Amazon RDS+EC2+PGroonga+ロジカルレプリケーションを使った低コスト高速全文検索

堀本泰弘

クリアコード

第28回 中国地方DB勉強会 in 岡山

2020-01-25

自己紹介

堀本　泰弘

全文検索エンジン

の開発・サポート

自己紹介

- 他のOSSへもコントリビュート
  - PostgreSQLのドキュメントの改善等
  - 問題は**「開発元」**で修正が弊社のポリシー

今日のテーマ

なるべく楽に
高機能で高速な全文検索
がしたい

実現のための要素

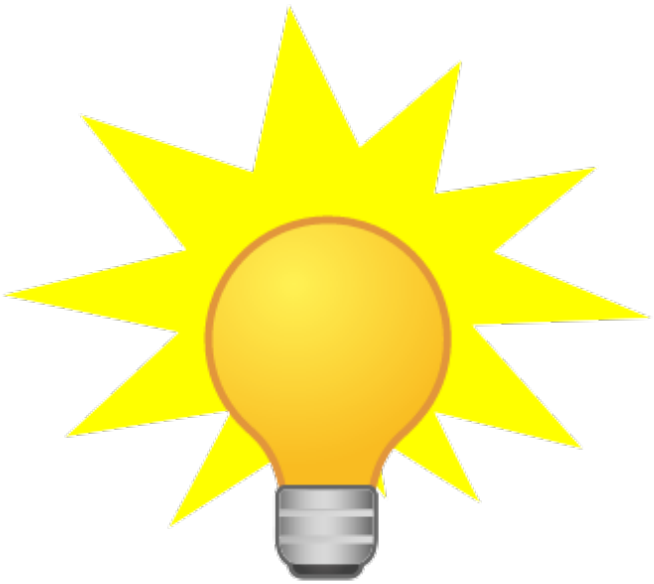
特に大事な要素

楽

楽

なぜ楽なのか？

設定が楽



Optimization

設定は最適化ずみ

構築が楽

A few minutes...

設定は最適化されているので

構築は数分!

スケールアップが楽

MAX

CPU:32

Memory:244GiB

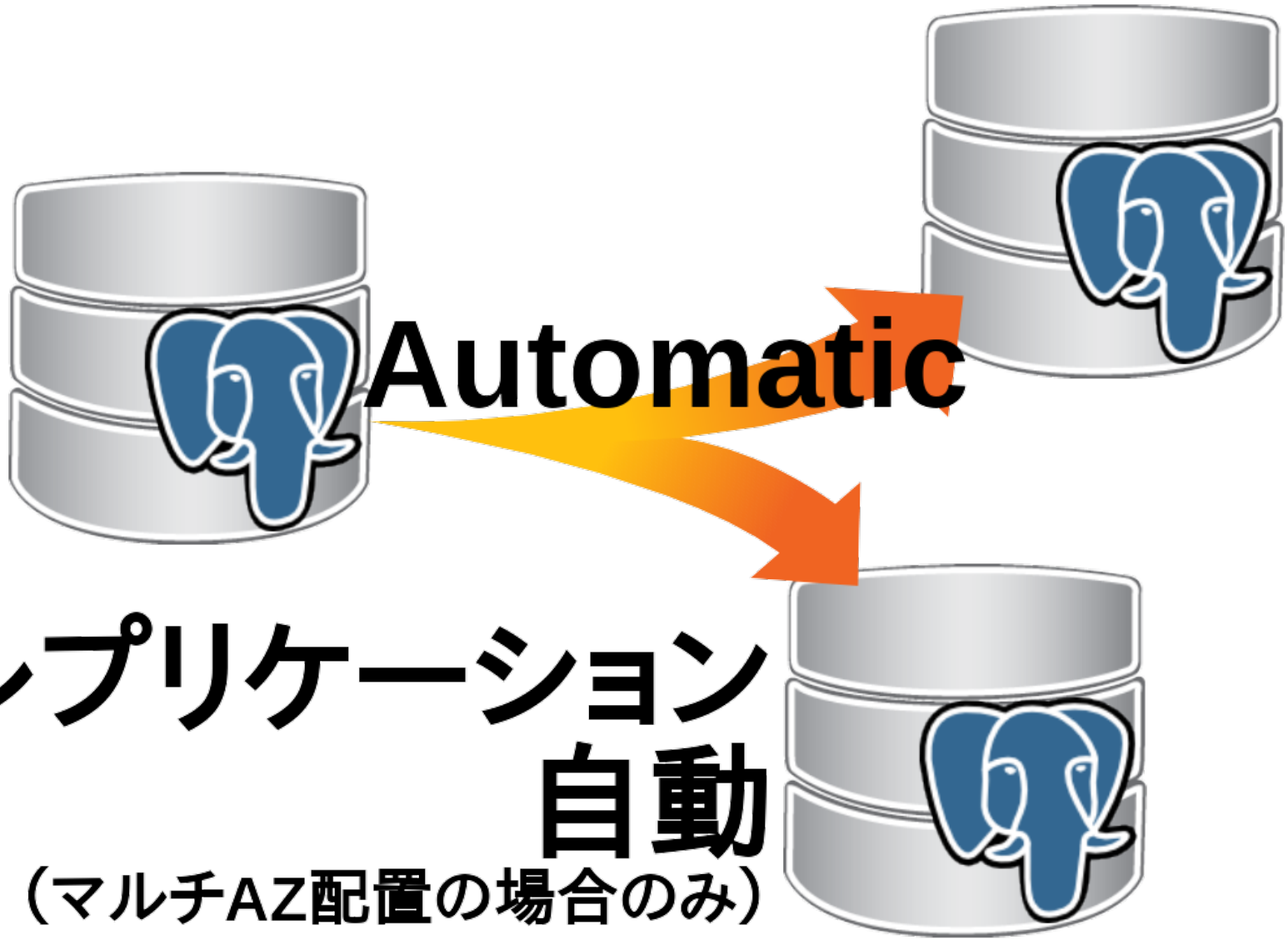
Storage:32TB

スケールアップが楽

スケーリングは
数分

Storageは
ダウンタイムなし

障害対策が楽

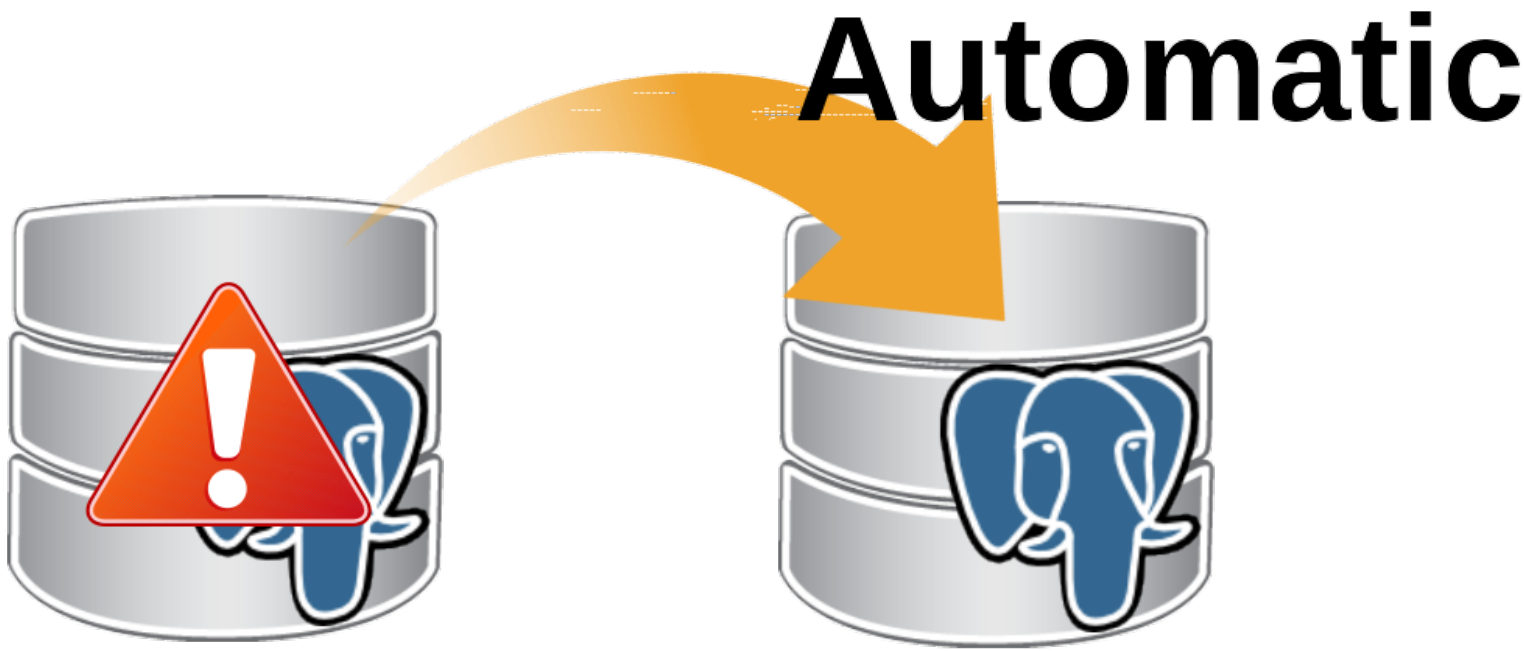


障害対策が楽

Automatic

バックアップ
自動

障害対策が楽



故障ハードの交換
自動

今日のテーマ

なるべく楽に
高機能で高速な全文検索
がしたい

全文検索

全文検索

なぜPGroongaなのか？

対応言語

全言語対応

速い

高速

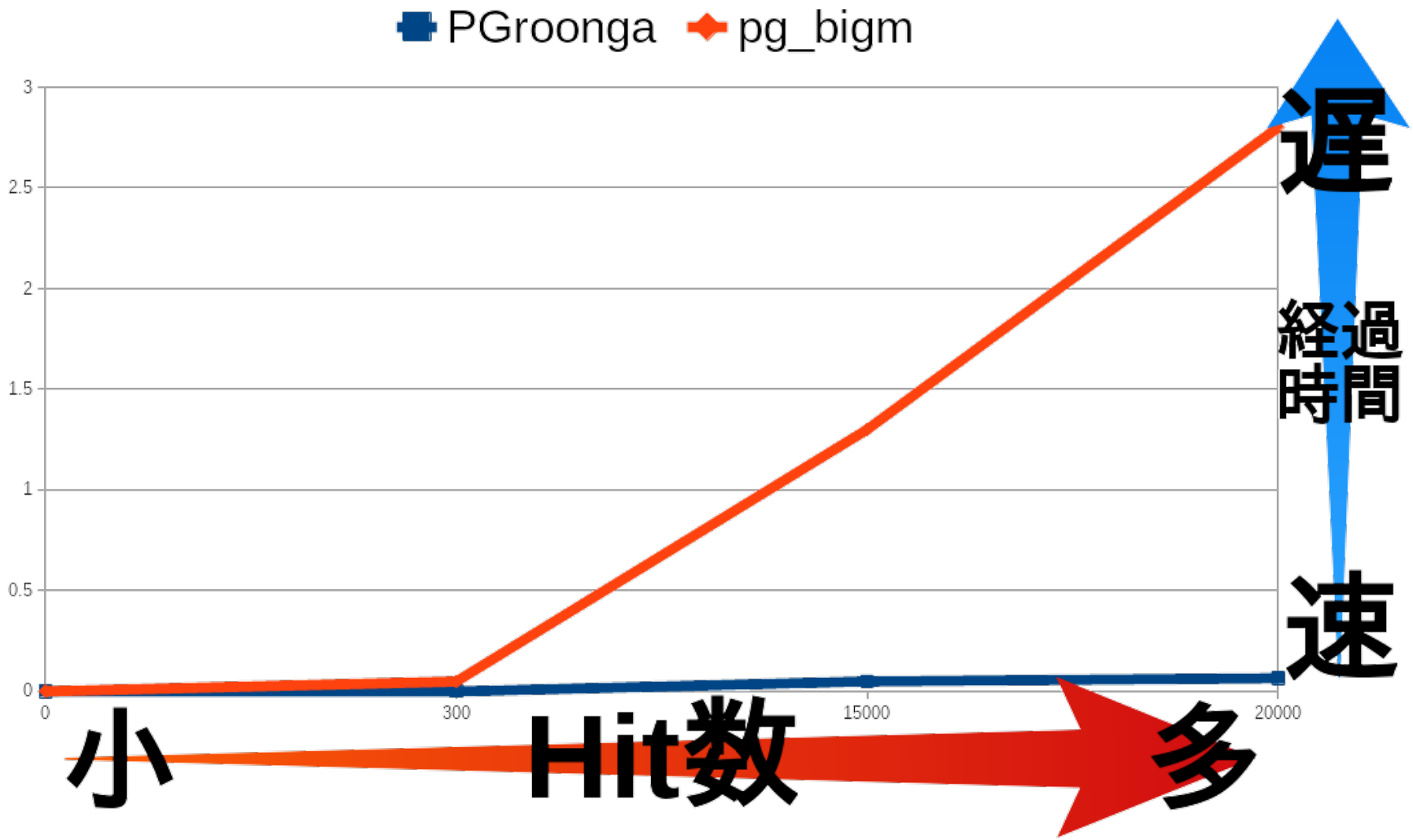
速い

ベンチマーク

速い

- 日本語Wikipedia
  - レコード数：約90万レコード
  - 平均テキストサイズ：6.7KiB

速い


PGroonga
pg_bigm
3
2.5
2
1.5
1
0.5
0
0
300
15000
20000
遅
経過
時間
速
小
Hit数
多

速い

- ベンチマークの詳細な条件は以下を参照

https://pgroonga.github.io/ja/reference/pgroonga-versus-pg-bigm.html

SQLが使える

専用のクエリー
を覚えなくてOK!

実例

具体的に
どう書くのか？

実行例：テーブル定義

```
CREATE TABLE entries (
  title text,
  content text
);
```

実行例：インデックス定義

```
-- 全文検索用インデックス
CREATE INDEX entries_full_text_search
  ON entries
  --「USING pgroonga」=「PGroongaを使う」
  USING pgroonga (title, content);
```

実行例：データ挿入

```
-- 普通に挿入するだけでよい
INSERT INTO entries
  VALUES ('PGroongaで高速全文検索！',
          '高速に全文検索したいですね！');
```

実行例：全文検索

```
SELECT title FROM entries
  WHERE
-- &@~で全文検索
-- 「検索」と「高速」をAND検索
    title &@~ '検索 高速' OR
    content &@~ '検索 高速';
```

実行例：LIKE

```
SELECT title FROM entries
   WHERE
-- LIKEでもインデックスが効く
--＝アプリを書き換えずに高速化可能
-- ただし&@~より性能が落ちる
      title LIKE '%検索%' OR
      content LIKE '%検索%';
```

機能

- 全文検索に必要そうな機能は一通り揃っている
  - 同義語検索
  - 類似文書検索
  - 読みがな検索
  - 入力補完 etc..

読みがな検索

「やきにく」
ってどう書きます
か？

読みがな検索

- やきにく
- 焼き肉
- 焼肉
- やき肉
- ヤキニク

読みがな検索

当然ですがどれも「やきにく」と読みます

読みがな検索

- 読みが同じなので、以下は全部同じものとして扱えます
  - やきにく
  - 焼き肉
  - 焼肉
  - やき肉
  - ヤキニク

読みがな検索

例えば
「やきにく」で検索すると

読みがな検索

- 「やきにく」**Hit!**
- 「焼き肉」**Hit!**
- 「焼肉」**Hit!**
- 「やき肉」**Hit!**
- 「ヤキニク」**Hit!**

読みがな検索

異体字

読みがな検索

「広」と「廣」

読みがな検索

例えば人名の
検索

読みがな検索

検索キーワード「広瀬」で

- 「広瀬」**Hit**
- 「廣瀬」**Hit**

となってほしい

読みがな検索

通常の検索
検索キーワード「広瀬」で

- 「広瀬」のみ**Hit**

読みがな検索

読みがな検索なら
検索キーワード「広瀬」で

- 「広瀬」**Hit**
- 「廣瀬」**Hit**

読みがな検索

両方ヒット！

読みがな検索

「広瀬」も
「廣瀬」も
読みが同じ

他にも

- それっぽい順でソート
- キーワードハイライト
- キーワードの周辺テキスト表示

他にも

- 電話番号検索
  - 090-1234-5678 と 090 1234 5678、(090)1234-5678 等
- fuzzy検索
  - typo対策
    - テクノロジーとテノクロジー

他にも

継続的に
メンテナンス
されている

他にも

PostgreSQL12にも対応！

高機能で高速

PGroongaなら高機能で高速に全文検索できる

Amazon RDS + PGroonga

しかし。。。

Amazon RDS + PGroonga

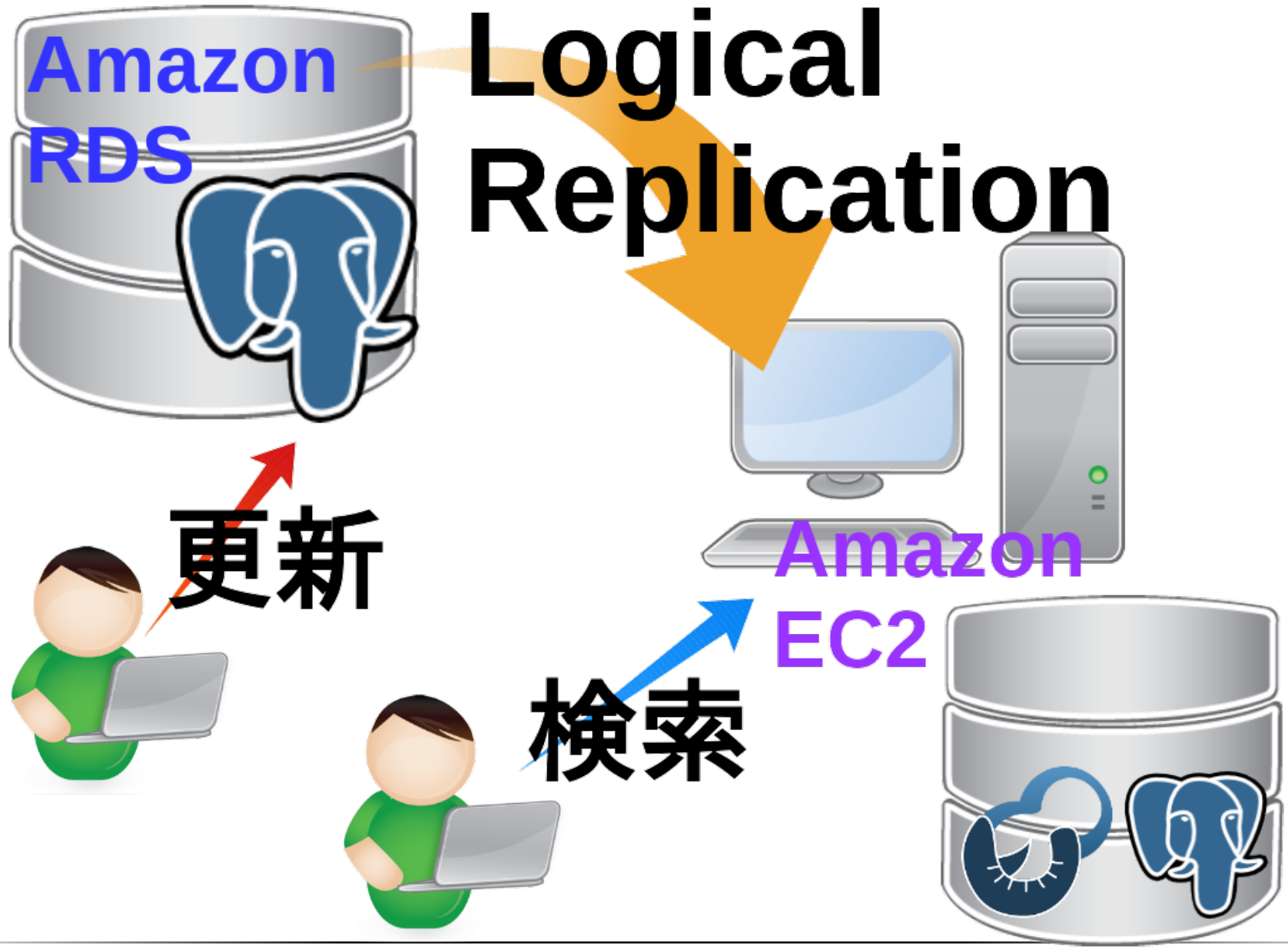
インストール不可

Amazon RDS + PGroonga

他の２つの要素の出番！

Amazon RDS + PGroonga

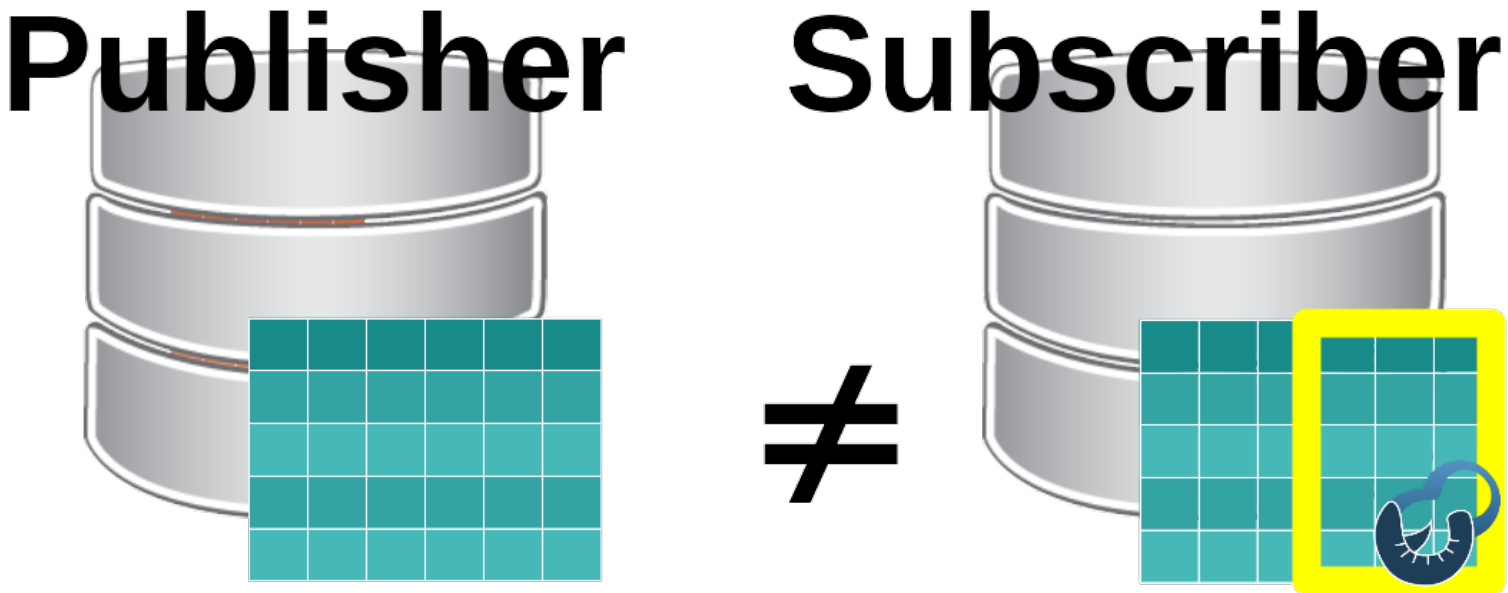
構成



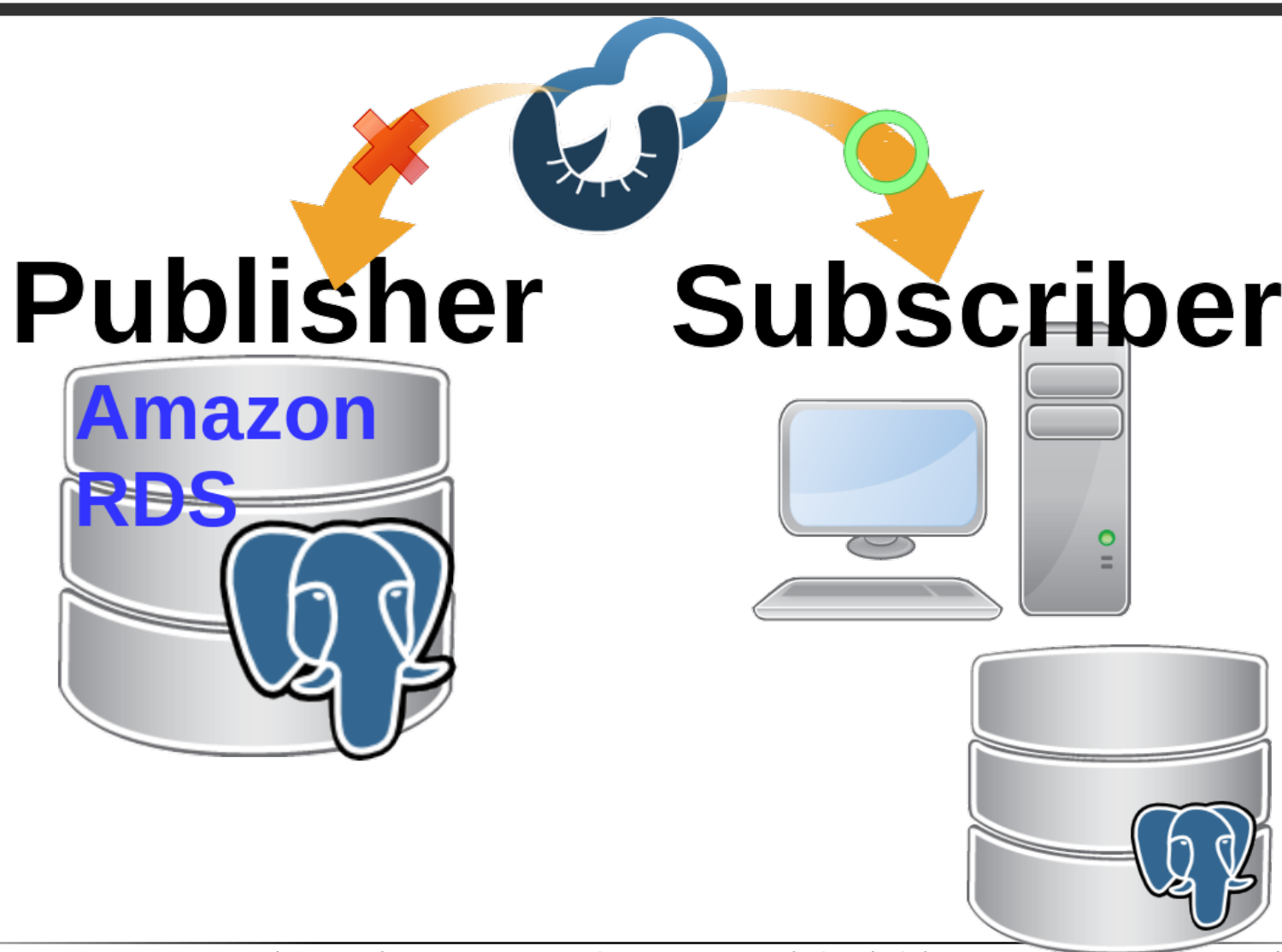
ロジカル
レプリケーションの特徴

複製元と複製先の構造が同一でなくてもよい

ロジカル
レプリケーションの特徴



ロジカル
レプリケーションの特徴



検索

Publisher

Logical Replication

Amazon RDS

Subscriber

Read Only!

更新

Publisher

Amazon RDS

Write Only!

Logical Replication

Subscriber

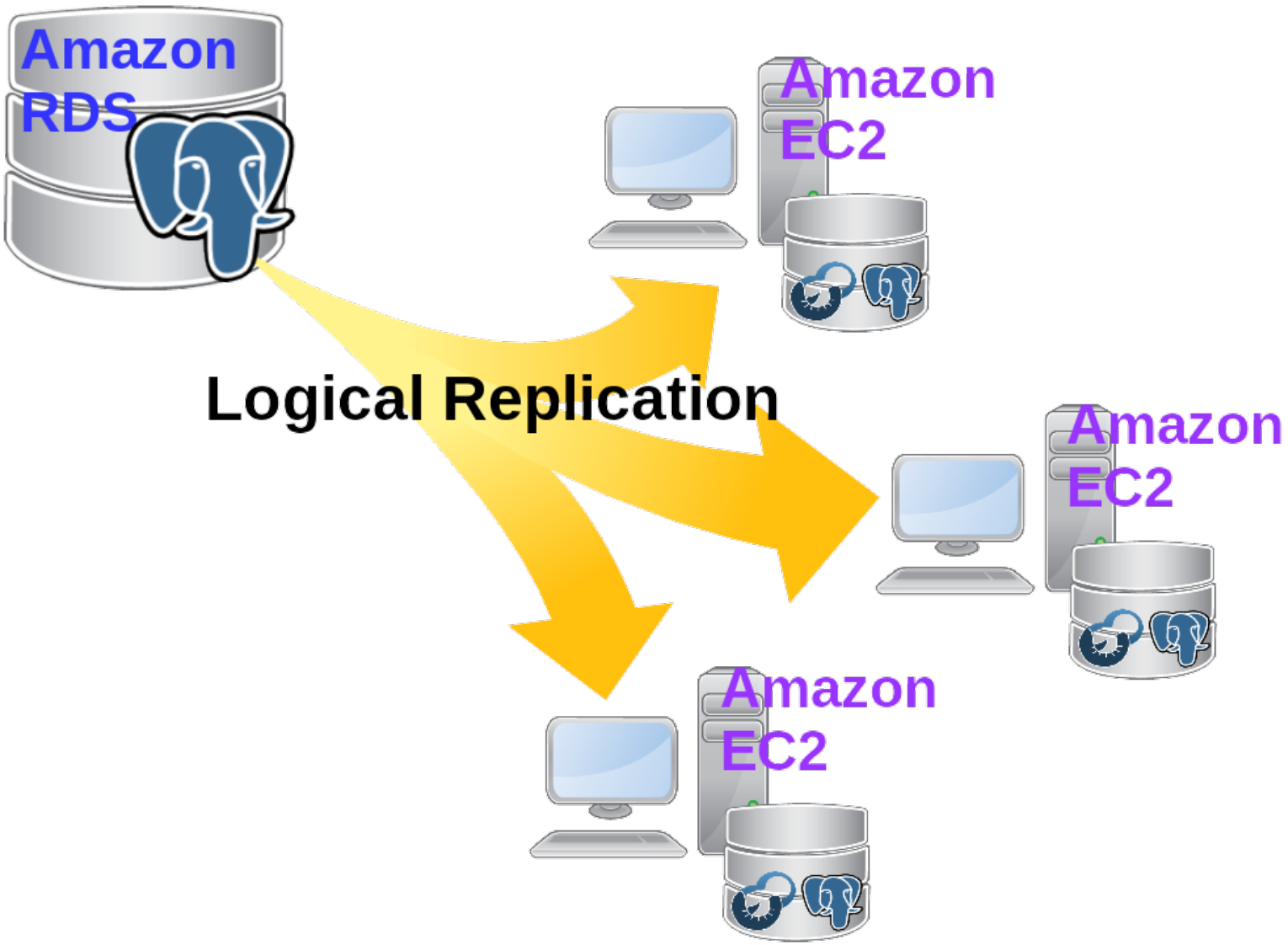
問題点

Subscriberが
1台しかない

問題点

- リクエストが増加しつづけた場合、この構成では耐えられない

負荷分散


Amazon
RDS
Amazon
EC2
Logical Replication
Amazon
EC2
Amazon
EC2

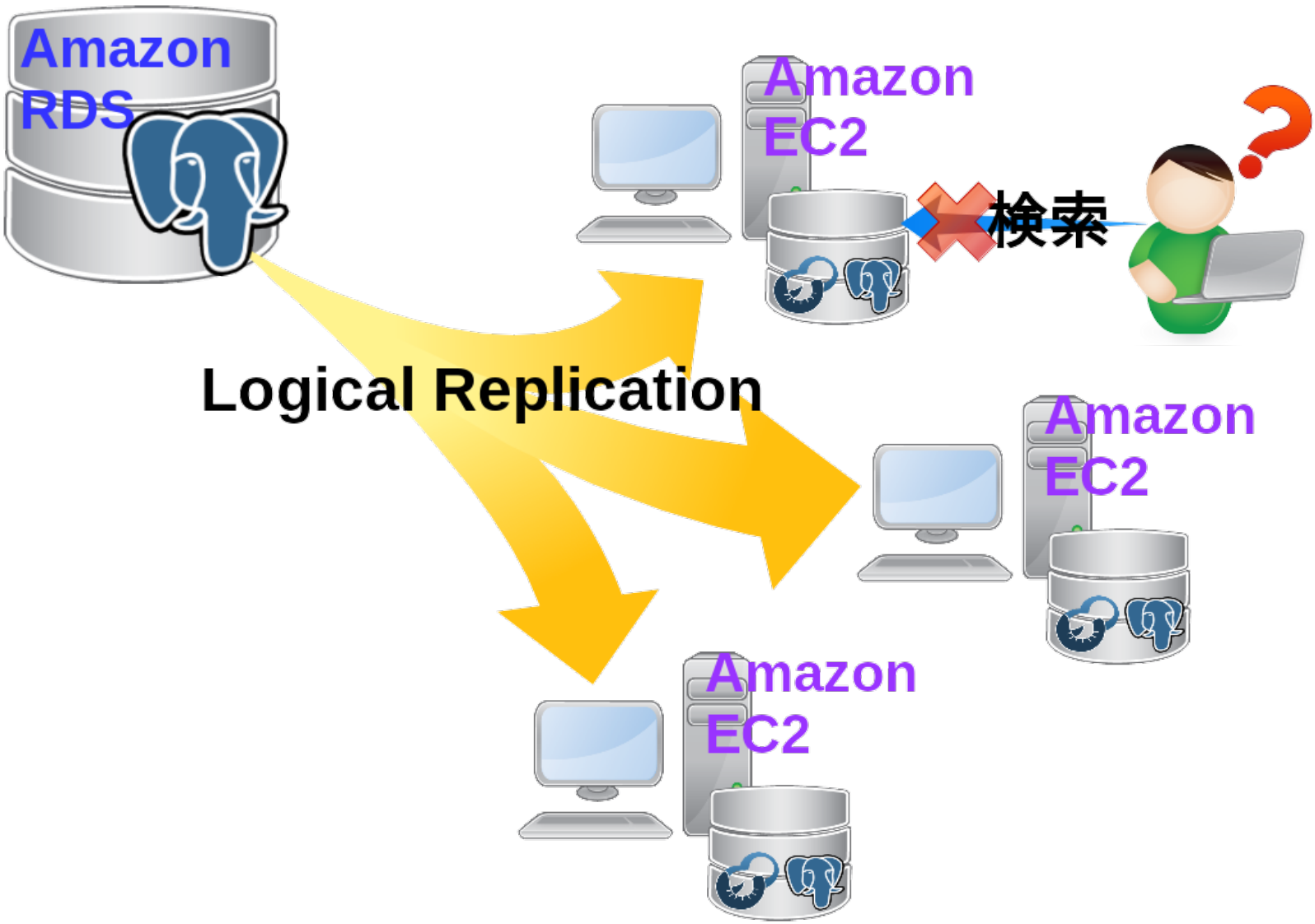
復旧

検索できなくなったら

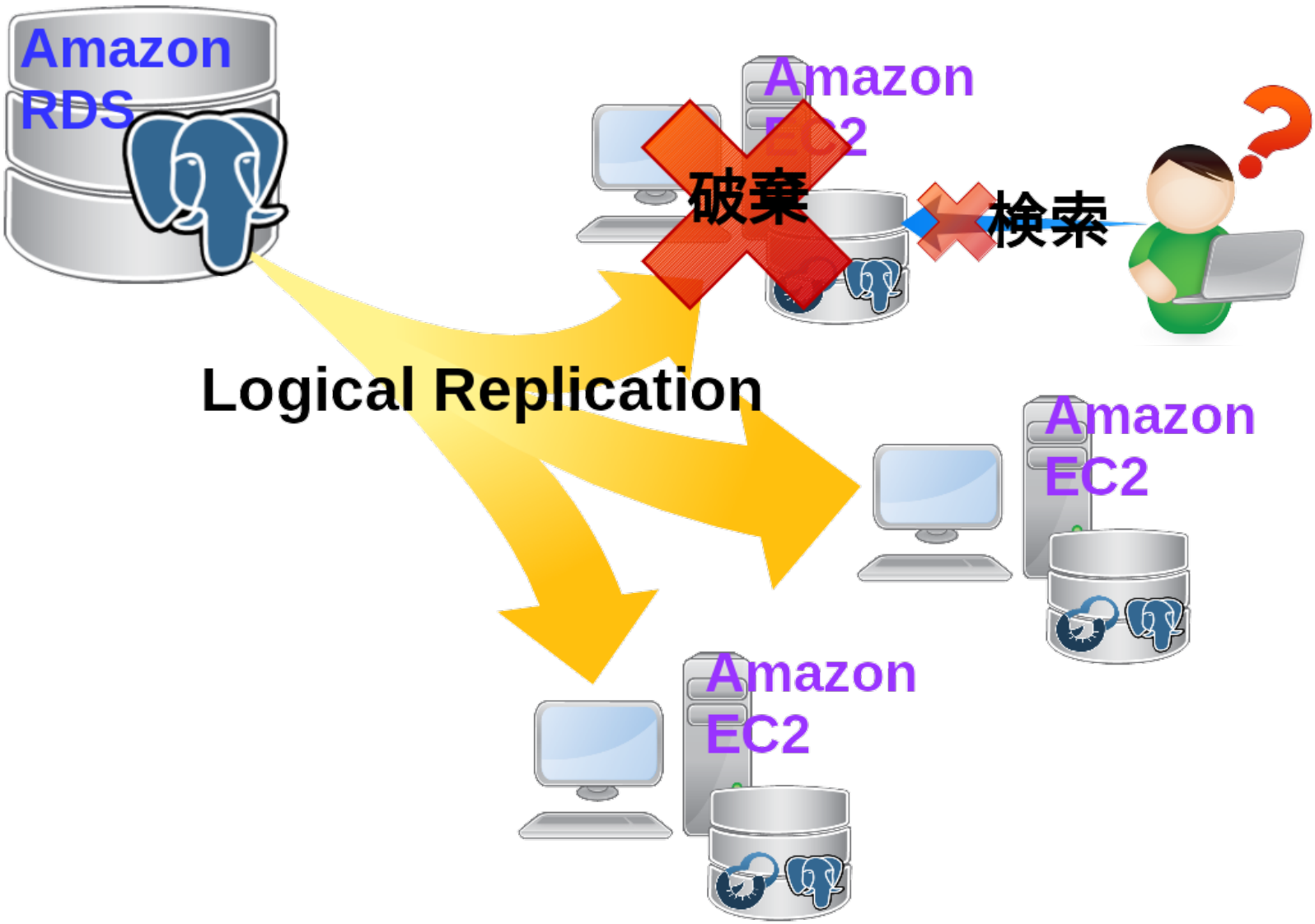
復旧

使い捨てる
復旧は
がんばらない

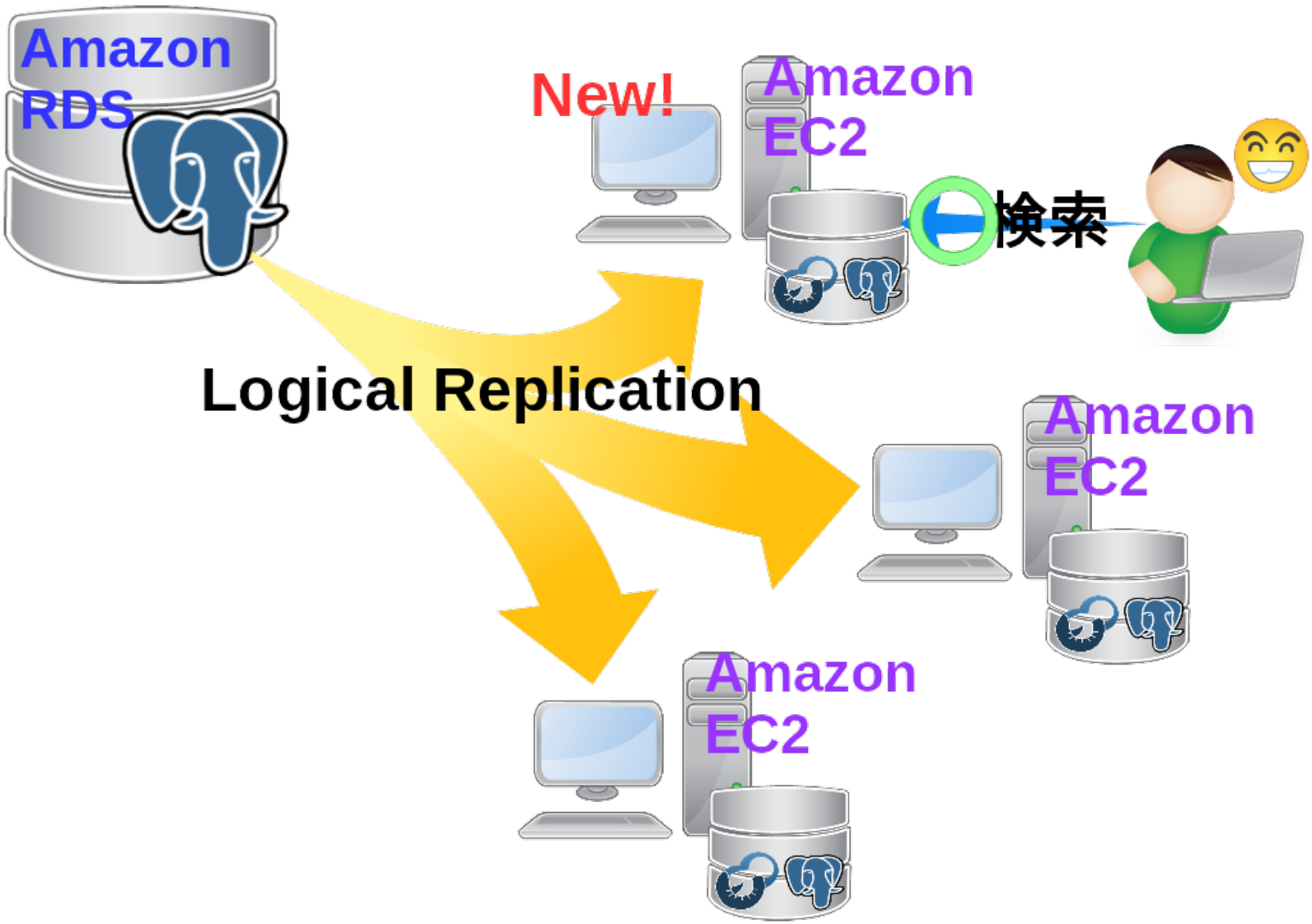
復旧


Amazon
RDS
Amazon
EC2
検索
Logical Replication
Amazon
EC2
Amazon
EC2

復旧


Amazon
RDS
Amazon
EC2
破棄
検索
Logical Replication
Amazon
EC2
Amazon
EC2

復旧


Amazon
RDS
New!
Amazon
EC2
検索
Logical Replication
Amazon
EC2
Amazon
EC2

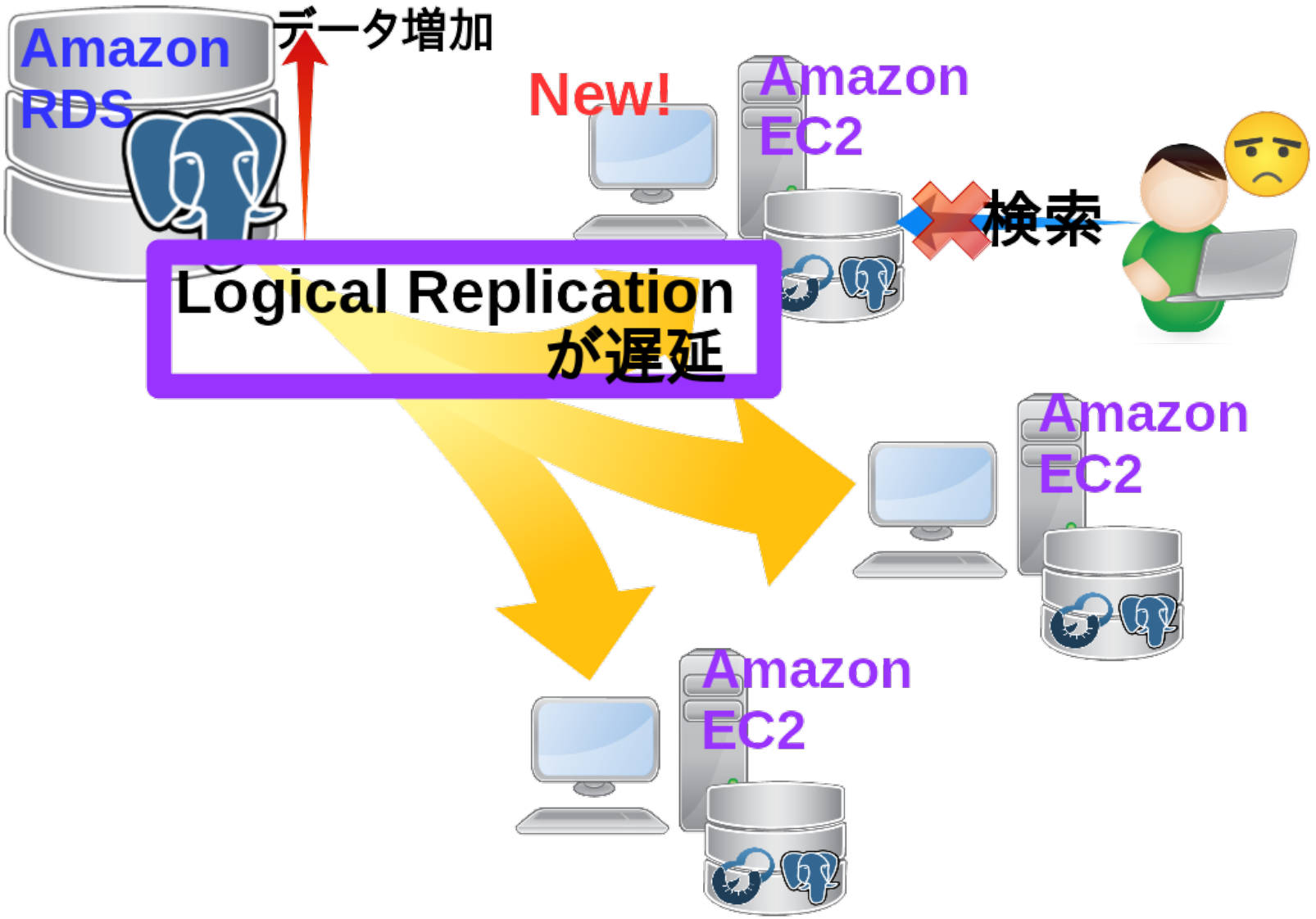
サービス開始時間

もう一つの問題

サービス開始時間

データが増えると
復旧が遅延

サービス開始時間


Amazon
RDS
データ増加
New!
Amazon
EC2
検索
Logical Replication
が遅延
Amazon
EC2
Amazon
EC2

停止時間の見積もり

許容できる停止時間は？

停止時間の見積もり

停止時間を
見積もる

停止時間の見積もり

- 例えば...
  - サービス復旧時間：1日
    - 新しくAmazon EC2を作成して、サービス開始できるまでの時間
  - 故障：1ヶ月に1回の頻度で故障

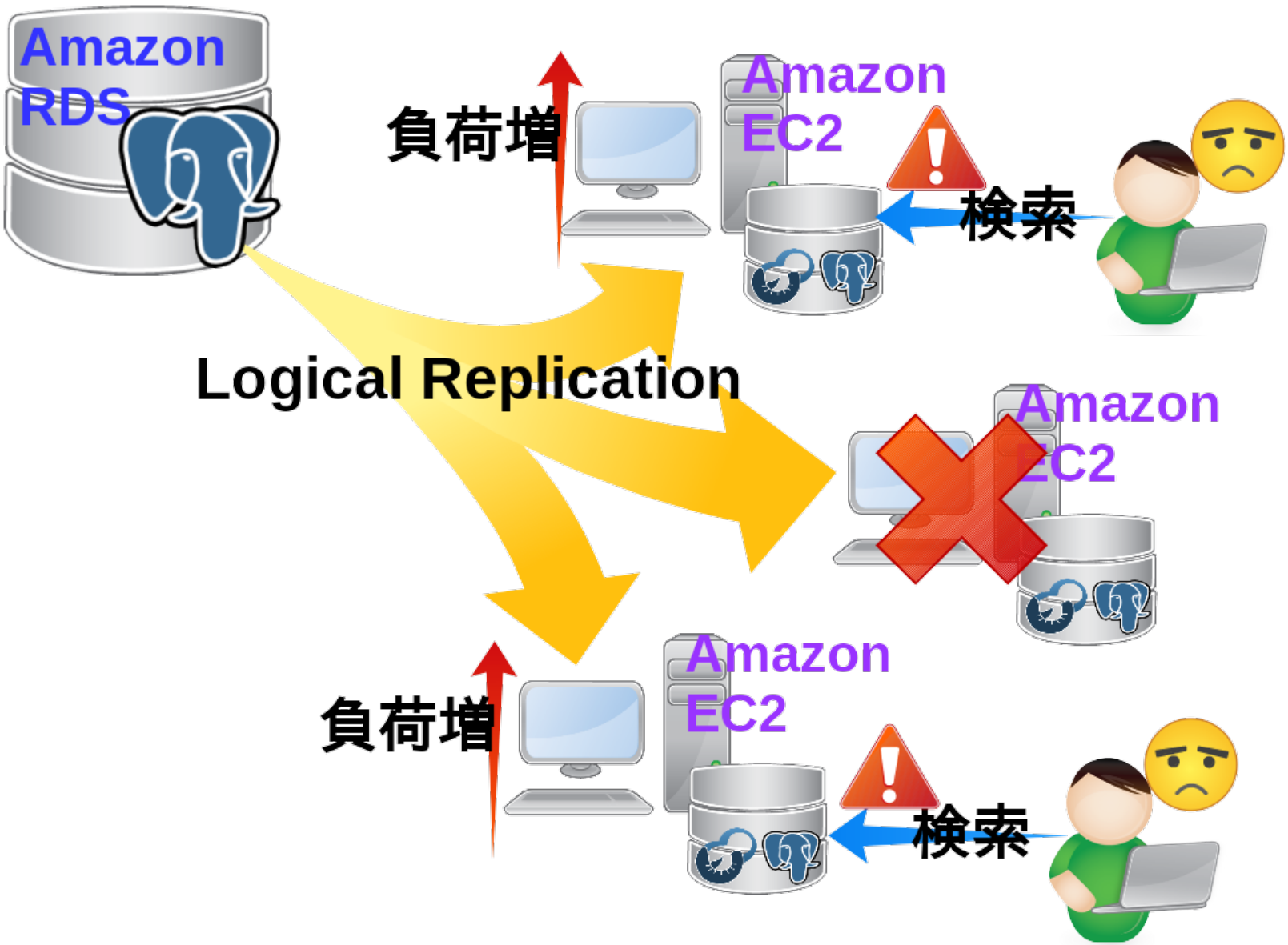
停止時間の見積もり

- リクエスト：
  - 各EC2には均等にリクエストが振り分けられる
- サービス継続：
  - Amazon EC2が3台あれば、サービスを提供可能なくらいの負荷

停止時間の見積もり

- Amazon EC2を3台で運用する場合
  - 1台でも故障するとサービス継続不可

停止時間の見積もり


Amazon
RDS
負荷増
Amazon
EC2
検索
Logical Replication
Amazon
EC2
負荷増
Amazon
EC2
検索

停止時間の見積もり

- この場合の稼働率
  - 平均故障間隔 = 365*24/12 = 730時間
  - 平均復旧時間 = 24時間
  - 稼働率 = 730/754 = 0.9681697612732095 ≒96.8%

停止時間の見積もり

つまり

停止時間の見積もり

- この構成では、1ヶ月に1日程度システムが停止する

停止時間の見積もり

- では、4台運用の場合ではどうなるのか？

停止時間の見積もり

- 1台あたりの稼働率：96.8%
- 1台あたりの故障率：

  100% - 96.8% = 3.2%

- 2台同時に故障する確率：

  0.032*0.032≒0.001=0.1%

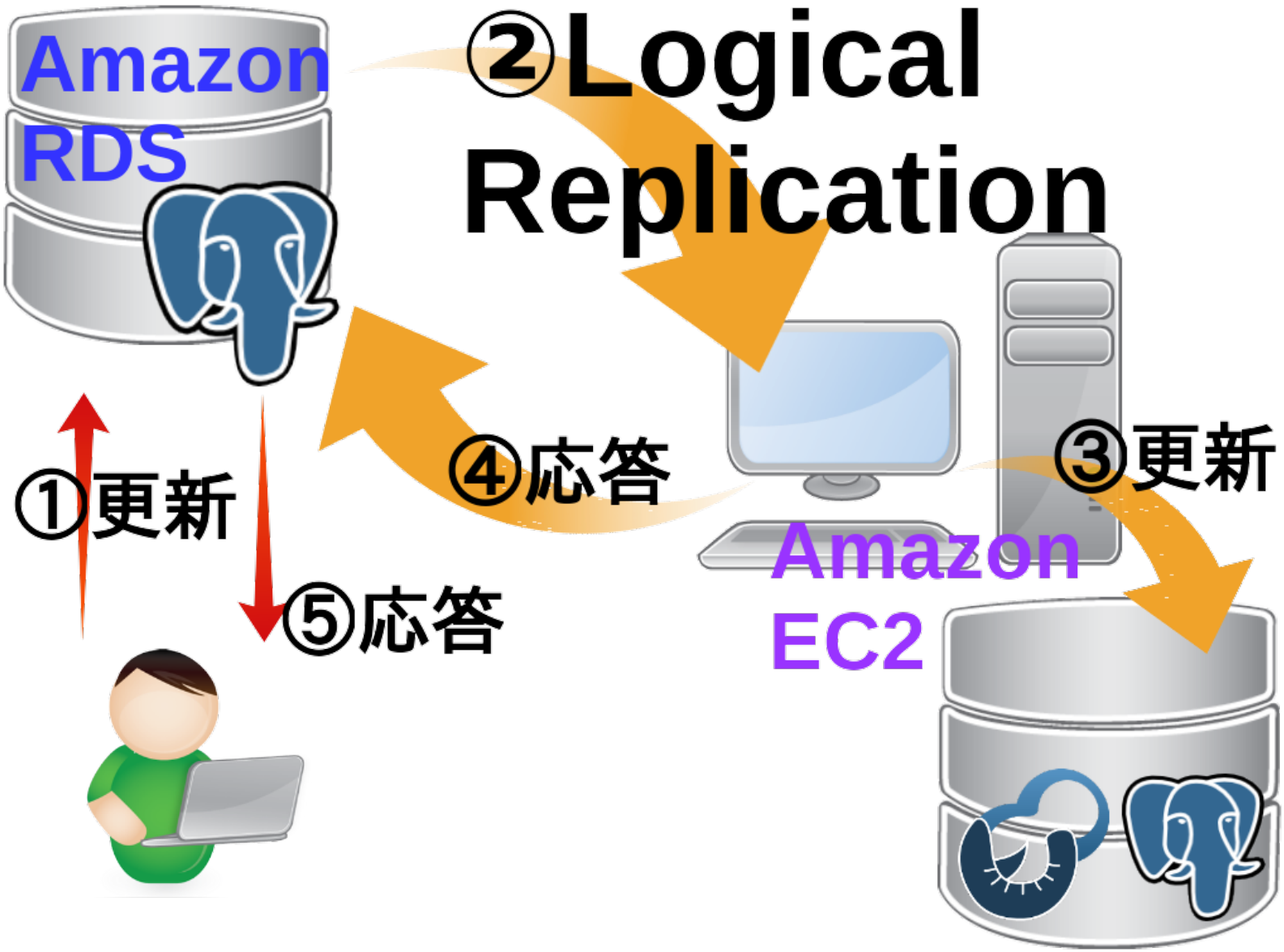
停止時間の見積もり

- したがって、1ヶ月に約45分程度システムが停止する

同期か非同期か

レプリケーションには同期と非同期がある

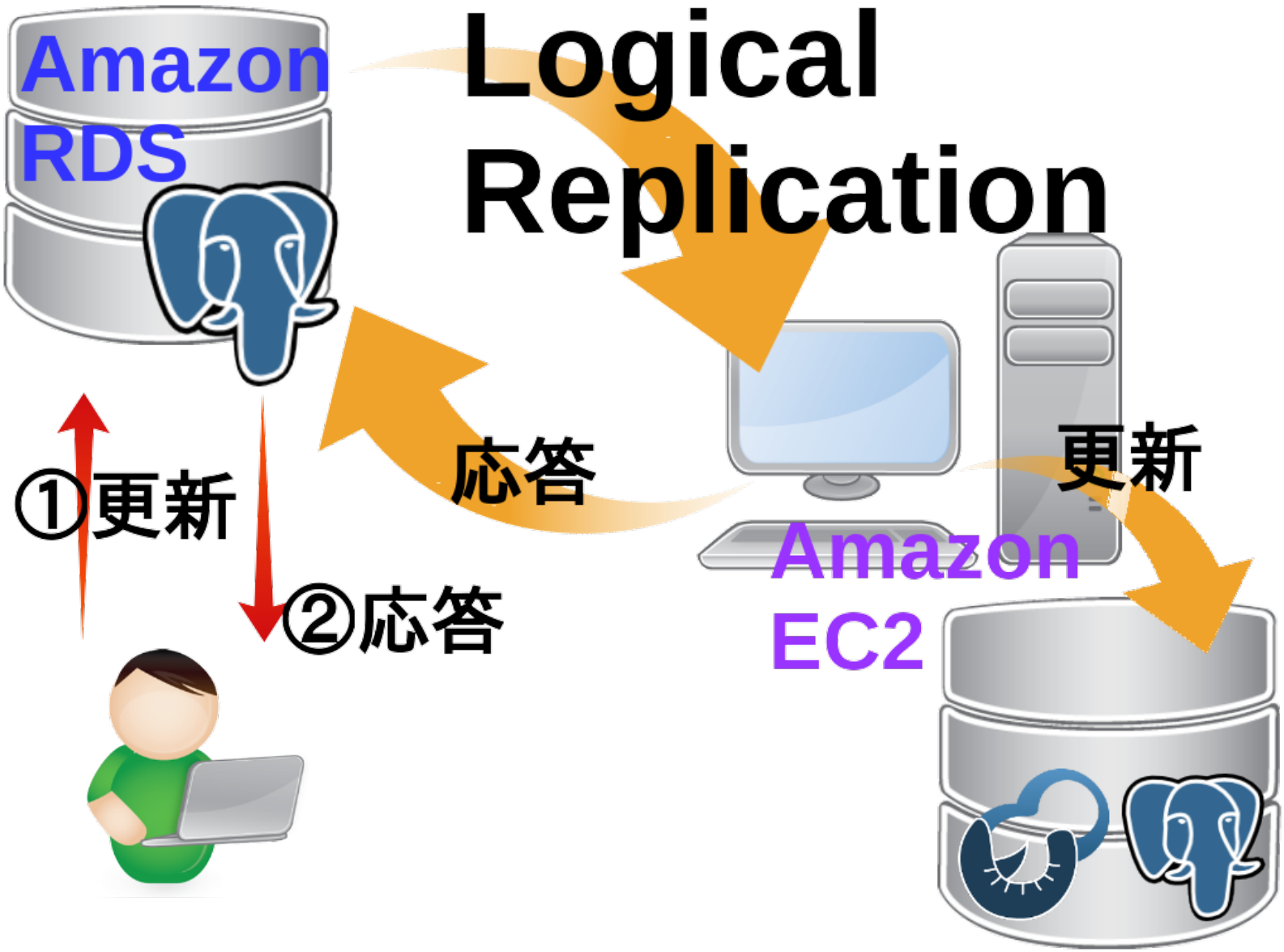
同期レプリケーション



同期レプリケーション

- 更新性能は落ちる
- マスターと同じデータが検索できることを保証

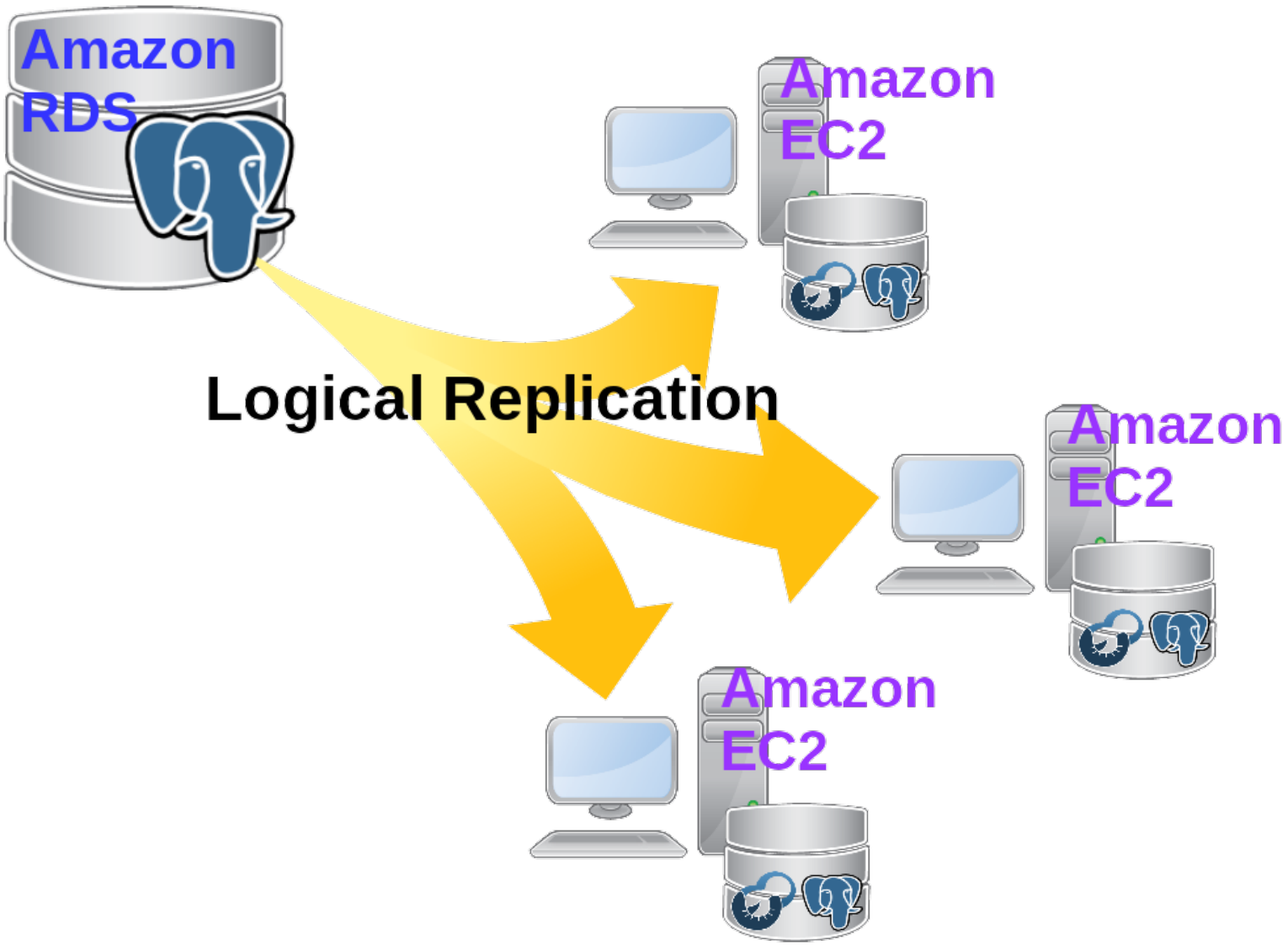
非同期レプリケーション



非同期レプリケーション

- 更新性能に影響はない
- タイミングによっては古いデータが見える

まとめ


Amazon
RDS
Amazon
EC2
Logical Replication
Amazon
EC2
Amazon
EC2

まとめ

なるべく楽に
高機能で高速な全
文検索が
できました！

最後に

- PGroongaについての疑問等は、GitHub、Gitterにて
- ドキュメントも充実
  - https://pgroonga.github.io/ja/

最後に

より突っ込んだお話がしたい場合は
↓↓
問い合わせ先：

https://www.clear-code.com/contact/?type=groonga

最後に

ご静聴ありがとうございました