Zabbixにおけるデータ保護と安全な運用の手法

2023年11月16日

NTTコムエンジニアリング株式会社

岩島　琉偉

© NTT Communications Corporation All Rights Reserved.

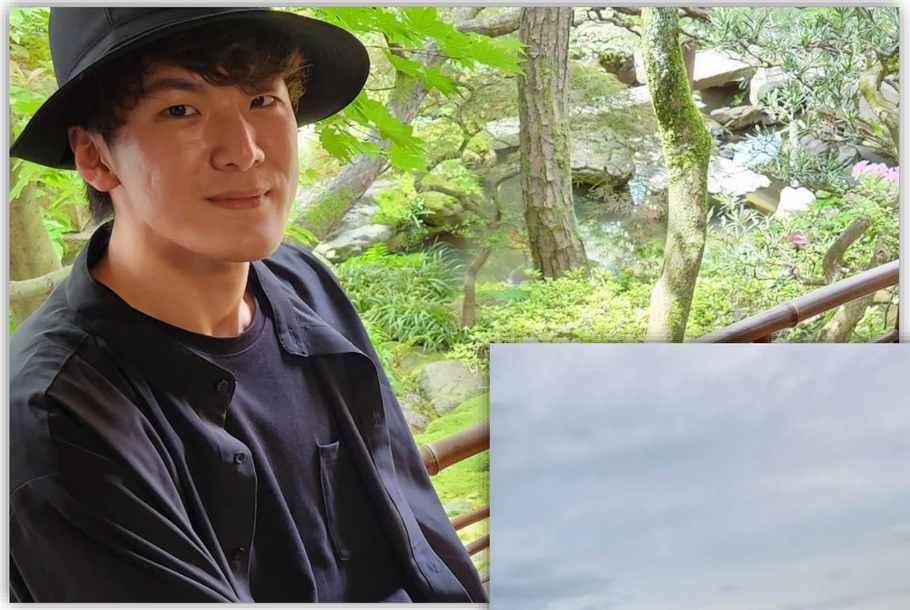
会社紹介

NTTコム エンジニアリング株式会社

- NTTコミュニケーションズのサービスにおける、 提案コンサル、設計構築・保守運用を担うVC子会社
- 自社の製品/サービスは「docomo Business」ブランドにて提供
- NTTコミュニケーションズまたはドコモビジネスソリューションズより販売

© NTT Communications Corporation All Rights Reserved.

登壇者紹介



氏名　：岩島　琉偉

所属　：NTTコム エンジニアリング株式会社
スマートオペレーションサービス部
オペレーションマネジメント部門

出身　：新潟県上越市

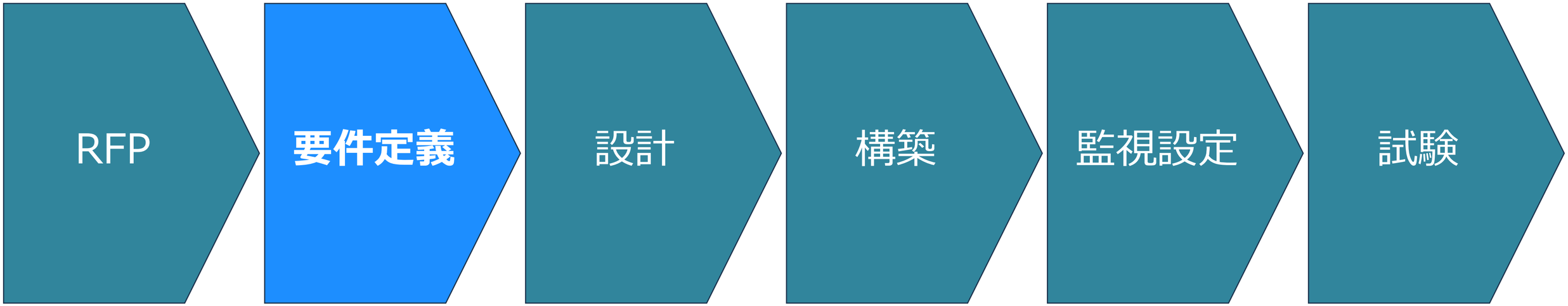
経歴　：2022年 NTTコミュニケーションズ入社
Zabbix関連部署に所属
2023年 NTTコム エンジニアリング
引き続きZabbix関連部署に配属
システム監視の提案・構築の業務に従事

© NTT Communications Corporation All Rights Reserved.

Zabbixのシステム構築において考慮することは？

© NTT Communications Corporation All Rights Reserved.

システム構築の流れ



© NTT Communications Corporation All Rights Reserved.

RASISという設計要素がある

- **Reliability（信頼性）**
- **Availability（可用性）**
- **Serviceability（保守性）**
- **Integrity（完全性）**
- **Security（機密性）**

品質管理

試験

バックアップ

冗長化

監視

システム更新

アクセス制御

暗号化

脆弱性管理

© NTT Communications Corporation All Rights Reserved.

保守性と可用性に着目してみる

Reliability（信頼性）

Availability（可用性）

Serviceability（保守性）

Integrity（完全性）

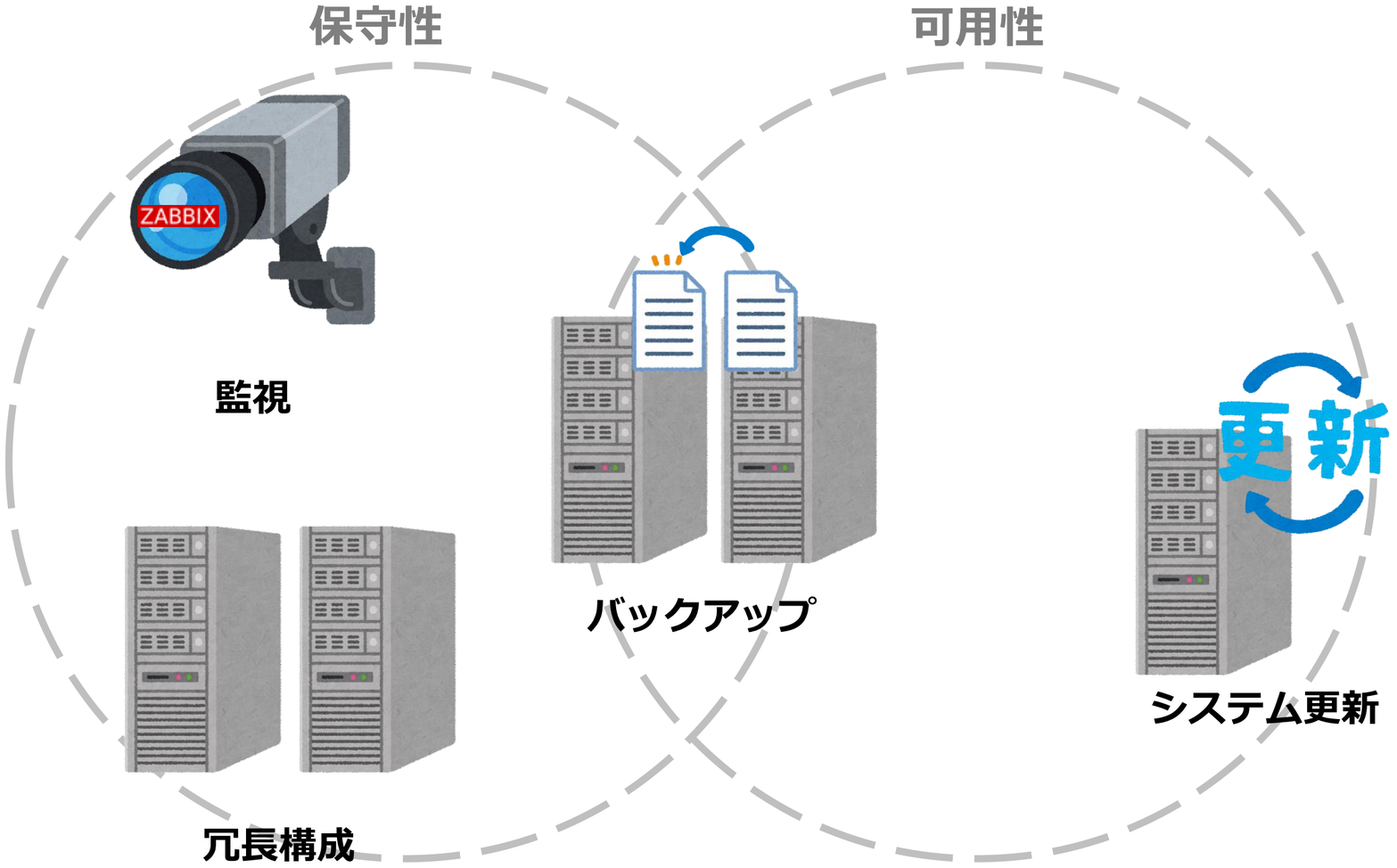
Security（機密性）

© NTT Communications Corporation All Rights Reserved.

Zabbixにおける保守性と可用性

© NTT Communications Corporation All Rights Reserved.

保守性と可用性の要素



システムの安定運用には保守性と可用性は不可欠

© NTT Communications Corporation All Rights Reserved.

システムの安定運用にはバックアップが重要

© NTT Communications Corporation All Rights Reserved.

バックアップの重要性

様々な要因によってデータが危険にさらされる

バックアップが無いと

- データ損失
- 長期間のサービス停止
- 余計なコストがかかる
- 信用低下

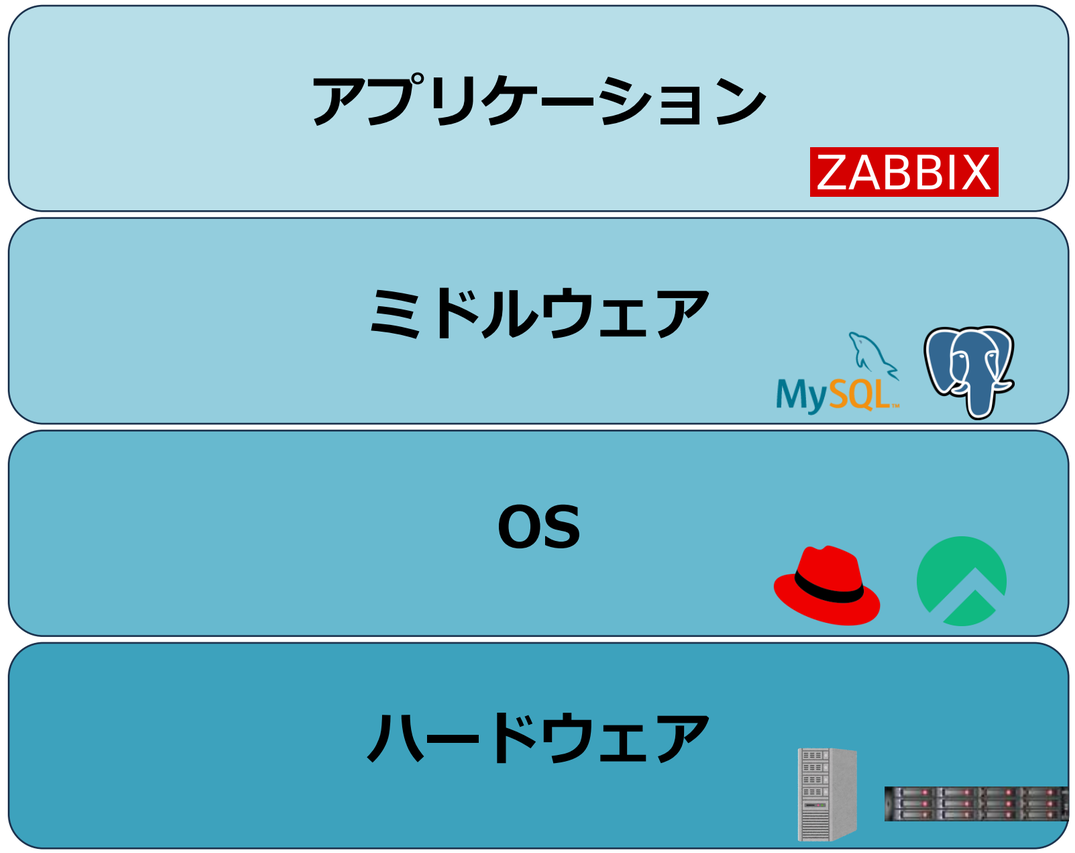
© NTT Communications Corporation All Rights Reserved.

Zabbixにおけるバックアップとは？

© NTT Communications Corporation All Rights Reserved.

システム構成の階層

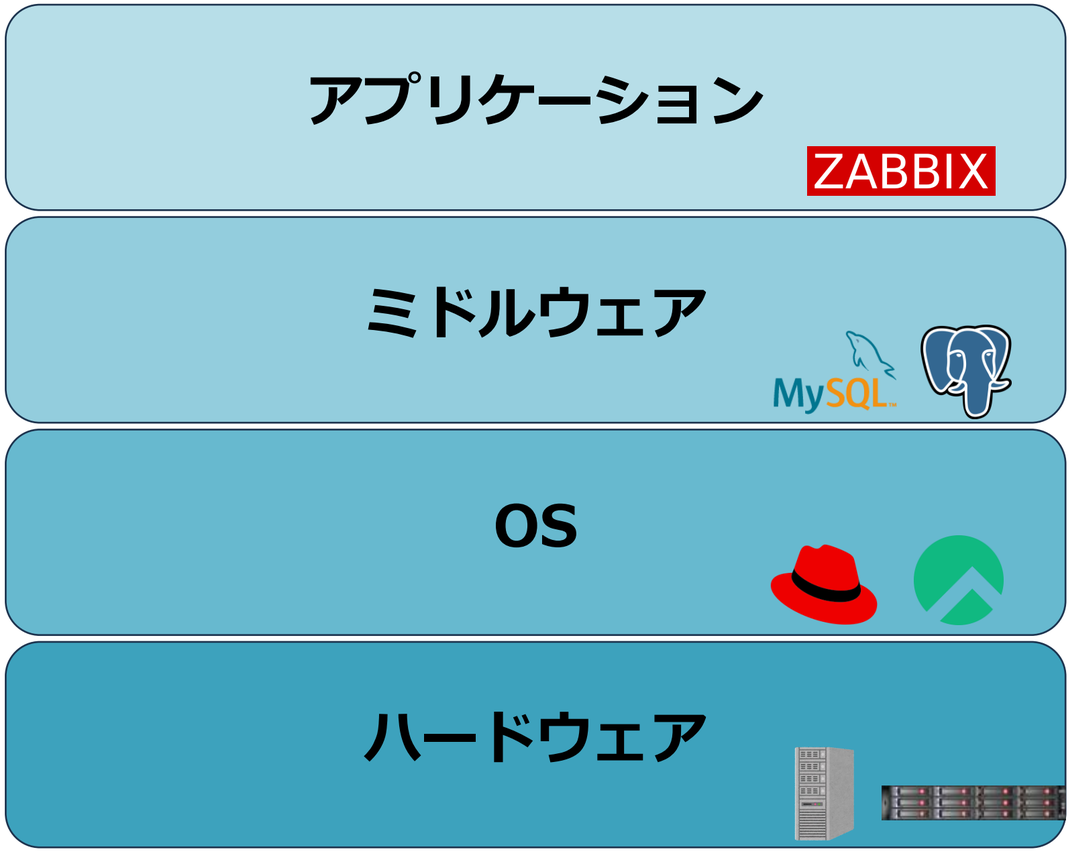
- アプリケーション
  Zabbix
- ミドルウェア
  MySQL, PostgreSQL
- OS
  RedHat, RockyLinux
- ハードウェア
  物理マシン
  仮想マシン



© NTT Communications Corporation All Rights Reserved.

各階層におけるバックアップ手法

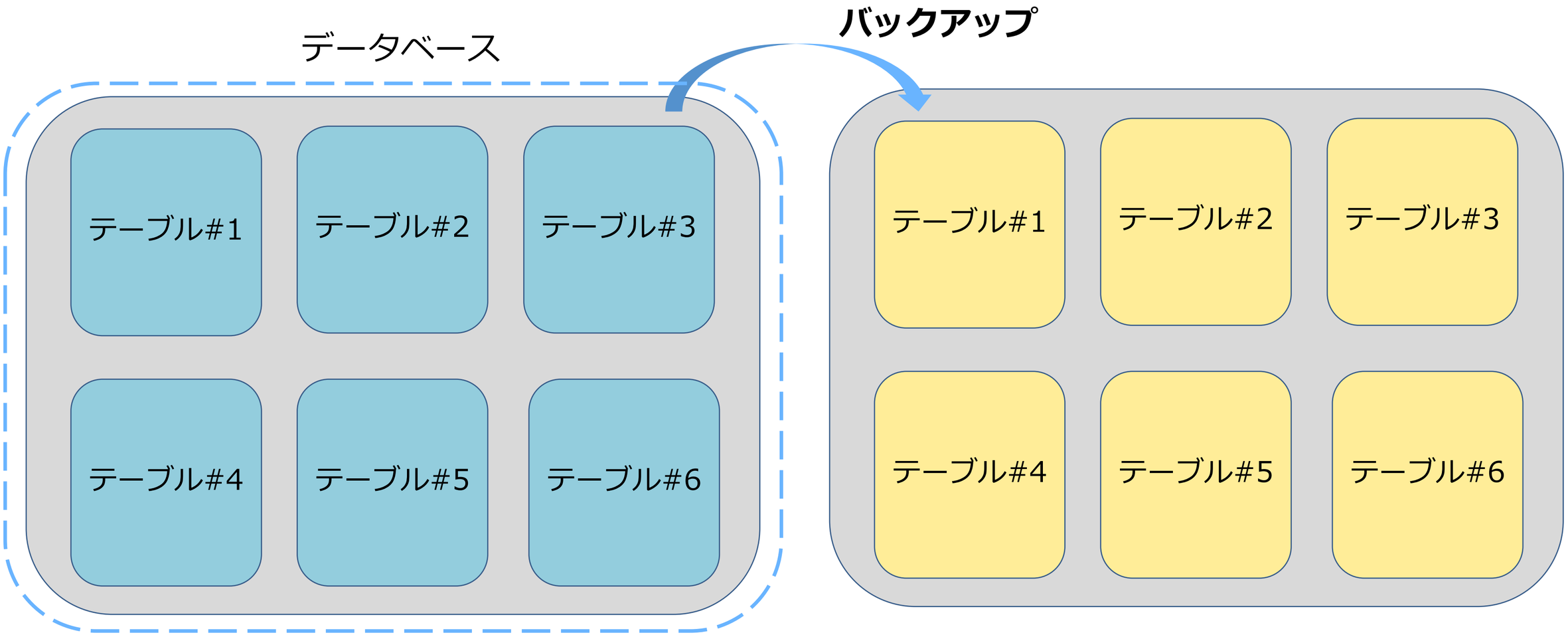
- アプリケーション
  ZabbixからXML出力等
- **ミドルウェア**
  DBMSの機能を用いたバックアップ
- OS
  システム全体をイメージ化する
- ハードウェア
  ストレージでバックアップ
  仮想マシンをバックアップ



© NTT Communications Corporation All Rights Reserved.

完全バックアップ

データベースのすべてのデータを一度にバックアップする方法。

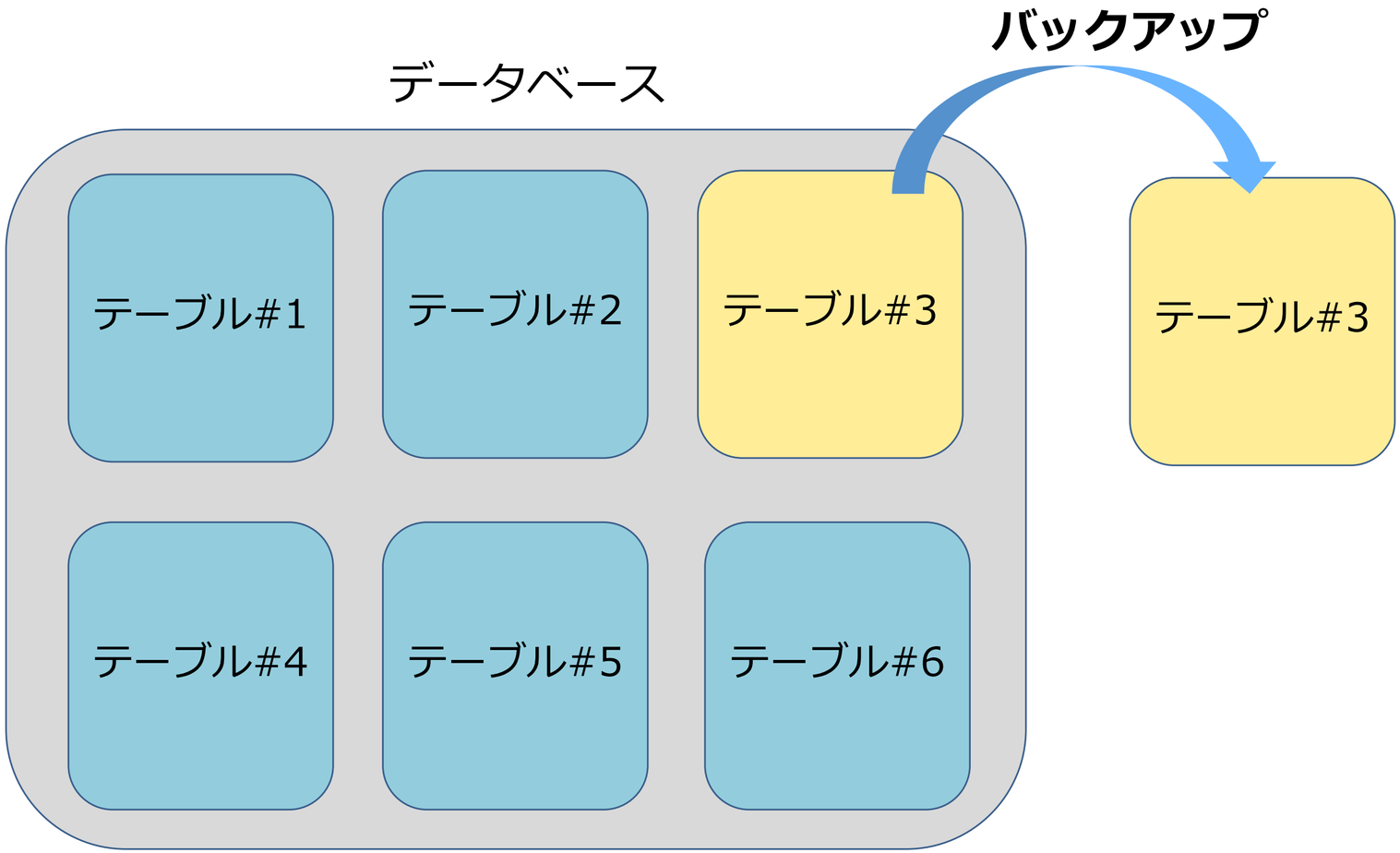


© NTT Communications Corporation All Rights Reserved.

Zabbixの場合は部分バックアップも可能

データベース内の特定のテーブルのみをバックアップ・リストアすることが可能。

Zabbixにおいては監視設定情報のみをバックアップすることができる。



© NTT Communications Corporation All Rights Reserved.

完全バックアップ

メリット

- 完全なデータ復旧が可能
- データ構造を意識しなくてもいい
- リストアが簡単に行える

デメリット

- データ量が大きいので負荷が高くなりやすい
- バックアップデータのサイズが大きくなる
  サイズが大きいため多くの世代を保持できない
- サイズが大きいためバックアップデータの取り回しが悪い

© NTT Communications Corporation All Rights Reserved.

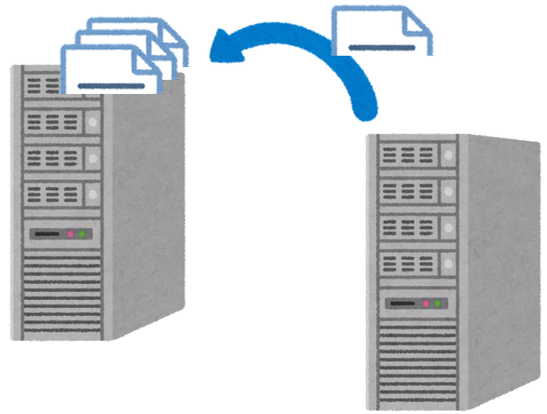
部分バックアップ

メリット

- データ量が小さく負荷がかからない
- バックアップデータのサイズが小さいため、多くの世代を保持できる
- サイズが小さいためバックアップデータの取り回しが良い

デメリット

- 部分的なデータ復旧しかできない
- バックアップする際にデータ構造を意識する必要がある
- リストアする際にテーブル構造に対する深い理解が必要
- リストアの手順を考える必要があり操作が複雑化する



© NTT Communications Corporation All Rights Reserved.

完全バックアップ・リストアの手順

コマンド例

- バックアップ方法

MySQL :

```
#mysqldump -u [USERNAME] -p [DBNAME] > backupfile.sql
```

PostgreSQL:

```
#pg_dump -U [USERNAME] -d [DBNAME] -f backupfile.sql
```

- リストア方法

MySQL :

```
#mysql -u [USERNAME] -p [DBNAME] < backupfile.sql
```

PostgreSQL:

```
#psql -U [USERNAME] -d [DBNAME] -f backupfile.sql
```

© NTT Communications Corporation All Rights Reserved.

部分バックアップ・リストアの手順

- バックアップ方法

MySQL :

```
#mysqldump -u [USERNAME] -p [DBNAME] [TABLENAME] > backupfile.sql
```

PostgreSQL:

```
#pg_dump -U [USERNAME] -d [DBNAME] [TABLENAME] -f backupfile.sql
```

- リストア方法

MySQL :

```
#mysql -u [USERNAME] -p [DBNAME] < backupfile.sql
```

PostgreSQL:

```
#psql -U [USERNAME] -d [DBNAME] -f backupfile.sql
```

理論上は可能だが、Zabbixでは安易な部分バックアップでは動かなくなる

© NTT Communications Corporation All Rights Reserved.

部分バックアップのリストアに関する注意点

© NTT Communications Corporation All Rights Reserved.

アイテムの設定をバックアップしたい場合

アイテムの監視設定情報だけほしいからアイテムのテーブルだけ取ってリストアすればいいんじゃね？

それだけだとリストアしても上手くいかないよ！！

アイテムのIDはZabbixの中で管理されているので矛盾が起きてしまう...

© NTT Communications Corporation All Rights Reserved.

部分バックアップによる問題例

アイテムのIDと、IDの管理情報が入っているidsテーブルとで矛盾が起きる問題を再現してみる。

1. idsテーブルを部分バックアップする

```
[root@localhost ~]# mysqldump -u root -p zabbix ids > /test/hogehoge.sql
Enter password:
[root@localhost ~]#
[root@localhost ~]# ls /test/
hogehoge.sql
```

© NTT Communications Corporation All Rights Reserved.

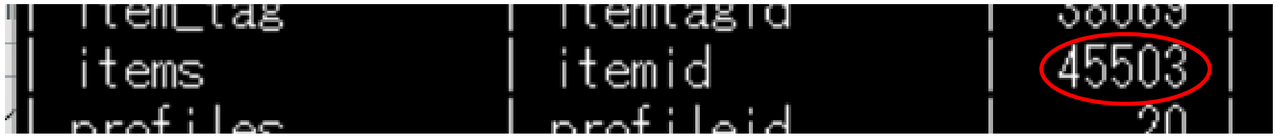
部分バックアップによる問題例

2．アイテムを2つ追加してみた

```
mysql> select * from ids;
+-----------------+-----------------+--------+
| table_name      | field_name      | nextid |
+-----------------+-----------------+--------+
| functions       | functionid      |  33084 |
| graphs          | graphid         |   2581 |
| graphs_items    | gitemid         | 222300 |
| item_discovery  | itemdiscoveryid |   8810 |
| item_preproc    | item_preprocid  |  94457 |
| item_tag        | itemtagid       |  38069 |
| items           | itemid          |  45503 |
| profiles        | profileid       |     20 |
| trigger_depends | triggerdepid    |  55563 |
| trigger_tag     | triggertagid    |   7834 |
| triggers        | triggerid       |  23319 |
| users           | userid          |      3 |
| users_groups    | id              |      5 |
+-----------------+-----------------+--------+
```

idsテーブル

idsテーブルのitemidのnextidの変化



© NTT Communications Corporation All Rights Reserved.

問題の発生

3. リストアしてみた

```
[root@localhost ~]#
[root@localhost ~]# ls /test/
hogehoge.sql
[root@localhost ~]#
[root@localhost ~]# cat /test/hogehoge.sql | mysql -u root -p zabbix
Enter password:
```

リストアを行ってからアイテムを登録しようとするとエラーが発生した。
監視設定情報のほとんどはIDで管理されているため、同様の事象が発生する。

アイテム作成失敗の図

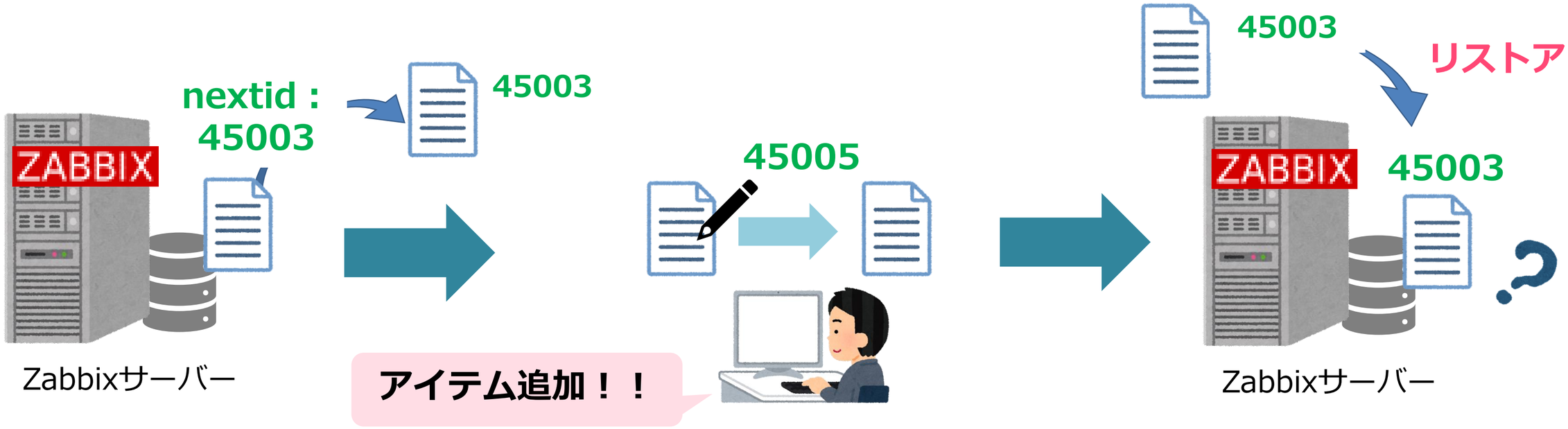
© NTT Communications Corporation All Rights Reserved.

リストアによるIDの矛盾

新たに追加されるアイテムやトリガー、ユーザーなどの情報に割り振られるID番号としてidsテーブルにはnextidが記録されている。

リレーショナルを考慮しないでバックアップを行うとIDの矛盾等が起きて予期せぬ動作を起こすことがある。

今回の検証では古いidsテーブルをリストアしたため、アイテムのnextidに矛盾が生じた。



IDの矛盾が起きないようにidsテーブルや、IDが関わる関連したテーブルも同時にバックアップを行わないといけない

© NTT Communications Corporation All Rights Reserved.

監視設定のみ部分バックアップする場合

バックアップ対象、データを削除するテーブル、バックアップ対象から除外にするテーブルを一部紹介する。

バックアップ対象	データを削除するテーブル	除外するテーブル
items	**alerts**	**history**
item_tag	**escalations**	**history_unit**
triggers	**task_acknowledge**	**history_text**
trigger_discovery	**task_check_now**	**history_log**
hosts	**task_data**	**trends**
hosts_groups	**task_result**	**trends_unit**
……	**……**	**……**
アイテムやトリガー、ホストなどのバックアップしたい監視設定が記録されているテーブル	アラートやタスクなどの中間データはバックアップした時点で行わないといけないものでリストアしても困る	ヒストリやドレンドの監視データを記録しているものはデータ量が大きいので除外

© NTT Communications Corporation All Rights Reserved.

バックアップ・リストアのまとめ

1. バックアップを怠ることによってデータ損失やサービス停止、信用低下が起きる可能性がある。
2. それぞれの環境や要件、目的に合わせたバックアップの手段を選択する。
3. 部分バックアップを行う際はリレーショナルを考慮したテーブルを選択する。
4. 監視設定のバックアップをする際は監視データの除外と中間データの削除を行い整合性を保つ。

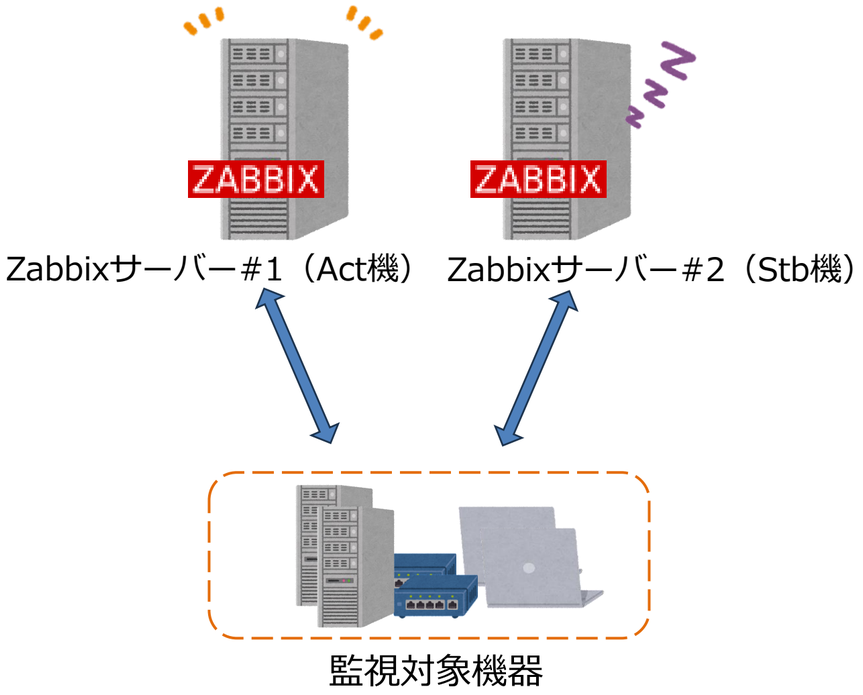
© NTT Communications Corporation All Rights Reserved.

部分バックアップを別のZabbixに入れれば Act-Act構成できる？

© NTT Communications Corporation All Rights Reserved.

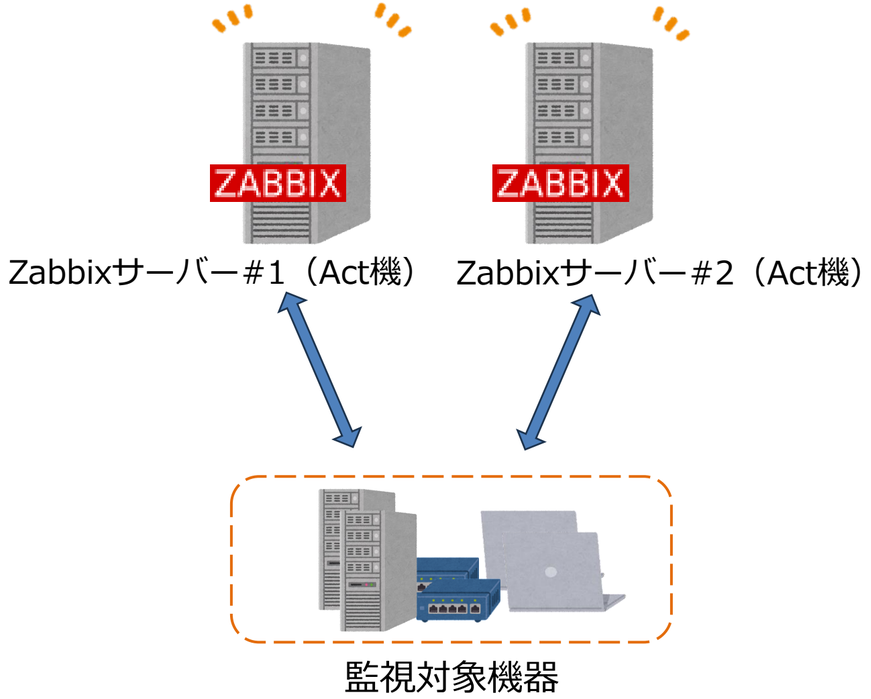
ZabbixのHA構成

Act-Stb構成



Stb機にAct機の監視設定を流し込むだけでよい

Act-Act構成



Act-Actはただ監視設定を流し込むだけでは問題がある

© NTT Communications Corporation All Rights Reserved.

例）こういう条件で運用した場合

状況

2号機のみに飛んでくるトラップがある構成。
このまま同期させると1号機のイベントで上書きしてしまう。

イベント情報を除外すればいい？

problemテーブルもeventsテーブルとセットのIDで管理されているので除外する必要がある。

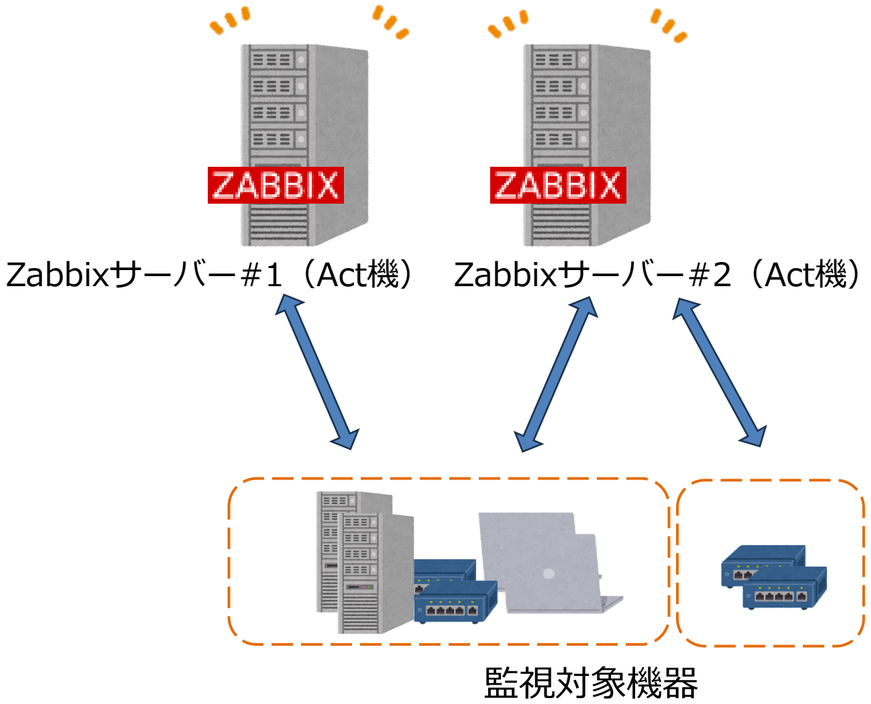
acknowledgesも受諾情報でeventsに紐づいているので同時に除外する必要がある。

この3つを除外するだけでいい？

テーブルを除外した場合、IDの情報を同期させるのは問題がある。

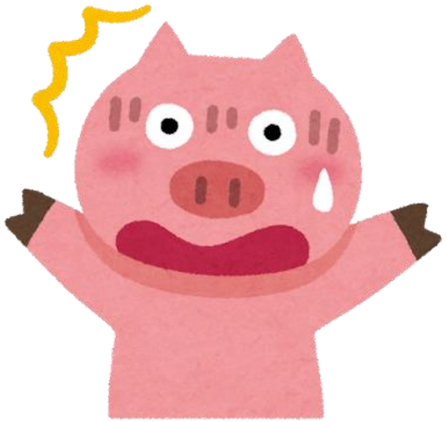
そのため、idsテーブルの情報は単純に同期させるわけにはいかない。

とあるAct-Act構成



© NTT Communications Corporation All Rights Reserved.

面倒くさいしidsテーブルの中身を空にしよう



いいんですかそんなことして！？

© NTT Communications Corporation All Rights Reserved.

いいよ～♪

idsテーブルに無いIDを採番したら、正しい
nextidを記録しておくよ

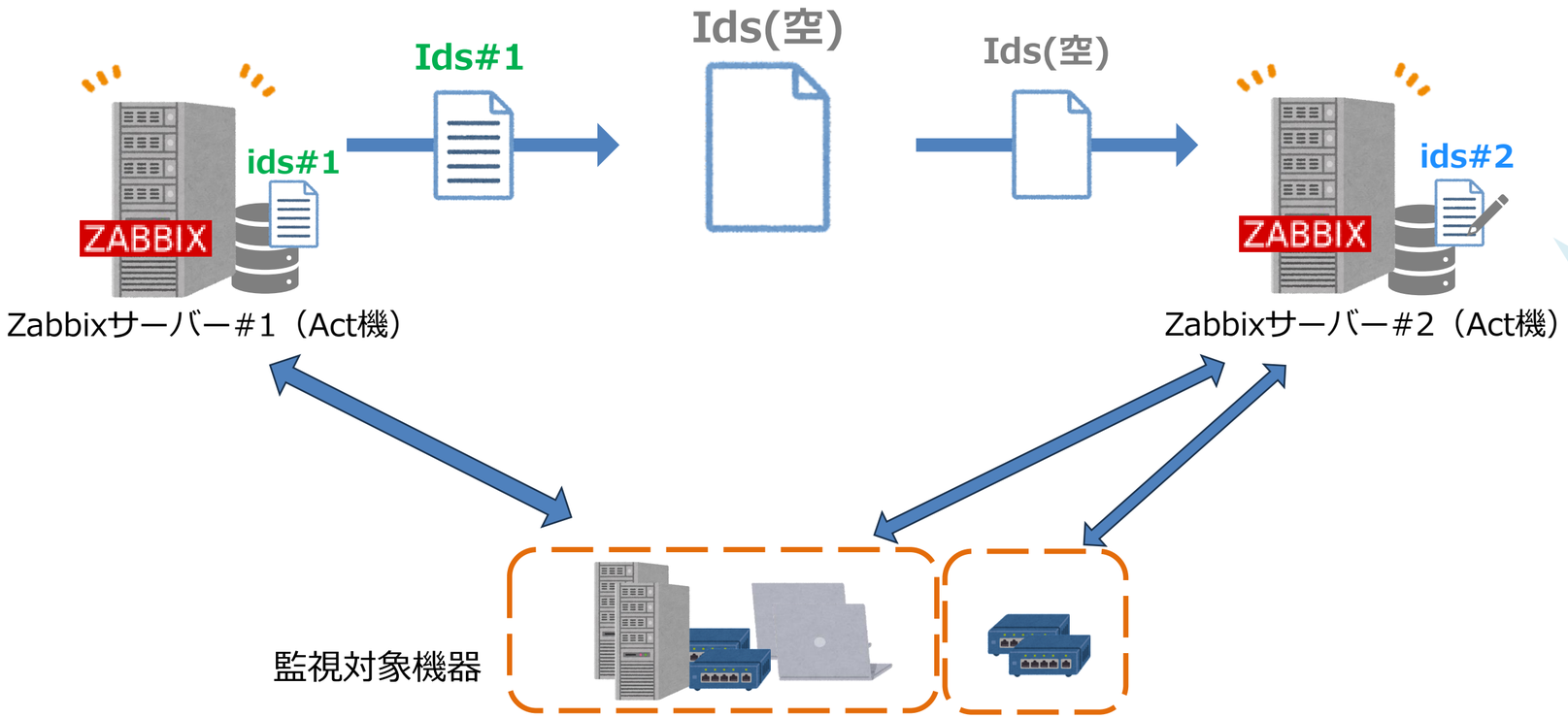
© NTT Communications Corporation All Rights Reserved.

IDの矛盾問題の解決

解決

同期の際にidsテーブルを空にすると、次のIDが振られる際に自動的に最後の番号+1のIDが挿入される。
そのためnextidの矛盾問題は無くなる。

これによって同期の際にacknowledgesのnextidに矛盾が起きることによって障害がクローズできない等の問題は解決される。



設定追加の時にidsテーブルに無いIDが採番されたら正しいnextidを記録しておくよ

© NTT Communications Corporation All Rights Reserved.

idsのデータを削除して検証

idsテーブルのnextidに矛盾があると様々な動作で支障が出る。

アイテム作成失敗の図

© NTT Communications Corporation All Rights Reserved.

idsのデータを削除して検証

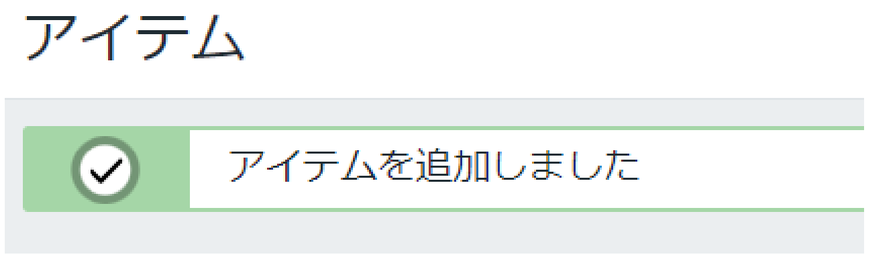
実際にidsテーブルの中身を空にしてアイテムを追加してみる。

```
mysql> select * from ids;
Empty set (0.00 sec)
```

mysql> truncate table ids; で中身を削除

```
mysql> select * from ids;
+------------+------------+--------+
| table_name | field_name | nextid |
+------------+------------+--------+
| items      | itemid     |  45506 |
| profiles   | profileid  |     58 |
+------------+------------+--------+
2 rows in set (0.00 sec)
```

新しくアイテムを追加した後idsを確認をする



成功！！

© NTT Communications Corporation All Rights Reserved.

Act-Act構成まとめ

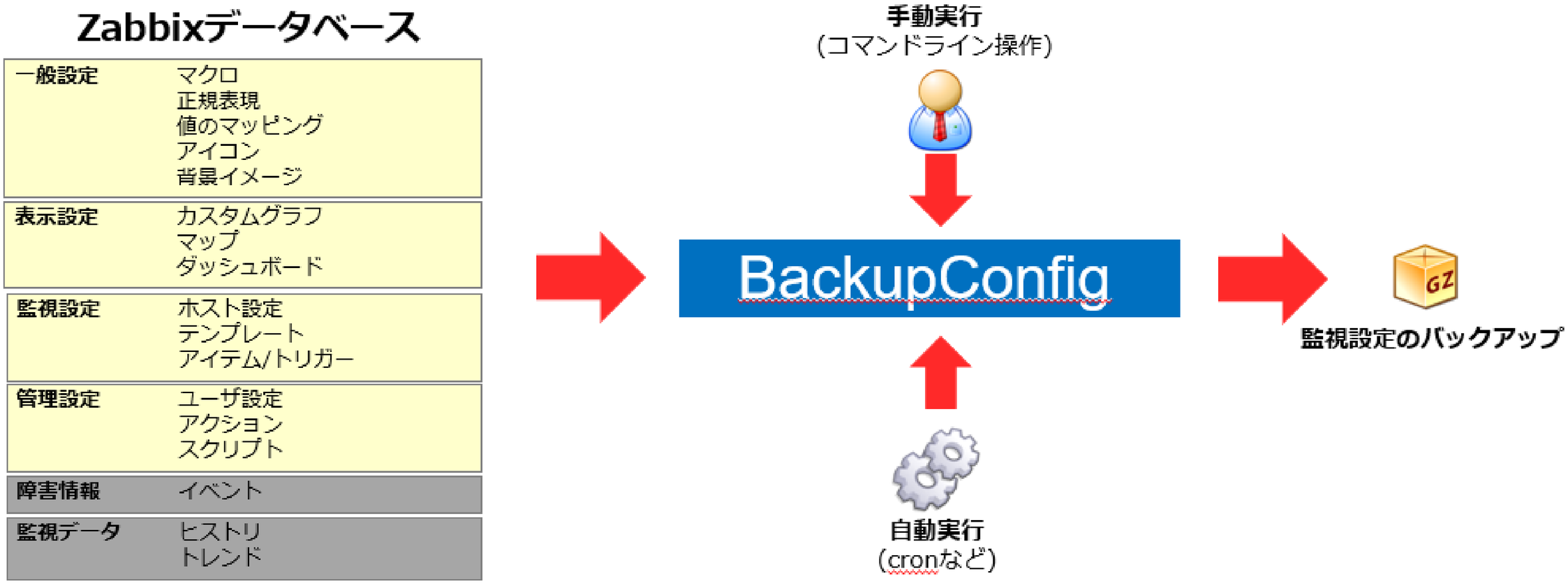
1. ZabbixのDBは様々なテーブルが連携しているため、欲しい情報に関連したテーブルも一緒にバックアップしなければならない。
2. 独立で動かすためには障害情報(problem)、イベント情報(events)、受諾情報(acknowledges)は同期してはいけない。
3. テーブルを除外した場合、idsテーブルの情報の整合性に気を付ける必要がある。

© NTT Communications Corporation All Rights Reserved.

補足

弊社では監視設定のバックアップをツール化して自動で行っています。

「BackupConfig」

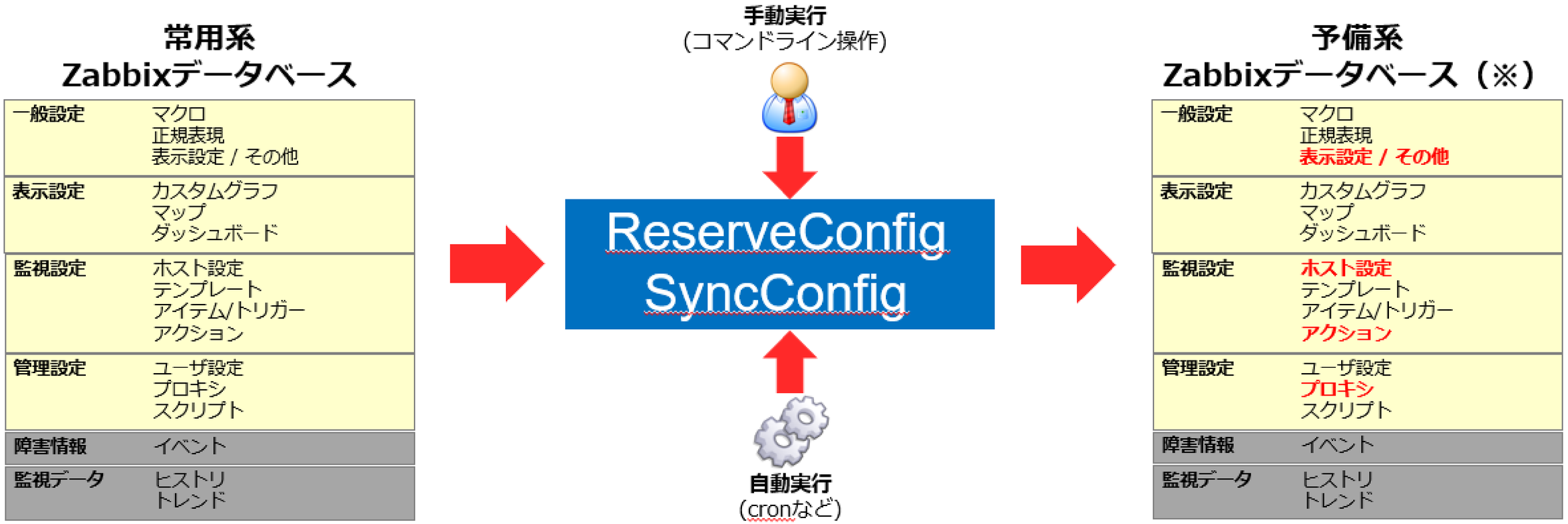


© NTT Communications Corporation All Rights Reserved.

補足

また、Act-Stb構成とAct-Act構成の同期も提供しています。

「**ReserveConfig**」



ほかにもZabbixの様々なツールを提供しております。

© NTT Communications Corporation All Rights Reserved.

まとめ

- ✓ 保守性と可用性を高めるためにはバックアップが重要
- ✓ Zabbixでは部分バックアップが有効な手法
- ✓ 部分バックアップではテーブルの構造を意識する
- ✓ Act-Act構成でDBを同期する場合、データの整合性に気を付ける

© NTT Communications Corporation All Rights Reserved.

ご清聴ありがとうございました。

© NTT Communications Corporation All Rights Reserved.