

Snowflake で何が起きているのかを把握する

観測で外側を閉じる — Snowflake 運用シリーズ 3/3

ある夜の典型

- ダッシュボードが突然遅い — 誰のクエリ？
- WHは動いているのに クレジットが溶ける
- 「さっきまで動いてた」 — 何が変わった？

事後調査用のビュー と 今すぐ切り分け は別物。

ACCOUNT_USAGE の落とし穴

- 遅延（最大 45 分～数時間）がある
- 粒度は 日次・集計 向き — 秒単位のインシデント には弱い
- 権限・ロールによって 見えない列 がある

今日のねらい

ClickOps 限界（1 本目）・dbt の切り分け（2 本目）を踏まえ、

Snowflake 上で「今何が起きているか」をチームで共有する方法を整理する。

2. 「把握」とは何か

何を見れば「起きている」と言えるか

把握の五層

層	例
クエリ	誰が何を走らせ、どれだけ時間・IO か
WH	サイズ・キュー・稼働 / 停止
クレジット	どの WH・どのワークロードが食うか
パイプライン	dbt run / 外部 Orchestrator の成否
人の操作	GRANT・DDL・WH 変更 (UI / SQL)

全部見えなくていいー インシデント種別ごとに 最低セット を決める。

シリーズとの位置づけ

- 1 本目: [Snowflake と ClickOps の限界](#) — 変更履歴が残りにくい
- 2 本目: [Snowflake と dbt](#) — 変換の内側とプラットフォームの外側
- 本日: 外側を観測 して、判断とアクションを速くする

3. データソースの地図

どこを見れば何が分かるか

リアルタイム寄り

- `INFORMATION_SCHEMA.QUERY_HISTORY` (セッション / 直近)
- `SNOWFLAKE.ACCOUNT_USAGE.QUERY_HISTORY` — 履歴・分析 (遅延あり)
- `WAREHOUSE_LOAD_HISTORY` / `WAREHOUSE_METERING_HISTORY`

監査・誰が何を

- `ACCESS_HISTORY` — オブジェクトへのアクセス
- `LOGIN_HISTORY` — ログイン・クライアント
- `GRANTS_TO_*` / `OBJECT_DEPENDENCIES` — 権限ドリフトの手がかり

落とし穴（共通）

- 遅延・粒度 — アラート設計と期待値を揃える
- 権限 — `IMPORTED PRIVILEGES` / 監視ロールの 最小 **GRANT**
- コスト — 大テーブル全スキャンは 別 **WH** で

dbt / Orchestrator との突合

- **dbt artifacts** (`manifest.json` / run results) — どのモデルがいつ
- Snowflake 側は **QUERY_TAG** / **SESSION** パラメータ で run を束ねる
- 外部 Orchestrator の **run id** を tag に載せると 一本の線 になる

4. 見せ方

ダッシュボード vs アラート vs 定期レビュー

ダッシュボード

- 探索・説明 向き（インシデント中の「全体像」）
- 陥穽: 作って終わり、誰も見ない

アラート

- 閾値・異常 で起きる（クレジット急増、長時間クエリ、WH キュー）
- 陥穽: ノイズ ー ランブックなしのアラートは無視される

毎朝見る 1 枚

- 定例レビュー 用の固定ビュー（前日クレジット TOP、失敗 run、長時間 TOP N）
- ダッシュボードより 軽く、アラートより 能動的
- 「把握できた」の 最低ライン をここで定義する

使い分けの目安

アラート → 今すぐ人が動く

毎朝 1 枚 → ドリフト・コストの早期発見

ダッシュボード → 調査・共有・振り返り

5. アクションに繋げる

調査から再発防止へ

調査の型

1. 症状（遅い / 高い / 落ちた）を時間で切る
2. **QUERY_HISTORY + tag** で犯人クエリ / run を特定
3. **WH / サイズ / 並列** を確認 — 一時対応か恒久か判断

再発防止の三つ

手段	例
ポリシー	本番 WH サイズ上限、ad hoc 用 WH 分離
自動化	長時間クエリ kill、自動 suspend、laC 差分
ランブック	「クレジット 2 倍」時のチェックリスト (SQL 付き)

人の判断の残し方

- 一時 XL 上げ → **Issue** / チケット に理由と戻し日
- 緊急 GRANT → 期限付き + ACCESS_HISTORY でレビュー
- ClickOps 限界 (1 本目) の 意図 を観測とセットで残す

3 本セットの結論

- コード化 — IaC / dbt で「あるべき姿」と履歴 (1・2 本目)
- 観測 — 遅延と粒度を理解した上で、クエリ・WH・クレジット・人を見る (本日)
- 人の判断 — 自動化できない部分を ランブック と記録 で残す

持ち帰り

- ACCOUNT_USAGE だけ ではインシデントに間に合わないことがある
- 把握の最低ライン（毎朝 1 枚 + 重要アラート）をチームで決める
- 観測は 禁止の代わり ではなく、早く気づいて直す ための土台

ありがとうございました

質問・失敗談（匿名化）歓迎・@ta93abe_