SRA OSS, INC.

Minimizing monitoring downtime - How to build Zabbix HA cluster

Toshihiro Akamatsu
SRA OSS, Inc. Japan

ZABBIX
PREMIUM PARTNER

How to minimize monitoring downtime

- One of solutions is HA cluster
- But …
- Which architecture is the best for you?
- How to build HA cluster?

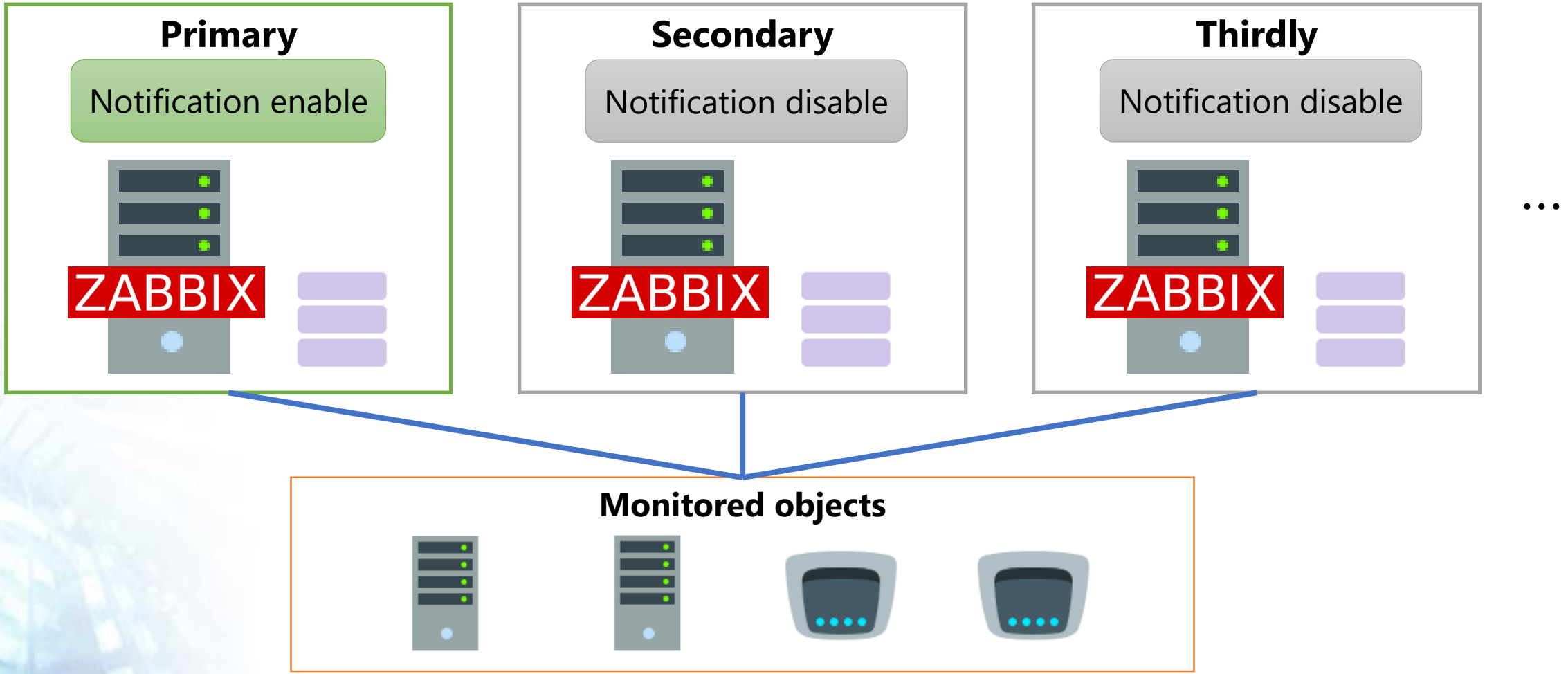
Active/Active V.S. Active/Passive

- Active/Active
  - Building multiple and individual Zabbix servers
  - Monitoring same objects by each Zabbix server
  - Enable to continue monitoring when primary is down

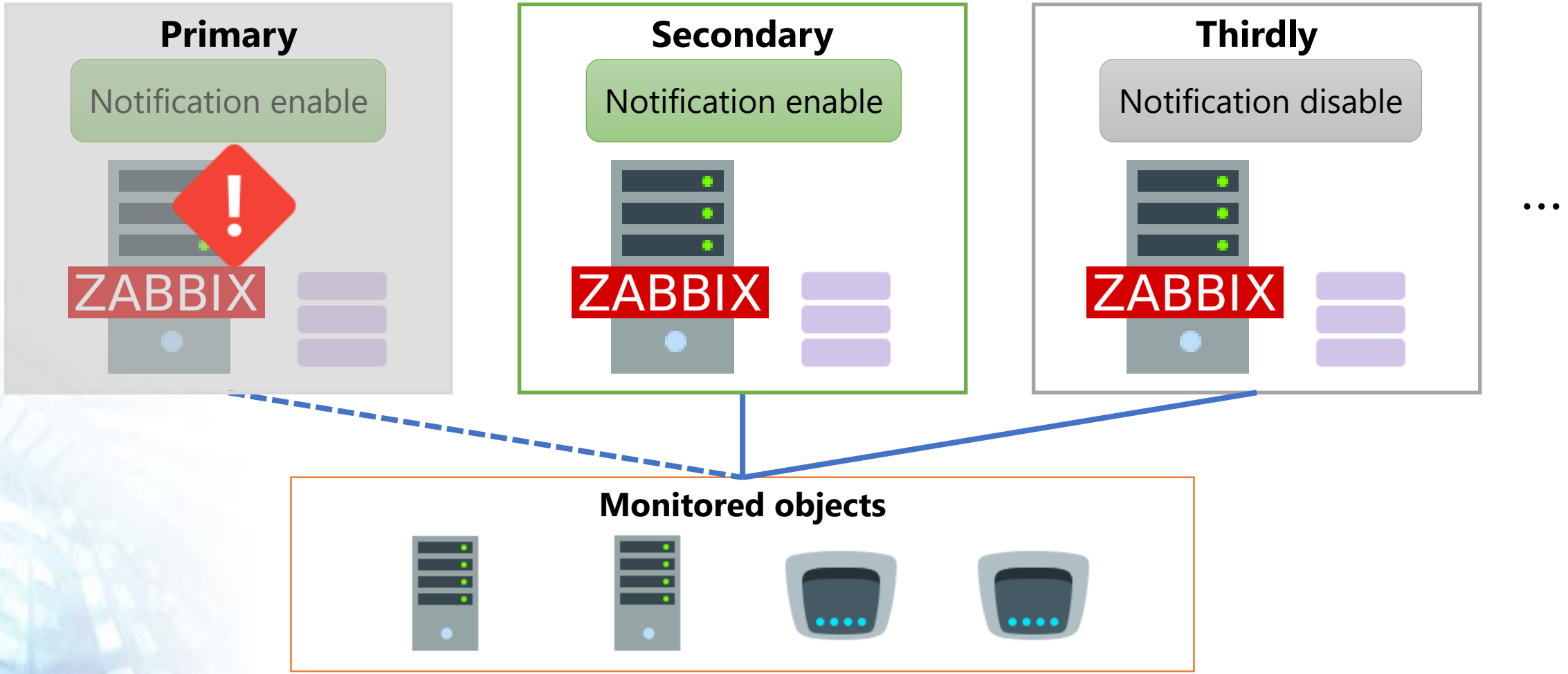
- Active/Passive
  - Building multiple Zabbix servers
  - Cluster software is needed to failover automatically when active is down
  - Monitoring is interrupted momentarily at failover

Zabbix Active/Active HA Cluster

Zabbix Active/Active HA cluster



Zabbix Active/Active HA cluster

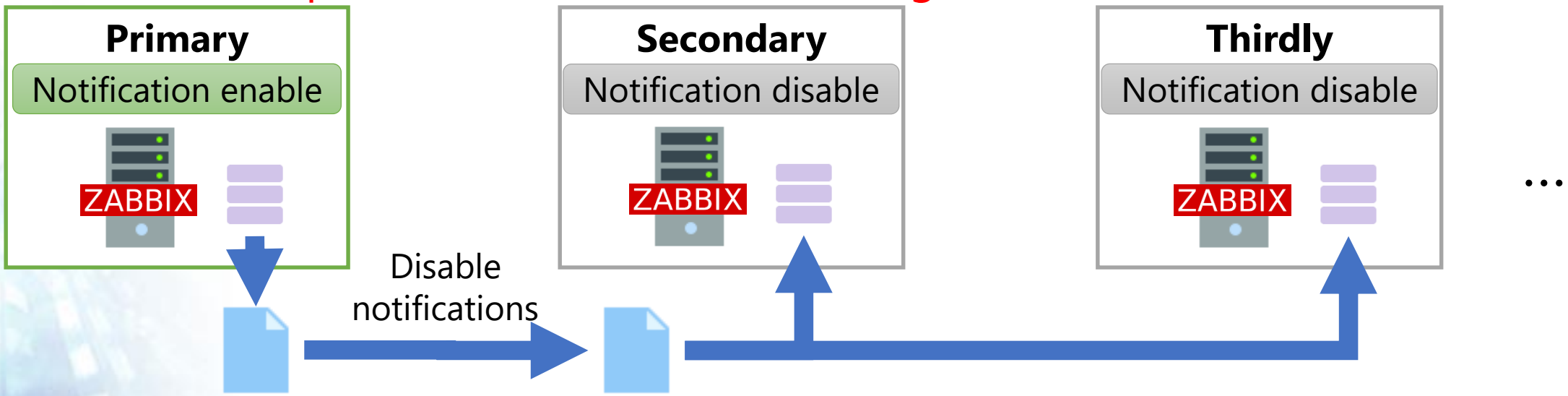


Considerations in building active/active cluster

- How to sync monitoring configurations
  - Need to sync monitoring configurations among all zabbix servers in active/active cluster
  - In addition, notifications (actions) should be disable except primary
- How to switch secondary to primary when primary is down
  - Need to enable configuration sync mechanism and notifications on secondary

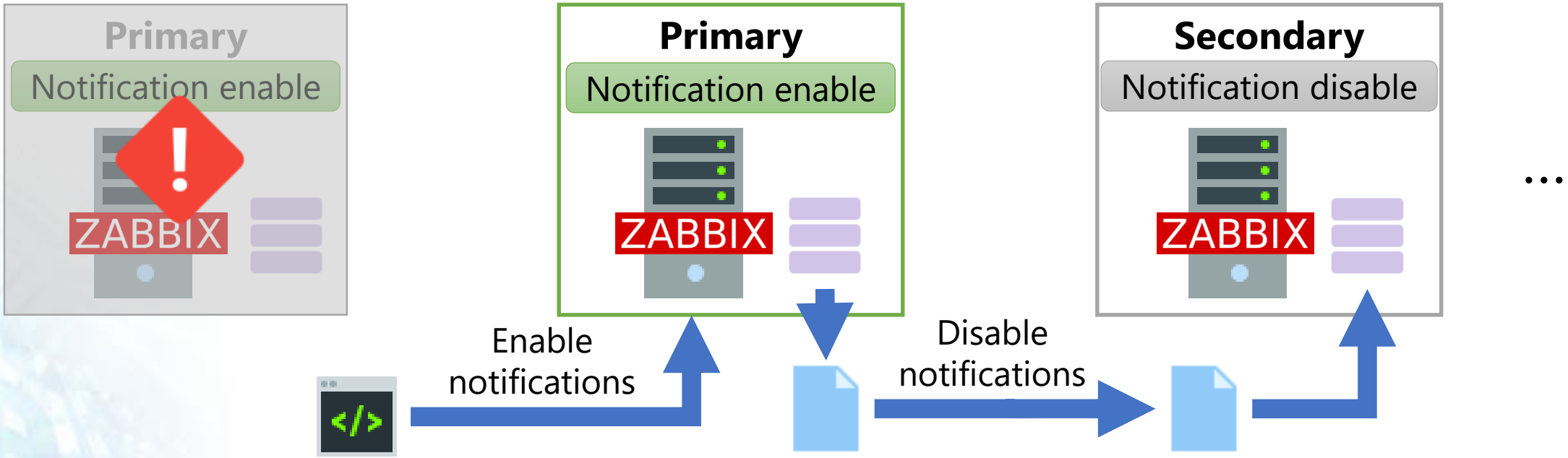
How to sync monitoring configurations

- For example …
  - Dump monitoring configuration tables from primary DB
  - Disable notifications and restore to other DBs
  - Prohibit from updating configuration except primary DB
  - Need to stop Zabbix server when restoring DBs



How to enable required notifications

- For example …
  - Execute script to enable notifications via Zabbix API or SQL



Summary of Zabbix active/active HA cluster

Pros

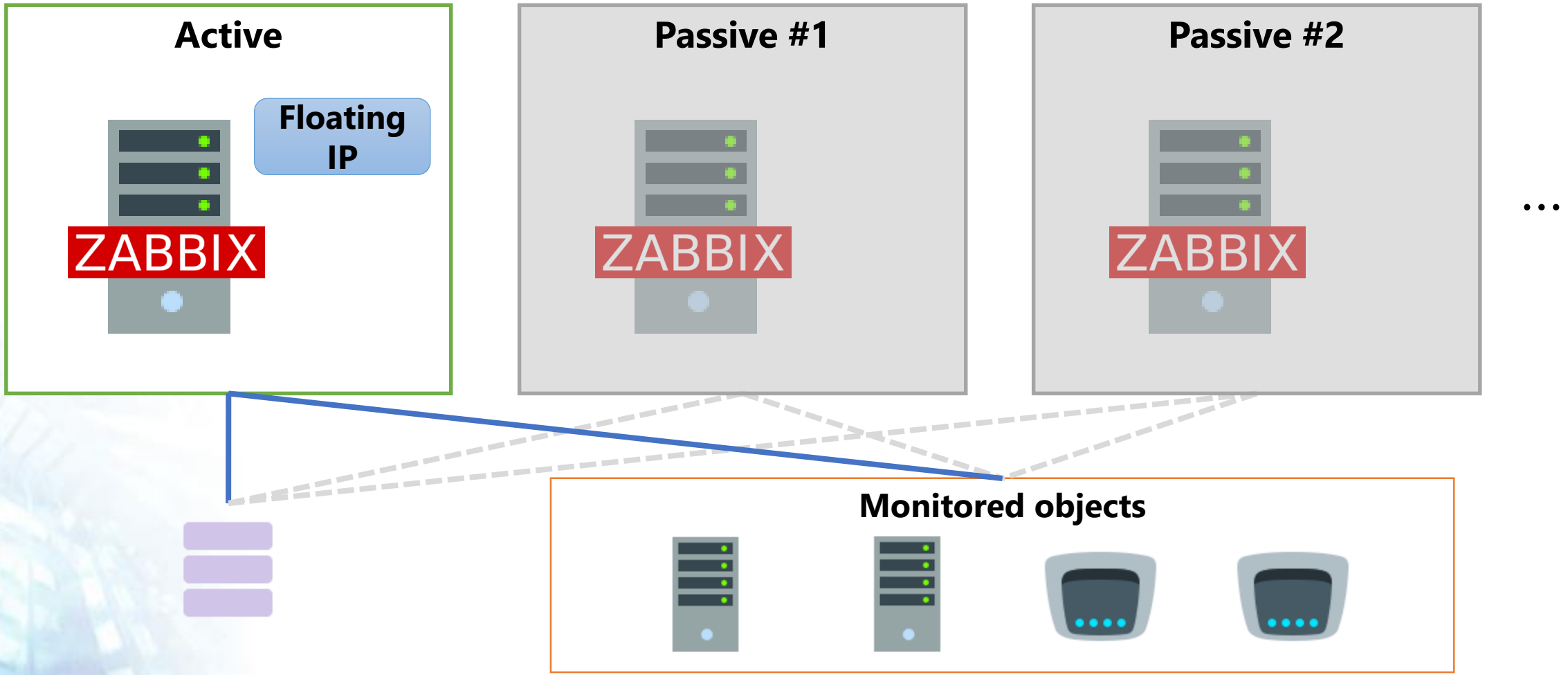
- Simple construction method
- Continuous monitoring when primary is down

Cons

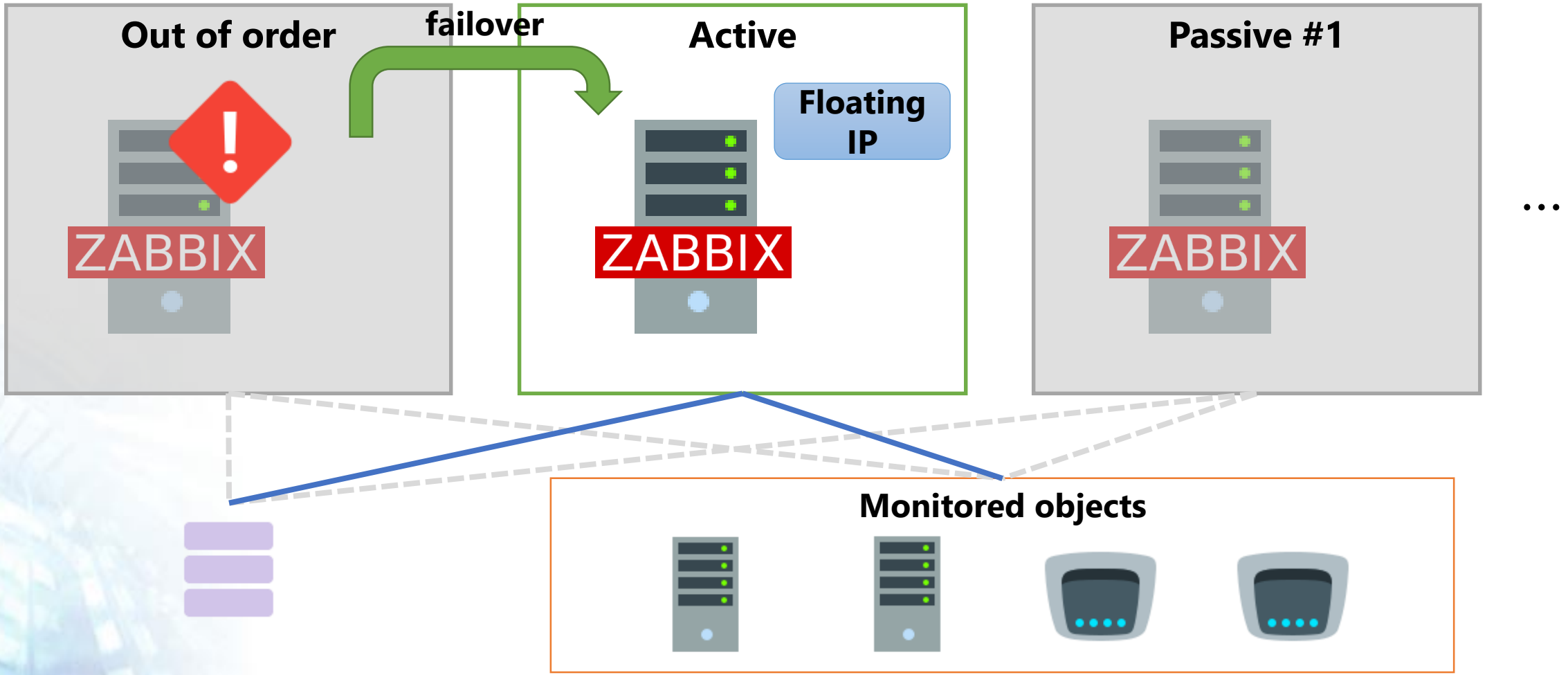
- Load of monitored object is higher than active/passive
- Need to consider how to sync configuration and switch method

Zabbix Active/Passive HA Cluster

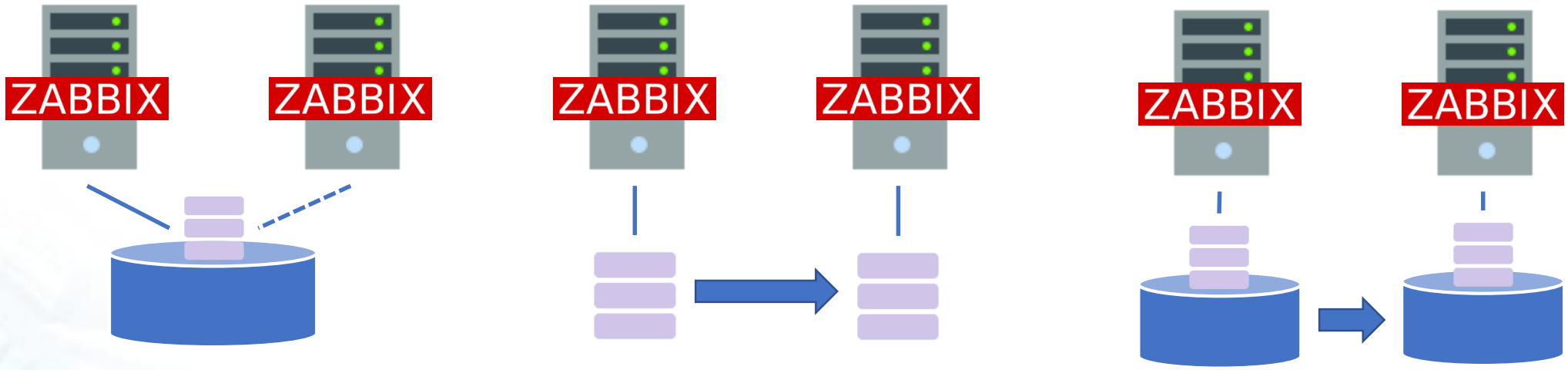
Zabbix Active/Passive HA cluster



Zabbix Active/Passive HA cluster



DB redundant method

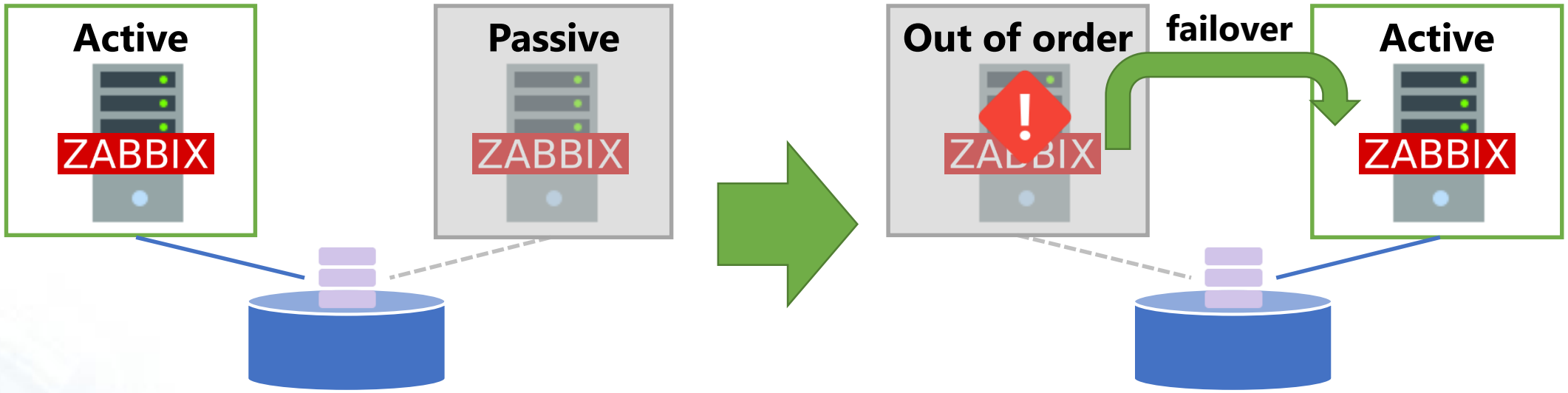


Shared storage

DB replication

Block device replication

DB redundancy with shared storage



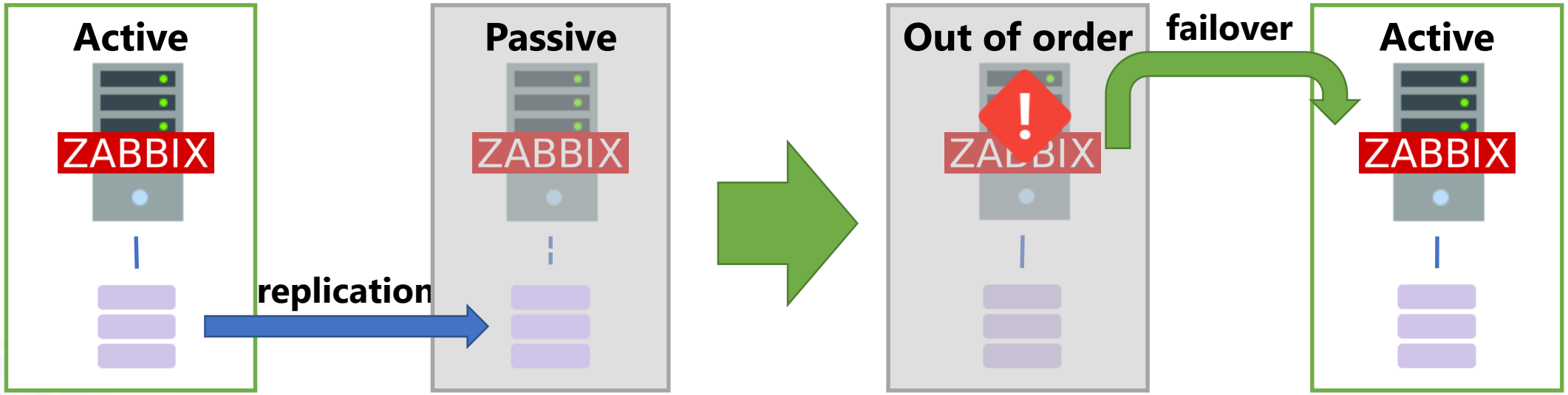
Pros

- No overhead by data sync
- Rapid failover without data loss

Cons

- Need specific device
- Storage device becomes single point of failure

DB redundancy with DB replication



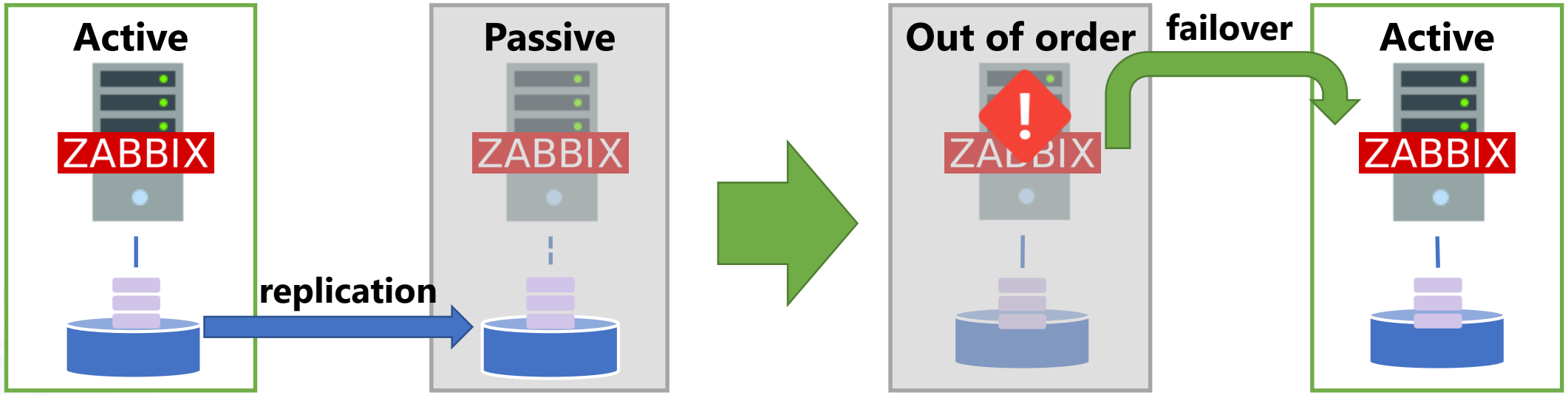
Pros

- Failover is relatively rapid
- Redundancy complete with DB feature only

Cons

- High technical cost
- Replication overhead

DB redundancy with block device replication



Pros

- Simple architecture
- Low monetary cost

Cons

- Taking time to failover
- Storage size increase by number of Zabbix servers

Considerations in building Active/Passive cluster

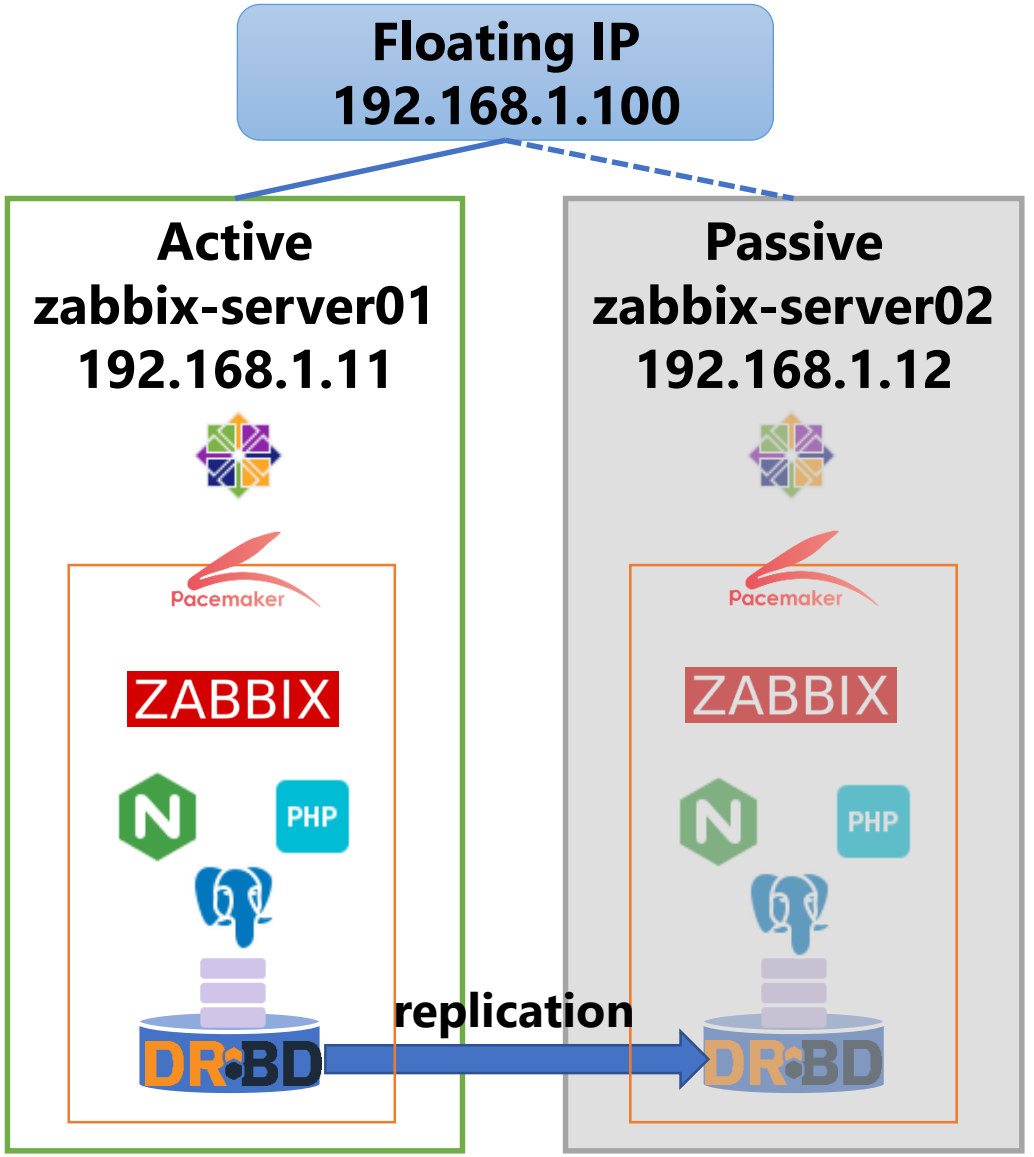
- How to prevent split-brain syndrome
  - Watchdog
  - STONITH (Shoot The Other Node In The Head)

- If monitoring SNMP trap or log files on Zabbix server, put log files on shared storage or replicated block device
  - Otherwise file re-read is occurred after failover

Active/Passive Cluster Setting Example

Environment

- OS: CentOS 8
- Software:
  - Zabbix 5.0.4
  - PostgreSQL 10.14-1
  - Web server software
    - Nginx 1.14.1-9
    - PHP-FPM 7.2.24-1
  - Cluster software
    - Pacemaker 2.0.3-5
    - Corosync 3.0.3-2
  - Block device replication software
    - DRBD 9.0.23



Zabbix setting

- Install Zabbix repository in both nodes

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

- Install Zabbix server and frontend in both nodes

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

- Edit Zabbix server's SourceIP to floating IP address in both nodes

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

Nginx and PHP-FPM setting

- Install Nginx in both nodes

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

- Edit Nginx conf file in both nodes

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

- Edit PHP-FPM conf file in both nodes

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

Pacemaker and Corosync setting

- Install Pacemaker and Corosync in both nodes

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

- Setup host name and IP address in both nodes

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

Pacemaker and Corosync setting

- Authorize cluster nodes

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

- Setup cluster

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

Pacemaker and Corosync setting

- Start cluster

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

- Disable STONITH and quorum policy

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

Pacemaker and Corosync setting

- Check cluster status

```
[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 setting

- Install DRBD in both nodes

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

DRBD setting

- Setup DRBD resource in both nodes

```
# 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 setting

- Create DRBD metadata

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

- Start DRBD

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

DRBD setting

- Check DRBD status

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

DRBD setting

- Sync DRBD

```
[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 setting

- Make filesystem and mount point

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

- Mount filesystem

```
[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 setting

- Install PostgreSQL in both nodes

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

- Make DB data directory

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

PostgreSQL setting

- Initialize DB data directory and start PostgreSQL

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

- Make Zabbix DB

```
[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
```

Resource setting

- Put filesystem and PostgreSQL under Pacemaker/Corosync control

```
[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
```

Resource setting

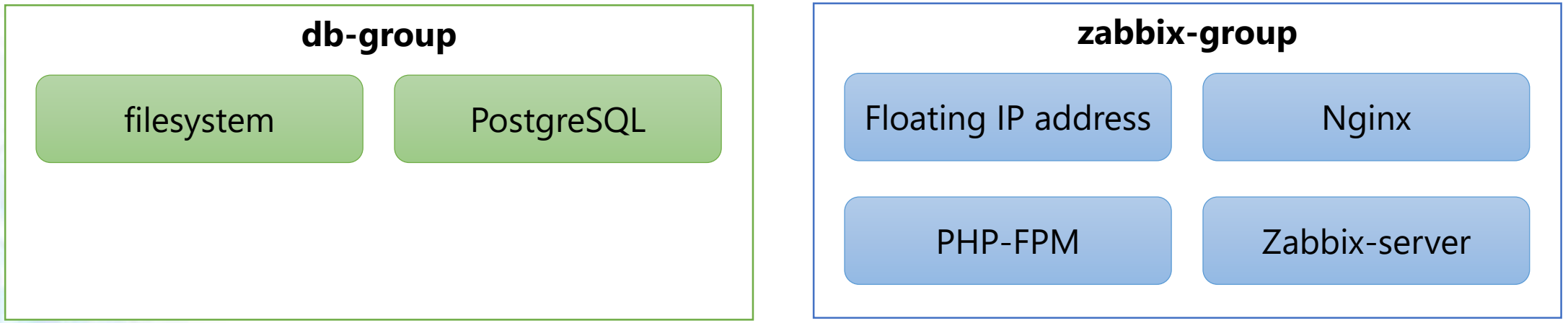
- Put floating IP address, Nginx and PHP-FPM under Pacemaker/Corosync control

```
[zabbix-server01] # pcs resource create fip 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
```

Resource setting

- Put Zabbix server under Pacemaker/Corosync control

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



Resource setting

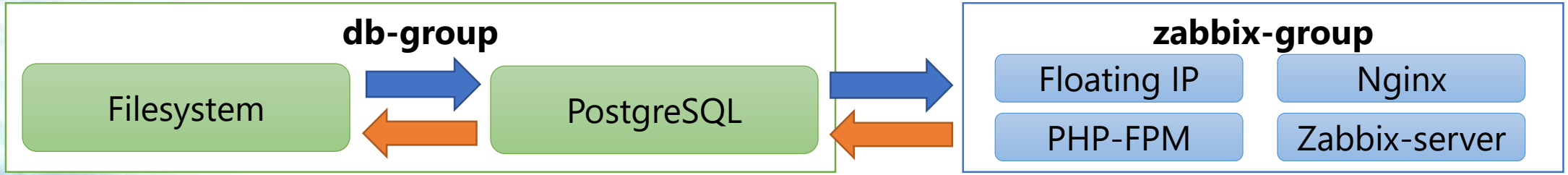
- Setup resource colocation constraint

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

- Setup resource order constraint

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

Resource start/stop order



Resource setting

- Check resources status

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

Full List of Resources:
  * Resource Group: zabbix-group:
    * fip       (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
…
```

Resource setting

- Check resource constraint

```
[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:
```

Resource setting

- Test resource failover

```
[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
```

For more information

- 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/

Thank you!