**Accounts and Assurances in Snowflake**

Managing accounts in Snowflake is all about controlling access to your data and keeping it secure. This involves setting up user roles and permissions while using key security features to protect your information. Snowflake ensures that only the right users have access to the right data, while providing the tools you need to maintain privacy and meet important security standards.

❓ **TOPIC GUIDANCE QUESTIONS:**

1. How does role-based access control (RBAC) help secure user access in Snowflake?
2. What is the importance of multi-factor authentication (MFA) in ensuring account security?
3. How does Snowflake ensure compliance with global standards like GDPR and HIPAA?
4. What features does Snowflake provide to audit and monitor activities within accounts?

**Accounts and Assurances in Snowflake**

Account Management - Snowflake accounts are tied to a specific region and cloud provider (AWS, Azure, or Google Cloud). Each account is uniquely identified and managed to ensure proper access and configuration across cloud platforms.

Account Identifiers - Every Snowflake account has a unique identifier that includes the account’s cloud provider and region. Knowing how to manage these identifiers ensures smooth account operation across global teams.

Access control framework - Snowflake’s approach to access control combines aspects from both Discretionary Access Control (DAC) and Role-based Access Control (RBAC).

Access control framework:

**Discretionary Access Control (DAC):** Each object has an owner, who can in turn grant access to that object.

**Role-based Access Control (RBAC):** Access privileges are assigned to roles, which are in turn assigned to users.

The key concepts to understanding access control in Snowflake are:

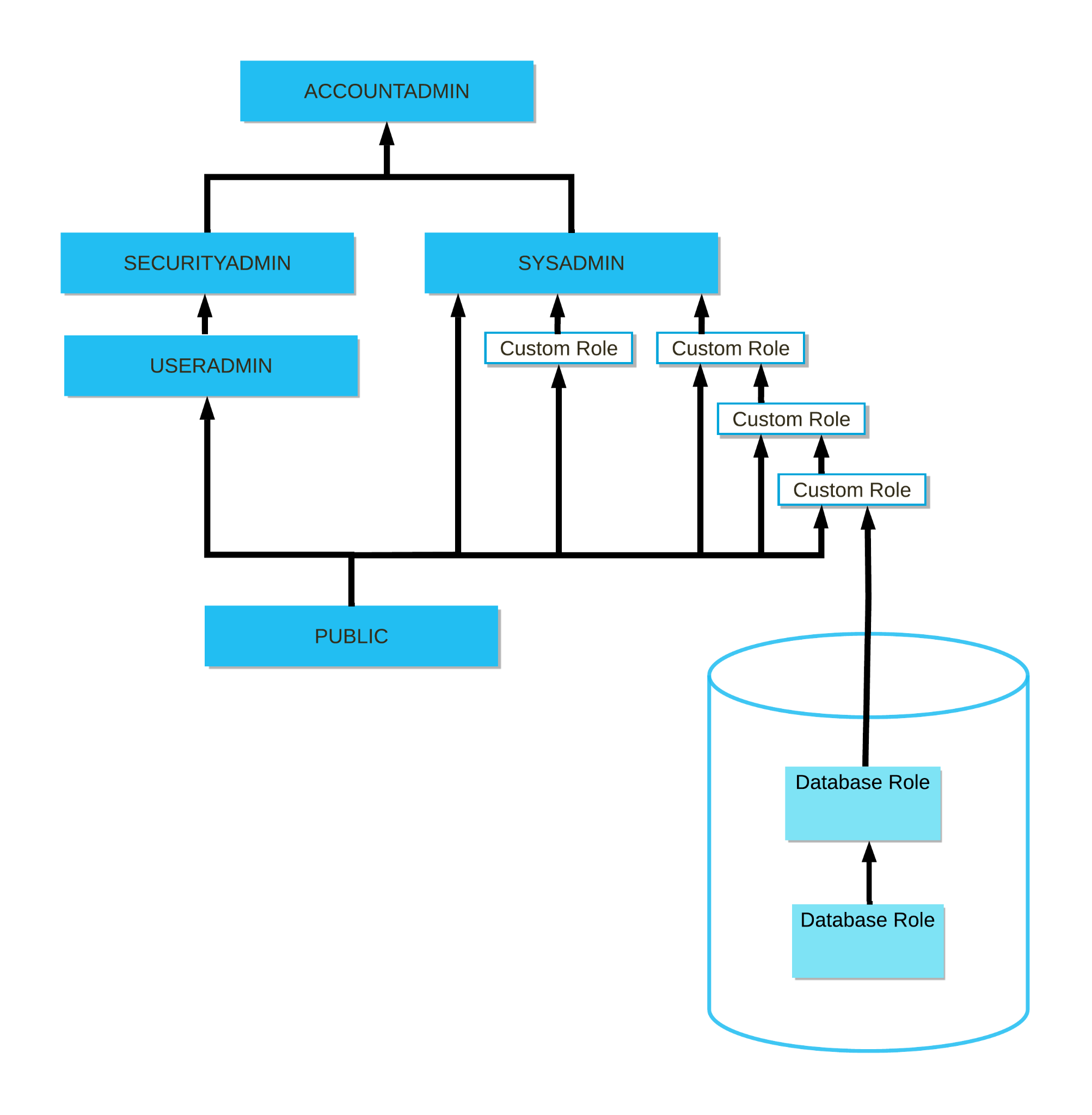
**Securable object:** An entity to which access can be granted. Unless allowed by a grant, access is denied.

**Role:** An entity to which privileges can be granted. Roles are, in turn, assigned to users. Note that roles can also be assigned to other roles, creating a role hierarchy.

**Privilege:** A defined level of access to an object. Multiple distinct privileges may be used to control the granularity of access granted.

**User:** A user identity recognized by Snowflake, whether associated with a person or program.

In the Snowflake model, access to securable objects is allowed via privileges assigned to roles, which are, in turn, assigned to users or other roles. Granting a role to another role creates a role hierarchy, which is explained in the Role hierarchy and privilege inheritance section(opens in a new tab)



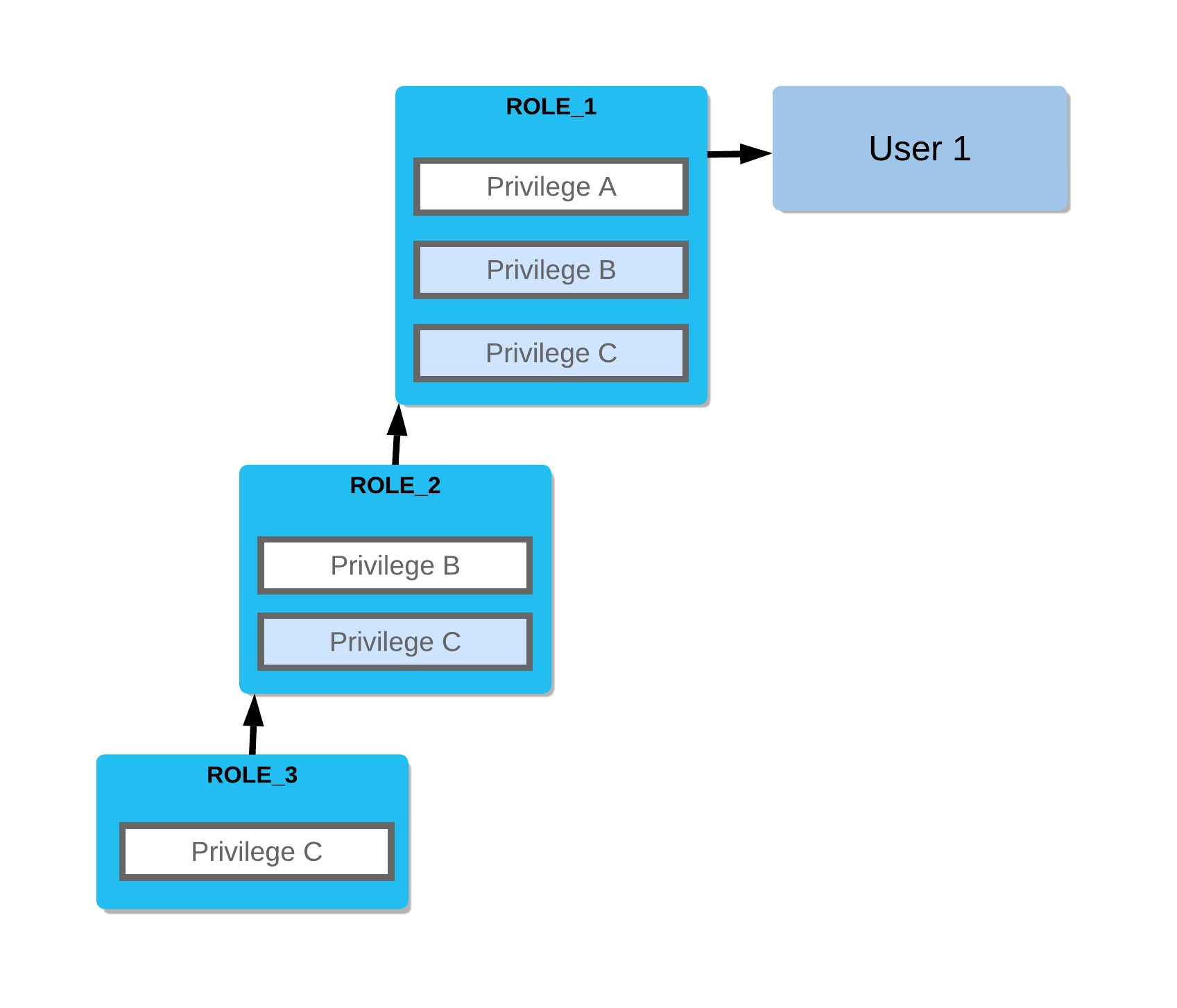
For a more specific example of role hierarchy and privilege inheritance, consider the following scenario:

Role 3 has been granted to Role 2.

Role 2 has been granted to Role 1.

Role 1 has been granted to User 1.

Try to create a diagram to understand this scenario.



**Creating custom read-only roles**

Suppose you need a role that is limited to querying all tables in a specific schema (for example, d1.s1). Users who execute commands using this role cannot update the table data, create additional database objects, or drop tables. The role is limited to querying table data.

GRANT USAGE

ON DATABASE d1

TO ROLE read\_only;

GRANT USAGE

ON SCHEMA d1.s1

TO ROLE read\_only;

GRANT SELECT

ON ALL TABLES IN SCHEMA d1.s1

TO ROLE read\_only;

GRANT USAGE

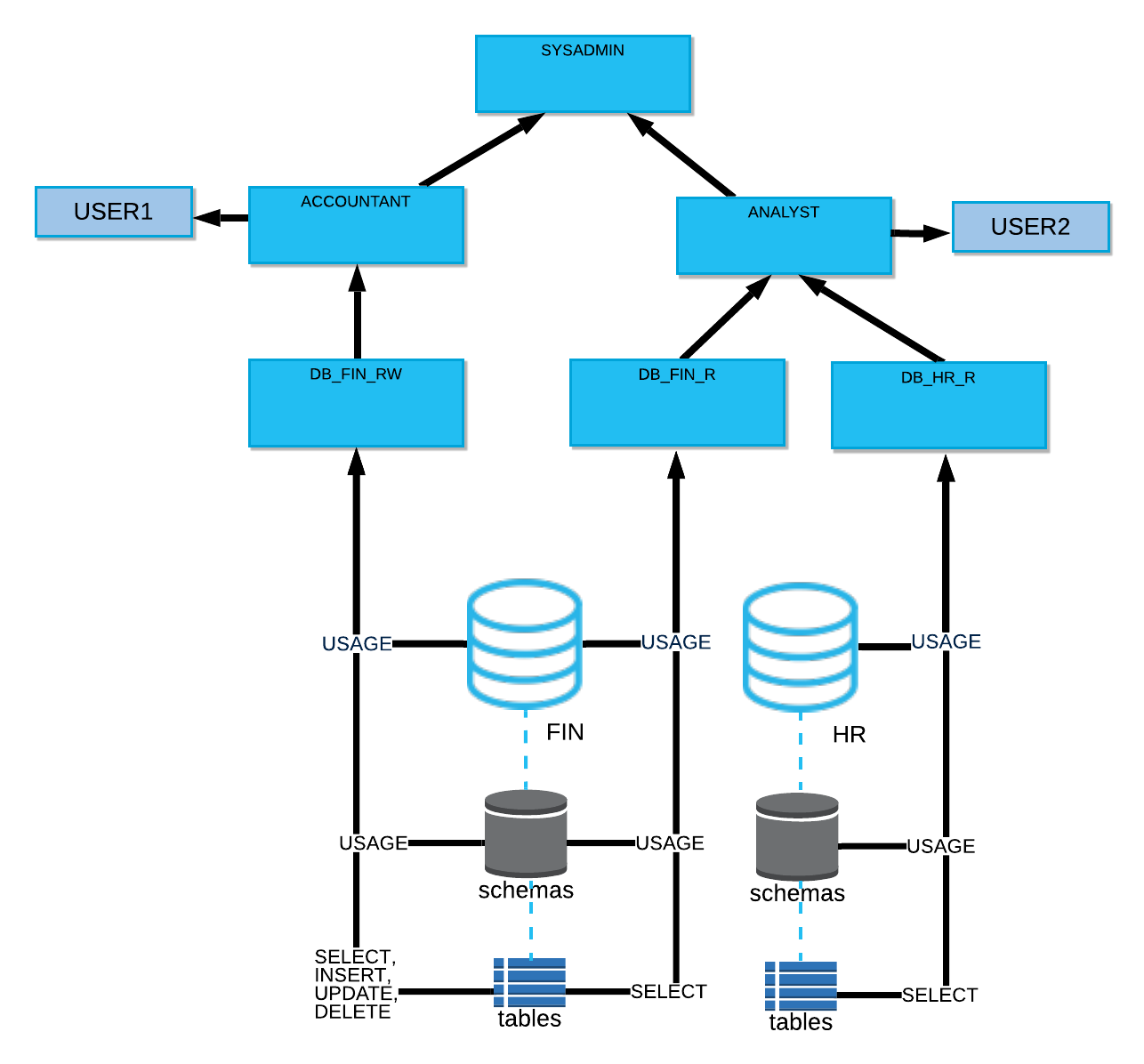
ON WAREHOUSE w1

TO ROLE read\_only;

**Example**

As a simple example, suppose two databases in an account, fin and hr, contain payroll and employee data, respectively. Accountants and analysts in your organization require different permissions on the objects in these databases to perform their business functions. Accountants should have read-write access to fin but might only require read-only access to hr because human resources personnel maintain the data in this database. Analysts could require read-only access to both databases.

|  |  |  |
| --- | --- | --- |
| **Custom Role** | **Description** | **Privileges** |
| **db\_hr\_r** | Access role that permits read-only access to tables in the hr database. | USAGE on database hr.  USAGE on all schemas in database hr.  SELECT on all tables in database hr. |
| **db\_fin\_r** | Access role that permits read-only access to tables in the fin database. | USAGE on database fin.  USAGE on all schemas in database fin.  SELECT on all tables in database fin. |
| **db\_fin\_rw** | Access role that permits read-write access to tables in the fin database. | USAGE on database fin.  USAGE on all schemas in database fin.  SELECT, INSERT, UPDATE, DELETE on all tables in database fin. |



**Creating a role hierarchy**

When creating custom roles, consider creating a role hierarchy ultimately assigned to a high-level administrator role. In general, the SYSADMIN role works well as the role all other roles are assigned to in a hierarchy, although it is important to note that any role with sufficient privileges could serve this function. The SYSADMIN role is a system-defined role that has privileges to create warehouses, databases, and database objects in an account and grant those privileges to other roles. In the default system hierarchy, the top-level ACCOUNTADMIN role manages the system administrator role.

Create a role hierarchy by granting a role to a second role. You can then grant that second role to a third role. The privileges associated with a role are inherited by any roles above that role in the hierarchy (that is, the parent role).

Grant a role to another role

Assign the role to a higher-level role in a role hierarchy. In this example, we are assigning the r1 role created in Creating custom roles to the SYSADMIN role. The SYSADMIN role inherits any object privileges granted to the r1 role:

GRANT ROLE r1

TO ROLE sysadmin;

Sign in to Snowsight.

In the navigation menu, select Admin » Users & Roles, and then select Roles.

Select Table and locate the role that you want to grant to another role. For example, r1.

In the section 0 roles have been granted R12, select Grant to Role.

For Role to receive grant, select SYSADMIN.

Select Grant.

**Granting privileges to a user**

A user with MANAGE GRANTS privileges on objects can grant privileges directly to users. For more information, see GRANT <privileges> … TO USER.

For example, to grant the USAGE privilege on a Streamlit application streamlitApp1 to user1 , execute the following commands:

GRANT USAGE ON WAREHOUSE w1 TO USER user1;

GRANT USAGE ON DATABASE d1 TO USER user1;

GRANT USAGE ON SCHEMA d1.s1 TO USER user1;

GRANT USAGE ON STREAMLIT `streamlitApp1` TO USER user1;

**Assigning future grants on objects**

To simplify grant management, future grants allow defining an initial set of privileges to grant on new (that is, future) objects of a certain type in a database or a schema. As new objects are created in the database or schema, the defined privileges are automatically granted to a specified role.

Considerations when using future grants

When future grants are defined on the same object type for a database and a schema in the same database, the schema-level grants take precedence over the database level grants, and the database level grants are ignored. This behavior applies to privileges on future objects granted to one role or different roles.

For example, the following statements grant different privileges on objects of the same type at the database and schema levels.

Grant the SELECT privilege on all future tables in database d1 to role r1:

GRANT SELECT ON FUTURE TABLES IN DATABASE d1 TO ROLE r1;

Grant the INSERT and DELETE privileges on all future tables in schema d1.s1 to role r2.

GRANT INSERT,DELETE ON FUTURE TABLES IN SCHEMA d1.s1 TO ROLE r2;

The future grants assigned to the r1 role on object types in schema d1.s1 are ignored completely. When new tables are created in schema d1.s1, only the future privileges defined on tables for the r2 role are granted.

Database level future grants apply to both regular and managed access schemas.

Creating a managed schema in Snowflake involves a few specific steps. When you create a managed schema, Snowflake automatically controls the permission management within that schema. Only the schema owner or a role with the MANAGE GRANTS privilege can manage access to the objects within the schema.

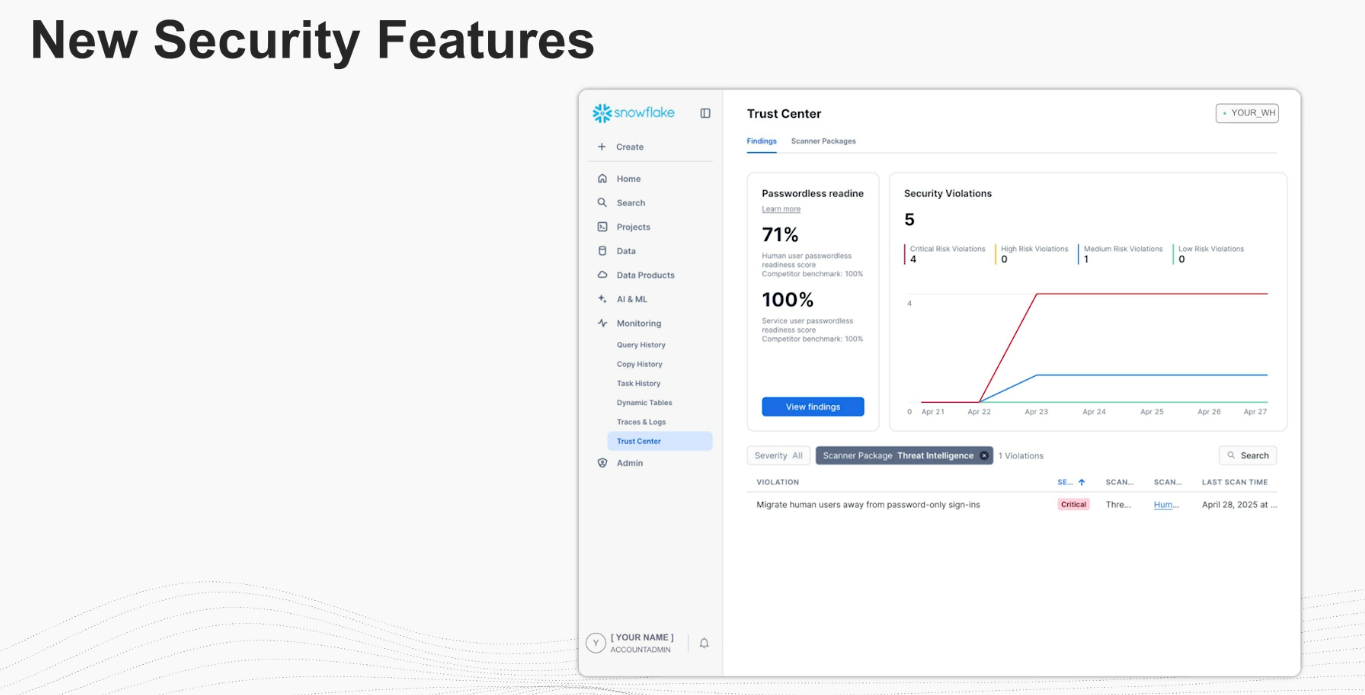
**MFA**

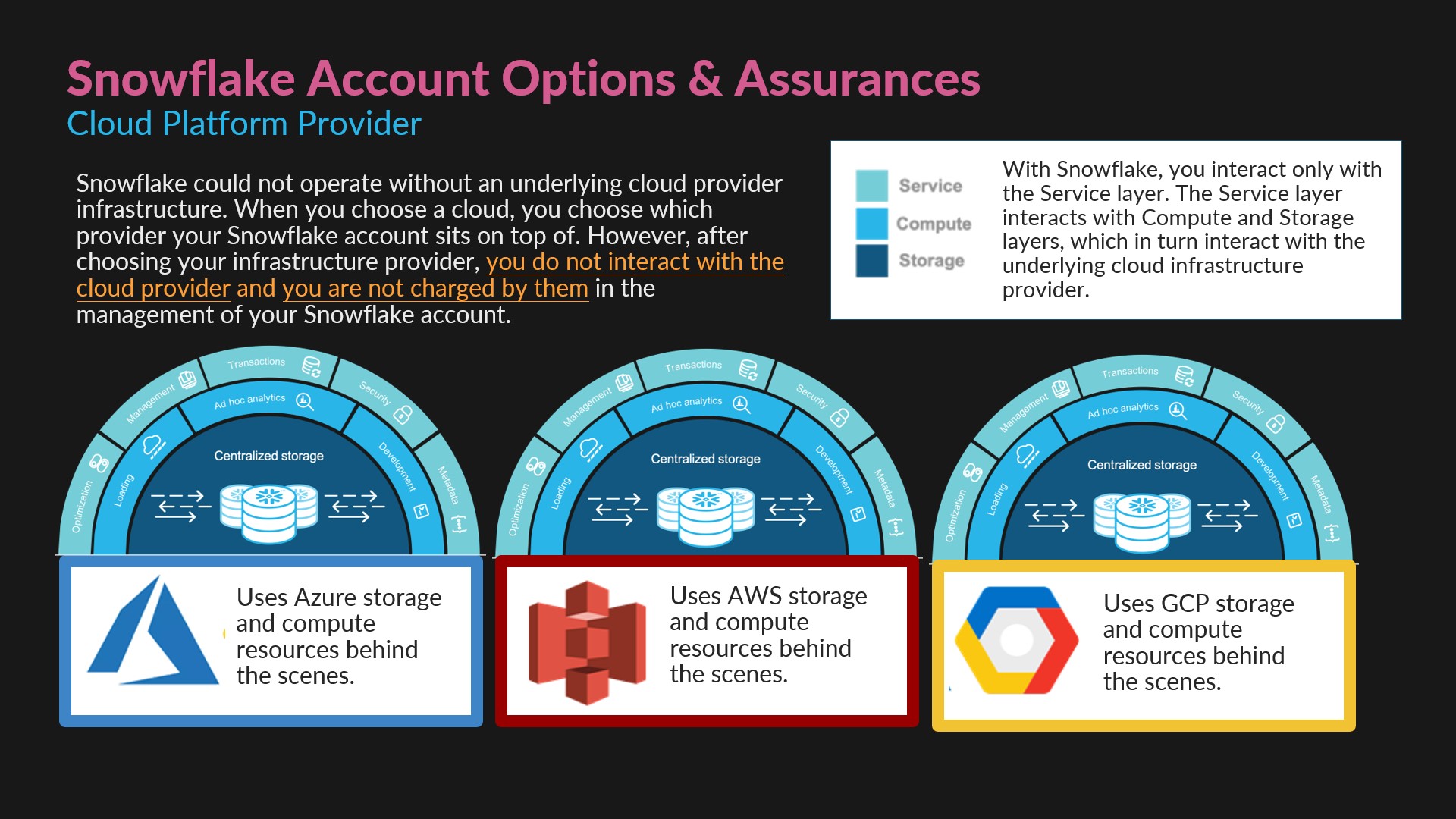
* Describe the planned adoption of new MFA Authentication requirements
* Use Trust Center Capabilities and Threat Intelligence Scanner Package
* Anticipate and plan for the upcoming deprecation of single-factor password sign-ins
* Enforce MFA at the User and Account Levels
* Use common MFA tools to authenticate
* Identify and troubleshoot common issues
* Get help and stay on top of evolving best practices

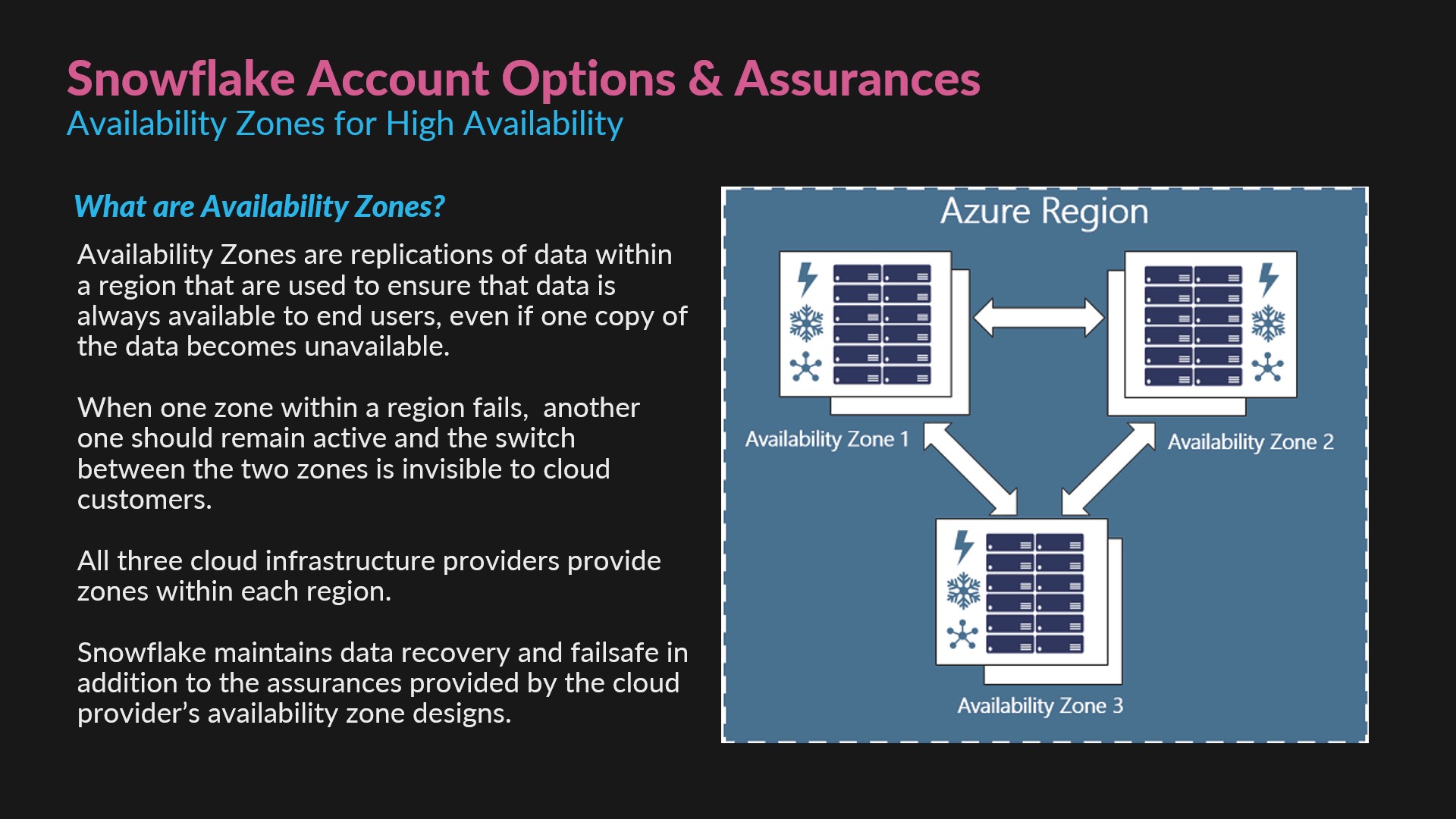
**Snowflake's Security Toolkit: Trust Center & Enforcement**

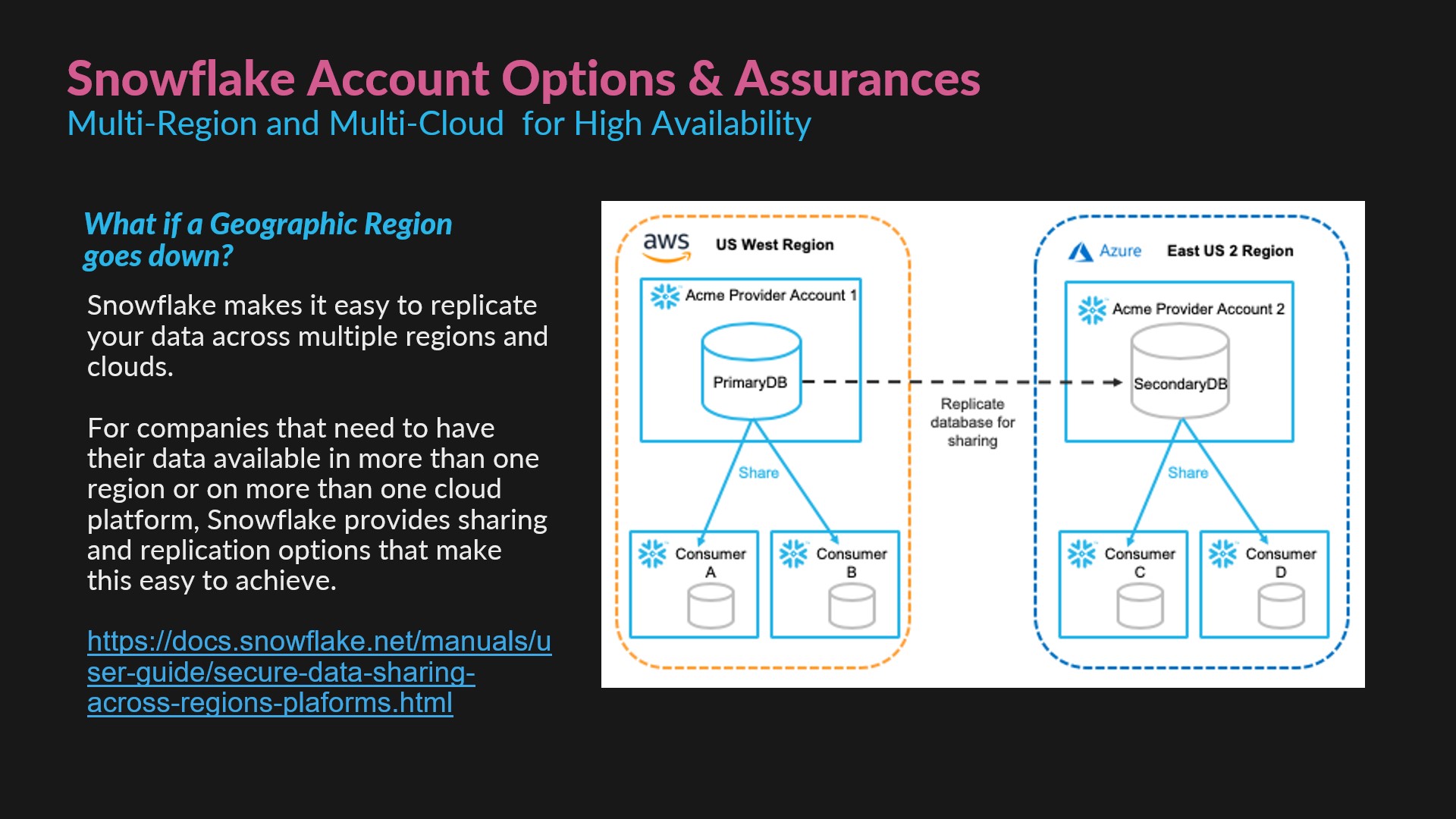
Snowflake provides capabilities like the Trust Center and Threat Intelligence Scanner Package to help you monitor your security posture and identify areas needing attention, such as users who could benefit from stronger authentication.

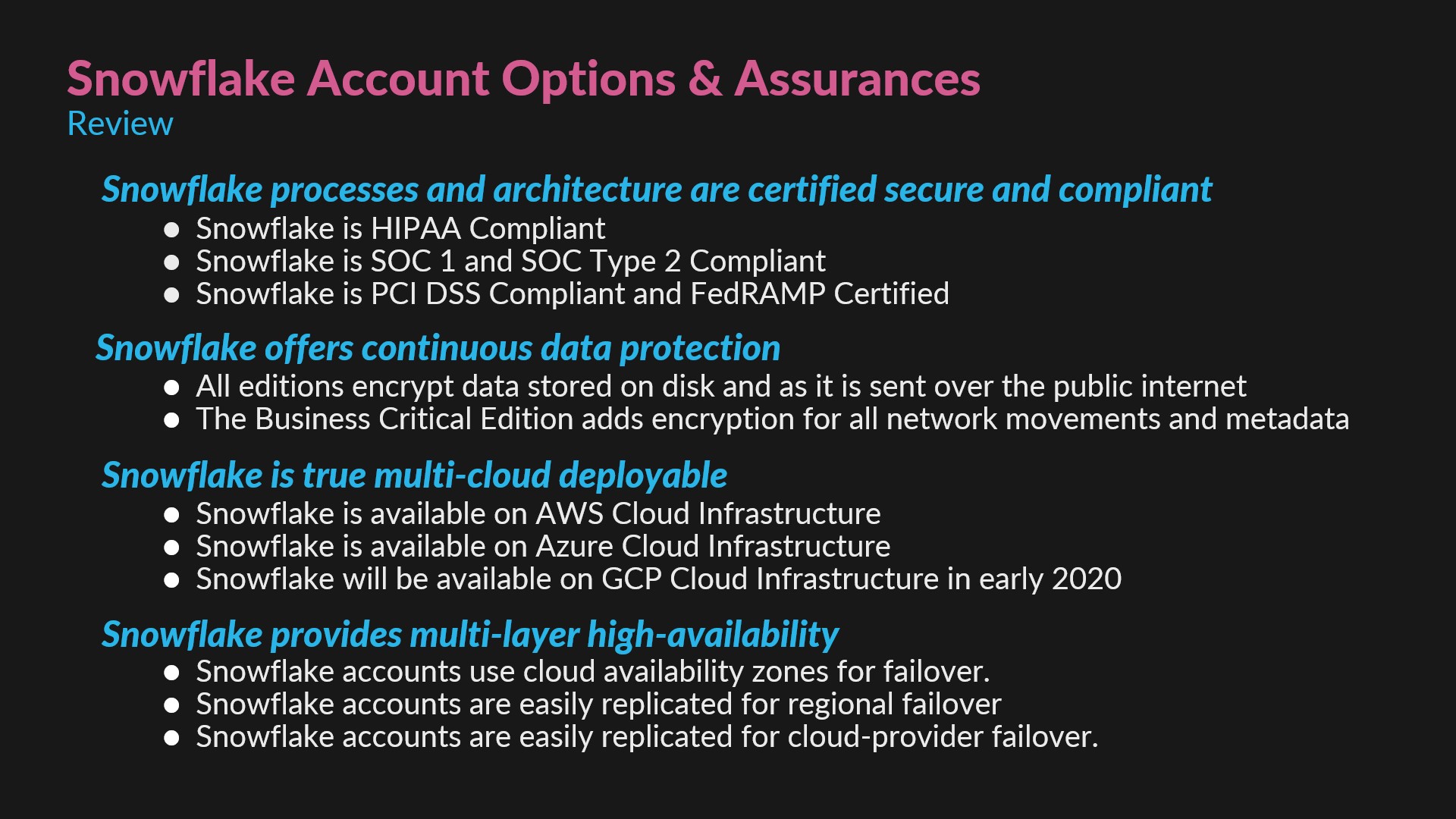
**Trust Center** is the interface that helps you manage risk posture natively for Snowflake accounts. It discovers security risks across cloud environments based on industry best practices and provides recommendations to resolve them. Below, we'll dive into the Trust Center, related security scanners, and how you can actively enforce MFA policies.











**USE CASE:** If your organization name is ABCDZE, and there are two accounts named TEST, one in the AWS us-east-2 region and the other in the Azure west-us-2 region, the new URLs will use the following structure:

First account: https://abcdze-test\_aws\_us\_east\_2.snowflakecomputing.com

Second account: https://abcdze-test\_azure\_west\_us\_2.snowflakecomputing.com

❓Guiding Questions:

1. What are the key elements to consider when creating or managing a Snowflake account?
2. How does role-based access control (RBAC) help in securing Snowflake accounts?
3. What is the importance of account identifiers, and how are they structured?
4. Why is it important to create specific roles in Snowflake?
5. How does giving the least amount of access necessary improve security?
6. What’s the best way to structure roles to match your organization?

**Why does role management matter?**

Every organization has different people doing different things—someone in marketing shouldn’t have access to your financial records, right? Snowflake helps you control this by assigning roles to users, making sure everyone has just the right level of access.

Creating Roles: When you create roles, you can match them to your organizational structure. For example, the finance team gets a role with a set of permissions, while the sales team gets another role with another set of permissions.

Assigning Users to Roles: After creating roles and granting privileges to them, you assign those roles to users. For instance, a financial analyst might need access to reports, but not the ability to change them. This prevents mistakes and keeps your data secure.

Least Privilege Principle: This simply means giving people the smallest amount of access necessary for their role. It reduces the risk of unauthorized access and accidental data breaches.

**Why do security assurances matter?**

With more and more companies relying on cloud storage, ensuring that data stays protected is vital. Snowflake provides security features that make it hard for unauthorized people to access your data—features like Multi-Factor Authentication (MFA) and encryption.

Multi-Factor Authentication (MFA): MFA adds an extra layer of security by requiring two forms of identification. It’s like adding a deadbolt to your front door—just a password isn’t enough!

Encryption: Snowflake ensures your data is encrypted both when it’s stored and when it’s moving between systems. Even if someone intercepts it, they won’t be able to read it without the decryption key.

Account Monitoring and Auditing: Snowflake keeps track of everything that happens in your account. That means you can see who accessed what data and when—perfect for spotting any suspicious behavior

**Real-World Example:**

An online retailer processing thousands of customer orders daily uses MFA and encryption to ensure that only authorized employees can access sensitive financial data. Regular audits help the company quickly spot any unusual activity, making sure the data stays secure.

1. How does MFA strengthen security for Snowflake accounts?
2. Why is encryption important when it comes to data storage?
3. What can account monitoring and auditing do to prevent security breaches?

**Why is compliance important?**

Following data privacy laws is non-negotiable. Failing to comply with regulations like GDPR or HIPAA can result in hefty fines and a loss of trust. Snowflake offers features that help your organization stay on the right side of the law.

Global Compliance Standards: Snowflake is designed to meet the requirements of privacy laws like GDPR (in Europe) and HIPAA (in healthcare). These standards help keep your customers’ data safe and your organization compliant.

Data Masking: This feature ensures that only authorized people can see sensitive information like Social Security Numbers or payment details. If someone without the right permissions tries to view the data, it appears masked.

Audit Trails: Snowflake keeps detailed logs of who accessed or modified data. This helps prove compliance and gives your organization a paper trail to follow in case of any legal inquiries.(Advanced topic)

<https://docs.snowflake.com/en/user-guide/access-history>

**Real-World Example**

A healthcare provider stores patient data in Snowflake. By using data masking, only medical professionals can see the patients’ Social Security Numbers, while administrative staff only see masked information. The provider also keeps audit logs to prove compliance with HIPAA regulations.

1. How does Snowflake help you follow global data privacy laws?
2. Why is data masking important, and how does it protect sensitive information?
3. How can audit trails help your organization stay compliant?

[**Managing Accounts in Your Organization**](https://docs.snowflake.com/en/user-guide/organizations-manage-accounts)

**QUIZ**

1. How does multi-factor authentication (MFA) enhance security in Snowflake?
2. It encrypts data stored in the cloud.
3. It requires users to provide two forms of identification, making unauthorized access more difficult.
4. It automatically grants access to users who forget their passwords.
5. It replaces the need for user passwords entirely.
6. Is Snowflake HIPAA compliant?
7. Yes
8. No
9. What are the names of the three Snowflake Editions offered when signing up for a trial account?
10. Standard
11. Free-Tier Basic
12. Enterprise
13. Ultra
14. Business Critical
15. True or False? Data is encrypted in every Snowflake account, regardless of which edition it is.
16. True
17. False
18. Which of the following industry compliance standards has Snowflake been audited and certified for? Select five.
19. Cloud GBDQ
20. SOC 1
21. SOC Type 2
22. PCI DSS
23. FedRAMP
24. HIPAA
25. When setting up a new Snowflake Account, what steps or choices must the enrollee complete? Select three.
26. Choose a Cloud Infrastructure Provider.
27. Choose a Cloud Compliance Standard.
28. Choose a Snowflake Edition.
29. Choose an Availability Zone.
30. Choose a Geographic Deployment Region.
31. Choose the billing and account management cloud provider.
32. When choosing a geographic deployment region, what factors might an enrollee consider? Select two.
33. End-user perceptions of glamorous or trendy geographic locations.
34. Proximity to the point of service.
35. Additional fees charged for regions with geo-political unrest.
36. Number of availability zones within a region.
37. Select three statements that are true.
38. A company can use more than one cloud infrastructure provider by setting up several Snowflake accounts.
39. A company can have its data stored in more than one geographical region by setting up several Snowflake accounts.
40. A company can use a combination of data sharing and replication to distribute data to various regions and cloud platforms.
41. A company can use availability zones to distribute data to various regions and cloud platforms.
42. What is a key benefit of Role-Based Access Control (RBAC) in Snowflake?
43. It automatically creates roles for every new user.
44. It limits access to only the data and features that a user needs, reducing security risks.
45. It allows users to bypass account identifiers.
46. It requires multi-factor authentication for all users.
47. What is the purpose of assigning roles to users in Snowflake?
48. To provide unrestricted access to all resources.
49. To grant specific permissions based on the user’s job role and responsibilities.
50. To restrict users from accessing their own data.
51. To automatically track user activities.

**ANSWER KEYS**

1. B
2. A
3. ACE
4. A
5. BCDEF
6. ACE
7. BD
8. ABC
9. B
10. B