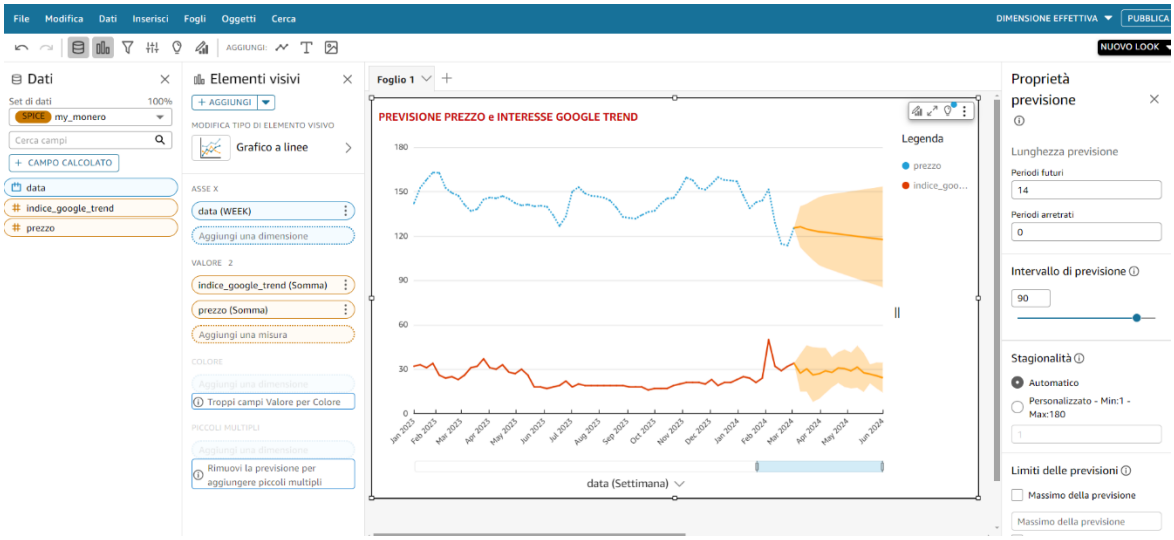
5.1 - Account Configuration

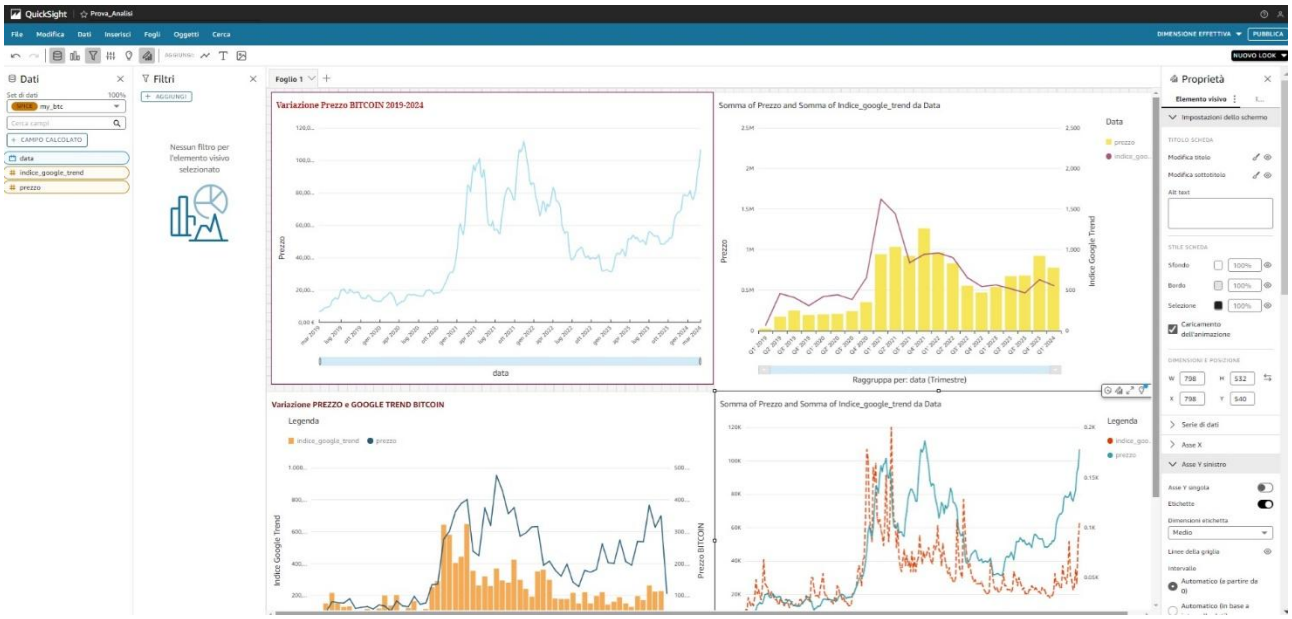
- An Amazon QuickSight instance has been created and configured for visual data analysis.
- A connection to the Redshift database has been established to allow access to the transformed data.

5.2 - Creating Graphs.

- Charts have been created to explore the potential of QuickSight that can help interpret results and make data-driven decisions.




File Modifica Dati Inserisci Fogli Oggetti Cerca
DIMENSIONE EFFETTIVA
PUBBLICA
NUOVO LOOK
Dati
Set di dati
my_monero
CAMPO CALCOLATO
data
indice_google_trend
prezzo
Elementi visivi
AGGIUNGI
MODIFICA TIPO DI ELEMENTO VISIVO
Grafico a linee
ASSE X
data (WEEK)
Aggiungi una dimensione
VALORE 2
indice_google_trend (Somma)
prezzo (Somma)
Aggiungi una misura
COLORE
Troppi campi Valore per Colore
PICCOLI MULTIPLI
Rimuovi la previsione per aggiungere piccoli multipli
Foglio 1
PREVISIONE PREZZO e INTERESSE GOOGLE TREND
Legenda
prezzo
indice_goo...
data (Settimana)
Proprietà previsione
Lunghezza previsione
Periodi futuri
14
Periodi arretrati
0
Intervallo di previsione
90
Stagionalità
Automatico
Personalizzato - Min:1 - Max:180
Limiti delle previsioni
Massimo della previsione


QuickSight
Prova_Analisi
File Modifica Dati Inserisci Fogli Oggetti Cerca
Dati
Filtri
Nessun filtro per l'elemento visivo selezionato
Foglio 1
Variazione Prezzo BITCOIN 2019-2024
Prezzo
data
Somma of Prezzo and Somma of Indice_google_trend da Data
Data
Indice Google Trend
Raggruppa per: data (Trimestre)
Variazione PREZZO e GOOGLE TREND BITCOIN
Legenda
Prezzo BITCOIN
Proprietà