Big Data Platforms – Final Project – Group 6
Stock Screener Dashboard

Objective:

The objective of the project is to create a dashboard for users which will allow them to evaluate stocks based on their fundamental data and historical performance. The dashboard would give the ability to select a stock and view relevant data and performance pertaining to it through pre-compiled reports.

Data Definition:

Universe data
Our analysis and data collection process is restricted to securities in this file.

Column name	Description
symbol	Unique identifier for a company or financial instrument.
name	Name of the company or financial instrument.
price	Current price of the company's stock or financial instrument.
exchange	Name of the exchange where the stock is traded.
exchangeShortName	Abbreviated name of the exchange.
type	Category or type of the company

Company Fundamental data

Column Name	Description
symbol	Unique identifier for a company or financial instrument.
price	Current price of the company's stock or financial instrument.
beta	Measure of the stock's volatility relative to the market.
volAvg	Average trading volume over a specified period.

mktCap	Market capitalization of the company.
lastDiv	Last dividend paid by the company.
range	Price range over a specified period.
changes	Change in price over a specified period.
companyName	Name of the company.
currency	Currency in which the stock is traded.
cik	Central Index Key used to identify securities.
isin	International Securities Identification Number.
cusip	Committee on Uniform Securities Identification Procedures number.
exchange	Name of the exchange where the stock is traded.
exchangeShortName	Abbreviated name of the exchange.
industry	Industry in which the company operates.
website	Company's website URL.
description	Description of the company or financial instrument.
ceo	Name of the company's Chief Executive Officer.

sector	Sector in which the company operates.
country	Country in which the company is headquartered.
fullTimeEmployees	Number of full-time employees.
phone	Contact phone number of the company.
address	Address of the company.
city	City where the company is located.
state	State where the company is located.
zip	Postal code of the company's location.
dcfDiff	Difference between Discounted Cash Flow (DCF) valuation and price.
dcf	Discounted Cash Flow (DCF) valuation.
image	URL of the company's logo or image.
ipoDate	Date of the company's Initial Public Offering (IPO).
defaultImage	URL of a default image for the company.
isEtf	Boolean indicating if the company is an Exchange-Traded Fund.
isActivelyTrading	Boolean indicating if the company's stock is actively traded.

isAdr	Boolean indicating if the company is an American Depositary Receipt (ADR).
isFund	Boolean indicating if the company is a fund.

## Stock price data

Column name	Description
Date	Date of the trading day
Open	Opening price of the stock during the trading day
High	Highest price of the stock during the trading day
Low	Lowest price of the stock on the trading day
Close	Closing price of the stock on the trading day1
Volume	Number of shares traded on the trading day
Dividends	Dividends paid on the trading day
StockSplits	Stock splits that occurred on the trading day
symbol	Symbol of the company (ticker)

Technical Stack:

- AWS cloud services
    - EC2: Elastic Compute Cloud, a web service that provides resizable compute capacity in the cloud.
    - Redshift: A fully managed cloud based data warehousing service
    - S3: A cloud based object storage service

- Python: A high level programming language

- Tableau: A data visualization software used for dashboarding

Architecture:
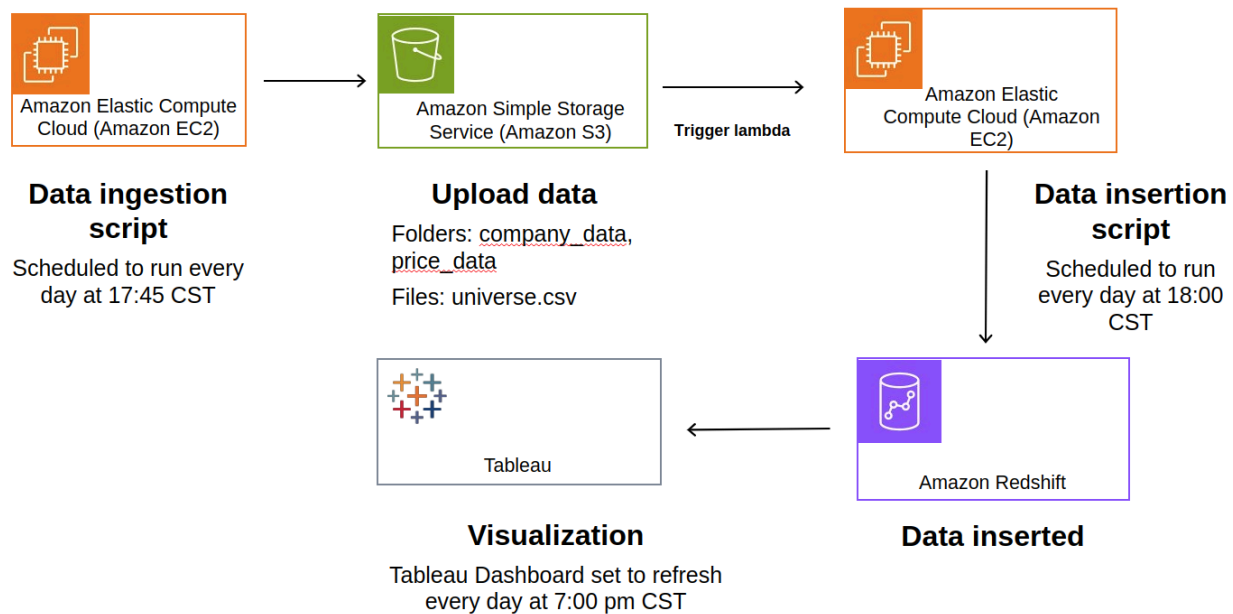
Daily Process:

1. The ingestion script runs at 17:45 CST every day and creates a folder with the current date and uploads the daily data into s3.
2. Post this run, another cron job script called insertion script is executed at 18:00 to insert that s3 data into the redshift table.
3. The tables are connected to the Tableau dashboard and it's scheduled to refresh at 19:00 everyday.

The below image shows Cron job schedule to run the data ingestion and the data insertion scripts and stores the logs with timestamps in the logs folder.

Scripts

Data Ingestion Script (Click on the link for the code)

1. Created a Python script that fetches company profile data and price history data for a list of symbols from Financial Modeling Prep API and Yahoo Finance API respectively.
2. It then saves the data into CSV files, organizes them into directories based on the date, and uploads the price data to an S3 bucket using AWS CLI.

We have identified a few companies from S&P 500 to get the data for each day.

Initial load:

Python Script: **Initial Load** (Change directory at the top before running)
Files: **Universe**, **Company data**, **Price**
NOTE: These csv files can be used directly for initial load)

The **full_load** folder contains the initial load data consisting of company information, stock universe data and 3 months stock data. The full load folder has 3 files: company_data.csv, stock_daily_data.csv and universe.csv. Schema of all these files are described in the previous section.

Contents of full_load folder:

Daily load:

Date	Open	High	Low	Close	Volume	Dividends	Stock	Splits symbol
2024-04-22	16.6900005340576	16.7299995422363	16.1700000762939	16.3099994659424	54105319	0	0	T
2024-04-22	129.979995727539	133.949996948242	129.535003662109	131.990005493164	3148359	0	0	RCL
2024-04-22	35.2999992370605	35.4099998474121	34.5400009155273	34.7599983215332	6593328	0	0	IP
2024-04-22	16.9400005340576	16.9799995422363	16.6499996185303	16.8299999237061	8298042	0	0	HPE
2024-04-22	113.099998474121	113.639999389648	111.75	111.98999786377	5253207	0	0	DIS
2024-04-22	159.949996948242	162.695007324219	158.440307617188	161.919998168945	4806020	0	0	CVX
2024-04-22	119.26000213623	121.565002441406	118.430000305176	120.559997558594	16071616	0	0	XOM
2024-04-22	616550	619765	611521	617284	18062	0	0	BRK-A
2024-04-22	406.839996337891	411.350006103516	406.25	408.779998779297	2175161	0	0	BRK-B
2024-04-22	60.0499992370605	60.5699996948242	58.2200012207031	59.6100006103516	1343100	0	0	CAVA
2024-04-22	3.79999995231628	3.9449999332428	3.63499999046326	3.8199999332428	6455212	0	0	NOVA
2024-04-22	0.557099997997284	0.595600008964539	0.529999971389771	0.552299976348877	2592652	0	0	AMWL
2024-04-22	25.0100002288818	28.4270992279053	24	24.6900005340576	727978	0	0	DXYZ
2024-04-22	2897.75	2911.57495117187	2867.02001953125	2884.25	247421	0	0	CMG
2024-04-22	1742.80004882813	1750	1730	1742	14247	0	0	WTM
2024-04-22	1.22000002861023	1.25999999046326	1.16499996185303	1.25999999046326	1280651	0	0	DOUG
2024-04-22	54.4000015258789	55.5200004577637	52.939998626709	54.7299995422363	4075250	0	0	W
2024-04-22	23.7800006866455	23.7800006866455	23	23.0400009155273	317205	0	0	PACS
2024-04-22	1.54999995231628	1.60000002384186	1.55999994277954	1.60000002384186	8583	0	0	LANV
2024-04-22	7710	7825	7647.02490234375	7806.7900390625	17823	0	0	NVR
2024-04-22	16.3700008392334	16.7450008392334	16.2999992370605	16.4200000762939	26714716	0	0	GOLD
2024-04-22	75.5	76.3399963378906	75.25	75.5800018310547	2446399	0	0	CNC
2024-04-22	2.10999989509583	2.10500001907349	2.05469989776611	2.0699999332428	83695	0	0	COOK
2024-04-22	89.1399993896484	91.4700012207031	88.129997253418	91.2300033569336	1864863	0	0	VLTO
2024-04-22	9.38000011444092	9.65999984741211	9.32499980926514	9.63000011444092	2787257	0	0	ZUO
2024-04-22	35.7299995422363	36.6399993896484	35.25	36.3499984741211	6215828	0	0	AA
2024-04-22	48.2200012207031	49.4099998474121	47.7999992370606	48.9500007629395	20121085	0	0	FCX

The above picture depicts how the daily data is downloaded and then dumped into s3. The folder structure is by date. The folder full_load contains the initial data. The daily load script creates a directory for a date if it does not exist and then downloads the data into that directory.

[**Data insertion script**](#) (Click on the link for the code)

This script is also stored in s3 and runs post the ingestion script. It connects to an Amazon Redshift database and copies data from a CSV file stored in an Amazon S3 bucket into a table named daily_data. It generates today's date, constructs the S3 file path based on the date, establishes a connection to Redshift, executes a SQL COPY command to load the CSV data into the daily_data table.

Before running the data ingestion and insertion scripts, let us configure the required AWS services.

AWS Services creation

EC2 instance

Steps to create EC2 instance:

1. Launch Instance: In the EC2 dashboard, click on the "Launch Instance" button to start the instance creation process.
2. Choose an Amazon Machine Image (AMI): Select a Linux AMI from the provided list. Choose Instance Type: In the "Choose Instance Type" step, select the t2.micro instance type from the list.

Number of instances | **Info**

```
1
```

Software Image (AMI)
Amazon Linux 2023 AMI 2023.4.2...read more
ami-07caf09b362be10b8

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

3. Configure Instance: Configure instance details such as network settings, subnet, IAM role (in our case LabRole), etc.

Instance ID
🗐 i-02aded4d9eaf51d0d (team6-bdi)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

```
EMR_EC2_DefaultRole                    ▼
```
🔄 Create new IAM role ☑

4. Add Storage: By default, a small EBS volume is attached to the instance.
5. Configure Security Group: Specify security group settings, including inbound and outbound rules for network traffic to and from your instance. Ensure that you allow SSH (port 22) access if you plan to connect to a Linux instance via SSH.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-055efdb88f2dd9dec	SSH ▼	TCP	22	C... ▼ 0.0.0.0/0 ✕		Delete
sgr-05cbd3be75fbe0bc6	Custom TCP ▼	TCP	0	C... ▼ 0.0.0.0/0 ✕		Delete
sgr-0611a5d5253b13aea	All traffic ▼	All	All	C... ▼ sg-0c0fcd5c760c69bc2 ✕		Delete

6. Create a Key Pair (pem file)
7. Launch Instance: After selecting your key pair, click on the "Launch Instances" button to launch your t2.micro EC2 instance.
8. View Instances: Once your instance is launched, you can view its status and details in the EC2 dashboard under "Instances".

Description of instance

Instances (1) Info		↻	Connect	Instance state ▼	Actions ▼	Launch instances ▼
Q Find Instance by attribute or tag (case-sensitive)				All states ▼		‹ 1 › ⚙

☐	Name ✎ ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm stat
☐	team6-bdi	i-02aded4d9eaf51d0d	⊘ Running ⊕ ⊖	t2.micro	⊘ 2/2 checks passed	View alarm

SSH into the EC2 instance:

Click on connect in the instance:

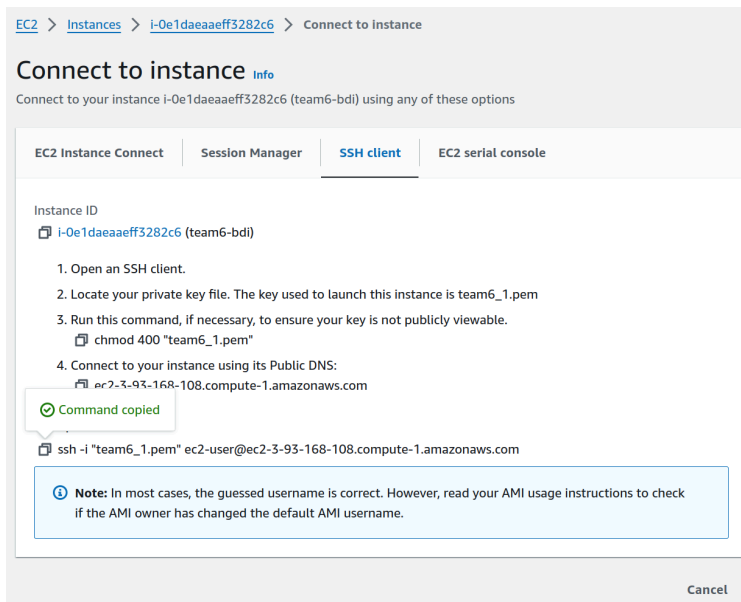| ↻	Connect	Instance state ▼	Actions ▼

Private IPv4 addresses
□ 172.31.22.115

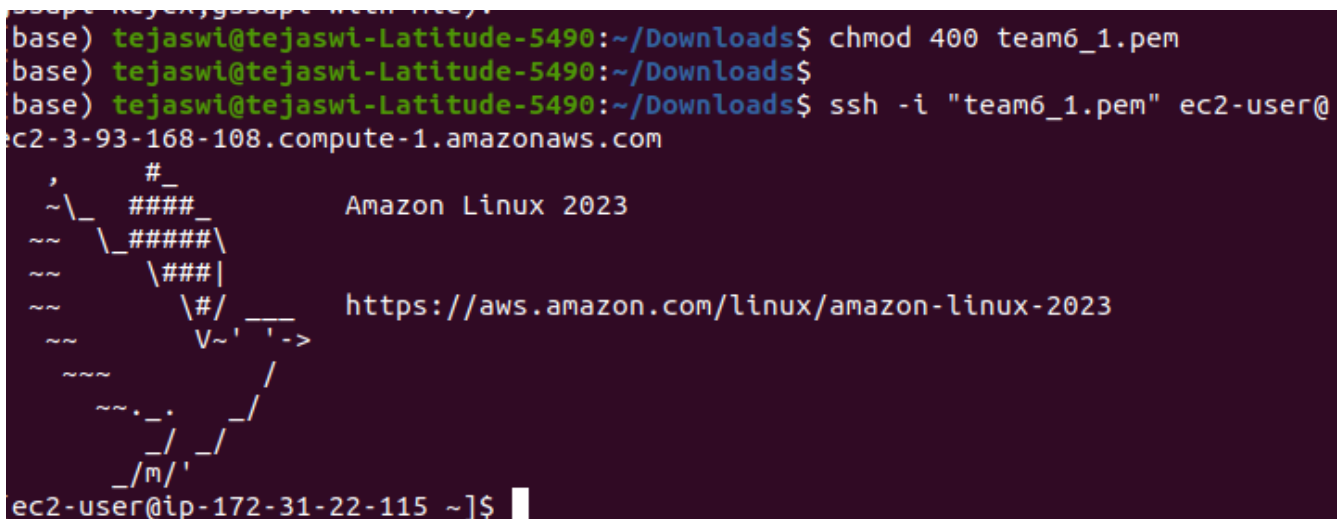Public IPv4 DNS
□ ec2-3-93-168-108.compute-1.amazonaws.com |
open address ↗

Since this is a linux instance, we can connect via ssh client.

copy the above command and paste it in your local terminal (linux), and for windows you can ssh into the instance via MobaXterm.

Before that, you have to change the permissions of your pem file using chmod as shown in the image below.

Install python, pip and crontab in the instance as shown in the screenshots below.

```
[ec2-user@ip-172-31-22-115 ~]$ sudo yum update

Last metadata expiration check: 0:56:31 ago on Mon May  6 02:36:54 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-22-115 ~]$ sudo yum install python3-pip
Last metadata expiration check: 0:56:40 ago on Mon May  6 02:36:54 2024.
Dependencies resolved.
```

```
[ec2-user@ip-172-31-22-115 ~]$ sudo yum install cronie
Last metadata expiration check: 0:56:53 ago on Mon May  6 02:36:54 2024.
Dependencies resolved.
==================================================================================
 Package              Arch         Version                    Repository      Size
==================================================================================
Installing:
 cronie               x86_64       1.5.7-1.amzn2023.0.2        amazonlinux     115 k
Installing dependencies:
```

To create python files, use scp command to copy files from local to ec2 instance.

```
scp -i team6.pem ec2-user@3.86.222.241:/home/ec2-user/tejaswi/project/daily_data.py /home/tejaswi/Documents/Sem2/BDI/daily_data_1.py
ssh -i "team6.pem" ec2-user@ec2-54-147-58-199.compute-1.amazonaws.com
```

S3:

Steps to create an S3 bucket:
1. Create a Bucket: In the S3 dashboard, click the "Create bucket" button.
2. Configure the Bucket: You will need to provide some basic information to configure your bucket:
   - Bucket Name: Choose a unique name for your bucket. This name must be globally unique across all existing bucket names in AWS.
3. Set Permissions: By default, buckets and the objects within them are private. Grant permissions to other AWS accounts or make the bucket publicly accessible.

General purpose buckets	Directory buckets

General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

[🔄] [🗐 Copy ARN] [Empty] [Delete] [**Create bucket**]

🔍 Find buckets by name   ‹ 1 › ⚙

	Name ▲	AWS Region ▽	IAM Access Analyzer	Creation date ▽
○	team6-bdi	US East (N. Virginia) us-east-1	View analyzer for us-east-1	April 21, 2024, 19:57:01 (UTC-05:00)

Redshift:

Created a cluster team6-bdi:

1. Create Cluster: In the Redshift dashboard, click on "Create cluster" to start the cluster creation process.
2. Cluster Configuration:
   - Cluster Identifier: Enter a unique name for your Redshift cluster (in our case team6-bdi)
   - Node Configuration: Choose "Single Node" for the cluster type.
   - Node Type: Select "dc2.large" from the dropdown list for a single dc2.large node.

Node type Info

Choose a node type that meets your CPU, RAM, storage capacity, and drive type requirements.

dc2.large ▼

Number of nodes

Enter the number of nodes that you need.

1

Range (1-32)

   - Cluster Permissions: In the "Network and security" section, you can choose to create a new Virtual Private Cloud (VPC) or use an existing one. Ensure that the cluster is publicly accessible by selecting "Yes" for "Publicly accessible".

3. Additional Configuration:
   - Set the admin username and manually add the admin password.

Database configurations

Admin user name
Enter a login ID for the admin user of your DB instance.

awsuser

The name must be 1-128 alphanumeric characters, and it can't be a reserved word ☑.

Admin password
Select an option to manage your admin password.

○ Manage admin credentials in AWS Secrets Manager Info
   AWS manages a KMS key that encrypts your data.
○ Generate a password
   Amazon Redshift generates an admin password.
● Manually add the admin password
   Manually enter the admin password.

Admin user password

- VPC Security Group: Configure the security group to control inbound and outbound traffic to your cluster. Ensure that appropriate ports are open for your use case. For public accessibility, the Redshift cluster needs to be associated with a security group that allows inbound traffic on port 5439 (Redshift default port) from your IP or from anywhere.

> ⓘ You can't change the VPC associated with this cluster after the cluster has been created. Learn more about getting started cluster in vpc ↗
> ✕

VPC security groups
This VPC security group defines which subnets and IP ranges the cluster can use in the VPC. For more information, see Learn more about Redshift clusters security groups ↗

Choose one or more security groups ▼

default ✕
sg-0673eb11915f263e2

Cluster subnet group Info
Choose the Amazon Redshift subnet group to launch the cluster in.

default ▼ | Create new subnet group | ↻

Availability Zone
Specify the Availability Zone to create the cluster in. Otherwise, Amazon Redshift chooses an Availability Zone for you.

No preference ▼

Enhanced VPC routing
Enabling this option routes network traffic between your cluster and data repositories through a VPC, instead of through the internet. Learn more about getting started cluster in vpc ↗

◉ Turn off
○ Turn on

Publicly accessible
For more information, see Learn more about Redshift clusters security groups ↗
☑ Turn on Publicly accessible
 Allow public connections to Amazon Redshift.

Elastic IP address
Select the Elastic IP address for connecting to the cluster.

▼

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-055efdb88f2dd9dec	SSH ▼	TCP	22	C... ▼ 🔍		Delete
				0.0.0.0/0 ✕		
sgr-05cbd3be75fbe0bc6	Custom TCP ▼	TCP	0	C... ▼ 🔍		Delete
				0.0.0.0/0 ✕		
sgr-	All traffic ▼	All	All	C... ▼ 🔍		Del

- IAM Roles : Associate LabRole that has the necessary policies in place

Associated IAM roles (1) Info
`[Set default ▼]` `[Manage IAM roles ▼]`

Create, associate, or remove an IAM role. You can associate up to 50 IAM roles. You can also choose an IAM role and set it as the default for this cluster.

| 🔍 Search for associated IAM role by name, status, or role type | | ‹ 1 › |

☐	IAM roles 🔗 ▽	Status ▽	Role type ▽
☐	LabRole	Not applied	--

4. Create Cluster: Review the configuration settings and then click "Create Cluster" to create the Redshift cluster
5. Wait for Cluster Creation: It may take several minutes for your cluster to be created. You can monitor the progress in the Redshift console.
6. Connect to Cluster: Once the cluster status changes to "Available", you can obtain the connection details such as the cluster endpoint, database name, and credentials from the cluster details page in the console.

team6-bdi

`[Actions ▼]` `[Edit]` `[Add partner integration]` `[Query data ▼]`

General information Info `[↻]`

| Cluster identifier	Status	Node type	Endpoint
team6-bdi	⊘ Available	dc2.large	🗗 team6-bdi.cwswinazre0j.us-east-1.redshift.amazonaws.com:5439/dev
Custom domain name	Date created	Number of nodes	
-	May 02, 2024, 16:06 (UTC-05:00)	1	JDBC URL
			🗗 jdbc:redshift://team6-bdi.cwswinazre0j.us-east-1.redshift.amazonaws.com:5439/dev
Cluster namespace ARN	Storage used		
🗗 arn:aws:redshift:us-east-1:905418465574:namespace:fb01fcb1-8afd-4f09-affa-de53d4d951bb	0.02% (0.03 of 160 GB used)		ODBC URL
	Multi-AZ		🗗 Driver={Amazon Redshift (x64)}; Server=team6-bdi.cwswinazre0j.us-east-1.redshift.amazonaws.com; Database=dev
Cluster configuration	No		
Production			

Cluster performance | Query monitoring | Zero-ETL integrations | Resource Policy | Schedules | Maintenance | Properties
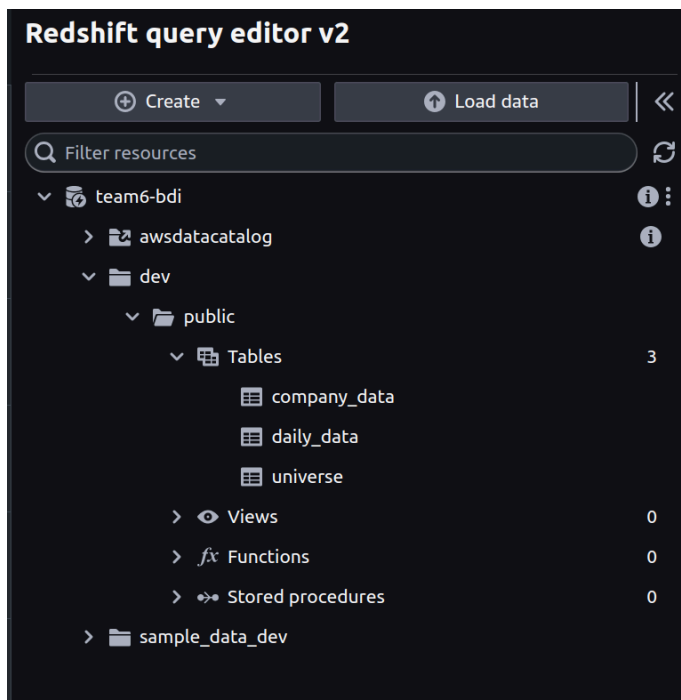
Create tables in redshift:

Open the query editor v2 and run the creation commands to create the three required tables.

[Table creation commands](#)
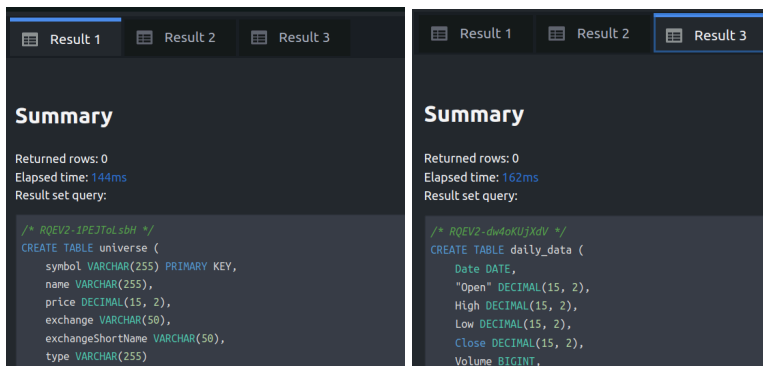
Database:
After running the create command, as seen in the image below, the database should now have 3 tables: company_data, universe, daily_data.

Initial loads:

company_data:

```
COPY dev.public.company_data (symbol, price, beta, volavg, mktcap, lastdiv, range, changes, companyname, currency, cik, isin, cusip, exchange, exchangeshortname, industry, website, description, ceo, sector, country, fulltimeemployees, phone, address, city, state, zip, dcfdiff, dcf, image, ipodate, defaultimage, isetf, isactivelytrading, isadr, isfund) FROM 's3://team-06-bdi/processed/company_data.csv' IAM_ROLE 'arn:aws:iam::992382855900:role/LabRole' FORMAT AS CSV DELIMITER ',' QUOTE '"' IGNOREHEADER 1 REGION AS 'us-east-1'
```

price_data

```
COPY dev.public.daily_data FROM 's3://team-06-bdi/processed/price_daily.csv' IAM_ROLE 'arn:aws:iam::992382855900:role/LabRole' FORMAT AS CSV DELIMITER ',' QUOTE '"' IGNOREHEADER 1 REGION AS 'us-east-1'
```

Result 1

Summary

ℹ Info:
- Load into table 'daily_data' completed, 100 record(s) loaded successfully.

Returned rows: 0
Query ID: 2344
Elapsed time: 1.4s
Result set query:

universe:

```
COPY dev.public.universe FROM 's3://team-06-bdi/processed/universe.csv' IAM_ROLE 'arn:aws:iam::992382855900:role/LabRole' FORMAT AS CSV DELIMITER ',' QUOTE '"' IGNOREHEADER 1 REGION AS 'us-east-1'
```

Result 1

Summary

ℹ Info:
- Load into table 'universe' completed, 100 record(s) loaded successfully.

Returned rows: 0
Query ID: 2368
Elapsed time: 1.2s
Result set query:

Querying price data, the latest data is from 2nd may as our AWS account credits expired.

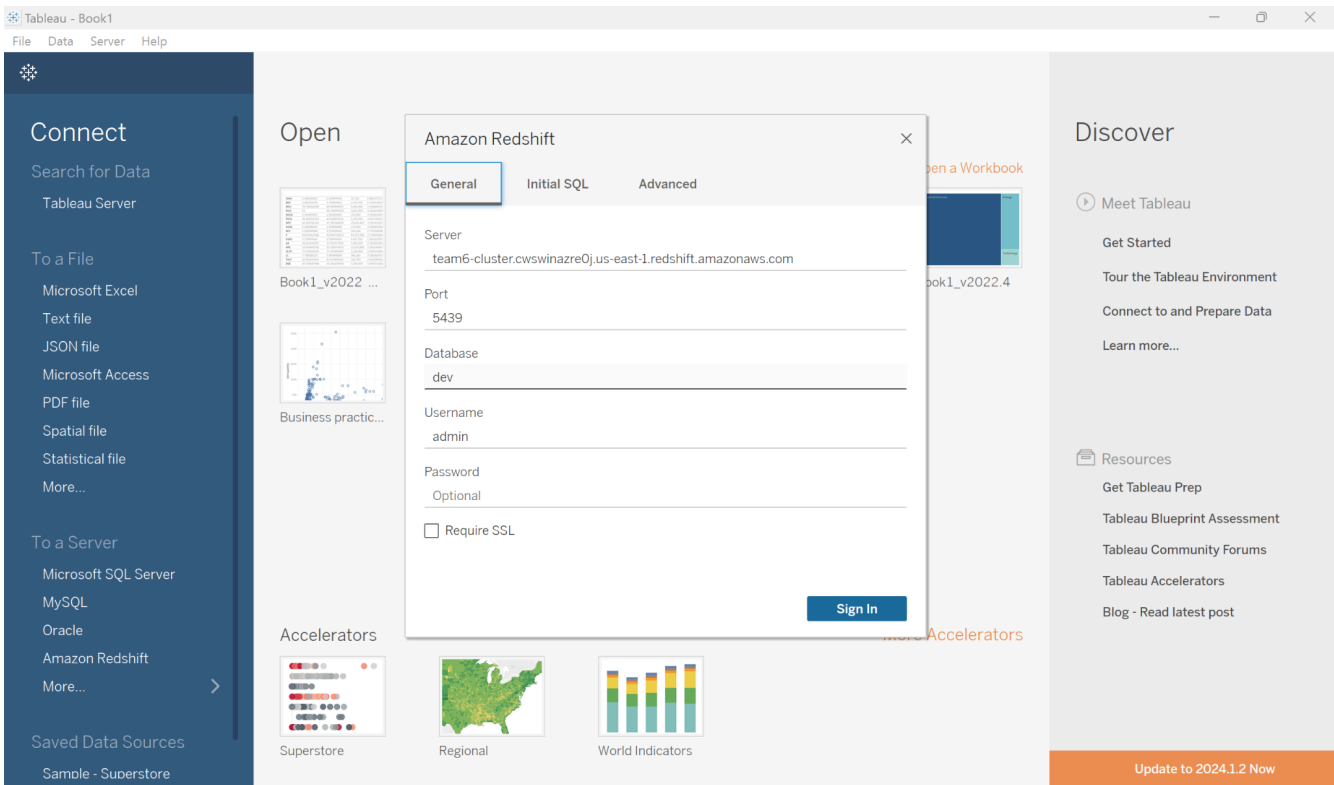|	date	open	high	low
☐	2024-05-02	16.98	16.98	16.72
☐	2024-05-02	139.11	139.11	136.32
☐	2024-05-02	35.5	36.07	35.12
☐	2024-05-02	16.61	16.64	16.26
☐	2024-05-02	111.43	112.87	110.55
☐	2024-05-02	160.41	161.77	159.89
☐	2024-05-02	116.26	117.26	115.9
☐	2024-05-02	609950	609950	600586.37

Visualizations:

Connecting to Amazon Redshift Tables in Tableau

To access data stored in Amazon Redshift and visualize it in Tableau, follow these steps:

1. Server: Enter the server address where your Amazon Redshift cluster is hosted.
2. Port: Specify the port number through which Tableau will connect to the Amazon Redshift cluster.
3. Database Name: Provide the name of the database in Amazon Redshift containing the tables you want to access.
4. Username: Input the username associated with your Amazon Redshift account.
5. Password: Enter the password corresponding to the provided username for authentication.

Ensure that the credentials provided are accurate and have the necessary permissions to access the desired tables.
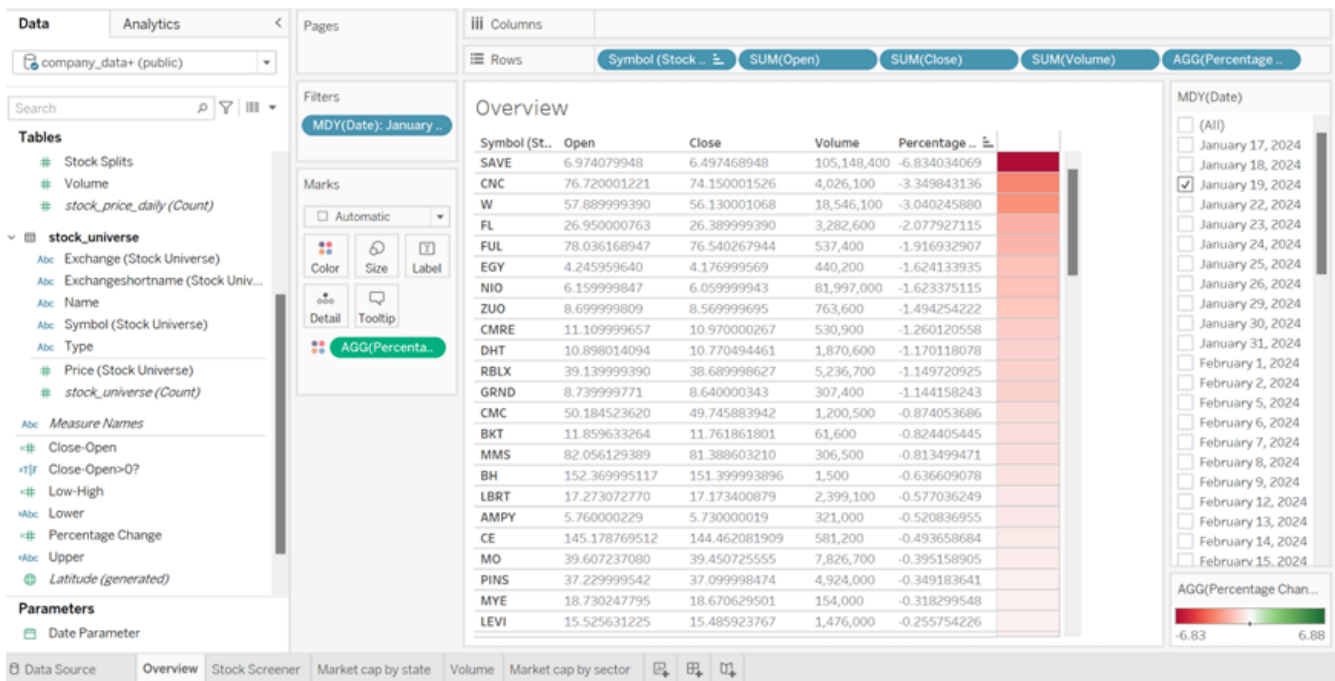
Overview Tab
The landing page of the dashboard, which displays a table of stock tickers and its details over time like open/close price, volume traded, and percentage change. It also provides a visual aid in the percentage change by color (Red if percentage change is negative, and Green if percentage change is positive) to provide a quick overview for the users. The users can sort and filter the table based on time and percentage change for some quick insights, as shown in the screenshots below.
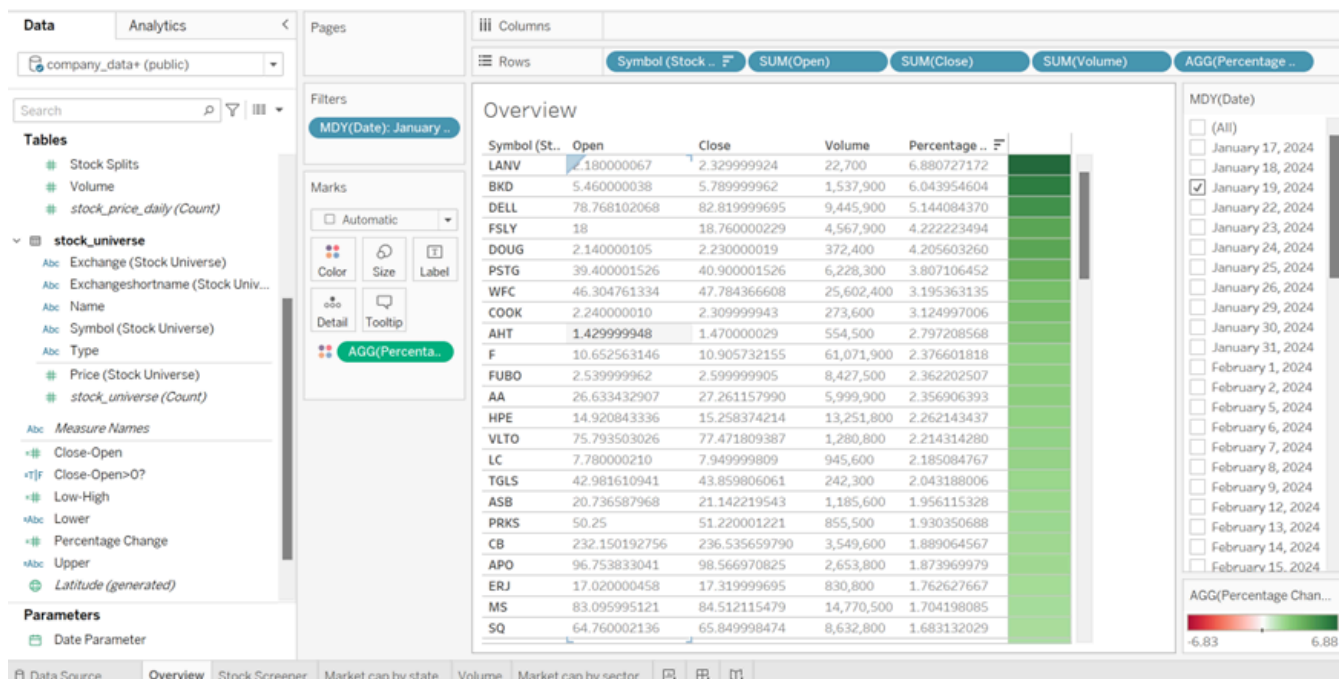
Tables and fields used-
- universe (Symbol)
- price_daily (Open, Close, Volume)
- Custom measure(s) - Percentage Change (calculated from open and close)

Overview of stocks on January 19th 2024 sorted by descending negative percentage change

Overview of stocks on January 19th 2024 sorted by descending positive percentage change


Stock Screener Tab
This tab provides a candlestick chart of stocks over time to analyze the trend patterns. It also provides information on the stock's change of price over a trading day. Users can filter a stock from the Companyname filter on the right side, and the chart will dynamically update accordingly. This tab can be used to help the user make trading decisions.

Tables and Fields Used-
- company_data (Companyname)

- stock_price_daily (Date, Close, Open, High, Low)
- Custom measure(s) -
  - Close - Open (Difference of Close and Open - Used for the candlesticks)
  - Close - Open > 0 (Used for colour coding the candlesticks by flagging green if the difference between Close and Open is positive, and red if the difference is negative)
  - Low-High (Difference of Low and High (Used to display the range of the stock price change over a trading day, represented by gray vertical lines on the candlestick)
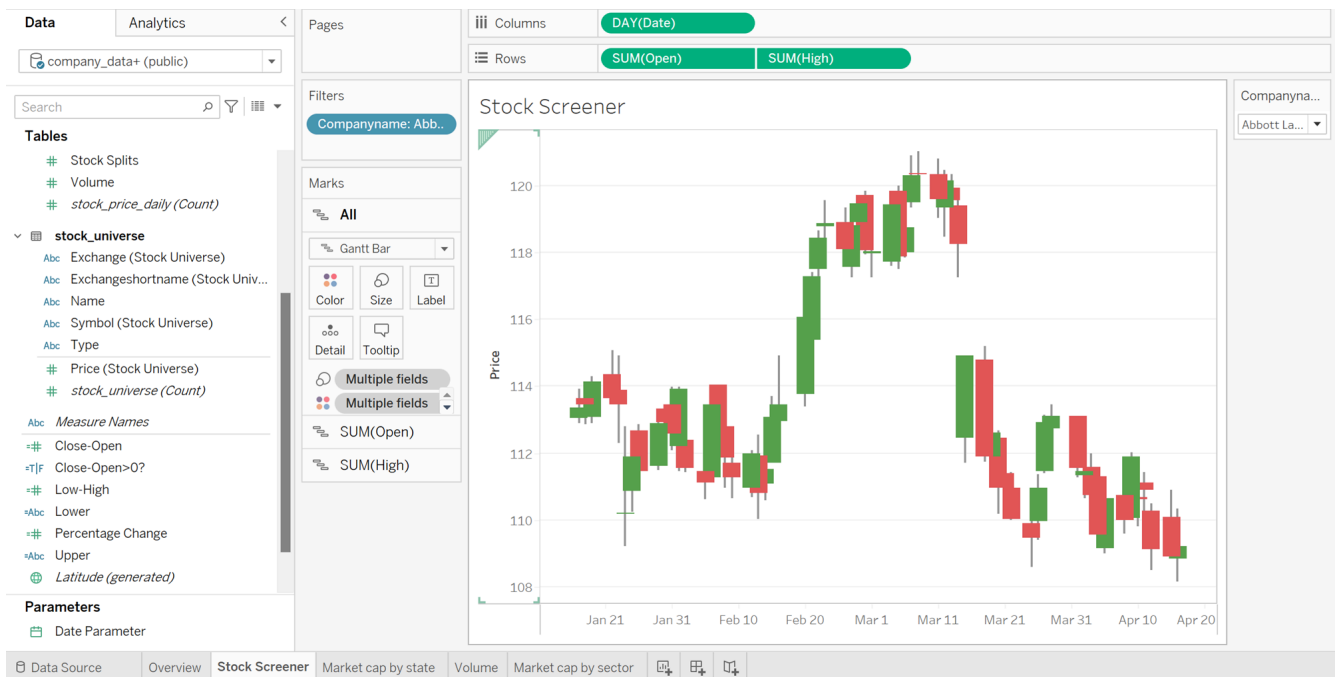

Market Capitalization by State
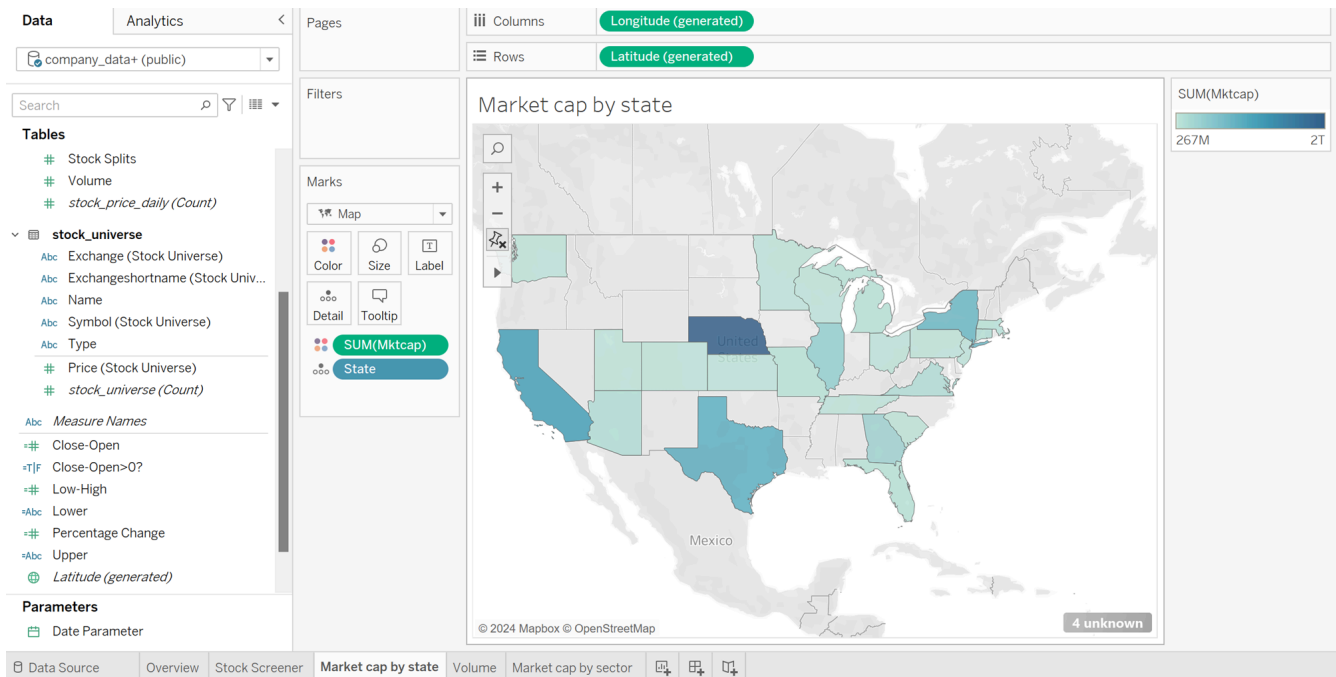
This worksheet visualizes market capitalization by state, utilizing both market cap data and state information. The color gradient represents the market size, ranging from $267 million to USD 3 trillion, allowing users to easily differentiate between states based on their market size. The state data is represented in a map format, providing a visual representation of market capitalization distribution across different geographic regions. This visualization enables users to quickly and intuitively understand the relative market sizes of different states and identify any regional trends or disparities in market capitalization.

Tables and Fields Used-

- State: This field represents the geographical states or regions for which market capitalization data is being visualized.
- Market Cap: This field contains the market capitalization data for each state, indicating the size of the market in terms of total market value. It is used to determine the color gradient on the map, representing the varying market sizes across different states.

**Volume Comparison of Different Stocks**
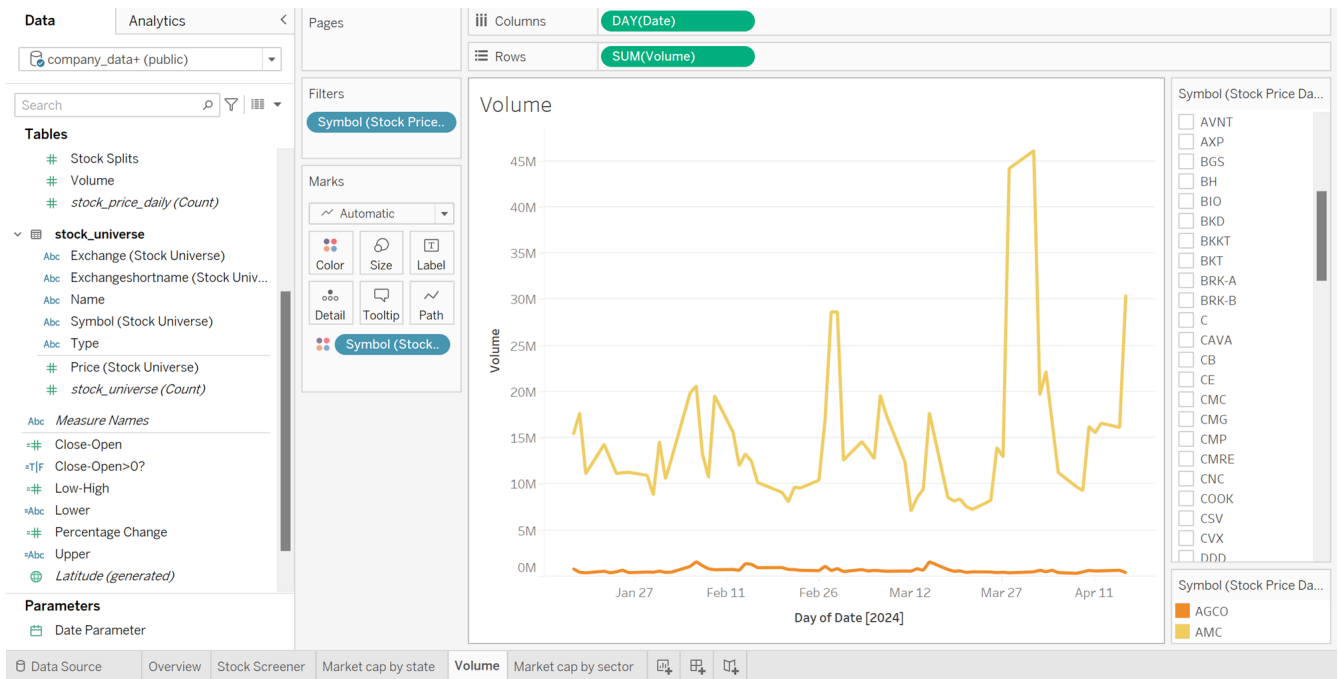
This visualization compares the trading volume of different stocks over time. The date is represented on the X-axis, while the trading volume is plotted on the Y-axis. Each stock symbol is assigned a distinct color, allowing users to add multiple symbols and compare their trading volumes within a single dashboard. This approach facilitates quick and effective analysis of trading activity across different stocks, enabling users to identify trends, patterns, and anomalies in trading volume over time.

Tables and Fields Used-

- Date: This field represents the date of trading activity, plotted along the X-axis to show changes over time.
- Volume: This field represents the trading volume of each stock, plotted along the Y-axis to visualize the volume of shares traded on each date.
- Symbol: This field identifies the stock symbol or ticker of each company. It is used to differentiate between different stocks and assign distinct colors to each symbol for easy identification and comparison within the visualization.

Market Capitalization by Sector
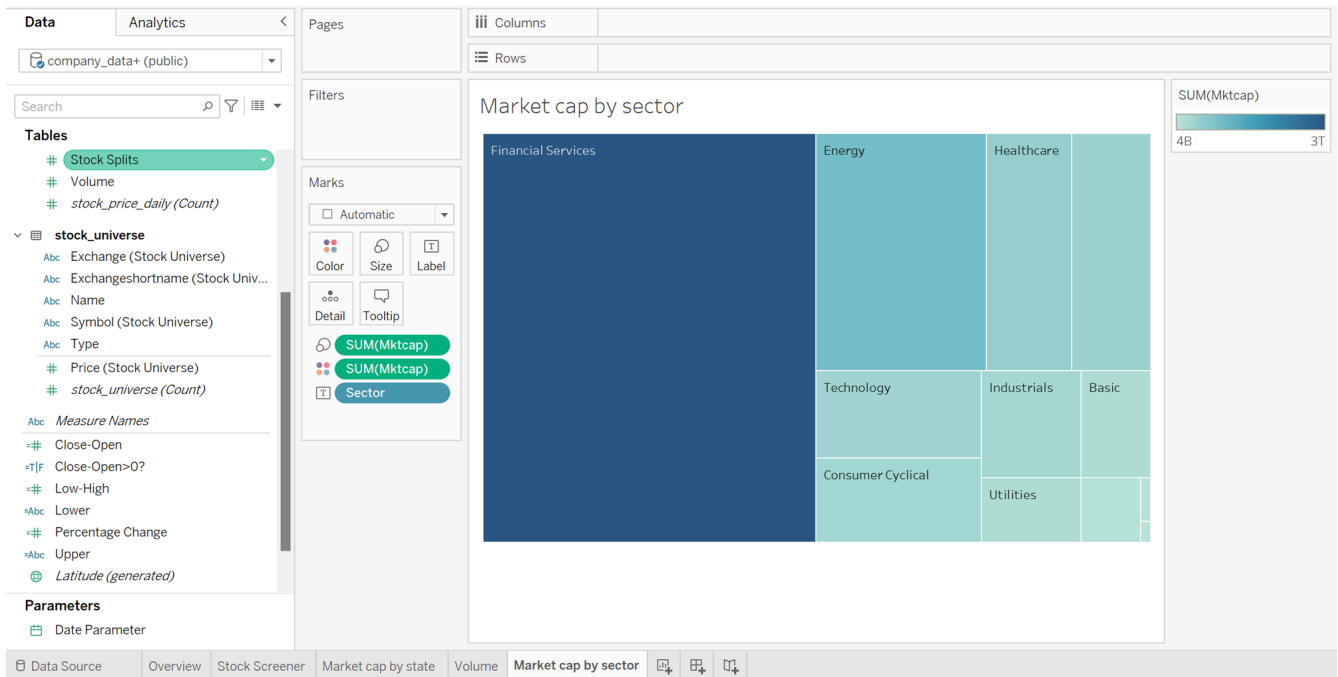
This worksheet visualizes market capitalization by sector, displaying the data using a color gradient that makes it easy to distinguish between different sectors. Additionally, the size of each sector's representation is proportional to its market capitalization, providing a clear visual representation of the relative sizes of the sectors within the market. The sector names are also displayed in text, ensuring that viewers can easily identify each sector. This visualization offers a comprehensive overview of market capitalization across different sectors, allowing for quick and effective analysis of market trends and sector performance.

Tables and Fields Used-

- Sector: This field represents the different sectors within the market, such as technology, healthcare, finance, etc. It is used to categorize the market capitalization data based on sector.

- Market Cap: This field contains the market capitalization data for each sector, indicating the size of the market in terms of total market value. It is used to determine the color gradient and size of each sector's representation, reflecting the varying market sizes and relative proportions of each sector within the market.

Automated Data Refresh and Visualization Updates

In order to maintain up-to-date visualizations for stock analysis, the following procedures have been implemented:

1. Scheduled Redshift Cluster Refresh: The Amazon Redshift cluster is set to refresh daily at 6:00 PM (local time). This ensures that the data used for analysis is consistently updated to reflect the latest information.
2. Visualizations Overview:
   - Stock Volume Comparison: Compares the trading volumes of various stocks over a specified period.
   - Stock Price Comparison: Displays the comparative stock prices of selected assets over time.
   - Market Cap by State: Illustrates the market capitalization distribution across different states or regions.
   - Stock Scanner: Provides insights into stocks meeting specific criteria or conditions, facilitating efficient stock selection.
3. Automated Refresh Process:
   - Once the Redshift cluster is refreshed at 6:00 PM daily, the data used in the visualizations is automatically updated.
   - This ensures that the stock volume comparison, stock price comparison, market cap by state, and stock scanner visualizations reflect the most recent data for informed decision-making.

By automating the data refresh process, stakeholders can rely on accurate and timely insights for effective stock analysis and decision-making.