

# Zabbix HA クラスタ構成について

– モニタリングのダウンタイムを最小限に抑える  
Zabbix HA クラスタを構築する方法 –

赤松 俊弘

SRA OSS, Inc. 日本支社

# Zabbix の冗長化

Active/Active ?  
Active/Standby ?

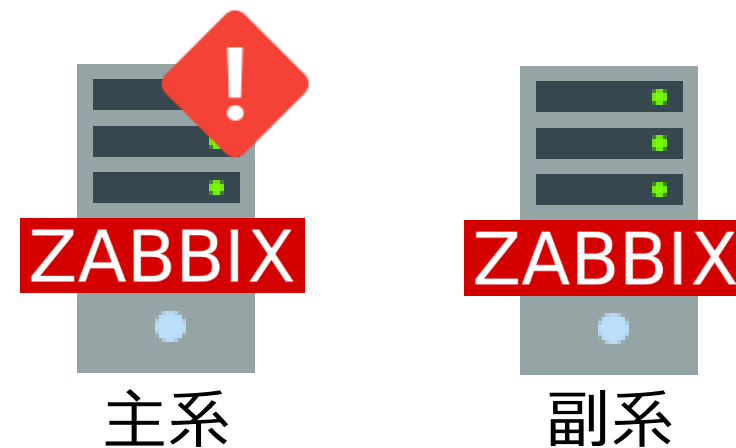
構成 ?  
構築方法 ?



# Active/Active V.S. Active/Standby

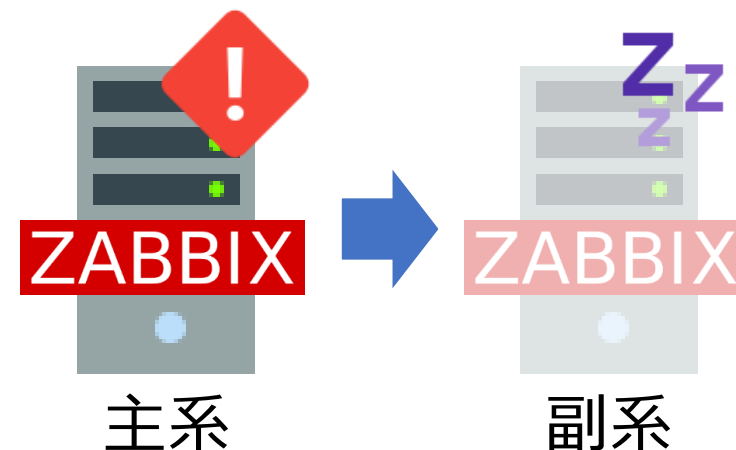
- Active/Active

- 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



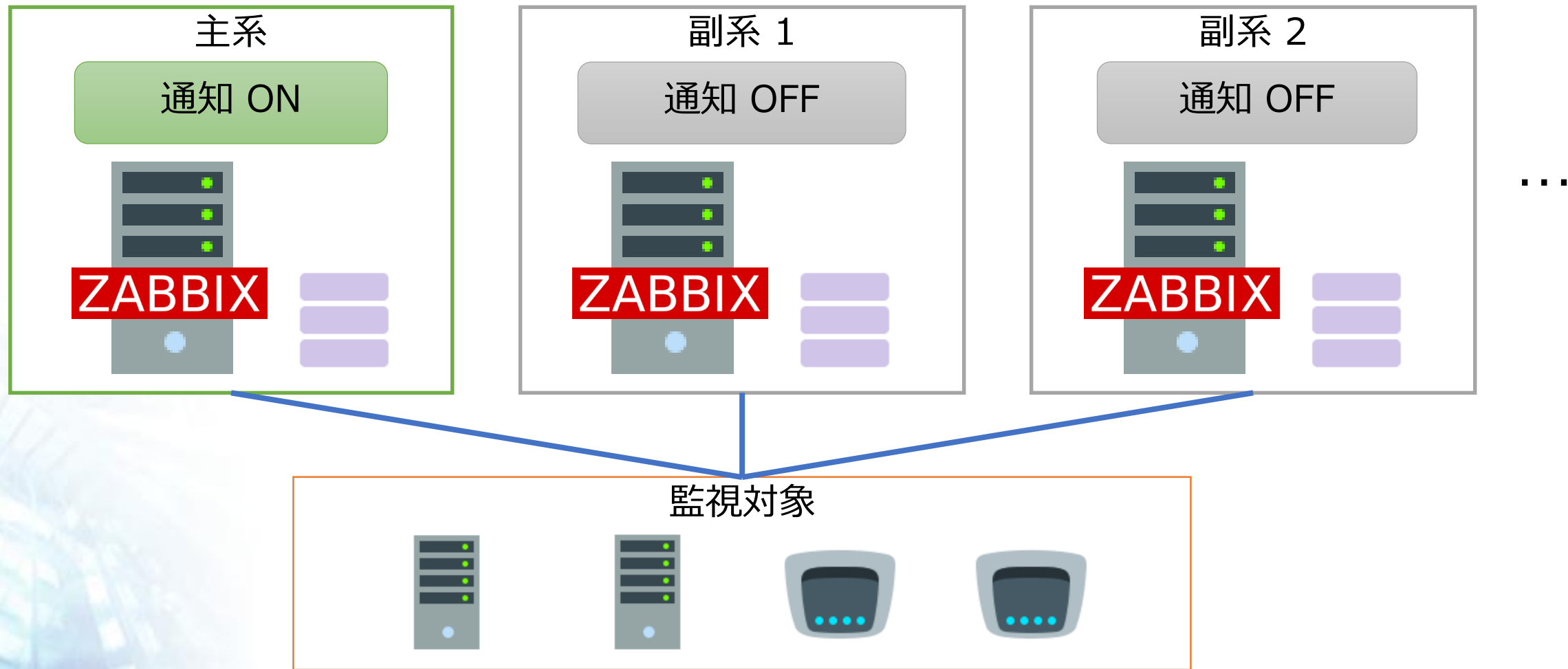
- Active/Standby

- 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

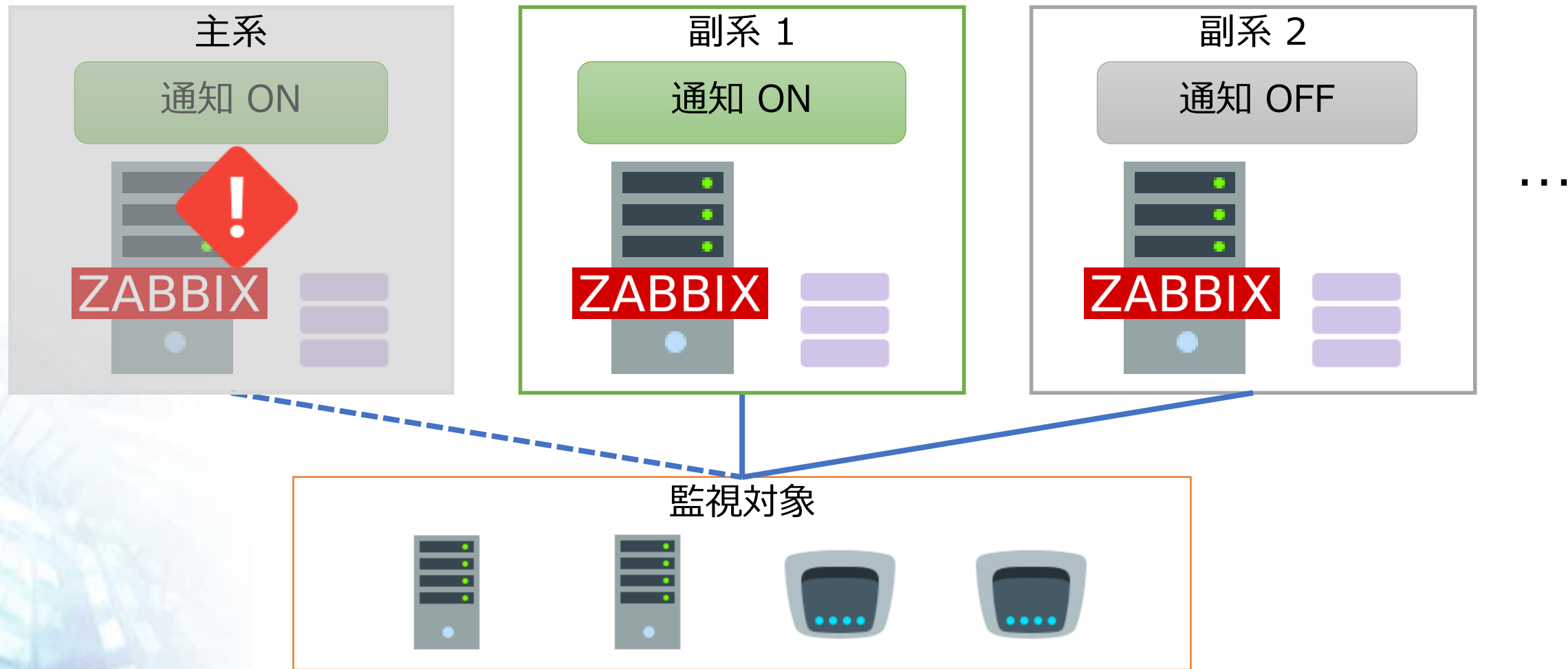


# Zabbix Active/Active HA クラスタ

# Zabbix Active/Active HA クラスタ



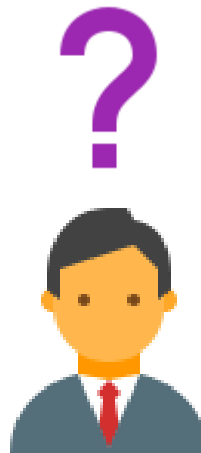
# Zabbix Active/Active HA クラスタ



# Active/Active クラスタ運用時の疑問

監視設定の同期？

主系/副系の切替？



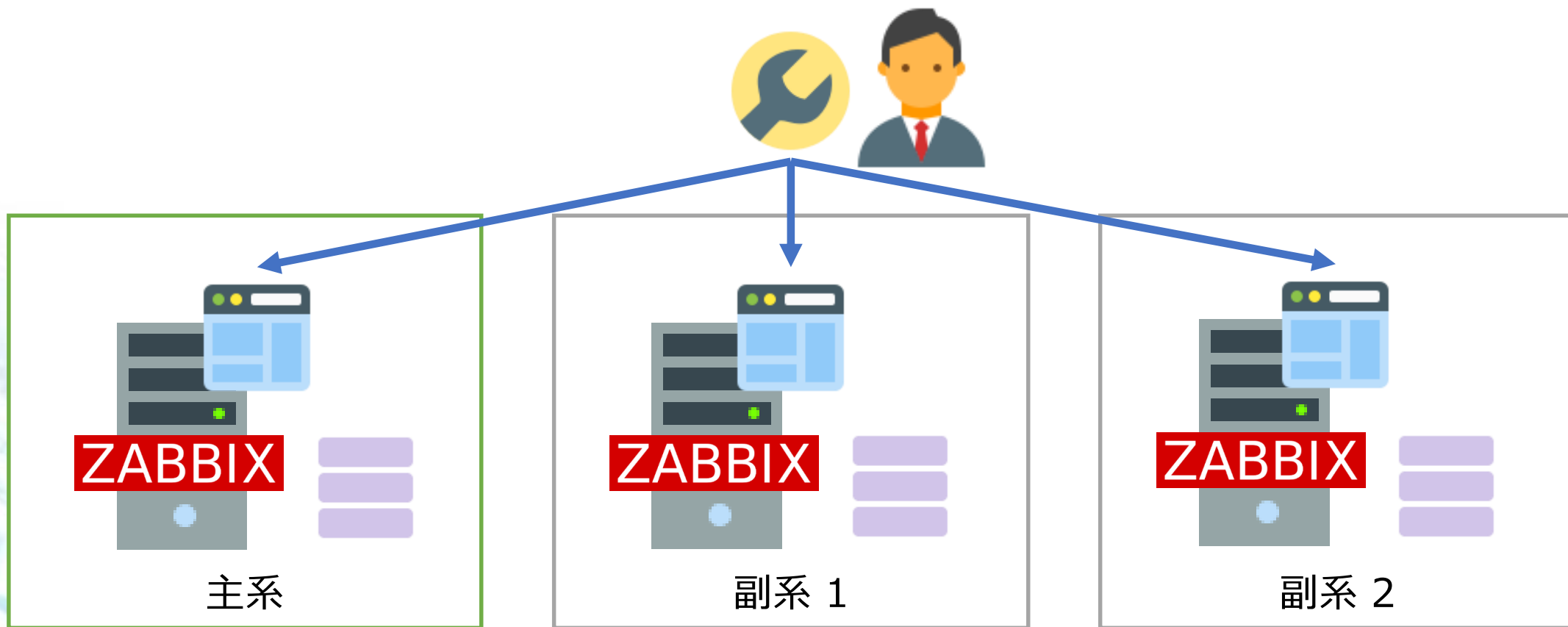
# Active/Active クラスタの考慮点

- 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



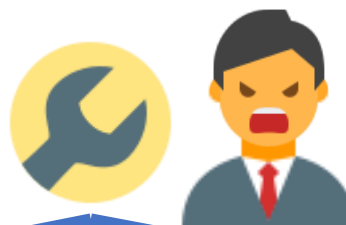
## 設定の同期方法の検討

- 全て手動で各 Web インタフェースから同一の設定を実施

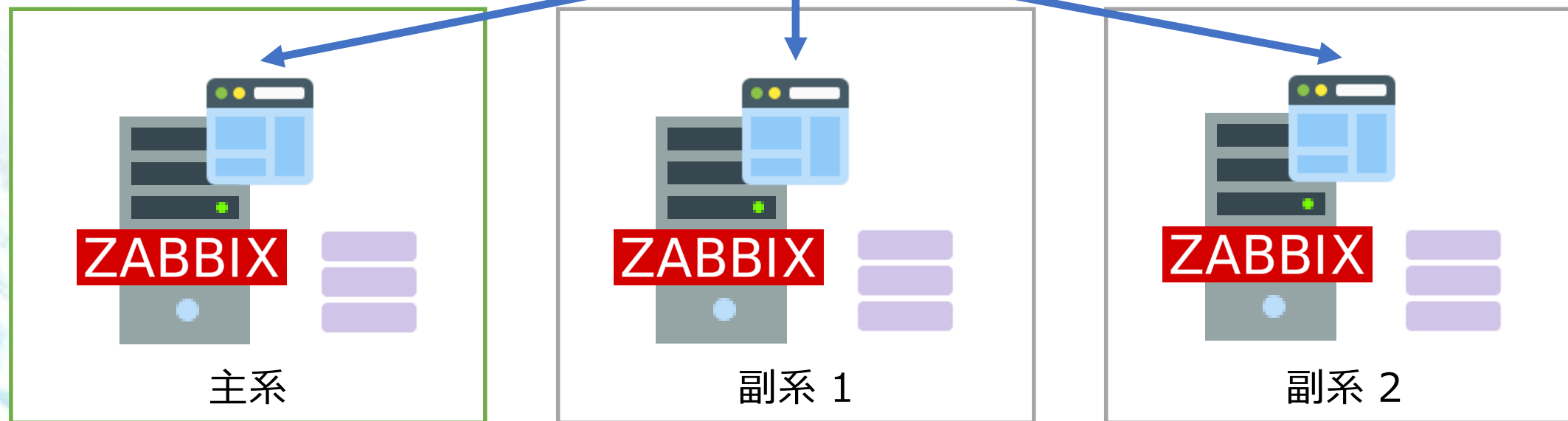


## 設定の同期方法の検討

- 全て手動で各 Web インタフェースから同一の設定を実施

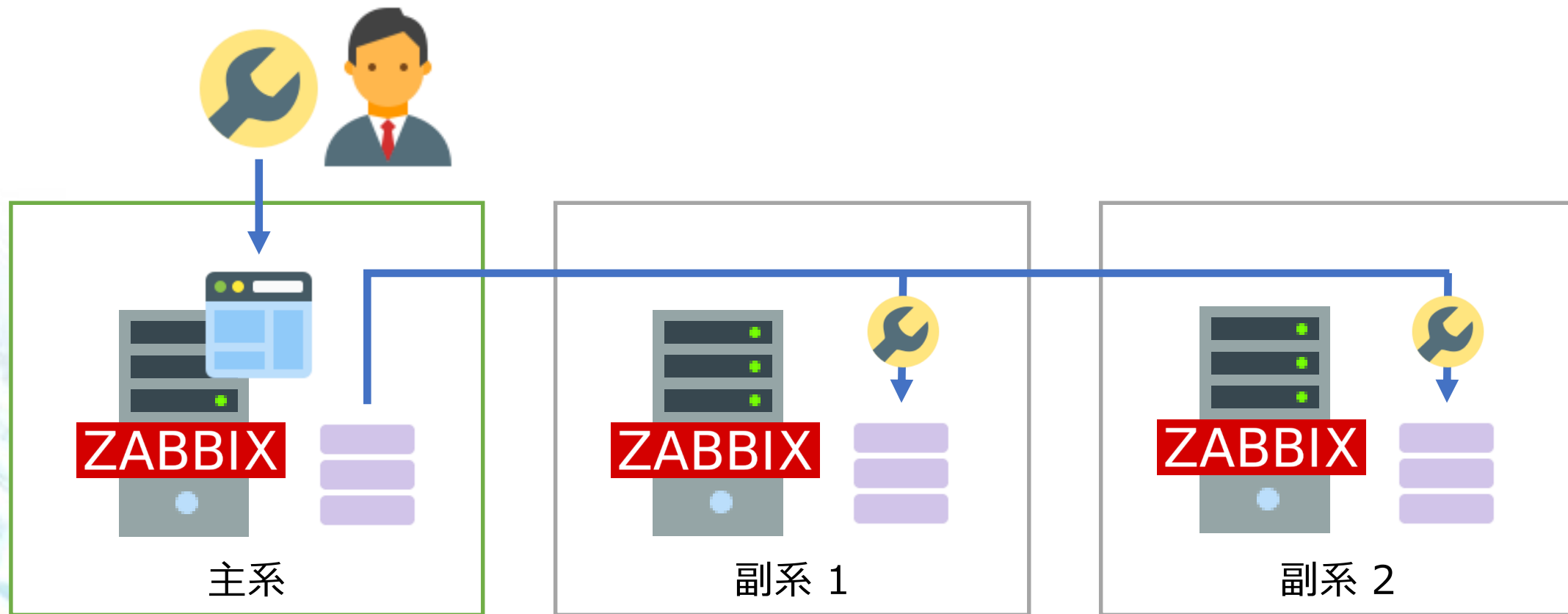


手間がかかる  
人為的ミスの可能性



## 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

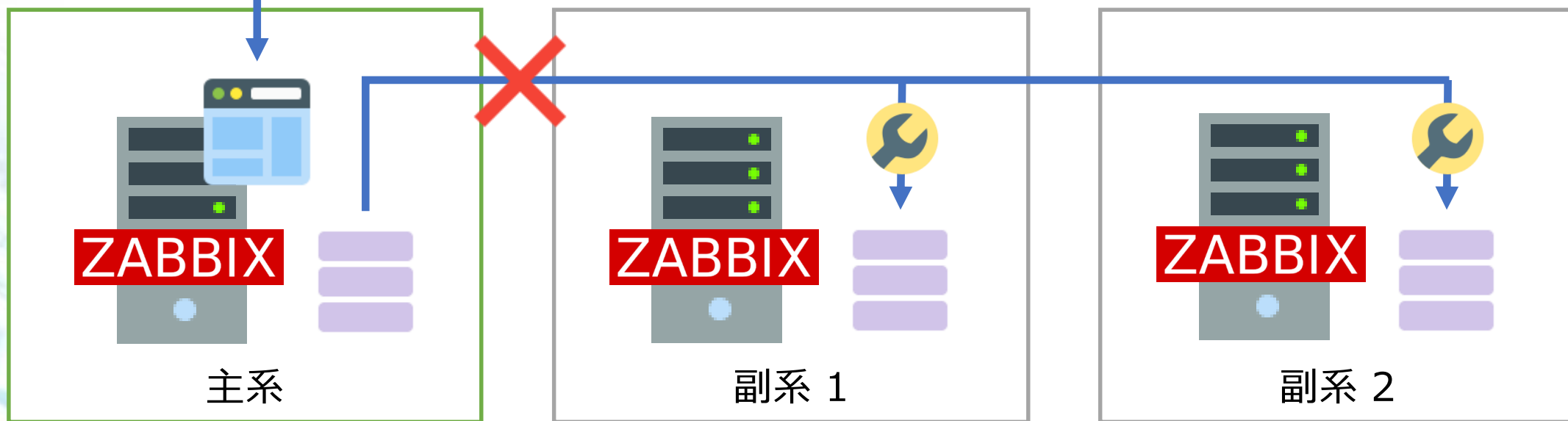


## 設定の同期方法の検討

- 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

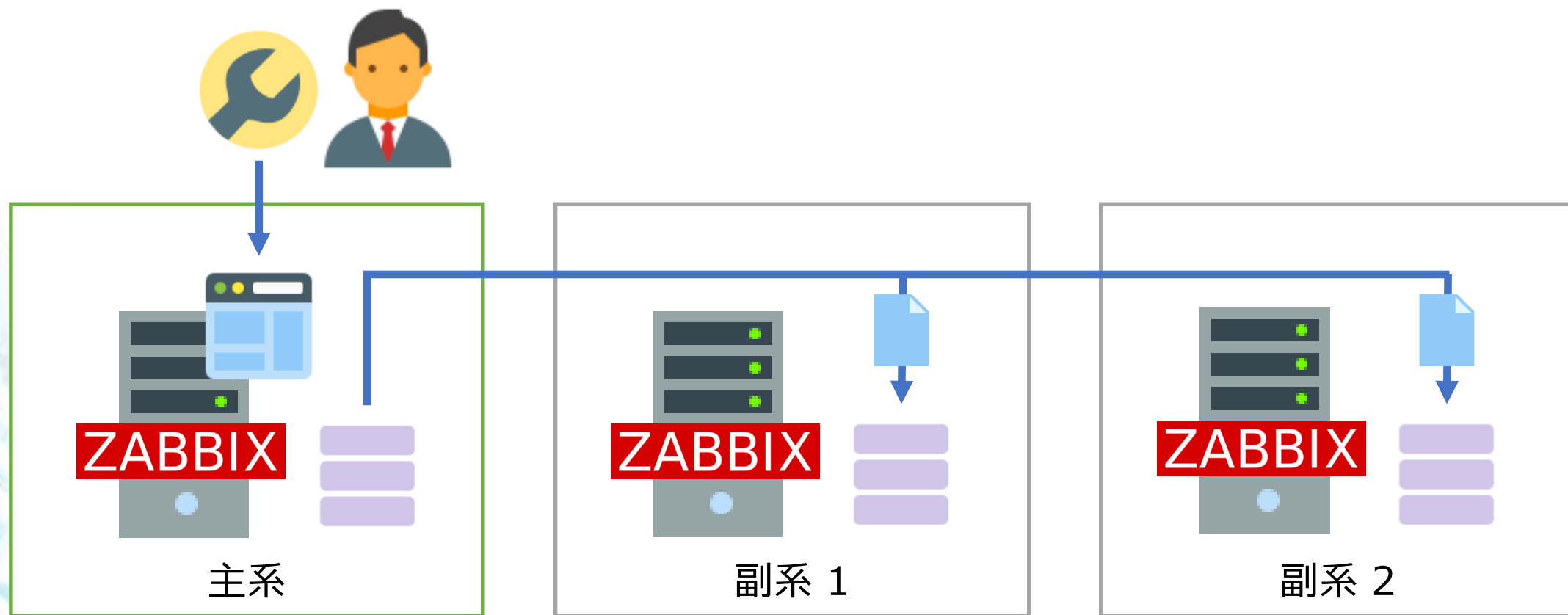


設定関連のテーブル内に監視結果に応じて Zabbix によって更新されるカラムがあるため  
レプリケーションが停止する可能性あり



## 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

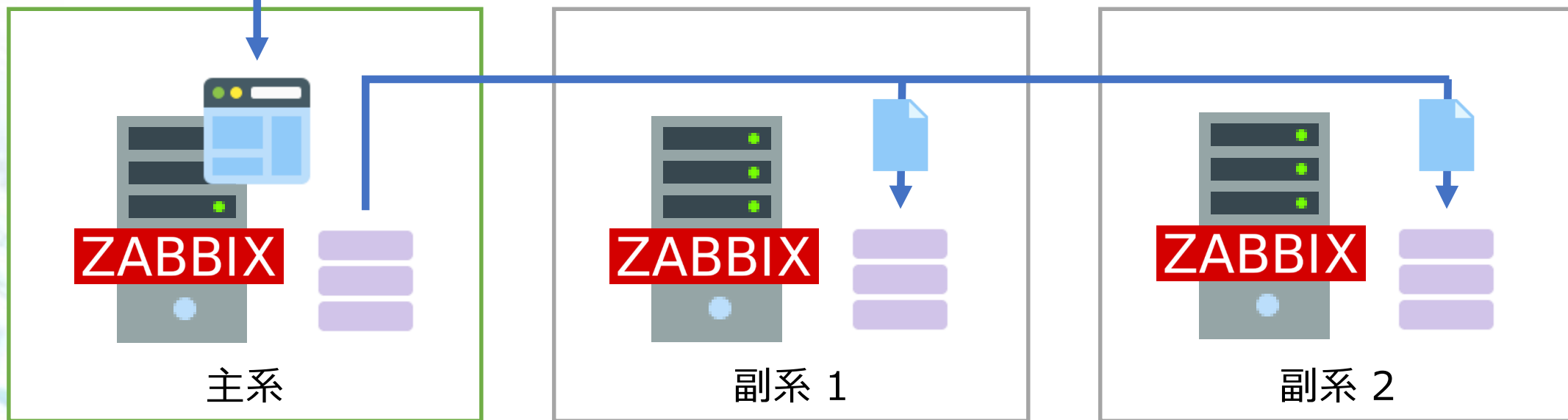


## 設定の同期方法の検討

- DB の設定関連テーブルのみをダンプ/リストア



リストア時には  
データ全体の整合性を保持する必要あり  
通知を何らかの方法で OFF にする必要あり

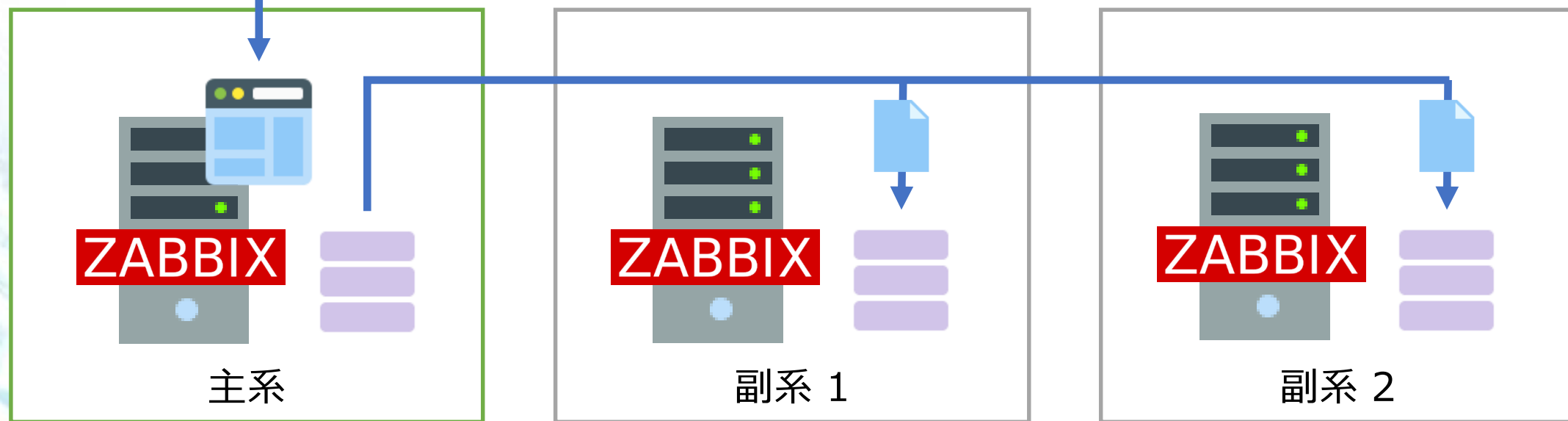


## 設定の同期方法の検討

- Zabbix 設定バックアップ同期ツール



コマンド 1 つで設定を副系に同期  
データ全体の整合性を担保、副系の通知を自動で OFF  
SRA OSS の Zabbix サポート経由でも提供可能



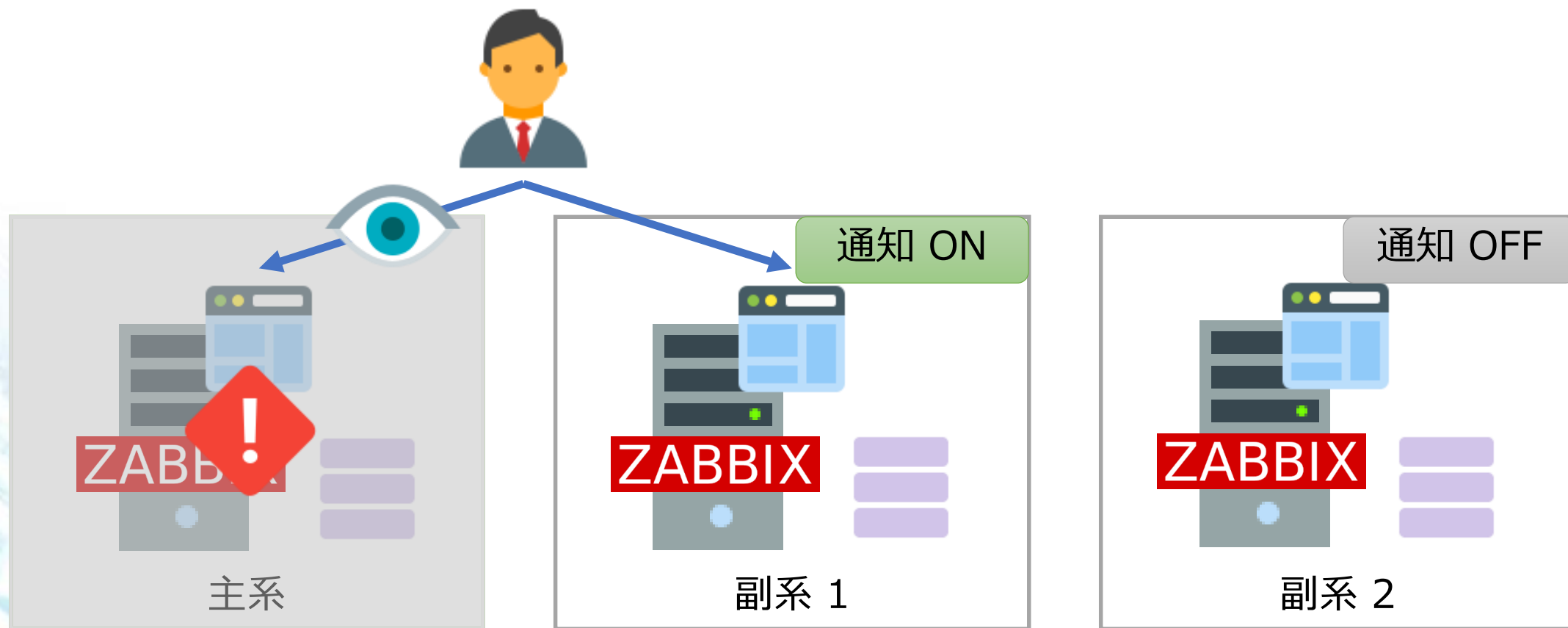
# Zabbix 設定バックアップツールの注意点

- 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問題



# 切替方法の検討

- 全て手動で切替を実施

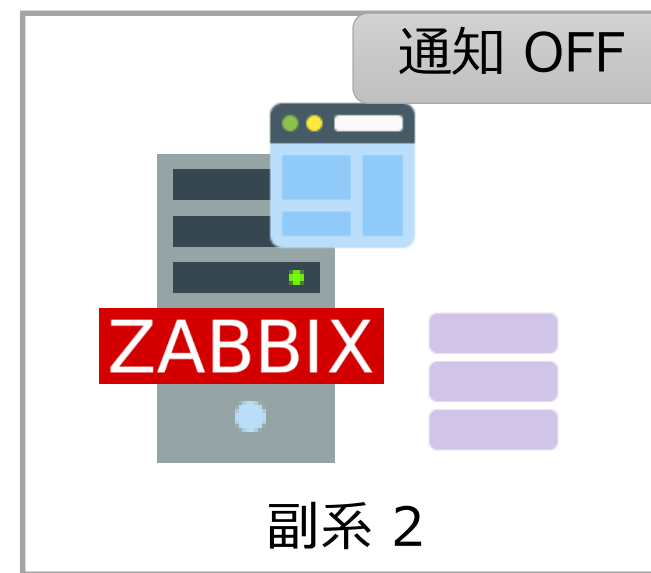
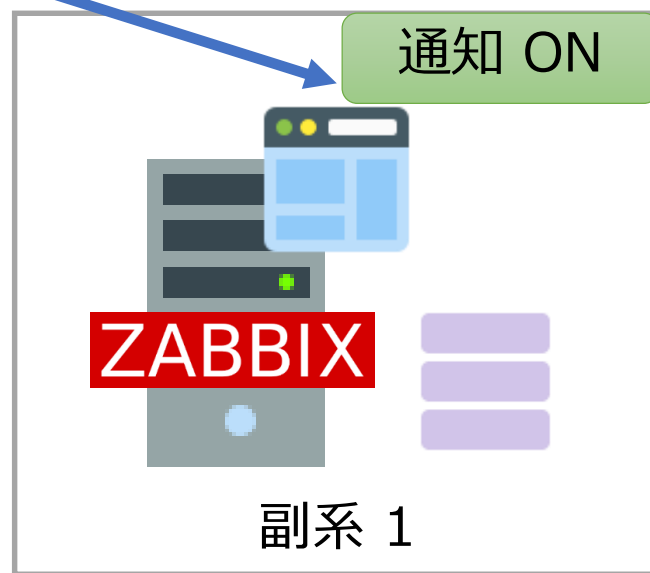
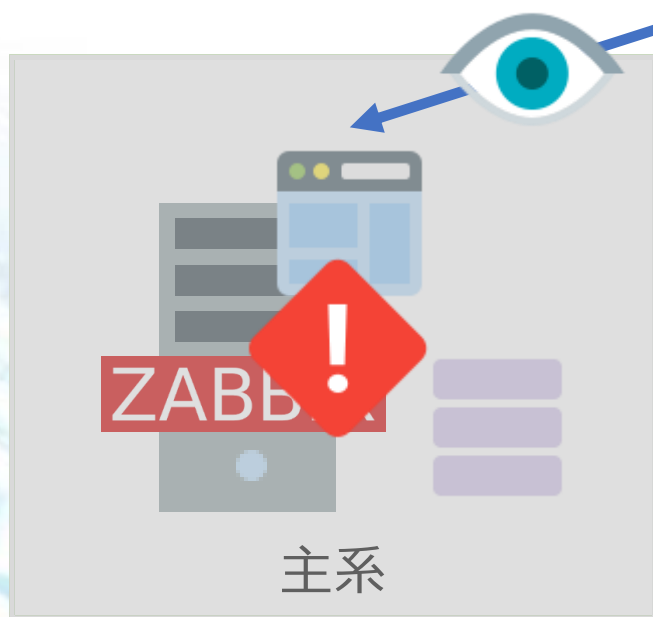


# 切替方法の検討

- 全て手動で切替を実施



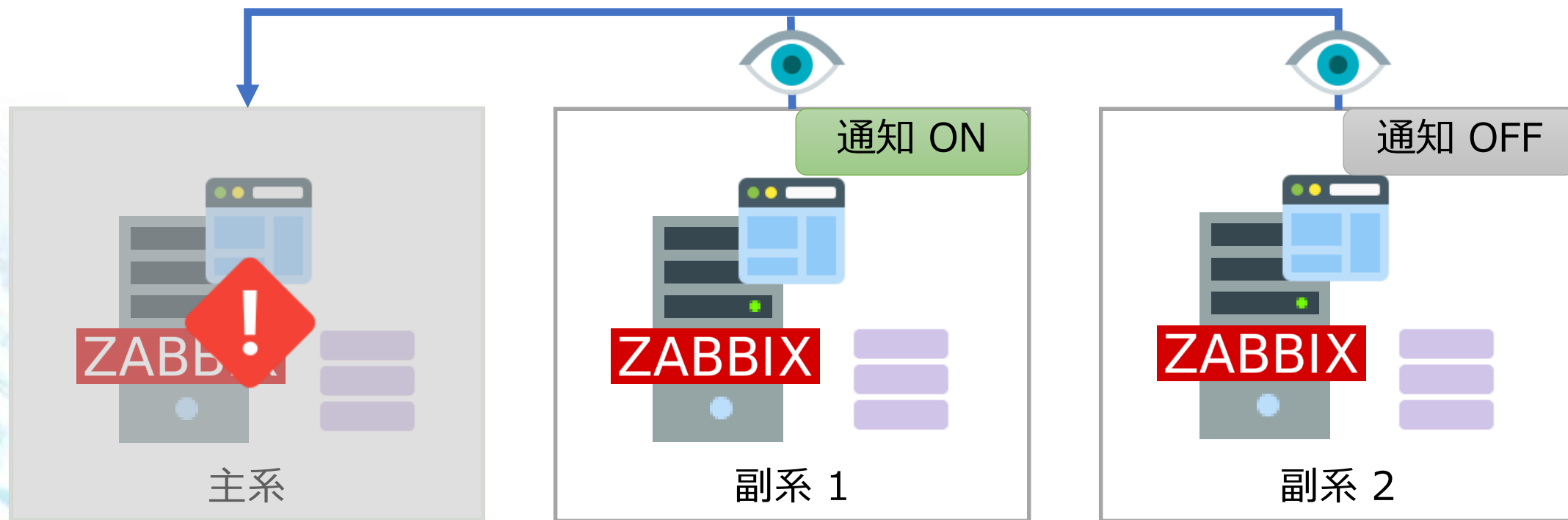
主系の障害検知に遅れると切替まで通知が来ない  
常に人が主系を監視するのは困難



...

# 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 ON にする  
アクションを副系の 1 つで ON にしておく



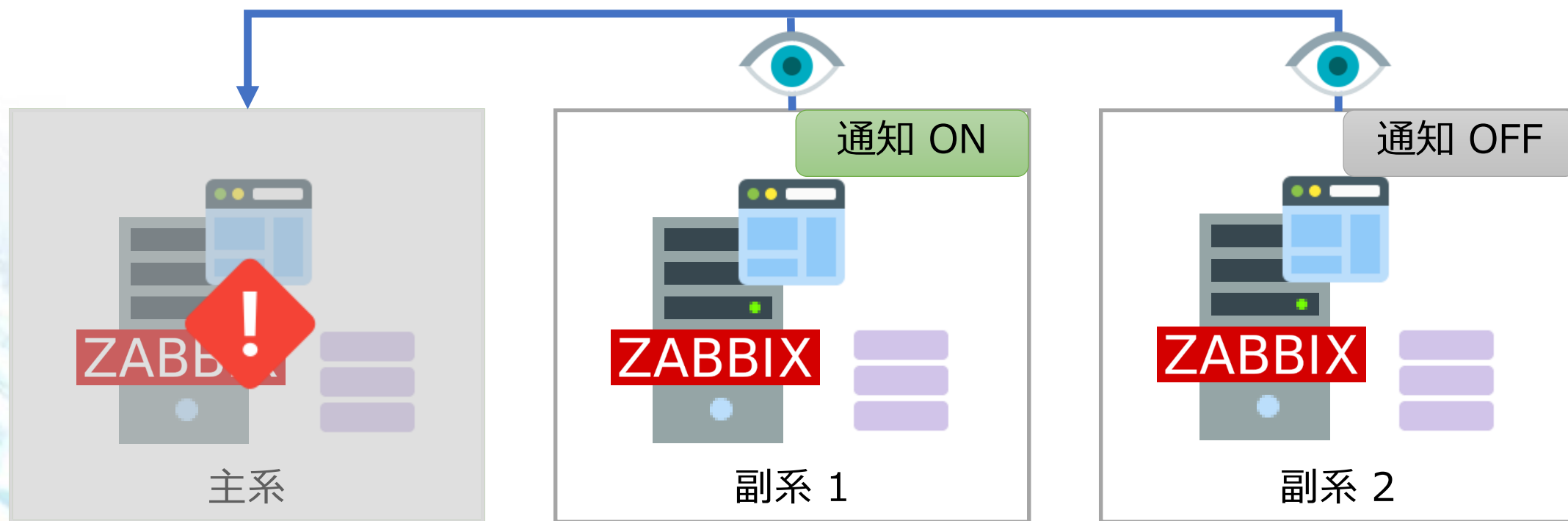
## 切替方法の検討

- Zabbix で監視と切替を実施



副系が複数ある場合は要工夫

- 副系で主系の障害検知時に自身の通知を ON にする  
アクションを副系の 1 つで ON にしておく



# Zabbix Active/Active HA クラスタまとめ

## 長所

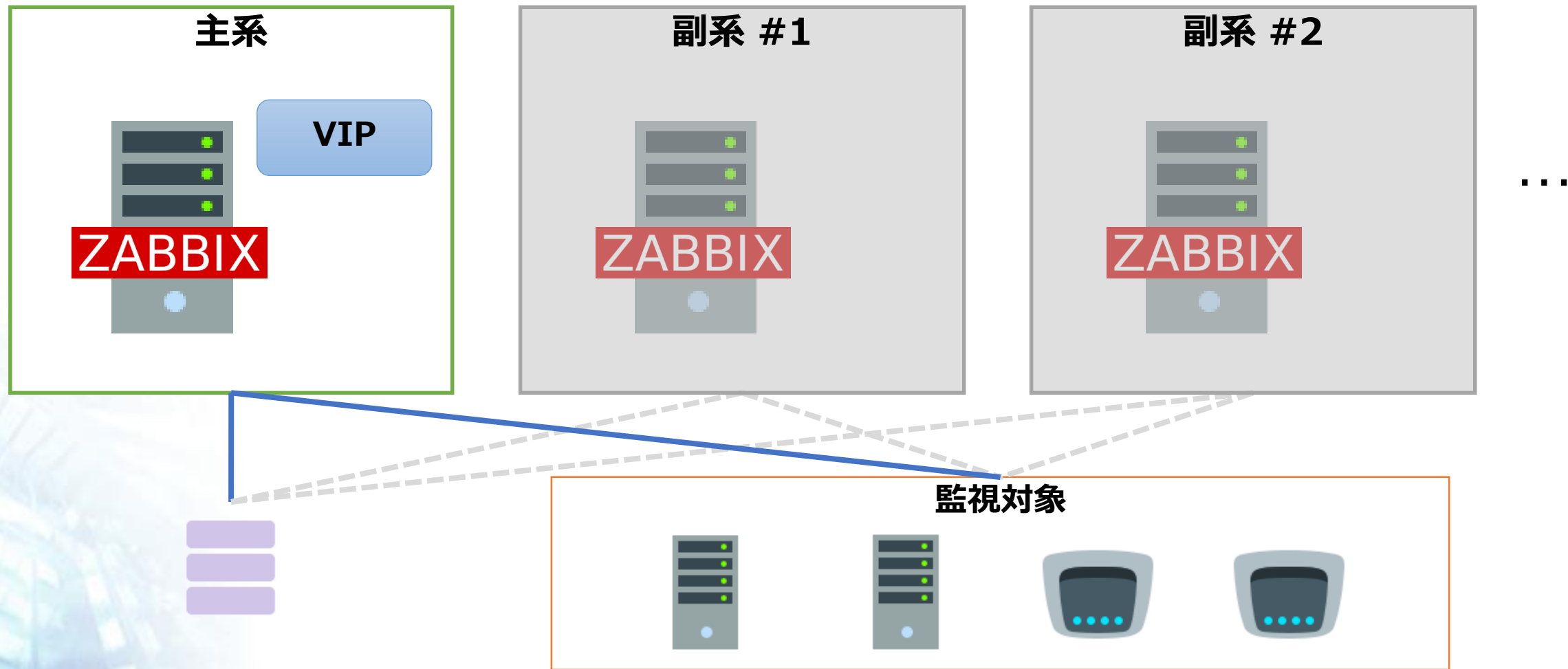
- 構築は簡単
- 障害発生時に監視の  
ダウンタイムなし

## 短所

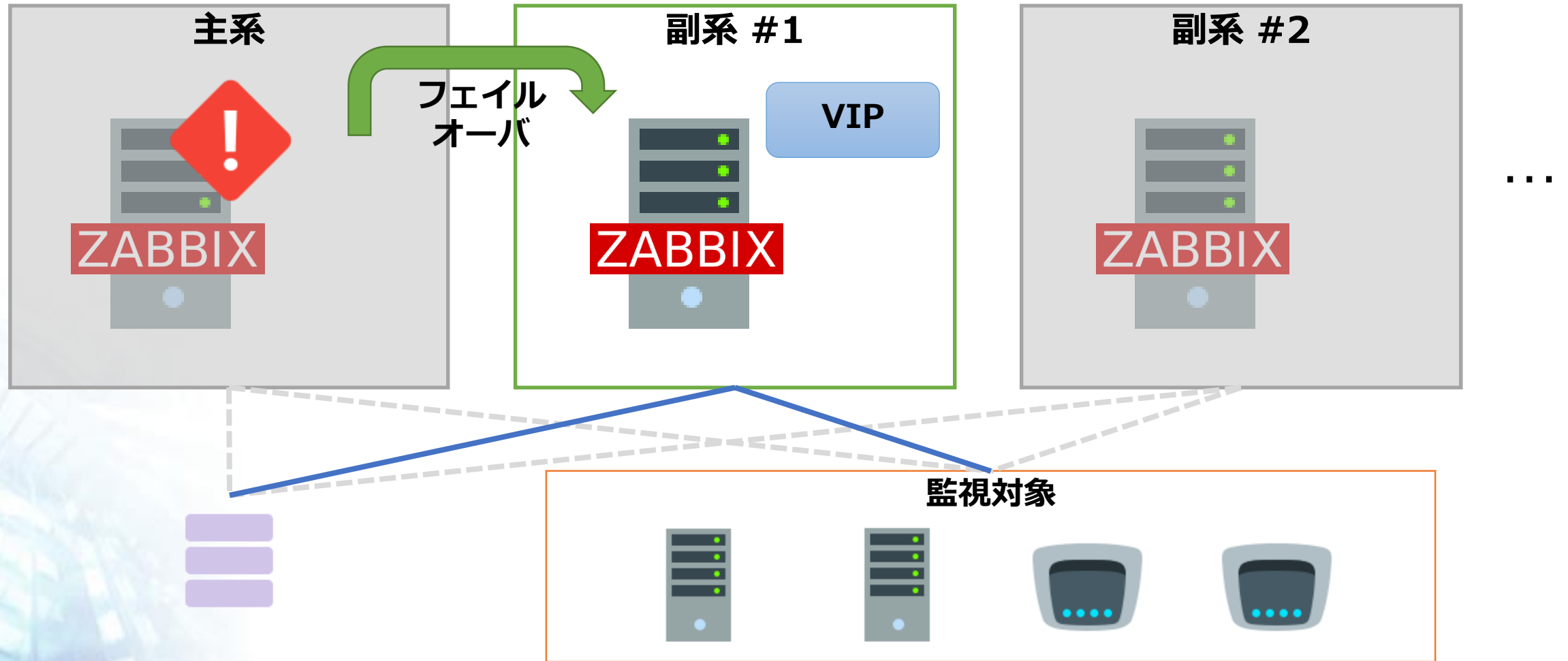
- 監視対象側の負荷は高い
- 設定の同期や切替に要工夫

# Zabbix Active/Standby HA クラスタ

# Zabbix Active/Standby HA クラスタ

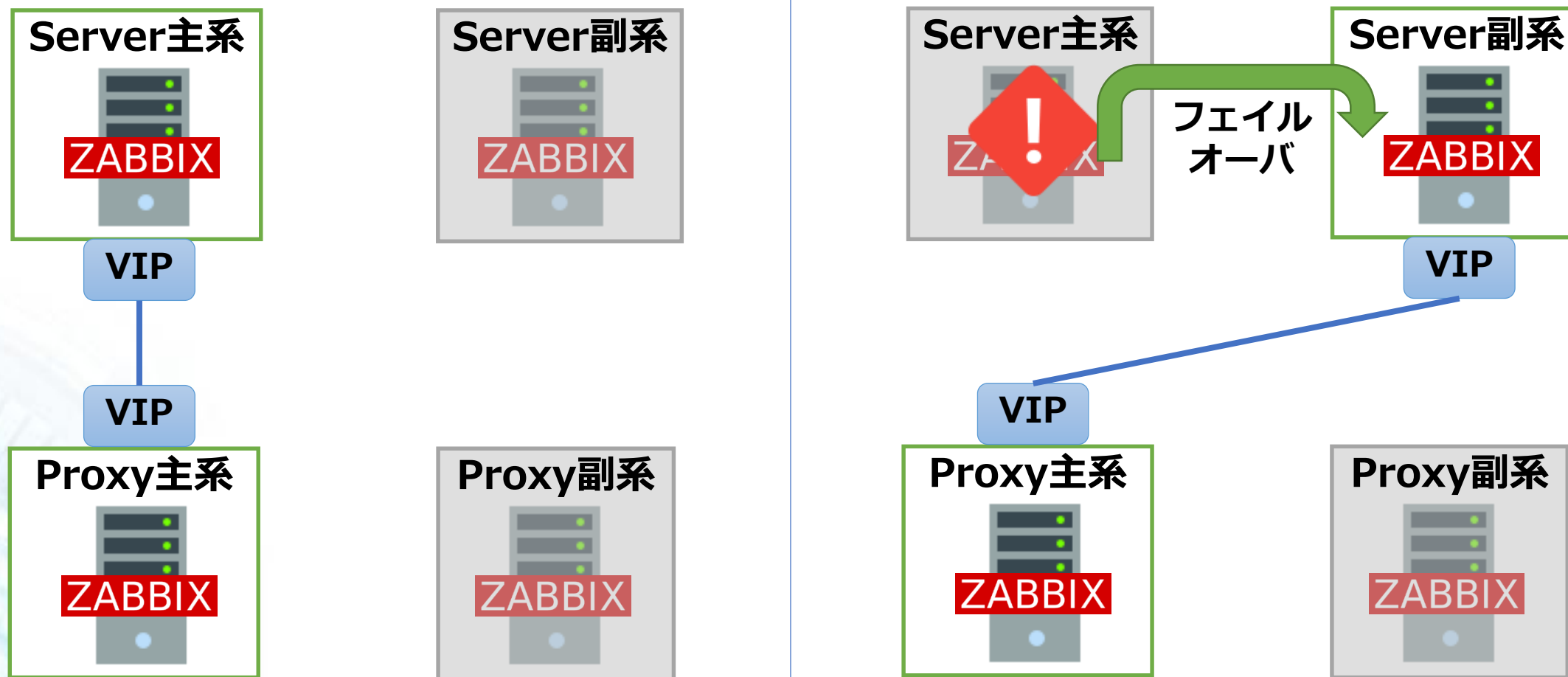


# Zabbix Active/Standby HA クラスタ

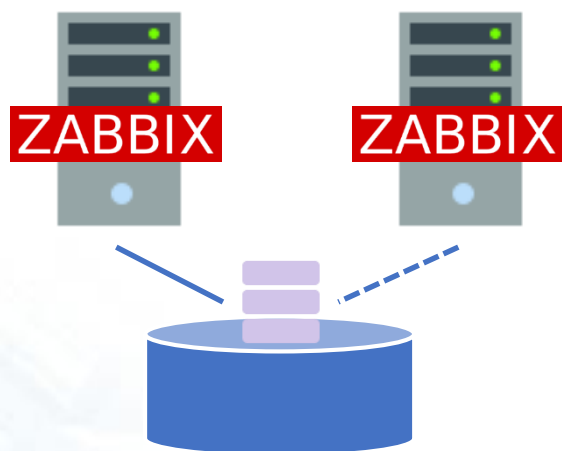




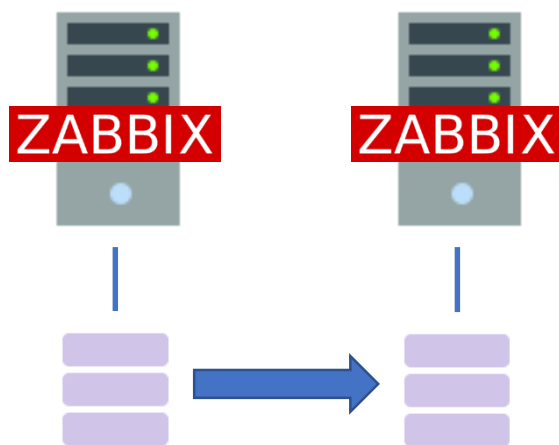
# Zabbix proxy を含む Active/Standby 構成



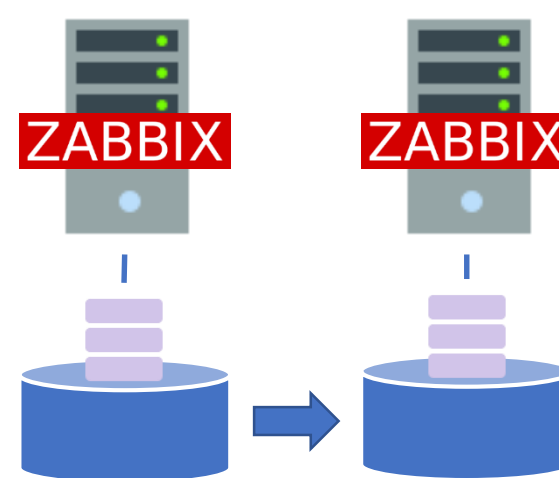
# Active/Standby DB 構成手法



共有ストレージ

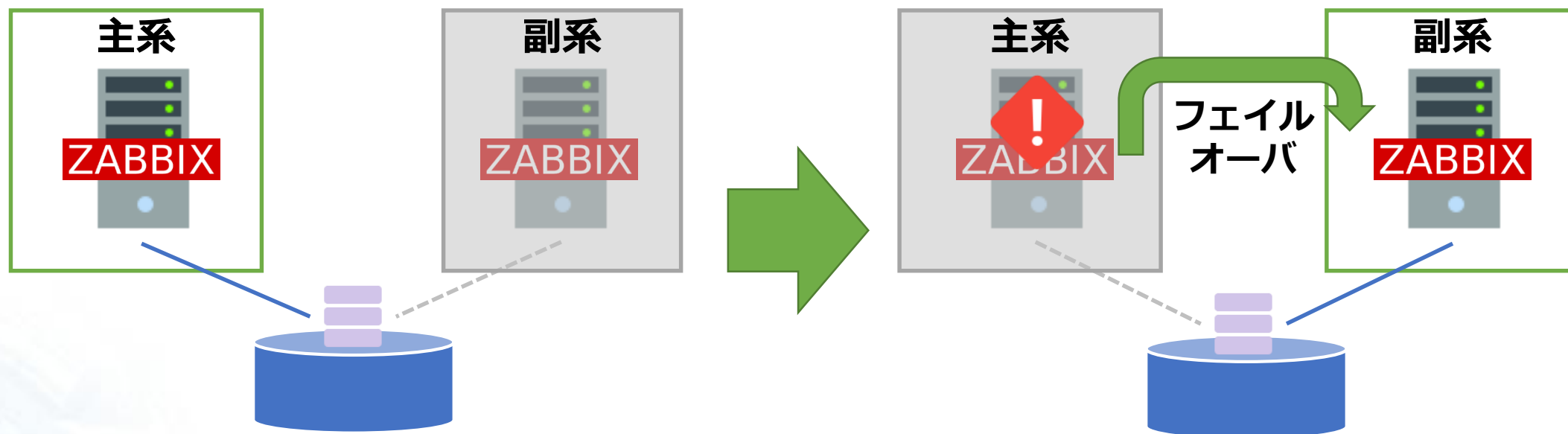


DB レプリケーション



ブロックデバイス  
レプリケーション

## 共有ストレージによる構成



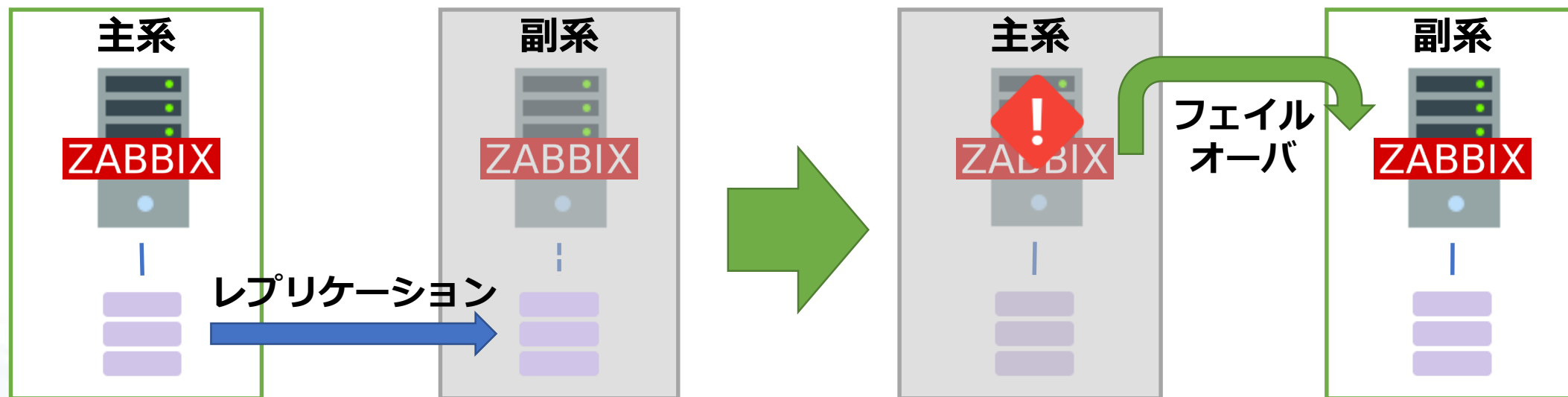
### 長所

- データ同期のオーバーヘッドなし
- 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
- フェイルオーバーに要時間
- ストレージが単一障害点

# DB レプリケーションによる構成



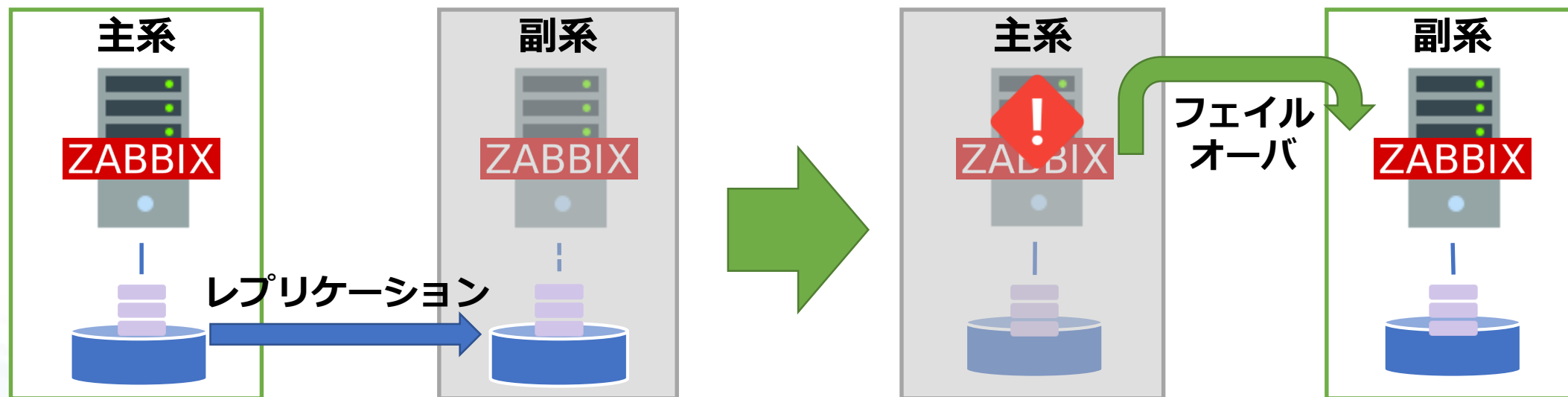
## 長所

- 導入費用コスト低
- 構成がシンプル
- フェイルオーバー速め

## 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

# 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

# Active/Standby クラスタの考慮点

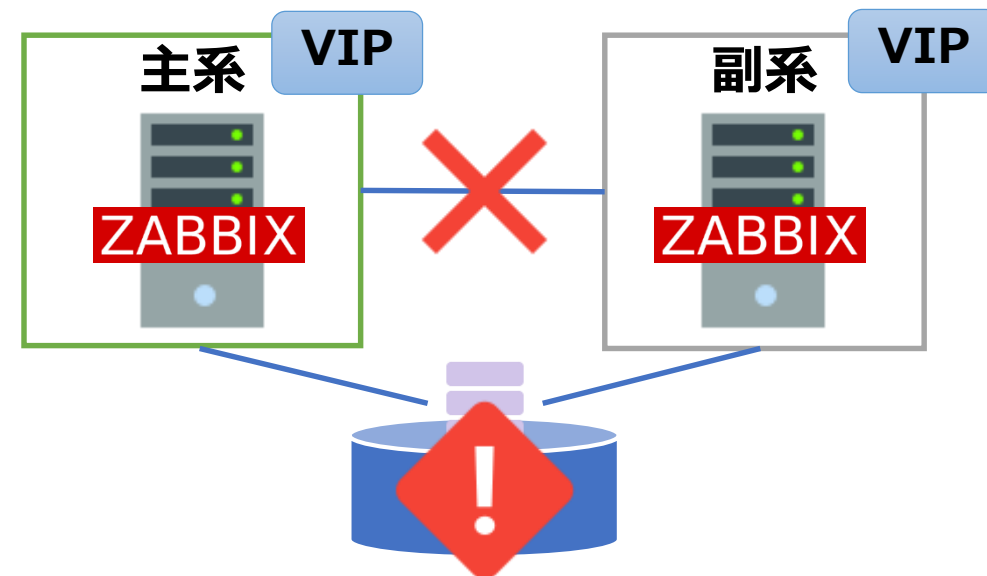
- スプリットブレイン対策
  - STONITH (Shoot The Other Node In The Head)
  - その他の排他制御機構
- Zabbix サーバ上のログ監視や SNMP トラップ監視
  - フェイルオーバー時に再読み込みの可能性あり
  - ログファイルや SNMP トラップファイルは共有ストレージに

# 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



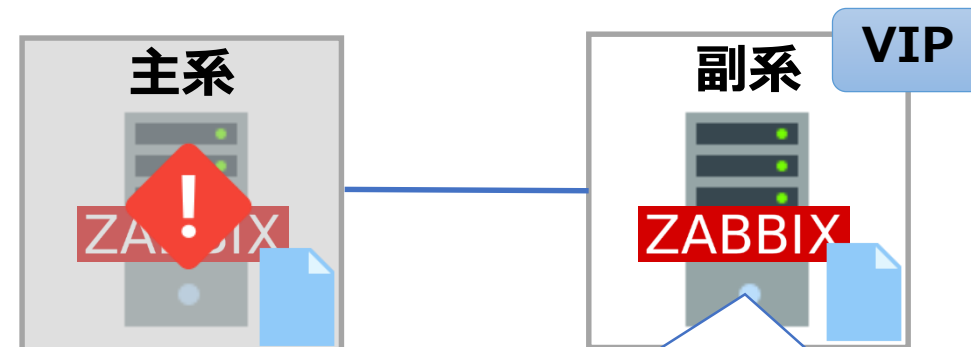
STONITH や共有ディスクの排他制御などで  
スプリットブレイン対策を！

# 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



フェイルオーバー時に再読み込み



サイズが小さくなってるから  
ローテーションされたな  
先頭から読もう！

VIP 経由のログ監視や SNMP トラップ監視は  
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！



# 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

# Zabbix ネイティブの HA 機能

- Zabbix 6.0 のロードマップ

## High availability, performance and scalability

- Proxies to provide automatic load balancing and HA  
★ Top voted!
- Native HA setup for Zabbix Server **In dev**  
★ Top voted!
- API performance improvements to speedup operations with 1M of devices
- Scalable history storage

It will include standard API for storage of history and trend data as well as support of new time-series storage engines.

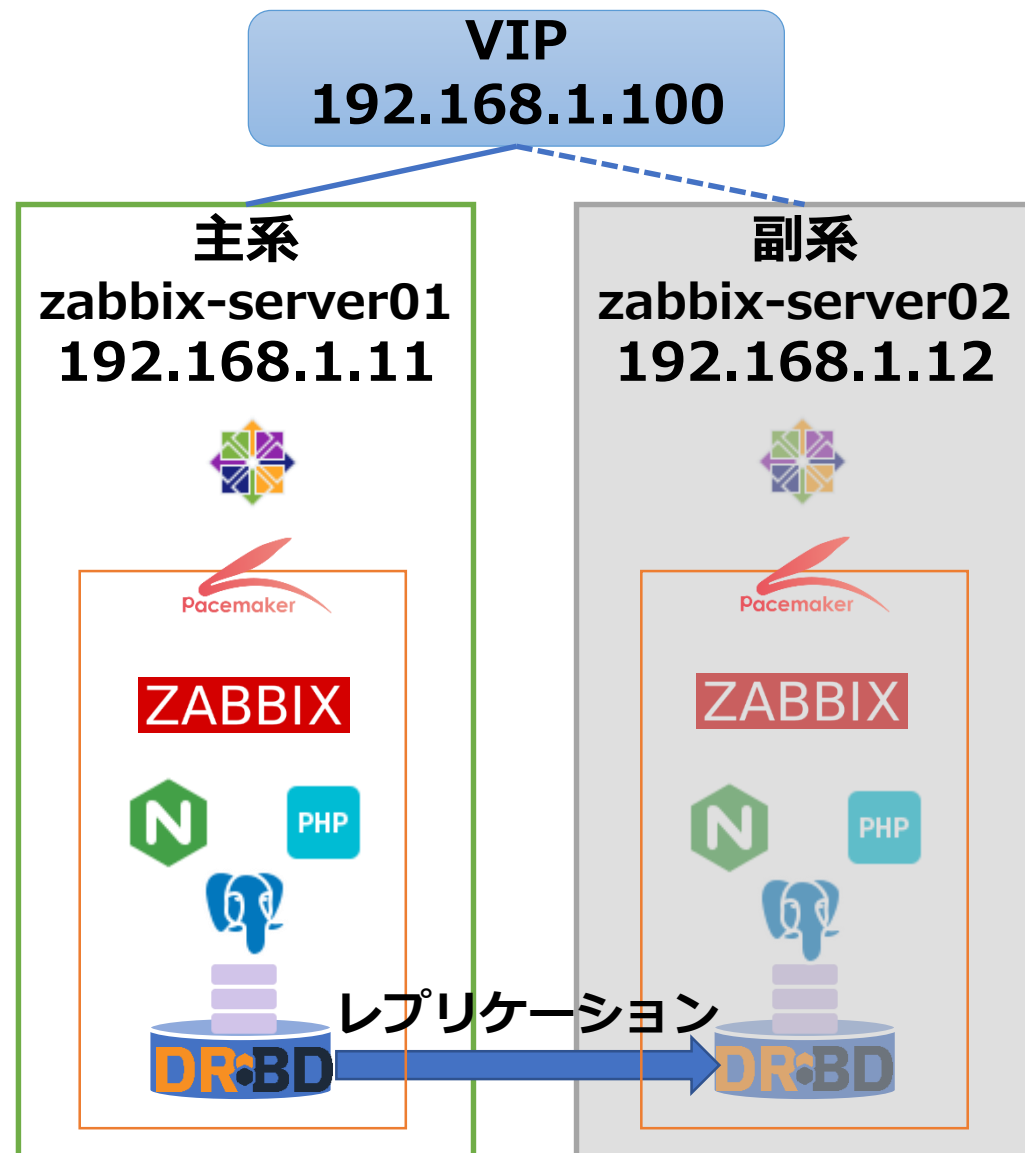
[ZBXNEXT-714](#) ★ Top voted!

<https://www.zabbix.com/jp/roadmap>

# Active/Standby クラスタ 設定例

# 環境

- OS: CentOS 8
- Software:
  - Zabbix 5.0.4
  - PostgreSQL 10.14-1
  - Web サーバソフトウェア
    - Nginx 1.14.1-9
    - PHP-FPM 7.2.24-1
  - クラスタソフトウェア
    - Pacemaker 2.0.3-5
    - Corosync 3.0.3-2
  - ブロックデバイスレプリケーション
    - DRBD 9.0.23



# 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

```
# rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.noarch.rpm  
# dnf clean all
```

- 両ノードで Zabbix サーバとフロント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
```

# 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>
```

# 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
```

# 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
```



# 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
```

# 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
```

# 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
```

# 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;
    }
}
```

# 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
```

# 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
```

# 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
```

## 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
```



# 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
```

# 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
```

## 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
```

# 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
```

# 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
```

## db-group

filesystem

PostgreSQL

## zabbix-group

VIP

Nginx

PHP-FPM

Zabbix-server

# 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



# 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
...
```

# 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:
```



# Pacemaker のリソース設定

- フェイルオーバーをテスト

```
[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
```

## 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](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/>



SRA OSS, INC.

**Thank you!**