

Snowflake と ClickOpsの限界

「ポチポチ」をシステムの外に出す —
Snowflake 運用シリーズ 1/3

あるある質問

- 本番の WH を誰が止めた？
- いつロールの GRANT が変わった？
- 検証と本番、今どっちを触っている？

全員が即答できないのが普通になる。

今日のねらい

Snowflake 運用が コンソールでポチポチ (**ClickOps**) に寄りすぎたとき、

- 何が 壊れやすい か
- どこまでが 現実的な限界 か

を言語化する。

2. ClickOps とは

良い面と、隠れコスト

ClickOps とは

UI / コンソール操作だけ でインフラ・設定を回す運用。

Snowflake だと: Worksheets、Admin、WH サイズ変更、GRANT のクリック操作。

良い面

- 速い — 障害時・検証時に即試せる
- 学習コストが低い — SQL より GUI の方が入りやすい
- 小規模チームでは十分 — 全員が同じ画面を見れば足りる

隠れコスト

- 誰が・いつ・何を 変えたかが残らない
- 手順が 属人化 する（「あの人しか知らない」）
- 本番と検証の 境界 が UI 操作だけでは守れない

3. 限界が出る 典型パターン

チームが育つと必ず当たる壁

権限・ロールのドリフト

- 緊急対応で一時 **GRANT** → 戻し忘れ
- ロール名は同じ、中身が環境ごとにバラバラ
- 「とりあえず ACCOUNTADMIN」が常態化

クレジット / WH のその場対応

- 遅い → **XL** に上げる → 請求が跳ねる
- 止め忘れ・サイズ戻し忘れ
- ポリシー より その場の判断 が勝つ

本番と検証の境界

- 同じアカウント内の DB / WH を 名前だけ で分けている
- UI のドロップダウンを間違える ヒューマンエラー
- 「本番禁止」は 運用ルール だけで、技術的に縛れない

監査・再現性・オンボーディング

- 監査: いつ誰が を UI ログだけでは説明しきれない
- 再現: 新メンバーが 同じ環境を再現 できない
- オンボ: 「先輩の横について覚える」が 唯一のドキュメント

4. 限界の正体

単一 UI では三つが残らない

状態・意図・変更履歴

残りにくいもの	意味
状態	今のロール・WH・ポリシーの正
意図	なぜそう変えたか（一時対応か恒久か）
変更履歴	いつ誰が何から何へ

ClickOps 単体では **Git** のような正 が立てにくい。

だから壊れる

- ドリフトに 気づくのが遅い
- 障害時の ロールバック が「もう一度ポチポチ」
- シリーズ 2・3 (dbt / 観測) へ進む 動機 になる

5. 脱 ClickOps の方向性

「禁止」ではなく「外に出す」

IaC / Git

- ロール・WH・DB をコード で定義
- PR・レビュー・差分で 意図と履歴 を残す
- Snowflake: Terraform / schemachange など

ポリシー

- 本番変更は パイプライン経由のみ
- クレジット上限・WH サイズの ガードレール
- ACCOUNTADMIN の 日常利用禁止 (break-glass は別)

観測

- QUERY_HISTORY / ACCESS_HISTORY で 誰が何を
- コスト・WH 利用率の アラート
- 「ポチポチ禁止」より 異常検知で早く気づく

橋渡し

本日: 問題提起 (1/3)

- 2本目: dbt など 変換・パイプライン の話
- 3本目: 観測・把握 で運用を閉じる

持ち帰り

- ClickOps は 悪 ではない。小さく始めるには 最適
- 限界は スケール・監査・属人化 に見える
- 「ポチポチ禁止」ではなく ポチポチをシステムの外に出す

ありがとうございました

質問・失敗談（匿名化）歓迎・@ta93abe_