SRA OSS, INC.

Zabbix クラスタ構成解説

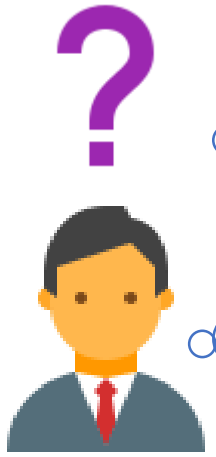
SRA OSS, Inc. 日本支社
OSS 事業本部　基盤技術グループ
赤松　俊弘

ZABBIX
PREMIUM PARTNER

Zabbix の冗長化

Active/Active ?
Active/Standby ?

構成 ?
構築方法 ?

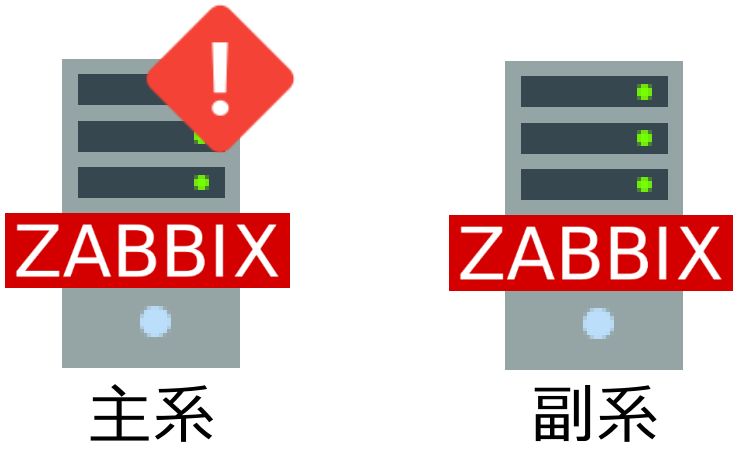


ネイティブ機能 ?
サードパーティ製品 ?

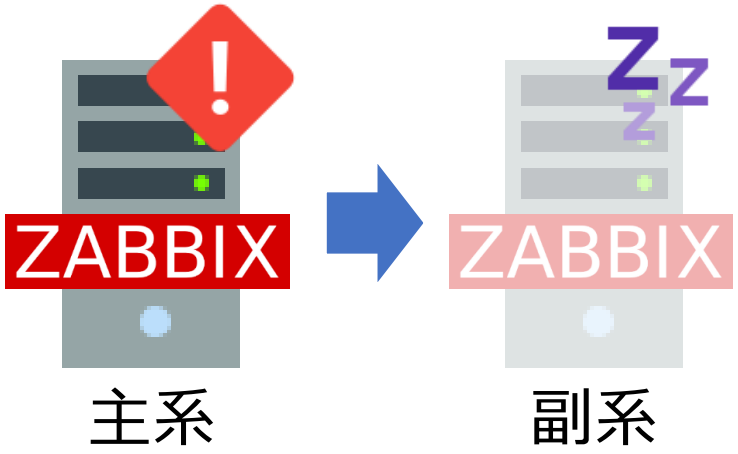
© 2022 SRA OSS, Inc. Japan

Active/Active V.S. Active/Standby

- Active/Active
  - 主系副系並行同時稼働
  - 主系故障時ダウンタイムなしで継続監視可能
  - 監視履歴は各系で異なる



- Active/Standby
  - 主系のみ稼働
  - 主系故障時にはクラスタソフトで副系を起動
  - 副系が起動するまで多少のダウンタイム有
  - 監視履歴は各系で同一

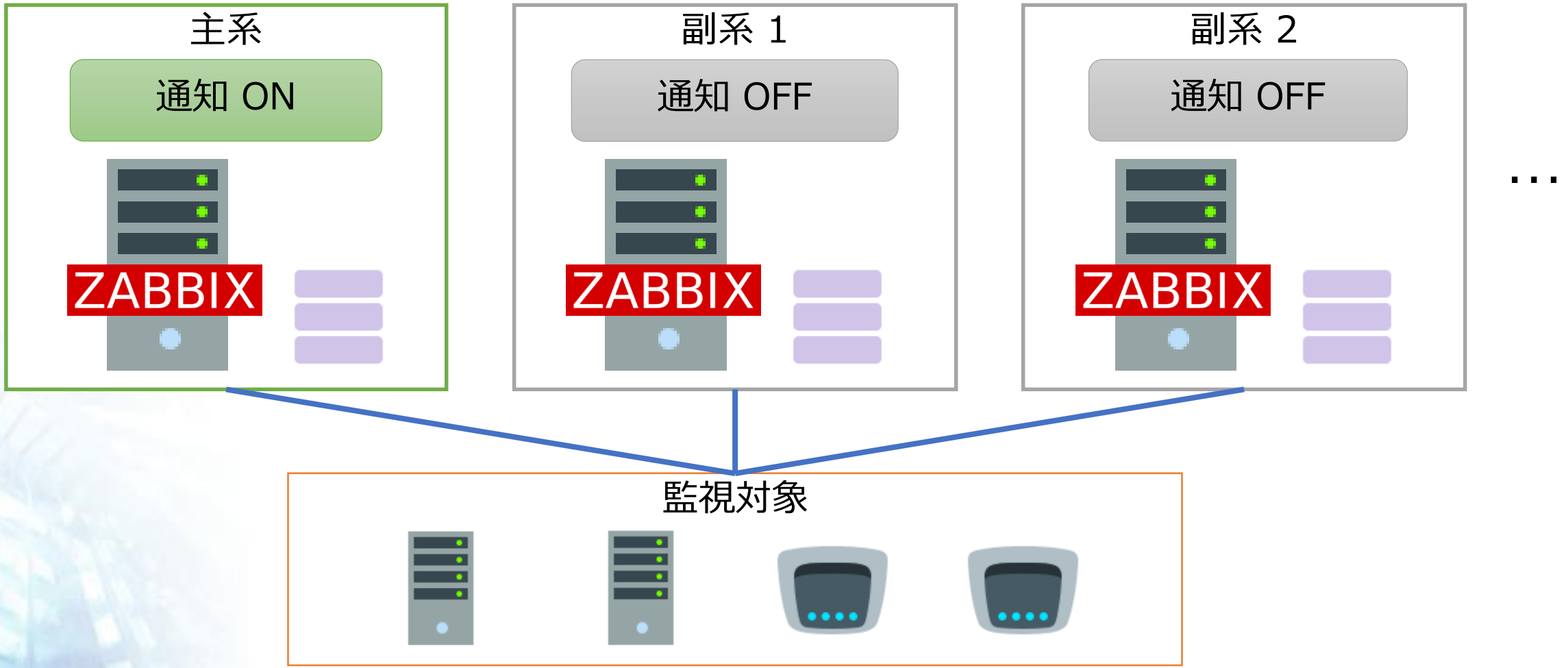


© 2022 SRA OSS, Inc. Japan

Zabbix Active/Active HA クラスタ

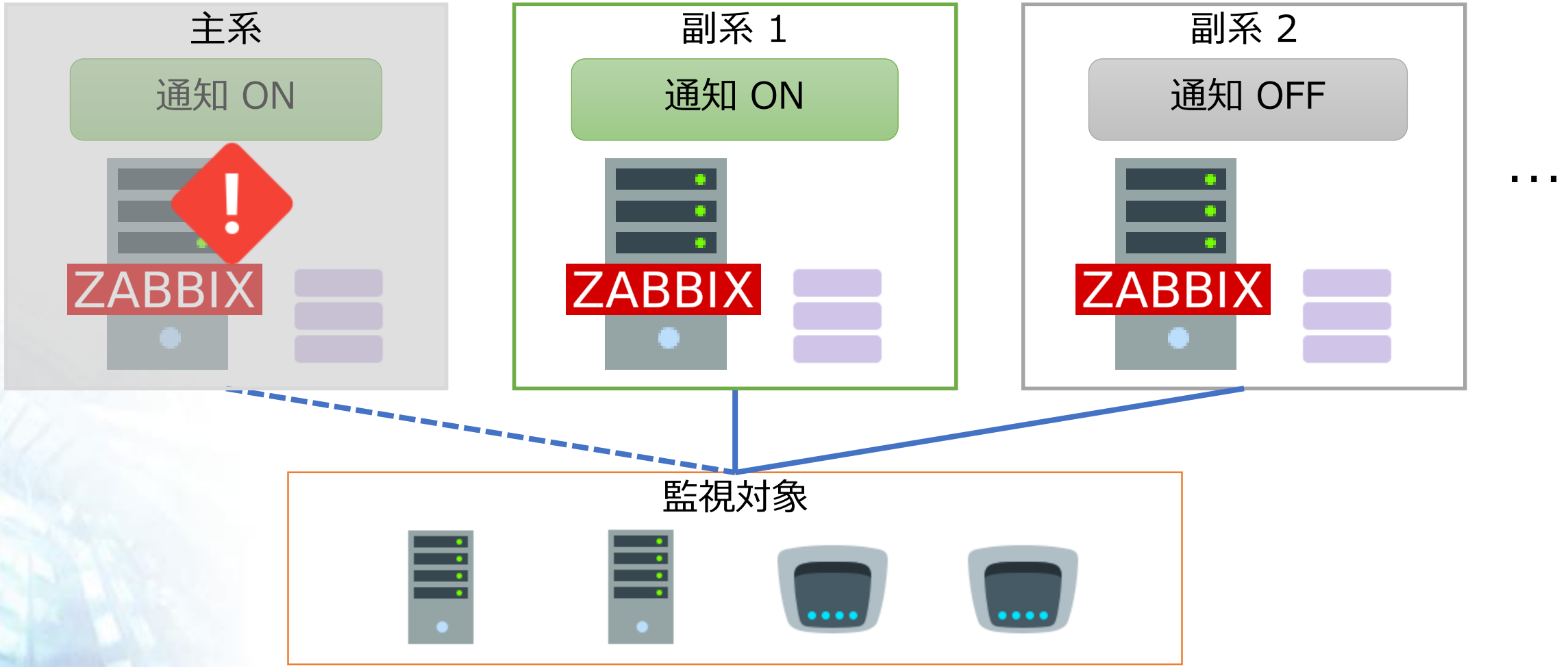
© 2022 SRA OSS, Inc. Japan

Zabbix Active/Active HA クラスタ



© 2022 SRA OSS, Inc. Japan

Zabbix Active/Active HA クラスタ



© 2022 SRA OSS, Inc. Japan

Active/Active クラスタ運用時の疑問

© 2022 SRA OSS, Inc. Japan

Active/Active クラスタの考慮点

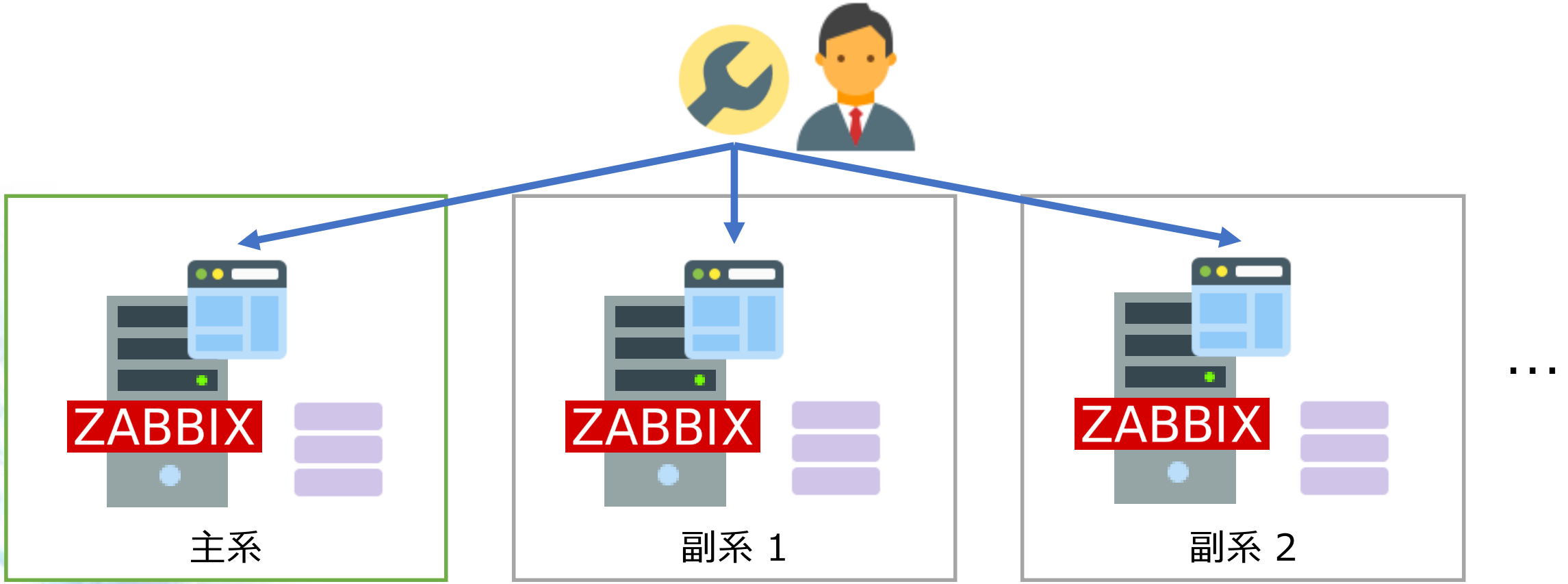
- 設定の同期
  - 全 Zabbix 間で設定を同一にする必要あり
  - 副系では通知を OFF にする必要あり

- 主系から副系への切替作業
  - 主系の障害検知
  - 切替時に副系で通知を ON にする必要あり

© 2022 SRA OSS, Inc. Japan

設定の同期方法の検討

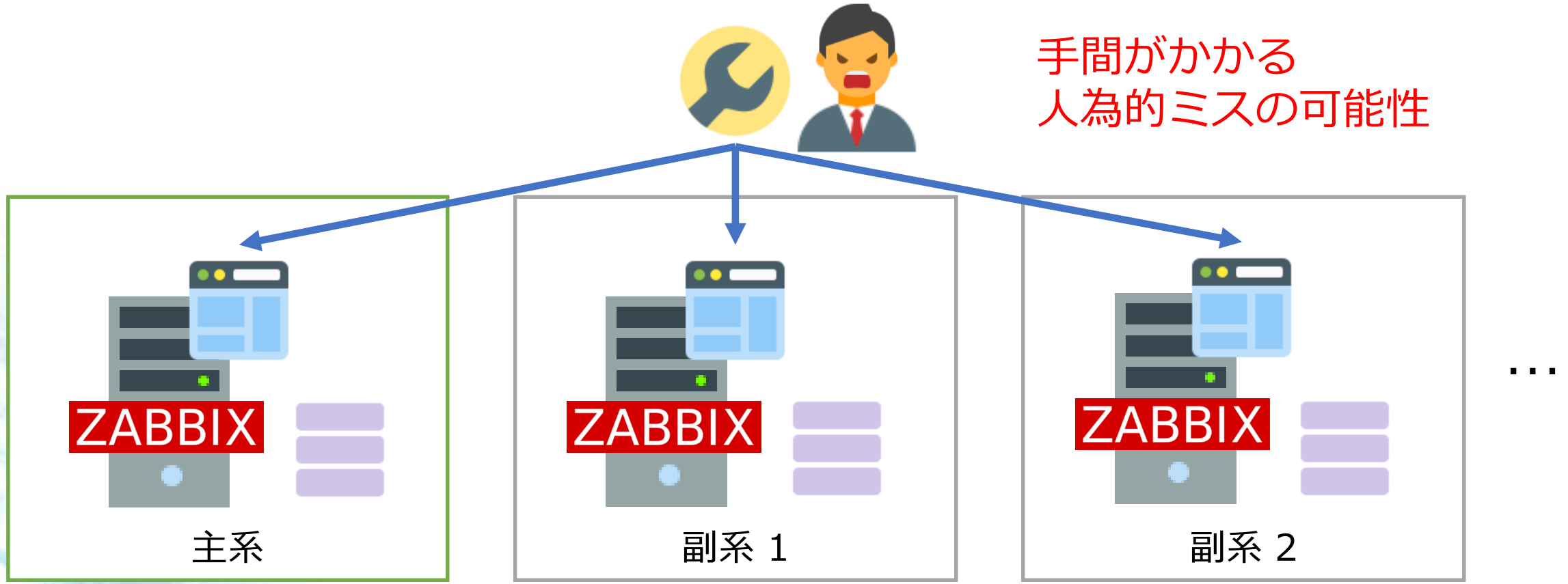
- 全て手動で各 Web インタフェースから同一の設定を実施



© 2022 SRA OSS, Inc. Japan

設定の同期方法の検討

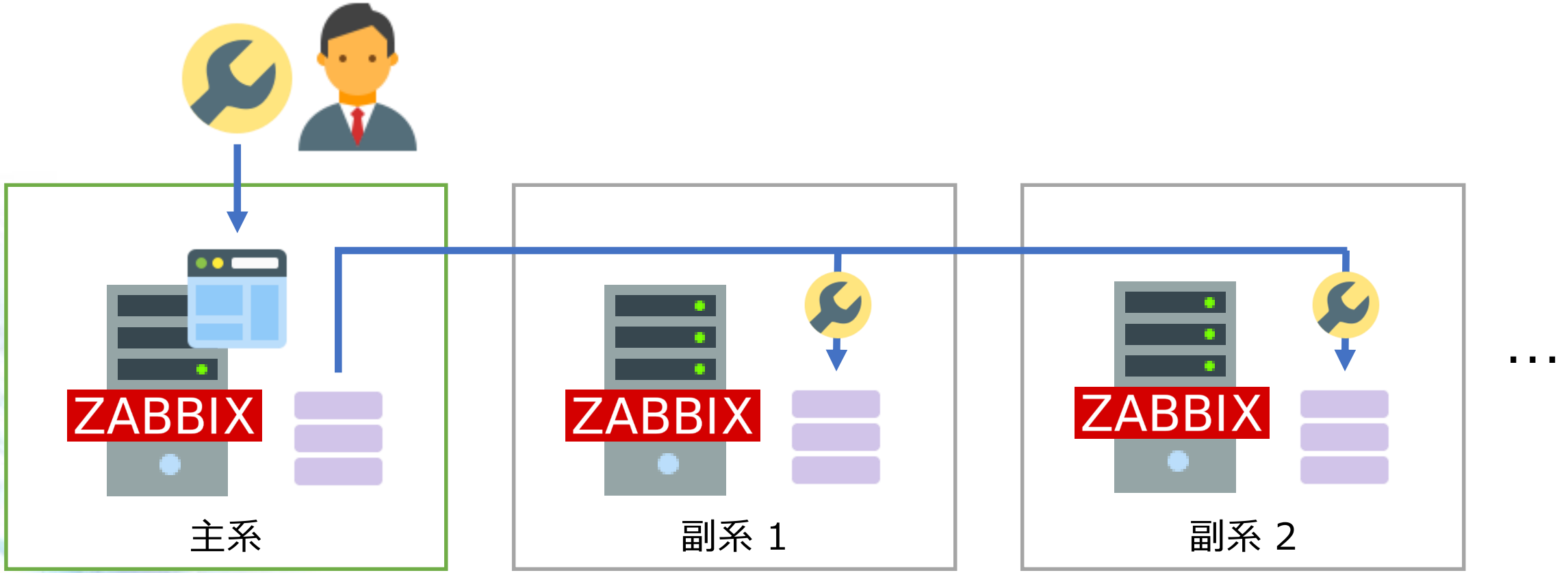
- 全て手動で各 Web インタフェースから同一の設定を実施 △



© 2022 SRA OSS, Inc. Japan

設定の同期方法の検討

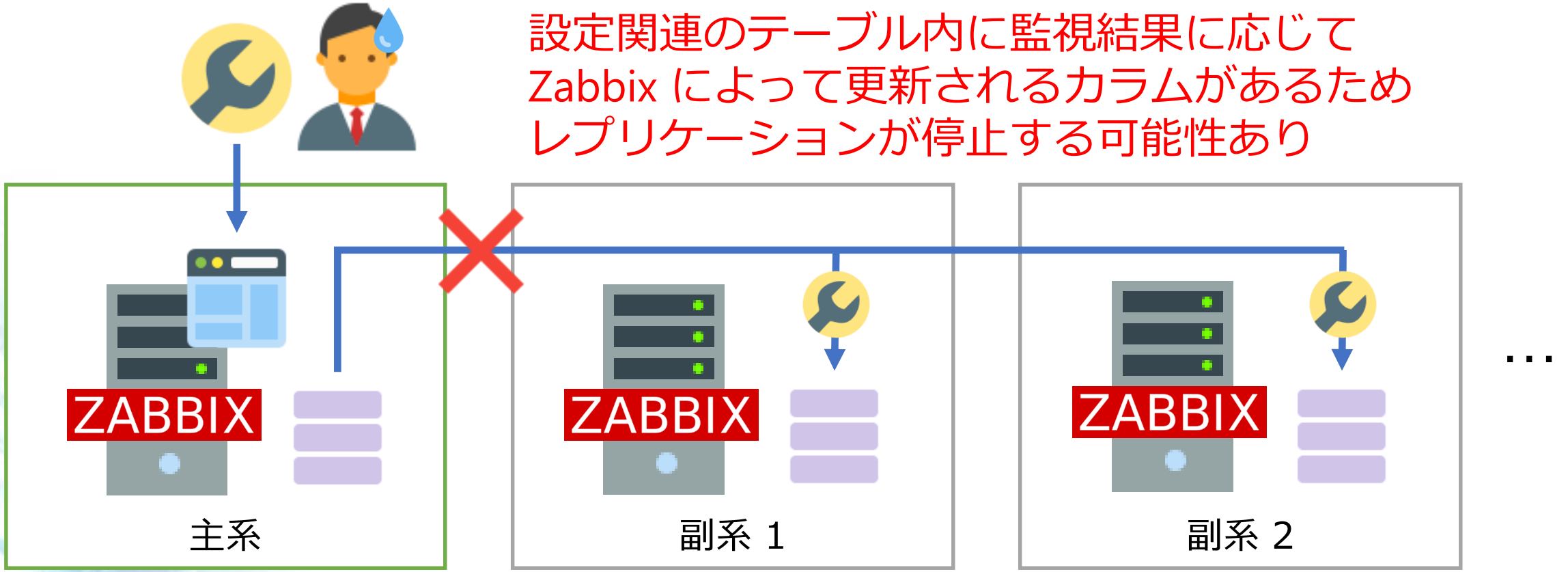
- レプリケーション機能で DB の設定関連テーブルのみを同期



© 2022 SRA OSS, Inc. Japan

設定の同期方法の検討

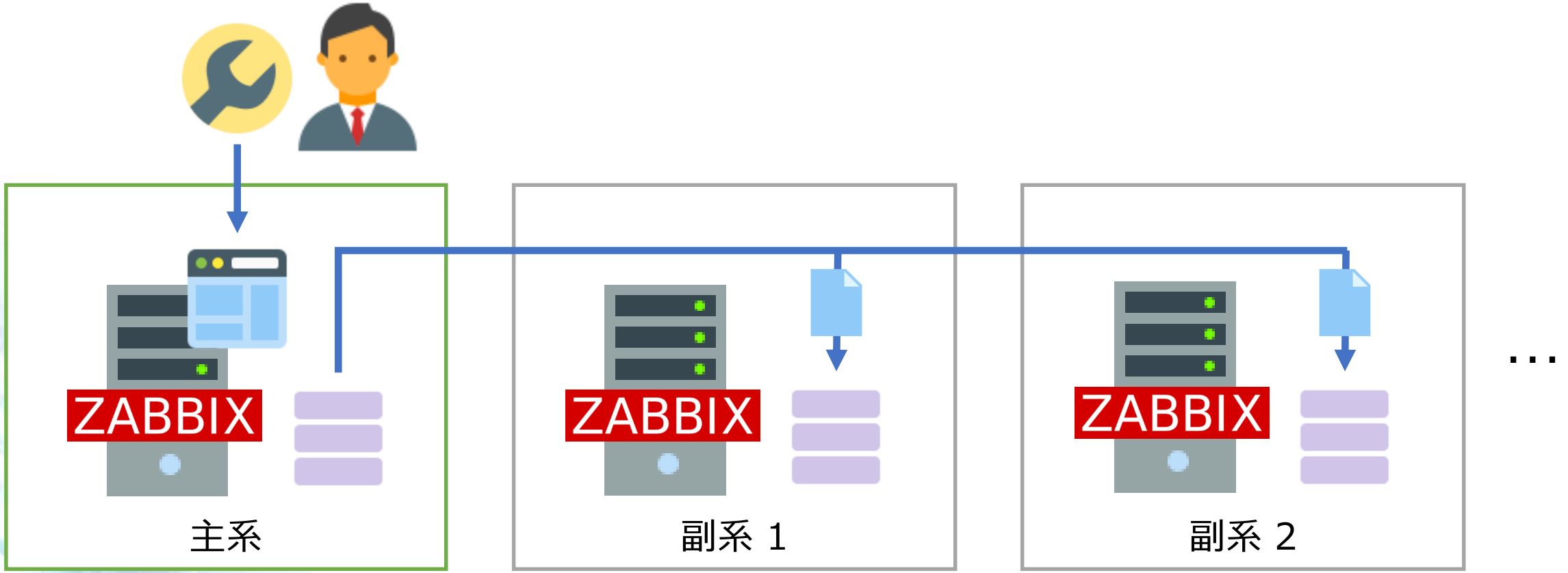
- レプリケーション機能で DB の設定関連テーブルのみを同期 ✕



© 2022 SRA OSS, Inc. Japan

設定の同期方法の検討

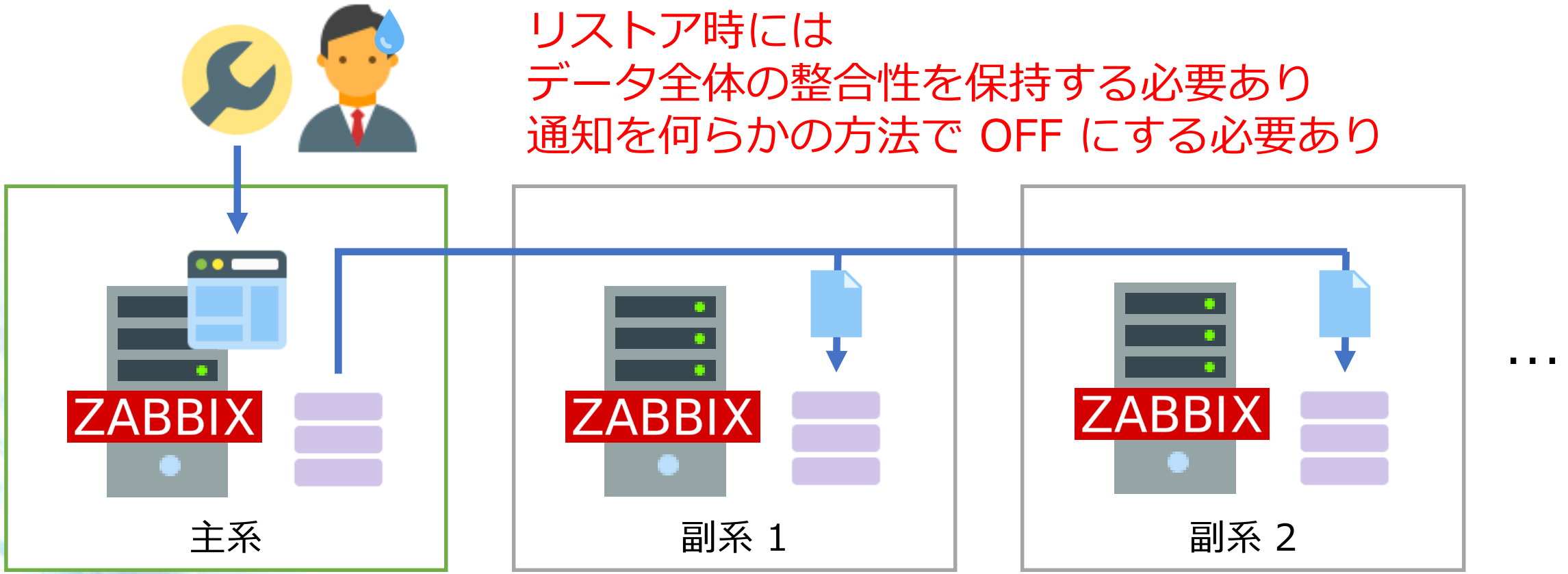
- DB の設定関連テーブルのみをダンプ/リストア



© 2022 SRA OSS, Inc. Japan

設定の同期方法の検討

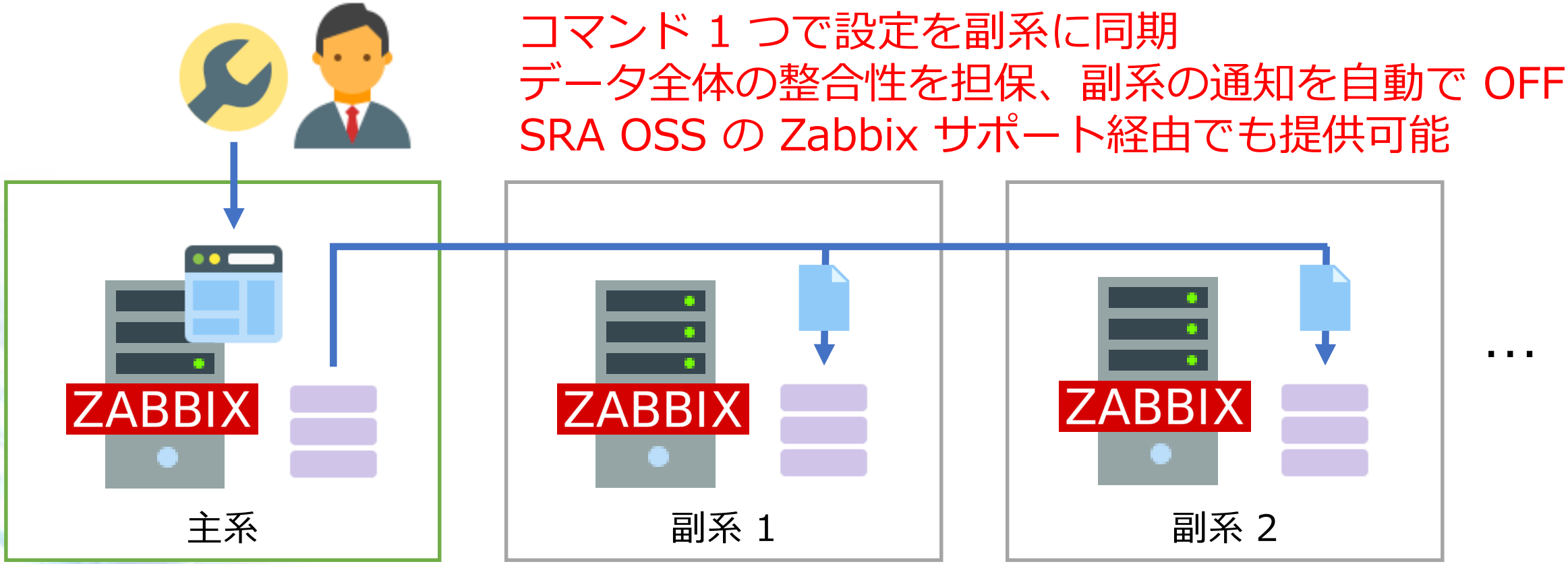
- DB の設定関連テーブルのみをダンプ/リストア △



© 2022 SRA OSS, Inc. Japan

設定の同期方法の検討

- Zabbix 設定バックアップ同期ツール ○



© 2022 SRA OSS, Inc. Japan

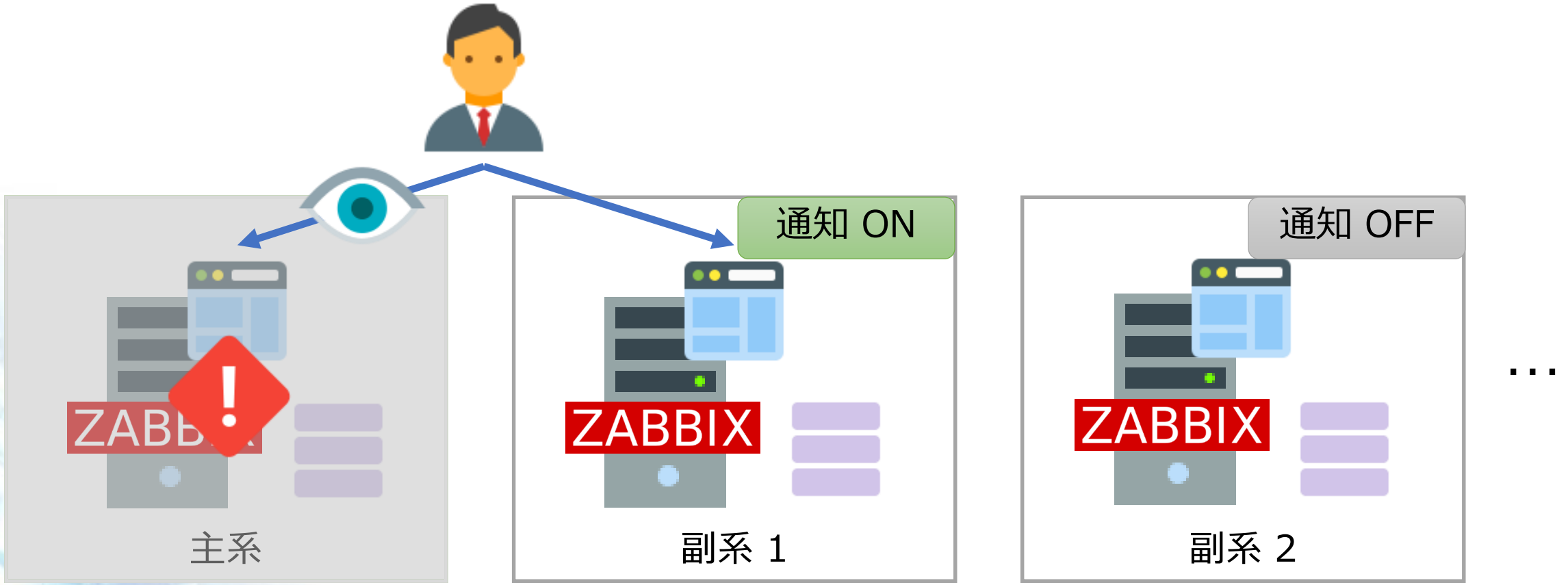
Zabbix 設定バックアップツールの注意点

- 副系では設定を追加・削除できない
  - データの整合性をチェックしているため、リストアできなくなる
- 副系ではディスカバリとエージェントの自動登録は無効化される
  - これもデータの整合性の都合

© 2022 SRA OSS, Inc. Japan

切替方法の検討

- 全て手動で切替を実施

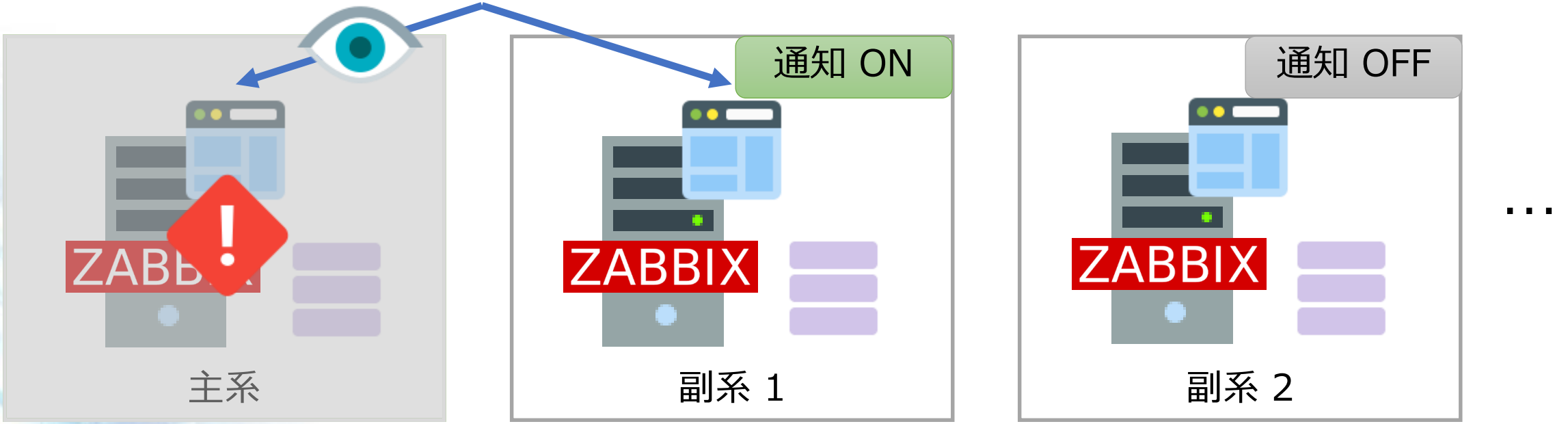


© 2022 SRA OSS, Inc. Japan

切替方法の検討

- 全て手動で切替を実施 △

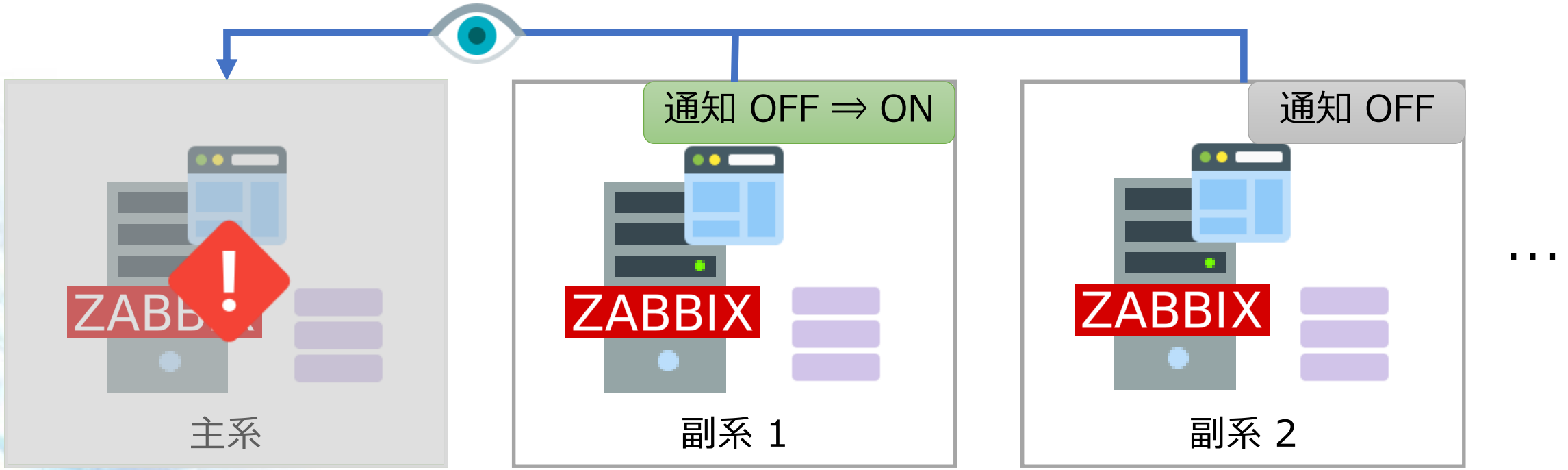
主系の障害検知に遅れると切替まで通知が来ない
常に人が主系を監視するのは困難



© 2022 SRA OSS, Inc. Japan

切替方法の検討

- Zabbix で監視と切替を実施
  - 副系で主系の障害検知時に自身の通知をアクション経由で ON にする

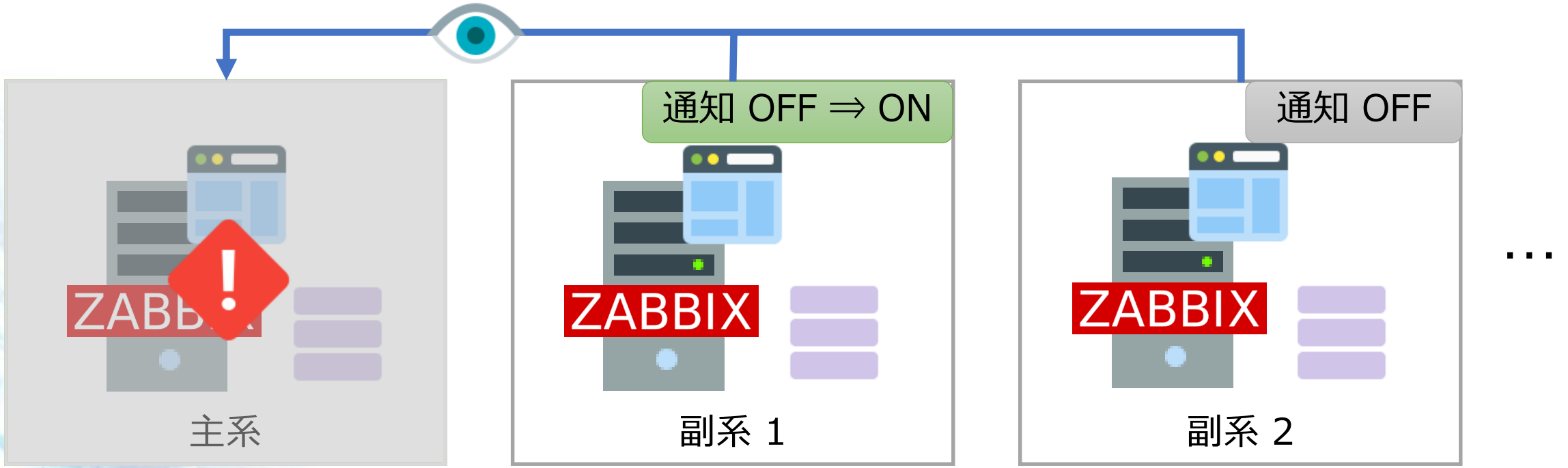


© 2022 SRA OSS, Inc. Japan

切替方法の検討

- Zabbix で監視と切替を実施 △
  - 副系で主系の障害検知時に自身の通知をアクション経由で ON にする

副系で主系監視の通知のみ ON にしておく必要あり



© 2022 SRA OSS, Inc. Japan

Zabbix Active/Active HA クラスタまとめ

長所

- 構築は簡単
- 障害発生時に監視のダウンタイムなし

短所

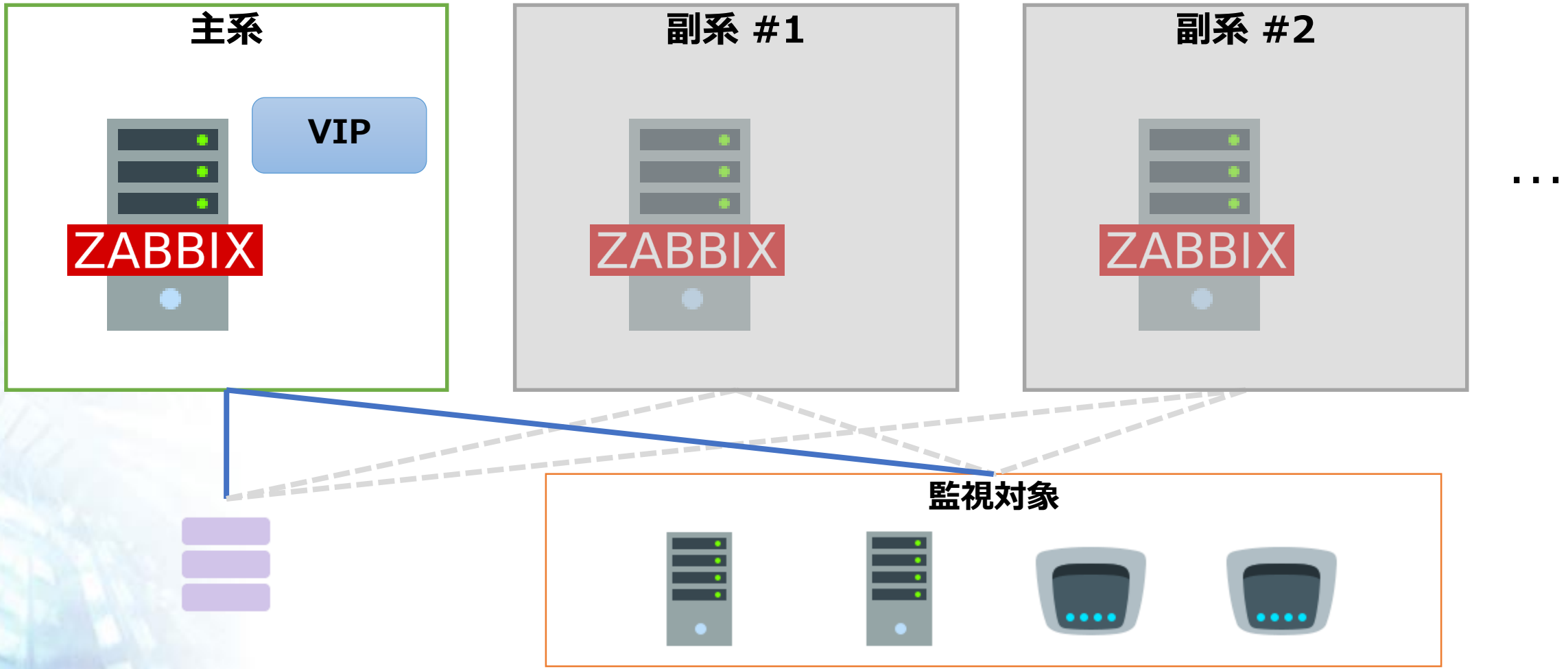
- 監視対象側の負荷は高め
- 設定の同期や切替に要工夫

© 2022 SRA OSS, Inc. Japan

Zabbix Active/Standby HA クラスタ

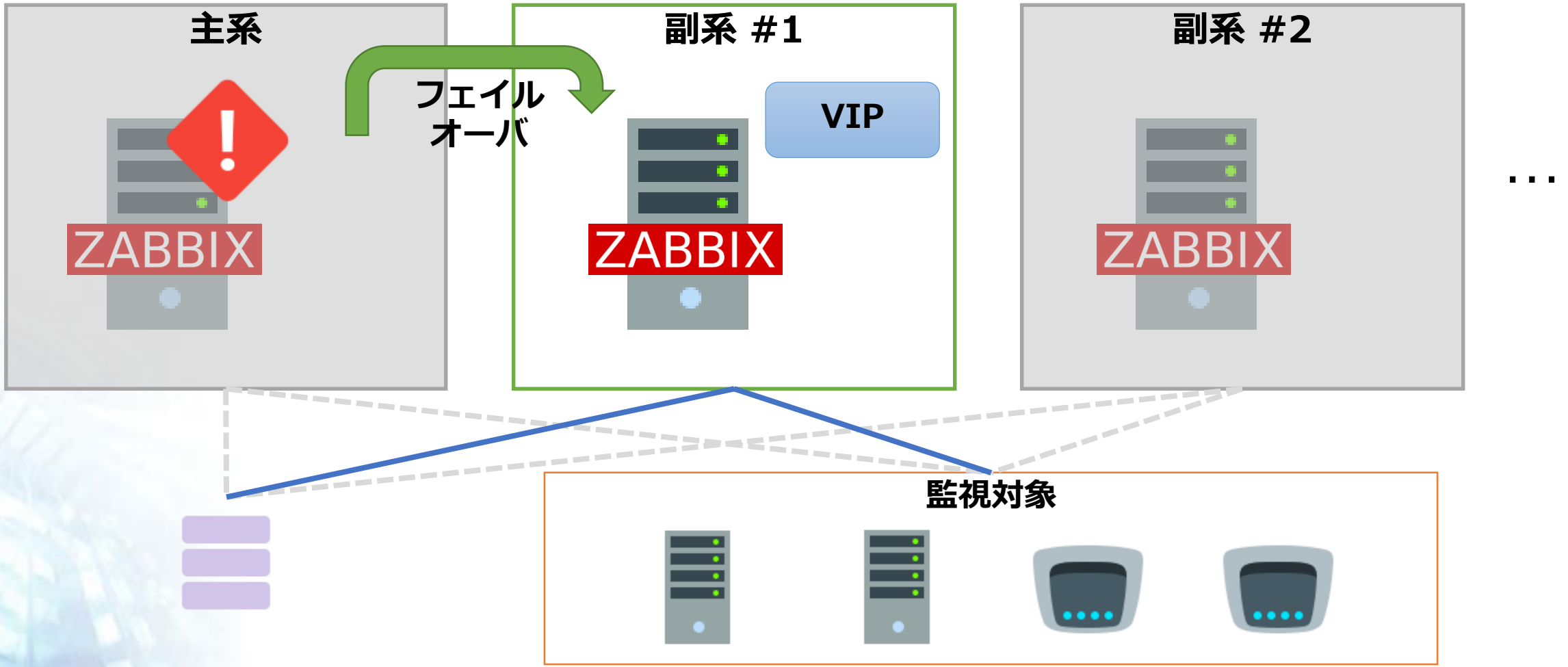
© 2022 SRA OSS, Inc. Japan

Zabbix Active/Standby HA クラスタ



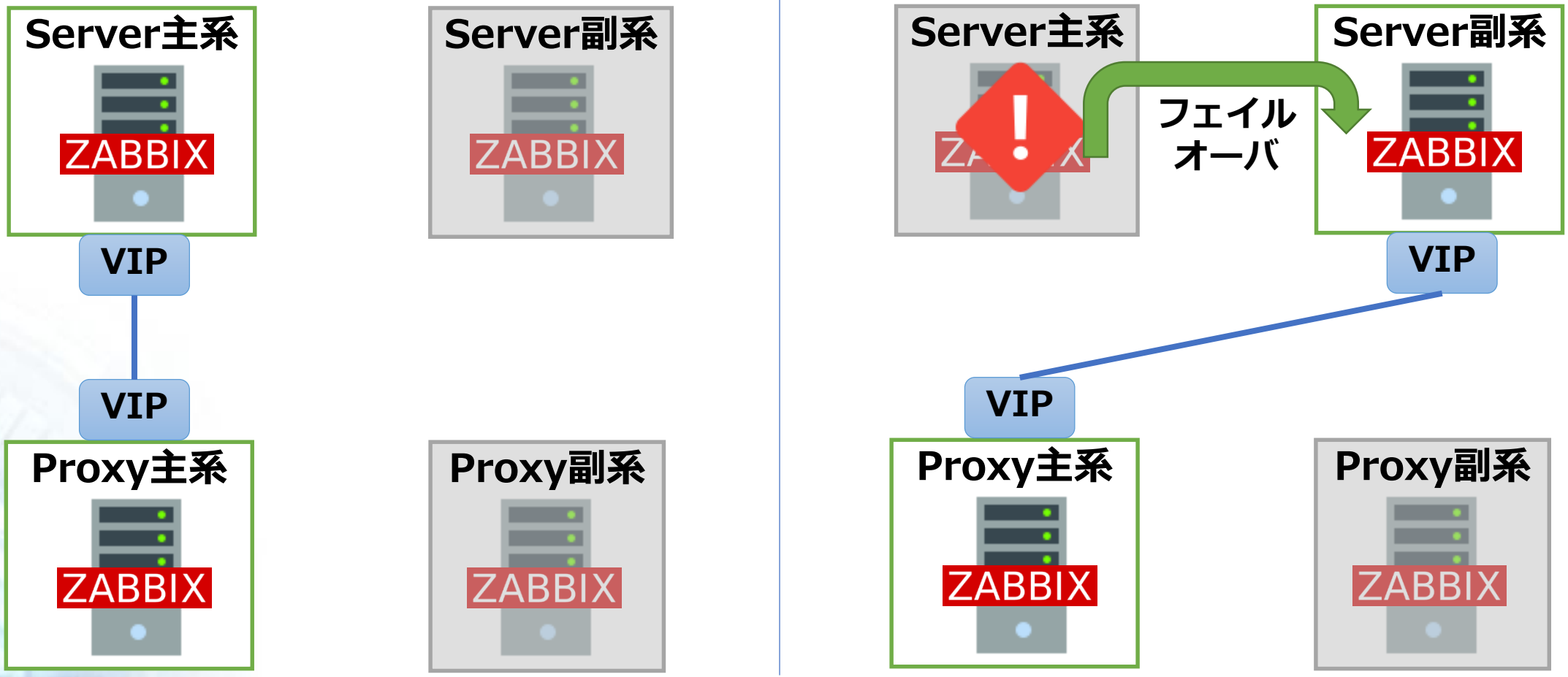
© 2022 SRA OSS, Inc. Japan

Zabbix Active/Standby HA クラスタ



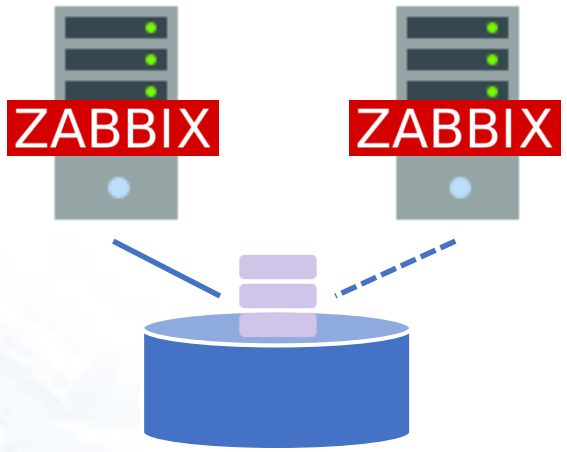
© 2022 SRA OSS, Inc. Japan

Zabbix proxy を含む Active/Standby 構成

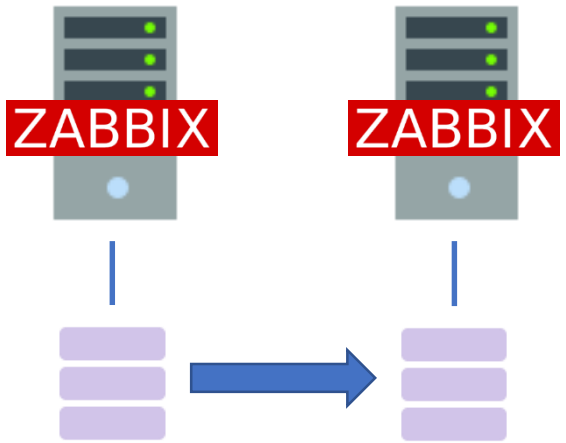


© 2022 SRA OSS, Inc. Japan

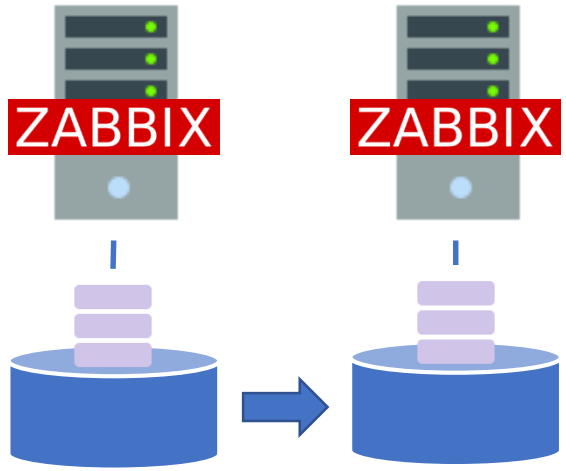
Active/Standby DB 構成手法



共有ストレージ



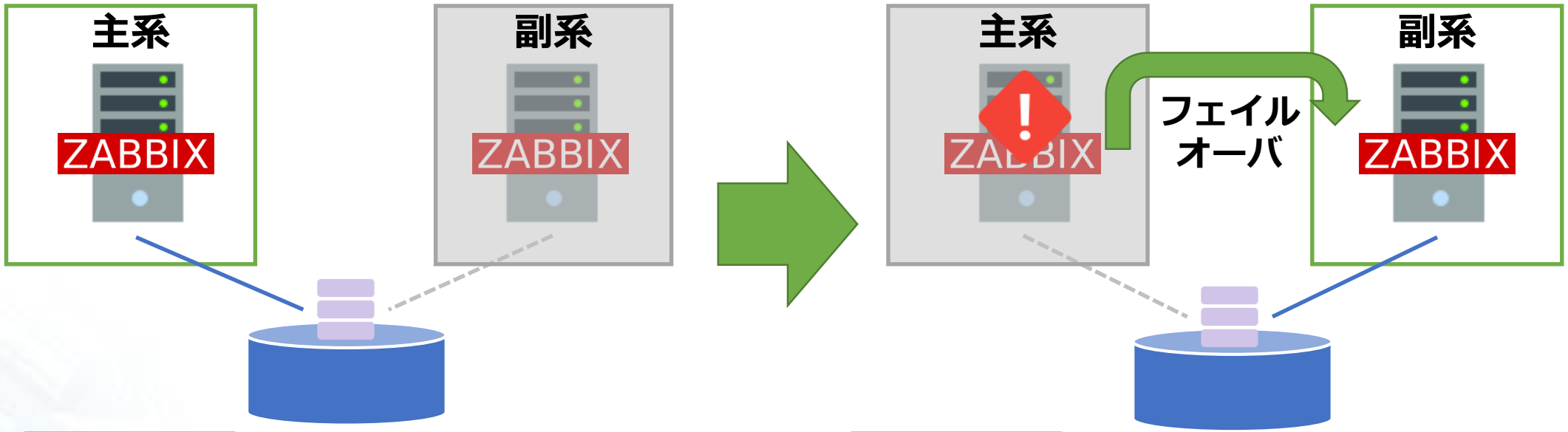
DB レプリケーション



ブロックデバイス
レプリケーション

© 2022 SRA OSS, Inc. Japan

共有ストレージによる構成



長所

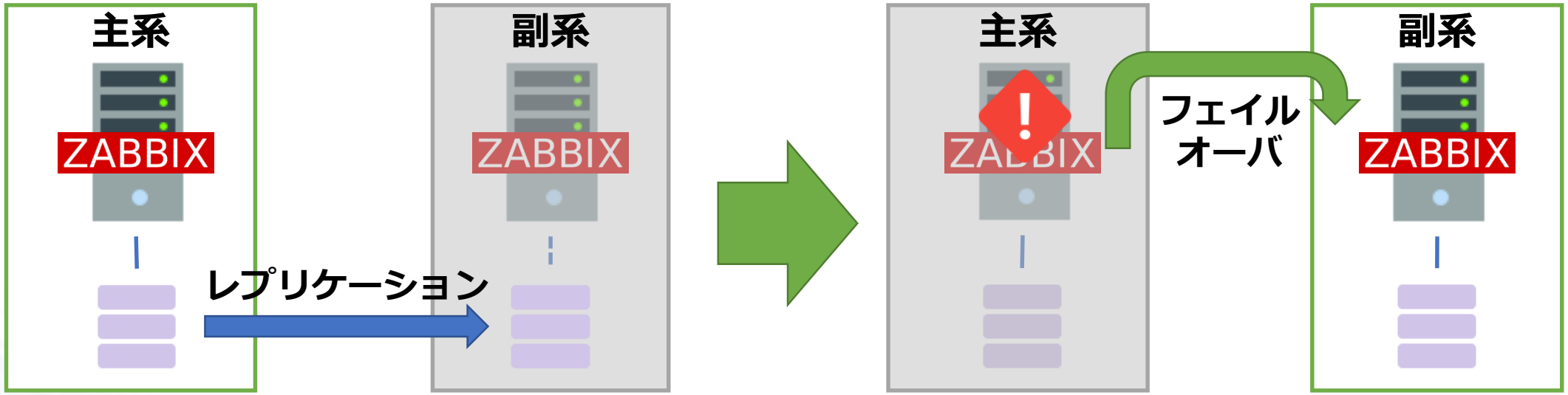
- データ同期のオーバヘッドなし
- データの不整合発生低

短所

- 共有ストレージの費用コスト高
- ストレージが単一障害点

© 2022 SRA OSS, Inc. Japan

DB レプリケーションによる構成



長所

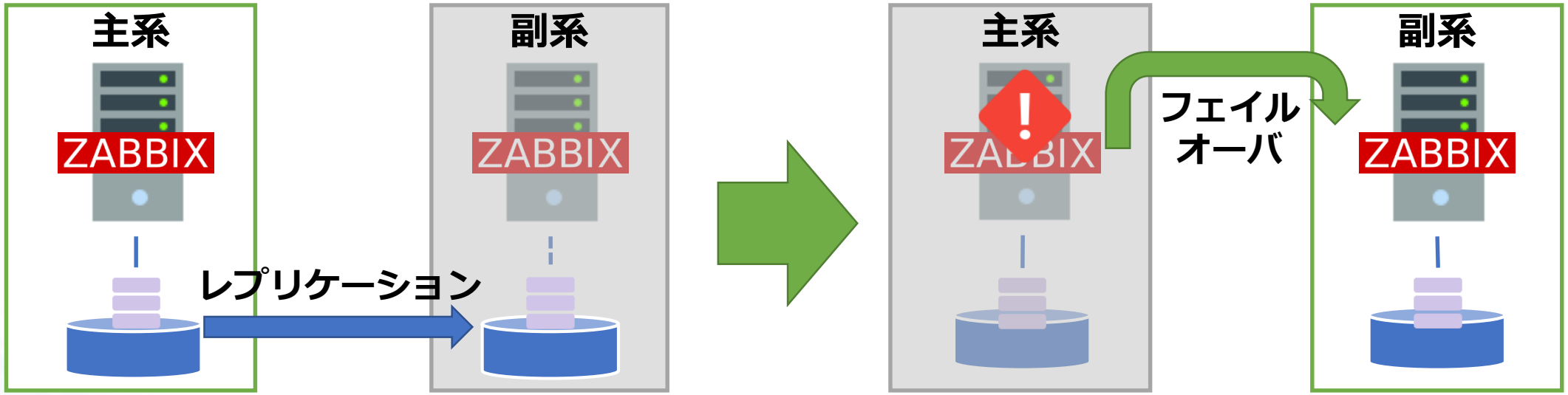
- 導入費用コスト低
- 構成がシンプル
- フェイルオーバ速め

短所

- 導入技術コスト高
- レプリケーション遅延の考慮が必要

© 2022 SRA OSS, Inc. Japan

ブロックデバイスレプリケーションによる構成



長所

- 導入費用コスト低
- レプリケーション遅延の考慮不要
- データロストのリスクなし

短所

- 導入技術コスト高
- 管理コスト高

© 2022 SRA OSS, Inc. Japan

Active/Standby クラスタの考慮点

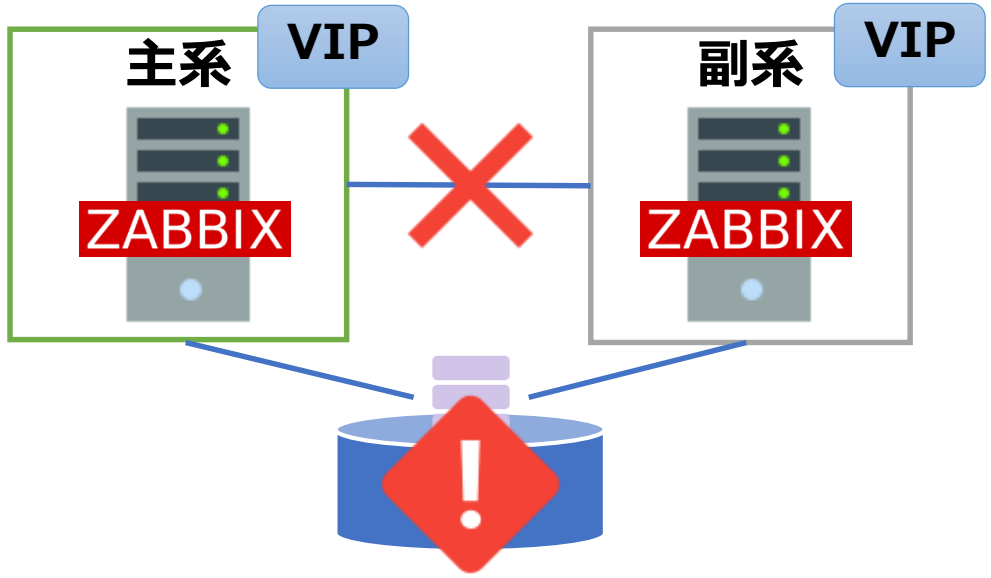
- スプリットブレイン対策
  - STONITH (Shoot The Other Node In The Head)
  - その他の排他制御機構

- Zabbix サーバ上のログ監視や SNMP トラップ監視
  - フェイルオーバ時に再読み込みの可能性あり
  - ログファイルや SNMP トラップファイルは共有ストレージに

© 2022 SRA OSS, Inc. Japan

スプリットブレイン

- ノード間のネットワーク分断で
  両系でサービスが起動



サービスへのアクセス不能
データベースの破壊

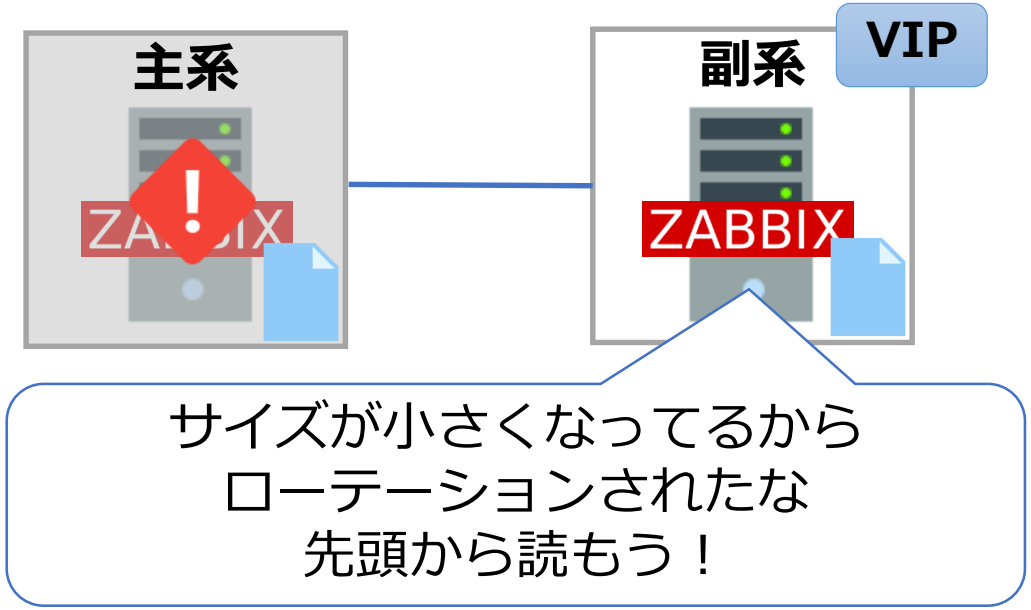
STONITH や共有ディスクの排他制御などで
スプリットブレイン対策を！

© 2022 SRA OSS, Inc. Japan

ログや SNMP トラップの再読み込み

- ログ監視や SNMP トラップは
  ローテーション対応のため
  ファイルサイズや更新日時を記憶

フェイルオーバ時に再読み込み



対象ファイルを共有ディスクや
レプリケーションされるブロックデバイス上に

© 2022 SRA OSS, Inc. Japan

Zabbix Active/Standby HA クラスタまとめ

長所

- 障害時はクラスタソフトで自動切替が可能
- 環境に応じた構成を選択可能

短所

- 構築はやや複雑
- 障害時に短時間の監視のダウンタイムあり

© 2022 SRA OSS, Inc. Japan

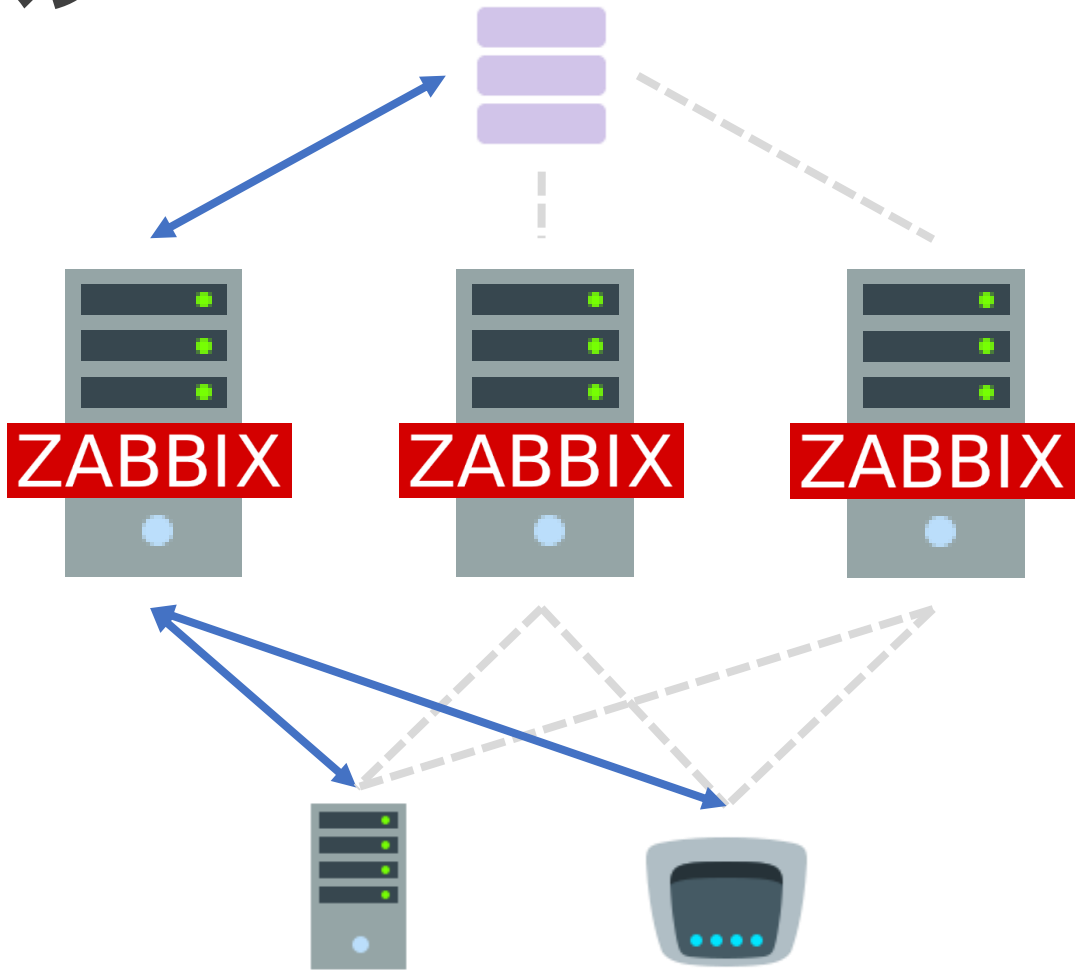
Zabbix server の HA クラスタ機能

© 2022 SRA OSS, Inc. Japan

Zabbix server の HA クラスタ

Zabbix 6.0 からサードパーティ製の
クラスタソフトウェアなしで
Zabbix server の HA クラスタを
構築できる機能が追加

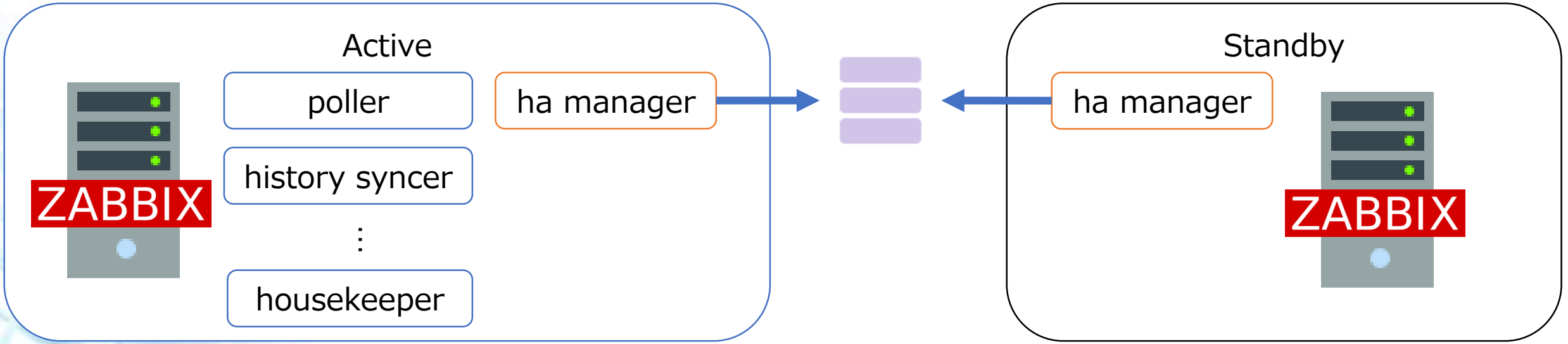
- Active/Standby
- 1 つの DB を共有
- マイナーバージョン間での互換性あり



© 2022 SRA OSS, Inc. Japan

HA クラスタの仕組み

- ha manager プロセスがクラスタを制御
  - アクティブ状態の Zabbix server では、全ての子プロセスが起動
  - スタンバイ状態の Zabbix server では、ha manager のみが起動
  - ha manager は DB にハートビートを送信



© 2022 SRA OSS, Inc. Japan

ha manager プロセス

- 自ノードの情報を DB に登録・更新

```
zabbix=# select * from ha_node;
        ha_nodeid         |       name       |     address     | port  | lastaccess | status |       ha_sessionid
--------------------------+------------------+-----------------+-------+------------+--------+--------------------------
 ckvupwi8n0001rwnxvdlwfv0s | Zabbix server #3 | 133.137.175.151 | 10051 | 1636696278 |      0 | ckvvx4pae0000f9nxezi9hczg
 ckvupsmeq0001egpvksl6c5ih | Zabbix server #2 | 133.137.175.28  | 10051 | 1636710640 |      1 | ckvupys780000temnjy04t2pb
 ckvupihk70001z8mkpw5cg0u3 | Zabbix server #1 | 133.137.175.61  | 10051 | 1646708964 |      3 | cl08vj7000000r6mkoj1ie0k3
```

- 他ノードの状況を監視
  - Active ノードがいない or 一定時間ハートビートが確認できない
    ⇒ 自ノードを Active としてフェイルオーバ

© 2022 SRA OSS, Inc. Japan

フェイルオーバ

- 全ノードが 5 秒おきにハートビートを送信
  - アクティブノードに異常（DB 上でハートビート未受信）があると、
    最初に検知したスタンバイノードが引き継ぐ (どのノードが引き継ぐかは指定不可)

- Active ノードが更新されない場合
  - フェイルオーバ遅延時間 (デフォルト1分) に達するまで待機
  - 遅延時間経過後にいずれかのスタンバイノードが引き継ぐ

- フェイルオーバ遅延時間は、ランタイムコマンドから設定

```
# zabbix_server -R ha_set_failover_delay=2m
```

© 2022 SRA OSS, Inc. Japan

HA クラスタの設定 – Zabbix server

- /etc/zabbix/zabbix_server.conf
  - HANodeName にノード名を設定
    - このパラメータが設定されてると Zabbix server は HA モードになる

```
HANodeName=Zabbix server #1
```

  - NodeAddress にノードアドレスを設定
    - フロントエンドがアクティブノードを判断するのに利用

```
NodeAddress=133.137.175.61
```

  - DBHost に同じ DB を設定
    - 全ノードに同一の DB を参照させる

© 2022 SRA OSS, Inc. Japan

HA クラスタの設定 – フロントエンド

- /etc/zabbix/web/zabbix.conf.php
  - $ZBX_SERVER と $ZBX_SERVER_PORT をコメントアウト

```
// $ZBX_SERVER                  = '';
// $ZBX_SERVER_PORT             = '';
```

現在のアクティブノードの IP アドレス、ポートは DB から取得
Zabbix server ごとにフロントエンドを用意する必要なし

© 2022 SRA OSS, Inc. Japan

HA クラスタの設定 – Zabbix agent

- /etc/zabbix/zabbix_agent.conf
  - パッシブ
    - Server パラメータに全ノードのアドレスを**カンマ区切り**で設定

```
Server=133.137.175.61,133.137.175.28,133.137.175.151
```

  - アクティブ
    - ServerActive パラメータに全ノードのアドレスを**セミコロン区切り**で設定
      (Zabbix 6.0 より有効な設定方法)

```
ServerActive=133.137.175.61;133.137.175.28;133.137.175.151
```

© 2022 SRA OSS, Inc. Japan

HA クラスタの設定 – Zabbix proxy

- /etc/zabbix/web/zabbix_proxy.conf
  - パッシブ
    - Server パラメータに全ノードのアドレスを**カンマ区切り**で設定

```
Server=133.137.175.61,133.137.175.28,133.137.175.151
```

  - アクティブ
    - Server パラメータに全ノードのアドレスを**セミコロン区切り**で設定

```
Server=133.137.175.61;133.137.175.28;133.137.175.151
```

© 2022 SRA OSS, Inc. Japan

HA クラスタの状態確認および監視

- 「レポート」 ⇒ 「システム情報」画面
  - HA クラスタの状態と各ノードの状態
- 「システム情報」ウィジェット
  - HA クラスタの状態もしくは各ノードの状態
- ha_status ランタイムコントロールオプション
  - 各ノードの状態（zabbix_server.log に出力）
- zabbix[cluster,discovery,nodes] インターナルアイテム
  - 各ノードの状態（JSON 形式）

© 2022 SRA OSS, Inc. Japan

「レポート」⇒「システム情報」画面

- ノード名
- アドレス
- 最終アクセスからの時間
- ステータス（アクティブ/スタンバイ/停止中/利用不可）

1秒あたりの監視項目数(Zabbixサーバーの要求パフォーマンス)		1.69	
HAクラスター		有効	フェールオーバーの遅延： 1分

名前	アドレス	最終アクセスからの時間	ステータス
Zabbix server #1	133.137.175.61:10051	2s	アクティブ
Zabbix server #3	133.137.175.151:10051	1m 44s	停止中
Zabbix server #2	133.137.175.28:10051	3s	スタンバイ

© 2022 SRA OSS, Inc. Japan

「システム情報」ウィジェット

システムステータス

システム情報

パラメータ	値	詳細
Zabbixサーバーの起動	はい	133.137.175.61:10051
ホスト数 (有効/無効)	5	5 / 0
テンプレート数	290	
アイテム数 (有効/無効/取得不可)	136	127 / 0 / 9
トリガー数 (有効/無効 [障害/正常])	71	71 / 0 [6 / 65]
ユーザー数 (オンライン)	3	1
1秒あたりの監視項目数(Zabbixサーバーの要求パフォーマンス)	2.72	
HAクラスター	有効	フェールオーバーの遅延： 1分

HA ノード

システム情報

名前	アドレス	最終アクセスからの時間	ステータス
Zabbix server #1	133.137.175.61:10051	4s	アクティブ
Zabbix server #3	133.137.175.151:10051	2h 50m 37s	停止中
Zabbix server #2	133.137.175.28:10051	3s	スタンバイ

フェールオーバーの遅延： 1分

© 2022 SRA OSS, Inc. Japan

ha_status ランタイムコントロールオプション

- 標準出力と zabbix_server.log に各ノードの状態を出力

```
# zabbix_server -R ha_status
Failover delay: 60 seconds
Cluster status:
#  ID                          Name              Address                 Status   Last Access
1. ckvupihk70001z8mkpw5cg0u3   Zabbix server #1  133.137.175.61:10051    active   0s
2. ckvupsmeq0001egpvksl6c5ih   Zabbix server #2  133.137.175.28:10051    standby  1s
3. ckvupwi8n0001rwnxvdlwfv0s   Zabbix server #3  133.137.175.151:10051   stopped  8m 12s
```

© 2022 SRA OSS, Inc. Japan

zabbix[cluster,discovery,nodes]

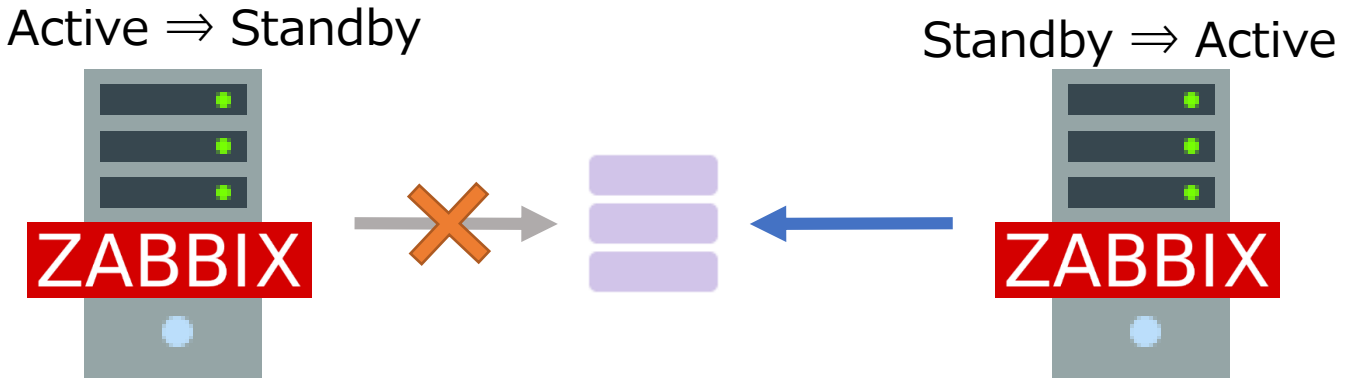
- 各ノードの情報を JSON 形式で取得
  - id
  - name
  - status
    - 0: standby
    - 1: stopped
    - 2: unavailable
    - 3: active
  - lastaccess (最終アクセス時刻 Unixtime)
  - address
  - db_timestamp（現在時刻 Unixtime）
  - lastaccess_age（最終アクセスからの経過秒）

```
[
  {
    "id":"ckvupihk70001z8mkpw5cg0u3",
    "name":"Zabbix server #1",
    "status":3,
    "lastaccess":1636623048,
    "address":"133.137.175.61:10051",
    "db_timestamp":1636623050,
    "lastaccess_age":2
  },
  ...
]
```

© 2022 SRA OSS, Inc. Japan

スプリットブレイン

- 各 Zabbix server ノードは DB のみと通信
- DB との接続が切れた場合には Standby に自動切替

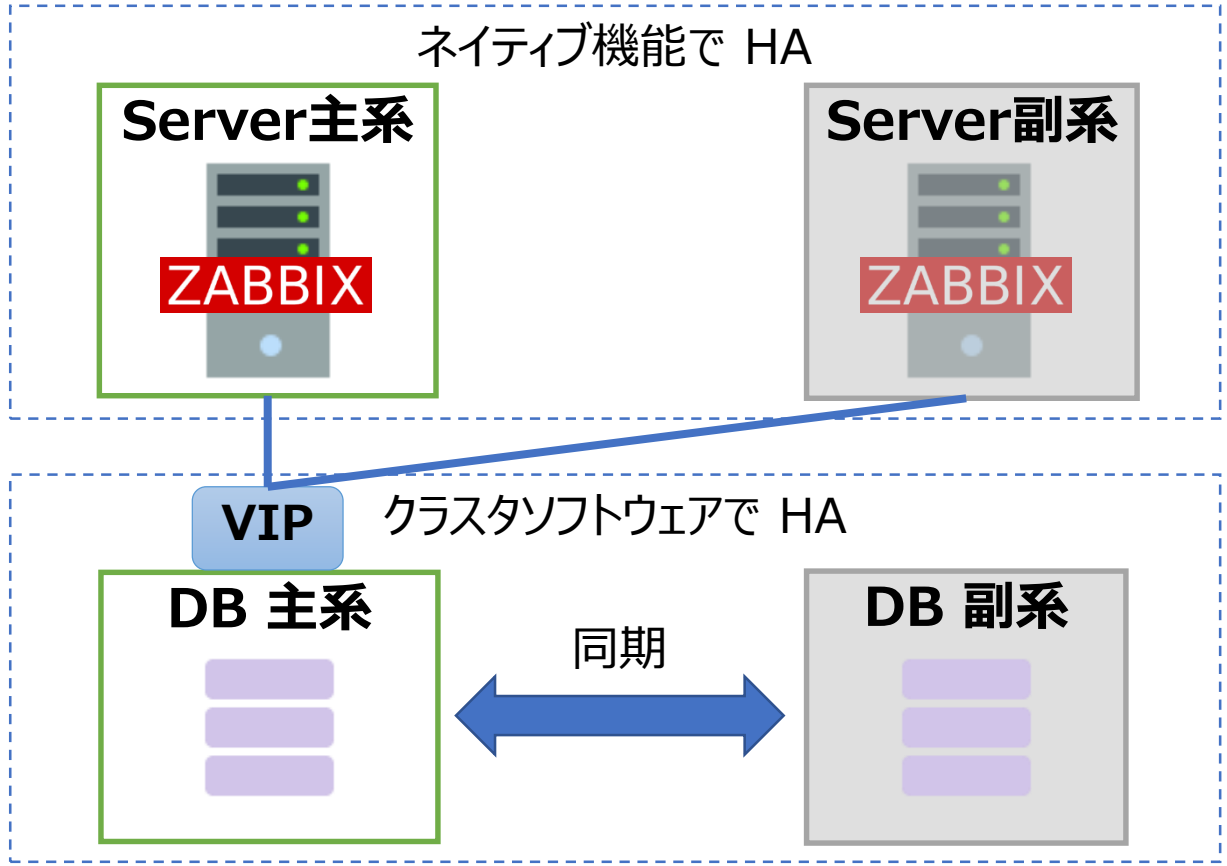


基本的にスプリットブレインが発生する心配なし

© 2022 SRA OSS, Inc. Japan

DB のクラスタ化

- Zabbix server の利用するデータベースの HA は含まれない
  - 従来通りサードパーティ製の
    クラスタソフトウェアなどを利用する
  - クラウドのマネージドサービスを利用



© 2022 SRA OSS, Inc. Japan

Zabbix server の HA クラスタ機能の注意点

- SNMP などは全ての Zabbix server ノードと通信できるように設定
- ファイアウォールも全ての Zabbix server ノードと通信できるように設定
- Zabbix proxy の HA 機能は Zabbix 6.4 で実装予定

© 2022 SRA OSS, Inc. Japan

オープンソースとともに

© 2022 SRA OSS, Inc. Japan

Appendix

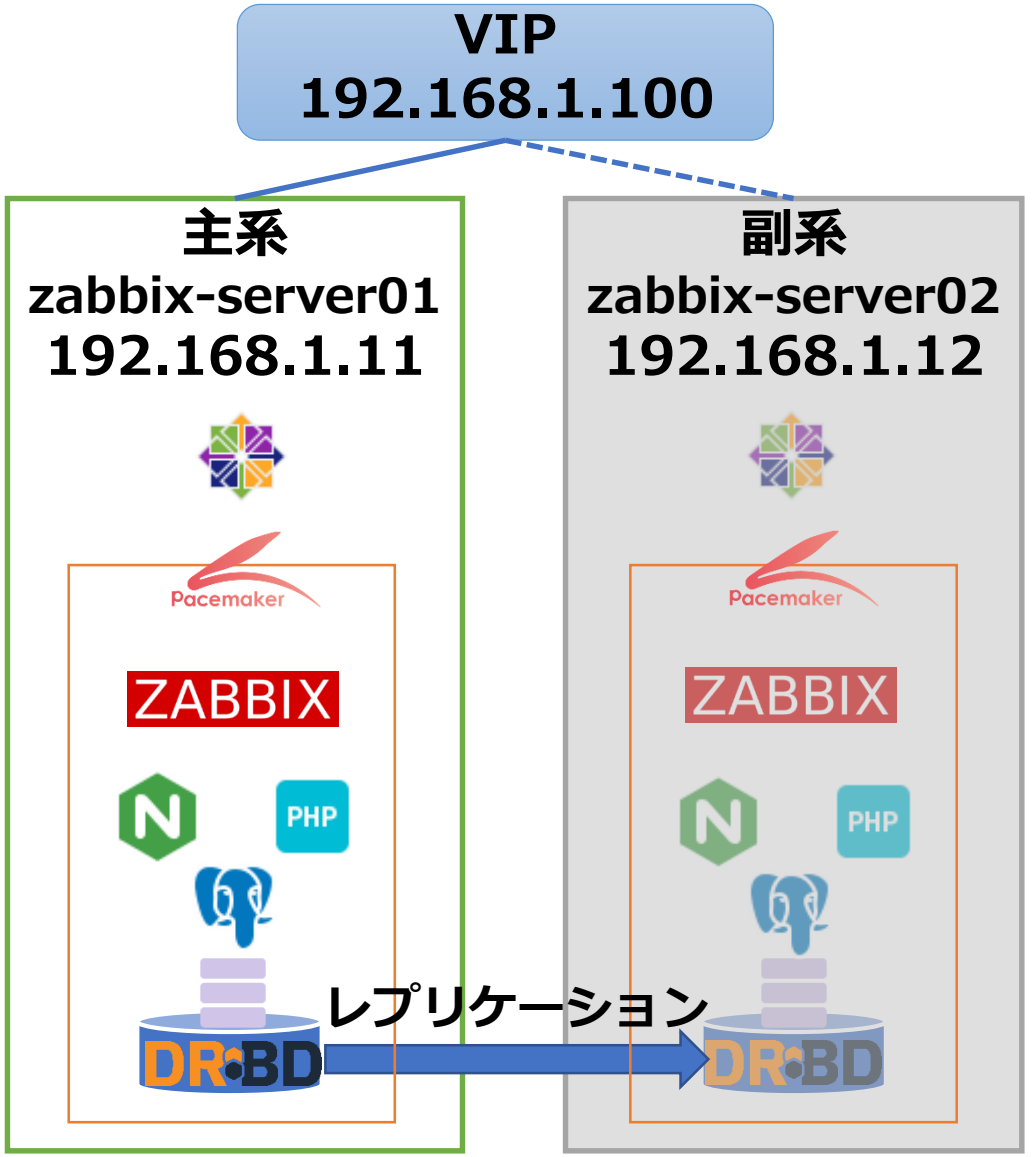
© 2022 SRA OSS, Inc. Japan

Pacemaker/Corosync/DRBD を利用した Zabbix Active/Standby クラスタ 設定例

© 2022 SRA OSS, Inc. Japan

環境

- OS: CentOS 8
- Software:
  - Zabbix 6.0.4
  - PostgreSQL 14
  - Web サーバソフトウェア
    - Nginx
    - PHP-FPM
  - クラスタソフトウェア
    - Pacemaker
    - Corosync
  - ブロックデバイスレプリケーション
    - DRBD



© 2022 SRA OSS, Inc. Japan

Zabbix のインストールおよび設定

- 両ノードで Zabbix のレポジトリをインストール

```
# dnf install https://repo.zabbix.com/zabbix/6.0/rhel/8/x86_64/zabbix-release-6.0-1.el8.noarch.rpm
# dnf clean all
```

- 両ノードで Zabbix サーバとフロンエンドをインストール

```
# dnf install zabbix-server-pgsql zabbix-web-pgsql zabbix-nginx-conf
# systemctl disable zabbix-server
```

- 両ノードで Zabbix サーバの SourceIP パラメータを VIP に設定

```
# vi /etc/zabbix/zabbix_server.conf
SourceIP=192.168.1.100
```

© 2022 SRA OSS, Inc. Japan

Nginx および PHP-FPM の設定

- 両ノードで Nginx と PHP-FPM をインストール

```
# dnf install nginx php-fpm
# systemctl disable nginx
# systemctl disable php-fpm
```

- 両ノードで Nginx の設定ファイルを編集

```
# vi /etc/nginx/conf.d/zabbix.conf
listen 80;
server_name 192.168.1.100;
```

- 両ノードで PHP-FPM の設定ファイルを編集

```
# vi /etc/php-fpm.d/zabbix.conf
php_value[date.timezone] = <your timezone>
```

© 2022 SRA OSS, Inc. Japan

Pacemaker、Corosync のインストール・設定

- 両ノードで Pacemaker および Corosync をインストール

```
# dnf --enablerepo=HighAvailability install pacemaker corosync pcs
# systemctl start pcsd
# systemctl enable pcsd
```

- 両ノードでホスト名と IP アドレスの紐づけを設定

```
# vi /etc/hosts
192.168.1.11 zabbix-server01
192.168.1.12 zabbix-server02
```

© 2022 SRA OSS, Inc. Japan

Pacemaker、Corosync の設定

- クラスタノードを承認

```
[zabbix-server01] # passwd hacluster
[zabbix-server01] # pcs host auth ¥
> zabbix-server01 zabbix-server02 ¥
> -u hacluster
Password: <hacluster’s password>
```

- クラスタをセットアップ

```
[zabbix-server01] # pcs cluster setup zabbix-cluster ¥
> zabbix-server01 zabbix-server02
```

© 2022 SRA OSS, Inc. Japan

Pacemaker、Corosync の設定

- クラスタを起動

```
[zabbix-server01] # pcs cluster start --all
[zabbix-server01] # pcs cluster enable --all
```

- STONITH およびクォーラムポリシーを無効化

```
[zabbix-server01] # pcs property set stonith-enabled=false
[zabbix-server01] # pcs property set no-quorum-policy=ignore
```

© 2022 SRA OSS, Inc. Japan

Pacemaker、Corosync の設定

- クラスタの状態チェック

```
[zabbix-server01] # pcs cluster status
Cluster Status:
 Cluster Summary:
   * Stack: corosync
   * Current DC: zabbix-server01 (version 2.0.3-5.el8_2.1-4b1f869f0f) - partition with quorum
   * Last updated: Wed Oct 14 14:01:44 2020
   * Last change:  Wed Oct 14 14:00:30 2020 by hacluster via crmd on zabbix-server01
   * 2 nodes configured
   * 0 resource instances configured
 Node List:
   * Online: [ zabbix-server01 zabbix-server02 ]

PCSD Status:
  zabbix-server01: Online
  zabbix-server02: Online
```

© 2022 SRA OSS, Inc. Japan

DRBD のインストール

- 両ノードで DRBD をインストール

```
# dnf install elrepo-release
# dnf install kmod-drbd90 drbd90-utils
# systemctl enable drbd
```

© 2022 SRA OSS, Inc. Japan

DRBD の設定

- 両ノードで DRBD リソースを設定

```
# vi /etc/drbd.d/drbd0.res
resource drbd0 {
        protocol C;
        disk /dev/sdb1;
        device /dev/drbd0;
        meta-disk internal;
        on zabbix-server01 {
                address 192.168.1.1:7789;
        }
        on zabbix-server02 {
                address 192.168.1.2:7789;
        }
}
```

© 2022 SRA OSS, Inc. Japan

DRBD の設定

- DRBD のメタデータを作成

```
[zabbix-server01] # drbdadm create-md drbd0
[zabbix-server02] # drbdadm create-md drbd0
```

- DRBD を起動

```
[zabbix-server01] # drbdadm up drbd0
[zabbix-server02] # drbdadm up drbd0
```

© 2022 SRA OSS, Inc. Japan

DRBD の設定

- DRBD の状態チェック

```
[zabbix-server01] # drbdadm status drbd0
drbd0 role:Secondary
  disk:Inconsistent
  zabbix-server02 role:Secondary
    peer-disk:Inconsistent
```

© 2022 SRA OSS, Inc. Japan

DRBD の設定

- DRBD の同期

```
[zabbix-server01] # drbdadm primary --force drbd0
[zabbix-server01] # drbdadm status drbd0
drbd0 role:Primary
  disk:UpToDate
  zabbix-server02 role:Secondary
    peer-disk:UpToDate

[zabbix-server01] # drbdadm secondary drbd0
[zabbix-server01] # drbdadm status drbd0
drbd0 role:Secondary
  disk:UpToDate
  zabbix-server02 role:Secondary
    peer-disk:UpToDate
```

© 2022 SRA OSS, Inc. Japan

DRBD の設定

- ファイルシステムとマウントポイントを作成

```
[zabbix-server01] # mkfs.xfs /dev/drbd0
[zabbix-server01] # mkdir /mnt/drbd
[zabbix-server02] # mkdir /mnt/drbd
```

- ファイルシステムをマウント

```
[zabbix-server01] # mount /dev/drbd0 /mnt/drbd
[zabbix-server01] # drbdadm status drbd0
drbd0 role:Primary
  disk:UpToDate
  zabbix-server02 role:Secondary
    peer-disk:UpToDate
```

© 2022 SRA OSS, Inc. Japan

PostgreSQL のインストールおよび設定

- 両ノードで PostgreSQL をインストール

```
# dnf install postgresql-server
# systemctl disable postgresql
```

- DB のデータディレクトリを作成

```
[zabbix-server01] # mkdir /mnt/drbd/pgdata
[zabbix-server01] # chmod 700 /mnt/drbd/pgdata
[zabbix-server01] # chown postgres:postgres /mnt/drbd/pgdata
```

© 2022 SRA OSS, Inc. Japan

PostgreSQL の設定

- DB のデータディレクトリを初期化および PostgreSQL を起動

```
[zabbix-server01] # sudo -u postgres initdb -D /mnt/drbd/pgdata ¥
> --encoding=utf8 --no-locale
[zabbix-server01] # pg_ctl –D /mnt/drbd/pgdata start
```

- Zabbix DB を作成

```
[zabbix-server01] # sudo -u postgres createuser --pwprompt zabbix
[zabbix-server01] # sudo -u postgres createdb -O zabbix zabbix
[zabbix-server01] # zcat /usr/share/doc/zabbix-server-pgsql/create.sql.gz ¥
> | sudo -u zabbix psql zabbix
```

© 2022 SRA OSS, Inc. Japan

Pacemaker のリソース設定

- ファイルシステムおよび PostgreSQL を Pacemaker にリソース設定

```
[zabbix-server01] # pcs resource create filesystem ocf:heartbeat:Filesystem ¥
> device=/dev/drbd0 directory=/mnt/drbd fstype=xfs ¥
> op monitor interval=10s --group db-group

[zabbix-server01] # pcs resource create pgsql ocf:heartbeat:pgsql ¥
> pgctl=/bin/pg_ctl psql=/bin/psql pgdata=/mnt/drbd/pgdata ¥
> op monitor interval=30s --group db-group
```

© 2022 SRA OSS, Inc. Japan

Pacemaker のリソース設定

- VIP、Nginx、PHP-FPM を Pacemaker にリソース設定

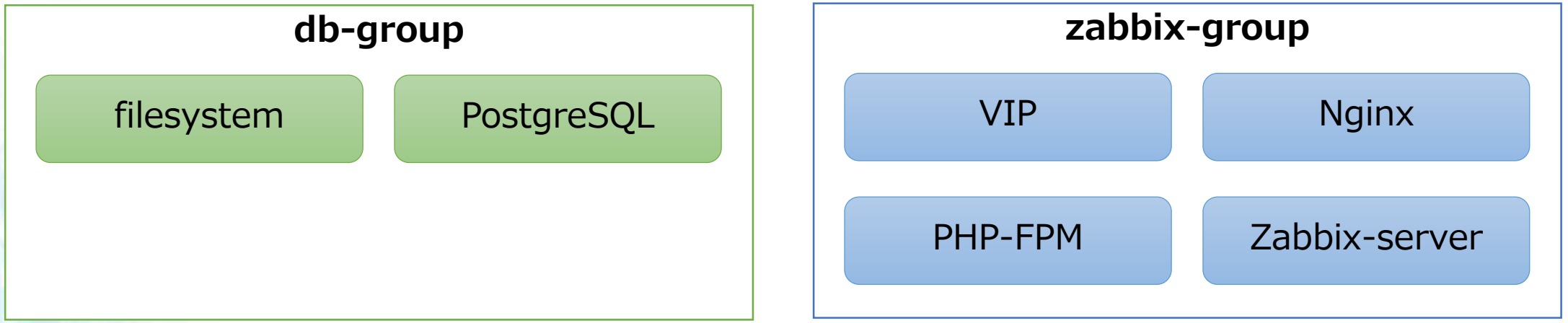
```
[zabbix-server01] # pcs resource create vip ocs:heartbeat:IPaddr2 ¥
> ip=192.168.1.100 cidr_netmask=24 ¥
> op monitor interval=5s --group zabbix-group

[zabbix-server01] # pcs resource create nginx ocf:heartbeat:nginx ¥
> configfile=/etc/nginx/nginx.conf ¥
> op monitor interval=30s --group zabbix-group

[zabbix-server01] # pcs resource create php-fpm systemd:php-fpm ¥
> op monitor interval=30s --group zabbix-group
```

© 2022 SRA OSS, Inc. Japan

Pacemaker のリソース設定

- Zabbix サーバを Pacemaker にリソース設定

```
[zabbix-server01] # pcs resource create zabbix-server systemd:zabbix-server ¥
> op monitor interval=30s --group zabbix-group
```



© 2022 SRA OSS, Inc. Japan

Pacemaker のリソース設定

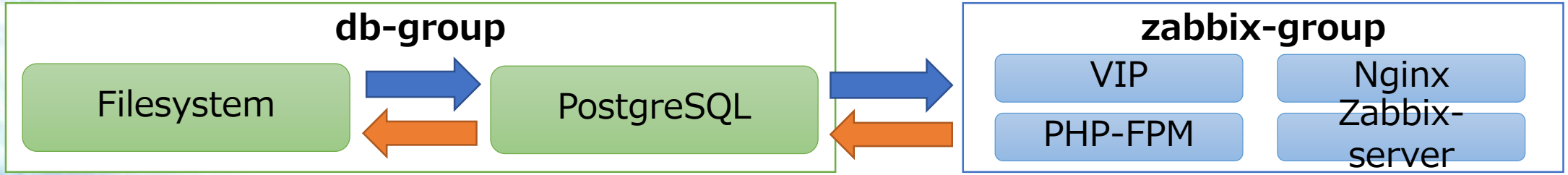
- リソースグループのコロケーション制約を設定

```
[zabbix-server01] # pcs constraint colocation add zabbix-group ¥
> with db-group INFINITY
```

- リソースグループのオーダー制約を設定

```
[zabbix-server01] # pcs constraint order filesystem then start pgsql
[zabbix-server01] # pcs constraint order db-group then start zabbix-group
```

Resource start/stop order



© 2022 SRA OSS, Inc. Japan

Pacemaker のリソース設定

- リソースの状態チェック

```
[zabbix-server01] # pcs status
…
Node List:
  * Online: [ zabbix-server01 zabbix-server02 ]

Full List of Resources:
  * Resource Group: zabbix-group:
    * vip       (ocf::heartbeat:IPaddr2):       Started zabbix-server01
    * nginx     (ocf::heartbeat:nginx): Started zabbix-server01
    * php-fpm   (systemd:php-fpm):      Started zabbix-server01
    * zabbix-server     (systemd:zabbix-server):        Started zabbix-server01
  * Resource Group: db-group:
    * filesystem        (ocf::heartbeat:Filesystem):    Started zabbix-server01
    * pgsql     (ocf::heartbeat:pgsql): Started zabbix-server01
…
```

© 2022 SRA OSS, Inc. Japan

Pacemaker のリソース設定

- リソースの制約チェック

```
[zabbix-server01] # pcs constraint list
Location Constraints:
Ordering Constraints:
  start filesystem then start pgsql (kind:Mandatory)
  start db-group then start zabbix-group (kind:Mandatory)
Colocation Constraints:
  zabbix-group with db-group (score:INFINITY)
Ticket Constraints:
```

© 2022 SRA OSS, Inc. Japan

Pacemaker のリソース設定

- フェイルオーバをテスト

```
[zabbix-server01] # pcs node standby zabbix-server01
[zabbix-server01] # pcs status
…
Node List:
 * Node zabbix-server01: standby
  * Online: [ zabbix-server02 ]

Full List of Resources:
  * Resource Group: zabbix-group:
    * fip       (ocf::heartbeat:IPaddr2):       Started zabbix-server02
    * nginx     (ocf::heartbeat:nginx): Started zabbix-server02
    * php-fpm   (systemd:php-fpm):      Started zabbix-server02
    * zabbix-server     (systemd:zabbix-server):        Started zabbix-server02
  * Resource Group: db-group:
    * filesystem        (ocf::heartbeat:Filesystem):    Started zabbix-server02
    * pgsql     (ocf::heartbeat:pgsql): Started zabbix-server02
…
[zabbix-server01] # pcs node unstandby zabbix-server01
```

© 2022 SRA OSS, Inc. Japan

参考情報

- Zabbix 設定バックアップ同期ツール
  - https://enterprise.zabbix.co.jp/documents/zabbix-backup-sync
- RedHat 8: Configuring and managing high availability clusters
  - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_and_managing_high_availability_clusters/index
- Pacemaker
  - https://clusterlabs.org/
- Corosync
  - http://corosync.github.io/corosync/
- DRBD
  - https://www.linbit.com/drbd/

© 2022 SRA OSS, Inc. Japan