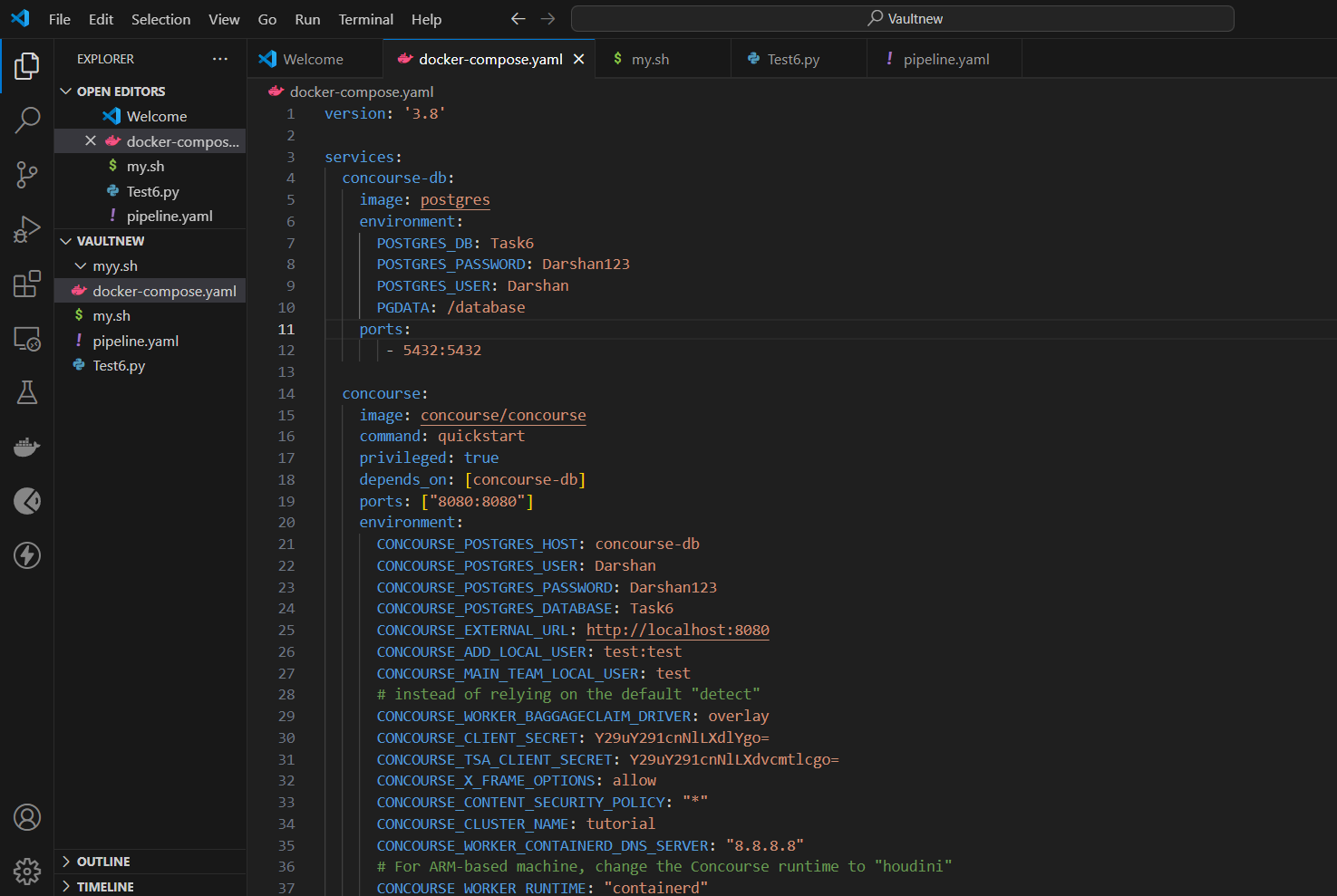
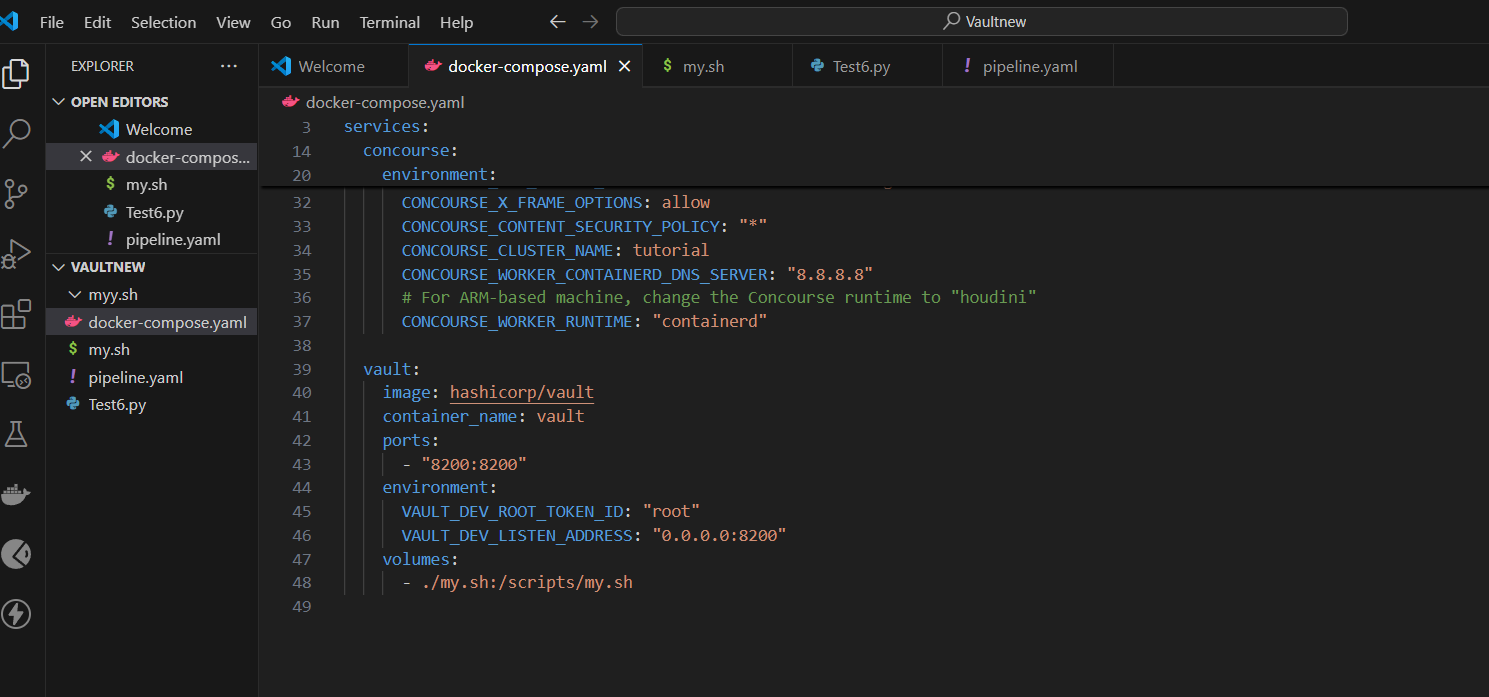
Task 6. Setup concourse through docker

* Connect concourse to fetch user and password from vault (Shell File) - (Sub-Task 1).
* Use those credential to login into <https://www.screener.in/> - (Sub-Task 2).
* Search Reliance data from screener and download an Excel (Sub-Task 3).
* web-scrape that excel data (ProfitandLoss Tab) and put that into our data base.

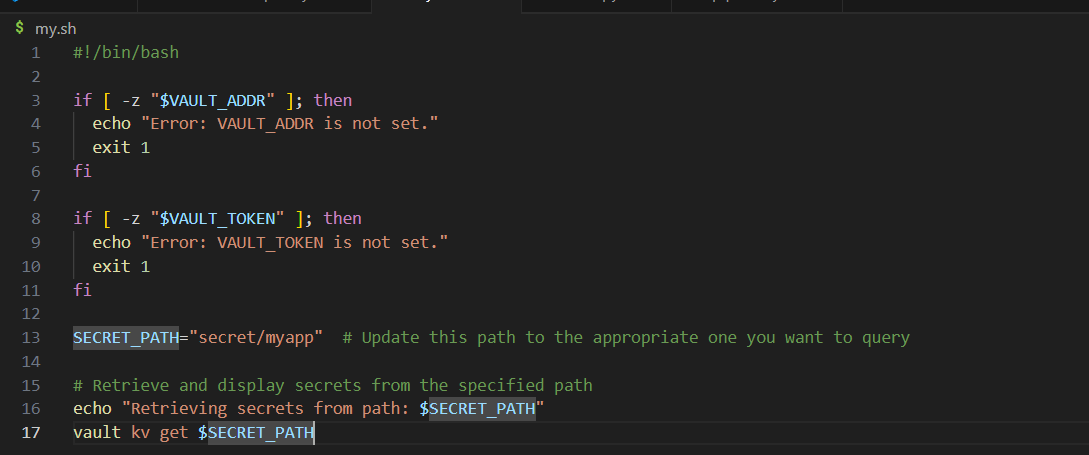
**Step1:**

**Created a new docker-compose**

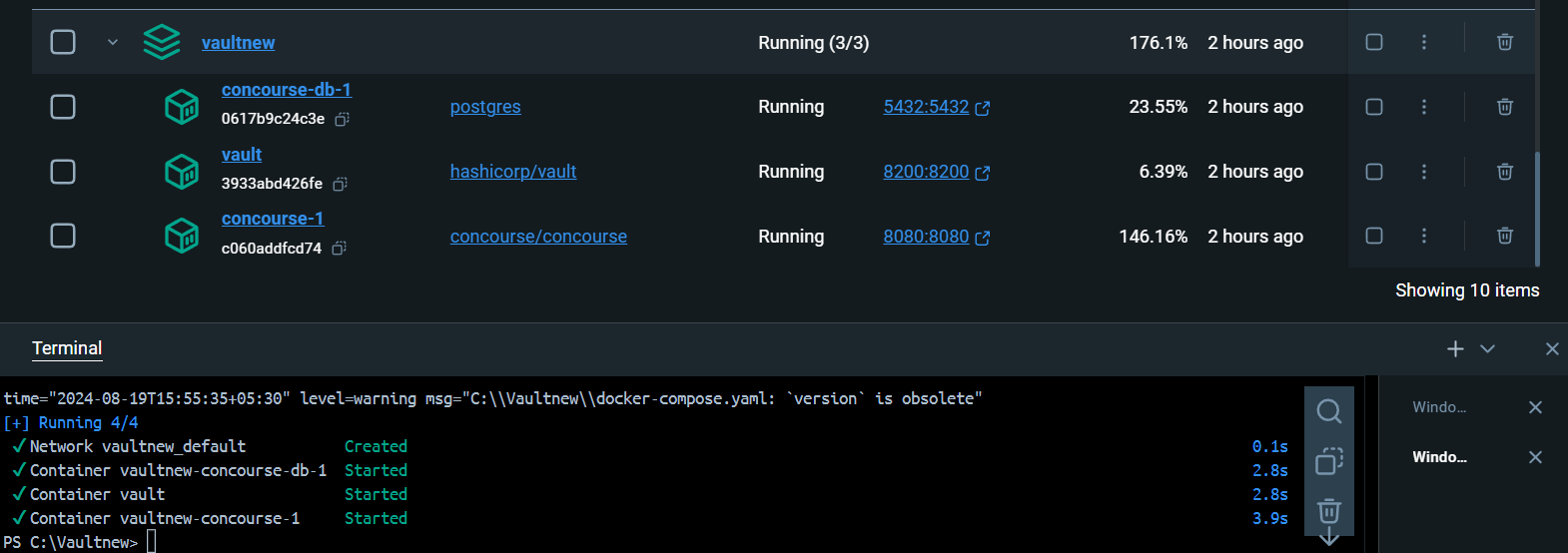




**Made a new vault connection as well:**



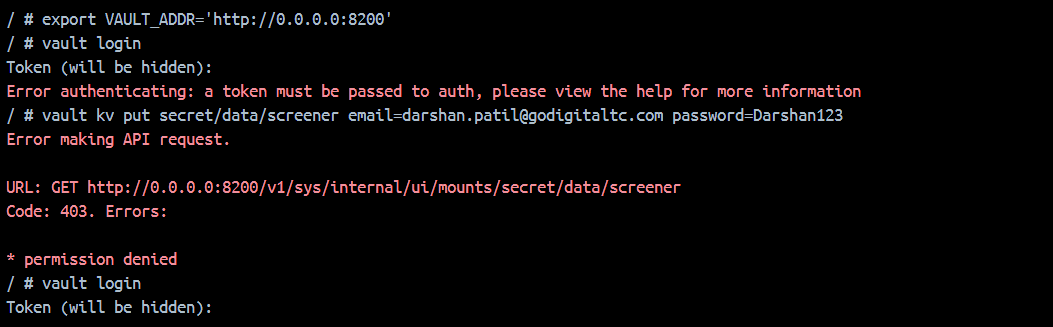
**Result:**



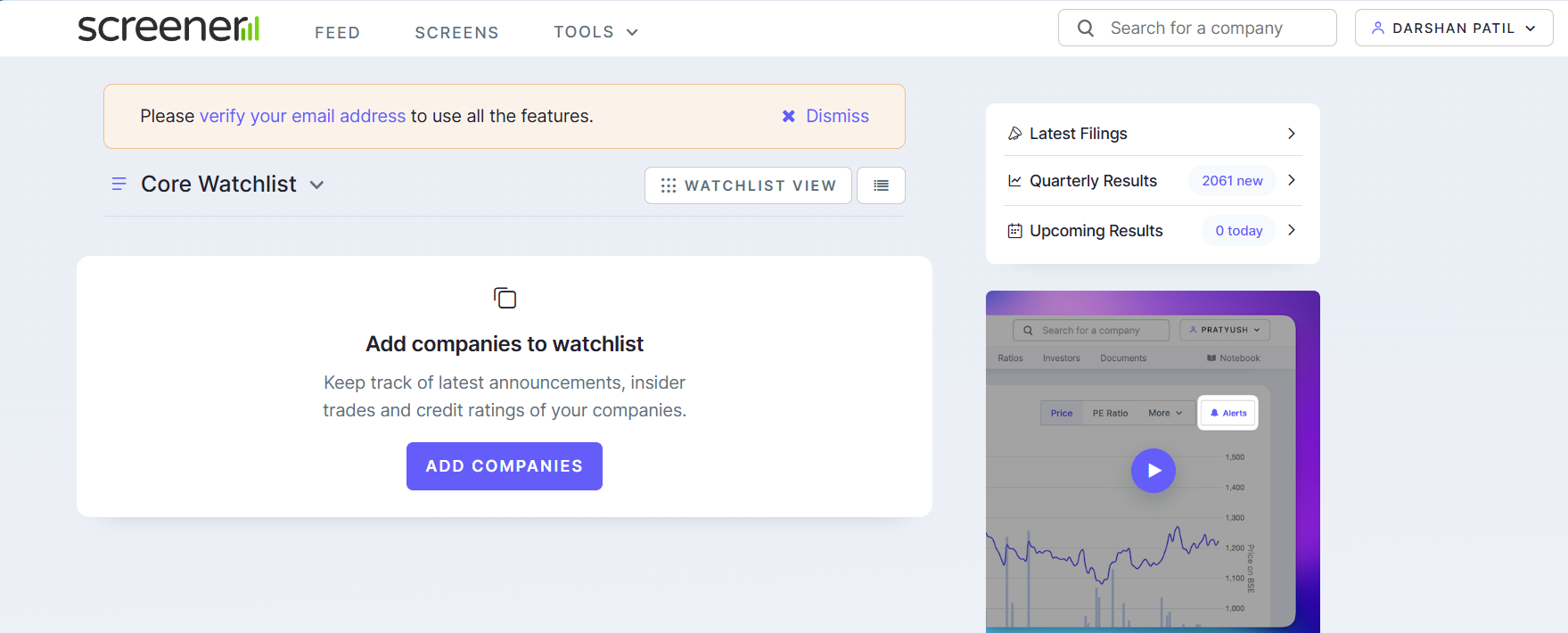
**Step2:**

**Created account and then made login on screener**

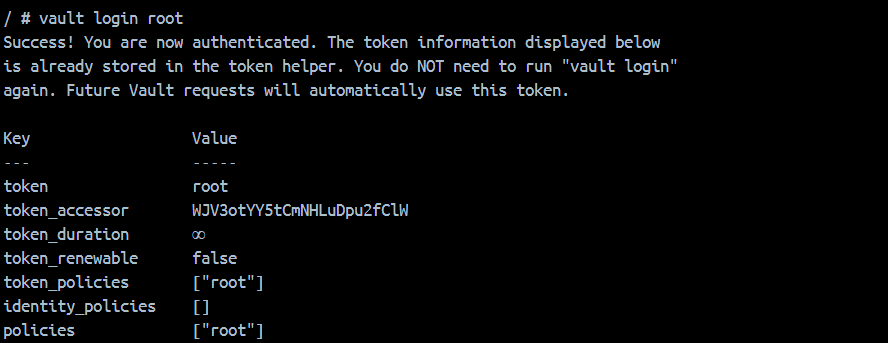
**Now in shell of vault,**

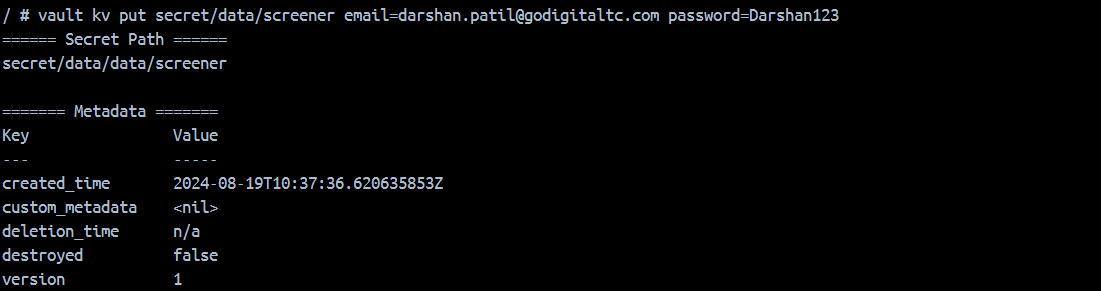


**Need to create login in screener first:**

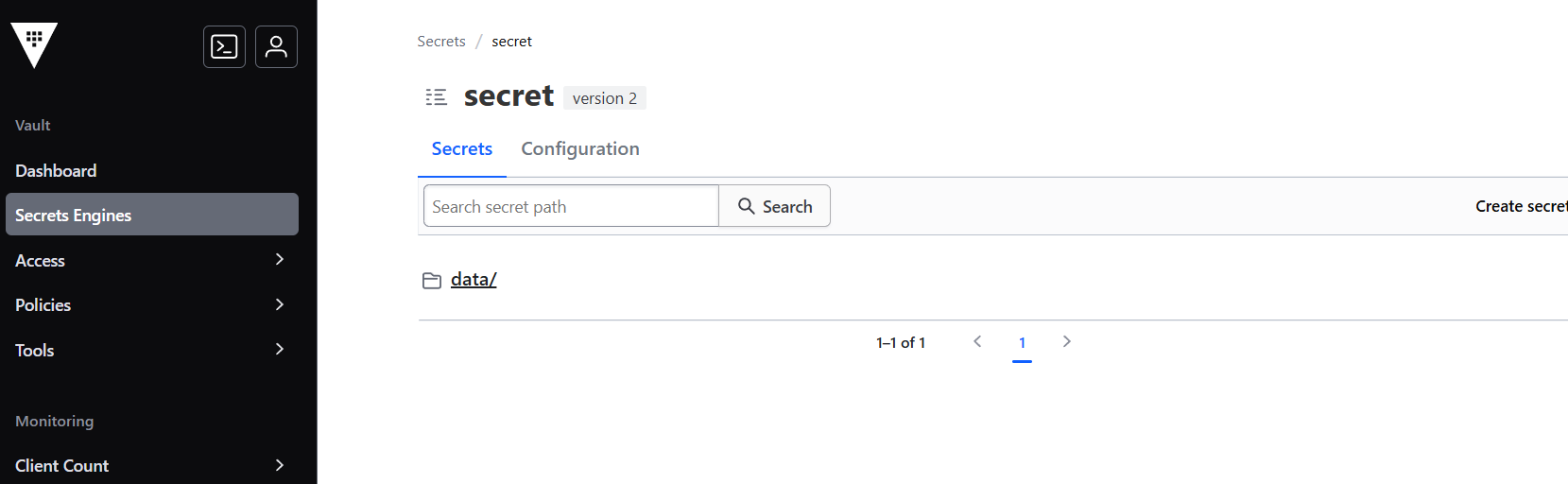


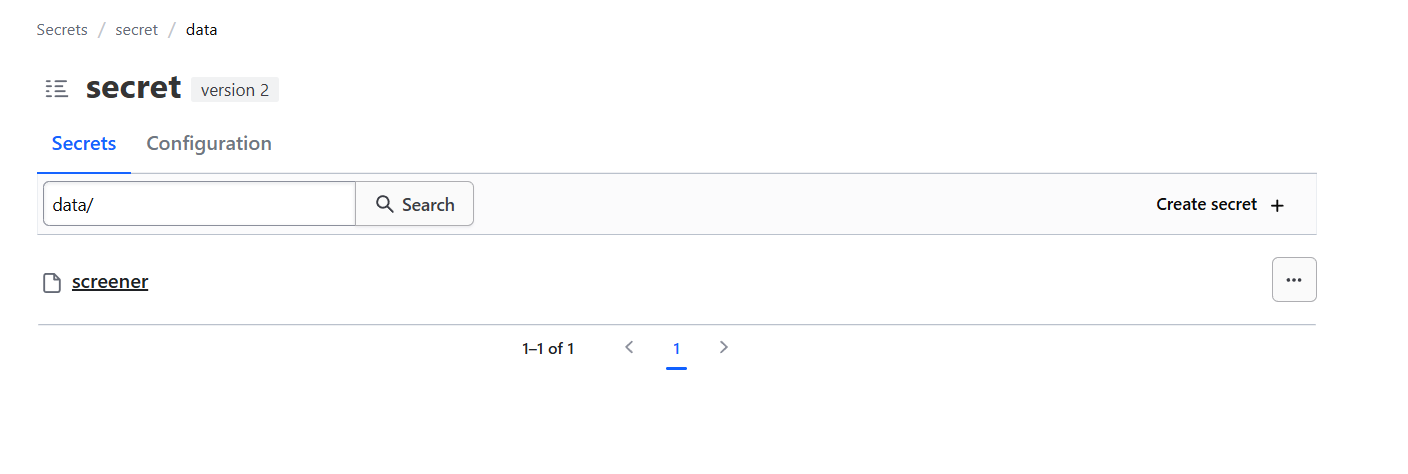
**Now come again in vault shell:**

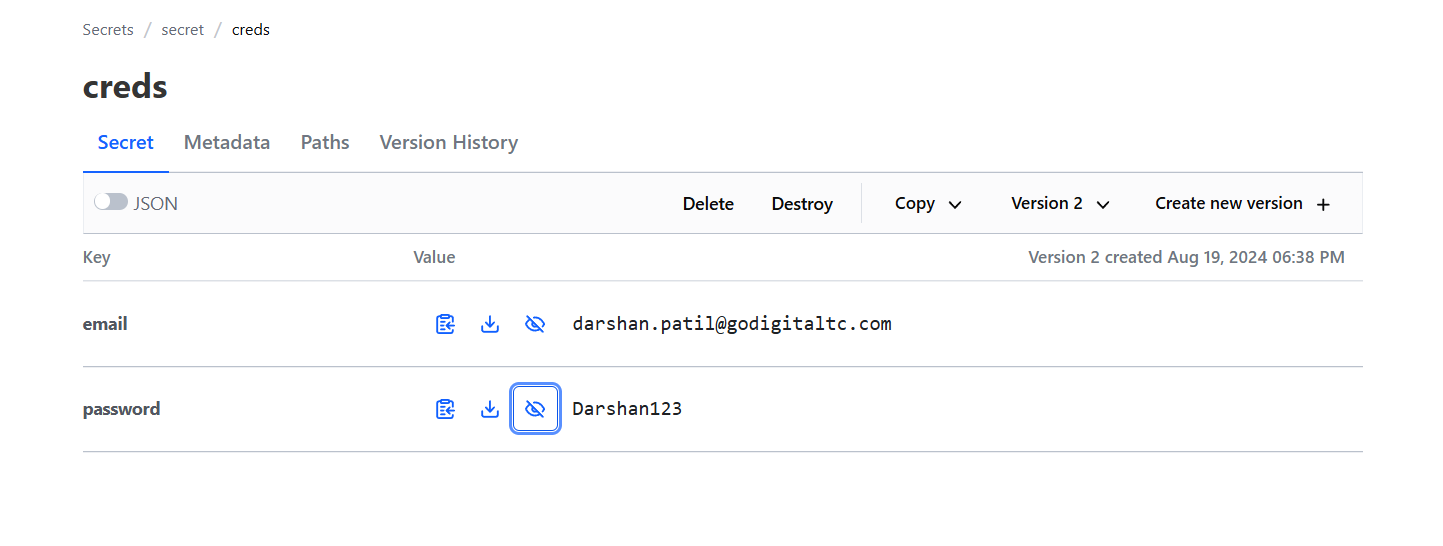




**As a result, I can see those credentials into vault:**



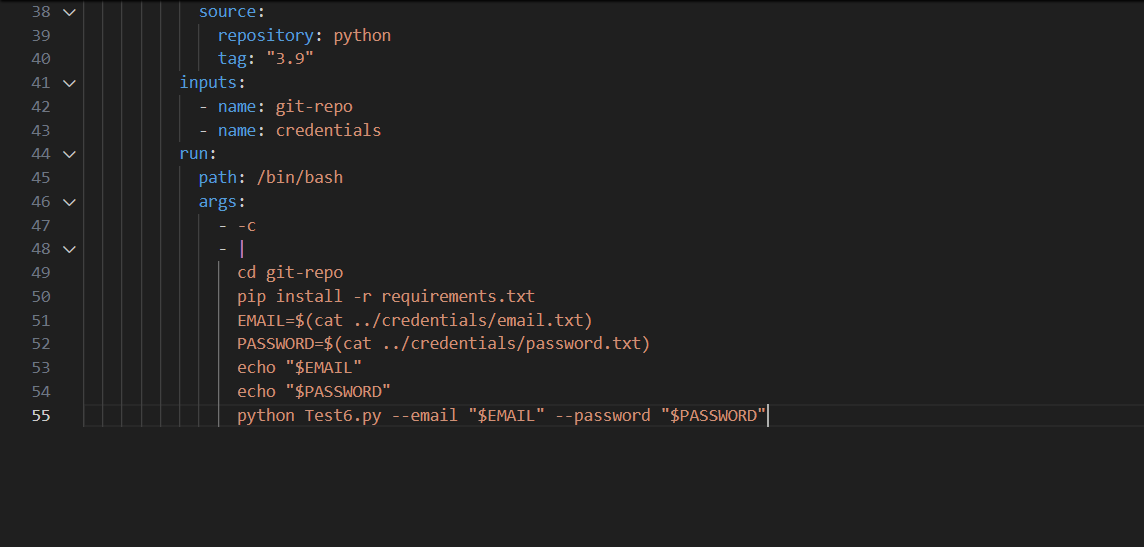
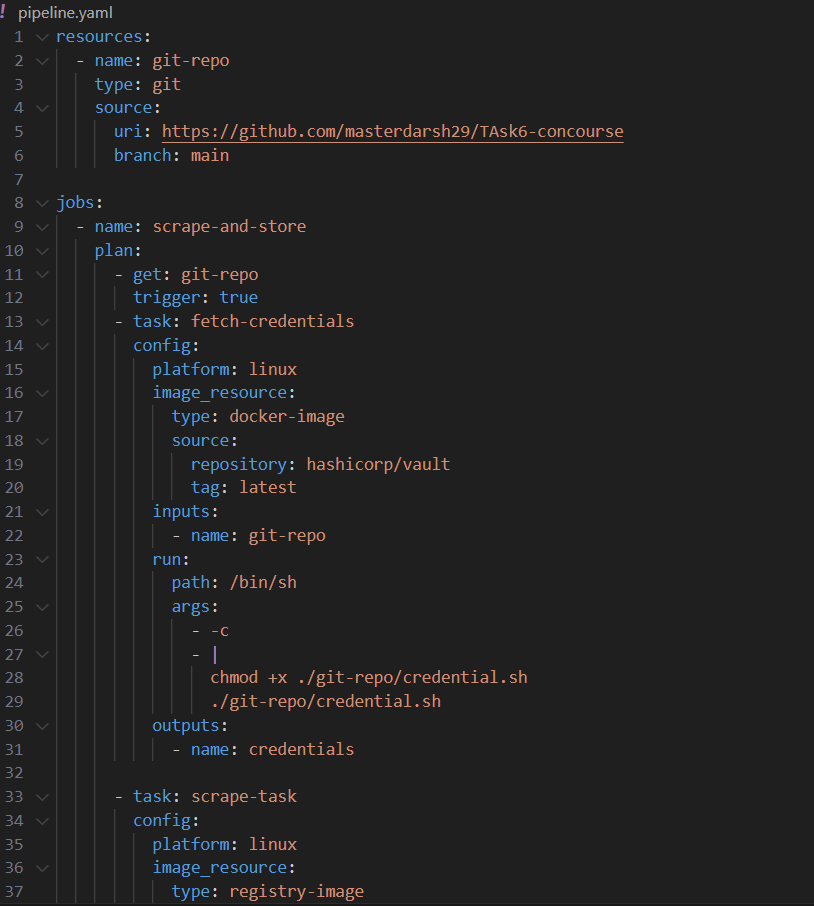




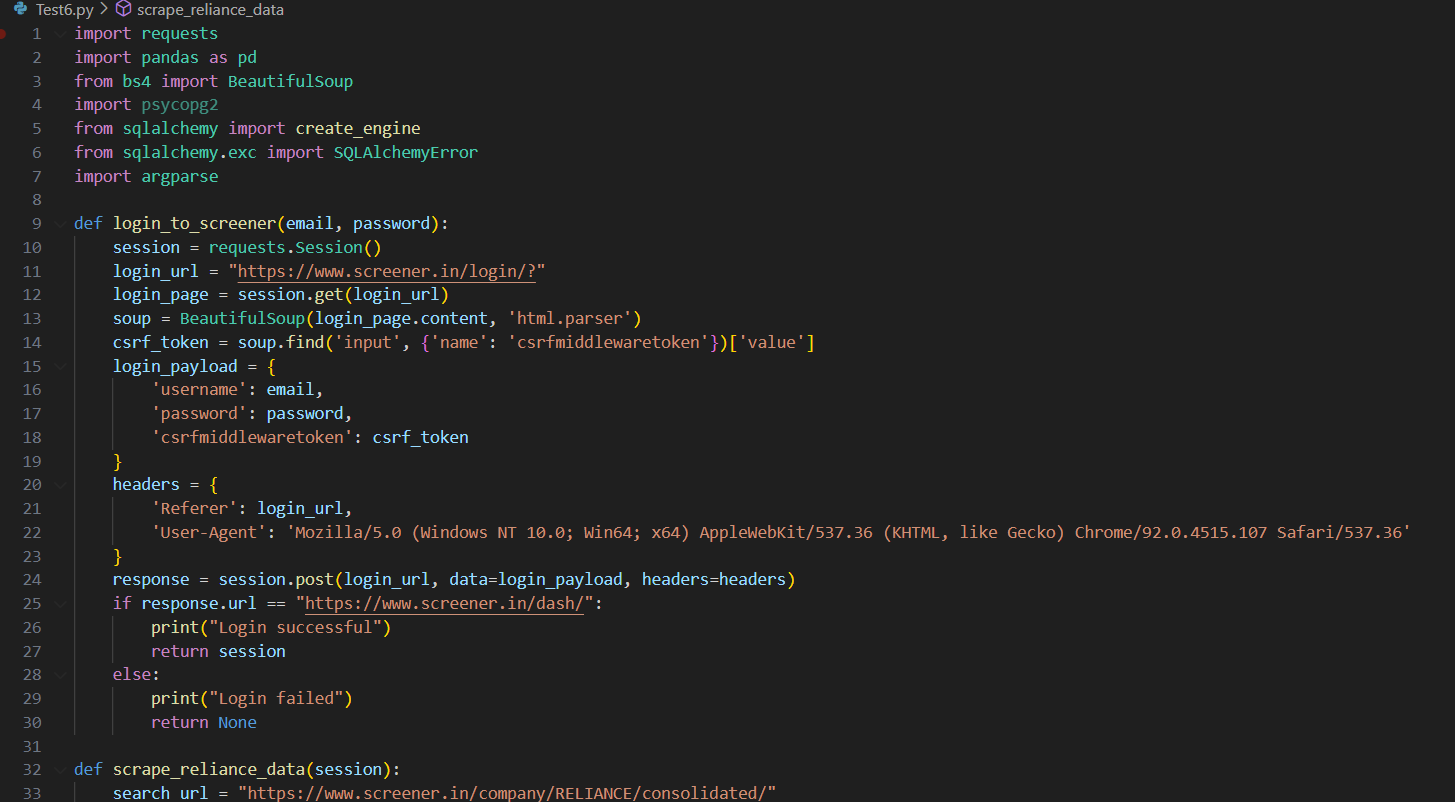
------------------------------------------Subtask1 & 2 done-------------------------------------------------------------------

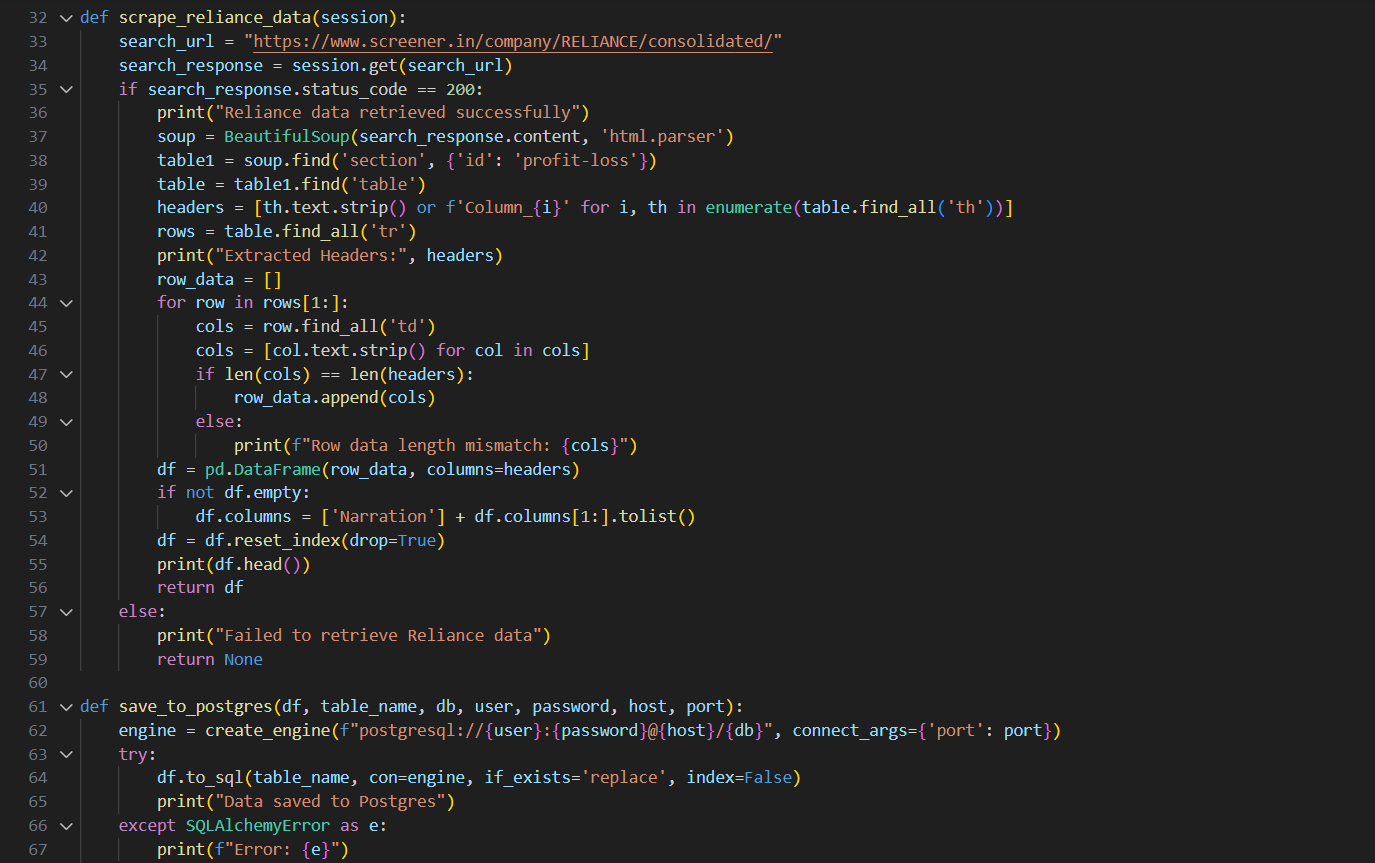
**Step3:**

**Make a pipeline.yaml and Test6.py file.**



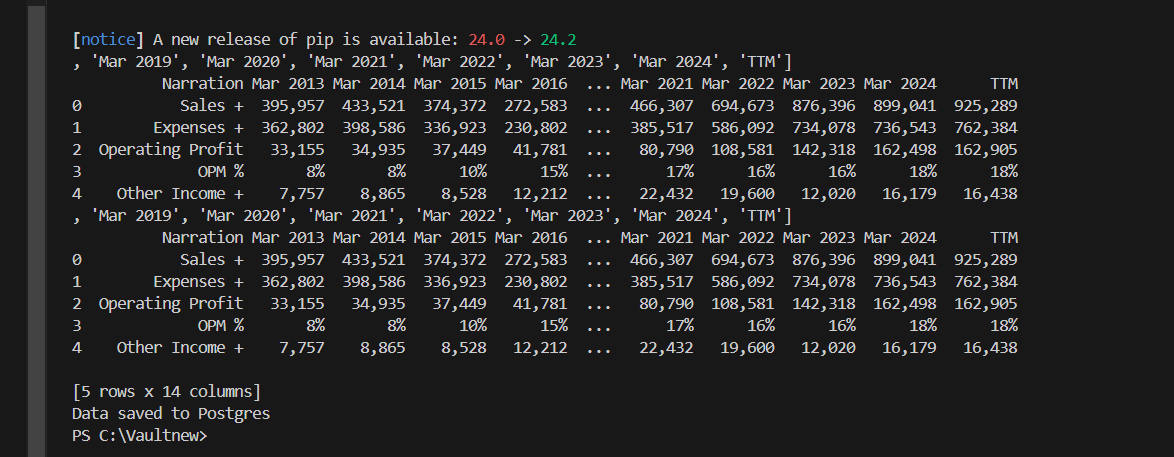
**The python\_script:**







**As in local system I can able to fetch the data here,**



**This Python script appears to be designed to:**

1. **Log in to the Screener.in website using provided email and password credentials.**
2. **Scrape financial data for Reliance Industries from the website.**
3. **Save the scraped data to a PostgreSQL database.**

**Here's a high-level overview of the script's components:**

* **login\_to\_screener: Logs in to Screener.in using the provided credentials and returns a session object.**
* **scrape\_reliance\_data: Uses the session object to scrape financial data for Reliance Industries and returns a Pandas DataFrame.**
* **save\_to\_postgres: Saves the scraped data to a PostgreSQL database using the provided connection details.**

**The script uses various libraries, including:**

* **requests for making HTTP requests**
* **BeautifulSoup for parsing HTML content**
* **pandas for data manipulation and storage**
* **sqlalchemy for interacting with the PostgreSQL database**

**The script also uses argparse to parse command-line arguments for the email, password, and database connection details.**

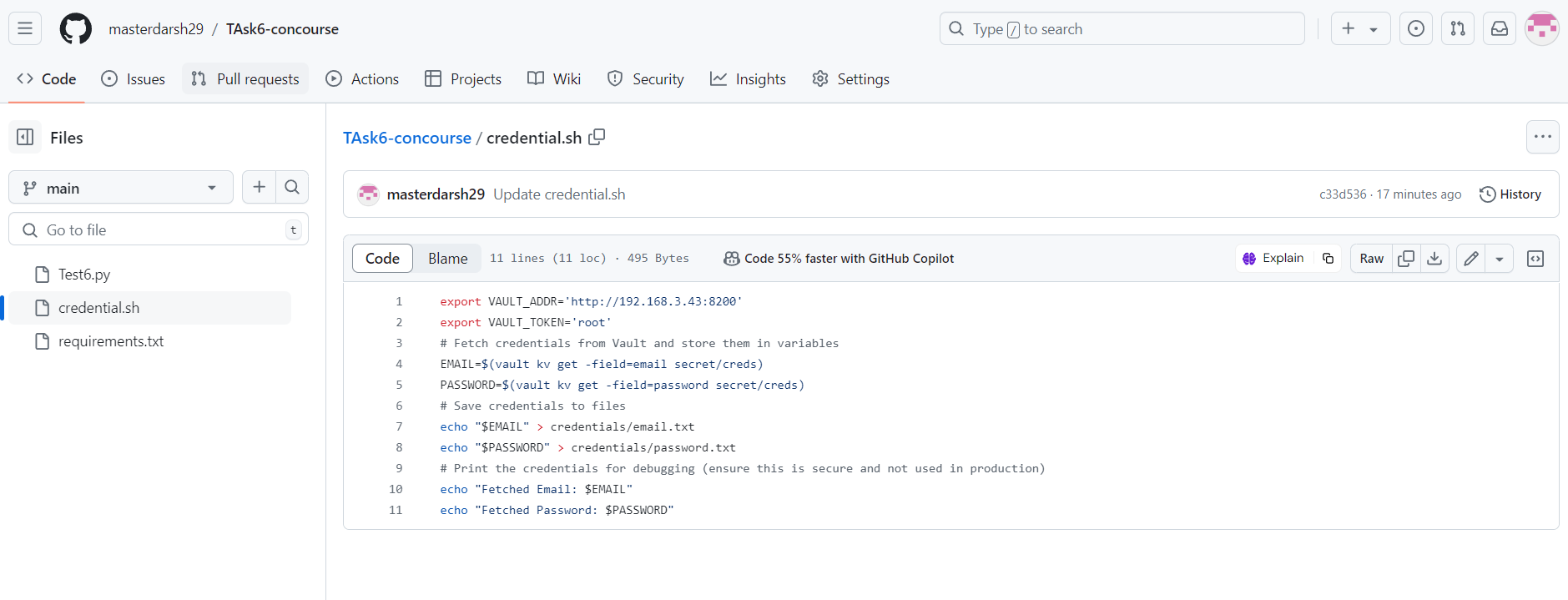
**To integrate this script with your Concourse pipeline, you can:**

1. **Store the script in your Git repository.**
2. **Update your pipeline.yaml file to include a task that runs the script using the python image.**
3. **Pass the required arguments to the script using environment variables or input parameters.**

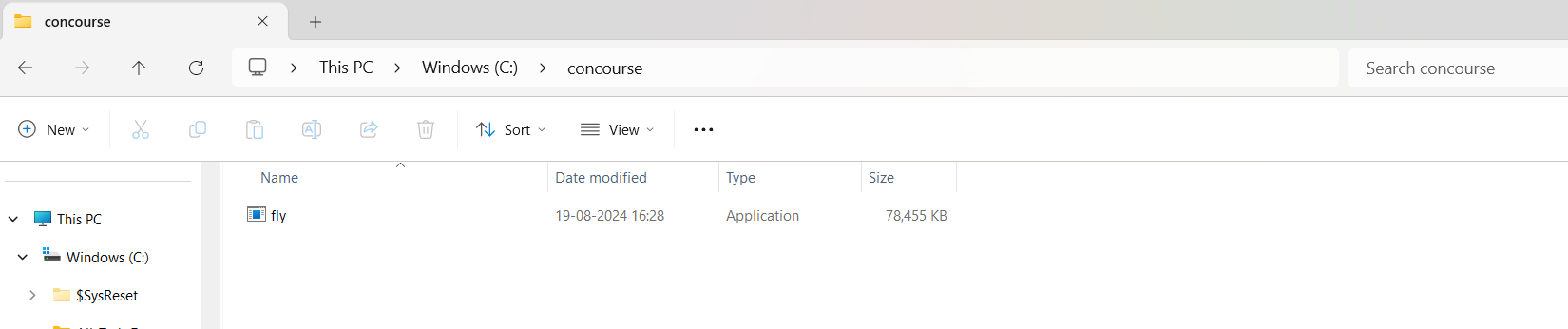
**Step4:**

**Made a github repository and stored python script, requirements and credentials.sh file**

[**https://github.com/masterdarsh29/TAsk6-concourse**](https://github.com/masterdarsh29/TAsk6-concourse)



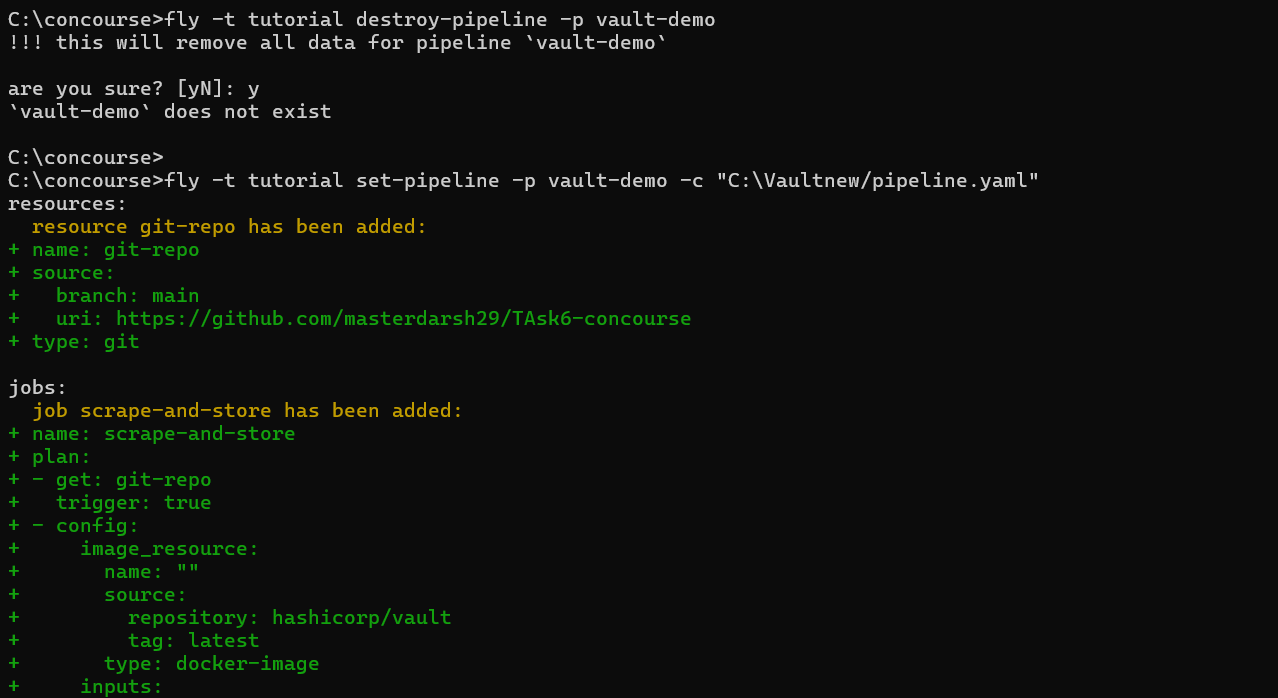
* **Now, to store all these files and pipelines those files need to create username and password and with the help of fly commands you can destroy, set and unpause progress of pipelining.**
* **Before running fly commands need to install it on desktop**



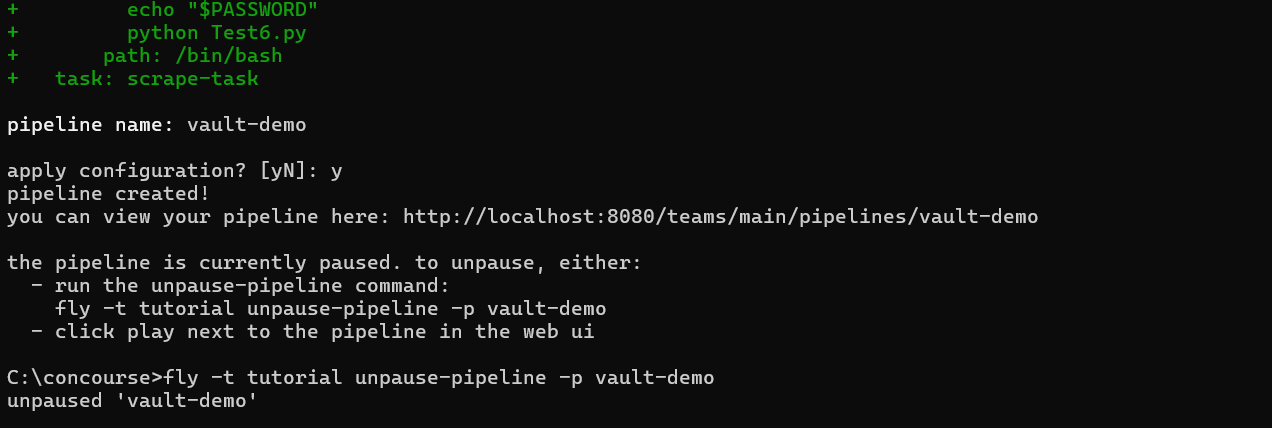
**Then open cmd in concourse**

**Run following commands step by step**

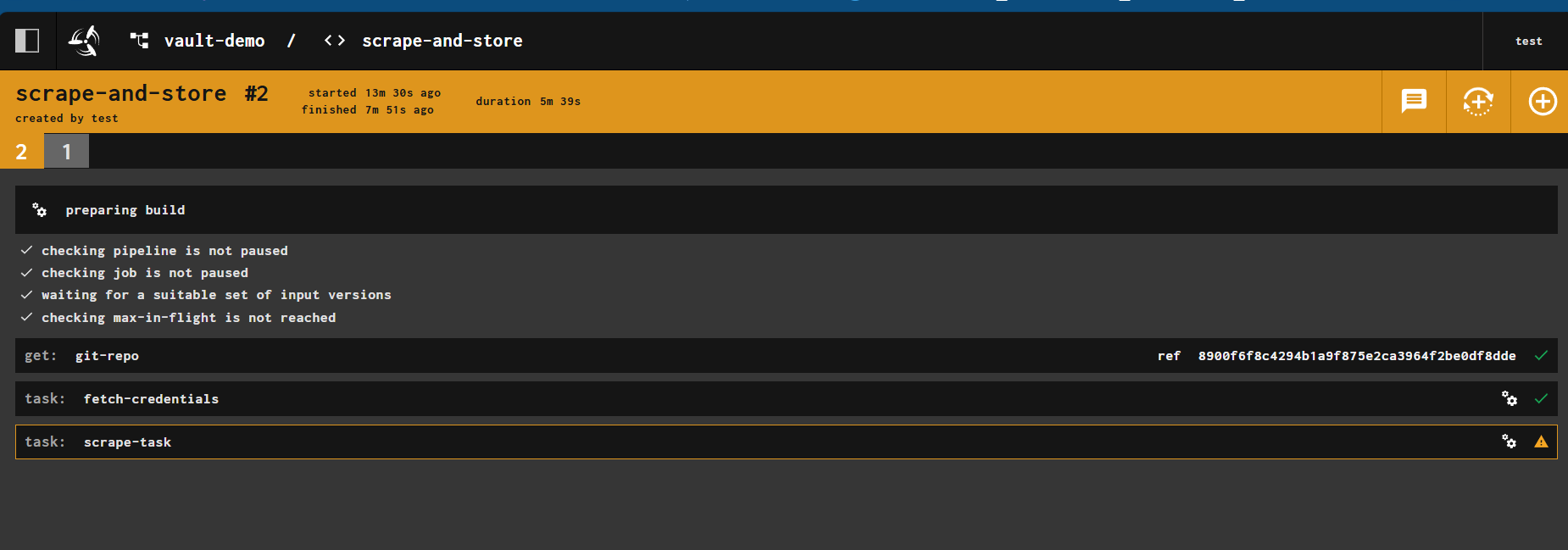




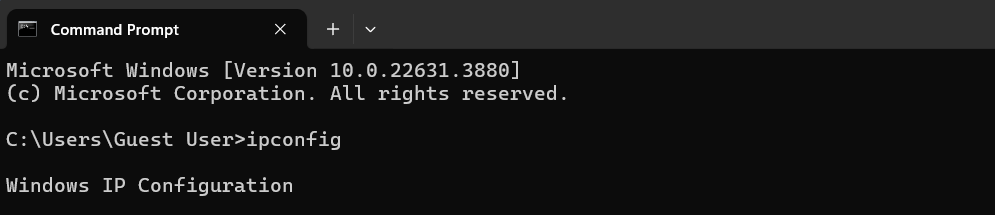
Then unpause

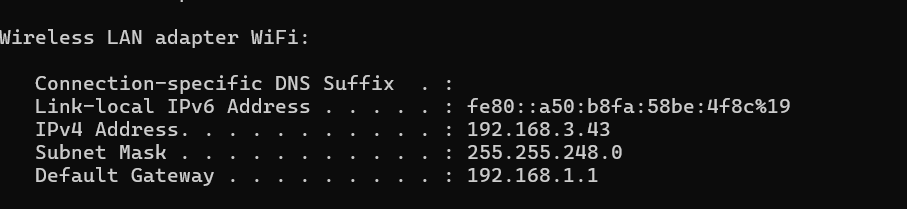


**Now, after this whenever you will concerned any error while scrap-store in concourse**



* Then you need to execute all fly commands from start.
* Destroy, set, unpause
* Then go and start scrap and store from vault-demo









* **Then in git repository, in credential.sh change ip address as needed and then re reun fly commands**

**fly -t tutorial login -c http://localhost:8080 -u test -p test**

**As followed by   
destroy:**

**fly -t tutorial destroy-pipeline -p vault-demo**

**set:**

**fly -t tutorial set-pipeline -p vault-demo -c “C:/Vaultnew/pipeline.yaml”**

**unpause:**

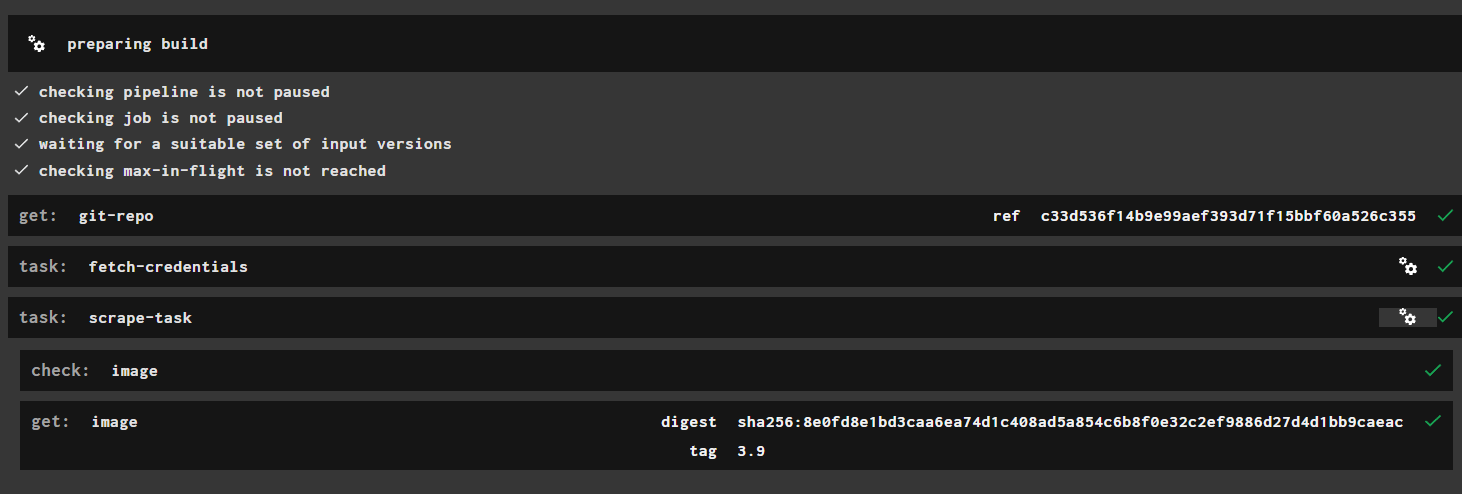
**fly -t tutorial unpause-pipeline -p vault-demo**

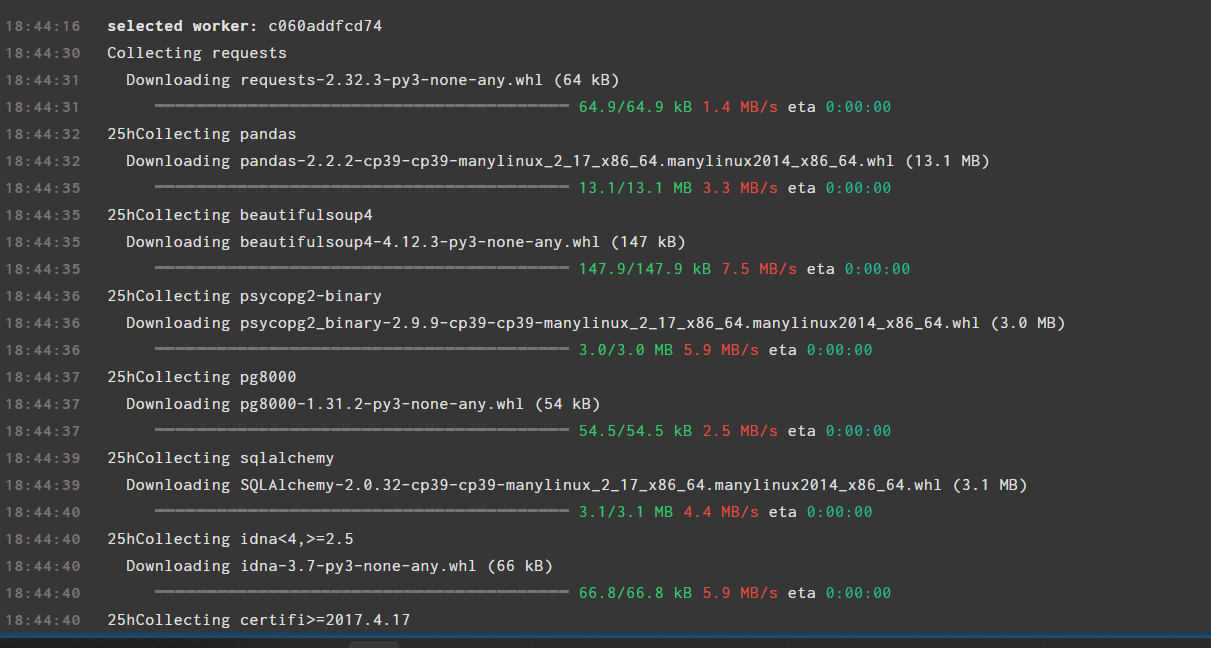
**Step5:**

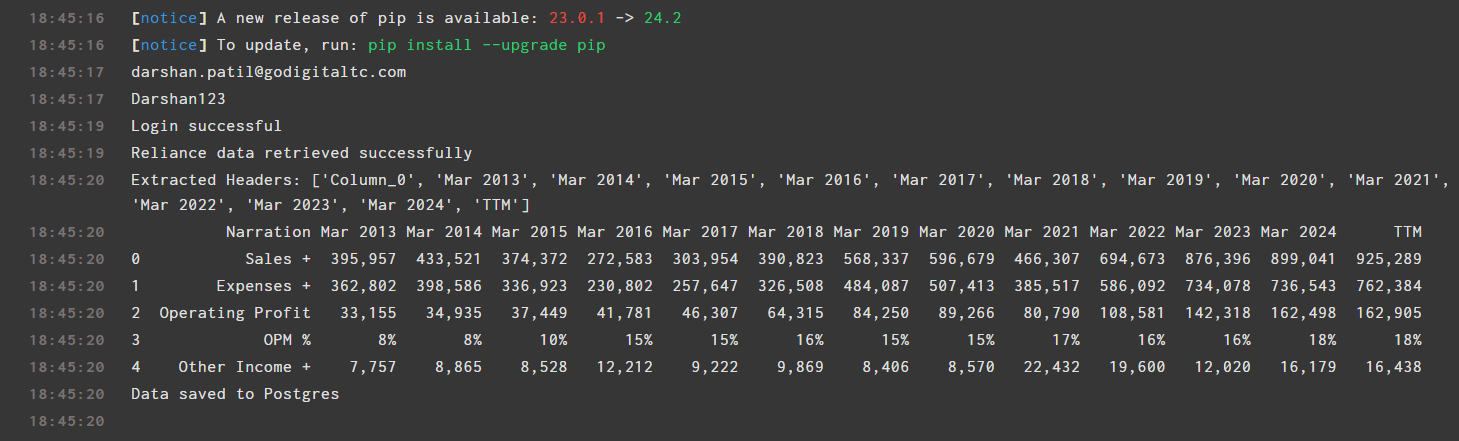
**Now open vault-demo**



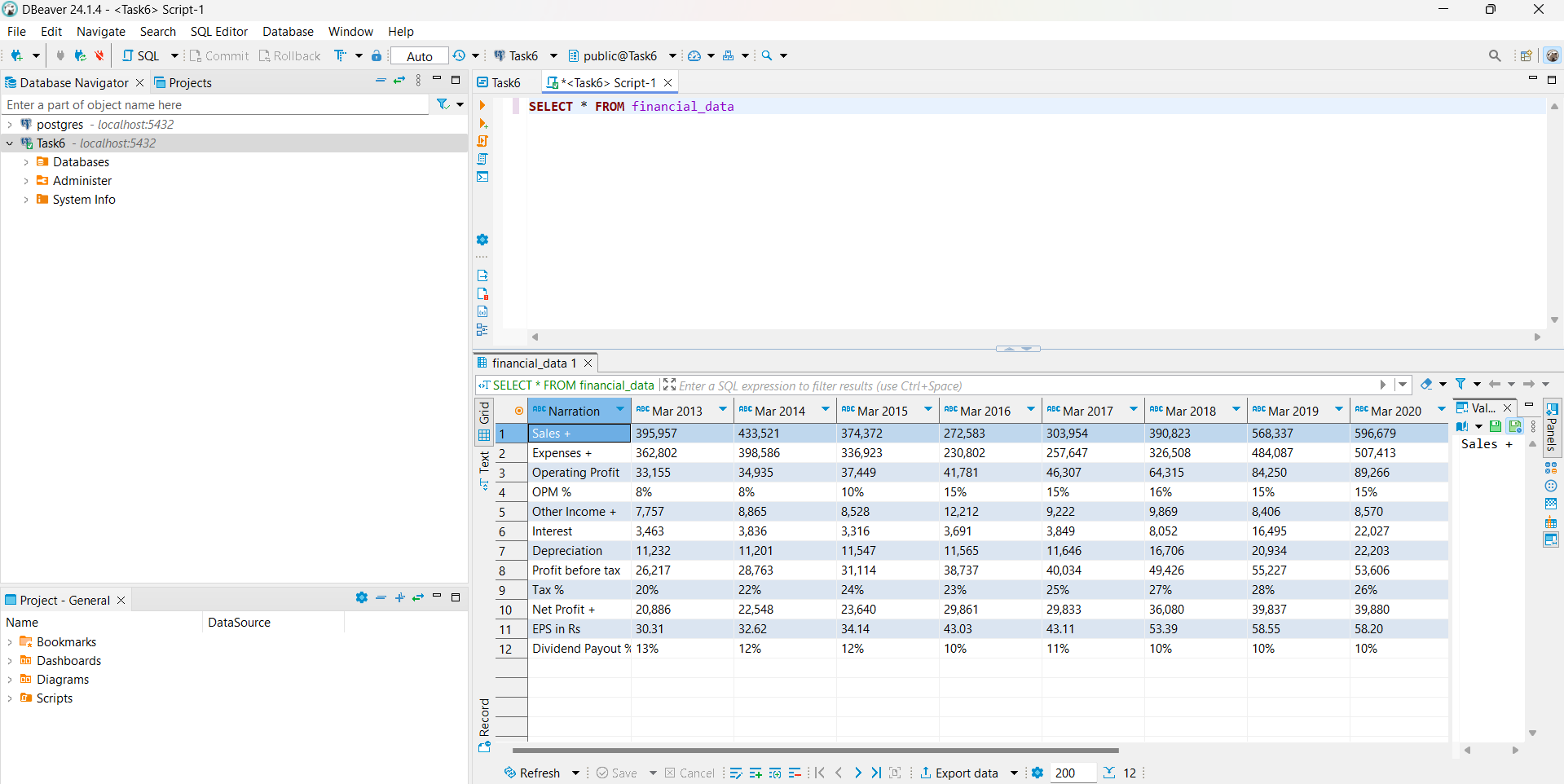
**Try to scrap-and-store your pipeline and fetch data into your postgres connection database.**







**Here your scrapped data from screener site has been successfully stored into your database.**



---------------------------------------------SubTask3-done--------------------------------------------------------------------

**Issues Got:**

* 1. **Required to made a separate docker-compose for al configuration of concourse-db and vault.**
  2. **Credentials issue (as in vault there was email, password data, but in credentials there was something username, password) so stucked on that issue for a while.**
  3. **So, whenever tried to run scrap files in vault-demo, always needed to re-run it, that time I forgot to destroy previous vault-demo.   
       
     Resource followed:**

**https://concourse-ci.org/managing-teams.html**