

AuroraからTiDBへ!!

安定運用タイトルで挑んだTiDB移行と
NewSQLの可能性

はじめに

今回お話しする内容は、
運用6年目のとある音ゲープロジェクトを
AuroraからTiDBへ移行した事例についてです

TiDB GAME DAY
2025

運用中タイトルでも移行できる！！

ゲーム業界初AuroraからTiDBへの
ゲームデータ移行の舞台裏

笑顔あふれるサービス提供
altplus

株式会社オルトプラス 楠元・浦谷



撮影OK



SNS投稿OK

運用6年目のタイトルの挑戦！

音ゲーアプリだけに
AuroraからTiDBへ「テンボよく」移行！！

笑顔あふれるサービス提供
altplus

CEDEC2025 7/24 18:30-

株式会社オルトプラス 浦谷 和茂

移行事例で登壇したセッションは2本あるので
今回はテイストを変えてお伝えします！！

- TiDB導入の背景
- NewSQLの水平スケールを支える仕組み
- なぜTiDBなのか
- TiDB Cloud選定の決め手
- TiDB Cloudへの移行
- 移行後について



株式会社オルトプラス 技術部/エンジニアマネージャー

kazushige uratani

浦谷 和茂

2023年6月にオルトプラスに入社

プロジェクトを横断的に見ていますー！！ SRE見えていますー！！

入社してからずっと「TiDBの推し活」をしました(笑)

TiDBのユーザーコミュニティ「TiUG」も盛り上げTi！！

笑顔あふれるセカイを増やす

オルトプラスについて



笑顔あふれるセカイを増やす

altplus



オルトプラスの主なタイトル

Everybody Shogi
(えぶりばでい将棋)



新感覚カジュアル将棋
パズルゲーム

忘却前夜



戦略的なバトルが特徴
のスマホ向けカードRPG

戦国小町苦勞譚 語絵巻
- カタリエマキ -



フルボイス化&平沢下戸先生の
魅力溢れるヴィジュアルで
再構築したノベルアプリ

笑顔あふれるセカイを増やす

TiDB導入の背景

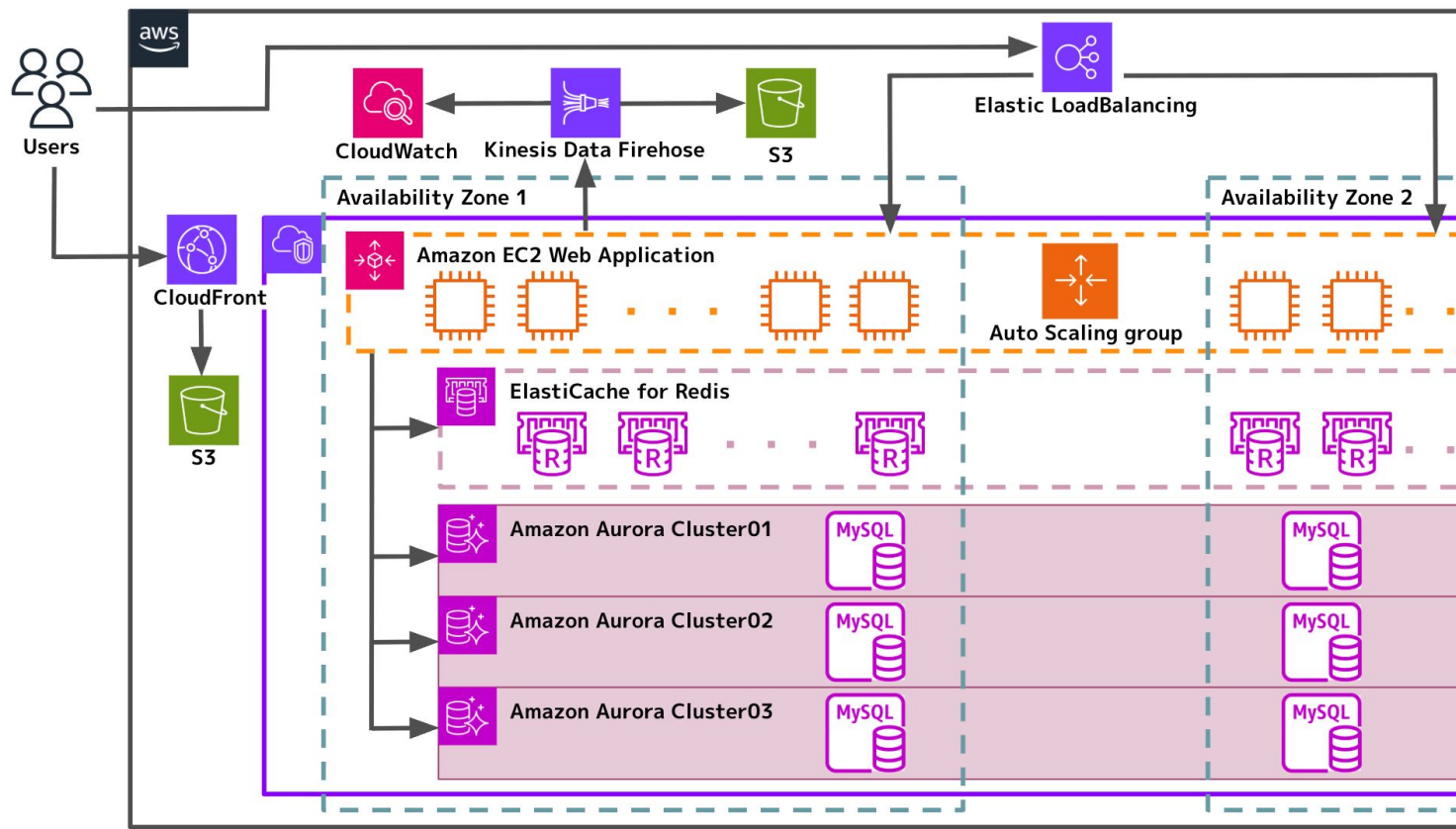


運用6年目を迎えた音ゲープロジェクトでは、
Auroraクラスターで安定運用していました。

それではなぜ
安定運用中のタイトルを
TiDBに刷新したのでしょうか？

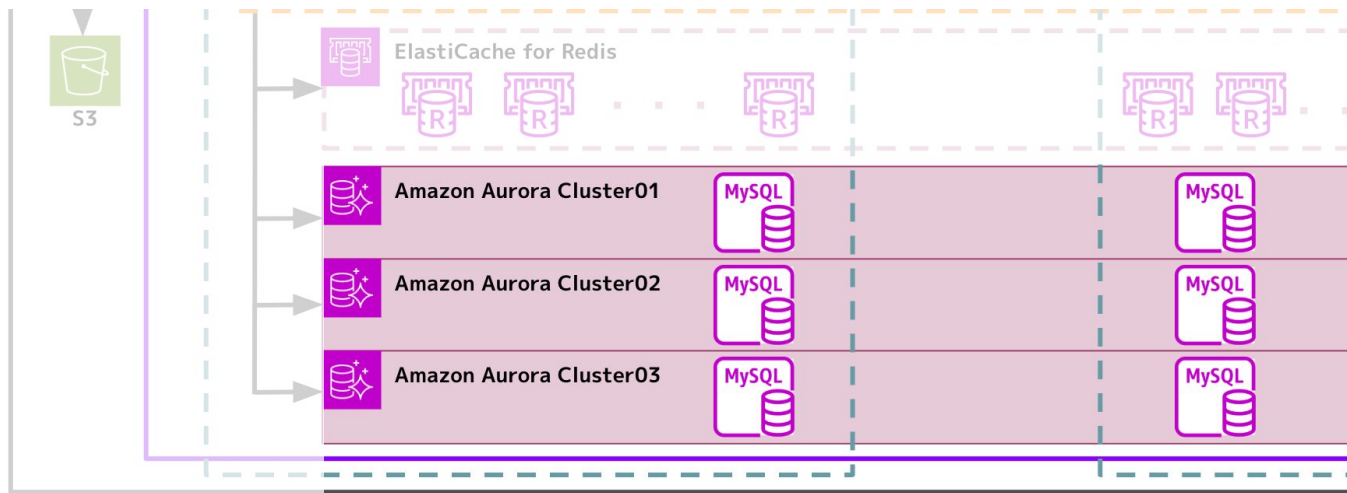


TiDB導入の背景：移行前構成図



aws

Auroraクラスター3台（Provisioned）で運用



Auroraは1クラスターで完結する前提の設計だが、
ゲームの成長にあわせて**クラスターを横に増やして対応**



クラスター増加に伴って**ベースコストが増加**

問題点

スケールはクラスター単位

- AuroraはWriterを増やせない → Range Sharding構成で運用
- イベント時など高負荷時 → 各クラスターでリソース調整が必要

理想

Writerがスケールし、かつシャーディング不要で運用できるデータベースがあれば・・・

笑顔あふれるセカイを増やす

NewSQLの水平スケールを支える仕組み



データベースの種類と特徴

データベース	特徴
RDB	強い整合性・複雑なJOIN・トランザクションが得意 スケールアップ中心で、スケールアウトは苦手
NoSQL	高いスケーラビリティ・柔軟なスキーマ SQL/トランザクション前提の設計には不向き
NewSQL	RDBのトランザクション/SQL互換性 NoSQL並みのスケールアウト性能 水平方向にスケール可能なアーキテクチャで、 従来のRDBが苦手とする書き込み負荷にも対応できる

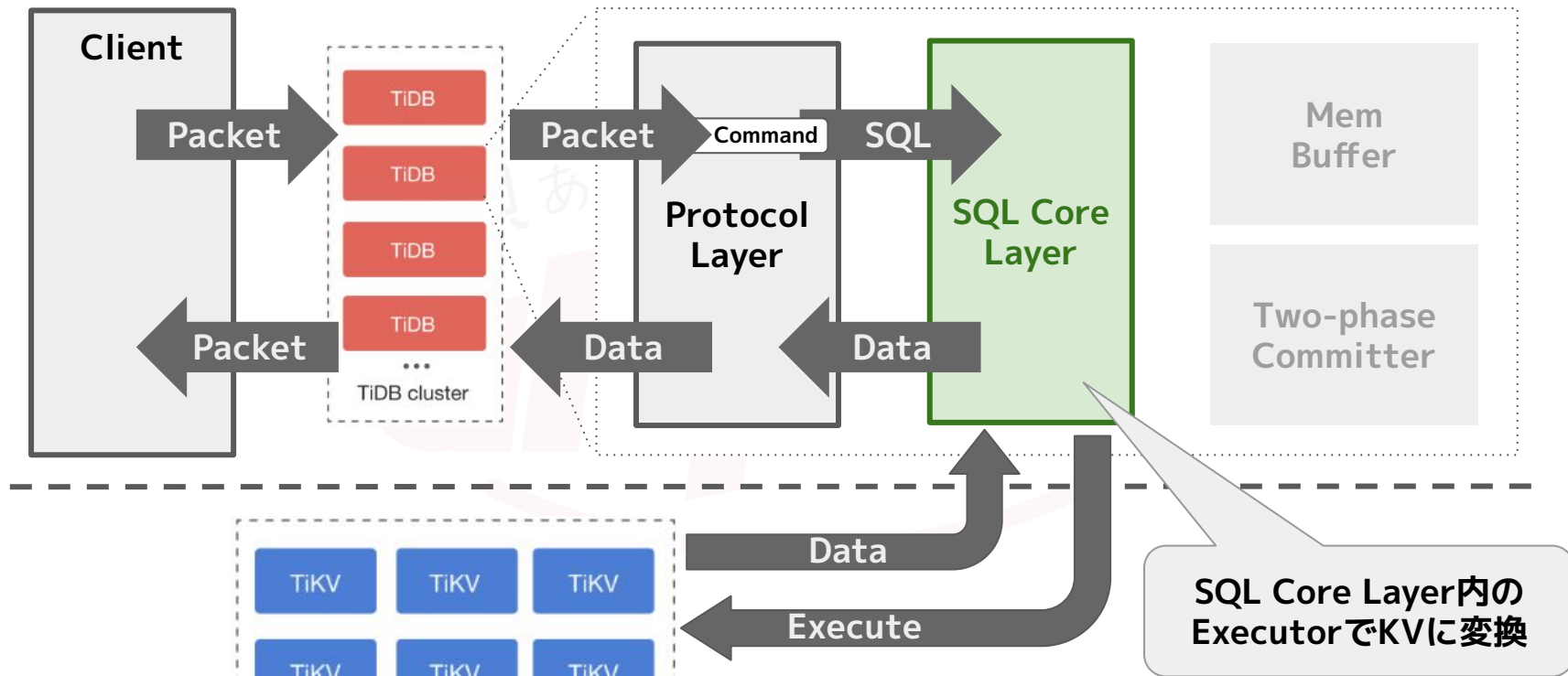
NewSQLが水平方向にスケールできるのは、
データを**Key-Value形式**に分割して
分散管理できる設計になっているためです！！

**一意なキーをもとにデータを分割できるため、
各ノードが独立して読み書きが可能！！**

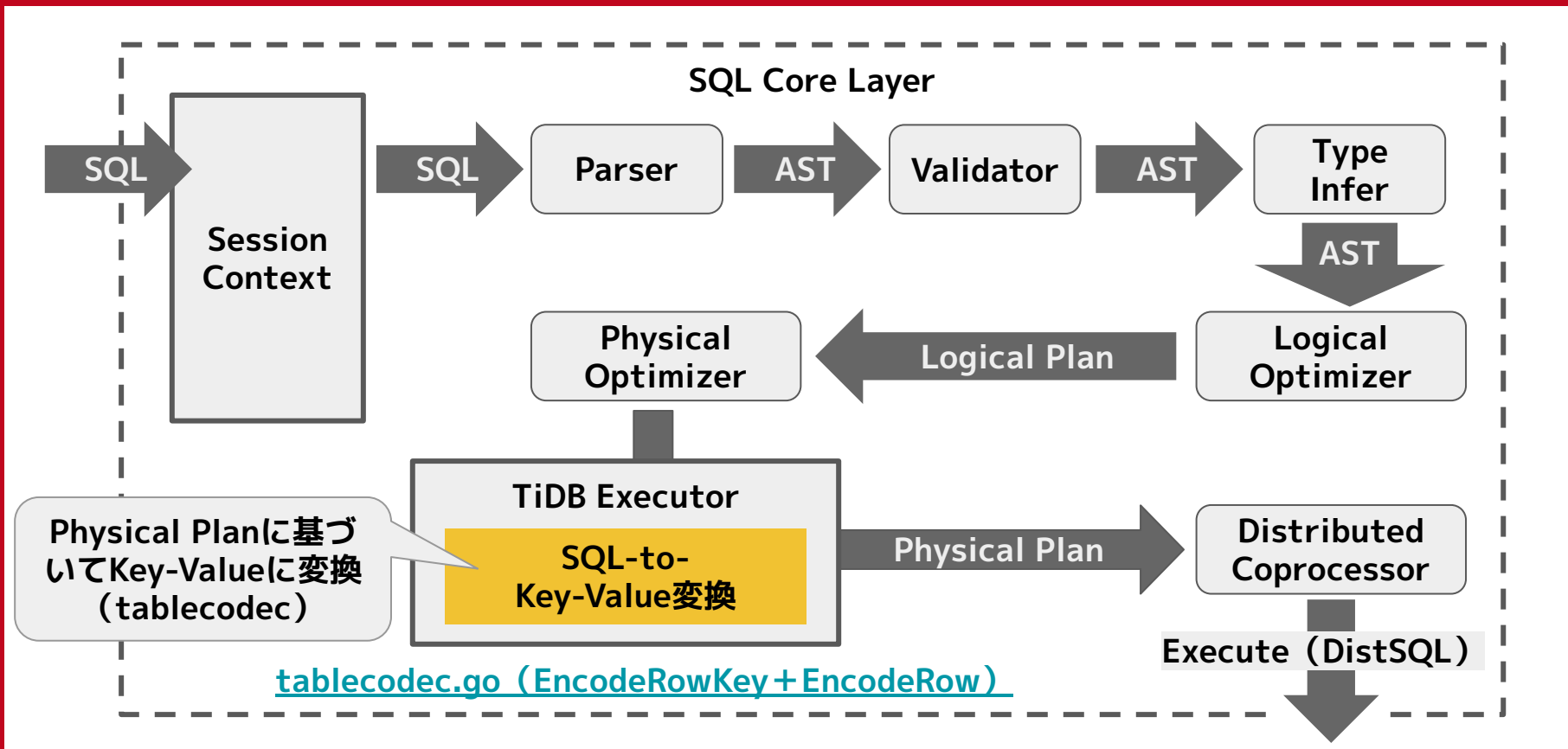
TiDBも同様SQLを内部で**Key-Valueに変換**して保存することで、1台のWriterではなく「**クラスタ全体で書き込み処理ができる**」構造になっています。

TiDBでは「SQL Core Layer」の
「Executor」内でKey-Valueに変換

NewSQLの水平スケールを支える仕組み：SQLレイヤーのデータフロー（全体像）



NewSQLの水平スケールを支える仕組み：SQL Core Layerのデータフロー



NewSQLの水平スケールを支える仕組み：RowからKey-Valueへの変換イメージ

[ROW]

1, "TiDB", "SQL Layer", 10

2, "TiKV", "KV Engine", 20

3, "PD", "Manager", 30

tablecodec

[Key-Value]

t10_r1 --> ["TiDB", "SQL Layer", 10]

t10_r2 --> ["TiKV", "KV Engine", 20]

t10_r3 --> ["PD", "Manager", 30]

ドキュメントを参照するとより理解が深まります！！

[SQLレイヤーのアーキテクチャ]

[https://docs.pingcap.com/ja/tidbcloud/tidb-computing/
#architecture-of-sql-layer](https://docs.pingcap.com/ja/tidbcloud/tidb-computing/#architecture-of-sql-layer)

[キーと値のマッピング関係の例]

[https://docs.pingcap.com/ja/tidbcloud/tidb-computing/
#example-of-key-value-mapping-relationship](https://docs.pingcap.com/ja/tidbcloud/tidb-computing/#example-of-key-value-mapping-relationship)

笑顔あふれるセカイを増やす

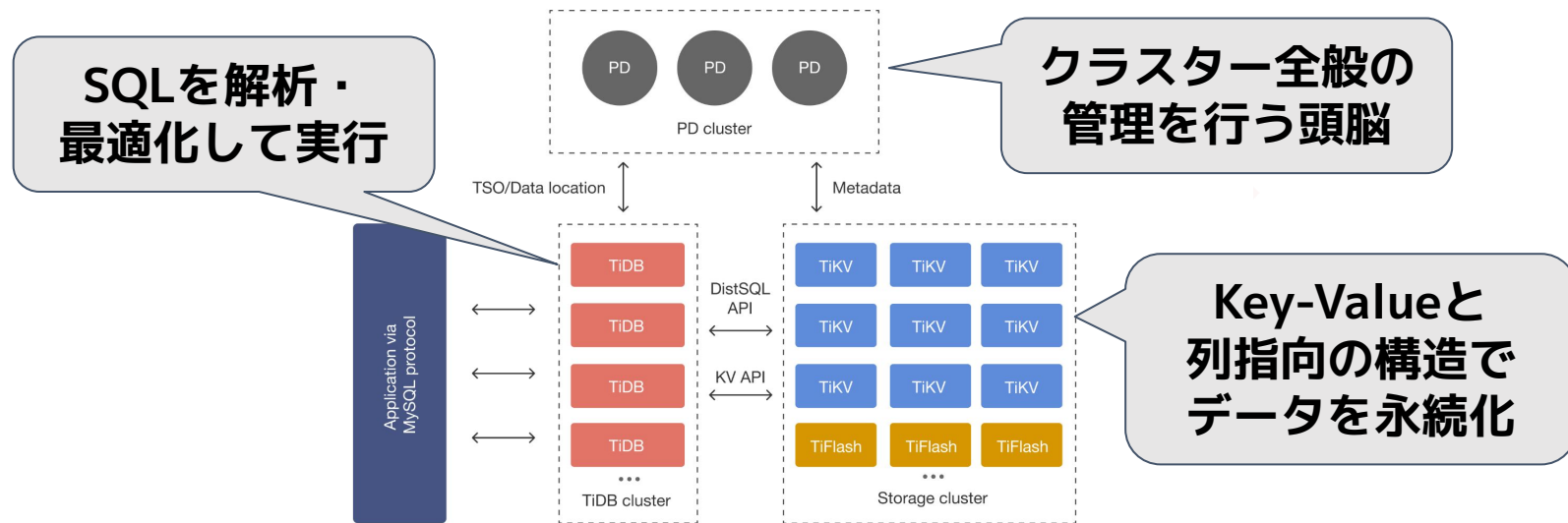
なぜTiDBなのか



Auroraは高い安定性とマネージド運用を実現し、
長年その恩恵を受けてきました。

一方で、より柔軟なスケールや分散構成を求めて、
NewSQLであるTiDBを選択しました！！

TiDBは3つのコンポーネントに分かれているため、
より柔軟なスケールや分散構成が可能です



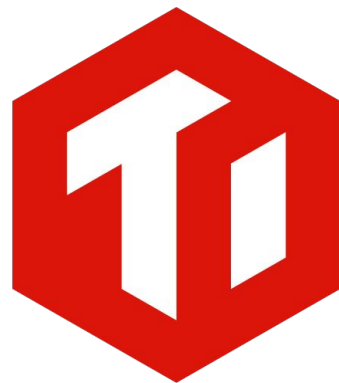
<https://docs.pingcap.com/tidb/stable/tidb-architecture>

なぜTiDBなのか



TiDBのStorage cluster「TiKV」のスケールで
Writerのスケールが可能に！！

Writerのスケール
シャーディング不要



TiDBの分散機能によりデータと負荷が
自動分散されシャーディングが不要に！！

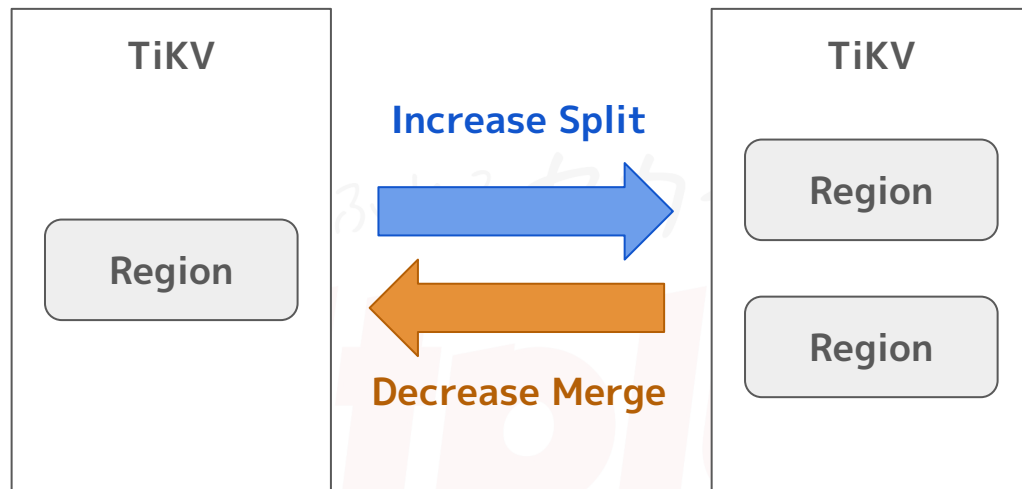
NewSQLの中で、
TiDBを選んだ理由は何故か？



①シャーディング不要で柔軟なスケール

- TiDBは自動でデータが分散される
 - シャーディング構成が不要
- 必要なリソースに合わせたスケール
 - 書き込み性能もノード追加で拡張可能

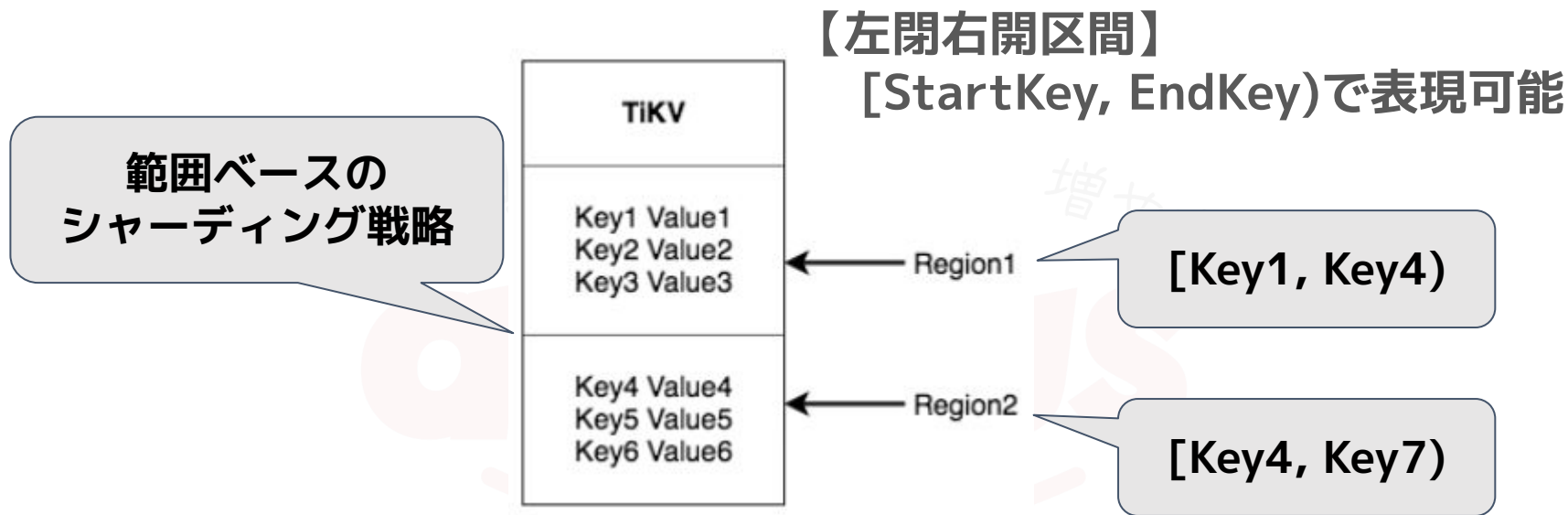
①シャーディング不要で柔軟なスケール：自動シャーディングの仕組み



96MiB以上で分割して20MiB以下でマージする
(上記サイズはデフォルトで変更可能)

v8.4.0以降では、Regionのデフォルトサイズが96MiBから256MiBに変更

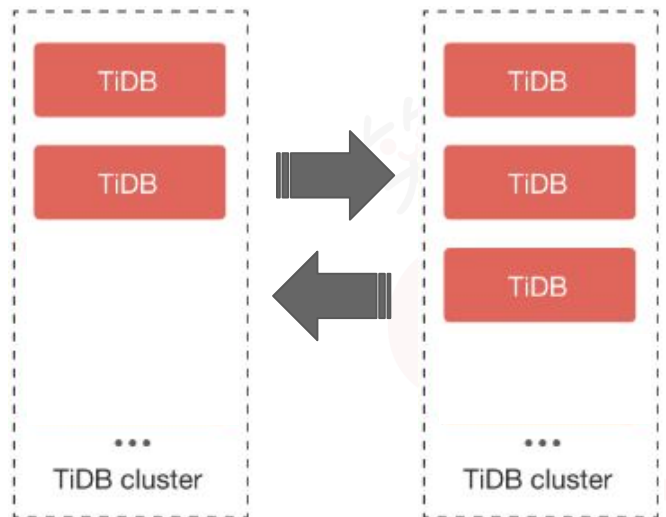
①シャーディング不要で柔軟なスケール：自動シャーディングの仕組み



<https://docs.pingcap.com/ja/tidb/stable/tidb-storage#region>

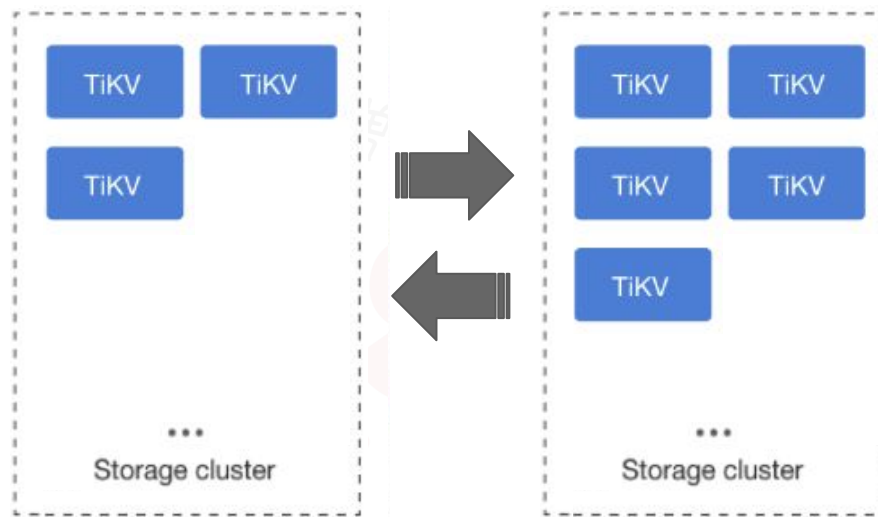
①シャーディング不要で柔軟なスケール：コンポーネント単位でのスケール

コンピューティング



[短期的]
QPSやコネクション数を元に増減

ストレージ



[短・中・長期的]
QPS・IOPS・ストレージを元に増減

②MySQLとの互換性の高さ

- MySQL5.7 + 8.0 (v7.5 LTS)
 - 従来のMySQLとほぼ同じ感じで使える

但し、FULLTEXT INDEX、ストアドプロシージャ、トリガー等サポートしていない機能も一部あります

②MySQLとの互換性の高さ：FULLTEXT INDEXについて



現時点では特定のAWSリージョンのTiDB Cloud StarterとEssentialのみ対応

AWS: Frankfurt (eu-central-1) and Singapore (ap-southeast-1)

<https://docs.pingcap.com/tidbcloud/vector-search-full-text-search-sql/>

②MySQLとの互換性の高さ：各データベースへの接続イメージ

MySQLクライアント

`mysql -uxxx -p -hxxx -port=3306`

MySQL Client

`mysql -uxxx -p -hxxx -port=4000`

ホストとportを変更
するだけで接続先が変わる

データベース

port:3306

MySQL

TiDB

TiDB

TiDB

TiDB

...

TiDB cluster

port:4000

自分は音ゲーの運用について日頃から考えていたことがありました。

- Auroraクラスター3つの運用はコストがかかるのでは？
- シャーディングによるリソース調整が難しいのでは？

TiDBが向いているのではないか？

弊社CTOと毎週話す機会があり提案



まずは音ゲーに入れたらどうだろう？



やっちまおうぜ！

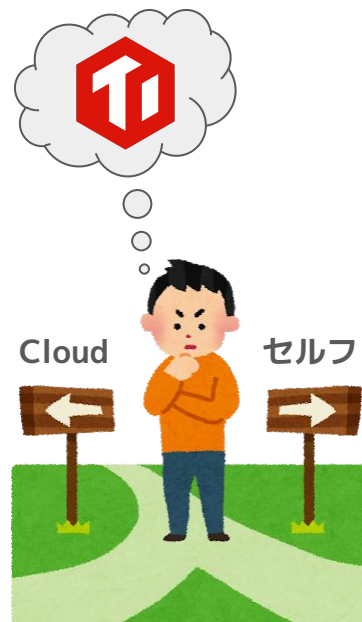
弊社CTOの有馬さんはこんな感じ（笑）

笑顔あふれるセカンドを増やす

TiDB Cloud選定の決め手



TiDBはセルフマネージドとフルマネージドを選ぶが、
それではなぜ、
TiDB Cloudを選定したのか？

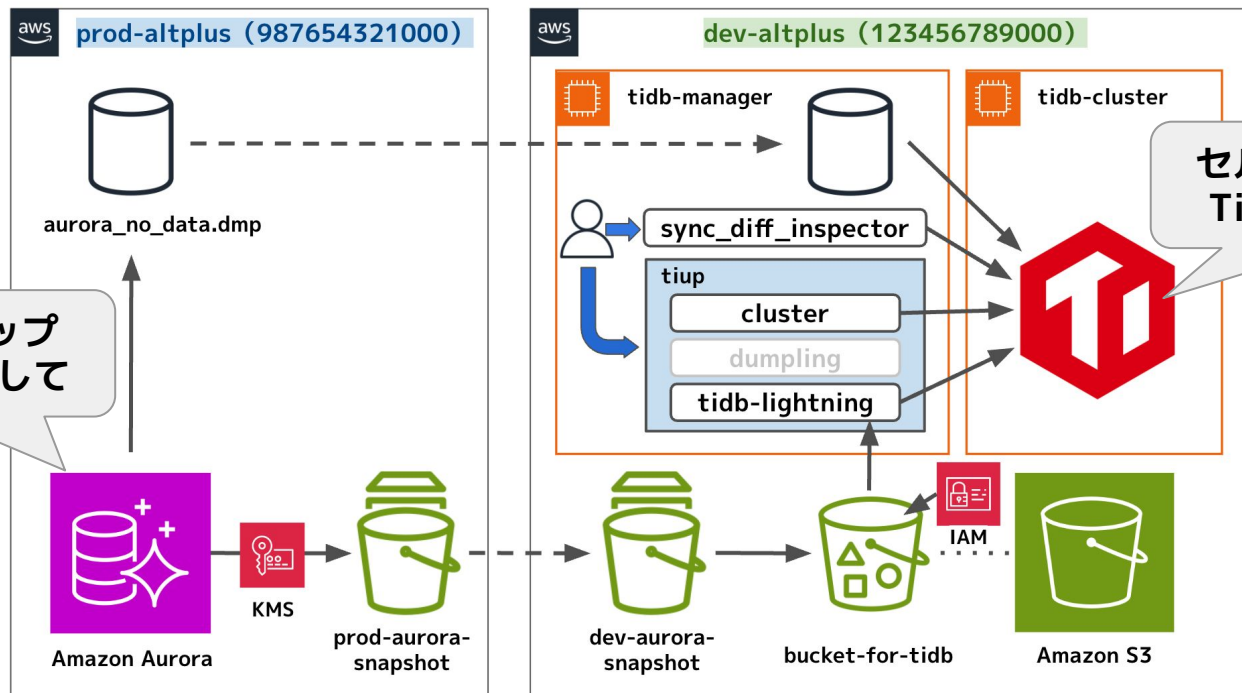


① マネージドサービス

- クラウド上でフルマネージド提供
 - インフラ構築や運用管理の負担を削減できる
 - サポートによる対応の恩恵を受けやすい
- スケーリングやバックアップも自動化
 - チームの運用コストや導入のハードルも低い

①マネージドサービス：セルフマネージドはどんなの？

セルフマネージドのTiDB環境を構築して移行検証をしていた時もありました
TiUPコマンドを使用した運用も自動化できるので使いやすかったです！！



Auroraのスナップ
ショットを取得して

セルフマネージドの
TiDBへデータ移行

①マネージドサービス：マネージド・セルフマネージド比較

観点	マネージド (TiDB Cloud)	セルフマネージド (ソフトウェア)
導入ハードル	学習コストが低く導入が容易	専門知識と初期構築が必要
運用負荷	クラウド上でフルマネージド提供 (インフラ構築や運用管理の負担削減)	構築や保守などは自社で実施 (初期構築や運用ノウハウの負担がある)
バックアップ	自動バックアップ機能を提供	設定や保守は自社で対応
アップデート	自動的に最新バージョンへ反映	バージョン管理や検証を自社で実施
サポート	チケットベースの公式サポート (Jiraのチケット対応で迅速)	社内リソース依存で負担が大きい (ただしテクニカルサポートは可能)

②シンプルで直感的なUI

- TiDB CloudのUIは操作がわかりやすい
 - 学習コストが低い
 - 画面から直感的にわかる
- チームメンバーに説明する時間が減り、導入初期のハードルが低い

②シンプルで直感的なUI：UIがシンプルで直感的だと感じる理由

①OrganizationやProjectを選択

②ナビゲーションペインから項目を選択

③選択内容が表示される

（左）のナビゲーションペインで選択した内容が、（右）のメインに表示されるUI構成のため、シンプルで直感的に操作できると感じています。

笑顔あふれるセカンドを増やす

TiDB Cloudへの移行



①移行手順と工夫

ドキュメントを確認しながら移行の方法を検討

The screenshot shows the TiDB Docs website. The top navigation bar includes links for TiDB Docs, TiDB Cloud, TiDB Self-Managed, Learning Center, and Contact Us. A secondary bar promotes TiDB Labs. The left sidebar shows a version selector set to v8.5 and a navigation menu with items like Docs Home, About TiDB Self-Managed, Get Started, Develop, Deploy, Migrate (expanded), Overview, Migration Tools (highlighted), and Import Best Practices. The main content area displays the 'TiDB Migration Tools Overview' page, which mentions 7 contributors and describes various data migration tools. It also includes a section for 'TiDB Data Migration (DM)' with a list of user scenarios, upstreams, and downstreams.

TiDB Docs TiDB Cloud TiDB Self-Managed Learning Center Contact Us

Explore the new TiDB Labs, learn distributed SQL database and

v8.5

Docs Home

> About TiDB Self-Managed

> Get Started

> Develop

> Deploy

▼ Migrate

Overview

Migration Tools

Import Best Practices

TiDB Migration Tools Overview

7 Contributors

TiDB provides multiple data migration tools for different scenarios such as backup and restore, and data replication.

This document introduces the user scenarios, supported upstreams and downstreams of these tools. You can choose the right tool according to your needs.

TiDB Data Migration (DM)

- **User scenario:** Data migration from MySQL-compatible databases:
- **Upstream:** MySQL, MariaDB, Aurora
- **Downstream:** TiDB

検討した結果下記ツールを採用（セルフマネージド）

ツール	説明
Dumpling (エクスポート)	テーブル定義やバックアップを取得する際に使用 出力ファイルは「SQL、CSV」、出力先は「ローカル、S3」、 分割サイズは256MiBが推奨
sync-diff-inspector (整合性チェック)	移行データを比較する用途で使用（チェックサム） テーブル定義或いはデータ不一致の場合は修正用のQUERYを出力 ※動いている環境で実行すると差分がでてしまうので注意

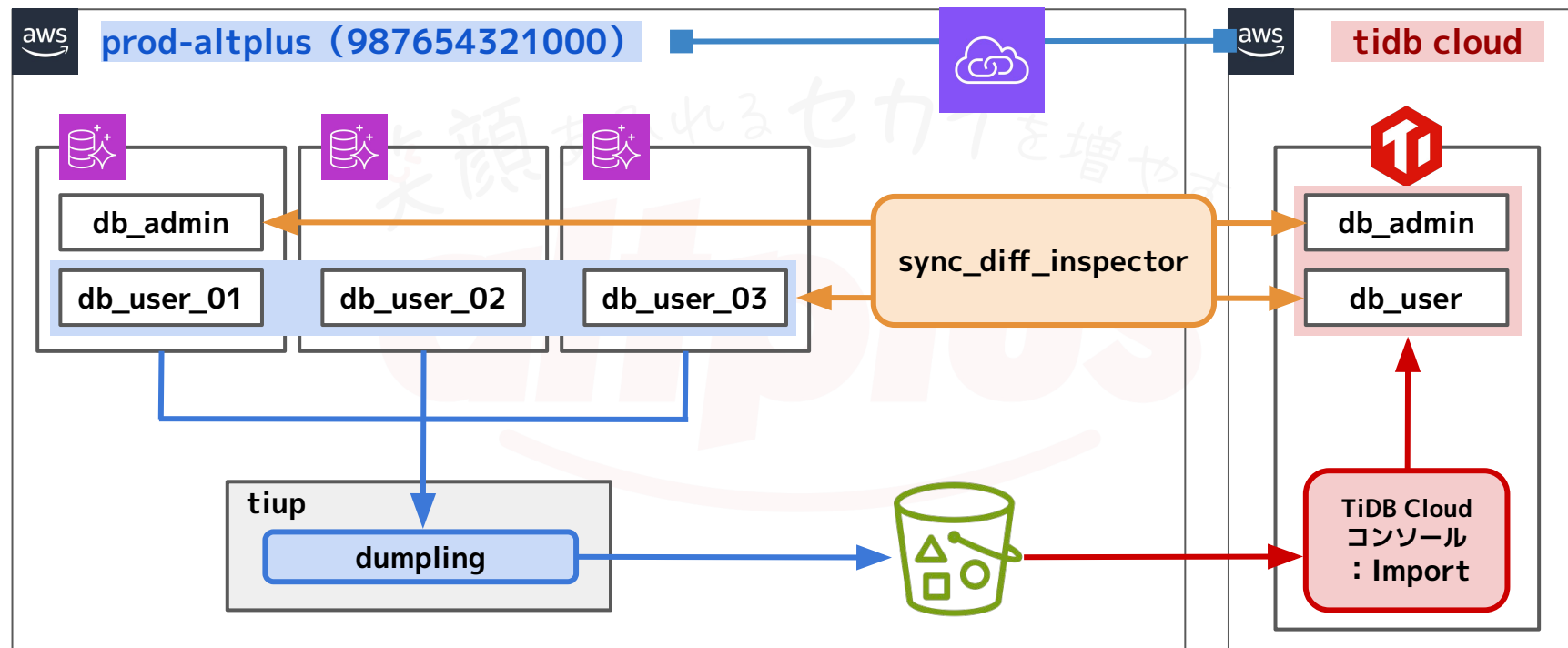
<https://docs.pingcap.com/tidb/stable/migration-tools/>

インポート処理のみTiDB Cloudのマネージド機能を使用

ツール	説明
TiDB Cloud コンソール (インポート)	TiDBクラスターへの初期データインポートで使用 Dumpingでエクスポートしたファイルを、 TiDB Cloudのインポート機能を使用してClusterに取り込む Physicalモードで、TiKVに直接データ（KVペア）を登録

<https://docs.pingcap.com/tidbcloud/import-sample-data/>

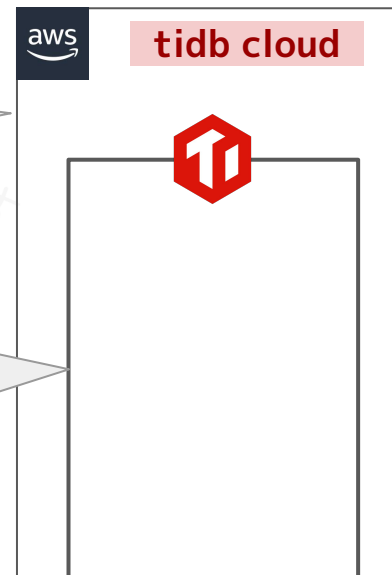
移行のイメージ（全体像）



[準備]TiDBクラスター作成

TiDB CloudのOrganizationを作成

TiDB Clusterを作成



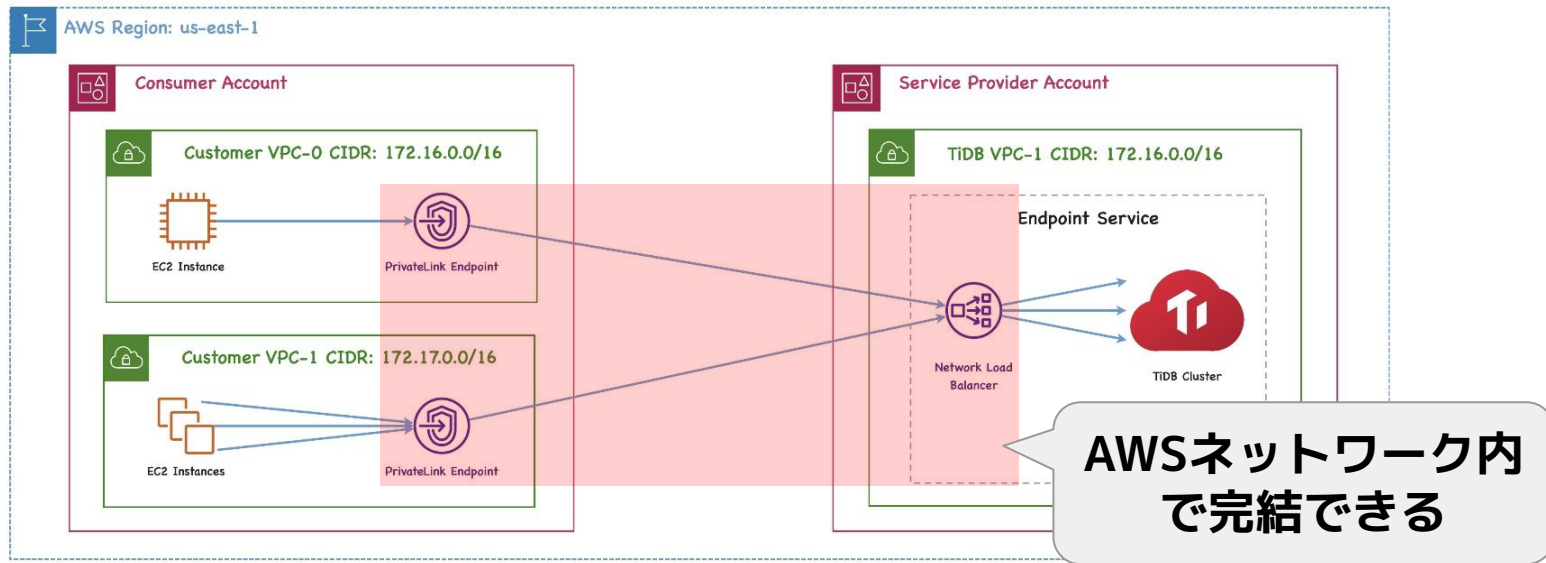
Organization作成後に「TiDB Cluster」の作成が可能になります

[準備]プライベートエンドポイント接続設定



TiDB Cloudでは、PrivateLinkを使用したプライベートエンドポイントは必須ではありませんが推奨されています

通信がパブリックインターネットを経由しない



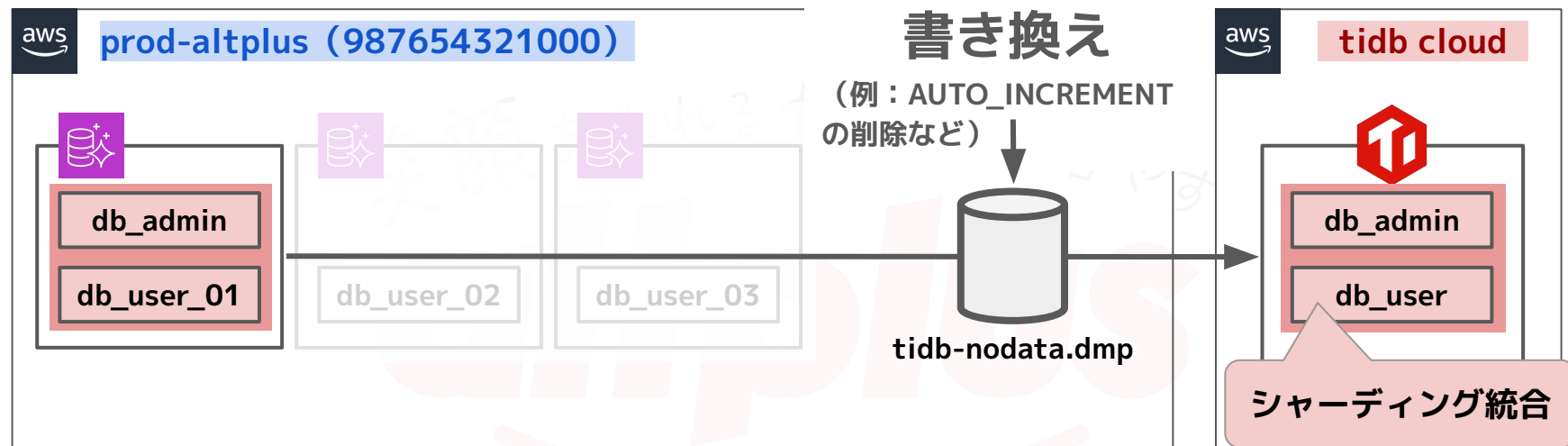
パブリックエンドポイントは簡易ですが、
セキュリティ要件が高い環境ではPrivateLinkの利用が望ましいです！！

VPC Peeringも似た接続方法なのですが . . .



VPC PeeringはTiDB Cloud Starter非対応
のためPrivateLink一択になりました！！

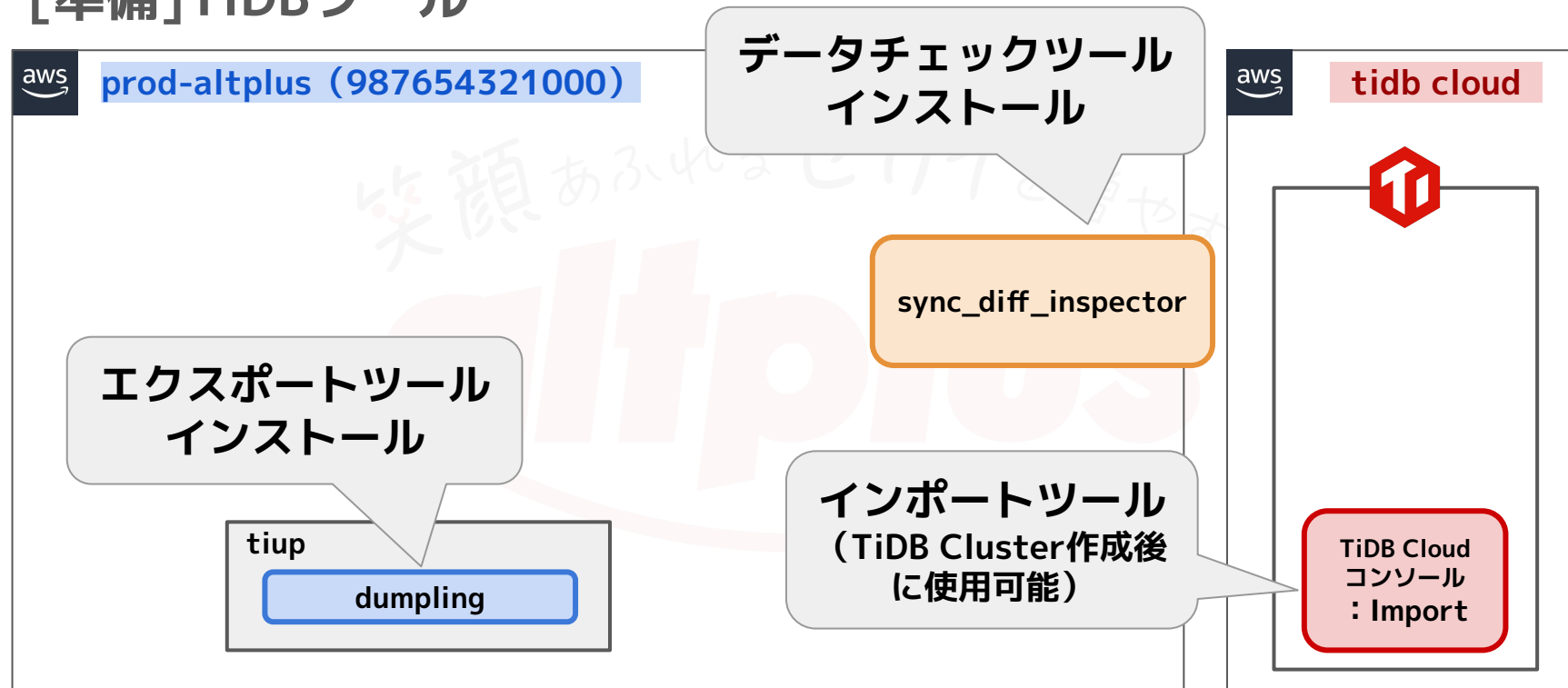
[準備]TiDBテーブル定義変更対応



シャーディングされたテーブル定義は「01」のみを取得し、TiDB向けにdmpファイルを書き換えてリストアを実施する

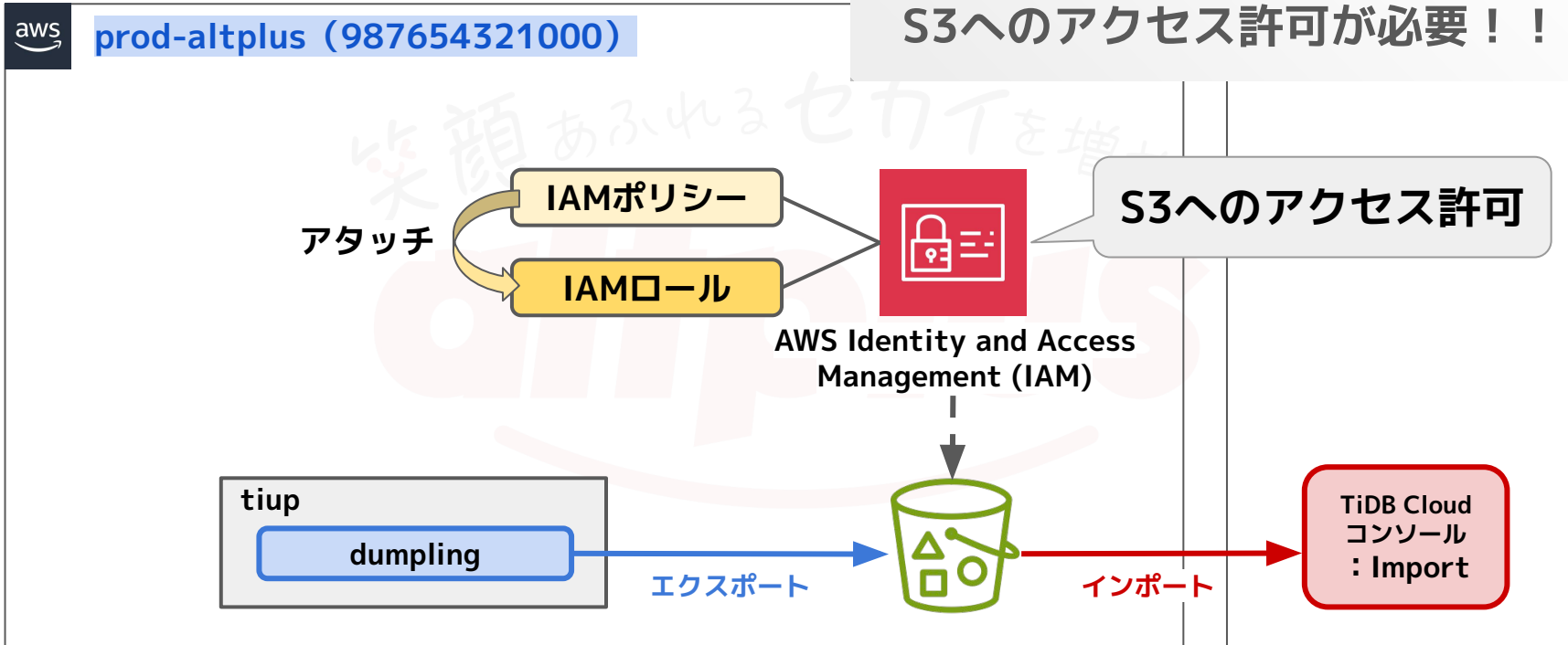
TiDB Cloudへの移行：①移行手順と工夫

[準備]TiDBツール

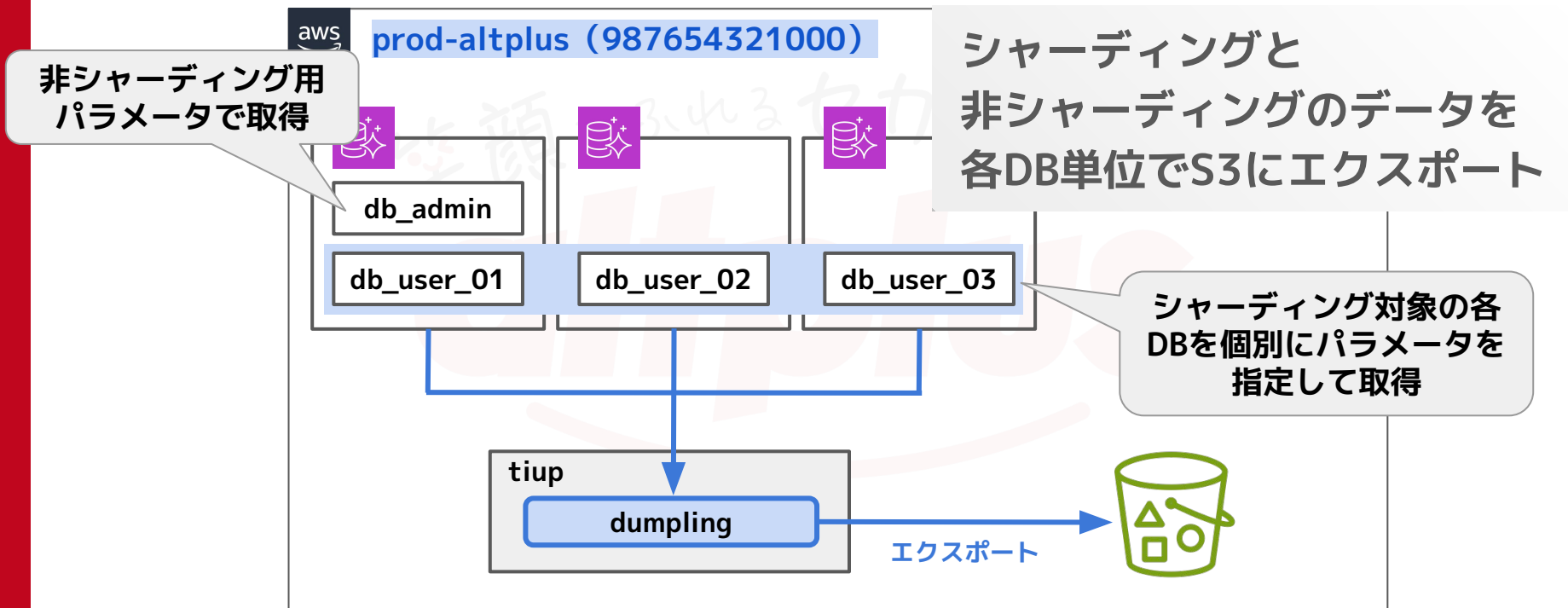


[準備]S3バケット

エクスポート・インポート時に
S3へのアクセス許可が必要！！



[実行]Dumplingを使用したエクスポート

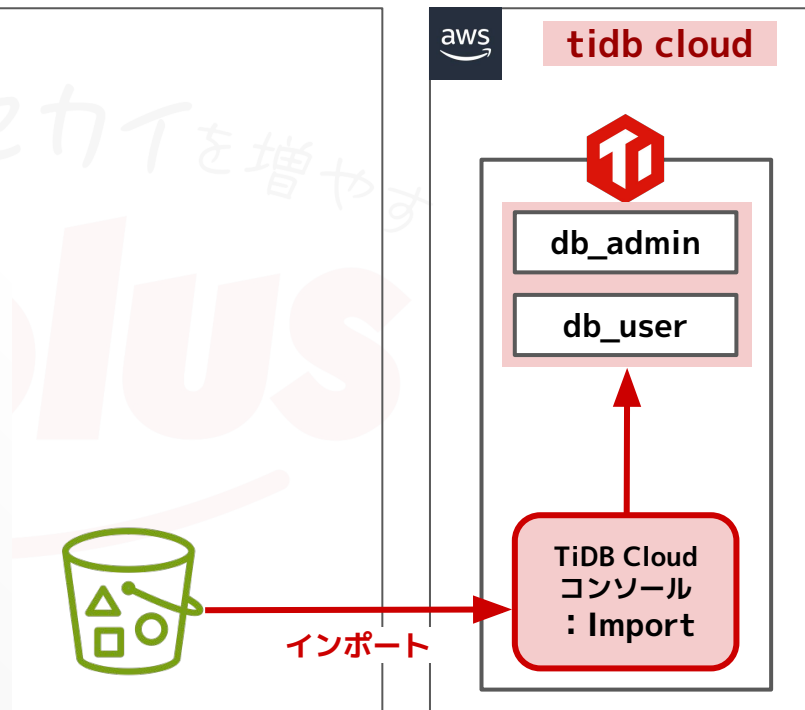


[実行]TiDB Cloudコンソールを使用したインポート

aws prod-altplus (987654321000)

TiDB Cloudコンソールを使用して
直感的なインポートを実行できます

インポート実施前に
「S3URL」や「ロールのARN」を
控えておく必要があります



[補足]TiDB Cloudでのインポート：TiDB Cloudコンソール

オルトプラスの例

Overview

Data

Import

Data Migration

Changefeed

Backup

Monitoring

Diagnosis

Metrics

Nodes

Alerts

Events

Calibrate Resource

Security

Networking

Import

Import data from Cloud Storage
Support CSV, Parquet, and SQL files

Import Data from Amazon S3

Source

Included Schema Files
☐ Yes ☒ No

Data Format
☐ CSV ☐ Parquet ☐ Aurora Snapshot ☒ SQL

Folder URI
s3://tidbcloud-samples/data-ingestion/

Use folder URI instead of file URI. The path should end in '/'.
S3URLを設定

Bucket Access
☒ AWS Role ARN ☐ AWS Access Key

Role ARN
ARN of AWS IAM Role

ARN of AWS IAM Role
Not sure what is Role ARN? [Click here to create new one with AWS CloudFormation](#)

Connect

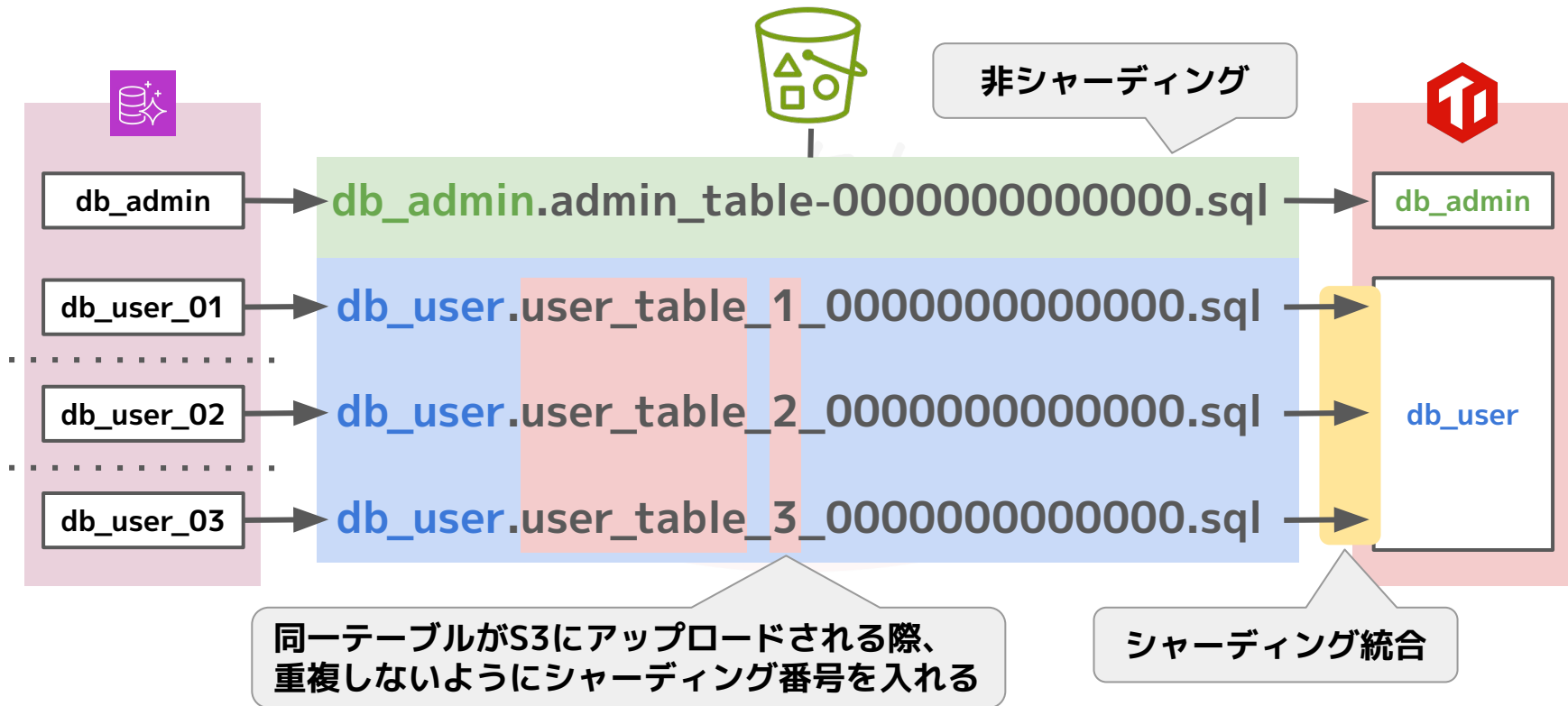
Schemaはあるので「No」

ファイルタイプ：SQL

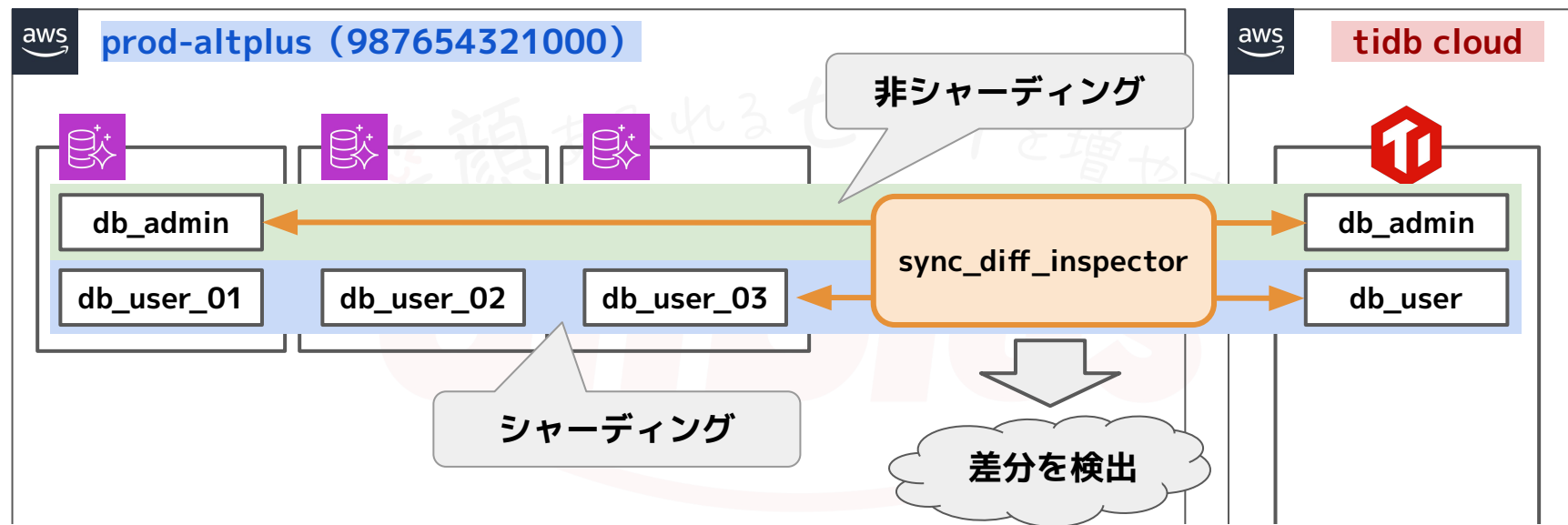
ロールのARNを設定

上記全て入力すると「Connect」が活性化
ボタン押下時にOKであればインポートの画面へ

[補足]エクスポート→インポート（シャーディング統合）イメージ



[実行]sync_diff_inspectorを使用したデータ整合性チェック



インポート前後のデータを比較し、差分を検出
(非シャーディングとシャーディングで分けて実行する必要があります)

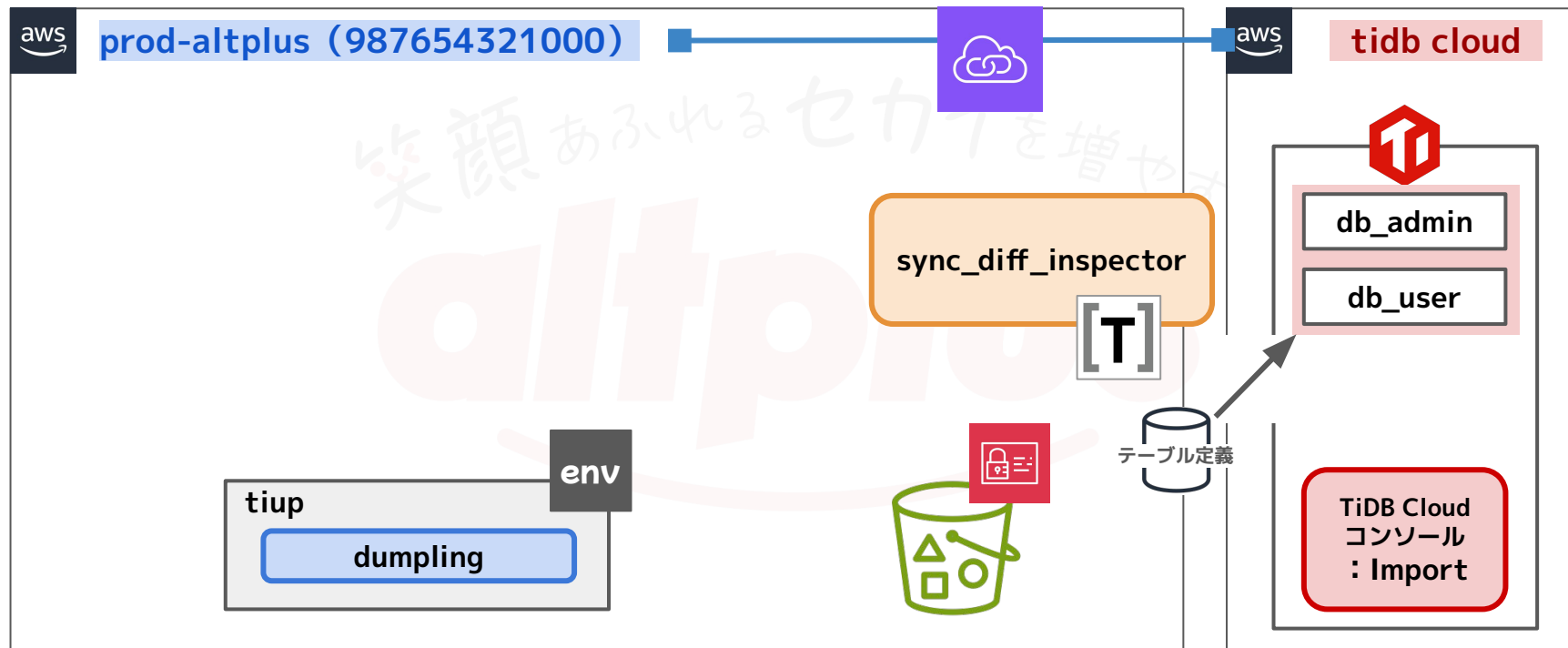
②本番移行

TiDB移行に必要なタスクについて

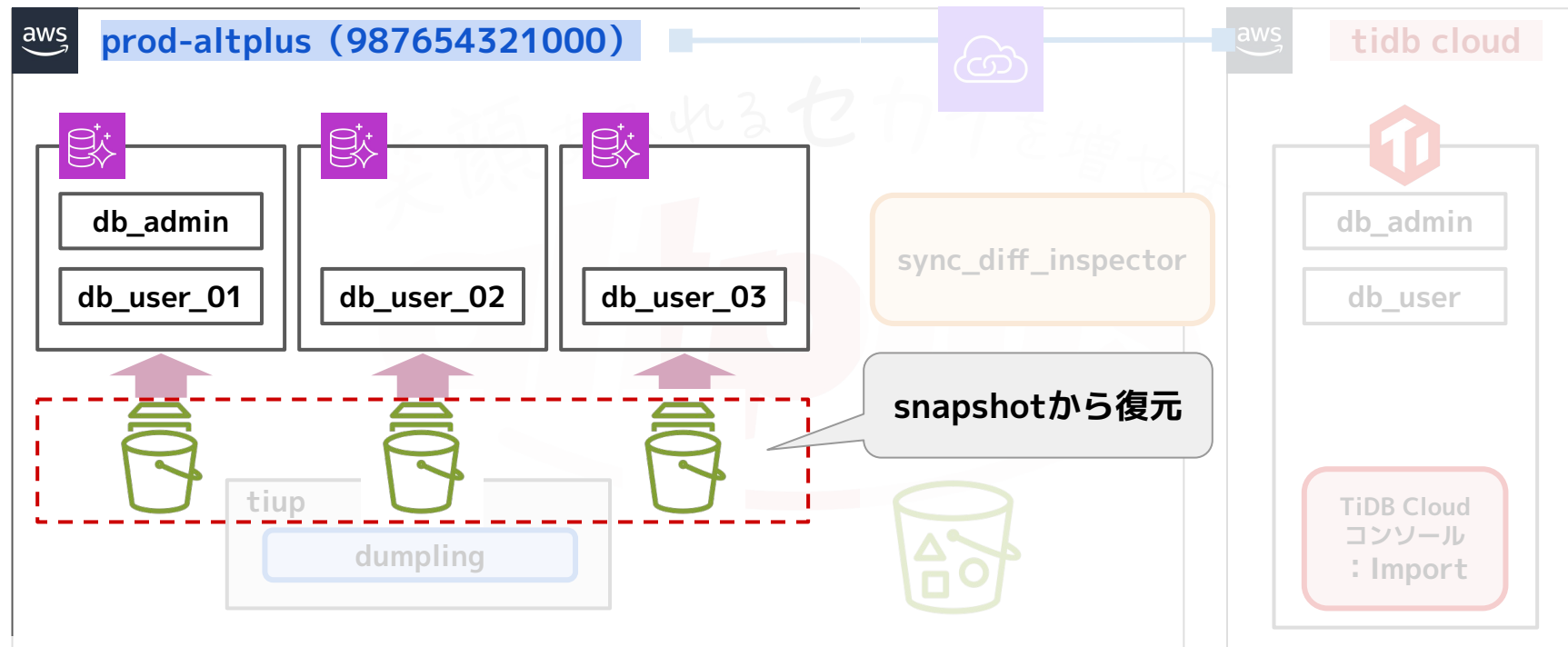
- tidb-manager環境構築
 - dumping
 - パラメータ設定
 - sync_diff_inspector
 - TOML設定
- TiDB Cloud Dedicated作成
 - テーブル作成
 - ユーザー作成
- PrivateLink作成
- TiDB CloudのS3アクセス設定
- mysql_config_editor設定
- データ整合性チェック
 - sync_diff_inspector
- エクスポート実行
 - Dumping
- インポート実行
 - TiDB Cloudコンソール

青：本番移行前に実施可能、赤：本番移行時

本番移行前に実施可能な環境を構築



snapshotから復元してリハーサル環境を構築



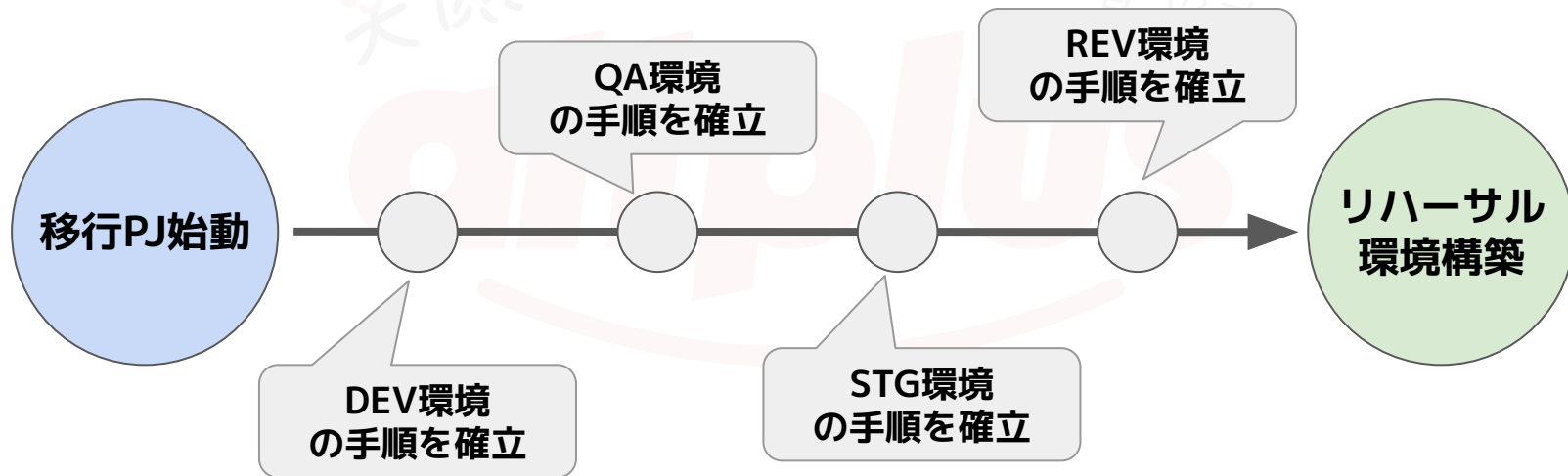
特にsync_diff_inspectorはカラム単位での比較となるため処理に時間がかかる



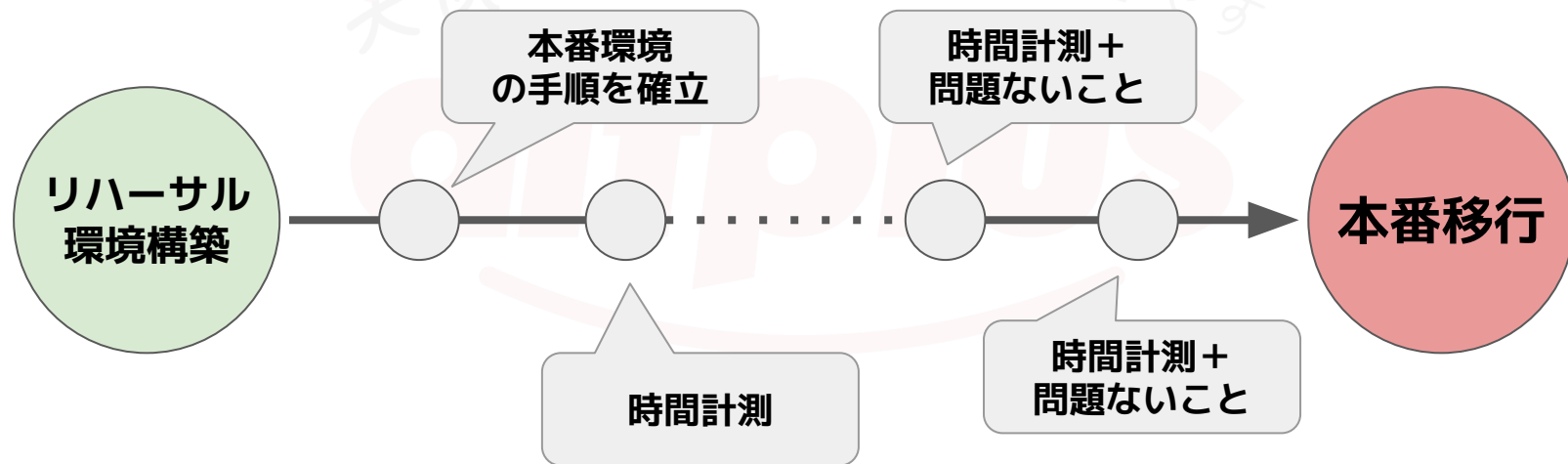
事前実施でメンテナンス時間を短縮

メンテナンス直前の比較で問題なければ本番も問題なしと判断可能

本番を想定したリハーサルを行うためにも、
各環境での手順を先に固めてからリハーサルに臨みました！！

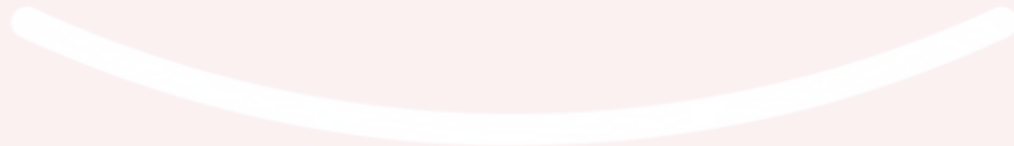


リハーサルで処理時間を計測し、問題が発生しないことを確認
「再現性のあるリハーサル」を実施したおかげで、
本番移行も無事問題なく完了することができました！！



笑顔あふれるセカイを増やす

移行後について



①移行直後のドタバタ劇

移行直後に発生したトラブル — スパイクアクセス

リリース後に負荷高騰でユーザー
からの問い合わせが殺到！！



①移行後のドタバタ劇：移行直後に発生したトラブルー スパイクアクセス

リリース10分後、
プレゼント機能へのアクセスが一気に集中。



結果的にはEBSのIOPS上限に張り付き、
応答が一時的に悪化しました。

①移行後のドタバタ劇：移行直後に発生したトラブルー スパイクアクセス

原因

プレゼント機能のレンジスキャン

一次対応

TiKVのスケールアウトで対応
書き込みのスケールは「NewSQLの強み」

二次対応

必要な件数（200件）のみを対象に
スキャンするよう最適化

リリース直後の負荷もTiKVの
スケールアウトで回避！！

その後はユーザー影響なく、
安定運用継続！！



移行翌日のトラブルー スケール戦略の誤判断

予定外のスケールインの調整も、
ユーザー流入直前で停止！！



少しでもコストカットができればと考えて
前日増やしたTiKVをスケールインしました。

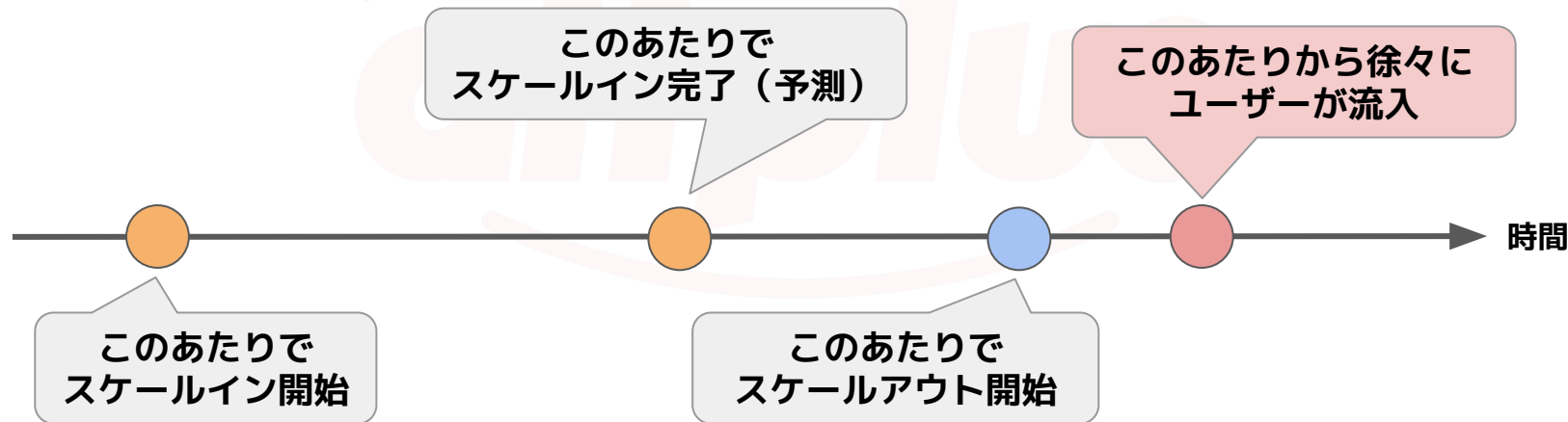


結果的にはユーザー流入前にPingCAPさんに
スケールインを一時停止いただきました💧

①移行後のドタバタ劇：移行翌日のトラブルー スケール戦略の誤判断

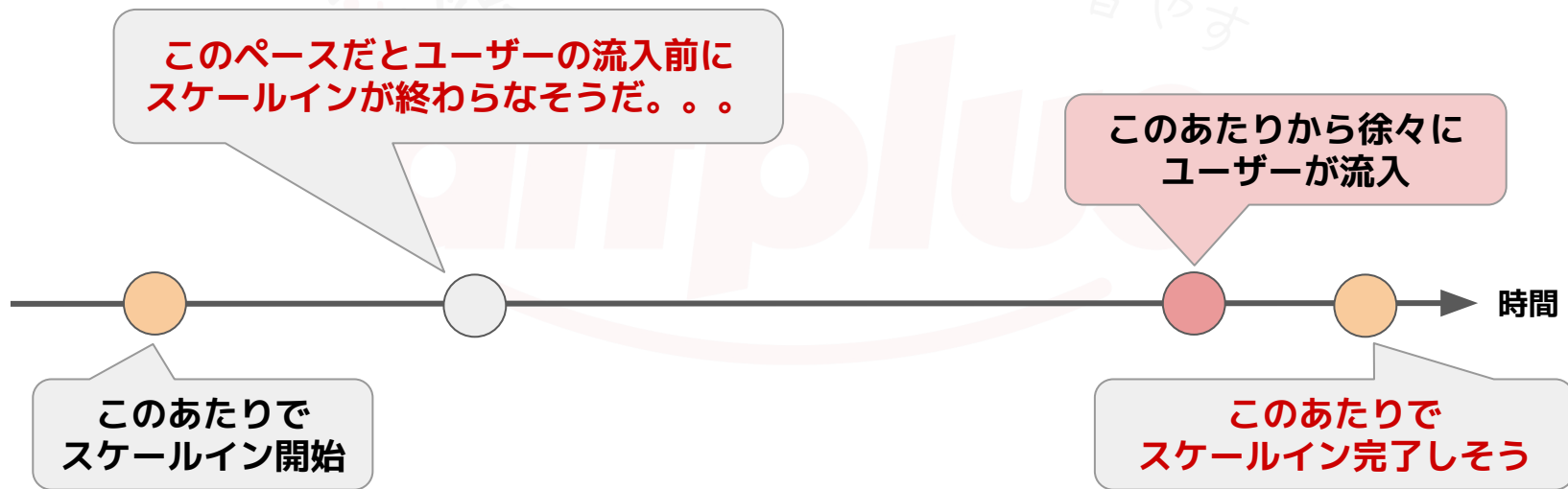
当初の予定

ユーザーの流入前にスケールインが完了するので
流入前のタイミングでスケールアウトしよう！！



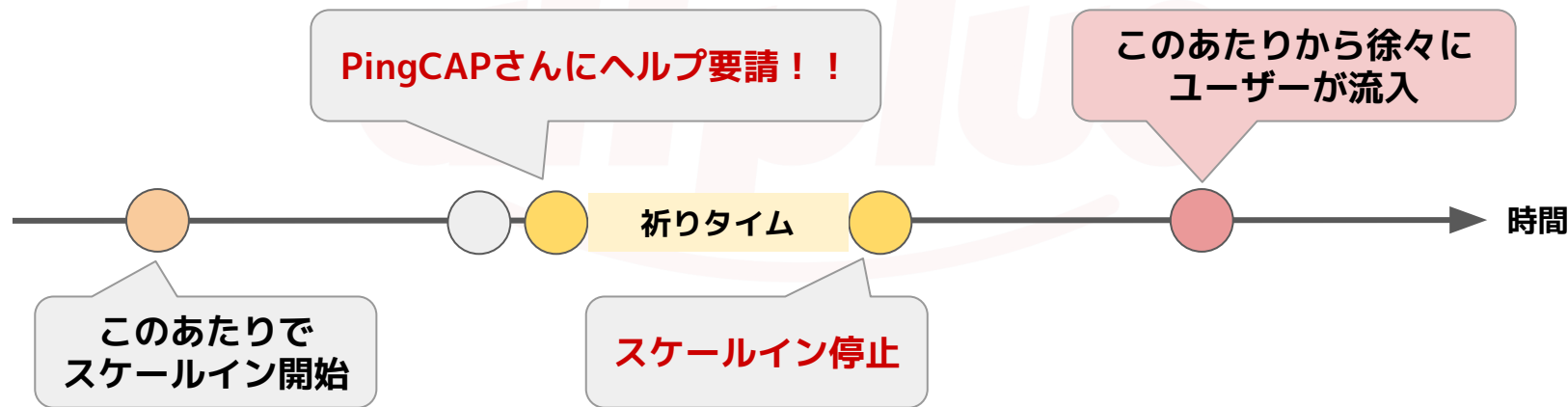
現実①

スケールインの終了が予想と違うかもと焦りが・・・



現実②

PingCAPさんに助けを求めました😭
その結果スケールインが止まりました！！



Regionの移動を途中で停止する



LeaderのRegionが均等に分散されない状態

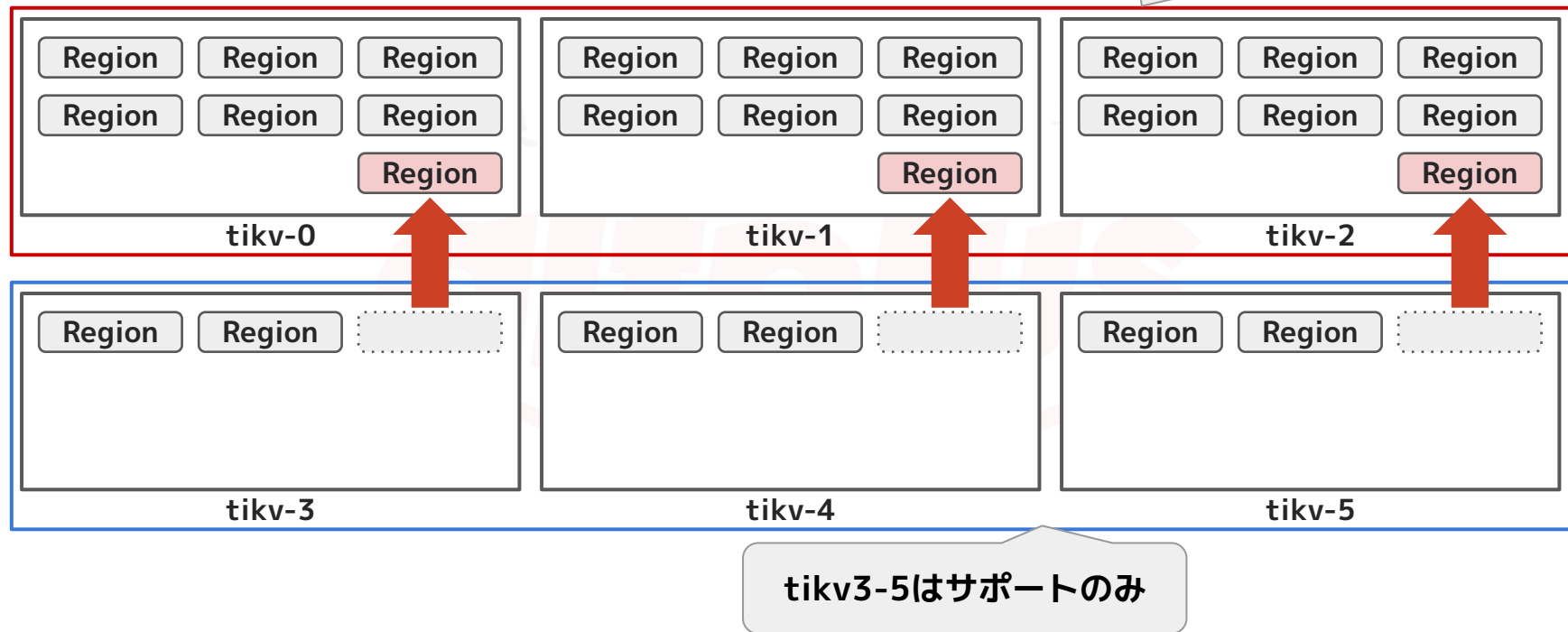


リソース使用に偏りが発生する

①移行後のドタバタ劇：移行翌日のトラブルー スケール戦略の誤判断

移動を停止中

tikv0-2に負荷が集中する



①移行後のドタバタ劇：移行翌日のトラブルー スケール戦略の誤判断

原因

コストカットを意識し過ぎて予定外のスケールインを実施

一次対応

PingCAPさんがスケールイン停止の対応
ユーザー影響なしで回避

二次対応

ピークタイムが過ぎた深夜0時過ぎに
スケールインを再開（PingCAPさん）

PingCAPさんのサポートのおかげで
ユーザー影響なしで回避！！

サポートのありがたみを
改めて実感しました！！



②TiDB Cloud APIでの自動化

**TiDB Cloud API（β版）を使用した
自動化を行いました！！**

**TiDB Cloud Dedicatedの操作をコンソールを
介さず実行できるためとても便利です！！**

TiDB Cloud APIの特徴

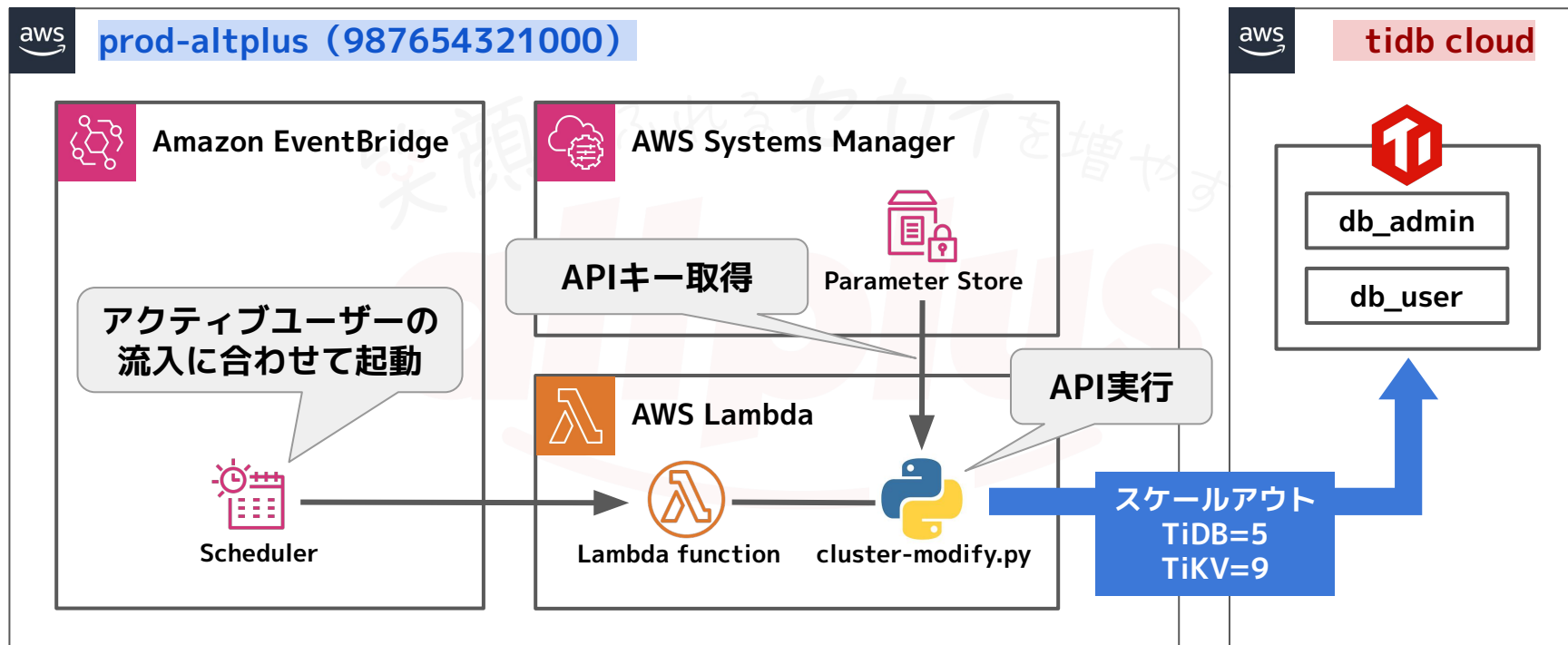
- REST形式で提供されるAPI
 - コンソールの主要操作をコードで実行可能
- 主要な操作
 - クラスターの作成・削除
 - スケーリング
 - バックアップおよびリストア
 - インポート（Physical）

準備するもの

- API Key
 - TiDB Cloudコンソールから作成
 - キーの管理はParameter Storeなど
- スクリプトの準備
 - ドキュメントを確認して実装する
 - パラメータはJSON

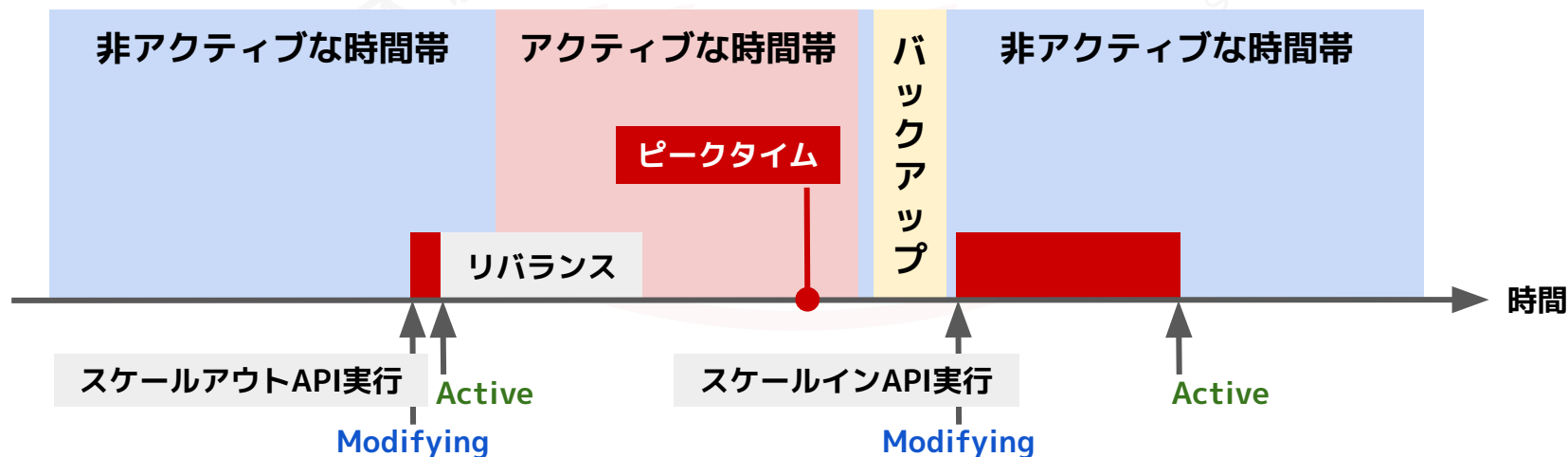
<https://docs.pingcap.com/ja/tidbcloud/api-overview/>

スケールアウトの例



オルトプラスでの運用例（毎日実行：TiKV）

- スケールアウトはアクティブな時間帯の直前で実行
- スケールインはバックアップ後に実行（念の為）



③自動化がメンテナンスと干渉しないために

TiDB Cloudはメンテナンスウィンドウがあり、自動化の設定によっては「意図しない接続断」につながるケースがあります。

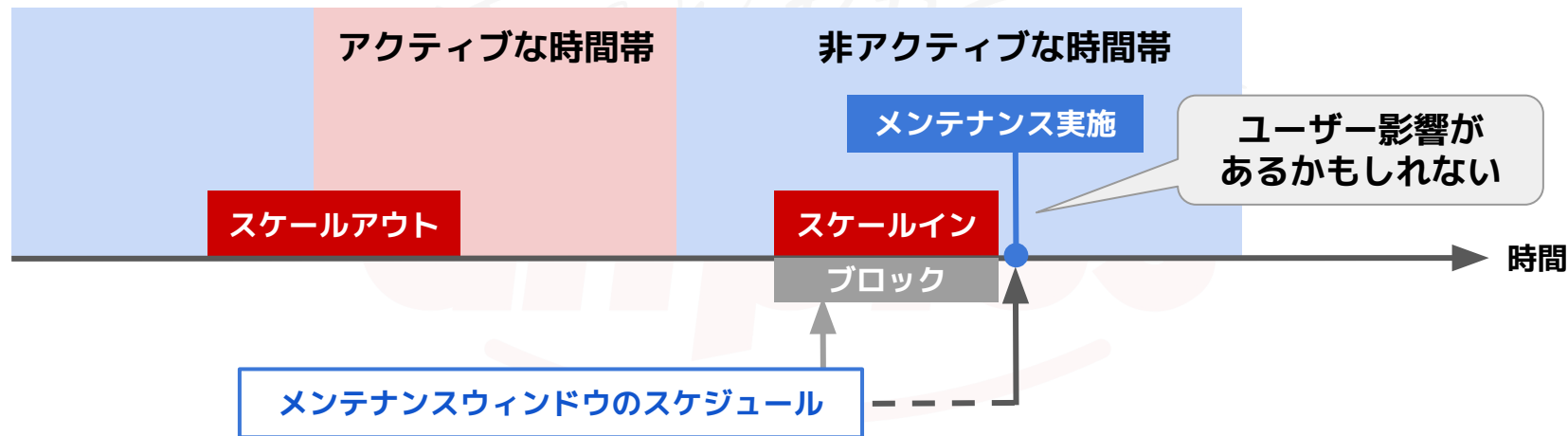
そのため、APIなどで自動化する場合はメンテナンスとの干渉を避ける工夫が必要です。

メンテナンス期間中に許可されない操作

- クラスター操作（変更/一時停止/再開）
- セキュリティ設定の変更
- PrivateLink/VPC Peeringの作成
- インポート/移行ジョブ/Changefeedの作成
- 移行ジョブ/Changefeedのスケール変更

<https://docs.pingcap.com/ja/tidbcloud/configure-maintenance-window/>

例えば、スケールイン時にメンテナンス実施すると、
スケールイン後にメンテナンスが実施される



影響は最小限でも一時的な接続中断は発生する可能性がある

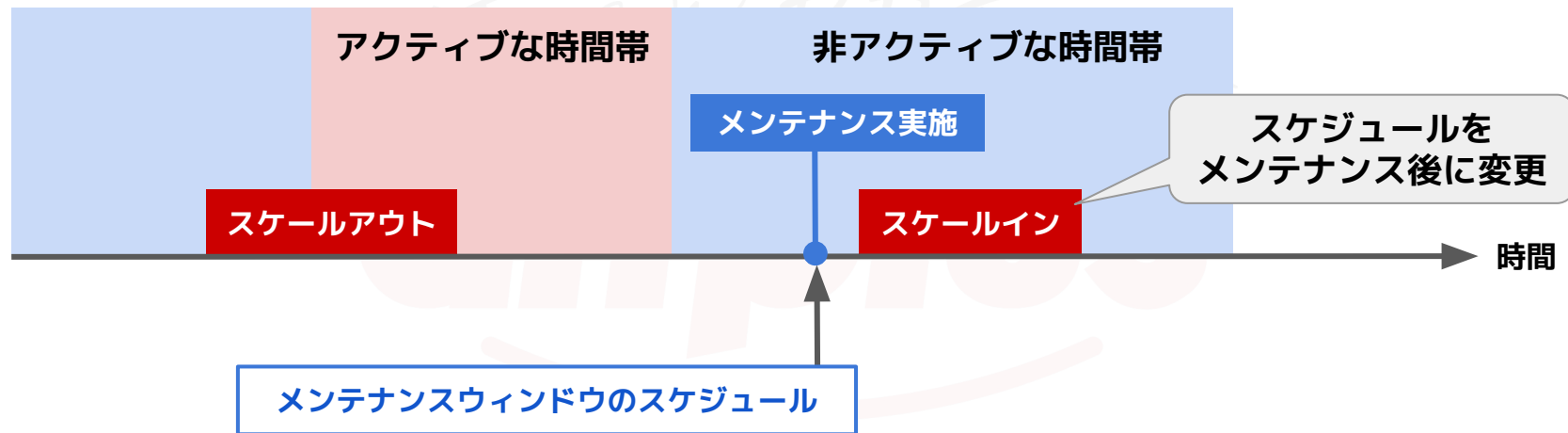
メンテナンスウィンドウの通知をメールで確認する

- メンテナンス期間の開始2週間前
 - 緊急メンテナンスタスクを除く
- メンテナンスウィンドウが開始する72時間前
- メンテナンスウィンドウが開始される時間
- メンテナンスウィンドウが完了した時間

メールだけでは気づきにくいためSlackやPagerDutyでの可視化を推奨

移行後について：③自動化がメンテナンスと干渉しないために

メンテナンス実施後にスケールインを実施することで、
ユーザー影響がない運用が可能となる



「メンテナンス後に実行する」工夫で安定運用につながります！！

- NewSQLの観点からTiDB Cloud刷新の意義を整理
- 移行の手順と移行に使用したツールを紹介
- 移行後のトラブル対応と運用改善について共有



TiUG TiDB
User Group

TiDBユーザーがつながるコミュニティ

What is TiUG?

TiDB に興味のある方々と情報共有や交流を行い **Ti !!**
みんなで **TiDB** を盛り上げ **Ti !!**



01

2ヶ月に1回を
目安に開催

02

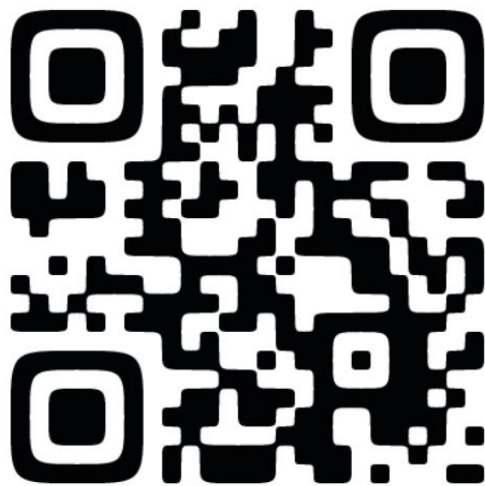
TiDB の
技術や活用情報、
ユースケースなどを
ユーザー同士で
共有・交換

03

TiDB に興味がある方
なら誰でも大歓迎！

Please Join!!

Connpass



<https://tiug.connpass.com>

笑顔あふれるセカイを増やす

**もっとTiDBを活用して、
NewSQLの力で楽しもうー！！**



ご清聴ありがとうございました