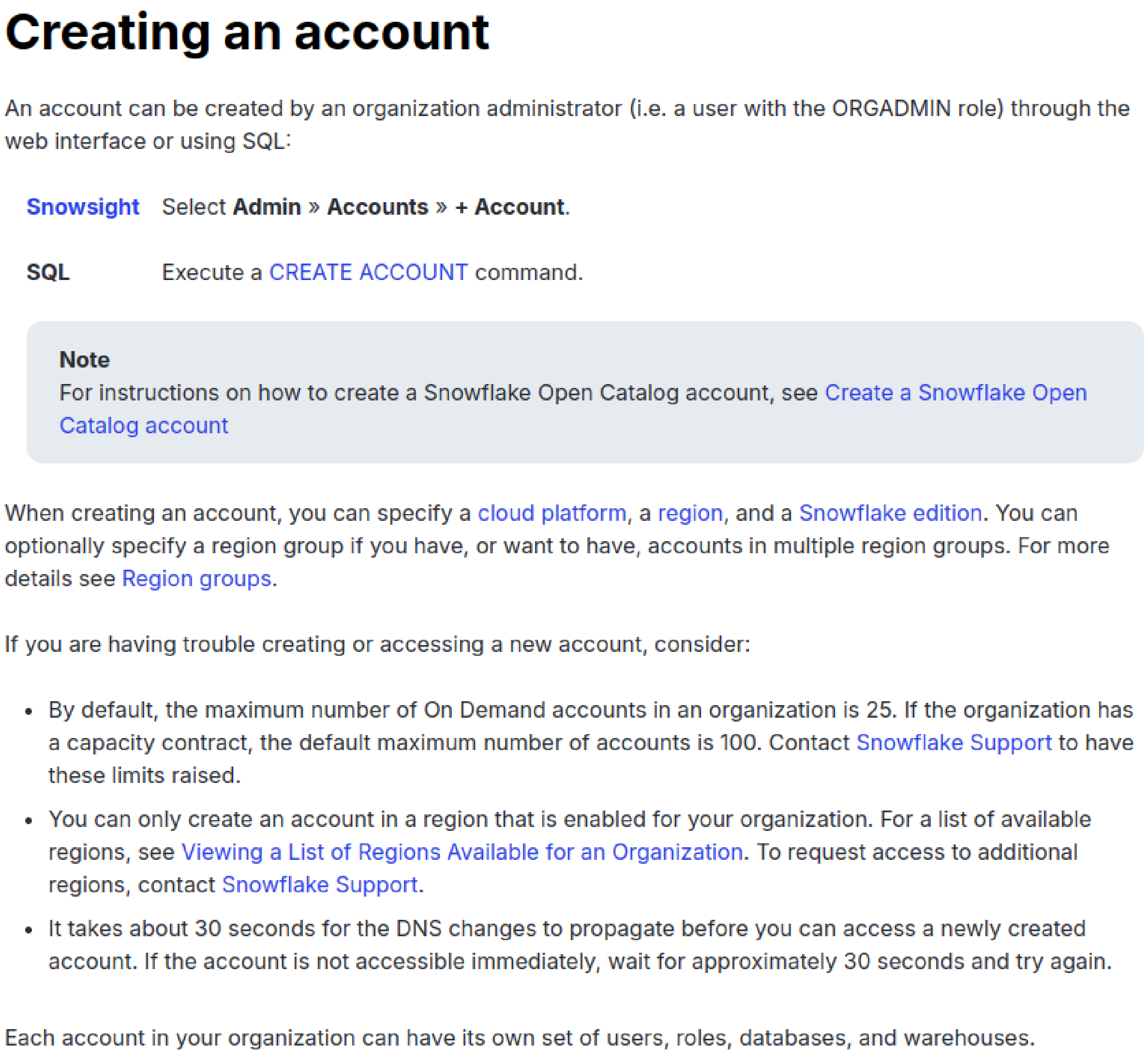
In this lab, we are going to focus on managing accounts in Snowflake. This will include the following:

- Creating Accounts
- Creating Organization Accounts
- Dropping Accounts
- Restoring Accounts
- Password Policies



Creating an account

An account can be created by an organization administrator (i.e. a user with the ORGADMIN role) through the web interface or using SQL:

Snowsight	Select **Admin** » **Accounts** » **+ Account**.
SQL	Execute a CREATE ACCOUNT command.

Note
For instructions on how to create a Snowflake Open Catalog account, see Create a Snowflake Open Catalog account

When creating an account, you can specify a cloud platform, a region, and a Snowflake edition. You can optionally specify a region group if you have, or want to have, accounts in multiple region groups. For more details see Region groups.

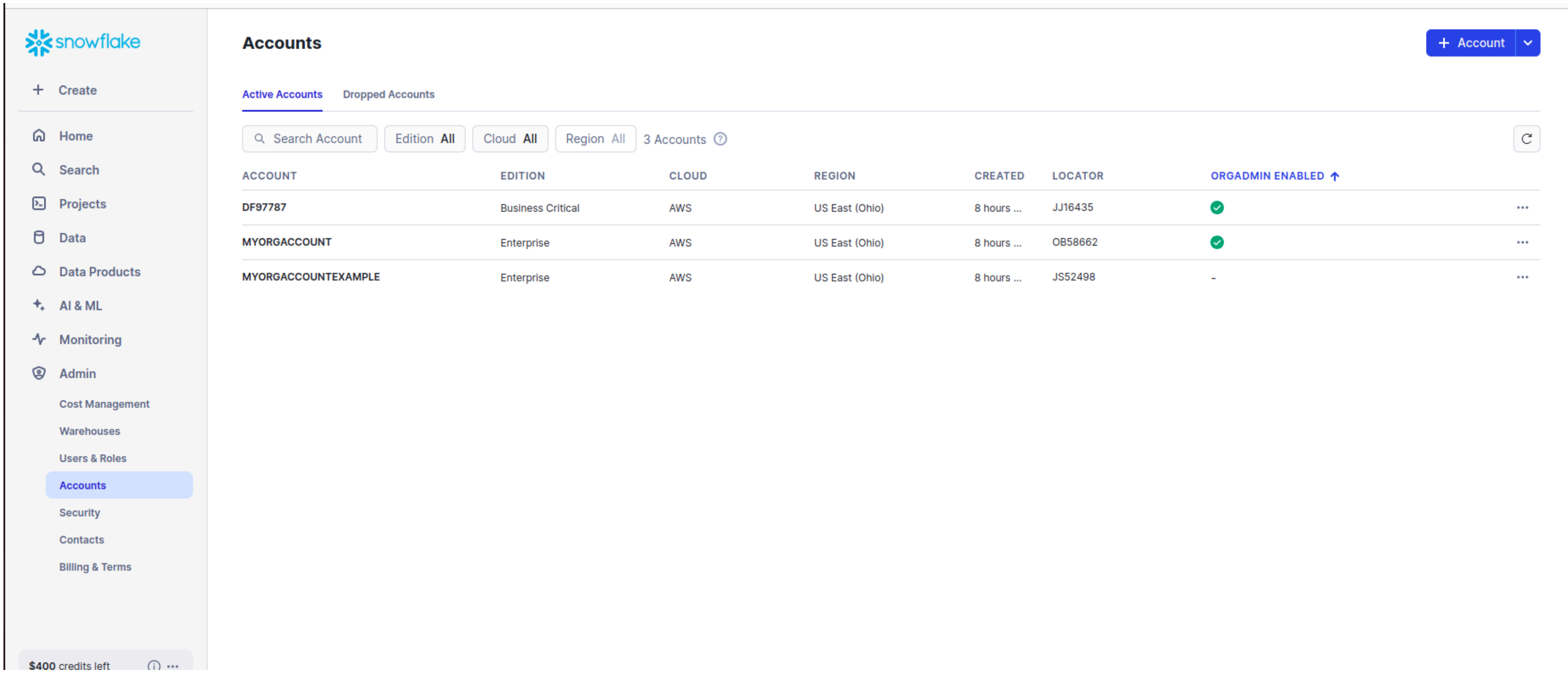
If you are having trouble creating or accessing a new account, consider:

- By default, the maximum number of On Demand accounts in an organization is 25. If the organization has a capacity contract, the default maximum number of accounts is 100. Contact Snowflake Support to have these limits raised.
- You can only create an account in a region that is enabled for your organization. For a list of available regions, see Viewing a List of Regions Available for an Organization. To request access to additional regions, contact Snowflake Support.
- It takes about 30 seconds for the DNS changes to propagate before you can access a newly created account. If the account is not accessible immediately, wait for approximately 30 seconds and try again.

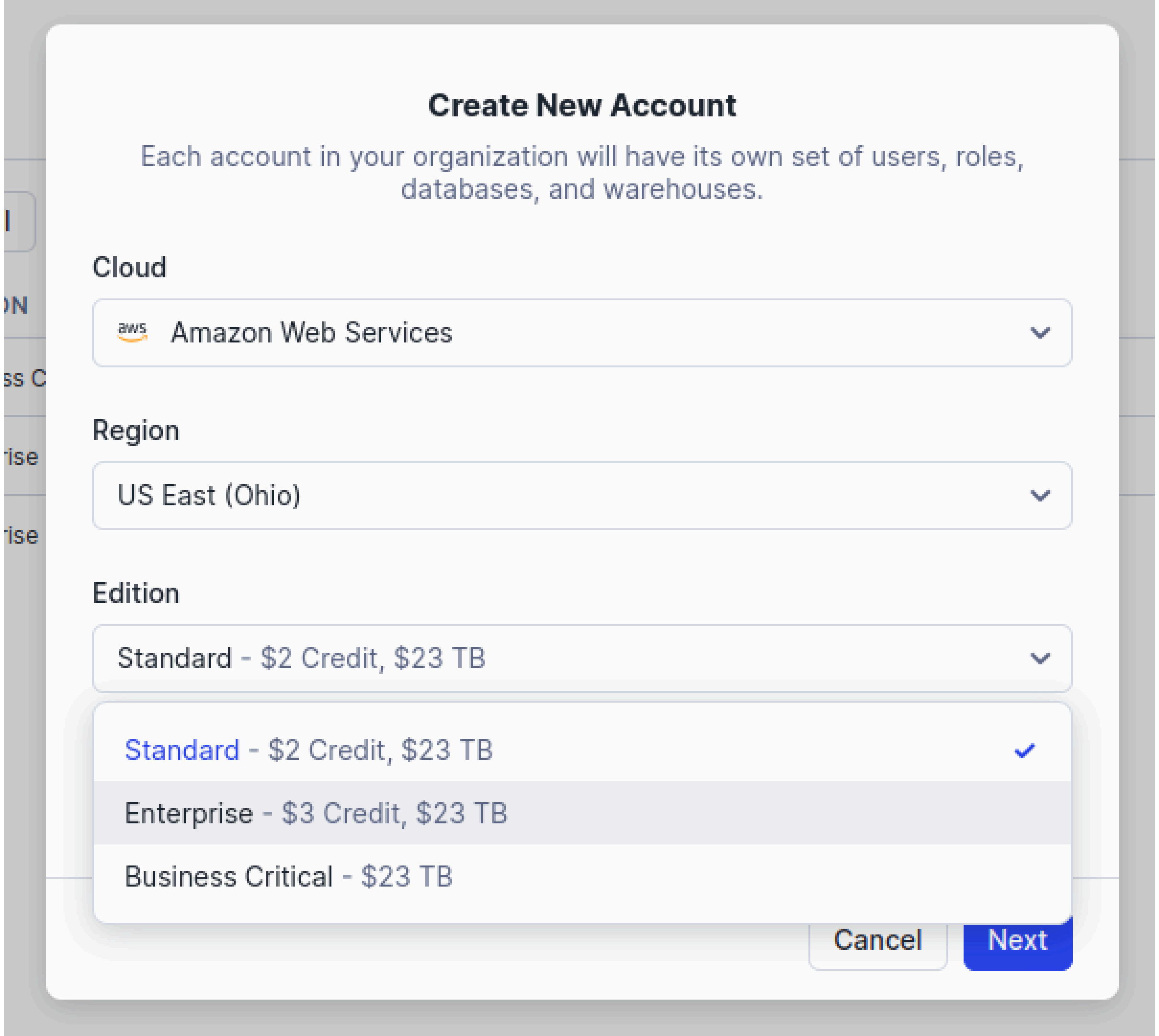
Each account in your organization can have its own set of users, roles, databases, and warehouses.

Let’s first create an account using the Snowsight interface.

- To do so, we are going to go to **Admin -> Accounts -> +Account**
- Next we are going to select our account information, such as name, cloud platform, region and edition
- Once we save, we will see the account available



ACCOUNT	EDITION	CLOUD	REGION	CREATED	LOCATOR	ORGADMIN ENABLED ↑
DF97787	Business Critical	AWS	US East (Ohio)	8 hours ...	JJ16435	✓
MYORGACCOUNT	Enterprise	AWS	US East (Ohio)	8 hours ...	OB58662	✓
MYORGACCOUNTEXAMPLE	Enterprise	AWS	US East (Ohio)	8 hours ...	JS52498	-



Create New Account

Each account in your organization will have its own set of users, roles, databases, and warehouses.

Cloud

Amazon Web Services

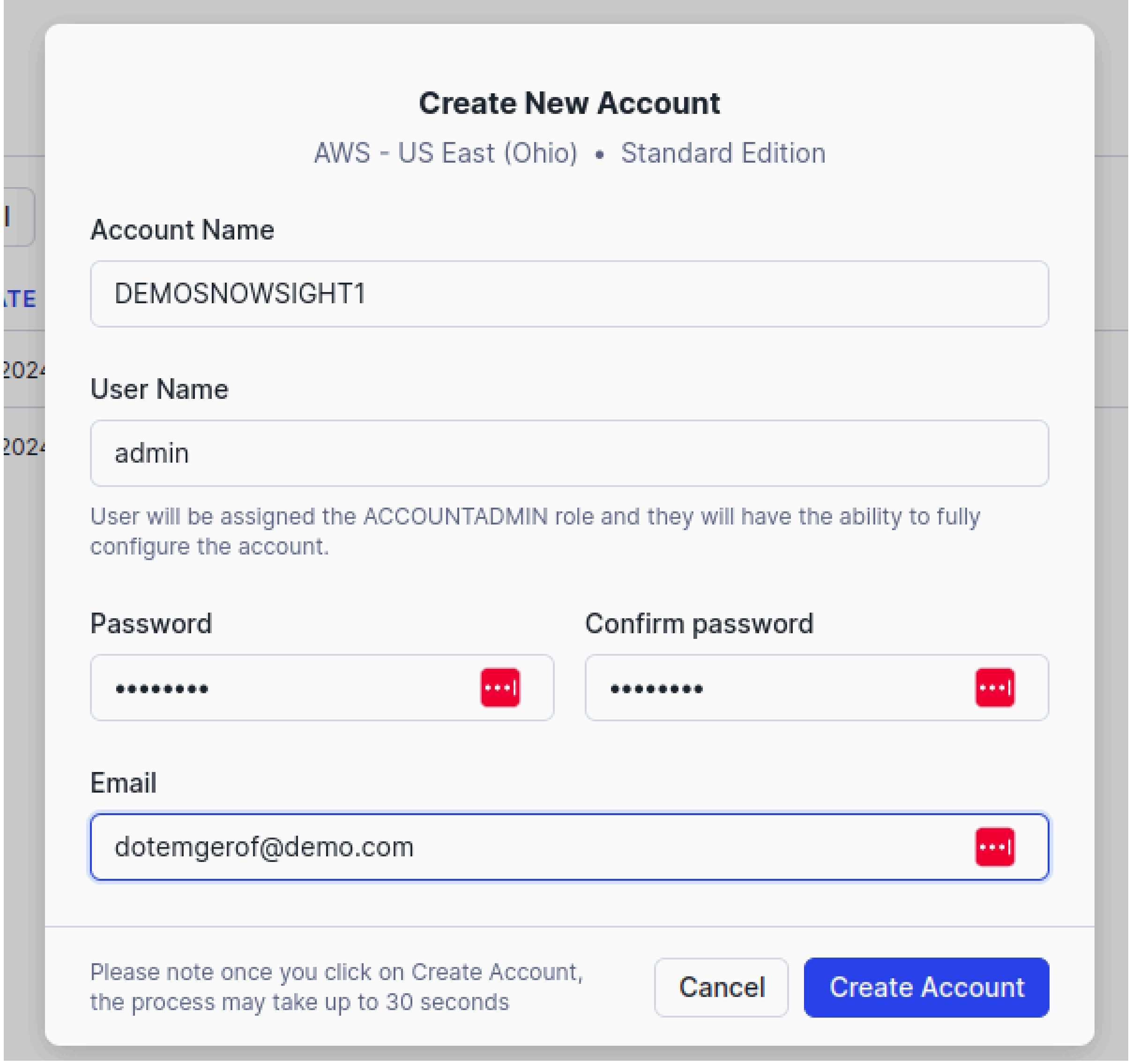
Region

US East (Ohio)

Edition

Standard - $2 Credit, $23 TB

- Standard - $2 Credit, $23 TB ✓
- Enterprise - $3 Credit, $23 TB
- Business Critical - $23 TB



Create New Account

AWS - US East (Ohio) • Standard Edition

Account Name

DEMOSNOWSIGHT1

User Name

admin

User will be assigned the ACCOUNTADMIN role and they will have the ability to fully configure the account.

Password

•••••••••

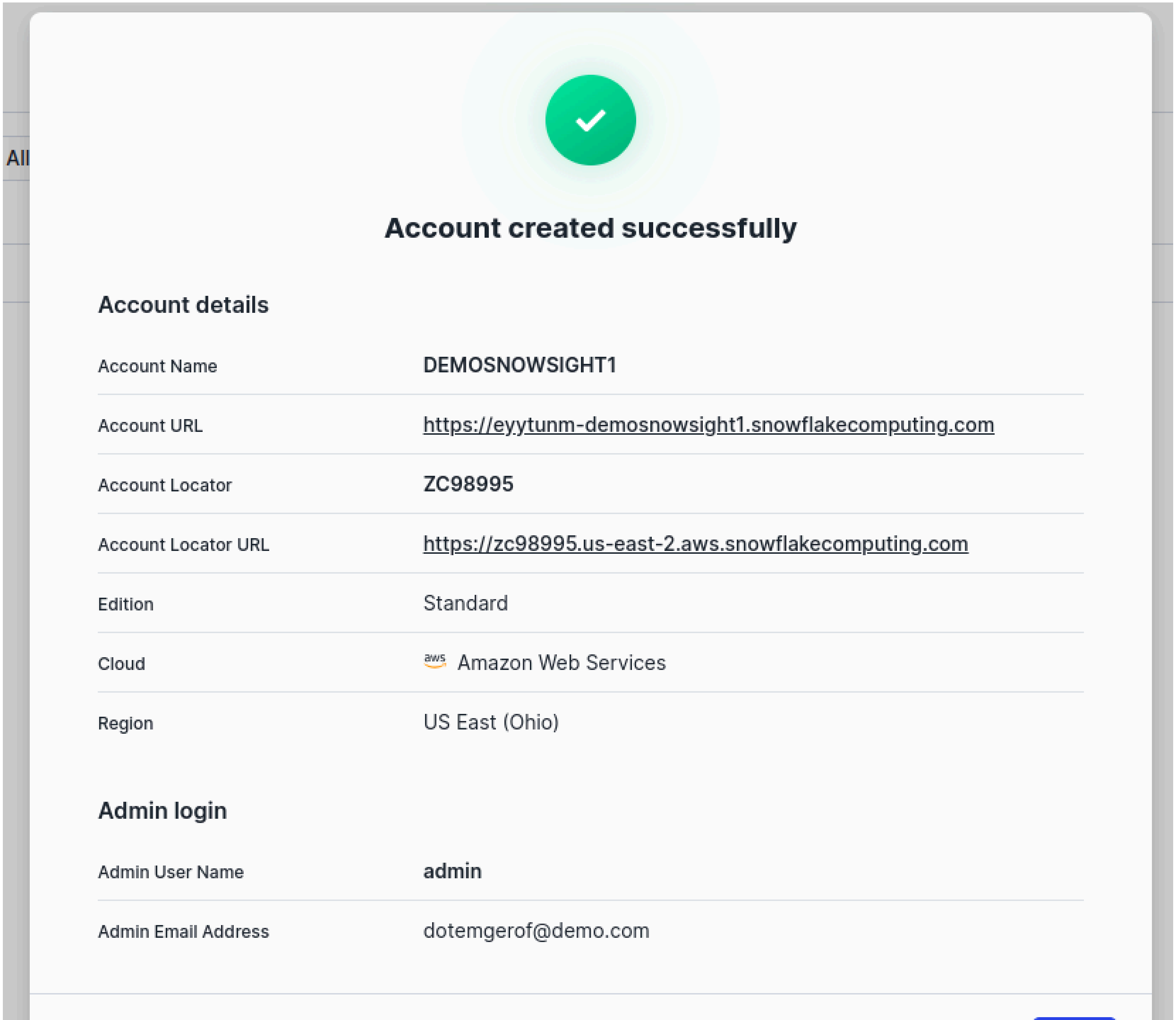
Confirm password

•••••••••

Email

dotemgerof@demo.com

Please note once you click on Create Account, the process may take up to 30 seconds



Account created successfully

Account details

Account Name	**DEMOSNOWSIGHT1**
Account URL	https://eyytunm-demosnowsight1.snowflakecomputing.com
Account Locator	**ZC98995**
Account Locator URL	https://zc98995.us-east-2.aws.snowflakecomputing.com
Edition	Standard
Cloud	aws Amazon Web Services
Region	US East (Ohio)

Admin login

Admin User Name	**admin**
Admin Email Address	dotemgerof@demo.com

You can also complete the same thing using SQL commands:

```
USE ROLE orgadmin;

--Create a regular Snowflake account
CREATE ACCOUNT DEMOSNOWSIGHT2
  ADMIN_NAME = admin
  ADMIN_PASSWORD = 'TestPassword1'
  FIRST_NAME = Jane
  LAST_NAME = Smith
  EMAIL = 'myemail43G5G45@demo.com'
  EDITION = enterprise
  REGION = aws_us_west_2;
```

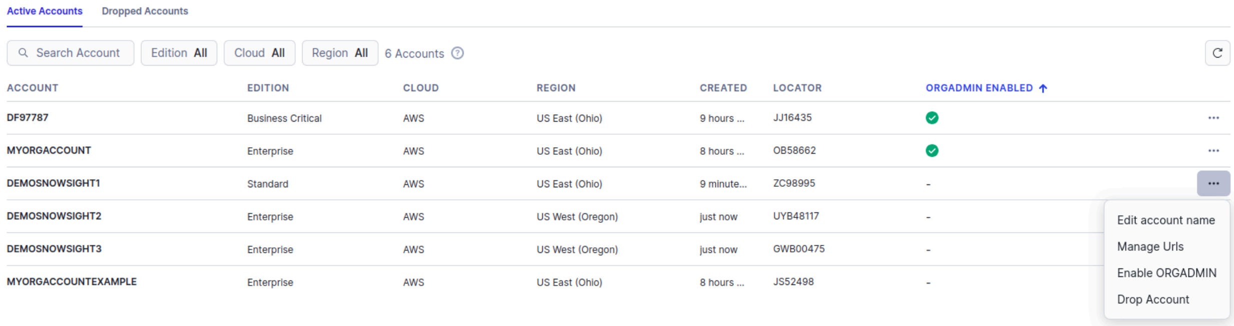
Renaming an account

An organization administrator (i.e. a user granted the ORGADMIN role) can rename an account.

When an account is renamed, Snowflake creates a new account URL that is used to access the account. During the renaming, the administrator can accept the default to save the original account URL so users can continue to use it, or they can delete the original URL to force users to use the new URL. Saved URLs can be deleted at a later time. You cannot save the original URL for a reader account.

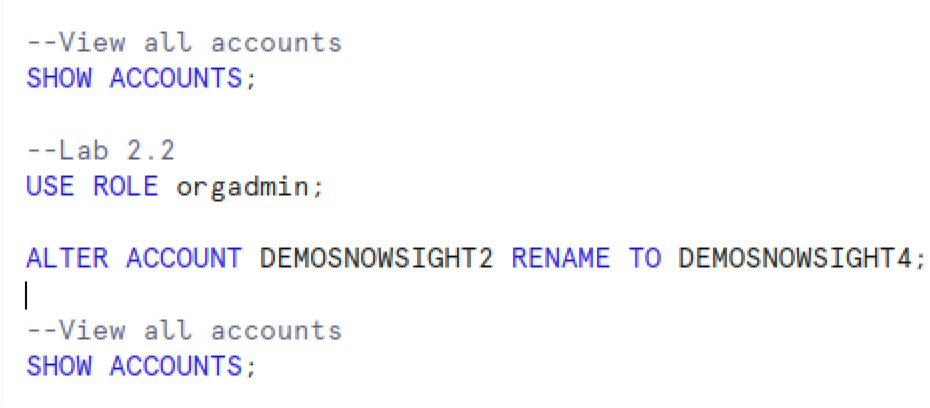
Organization administrators cannot rename an account while they are logged in to it, so they must log in to a different account before executing the renaming command. If your organization consists of a single account that needs to be renamed, contact Snowflake Support.

Note
Renaming an account has no effect on replication and failover.

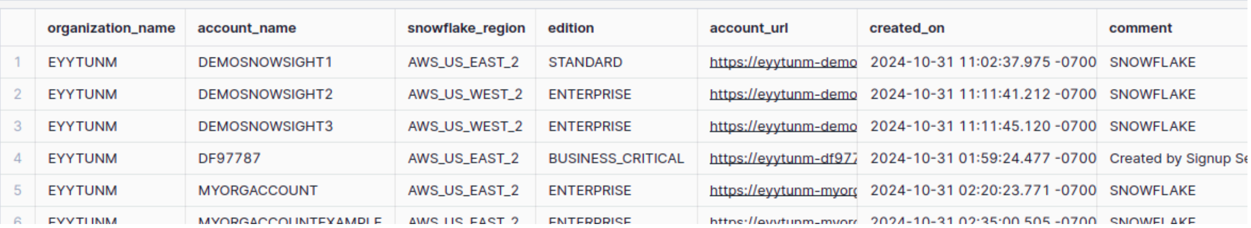


ACCOUNT	EDITION	CLOUD	REGION	CREATED	LOCATOR	ORGADMIN ENABLED ↑
DF97787	Business Critical	AWS	US East (Ohio)	9 hours ...	JJ16435	✓
MYORGACCOUNT	Enterprise	AWS	US East (Ohio)	8 hours ...	OB58662	✓
DEMOSNOWSIGHT1	Standard	AWS	US East (Ohio)	9 minute...	ZC98995	-
DEMOSNOWSIGHT2	Enterprise	AWS	US West (Oregon)	just now	UYB48117	-
DEMOSNOWSIGHT3	Enterprise	AWS	US West (Oregon)	just now	GWB00475	-
MYORGACCOUNTEXAMPLE	Enterprise	AWS	US East (Ohio)	8 hours ...	JS52498	-

You can use the following commands to rename an account:

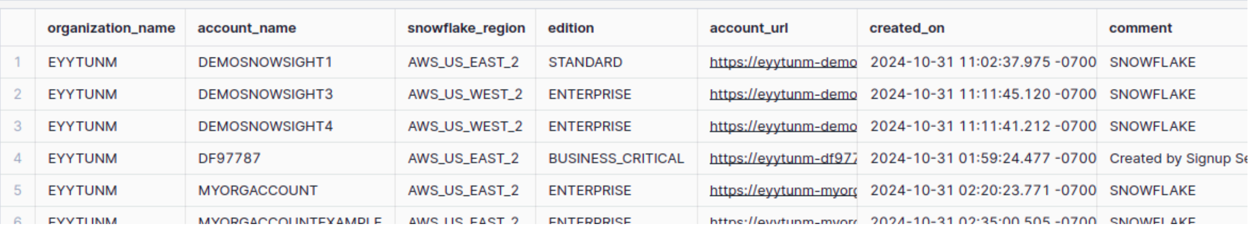


```
--View all accounts
SHOW ACCOUNTS;

--Lab 2.2
USE ROLE orgadmin;

ALTER ACCOUNT DEMOSNOWSIGHT2 RENAME TO DEMOSNOWSIGHT4;

--View all accounts
SHOW ACCOUNTS;
```



	organization_name	account_name	snowflake_region	edition	account_url	created_on	comment
1	EYYTUNM	DEMOSNOWSIGHT1	AWS_US_EAST_2	STANDARD	https://eyytunm-demo	2024-10-31 11:02:37.975 -0700	SNOWFLAKE
2	EYYTUNM	DEMOSNOWSIGHT2	AWS_US_WEST_2	ENTERPRISE	https://eyytunm-demo	2024-10-31 11:11:41.212 -0700	SNOWFLAKE
3	EYYTUNM	DEMOSNOWSIGHT3	AWS_US_WEST_2	ENTERPRISE	https://eyytunm-demo	2024-10-31 11:11:45.120 -0700	SNOWFLAKE
4	EYYTUNM	DF97787	AWS_US_EAST_2	BUSINESS_CRITICAL	https://eyytunm-df977	2024-10-31 01:59:24.477 -0700	Created by Signup Se
5	EYYTUNM	MYORGACCOUNT	AWS_US_EAST_2	ENTERPRISE	https://eyytunm-myorg	2024-10-31 02:20:23.771 -0700	SNOWFLAKE
6	EYYTUNM	MYORGACCOUNTEXAMPLE	AWS_US_EAST_2	ENTERPRISE	https://eyytunm-myorg	2024-10-31 02:35:00.505 -0700	SNOWFLAKE

After the command:



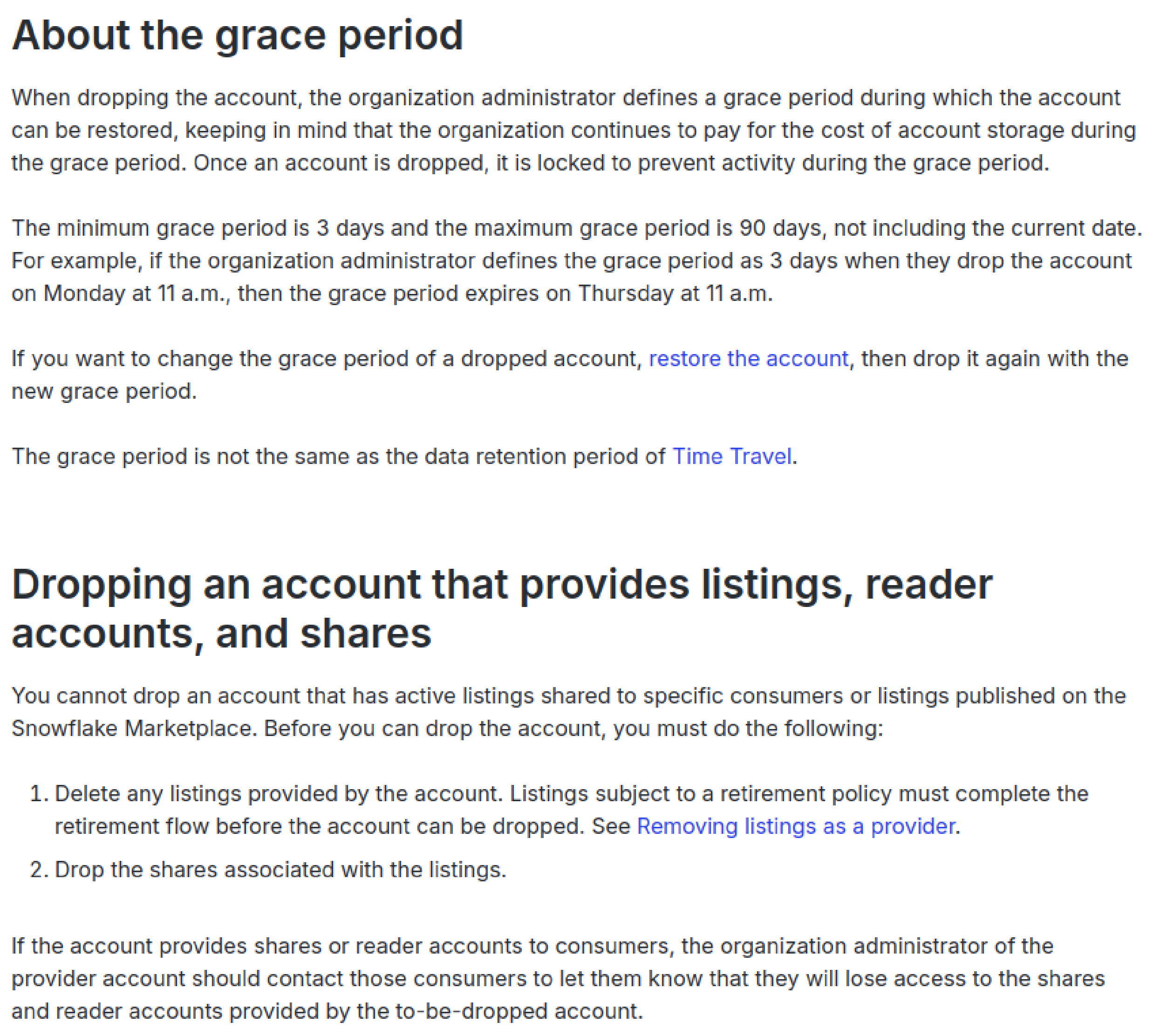
	organization_name	account_name	snowflake_region	edition	account_url	created_on	comment
1	EYYTUNM	DEMOSNOWSIGHT1	AWS_US_EAST_2	STANDARD	https://eyytunm-demo	2024-10-31 11:02:37.975 -0700	SNOWFLAKE
2	EYYTUNM	DEMOSNOWSIGHT3	AWS_US_WEST_2	ENTERPRISE	https://eyytunm-demo	2024-10-31 11:11:45.120 -0700	SNOWFLAKE
3	EYYTUNM	DEMOSNOWSIGHT4	AWS_US_WEST_2	ENTERPRISE	https://eyytunm-demo	2024-10-31 11:11:41.212 -0700	SNOWFLAKE
4	EYYTUNM	DF97787	AWS_US_EAST_2	BUSINESS_CRITICAL	https://eyytunm-df977	2024-10-31 01:59:24.477 -0700	Created by Signup Se
5	EYYTUNM	MYORGACCOUNT	AWS_US_EAST_2	ENTERPRISE	https://eyytunm-myorg	2024-10-31 02:20:23.771 -0700	SNOWFLAKE
6	EYYTUNM	MYORGACCOUNTEXAMPLE	AWS_US_EAST_2	ENTERPRISE	https://eyytunm-myorg	2024-10-31 02:35:00.505 -0700	SNOWFLAKE

Dropping an account

The organization administrator (i.e. a user with the ORGADMIN role) can drop an account to delete it from the system. A dropped account is not deleted immediately, but rather enters a grace period during which the administrator can restore (“undrop”) the account. When the grace period expires, Snowflake purges the dropped account from the system.

The organization administrator cannot drop an account while they are logged in to it; they must log in to a different ORGADMIN account before executing the DROP ACCOUNT command. This means that the organization administrator cannot drop the last account in the organization. If your organization consists of a single account that needs to be deleted, contact Snowflake Support.

> **Tip**
> Because Snowflake does not permanently delete an account when it is initially dropped, you cannot immediately create a new account with the same name as the one you just dropped. As a workaround, rename the account before dropping it.



About the grace period

When dropping the account, the organization administrator defines a grace period during which the account can be restored, keeping in mind that the organization continues to pay for the cost of account storage during the grace period. Once an account is dropped, it is locked to prevent activity during the grace period.

The minimum grace period is 3 days and the maximum grace period is 90 days, not including the current date. For example, if the organization administrator defines the grace period as 3 days when they drop the account on Monday at 11 a.m., then the grace period expires on Thursday at 11 a.m.

If you want to change the grace period of a dropped account, restore the account, then drop it again with the new grace period.

The grace period is not the same as the data retention period of Time Travel.

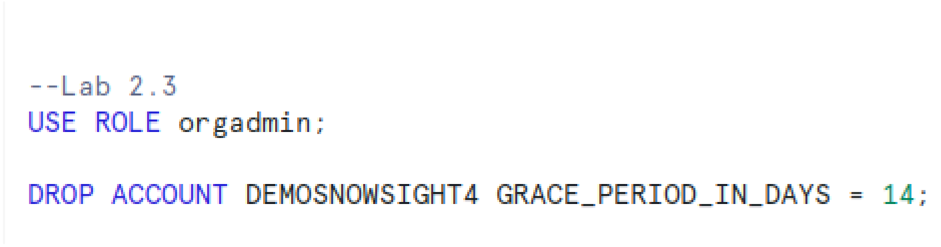
Dropping an account that provides listings, reader accounts, and shares

You cannot drop an account that has active listings shared to specific consumers or listings published on the Snowflake Marketplace. Before you can drop the account, you must do the following:

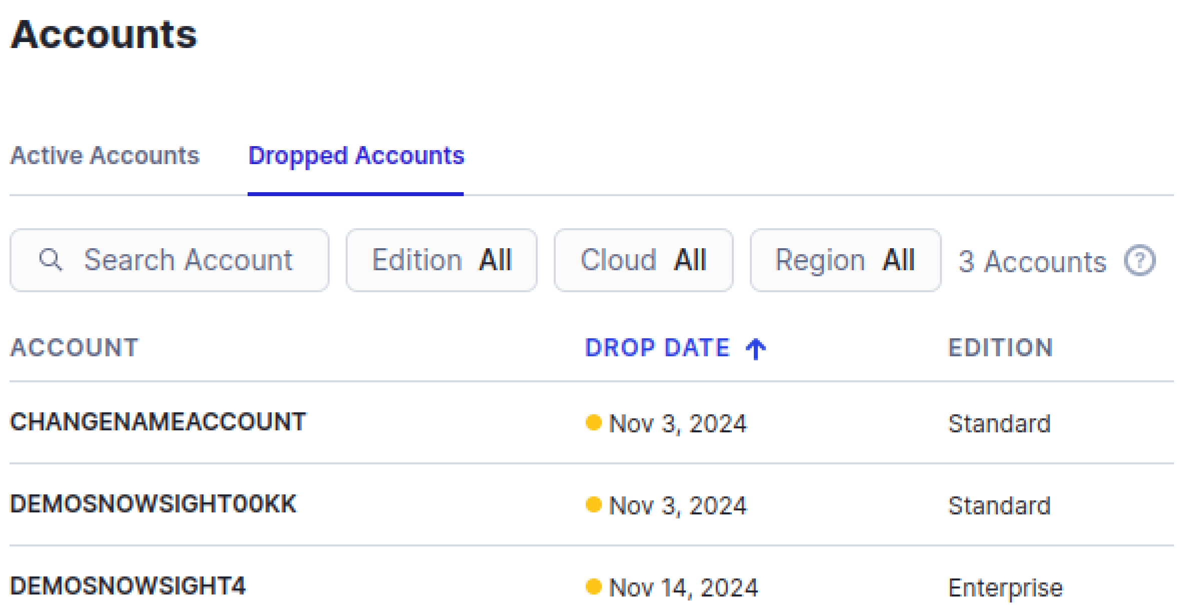
1. Delete any listings provided by the account. Listings subject to a retirement policy must complete the retirement flow before the account can be dropped. See Removing listings as a provider.
2. Drop the shares associated with the listings.

If the account provides shares or reader accounts to consumers, the organization administrator of the provider account should contact those consumers to let them know that they will lose access to the shares and reader accounts provided by the to-be-dropped account.

To drop an account, use the following command:



```
--Lab 2.3
USE ROLE orgadmin;

DROP ACCOUNT DEMOSNOWSIGHT4 GRACE_PERIOD_IN_DAYS = 14;
```



Accounts

Edition All | Cloud All | Region All | 3 Accounts

ACCOUNT	DROP DATE ↑	EDITION
CHANGENAMEACCOUNT	● Nov 3, 2024	Standard
DEMOSNOWSIGHT00KK	● Nov 3, 2024	Standard
DEMOSNOWSIGHT4	● Nov 14, 2024	Enterprise

Use Undrop to restore the account.

Organization accounts

PREVIEW FEATURE — OPEN

Available to all non-government accounts that are Enterprise Edition (or higher).

To inquire about upgrading, please contact Snowflake Support.

An *organization account* is a special type of account that organization administrators use to perform tasks that affect the entire organization. For example, administrators use the organization account to do the following:

- View organization-level data collected from all accounts in the organization, including the query history from each account.
- Enable Snowflake Marketplace terms for the entire organization.
- Manage the lifecycle of accounts in an organization, including creating and deleting accounts.
- Enable replication for an account.

Before this preview, administrators needed to perform these organization-level tasks using an account that had the ORGADMIN role enabled. These *ORGADMIN-enabled accounts* are different from the organization account. Unlike the classic approach where an organization might have multiple ORGADMIN-enabled accounts, there is only one organization account.

During the preview of organization accounts, organization administrators can still use an ORGADMIN-enabled account to manage the lifecycle of accounts (for example, creating and deleting accounts). After organization accounts become generally available, there will be a transition period, after which administrators will use the organization account for all organization-level tasks.

Create the organization account

> **Note**
> Creating the organization account results in the ORGANIZATION_USAGE schema being populated with data, which incurs additional costs for your organization.

To create the organization account:

1. Choose an existing account from which you will create the organization account. This existing account must have the ORGADMIN role enabled.
2. Sign in to the account you are using to create the organization account.
3. Switch to the ORGADMIN role. For example:

   ```
   USE ROLE ORGADMIN;
   ```

4. Execute the CREATE ORGANIZATION ACCOUNT command. For example:

   ```
   CREATE ORGANIZATION ACCOUNT myorgaccount
       ADMIN_NAME = admin
       ADMIN_PASSWORD = 'TestPassword1'
       EMAIL = 'myemail@myorg.org'
       MUST_CHANGE_PASSWORD = true
       EDITION = enterprise;
   ```

Password policies

A password policy specifies the requirements that must be met to create and reset a password to authenticate to Snowflake.

Snowflake provides two options for password policies:

- A built-in password policy to facilitate the initial user provisioning process.
- A schema-level password policy object that can be set at the level of the Snowflake account, an individual user, or both depending on the use cases and needs of the user administrator.

For details on best practices and each of the password policy options, see:

- Best practices for password policies and passwords
- Snowflake-provided password policy
- Custom password policy for the account and users

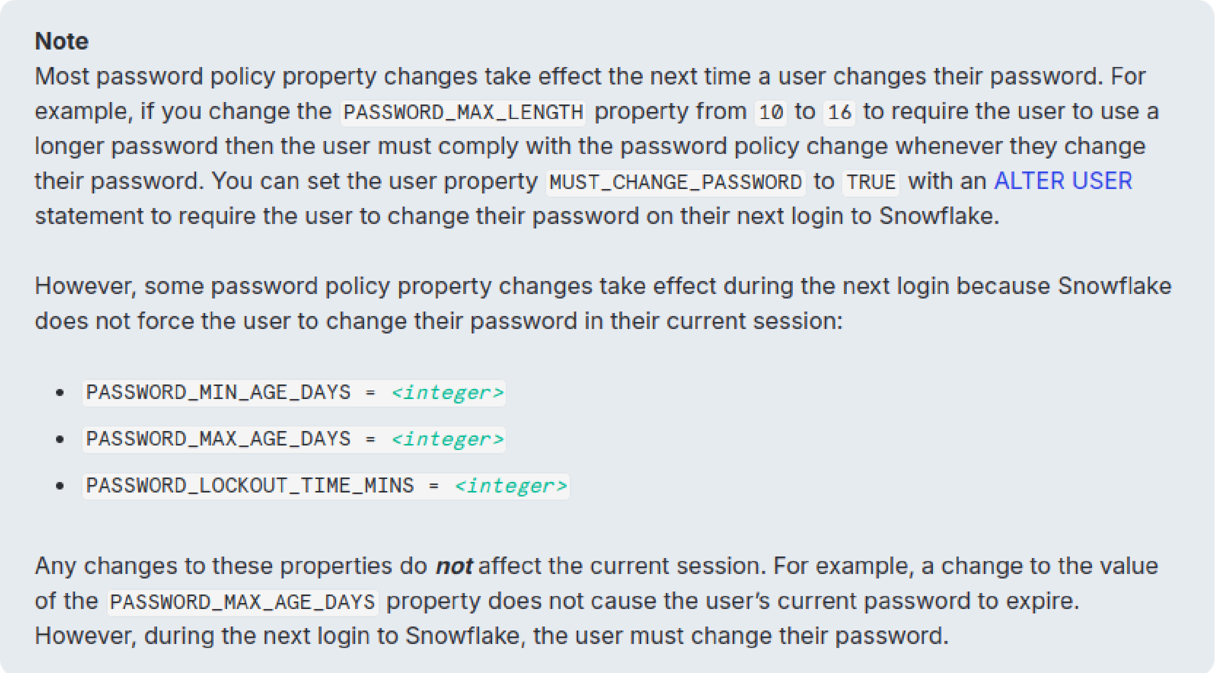
Custom password policy for the account and users

The custom password policy is a schema-level object that specifies the requirements that must be met to create and reset a password to authenticate to Snowflake, including the number of attempts to enter the password successfully and the number of minutes before a password can be retried (i.e. the "lockout" time).

The password policy requirements for a password include upper or lowercase letters, special characters, numbers, and password length to meet security requirements for users and clients to authenticate to Snowflake. Password policies that require strong passwords help to meet security guidelines and regulations.

Snowflake supports setting a password policy for your Snowflake account and for individual users. Only one password policy can be set at any given time for your Snowflake account or a user. If a password policy exists for the Snowflake account and another password policy is set for a user in the same Snowflake account, the user-level password policy takes precedence over the account-level password policy.

The password policy applies to new passwords that are set in your Snowflake account. To ensure that users with existing passwords meet the password policy requirements, require users to change their password during their next login to Snowflake as shown in Step 6: Require a password change (in this topic).



Note

Most password policy property changes take effect the next time a user changes their password. For example, if you change the `PASSWORD_MAX_LENGTH` property from `10` to `16` to require the user to use a longer password then the user must comply with the password policy change whenever they change their password. You can set the user property `MUST_CHANGE_PASSWORD` to `TRUE` with an ALTER USER statement to require the user to change their password on their next login to Snowflake.

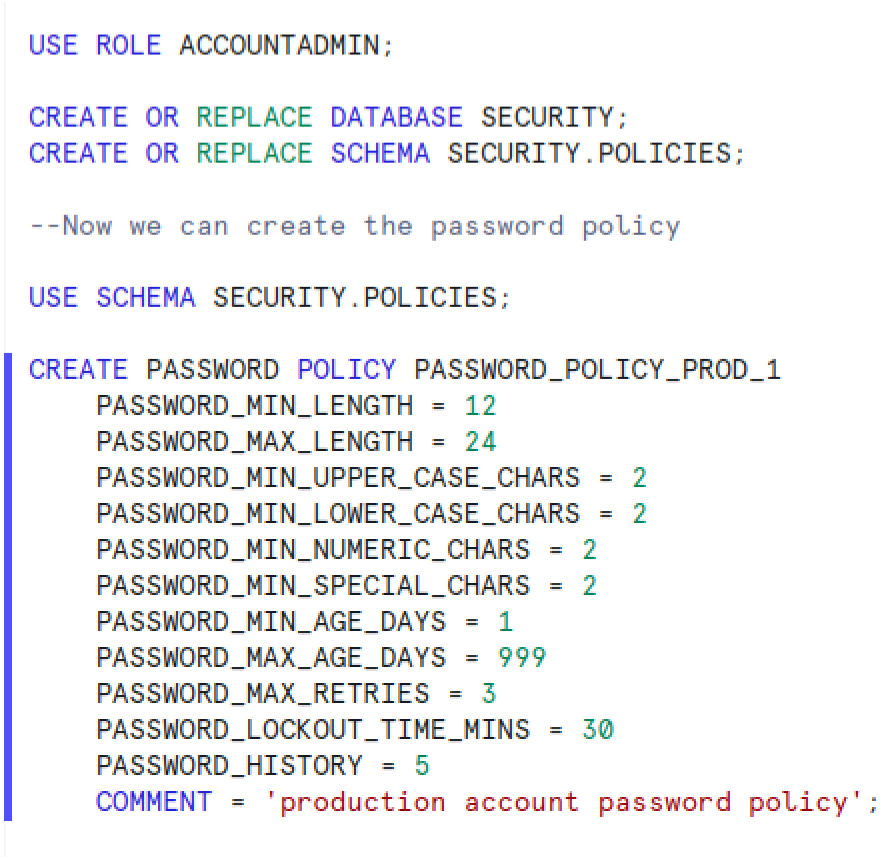
However, some password policy property changes take effect during the next login because Snowflake does not force the user to change their password in their current session:

- `PASSWORD_MIN_AGE_DAYS = <integer>`
- `PASSWORD_MAX_AGE_DAYS = <integer>`
- `PASSWORD_LOCKOUT_TIME_MINS = <integer>`

Any changes to these properties do ***not*** affect the current session. For example, a change to the value of the `PASSWORD_MAX_AGE_DAYS` property does not cause the user's current password to expire. However, during the next login to Snowflake, the user must change their password.

To create a password policy, you will need to do the following:

1. Create a database and schema to manage the policies.
2. Create the password policy
3. Apply to accounts or users



```
USE ROLE ACCOUNTADMIN;

CREATE OR REPLACE DATABASE SECURITY;
CREATE OR REPLACE SCHEMA SECURITY.POLICIES;

--Now we can create the password policy

USE SCHEMA SECURITY.POLICIES;

CREATE PASSWORD POLICY PASSWORD_POLICY_PROD_1
    PASSWORD_MIN_LENGTH = 12
    PASSWORD_MAX_LENGTH = 24
    PASSWORD_MIN_UPPER_CASE_CHARS = 2
    PASSWORD_MIN_LOWER_CASE_CHARS = 2
    PASSWORD_MIN_NUMERIC_CHARS = 2
    PASSWORD_MIN_SPECIAL_CHARS = 2
    PASSWORD_MIN_AGE_DAYS = 1
    PASSWORD_MAX_AGE_DAYS = 999
    PASSWORD_MAX_RETRIES = 3
    PASSWORD_LOCKOUT_TIME_MINS = 30
    PASSWORD_HISTORY = 5
    COMMENT = 'production account password policy';
```

```
--Apply the policy to an account

ALTER ACCOUNT SET PASSWORD POLICY security.policies.password_policy_prod_1;

--Apply the policy to a user

CREATE USER test_user1;

ALTER USER test_user1 SET PASSWORD POLICY security.policies.password_policy_user;

--To reset a password policy, use UNSET

ALTER ACCOUNT UNSET PASSWORD POLICY;
```