THE DATA CLOUD IS A GLOBAL NETWORK

One global, unified system connecting companies and data providers to the most relevant data for their business.
ELEMENTS OF THE DATA CLOUD

PLATFORM + CONTENT

- Partner Data
- 3rd-Party Data
- SaaS Data
- Customer Data
- Applications
- Data Services
Built-in security & governance features protect the data you load and use in Snowflake

- Snowflake uses the industry-standard “shared responsibility” model
- Snowflake personnel do not have unauthorized access to customer data
- Snowflake personnel do not collect, delete, update, disclose, or use customer data

Snowflake uses sophisticated mechanisms to keep the platform safe and stable

- Orchestrated security built in to the fabric of the platform
  - Automated controls in place for all functions
  - Constant monitoring
  - Analysis to detect and mitigate threats quickly
- Customers never have direct access (e.g., "SSH") to the Snowflake VPC/VNET
- All access to customer data is through the Snowflake Service application layer
- Customer data is decrypted only in memory on dedicated Virtual Warehouse VMs:
  - Only the data required to process the command is decrypted
  - Virtual Warehouses are ephemeral, meaning they run only when needed

Customer data is encrypted at rest using dedicated encryption keys. Storage is governed by dedicated IaaS user principals.
# SNOWFLAKE SECURITY & GOVERNANCE AT A GLANCE

## Network Controls
- All communication secured using TLS 1.2 with HSTS enforced for all client communications, and controlled by Network Policies (IP Allowlisting)
- Integration with CSP Private Networking
  - GCP Private Service Connect
  - AWS Privatelink
  - AWS VPC ID S3 policies
  - Azure Private Link
  - Azure cross-VNet rules for Blob access
- Choose from any of the Snowflake-supported cloud regions

## Identity & Access
- SCIM user management
- Native Snowflake credentials
  - Password policies
  - Multi-Factor Authentication
  - Key Pair Authentication
- Federated Identity
  - SAML 2.0-based SSO
  - OAuth 2.0 delegated authorization
- Session control through policies

## Data Governance
- Built-in Features & partner integrations
- RBAC & DAC
- Column-level security
  - Using views & UDFs
  - Dynamic data masking
  - External tokenization
- Row access policy
- Tagging
- Classification
- Anonymization

## Data Protection
- Account, region, cloud, and data-level recovery & failover
  - Fail-Safe
  - Time Travel
  - Cross-cloud & region replication & failover
  - AWS, Azure, GCP redundancy

## Encryption
- Customer data always encrypted in flight
- Data-at-rest always encrypted using a hierarchical key model
  - Rooted in the CSP’s HSM
  - Automated key rotation & re-keying
  - BYOK with “Tri-Secret Secure”

## Compliance & Legal
- SOC 2 Type II 12 Month Coverage Period
- SOC 1 Type II 6 Month Coverage Period
- HITRUST Certified
- PCI Security Standards Council
- FedRAMP Moderate (Available from OMB MAX)

## Auditing
- Comprehensive audit trail for all activities by all users from login
INFOSEC & COMPLIANCE at a Glance
All reports, attestations, documentation, and certifications

Third-Party Reports & Certifications
- Snowflake SOC 2 Type II Report
- Snowflake SOC 1 Type II Report
- Snowflake PCI-DSS-AOC-Final Report
- HIPAA/HITRUST Reports (proving ability to enter into BAA)
- Snowflake’s ISO 27001 Certificate
- FedRAMP Moderate (on OMB MAX)
- IRAP Protected
- CyberGRX Report
- Penetration Test Results

Snowflake’s Policy Documentation
- Snowflake Security Policy
- https://www.snowflake.com/legal/ for Acceptable Use, Support, and more

Snowflake Internal Controls & Testing
- DRP, BCP, and Information System Contingency Plans
- Security Incident Process
- Staff Training, Onboarding, and Access Policies

Snowflake Self-Assessment Reports
- CAIQ
- SIG Lite
- Red Team Pen Tests
SNOWFLAKE SECURITY & GOVERNANCE AT A GLANCE

Achieve: Compliance benchmarks, Privacy goals
**Network Controls – Secure Communication**

**Common Connection Pattern for Drivers & Connectors**
- Every [driver & connector](#) connects the same way
- All communication encrypted end-to-end
  - All customer data flows solely over HTTPS
  - Connections encrypted using TLS 1.2 from client through to the Snowflake Service
  - HSTS enforced for all client communications
- Data encrypted at rest

**Common Access Control for all Sessions**
- IP allowlisting available to restrict client communication to specific IP addresses using [customer-configured Network Policies](#)
- Authentication required for all connections
Network policies can be applied at three levels

1. **Snowflake Account**
   - All traffic will use this policy, unless there is a more specific one.
   - Control is applied at authentication time.

2. **Outside Integration**
   - Applies to traffic at the integration endpoint only.
     For example: SCIM or OAuth security integrations.

3. **User Specific**
   - Applies to the specific user only.
   - Best practice for users used as service accounts.

*The most specific policy always wins.*
Authoritative Directory in Control

- User creation, changes, & deletes
- Roles driven by group membership
- Use Okta, Azure AD, or any system that speaks SCIM, or...
- Any system that can use SQL

**USER PROVISIONING WITH SCIM**
How to do Authentication & Delegated Authorization for Snowflake

Please note: Partners shown are a sample list and not the full list of supported platforms.

- Users may have multiple authN & authZ methods all configured at once
- “External Browser” mode can bring SSO to desktop apps that do not natively support it
- Integrate easily with a secrets management platform like Hashicorp Vault
SNOWFLAKE AUTHENTICATION

How to do Authentication & Delegated Authorization for Snowflake
SNOWFLAKE GOVERNANCE

Know Your Data
Understand, classify, and track data and its usage

Protect Your Data
Secure sensitive data with policy-based access controls

Unlock Your Data
Securely collaborate and share data across teams
<table>
<thead>
<tr>
<th>name</th>
<th>gender</th>
<th>age</th>
<th>zip_code</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>male</td>
<td>39</td>
<td>79007</td>
<td>123-555-1234</td>
</tr>
<tr>
<td>Jane Doe</td>
<td>female</td>
<td>50</td>
<td>77001</td>
<td>333-555-1236</td>
</tr>
<tr>
<td>Mary Taylor</td>
<td>trans-fem</td>
<td>46</td>
<td>77020</td>
<td>222-333-1111</td>
</tr>
<tr>
<td>Gene Marshall</td>
<td>non-binary</td>
<td>48</td>
<td>77042</td>
<td>555-555-1234</td>
</tr>
<tr>
<td>Michael Gaines</td>
<td>male</td>
<td>75</td>
<td>79003</td>
<td>666-666-1357</td>
</tr>
</tbody>
</table>

**We start with a table**

- The table has been instantiated from the encrypted, at-rest files (micro-partitions)
- The information in the table is opaque to Snowflake
- This table contains data “in the clear,” but you might load data that’s been modified in some way to protect it as well
**RBAC & DAC Protect the table**
- Every object in Snowflake is subject to these controls, and they are at the whole-object level
- RBAC inheritance and other RBAC features apply
- The customer controls RBAC completely
- DAC (Discretionary Access Control) applies to the role that owns the object, unless the object is subject to Managed Schema Access

**You may also create Views & UDFs**
- These are mostly used to redact or transform rows, columns, or even cells, and create a new object.
- The new object has RBAC and DAC controls.
We can use Policy controls for Columns and Rows

- Prevent View/UDF explosion
- Table/View owners and privileged users (such as ACCOUNTADMIN) unauthorized to data by default
- Ensure controls are applied in any context where the object’s data is used

We get more ease of management

- Centrally manage policies
- Apply a single policy to multiple tables
- Built-in separation of duty: policy admins assign and users are subject to policy controls
- All application and use is fully audited
We can leverage **Column-Level Security** to dynamically mask data at query time:

- No change to the stored data
- Mask or partially mask using constant value, hash, and custom functions
- Unmask for authorized users only

**Example:**

```sql
create or replace masking policy FOO
as (val string) returns string ->
    case
    when is_granted_to_invoker_role('SEECLEAR')
    then val
    when current_role('ONLYPART')
    then regexp_replace(val, '[0-9]', '*', 7)
    when is_role_in_session('CRYPTO')
    then decrypt_raw(val, KEY, IV, ...)
    when is_role_in_session('BESPOKE')
    then user_defined_func(val, baz, ...)
    else '** masked **'
    end;
```
Example using policy:
create or replace masking policy BAR as (val string)
returns string ->
case
  when is_granted_to_invoker_role('SEETOKENS')
    then val
  when current_role('GETREAL')
    then detok_ext_func(val, CURRENT_USER(), ...)
  else '** masked **'
end;

Example using SQL outside policy:
SELECT detok_ext_func(T1.phone) AS REAL_PHONE,
       T1.GENDER,
       T2.ZIP
FROM T1
JOIN T2
ON T2.PHONE = T1.PHONE
;

<table>
<thead>
<tr>
<th>name</th>
<th>gender</th>
<th>age</th>
<th>zip_code</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>male</td>
<td>39</td>
<td>79007</td>
<td>46ryn28ahr6</td>
</tr>
<tr>
<td>Jane Doe</td>
<td>female</td>
<td>50</td>
<td>77001</td>
<td>7dhe8ajs64te</td>
</tr>
<tr>
<td>Mary Taylor</td>
<td>trans-fem</td>
<td>46</td>
<td>77020</td>
<td>63gds74y2jwe</td>
</tr>
<tr>
<td>Gene Marshall</td>
<td>non-binary</td>
<td>48</td>
<td>77042</td>
<td>84yr75yw3456</td>
</tr>
<tr>
<td>Michael Gaines</td>
<td>male</td>
<td>75</td>
<td>79003</td>
<td>9fhr64gsrw21</td>
</tr>
</tbody>
</table>

Ingest protected (PII/PHI) data as Externally Tokenized
- Using tokenization provider functionality upstream from Snowflake
- De-tokenize for authorized users at query time
  - Tokenization provider called using a Snowflake External Function to de-tokenize data
  - For unauthorized users, third-party service is not called
  - Can be used in policy or outside

External Tokenization

RBAC & DAC

Views & UDFs

Column-Level Security
Filter rows at query time based on user role and lookup table

- Policy contains condition(s) to allow or filter out rows
- Policy is applied to one or more table, view, or external table in an account
- Dynamically generated predicate filters out rows the user is not authorized to see at query time
- Can be combined with other controls

Example:
```sql
create or replace row access policy FOO as (this_zip varchar) returns boolean -> 'allseeing_role' = current_role()
or exists (select 1 from zip_mapping_table where info_reader = current_role() and zip_code = this_zip);
```
**Tagging**

**RBAC & DAC**

**Views & UDFs**

**Column-Level Security**

**Row Access Policy**

**Tagging**

---

### Data Governance/Tagging

**Keep track of sensitive data for visibility and compliance**

- Assign tags to sensitive columns, tables, external tables, or views
- Easily audit sensitive objects without appropriate security policies
- Assign tags to virtual warehouses, Snowpipe, materialized views, and clustered tables to keep track of resource usage for cost visibility and attribution

**Manage tags with flexible administration models**

- Centralized tag creation and assignment for centrally managed governance
- Decentralized tag assignment controlled by privileges for object owner-supplied tag value
<table>
<thead>
<tr>
<th>name</th>
<th>gender</th>
<th>age</th>
<th>zip_code</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>male</td>
<td>39</td>
<td>79007</td>
<td>123-555-1234</td>
</tr>
<tr>
<td>Jane Doe</td>
<td>female</td>
<td>50</td>
<td>77001</td>
<td>333-555-1236</td>
</tr>
<tr>
<td>Mary Taylor</td>
<td>trans-fem</td>
<td>46</td>
<td>77020</td>
<td>222-333-1111</td>
</tr>
<tr>
<td>Gene Marshall</td>
<td>non-binary</td>
<td>48</td>
<td>77042</td>
<td>555-555-1234</td>
</tr>
<tr>
<td>Michael Gaines</td>
<td>male</td>
<td>75</td>
<td>79003</td>
<td>666-666-1357</td>
</tr>
</tbody>
</table>

**Classification**

- Classify columns containing personal data
  - Automatically detect columns with personal data
  - Apply Snowflake-defined semantic and privacy category system tags
  - Assign data access policies to columns based on system tags
  - Use system tags to audit personal data among millions of columns
  - Seamlessly integrates across Snowflake’s governance capabilities and Data Sharing
  - Populate and manipulate tags using third-party GRC, MDM, and other classification solutions
Anonymization

RBAC & DAC

Views & UDFs

Column-Level Security

Row Access Policy

Tagging

Classification

Anonymization

Protect personal data and retain analytical value
- Set level of protection to meet your internal policies
- Optimize for your analytic use case
- Help comply with privacy regulations

Maintain protection during updates
- Dynamically applied
- Role-based controls dictate who can view personal data at query time

Industry standard built for Snowflake
- Utilize native k-Anonymity algorithm
- Integrated, centralized access controls
- Share data internally and externally while protecting personal information

Anonymization

Data Governance/
Security & Governance delivered

- Automated controls
- Centralized governance with delegated authority
- Optimized analysis without sacrificing privacy and compliance concerns
- Compatible with third-party platforms
- Data-driven policies

**Anonymization**
- Automatically produced anonymized View

**Dynamic Data Masking**
- External Tokenization

**Row Access Policy**
- Tagging

**Column-Level Security**
- Views & UDFs
- Classification

**RBAC & DAC**
- All Data Gov Features

<table>
<thead>
<tr>
<th>Name</th>
<th>Gender</th>
<th>Age</th>
<th>Zip Code</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>male</td>
<td>39</td>
<td>79007</td>
<td>123-555-1234</td>
</tr>
<tr>
<td>Jane Doe</td>
<td>female</td>
<td>50</td>
<td>77001</td>
<td>333-555-1236</td>
</tr>
<tr>
<td>Mary Taylor</td>
<td>trans-fem</td>
<td>46</td>
<td>77020</td>
<td>222-333-1111</td>
</tr>
<tr>
<td>Gene Marshall</td>
<td>non-binary</td>
<td>48</td>
<td>77042</td>
<td>555-555-1234</td>
</tr>
<tr>
<td>Michael Gaines</td>
<td>male</td>
<td>75</td>
<td>79003</td>
<td>666-666-1357</td>
</tr>
</tbody>
</table>
GDPR, CCPA, and other emerging regulations allow individuals to request the deletion of their personal information, unless exceptions apply. This means your Snowflake recovery policies must properly align to your organization’s compliance policies.

Snowflake provides product features for customers to meet the demands of data privacy regulations.

- Time Travel & Fail-safe have data retention implications
- Up to 90 days (Time Travel)
- 7 days (Fail-safe)
- For PII erasure requests, you must consider Time Travel and its setting.

More on Time Travel & Fail-safe
Cross-Cloud & Cross-Region Replication
- Business continuity & disaster recovery
- Secure data sharing across regions/clouds
- Data portability for account migrations

Zero Performance Impact on Primary
- Asynchronous replication

Reduced Data Loss
- Incremental refreshes

Instant Recovery
- Read: Readable secondary databases
- Write: Database failover

Secure
- Data encrypted at-rest & in-transit
- Tri-Secret Secure compatible

Cost Effective
- Replication costs: Data transfer & compute (serverless)
- Control which databases to replicate

More about Database Replication & Failover
Hierarchical Key Model

- Hierarchical key model rooted in the CSP’s HSM
  - GCP: [Cloud HSM](#)
  - AWS: [Cloud HSM](#)
  - Azure: [Dedicated HSM](#)
- All data at rest is encrypted by default, with no configuration required

**Account Master Key**

**Object Master Keys**
- (e.g. Table Master Keys, Result Master Keys, Stage Master Keys)

**File Keys**

More resources on Key Management
**Key Rotation**
- Snowflake rotates keys every 30 days
- Process is transparent to customer and queries

**Key Re-Keying**
- Yearly re-keying re-encrypts data on the key's birthday
- Re-keying requires Enterprise Edition or better
- Process is transparent to customer and queries

More resources on Key Management
Hierarchical Key Model using Tri-Secret Secure

- Hierarchical key model adds a hybrid HYOK & BYOK model to give the customer control.
- Customer holds key in their CSP Key Management and brings key materials to Snowflake to be part of the key-encrypting key (the Account Master Key or AMK).
- CSP-supported key managers:
  - GCP: Cloud KMS
  - AWS: AWS KMS
  - Azure: Key Vault
AUDIT LOGGING – ACCOUNT USAGE

Auditing tracks every user’s activity at all times in full detail

Kept in a tamper-proof area of your account for 365 days

All supplied drivers and connectors also have extended logging

Possibly Interesting Blocked AuthN Events

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>ERROR_MESSAGE</th>
<th>CLIENT_IP</th>
<th>USER_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:09:52</td>
<td>INCOMING_IP_BLOCKED</td>
<td>2471555</td>
<td>UNIVERSE-1-DEMO-US-MIKE</td>
</tr>
<tr>
<td>08:02:33</td>
<td>USER_NAME_MISMATCH</td>
<td>182209</td>
<td>SETHY.YOUSSEF@SNOWFLAKESEI</td>
</tr>
<tr>
<td>12:37:35</td>
<td>PASSWORD_EXPIRED</td>
<td>877777</td>
<td>DMITRI</td>
</tr>
<tr>
<td>03:28:45</td>
<td>USER_ACCESS_DISABLED</td>
<td>373737</td>
<td>SPIDERMAN</td>
</tr>
<tr>
<td>22:00:29</td>
<td>USER_ACCESS_DISABLED</td>
<td>121212</td>
<td>UATU</td>
</tr>
<tr>
<td>22:00:26</td>
<td>USER_ACCESS_DISABLED</td>
<td>121212</td>
<td><a href="mailto:THE_PHILosopher@GMAIL.COM">THE_PHILosopher@GMAIL.COM</a></td>
</tr>
<tr>
<td>22:00:25</td>
<td>USER_ACCESS_DISABLED</td>
<td>121212</td>
<td>TEST1</td>
</tr>
<tr>
<td>22:00:23</td>
<td>USER_ACCESS_DISABLED</td>
<td>121212</td>
<td>SPIDERMAN</td>
</tr>
</tbody>
</table>

Why Are You Running as ACCOUNTADMIN?

<table>
<thead>
<tr>
<th>QUERY_TYPE</th>
<th>USER_NAME</th>
<th>START_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>DMITRI</td>
<td>2021-06-14 02:31:53.282 -0700</td>
</tr>
<tr>
<td>SHOW</td>
<td>BORING</td>
<td>2021-04-05 05:47:59.189 -0700</td>
</tr>
<tr>
<td>SHOW</td>
<td>SETHY</td>
<td>2021-02-24 08:58:16.539 -0800</td>
</tr>
<tr>
<td>SHOW</td>
<td>JSANDER</td>
<td>2020-11-19 08:15:10.216 -0800</td>
</tr>
<tr>
<td>ROLLBACK</td>
<td>WADEWILSON</td>
<td>2020-10-26 08:38:54.073 -0700</td>
</tr>
<tr>
<td>SELECT</td>
<td>SYSTEM</td>
<td>2020-10-14 15:38:47.130 -0700</td>
</tr>
<tr>
<td>SHOW</td>
<td>JOHN</td>
<td>2020-09-14 05:52:30.765 -0700</td>
</tr>
<tr>
<td>SHOW</td>
<td>EUGENE</td>
<td>2020-08-27 15:55:49.126 -0700</td>
</tr>
<tr>
<td>SHOW</td>
<td>RYANO</td>
<td>2020-08-23 12:29:40.470 -0700</td>
</tr>
</tbody>
</table>

Active Stages

<table>
<thead>
<tr>
<th>STAGE_LOCATION</th>
<th>LAST_LOAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>s3://aws-cse-testing/</td>
<td>2021-04-12 11:34:41.050 -0700</td>
</tr>
<tr>
<td>s3://aws-cse-testing</td>
<td>2020-10-19 12:56:49.375 -0700</td>
</tr>
</tbody>
</table>

AuthN Breakdown

- PASSWORD_MFA
- PASSWORD_NOMFA
- ALL_PASSWORD
- SAML
- OAuth
- KEYPAR

© 2022 Snowflake Inc. All Rights Reserved
Access History supplies audit data access to comply with regulatory requirements and data governance initiatives.

- Access log of the tables, views, and columns each query accesses
- Includes base objects (e.g. table serving a view) indirectly accessed by the query
- Data access history available for easy reporting as a Account Usage View

Discover unused data to determine whether to archive or delete the data.

Notify users prior to altering a table, view, or column.
Results can be further filtered using SQL predicates

Export through JDBC or as JSON for use in SIEM
How is the Snowflake Infosec Team monitoring the service?

Snowflake’s internal Critical Security Controls dashboard provides real-time risk visibility
- Access Control, Security Assessment & Authorization, Configuration Management, Security Awareness, etc. all represented on a single Dashboard
- Real-time monitoring of data loaded into Snowflake from internal and other relevant data sources

Snowflake uses CIS benchmark templates for configuration hardening
- Service configuration information is collected centrally in Snowflake
- Continuously and automatically tracked—unplanned changes cause alerts
- Part of the Snowflake Security Compliance Team’s dashboard

Snowflake undergoes independent pentests
- Comprehensive Web Application Penetration Test – Annually
- Internal Network Penetration Test – Annually
- Major Functionality Penetration Tests – As major functionality is released as part of the SDLC

Snowflake performs weekly vulnerability scans on infrastructure
- Vulnerabilities are remediated per Security Policy
- Remediation trends tracked using Snowflake
GDPR – GENERAL DATA PROTECTION REGULATION

What is it?

- GDPR is an EU regulation that went into effect on May 25, 2018
- Governs the protection and processing of EU personal data

What does it mean in the context of Snowflake?

Different requirements apply to different types of entities:

- **Controller** – Snowflake customers are responsible for complying with GDPR independently from Snowflake
- **Processor** – Snowflake is responsible for the following:
  - Putting data processing addendums in place with our customers and our vendors
  - Only using our customers' EU personal data to provide our service to them
  - Being transparent about how we handle and process our customers' EU personal data on their behalf and keeping accurate records
  - Securing customers' EU personal data in our service
  - Facilitating our customers' compliance with data subject requests
  - Notifying customers about changes to our list of subcontractors

Snowflake responsibilities are documented in a Data Processing Addendum (DPA on snowflake.com/legal).

Available for signature now
USEFUL LINKS

- [Snowflake Security Overview and Best Practices](#)
- [Snowflake Security Product Documentation](#)
- [Managing Governance in Snowflake](#)