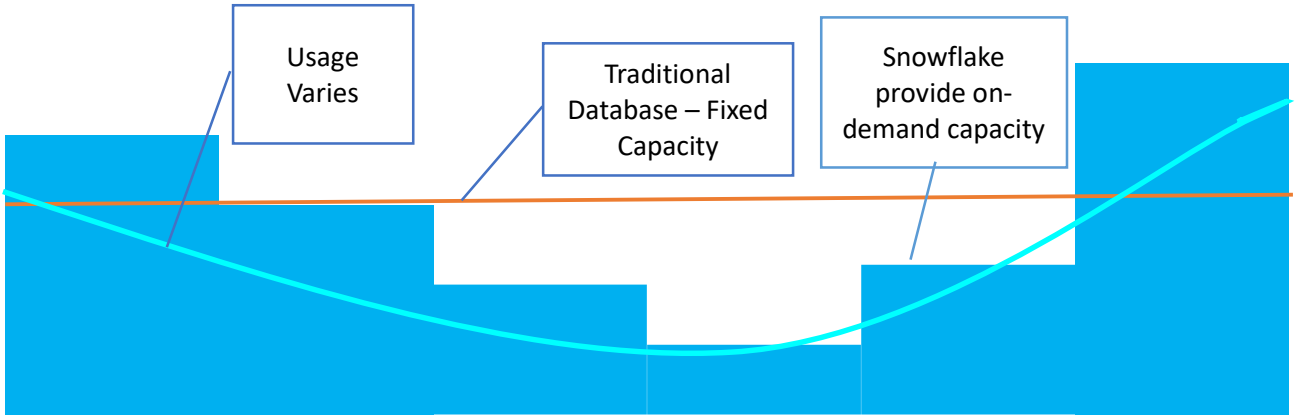
Project Title: Self-Service - Snowflake Compute Request

1. Introduction to Snowflake

Snowflake (database) is a cloud data warehouse that can store and analyze all your data records in one place. Snowflake does not tightly couple storage, compute, and database services — it can dynamically modify configurations and scale up or down resources independently.

The elastically scalable feature enables you to run virtually any number of workloads across many users at the same time without worrying about resource contention. Workloads can include use cases such as batch data processing to interactive analytics to complex data pipelines.

To support high compute demand, Snowflake can scale on-demand capacity and performance as needed, data teams no longer need to run upfront capacity planning exercises. Nor do they need to maintain costly oversized data warehouses that remain mostly underutilized.



Snowflake virtual warehouse (compute) match customers data warehouse usage patterns

Snowflake supports two ways to scale warehouses (compute):

Manually Scale up by resizing a warehouse:

Resizing a warehouse generally improves query performance, particularly for larger, more complex queries. It can also help reduce the queuing that occurs if a warehouse does not have enough compute resources to process all the queries that are submitted concurrently.

Automatic Scale out multi-cluster warehouse:

Scale out by adding clusters to a multi-cluster warehouse (requires Snowflake Enterprise Edition or higher) are designed specifically for handling queuing and performance issues related to large numbers of concurrent users and/or queries.

For more information on Snowflake warehouse (compute) please refer to: link

2. Project Introduction/Objective:

When an application or user need large warehouse (compute) to process complex batch processing, model execution, etc., The platform team review the request along with last 90 days usage trend and make decision by fulfilling the request with large compute or else decline the request based on database usage trend for the user or not convincing business justification.

Currently this request process is managed through email/chat with multiple follow up and operational delay towards review and warehouse provisioning.

The proposed self-service tool helps users to submit their requests through web application. Platform team get notification on the request and start the review supported by usage trend reports. As the decision is made (approved or rejected) the warehouse provisioning is automated, and the user receives notification on his/her request. Also, the process enables log capture in each stage that helps to track any lag in fulfillment process and provide evidence for compliance requirement.

3. Requirements Analysis:

We plan to use following python packages in this project

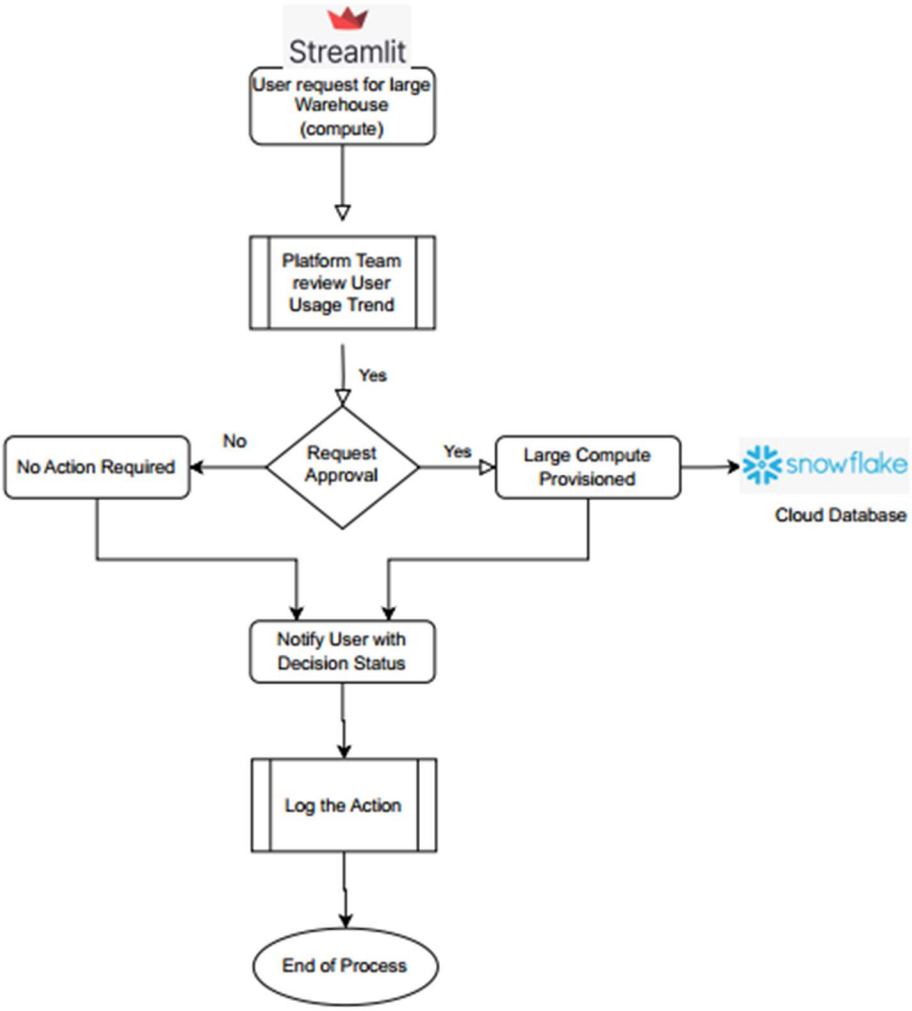
- Python 3.8 - Snowflake Snowpark is compatible only with Python 3.8 version
- Streamlit - A faster way to build and share data apps
- Altair 4.0 – Visualization in Python
- Pandas – Data Analysis Library
- Snowflake-connect-python - Snowflake connector

4. System Architecture

- Users submit a web based form/request for larger warehouse (compute)
- Requests show up in platform team dashboard with user workload metrics.
- Platform team review and make decision to approve/decline the request.
- Approved: Large warehouse (compute) is provisioned to user (e.g MEDIUM is approved) by connecting to Snowflake
- Rejected: Based on user workload trend, platform team decline the request
- Notification: User and his/her manager receive email on request status [Final update: SMTP server is not available, so skipped the email notification. However, the app display the status of the request)

5. System Process Flow

Self-service user tool request, approval and notification process flow.



6. Data Design:

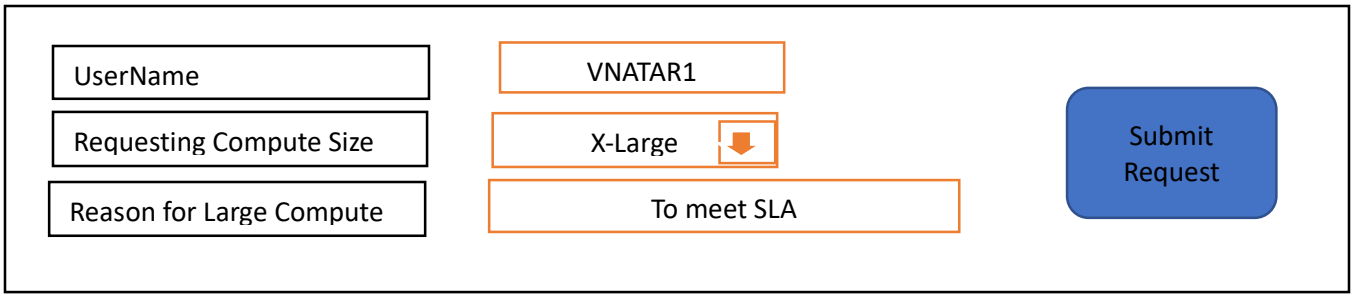
Persistent table is created in Snowflake database to record all user requests and the action taken by platform team.

- USER_NAME - Requester id
- REQUEST_COMPUTE - Snowflake Warehouse compute (size) requested.
- REQUEST_REASON - Business justification
- REQUEST_TIMESTAMP - Request submit timestamp

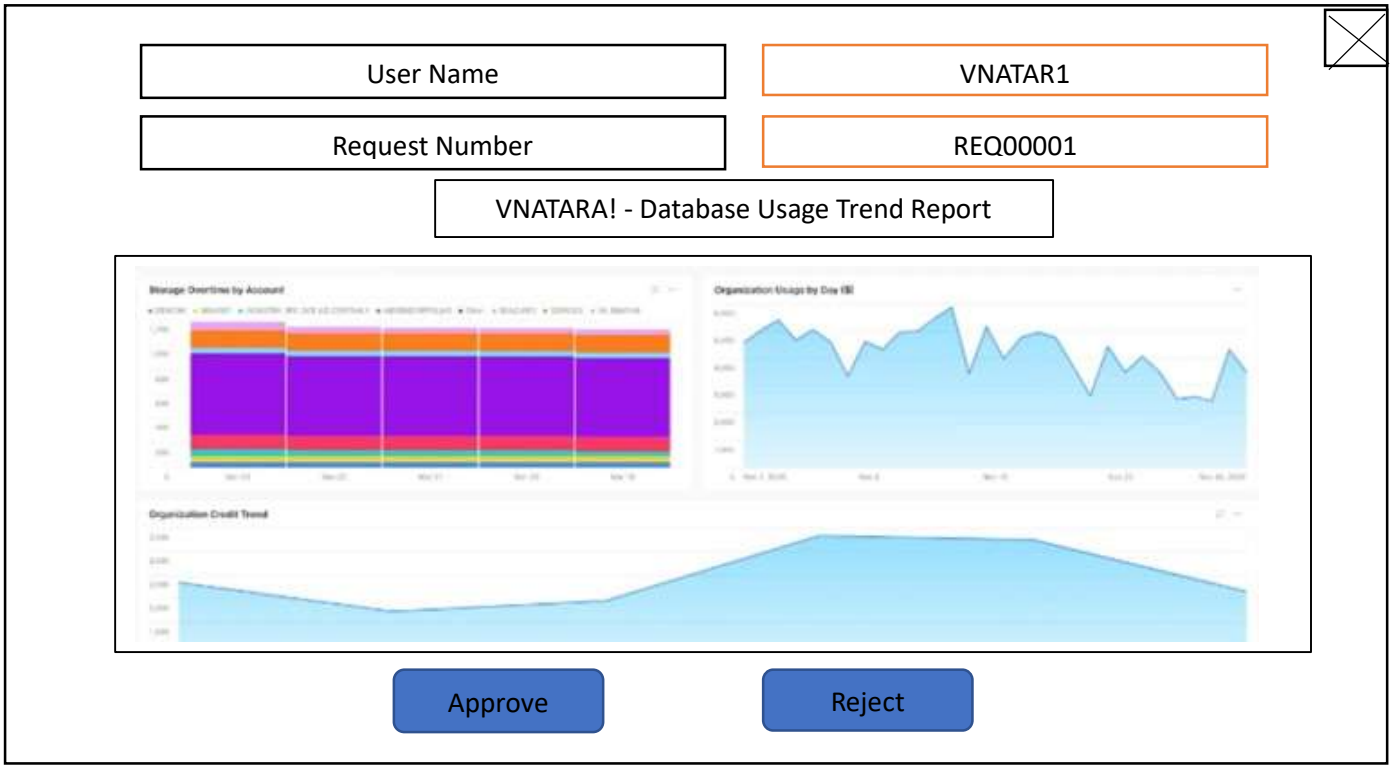
- REVIEWED_BY - Platform team member reviewing the request
- REVIEWED_TIMESTAMP - Review request timestamp
- REVIEWED_STATUS - Review status (approved/declined)
- APPROVED_COMPUTE - Warehouse compute approved
- APPROVED_TIMESTAMP - Request approved timestamp
- EMAIL_NOTIFICATION - Email notification sent (yes/no)

7. User Interface Design:

7.1. User request template:



7.2. Platform Admin Template:



[Final update: SMTP server is not available, so skipped the email notification. However, the app display the status of the request)

6.3. Email Template: Approved
Date: 06/28/2023 03:34pm Central

Hi Velu Natarajan,

Your request for the larger compute is approved. The access to warehouse is granted for next 8 hrs that expires at 06/20/2023 11:34pm Central.

Thanks.

6.4. Email Template: Rejected

Date: 06/28/2023 03:41pm Central

Hi Velu Natarajan,

Your request for the larger compute is not approved. The current trend show your existing warehouse can manage the new workload. For question, please reach Snowflake DBA team.

Thanks.

7. Module/Component Design:

- Web interface for users to submit computer requests.
- Web interface for platform team to review the request.
- Log tracker table in Snowflake database to record all user requests and action taken.

8. Testing Strategy:

8.1 Test-case-1:

Users choose warehouse (compute) size required, provide business justification and submit the request.

The request to show up in platform team dashboard

8.2 Test case-2:

Platform team review user trend report and approve the request

The approved compute should be granted to the user

Requester need to be notified through email

8.3 Test case-3:

Platform team review user trend report and reject the request

Requester need to be notified through email

9. Implementation Plan:

1. 07/02/2023 - Comprehensive software design document

2. 07/09/2023 - Script Stremlit Webpage

3. 07/16/2023 – Store persistent data in Snowflake

4. 07/21/2023 – Finish testing and documentation

5. 07/26/2023 – Submit final project after observing instructor comments

10. Reference:

1. Snowflake Warehouse (compute) sizing

Warehouse Size	Credits / Hour	Credits / Second	Notes
X-Small	1	0.0003	Default size for warehouses created using CREATE WAREHOUSE.
Small	2	0.0006	
Medium	4	0.0011	
Large	8	0.0022	
X-Large	16	0.0044	Default for warehouses created in the web interface.
2X-Large	32	0.0089	
3X-Large	64	0.0178	
4X-Large	128	0.0356	