

# Summary

Currently, Zabbix does not support High Availability (HA) operations out of the box. Zabbix is known to work in HA mode with the help of third-party HA clustering solutions (such as Pacemaker) but the barrier to entry (high learning curve for HA software, complicated network settings, error-prone cluster configuration) can be too high for small to medium Zabbix installations.

Zabbix should offer a simple but reliable native solution that would make HA installation easy to set up and operate.

It's understood that the simplicity aspect of the native Zabbix HA solution has its trade-offs and may not cover all HA requirements, especially for very big and complex installations. Thus, the native solution should be opt-in and not interfere with the third-party clustering software, should the user decide to go their own HA route.

## Use cases

1. I want my Zabbix server to be protected against hardware and software failures.
2. I want to minimize Zabbix downtime during software and hardware maintenance.
3. I want to be able to switch between HA and non-HA modes of operation when needed.

## Zabbix acceptance

1. Zabbix must provide an HA solution with the following features and requirements:
  - a. the HA solution must be opt-in
  - b. the HA solution must consist of multiple `zabbix_server` instances (nodes)
    - i. every node is configured separately (config file, scripts, encryption, data export)
    - ii. every node uses the same database
    - iii. every node has two modes of operation: active and standby
    - iv. only one node can be active (working) at a time
    - v. standby (inactive) node must:
      1. not do data collection, processing, or any other "normal" server activity; not listen ports
      2. keep the number of active database connections to a minimum
      3. verify `zabbix_server.conf` parameters at the start
    - vi. nodes must be compatible across minor Zabbix versions (i.e. no breaking changes in HA between minor versions)
  2. HA configuration must be introduced:
    - a. new optional parameter **node name** in `zabbix_server.conf`
    - b. new parameter **failover delay** in the 'config' table with the default value 60 seconds
      - i. **failover delay** must not be less than 10 seconds
    - c. new parameter **HA status** (true/false) in the 'config' table with the default value false
    - d. new **server nodes** table in the database:
      - i. list of server nodes; each record has attributes:
        1. **node name** (corresponds to **node name** from `zabbix_server.conf`)
        2. **address** (IP/host, port)
        3. **last access time**
        4. **node status** (active/standby/stopped)
  3. Zabbix server changes:
    - a. on startup
      - i. if **node name** is not provided in `zabbix_server.conf`:
        1. assume empty string as **node name**
        2. print error message and exit if **HA status** is true
        3. delete all records in **server nodes** table if **HA status** is false
        4. add a record with empty **node name** to **server nodes** table
          - a. set **address** to first entry of 'ListenIP' + 'ListenPort' from `zabbix_server.conf`
      - ii. if **node name** is provided in `zabbix_server.conf`:
        1. print error message and exit if **HA status** is false
        2. add a record to **server nodes** table if the **node name** doesn't exist in the table:
          - a. **address** must be set to the first entry of 'ListenIP' + 'ListenPort' from `zabbix_server.conf`
          - b. **node status** set to standby if there is a record with active status, set to active otherwise
        3. read the record with **node name** from **server nodes** table if the record exists, set the mode of operation to either active or standby

- b. during runtime
    - i. active and standby nodes must update their **last access time** every 5 seconds
      - 1. **last access time** must be updated by SQL timestamp function (e.g. `unix_timestamp()` for MySQL)
    - ii. active node must monitor its status in **server nodes** table every 5 seconds
      - 1. if the node became standby in **server nodes** table, it must:
        - a. stop writing to the database immediately, stop all processing and switch to standby mode
    - iii. standby node must monitor **last access time** of the active node in the database every 5 seconds
      - 1. if the **last access time** is  $>$  **failover delay** seconds in the past the standby node must switch itself to active mode (failover) and:
        - a. update the list of nodes accordingly (by assigning standby status to previously active node)
        - b. log failover operation to Zabbix server log
    - iv. standby node must log the message 'this server is running in standby mode' every hour to Zabbix server log
    - v. all nodes must re-query **failover delay** parameter periodically from the 'config' table
  - c. on shutdown
    - i. a node must set **node status** to 'stopped' as the last step of the shutdown procedure
  - d. protection against the following situations must be provided:
    - i. when multiple standby nodes switch to the active mode simultaneously (e.g. by table lock)
    - ii. when an active node is either overloaded or experiencing database connectivity problems and thus is unable to complete write operations before another node takes over (one possible solution is to set up watchdog/alarm to control if **last access time** update completed ok and within a reasonable time; e.g. if a **failover delay** set to 60s then we must complete in  $<$  50s)
    - iii. the sudden time change on the database server (a possible solution is to detect clock skew in the database by comparing current and previous timestamps in the **server nodes** table with a real (5 seconds) time interval between updates)
  - e. new command-line options for `zabbix_server` binary must be introduced:
    - i. 'show HA status and nodes' - to show the list of nodes with associated attributes and **HA status** (on/off)
      - 1. **last access time** for nodes must be displayed both in human-readable local time format and as time difference (e.g. "last seen 2m30s ago")
    - ii. 'remove node reference' - remove the specific node from **server nodes** table. The command must remove the node only if either:
      - 1. the node is in the stopped state
      - 2. or the node is stale (stale node has current time - **last access time**  $>$  **failover delay** + 1 minute)
    - iii. 'change failover delay' - change **failover delay** parameter in the 'config' table
    - iv. 'set HA status' - set **HA status**. **HA status** can be only changed if all nodes are either stopped or stale.
  - f. internal check '`zabbix[cluster,discovery,nodes]`' must be added:
    - i. it must be only supported by the server
    - ii. it must return JSON with all node parameters
  - g. remote monitoring check '`zabbix[stats,{$ADDRESS},{$PORT}]`' must be changed to include the same information as in '`zabbix[cluster,discovery,nodes]`'
4. Zabbix proxy changes:
  - a. Proxy must be modified to allow multiple addresses for 'Server' parameter of `zabbix_proxy.conf` in active mode
    - i. Proxy must try to establish the connection to every address from the list until the connection succeeds. Once the connection succeeded the proxy must stick with this address until this connection fails.
    - ii. Proxy must be able to use one TLS certificate for all addresses in the list
5. Zabbix agent and agent2 (agents) changes:
  - a. 'ServerActive' config file parameter must support a list of cluster node addresses as a single entry in the list of servers
    - i. agents must connect to every address from the list of cluster nodes until the connection succeeds. Once the connection succeeded the agent must stick with this address until this connection fails.
    - ii. the change to 'ServerActive' parameter must be backward-compatible with the current format
6. Template changes
  - a. by using `zabbix[cluster,discovery,nodes]` template 'Zabbix server' must:
    - i. provide LLD rule with item and trigger prototypes for node discovery
      - 1. triggers on node status change
    - ii. support metrics for each node:
      - 1. address
      - 2. last access time
      - 3. number of seconds between database `unix_timestamp()` and last access time

4. node status
  - b. by using modified 'zabbix[stats,{\$ADDRESS},{\$PORT}]' template 'Remote Zabbix server' must provide remote monitoring of cluster nodes similarly to 'Zabbix server'
7. Front-end changes:
  - a. System information widget must be updated to display:
    - i. **HA status** (on/off)
    - ii. If HA status is on then display:
      1. the list of cluster nodes from **server nodes** table (**last access time** must be displayed both in human-readable local time format and as time difference)
      2. **failover delay** parameter
  - b. Reports → System information must be updated to display the same HA information as System information widget
8. API changes:
  - a. API method to get the list of nodes must be added

## Nonfunctional requirements

1. N/A

## Decisions made

1. Very big installations can be slow to shut down. The slow failover process is not addressed by this ACC.
2. High availability for Zabbix database is out of scope.
3. High availability for Zabbix proxies is out of scope.
4. All nodes are equal, node priorities are not supported.
5. Every node in HA setup must be started and shut down manually.
6. Heartbeat times (last access time update interval) are not configurable.
7. List of IP addresses (Server parameter of zabbix\_proxy.conf, ServerActive for zabbix\_agentd.conf) for active proxies and agents must be maintained manually.
8. When a node detects slow database performance or experiences database connectivity problems, no steps to stop data collection must be taken.
9. No database details must be shown when displaying the list of cluster nodes.
10. No dedicated authentication mechanisms for nodes, as they need to possess the database credentials anyway.

## Open questions

1. N/A

## Documentation

1. The new chapter "High Availability Cluster" must be created in Zabbix documentation. This chapter must include the following:
  - a. General information about HA support in Zabbix:
    - i. How HA works in Zabbix
    - ii. What are the limitations of HA support in Zabbix
  - b. How to enable and disable HA
  - c. How to upgrade server when running in HA mode

## Changes log

- N/A