

GenAIQ Guide

GenAIQ Marketplace Install | GenAIQ Application Setup

Table of Contents

1. Install GenAIQ from Snowflake Marketplace	3
1.1. Snowflake Role	3
1.2. Snowflake Marketplace	3
1.3. Install the Application	4
1.4. Check the Application in Snowflake	6
2. GenAIQ Native Application	6
2.1. External PostgreSQL Instance	7
2.2. Snowflake Secrets for PostgreSQL	7
2.3. LiteLLM and Snowflake PAT Setup	8
2.3.1. Create a LiteLLM Master Key (Example)	8
2.3.2. Create a Service User and PAT in Snowflake	8
2.4. Optional Cost Monitoring Privilege	10
3. GenAIQ Manager	11
3.1. Create a PostgreSQL Instance	11
3.2. Enable Services in Snowflake Native App	12
3.3. Manage GenAIQ and Configure Host	14
3.4. Configure LiteLLM Credentials	14
3.4.1. Create a LiteLLM Master Key	15
3.4.2. Create a Service User and PAT in Snowflake	15
3.5. Deploy GenAIQ & Manage Services	16
3.6. Service Endpoints	17
3.7. GenAIQ Login (Keycloak)	18

1. Install GenAIQ from Snowflake Marketplace

1.1. Use the correct Snowflake role

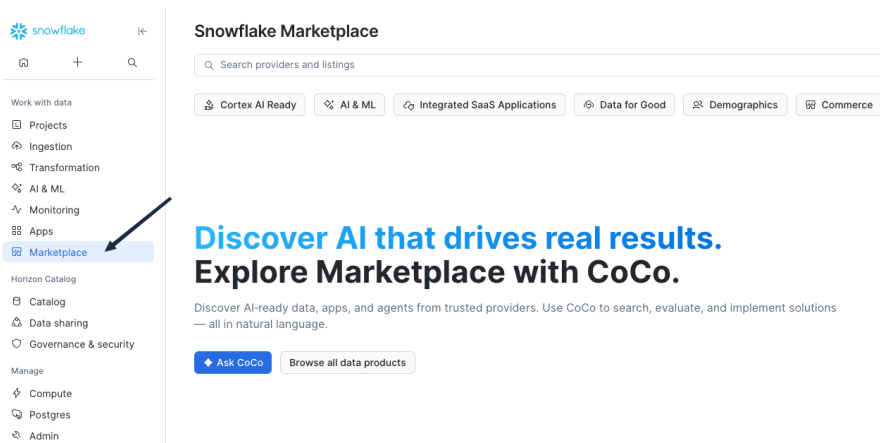
Sign in to **Snowsight** with a role that is allowed to install marketplace apps.

Recommended role:  **ACCOUNTADMIN**

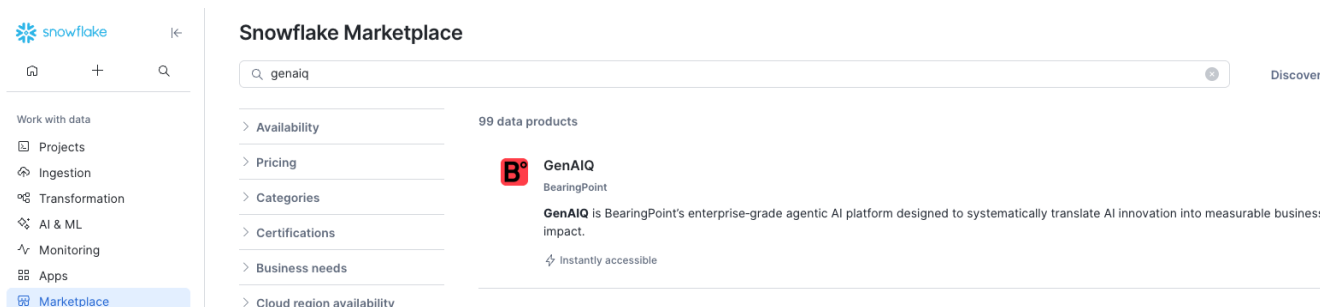
Alternatively, use a custom admin role that has the required marketplace and application privileges. Snowflake states that installing apps from listings requires the appropriate installation privileges, and paid listings may also require purchase privileges.

1.2. Open Snowflake Marketplace

In **Snowsight**, go to: Marketplace → Snowflake Marketplace

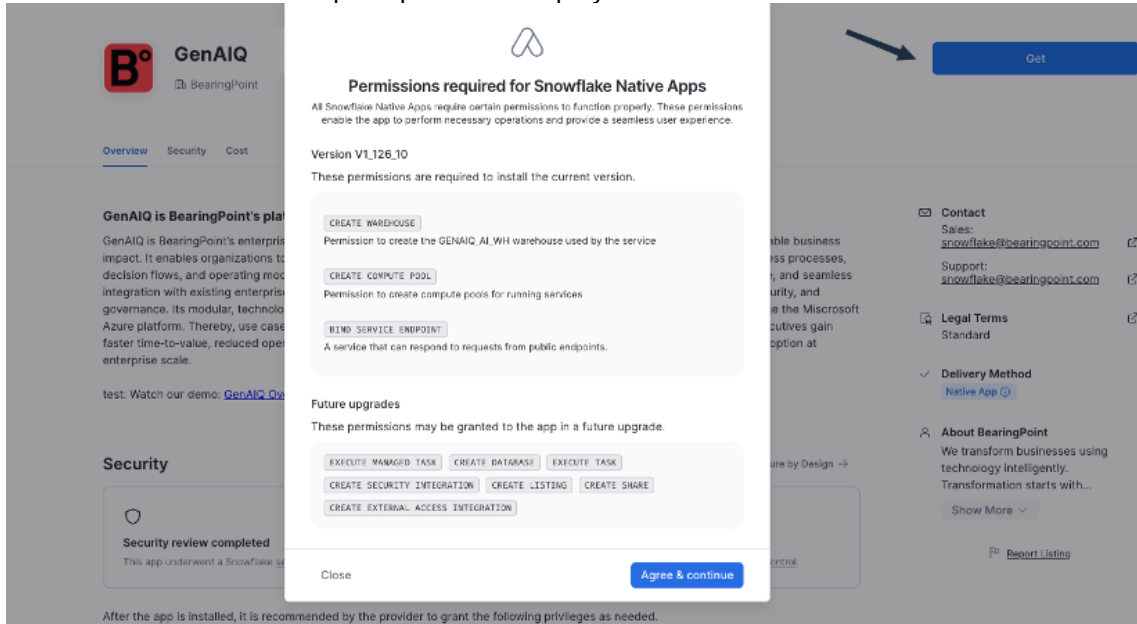


and search for: **GenAIQ**.



1.3. Install the application

Click on **GET** and the next prompt will be displayed:



For this version, GenAIQ requires the following permissions to complete the installation:

- **CREATE WAREHOUSE**
Permission to create the GENAIQ_AI_WH warehouse used by the service.
- **BIND SERVICE ENDPOINT**
Allows the app service to respond to requests from public endpoints.
- **CREATE COMPUTE POOL**
Allows the app to create compute pools required to run its services.

These permissions are required for the Snowflake Native App to function properly.

After the app is installed, the provider also recommends granting additional privileges as needed. In the post-installation view, Snowflake indicates that the app may need the following account-level privilege:

- **IMPORTED PRIVILEGES**
Required to access Snowflake account-level metadata, such as Account Usage views, for monitoring purposes.

In addition, the app may require object-level privileges in the Snowflake account, including:

- **USAGE** on **External Access Integration**
- **USAGE** on **PostgreSQL External Access**

- **READ** and **USAGE** on **PostgreSQL Password Secret**
- **READ** and **USAGE** on **PostgreSQL CA Certificate**

These object-level permissions appear to be related to enabling external connectivity, specifically access to a PostgreSQL external integration and the associated secrets or certificates.

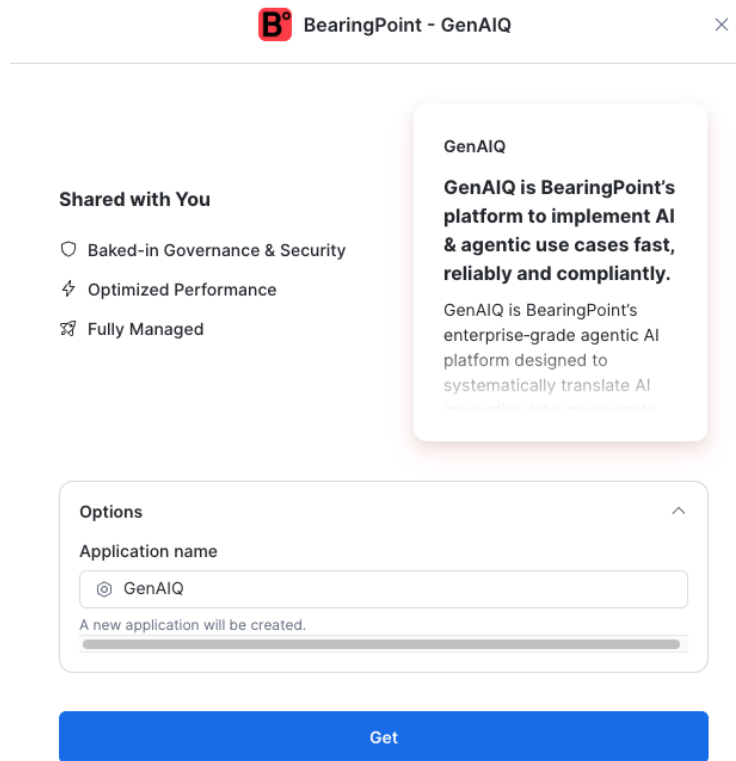
There are also cost considerations associated with the GenAIQ Native App:

- **Compute pool costs**
The app may create compute pools in the Snowflake account. Snowflake standard credits will be used to pay for these compute pools while the app is running.
- **Warehouse costs**
Snowflake standard credits will also be used for the processing time consumed by the virtual warehouse used to execute queries on this data product.
- **Storage costs**
The application object and any objects created by the application may incur Snowflake storage costs.

At this stage, the installation requires approval of the permissions for the current version. The additional post-installation privileges should be reviewed and granted only if they are required for the intended GenAIQ configuration.

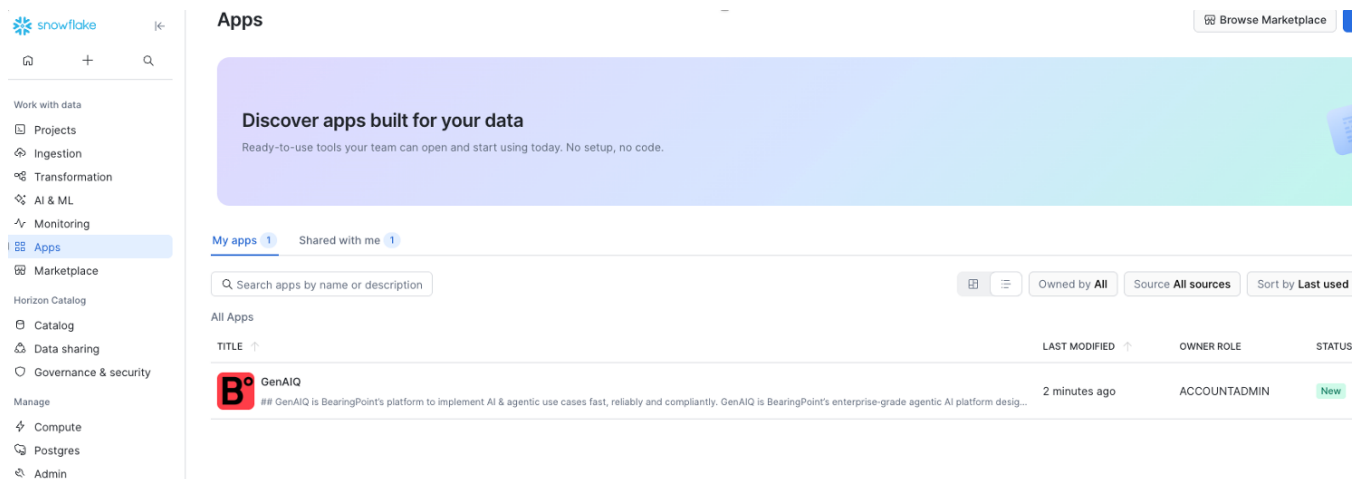
Please review the requested permissions and, if approved, click **Agree & continue** to proceed with the GenAIQ installation.

Once agreed, a new screen will pop-up with a short summary and the option of renaming the GenAIQ application.



1.4. Check the application in Snowflake

Once installed, the application should be visible in Snowflake -> Apps -> My apps. A conformation email will be sent on the email address of the Snowflake account.



2. GenAIQ Snowflake Native Application

GenAIQ is an AI-powered document assistant that helps accelerate business and administrative workflows. It runs entirely inside the consumer's Snowflake account using Snowpark Container Services (SPCS) and an external PostgreSQL database.

Features include:

- **Chat with Your Data:** Retrieve specific information from uploaded documents and knowledge bases
- **Summarizer:** Summarize entire documents for quick insights
- **Vector Analytics:** Advanced embedding and similarity search capabilities
- **PostgreSQL Integration:** Persistent data storage with relational database support

Prerequisites

Before running the setup wizard, prepare the following in your Snowflake account.

2.1. External PostgreSQL Instance

GenAIQ uses an external PostgreSQL database as its backend data store for:

- Document metadata and knowledge base management
- Session and interaction history
- Vector embeddings and similarity search indices
- Runtime configuration and service state

Requirements:

- PostgreSQL 12 or higher
- Network accessible from Snowflake Container Services compute nodes
- A dedicated PostgreSQL user with `CONNECT`, `CREATE SCHEMA` and `CREATE TABLE`
- PostgreSQL host and port (for example: `db-host.example.com:5432`)

2.2. Snowflake Secrets for PostgreSQL

You need two Snowflake secrets: one for the PostgreSQL password and one for the PostgreSQL CA certificate.

First, you can list existing secrets:

```
SHOW SECRETS IN ACCOUNT;
```

Create password secret (example):

```
CREATE OR REPLACE SECRET GENAIQ_DB.SECURITY.PG_PASSWORD  
  TYPE = GENERIC_STRING  
  SECRET_STRING = 'replace-with-your-postgres-password';
```

Create CA certificate secret (example):

```
CREATE OR REPLACE SECRET GENAIQ_DB.SECURITY.PG_CA_CERT
  TYPE = GENERIC_STRING
  SECRET_STRING = '-----BEGIN CERTIFICATE-----
MIIB...YOUR_FULL_POSTGRES_CA_CERT_HERE...ABCD
-----END CERTIFICATE-----';
```

Important:

- Paste the full certificate including `BEGIN CERTIFICATE` and `END CERTIFICATE` lines.
- The PostgreSQL CA certificate can be downloaded from Snowsight: **Manage** → **Postgres** → **:** → **Download Certificate** (see section 3.1, Step 4). The certificate is valid at account level.
- In the app configuration screen, bind these to the app references for PostgreSQL credentials and certificate.

a. LiteLLM And Snowflake PAT Setup

GenAIQ uses LiteLLM as a model gateway. You must provide:

- A LiteLLM master key (any strong secret string)
- A Snowflake PAT generated for a dedicated service user

2.3.1. Create a LiteLLM Master Key (example)

Generate a strong key (example command):

```
openssl rand -base64 48
```

Example output (do not use this exact value):

```
LITELLM_MASTER_KEY_EXAMPLE_kY6V8s5oP3wD2jN9qR4uB7mX1tZ0cA6eH3pL8vF2gQ
```

You will paste this value into the setup wizard field `LiteLLM Master Key`.

2.3.2. Create a Service User and PAT In Snowflake

Run the following as an account administrator (adjust names and grants for your environment):

```
-- 1) Role for service user
CREATE ROLE IF NOT EXISTS GENAIQ_SERVICE_ROLE
    COMMENT = 'Role for GenAIQ service user';

-- 2) Cortex access grants
GRANT DATABASE ROLE SNOWFLAKE.CORTEX_USER TO ROLE GENAIQ_SERVICE_ROLE;
GRANT DATABASE ROLE SNOWFLAKE.CORTEX_REST_API_USER TO ROLE GENAIQ_SERVICE_ROLE;
GRANT USE AI FUNCTIONS ON ACCOUNT TO ROLE GENAIQ_SERVICE_ROLE;

-- 3) Optional: network rule and network policy for PAT control
CREATE OR REPLACE NETWORK RULE GENAIQ_DB.SECURITY.GENAIQ_PAT_NETWORK_RULE
    TYPE = IPV4
    VALUE_LIST = ('0.0.0.0/0')
    MODE = INGRESS
    COMMENT = 'Allowed IP ranges for GenAIQ service user PAT auth';

CREATE OR REPLACE NETWORK POLICY GENAIQ_SERVICE_NETWORK_POLICY
    ALLOWED_NETWORK_RULE_LIST = ('GENAIQ_DB.SECURITY.GENAIQ_PAT_NETWORK_RULE')
    COMMENT = 'Network policy for GenAIQ service user PAT authentication';

-- 4) Service user
CREATE USER IF NOT EXISTS GENAIQ_SERVICE_USER
    TYPE = SERVICE
    DEFAULT_ROLE = GENAIQ_SERVICE_ROLE
    COMMENT = 'Service user for GenAIQ Cortex integration';

ALTER USER GENAIQ_SERVICE_USER
    SET NETWORK_POLICY = GENAIQ_SERVICE_NETWORK_POLICY;

GRANT ROLE GENAIQ_SERVICE_ROLE TO USER GENAIQ_SERVICE_USER;
ALTER USER GENAIQ_SERVICE_USER SET DEFAULT_ROLE = GENAIQ_SERVICE_ROLE;

-- 5) Minimum warehouse/database usage grants (example)
GRANT USAGE ON DATABASE GENAIQ_DB TO ROLE GENAIQ_SERVICE_ROLE;
GRANT USAGE ON SCHEMA GENAIQ_DB.PUBLIC TO ROLE GENAIQ_SERVICE_ROLE;
GRANT USAGE ON WAREHOUSE DEV_WH TO ROLE GENAIQ_SERVICE_ROLE;

-- 6) Create PAT
ALTER USER IF EXISTS GENAIQ_SERVICE_USER
    ADD PAT GENAIQ_SERVICE_PAT
        ROLE_RESTRICTION = 'GENAIQ_SERVICE_ROLE'
        DAYS_TO_EXPIRY = 90
        COMMENT = 'PAT for GenAIQ service user';

-- 7) Verify PAT exists
SHOW USER PROGRAMMATIC ACCESS TOKENS FOR USER GENAIQ_SERVICE_USER;
```

After ADD PAT, copy token_secret immediately. It is shown only once.

Example token format (fake value):

```
eyJraWQ1OiJFWEFNUEFmFiwImFsZyI6IkVMTjU2In0.eyJwIjoiriRkFLRV9QQVRfUEFZTE9BRCIsImV4cCI6MTg5MzQ1NjAwMH0.EXAMPLE_SIGNATURE_ONLY
```

You will paste this value into the setup wizard field `Snowflake PAT (OPENAI_API_KEY)`.

How GenAIQ Works

After configuration, the setup wizard deploys three container services:

- `CORE.KEYCLOAK_SVC` : authentication and identity provider
- `CORE.LITELLM_PROXY_SVC` : model API proxy used by the app
- `CORE.GENAIQ_INSTANCE` : main GenAIQ application UI and backend

Deployment sequence:

1. Configure PostgreSQL host and bind required secrets/integrations.
2. Create compute pool (`GENAIQ_COMPUTE_POOL`).
3. Deploy Keycloak service.
4. Deploy LiteLLM proxy service with your master key and PAT.
5. Deploy GenAIQ application service.
6. Open the generated endpoint URL from the Service Status section.

The setup app also provides:

- Suspend/Resume controls for services and compute pool
- Deployment and installation history
- Optional Cortex API usage cost monitoring (if account usage privilege is granted)

Optional Cost Monitoring Privilege

To use the in-app Cortex AI usage cost dashboard, grant:

```
IMPORTED PRIVILEGES ON DATABASE SNOWFLAKE DB
```

If not granted, GenAIQ still works normally, and the app will show a manual SQL query option for usage monitoring.

Quick Checklist Before You Deploy

- PostgreSQL instance is reachable from Snowflake and accepts connections.
- PostgreSQL password secret is created.
- PostgreSQL CA certificate secret is created.
- Required app references are bound in Snowsight.
- LiteLLM master key is generated.
- Snowflake service user PAT is created and copied.
- You have permissions to create/operate compute pools and services.

3. GenAIQ Manager

3.1 How to Create a PostgreSQL Instance

Step 1: Create PostgreSQL Instance in Snowflake

1. Open your Snowflake console
2. Navigate to **Data Engineering** → **Database** → **Databases**
3. Click **+ Create**
4. Select **PostgreSQL** as the database type
5. Configure the following:
 - **Name:** Give your database a meaningful name (e.g., `genaiq_db`)
 - **Host:** The system will assign this automatically
 - **Port:** Default is 5432
 - **Admin User:** Set to `snowflake_admin`
 - **Admin Password:** Create a strong password (you'll use this in the secret)
 - **Version:** Use PostgreSQL 14 or later
6. Review and click **Create**
7. Once created, copy the **Host** value provided (it will look like: `your-host.postgres.snowflake.app`)

Step 2: Create PostgreSQL Secret in Snowflake

The PostgreSQL secret stores your database credentials securely for use by the GenAI-Q application.

Using Snowflake UI:

1. Open Snowflake console and navigate to **Admin** → **Security** → **Secrets**
2. Click **+ Create** → **Secret**
3. Fill in:
 - **Name:** `PG_SECRET` (or your preferred name)
 - **Database & Schema:** Select where to store it (or use default)
 - **Secret Type:** `Username and password`
 - **Username:** `snowflake_admin`
 - **Password:** The password you set when creating the PostgreSQL instance
4. Click **Create**

Using SQL (alternative):

```
CREATE SECRET <database_name>.<schema_name>.PG_SECRET
TYPE = PASSWORD
USERNAME = 'snowflake_admin'
PASSWORD = '<your_postgres_password>'
```

```
COMMENT = 'PostgreSQL credentials for GenAI-Q application';
```

Step 3: Copy PostgreSQL Host

1. Go back to your PostgreSQL instance details
2. Copy the **Host** value (e.g., `your-host.postgres.snowflake.app`)
3. Paste it in the "Configure Host" tab above along with the port (e.g., `your-host.postgres.snowflake.app:5432`)

Step 4: Download the PostgreSQL CA Certificate

1. In Snowsight, navigate to **Manage** → **Postgres**
2. Click the ⋮ (more options) button next to **+ Create**
3. Select **Download Certificate**

Postgres

Status All Statuses ▾



4. **Note:** This CA certificate is valid at **account level** – it applies to all PostgreSQL instances in the Snowflake account, so it only needs to be downloaded once.
5. Open the downloaded file, copy its full content (including the `BEGIN CERTIFICATE` and `END CERTIFICATE` lines) and use it as the `SECRET_STRING` of the PostgreSQL CA certificate secret (see section 2.2).
6. Bind the CA certificate secret to the **PostgreSQL CA Certificate** app reference (same as the PostgreSQL password secret).

Important Notes:

- **Database User:** Must be `snowflake_admin`
- **Minimum Permissions:** The user needs permission to create tables and schemas
- **Network Access:** The PostgreSQL instance must be accessible from Snowflake compute pools
- **Secret Binding:** After creating the secret, you'll need to bind it to the GenAI-Q app reference (see "Enable Services" tab)

3.2 Enable Services in Snowflake Native App

After Configuring PostgreSQL Host

Once you've saved your PostgreSQL host, in the "Configure Host" tab above, you must complete the following steps in Snowflake:

Step 1: Open GenAI-Q App Configuration

1. Go to **Admin** → **Integrations & Security** → **Applications**
2. Find **GenAI-Q** in the list of installed Native Apps
3. Click on **GenAI-Q** to open the app details page
4. Click the **Configuration** tab at the top

Step 2: Bind PostgreSQL Secret Reference

1. Look for the **Reference Bindings** section
2. Find `pg_secret` in the list
3. Click **Configure** next to it
4. In the dropdown, select the PostgreSQL secret you created (e.g., `PG_SECRET`)
5. Click **Save**
6. Wait for the binding to complete (this may take 30 seconds)

Step 3: Bind External Access Integration (EAI)

1. In the same **Configuration** page, find the **External Access Integration** section
2. Look for `pg_eai` reference
3. Click **Configure** next to it
4. Select or create an External Access Integration that allows connectivity to your PostgreSQL instance
 - If you don't have one, you can create it by clicking **+ Create New**
 - Configure it to allow outbound connections to your PostgreSQL host on port 5432
5. Click **Save**
6. Wait for the binding to complete

Step 4: Verify Configuration

1. Once both bindings are complete (green checkmarks), your app is ready
2. Return to this setup wizard and continue with Step 1 (Configure Secrets & References)
3. You should see a success message confirming the secret reference is configured

Troubleshooting

- **Secret not visible?** Ensure you created it in the correct database/schema accessible to the Native App
 - **EAI not connecting?** Check that your PostgreSQL instance is accessible from Snowflake's network
 - **Still seeing warnings?** Refresh this page after completing the bindings in Snowsight
-

3.3 Manage GenAIQ and Configure Host



Manager

Manage your running GenAIQ service, compute pool, and configure additional settings.

Service Status & Endpoint

Service Status: ● RUNNING

Service Endpoint for accessing GenAIQ App:

<genaiq-instance-account>.snowflakecomputing.app

Manage Service & Compute Pool

Suspend
Resume

3.4 Configure LiteLLM Credentials

LiteLLM Credentials ↔

[configure LiteLLM](#) [Usable Models](#) [Enable Cortex Models](#)

These credentials are required to deploy the LiteLLM proxy service. They are passed securely into the container at deploy time and never stored in plaintext.

LiteLLM Master Key

sk-my-litellm-master-key 👁

Snowflake PAT

eyJraWQOI... 👁

Save LiteLLM Credentials

Under LiteLLM Credentials, **the Usable Models** section displays the list of supported model providers, including Anthropic, DeepSeek, OpenAI, Mistral, Snowflake, Meta, and others.

LiteLLM Credentials

[configure LiteLLM](#) [Usable Models](#) [Enable Cortex Models](#)

Supported model names for LiteLLM / Cortex routing

Use these exact values when enabling model.

Enable Cortex Models

LiteLLM And Snowflake PAT Setup

GenAIQ uses LiteLLM as a model gateway. You must provide:

- A LiteLLM master key (any strong secret string)
- A Snowflake PAT generated for a dedicated service user

3.4.1. Create a LiteLLM Master Key (example)

Generate a strong key (example command):

```
openssl rand -base64 48
```

Example output (do not use this exact value):

```
LITELLM_MASTER_KEY_EXAMPLE_kY6V8s5oP3wD2jN9qR4uB7mX1tZ0cA6eH3pL8vF2gQ
```

You will paste this value into the setup wizard field `LiteLLM Master Key`.

3.4.2. Create a Service User And PAT In Snowflake

Run the following as an account administrator (adjust names and grants for your environment but do not change the name of the role: `GENAIQ_SERVICE_ROLE`):

```
-- 1) Role for service user
CREATE ROLE IF NOT EXISTS GENAIQ_SERVICE_ROLE
    COMMENT = 'Role for GenAIQ service user';

-- 2) Cortex access grants
GRANT DATABASE ROLE SNOWFLAKE.CORTEX_USER TO ROLE GENAIQ_SERVICE_ROLE;
GRANT DATABASE ROLE SNOWFLAKE.CORTEX_REST_API_USER TO ROLE GENAIQ_SERVICE_ROLE;
GRANT USE AI FUNCTIONS ON ACCOUNT TO ROLE GENAIQ_SERVICE_ROLE;

-- 3) Optional: network rule and network policy for PAT control
CREATE OR REPLACE NETWORK RULE GENAIQ_DB.SECURITY.GENAIQ_PAT_NETWORK_RULE
    TYPE = IPV4
    VALUE_LIST = ('0.0.0.0/0')
    MODE = INGRESS
    COMMENT = 'Allowed IP ranges for GenAIQ service user PAT auth';

CREATE OR REPLACE NETWORK POLICY GENAIQ_SERVICE_NETWORK_POLICY
    ALLOWED_NETWORK_RULE_LIST = ('GENAIQ_DB.SECURITY.GENAIQ_PAT_NETWORK_RULE')
    COMMENT = 'Network policy for GenAIQ service user PAT authentication';

-- 4) Service user
```

```
CREATE USER IF NOT EXISTS GENAIQ_SERVICE_USER
  TYPE = SERVICE
  DEFAULT_ROLE = GENAIQ_SERVICE_ROLE
  COMMENT = 'Service user for GenAIQ Cortex integration';

ALTER USER GENAIQ_SERVICE_USER
  SET NETWORK_POLICY = GENAIQ_SERVICE_NETWORK_POLICY;

GRANT ROLE GENAIQ_SERVICE_ROLE TO USER GENAIQ_SERVICE_USER;
ALTER USER GENAIQ_SERVICE_USER SET DEFAULT_ROLE = GENAIQ_SERVICE_ROLE;

-- 5) Minimum warehouse/database usage grants (example)
GRANT USAGE ON DATABASE GENAIQ_DB TO ROLE GENAIQ_SERVICE_ROLE;
GRANT USAGE ON SCHEMA GENAIQ_DB.PUBLIC TO ROLE GENAIQ_SERVICE_ROLE;
GRANT USAGE ON WAREHOUSE DEV_WH TO ROLE GENAIQ_SERVICE_ROLE;

-- 6) Create PAT
ALTER USER IF EXISTS GENAIQ_SERVICE_USER
  ADD PAT GENAIQ_SERVICE_PAT
    ROLE_RESTRICTION = 'GENAIQ_SERVICE_ROLE'
    DAYS_TO_EXPIRY = 90
    COMMENT = 'PAT for GenAIQ service user';

-- 7) Verify PAT exists
SHOW USER PROGRAMMATIC ACCESS TOKENS FOR USER GENAIQ_SERVICE_USER;
```

After ADD PAT, copy `token_secret` immediately. It is shown only once.

Example token format (fake value):

```
eyJraWQ1OjJFWEFNUExFiIiwgImFsZyI6IkVtMjU2In0.eyJwIjoiriRkFLRV9QVRfUEFZTE9BRCIsImV4
cCI6MTg5MzQ1NjAwMH0.EXAMPLE_SIGNATURE_ONLY
```

You will paste this value into the setup wizard field `Snowflake PAT`.

3.5 Deploy GenAIQ & Manage Services

Create Compute Pool

Compute Pool Name: GENAIQ_COMPUTE_POOL

Manage Service

Under **Deploy GenAIQ & Manage Services**, on Manage Service, you can interact with service by Suspending,

Resuming or Dropping it.

Manage GenAIQ Service & Compute Pool

Suspend

Resume

Drop

API Cost Monitoring

Tracks GENAIQ_SERVICE_USER model usage and estimated USD cost for the last 7, 30 or 90 days.

Note: Cortex API usage in SNOWFLAKE.ACCOUNT_USAGE is not real-time and may take some time to appear after requests are made.

Cost by Model

user_name	model_name	total_requests	total_tokens	input_tokens	output_tokens	input_cost_usd	output_cost_usd	total_cost_usd
GENAIQ_SERVICE_USER	openai-gpt-5-mini	1	4,077	3,390	687	0.0014	0.0011	0.0025

3.6 Service Endpoints

After clicking on “**Show Endpoints**”, the following URLs list will be displayed.

Service Endpoint

Show Endpoints

Keycloak Service

Keycloak URL:

https://<keycloak-service-account>.snowflakecomputing.app

LiteLLM Proxy Service

LiteLLM URL:

https://<litellm-proxy-account>.snowflakecomputing.app

GenAIQ Instance

Service URL:

https://<genaiq-instance-account>.snowflakecomputing.app

Keycloak Service – Keycloak is the **identity and access management service** for the GenAIQ app. In simple terms, **Keycloak Service is responsible for:**

- Used for authentication and user access management.
- Managing user login
- Issuing authentication tokens
- Handling single sign-on style authentication
- Controlling user sessions

- Managing roles or access policies for the GenAIQ UI/API
- Allowing the GenAIQ app and backend services to authenticate requests securely

LiteLLM Proxy Service

Likely used as the gateway/proxy for LLM model calls.

GenAIQ Instance

The main GenAIQ application/service endpoint.

3.7 GenAIQ Login (Keycloak)

GenAIQ login is handled by **Keycloak**. After deployment, open the GenAIQ / Keycloak URL from the Service Endpoints section and sign in with the default credentials:

1. **Username:** admin
2. **Password:** changeme

Important: Change the default password immediately after the application is installed, before giving access to any other users.

How to change the password:

1. Open the **Keycloak URL** and log in with admin / changeme
2. Go to **Users** → select **admin** → **Credentials** tab → **Reset password**
3. Set a strong password, turn **Temporary** off and click **Save**
4. Log out and log in again with the new password to confirm the change
5. Store the new password securely (e.g., in your organization's password vault).