



Bluesky Mission Control

Quick Start

Connecting Bluesky to Snowflake

Last Update on **Aug 30, 2022**

Bluesky Mission Control enables enterprises to gain visibility into the workloads, usage, and cost of Snowflake. It also helps identify and optimize workloads to help enterprises improve the efficiency and performance of Snowflake.

This quick start document contains steps to get started with Bluesky Mission Control. The instructions below help you connect Bluesky to your Snowflake cloud deployment.

Summary

Bluesky connects to Snowflake via the [Snowflake Database](#) which contains metadata like historical queries, usage metrics, organizations and accounts. Bluesky uses such metadata to analyze underlying workloads, provide granular reporting, and provide efficiency optimization suggestions to improve performance and ROI of Snowflake.

Bluesky requires **read-only access** to the Snowflake Database and thus, will **not** be able to mutate any data in it. Bluesky requires **no** access to the actual business data (e.g. tables, materialized view, schema) in your Snowflake instance.

Set Up Instructions

- Please follow the instructions in **either** Fast Mode or Detailed Mode



Please make sure you are running commands in the role ACCOUNTADMIN (refer to [How to Switch To ACCOUNTADMIN Role](#)).

💡 To run the commands, you could simply copy & paste the following commands and directly run in the Snowflake console using ACCOUNTADMIN role (refer to [How to Run Commands in Snowflake UI Console](#)).

Fast Mode

The fastest way to set up the Bluesky connection is to use the Snowflake UI console. Simply copy & paste all the commands, select all commands, and click “Run”.



```
1 -- Step 1. Create a read only role called BLUESKY_READONLY for bluesky Data
2 create role BLUESKY_READONLY
3   comment = 'This role has read only permissions for Bluesky Data';
4
5 -- Step 2. Grant privileges on SNOWFLAKE DB to the BLUESKY_READONLY role, such that we can access the data in Snowflake for analysis
6 grant IMPORTED PRIVILEGES
7   on database SNOWFLAKE
8   to role BLUESKY_READONLY;
9
10
11 -- Step 3. Create a medium sized warehouse called 'bluesky_test_wh'
12 CREATE WAREHOUSE bluesky_test_wh WITH WAREHOUSE_SIZE = 'MEDIUM' WAREHOUSE_TYPE = 'STANDARD' AUTO_SUSPEND = 600 AUTO_RESUME = TRUE MIN_CLUSTER_COUNT = 1 MAX_CLUSTER_COUNT = 2 SCALING_POLICY = 'STANDARD';
13
14 -- Step 4. Grant the usage of the warehouse 'bluesky_test_wh' to the BLUESKY_READONLY role
15 grant usage
16   on WAREHOUSE bluesky_test_wh
17   to role BLUESKY_READONLY;
18
19
20 -- Step 5. Help us create an account (user_name is bluesky_test, password is 123)
21 CREATE USER bluesky_test PASSWORD = '123' DEFAULT_ROLE = BLUESKY_READONLY
22 MUST_CHANGE_PASSWORD = TRUE;
23
24 -- Step 6. Grant the BLUESKY_READONLY role to user bluesky_test
25 GRANT ROLE BLUESKY_READONLY TO USER bluesky_test;
26
27
28 -- Step 7. Set the default role as read_only for safety
29 alter user bluesky_test
30   set default_role = BLUESKY_READONLY;
31
32
33 -- Step 8. Create a DB ('bluesky_test_db') for us and grant us write access to the DB. The intuition is that based on our infrastructure, this step will make it easier for us to define our own views (e.g., sampling, or filter)
34 create DATABASE bluesky_test_db;
35
```

Script to Copy

Copy and paste the following into the Snowflake Console:

```
-- Step 1. Create a new role for Bluesky Data
CREATE ROLE BLUESKY_READONLY
  comment = 'This role has read only permissions for Bluesky Data';

-- Step 2. Give our role privileges on the internal SNOWFLAKE to allow us to analyze
historical queries and metadata
GRANT IMPORTED PRIVILEGES
  ON DATABASE SNOWFLAKE
  TO ROLE BLUESKY_READONLY;

-- Step 3. Create a small sized Warehouse
CREATE WAREHOUSE bluesky_wh WITH WAREHOUSE_SIZE = 'SMALL' WAREHOUSE_TYPE = 'STANDARD'
AUTO_SUSPEND = 60 AUTO_RESUME = TRUE MIN_CLUSTER_COUNT = 1 MAX_CLUSTER_COUNT = 2
SCALING_POLICY = 'STANDARD';

-- Step 4. Give our role USAGE on the Warehouse
GRANT USAGE ON WAREHOUSE bluesky_wh TO ROLE BLUESKY_READONLY;

-- Step 5. Create an user (user_name is bluesky_test, password is 123)
CREATE USER bluesky_test PASSWORD = '123' DEFAULT_ROLE = BLUESKY_READONLY
MUST_CHANGE_PASSWORD = TRUE;
```

```

-- Step 6. Grant the role to user
GRANT ROLE BLUESKY_READONLY TO USER bluesky_test;

-- Step 7. Set the default role as read_only for safety reasons
ALTER USER bluesky_test SET default_role = BLUESKY_READONLY;

-- Step 8. Create a DB for us and grant us write access to the DB. This step allows us
define our own views (e.g., sampling, or filters) for analysis purposes.
CREATE DATABASE bluesky_test_db;

-- Step 9. Grant all privileges of the newly created DB ("bluesky_test_db") to role
BLUESKY_READONLY
GRANT all PRIVILEGES ON DATABASE bluesky_test_db TO ROLE BLUESKY_READONLY;

-- Step 10. Grant monitoring permission to role BLUESKY_READONLY
GRANT monitor USAGE ON account TO ROLE BLUESKY_READONLY;

```

Detailed Steps

The following section contains the detailed commands on how to grant read-only roles to the Snowflake metadata database (identical to above section, but separate).

Step 1

Create a read only role called BLUESKY_READONLY for Bluesky Data

```

CREATE ROLE BLUESKY_READONLY
  comment = 'This role has read only permissions for Bluesky Data';

```

Step 2

Grant privileges on SNOWFLAKE DB to the BLUESKY_READONLY role, such that we can access the data in Snowflake for analysis

```

GRANT IMPORTED PRIVILEGES ON DATABASE SNOWFLAKE TO ROLE BLUESKY_READONLY;

```

Step 3

Create a small sized warehouse called "bluesky_wh"

```

CREATE WAREHOUSE bluesky_wh WITH WAREHOUSE_SIZE = 'SMALL' WAREHOUSE_TYPE = 'STANDARD'
AUTO_SUSPEND = 60 AUTO_RESUME = TRUE MIN_CLUSTER_COUNT = 1 MAX_CLUSTER_COUNT = 2
SCALING_POLICY = 'STANDARD';

```

Step 4

Grant the usage of the warehouse "bluesky_wh" to the BLUESKY_READONLY role

```
GRANT USAGE ON WAREHOUSE bluesky_wh TO ROLE BLUESKY_READONLY;
```

Step 5

Help us create an account (user_name is bluesky_test, password is 123)

```
CREATE USER bluesky_test PASSWORD = '123' DEFAULT_ROLE = BLUESKY_READONLY  
MUST_CHANGE_PASSWORD = TRUE;
```

Step 6

Grant the BLUESKY_READONLY role to user bluesky_test

```
GRANT ROLE BLUESKY_READONLY TO USER bluesky_test;
```

Step 7

Set the default role as read_only for safety

```
ALTER USER bluesky_test SET default_role = BLUESKY_READONLY;
```

Step 8

Create a DB ("bluesky_test_db") for us and grant us write access to the DB. The intuition is that based on our infrastructure, this step will make it easier for us to define our own views (e.g., sampling, or filters) over the data with large volume, which makes it flexible for us to perform analysis. Also, creating a new testing DB will make sure our write operations only happen in this test DB, decoupling with other product DBs.

```
CREATE DATABASE bluesky_test_db;
```

Step 9

Grant all privileges of the newly created DB ("bluesky_test_db") to role BLUESKY_READONLY

```
GRANT all PRIVILEGES ON DATABASE bluesky_test_db TO ROLE BLUESKY_READONLY;
```

Step 10

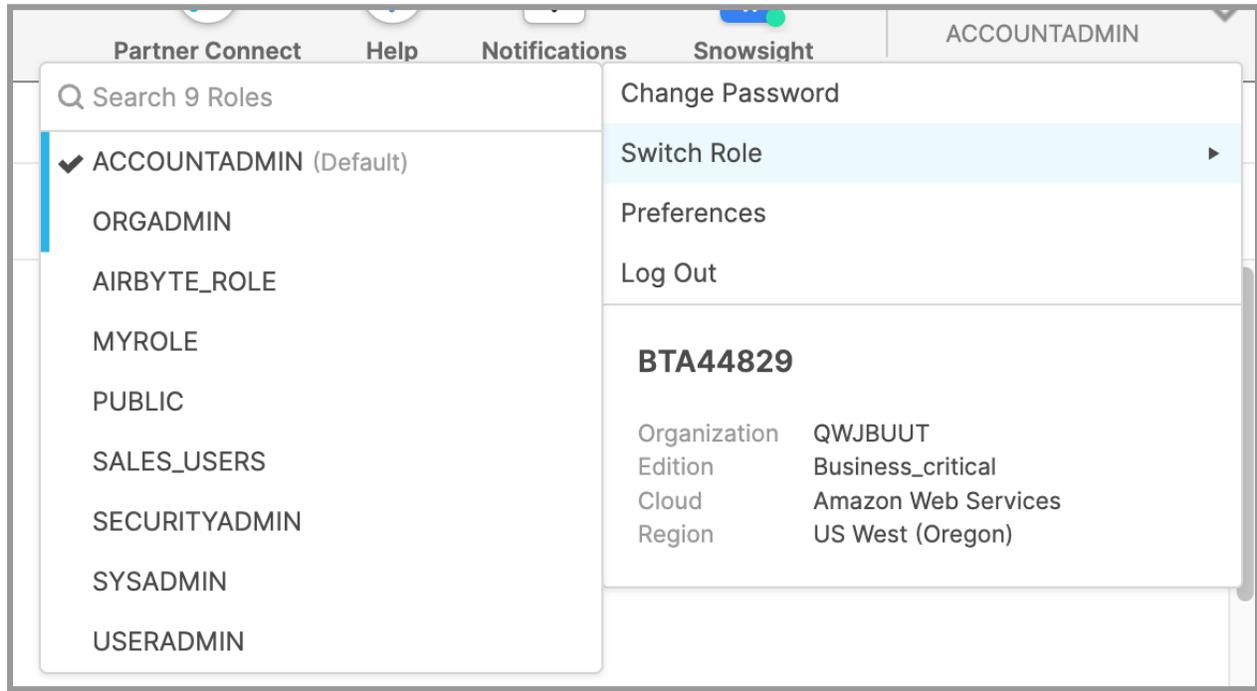
Allow BLUESKY_READONLY to monitor warehouse settings

```
GRANT monitor USAGE ON account TO ROLE BLUESKY_READONLY;
```

Appendix

How to Switch To ACCOUNTADMIN Role

It is easy to switch to ACCOUNTADMIN role directly from UI. Basically you could click on the button near the user-name in the top right corner, and select the ACCOUNTADMIN role.



If you cannot see this role, better to contact the Snowflake admin in your organization to run the commands, or help grant you the permission of the role:

```
GRANT ROLE ACCOUNTADMIN TO USER XXXX;
```