ZABBIX processes

/usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf
/usr/sbin/zabbix_server: configuration syncer [synced configuration in 0.169800 sec, idle 60 sec]
/usr/sbin/zabbix_server: alerter #2 started
/usr/sbin/zabbix_server: housekeeper [deleted 105688 hist/trends, 7874 items/triggers ...]
/usr/sbin/zabbix_server: timer #1 [updated 0 hosts, suppressed 0 events in 0.000589 sec, idle 59 sec]
/usr/sbin/zabbix_server: http poller #1 [got 0 values in 0.000610 sec, idle 5 sec]
/usr/sbin/zabbix_server: discoverer #1 [processed 1 rules in 0.004817 sec, idle 60 sec]
/usr/sbin/zabbix_server: history syncer #4 [processed 136 values, 92 triggers in 0.008205 sec, idle 1 sec]
/usr/sbin/zabbix_server: escalator #1 [processed 0 escalations in 0.000745 sec, idle 3 sec]
/usr/sbin/zabbix_server: java poller #1 [got 0 values in 0.000002 sec, idle 5 sec]
/usr/sbin/zabbix_server: snmp trapper [processed data in 0.000014 sec, idle 1 sec]
/usr/sbin/zabbix_server: proxy poller #1 [exchanged data with 0 proxies in 0.000003 sec, idle 5 sec]
/usr/sbin/zabbix_server: self-monitoring [processed data in 0.000007 sec, idle 1 sec]
/usr/sbin/zabbix_server: vmware collector #1 [updated 0, removed 0 VMware services in 0.000001 sec, idle 5 sec]
/usr/sbin/zabbix_server: task manager [processed 0 task(s) in 0.000002 sec, idle 5 sec]
/usr/sbin/zabbix_server: poller #1 [got 27 values in 0.303246 sec, idle 1 sec]
/usr/sbin/zabbix_server: unreachable poller #1 [got 0 values in 0.000003 sec, idle 5 sec]
/usr/sbin/zabbix_server: trapper #1 [processed data in 0.001295 sec, waiting for connection]
/usr/sbin/zabbix_server: icmp pinger #1 [got 0 values in 0.000004 sec, idle 5 sec]
/usr/sbin/zabbix_server: alert manager #1 [sent 0, failed 0 alerts, idle 5.007591 sec during 5.007593 sec]
/usr/sbin/zabbix_server: preprocessing manager #1 [queued 0, processed 236 values, idle 5.031538 sec during 5.032590 sec]
/usr/sbin/zabbix_server: preprocessing worker #1 started
It is already confusing? No?! Now, let's discuss Zabbix processes
Legend

- database (DB)
- internal process
- process interacting with outside
- cache/memory
- group of processes
- Zabbix component
- can have multiple parallel processes
1. read configuration file
2. load modules
3. allocate memory for caches
4. check DB (e.g. version)
5. get configuration data (hosts, templates, items, triggers, etc.)
6. put configuration into cache
7. start other processes

```
[root@zabbix ~]# systemctl start zabbix-server
```
Configuration syncer

- gets all configuration data from DB (e.g. hosts, items, triggers, etc.) and puts it into "Configuration cache".
- updates item queue (next check)
- then every 1 minute checks for updates and performs incremental changes in the cache (if any)
Self-monitoring

- All processes use internal monitoring cache to keep status (busy, idle).
- Self monitoring reads statuses and calculates busy/idle.
- Poller processes this data
- internal monitoring cache has no information about other caches (e.g. free).
- Poller uses cache allocators to get cache usage.
- Caches have statistics (e.g. history cache - received values).
- Poller may use DB to get number of historical values (there is an internal item).
Data gathering

- obtains item values/metrics and delivers to data pre-processing
- updates host availability (cache and DB)
- manages LLD* by creating entities, links to templates directly in DB

*LLD = Low Level Discovery
Data gathering
Data gathering - trappers

active check Trapper
- listening on TCP/10051 port by default
- receives data from:
  - active Zabbix agents
  - active proxies
  - Zabbix-sender utility (special item type)

### Option: StartTrappers=5

SNMP trapper
- direct interaction with the DB and Configuration cache
- uses snmptrapd to cache traps
  - perl_script (my Zabbix Devs)
  - snmptt

### Option: StartSNMPTrapper=0
Data gathering – http poller

- passive check (done by server/proxy)
- uses cURL (libcurl)
- retrieves http/https page headers/body
- http response code
- emulates browser
- do not require any agent running on a host
- can follow redirects (hard-coded to 10 (using cURL option CURLOPT_MAXREDIRS))

### Option: StartHTTPPollers=1
Data gathering – icmp pinger

performs so called: **Simple checks**
- icmpping[<target>,<packets>,<interval>,<size>,<timeout>]
- icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>]
- icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>, <mode>]

uses fping and fping6 system utility
by default evaluates 3 pings

### Option: StartPingers=1
Data gathering - unreachable poller

- processes items for unreachable or unavailable hosts.
- pollers, ICMP pingers, IPMI pollers and Unreachable pollers do direct DB access in both directions for unsupported items.

### Option: StartPollersUnreachable=1
### Option: UnreachablePeriod=45
### Option: UnavailableDelay=60
### Option: UnavailableDelay=15
Data gathering – java poller

- needs Zabbix-java-gateway
- Native support for monitoring JMX applications
- java app must be started with:
  - `Dcom.sun.management.jmxremote`

### Option: `JavaGateway=<IP/DNS>`
### Option: `StartJavaPollers=0`
Data gathering – ipmi manager

- uses ipmi pollers to get data
- manager uses configuration (queue)
- keeps a track of which poller was used to poll a device
- execution of ipmi script:
  - frontend -> trapper -> ipmi manager

### Option: StartIPMIPollers=0
Data gathering - poller

Processes checks for item types:
- zabbix agent (passive)
- external
- SNMP
- simple checks
- internal
- aggregate
- calculated
- ssh
- telnet
- DB monitoring

### Option: StartPollers=5
Data gathering - poller

vmware collector

vmware cache

value cache

internal monitoring cache

configuration cache

poller

database

get item configuration
update host availability

cache VMware data

generate VMware metrics

get values for aggregate and calculated checks

update host availability
process low level discovery

get self-monitoring metrics
Data gathering – vmware collector

- gets data from VCenter/Hypervisor using API
- uses VMware cache
- controlled by settings in Zabbix_server.conf
  
  ### VMwareFrequency=60 #config data
  ### VMwarePerfFrequency=60 #performance data

- grab all available information from VCenter
- poller extracts needed data -> pre-processing
- if something is missing -> task -> retrieved during next fetch cycle.
Proxy communication manages active and passive proxy communications:
- pushes or delivers configuration data for proxies
- retrieves values/metrics
- updates host availability
- updates proxy heartbeat for internal monitoring
- auto registration
- LLD processing
Proxy communication

configuration cache
- get item configuration data
- update host availability
- update proxy statistics

get proxy configuration (hosts, items, ...)
- get remote tasks
- update host availability
- create auto registration events
- create discovery events
- create remote task results
- perform low level discovery

proxy poller
- send: configuration, tasks
- receive: historical data, auto registration data, discovery data, host availability data, tasks
- get: configuration, tasks

trapper
- send: historical data, auto registration data, discovery data, host availability data, tasks

data pre-processing
- send values

Zabbix proxy (passive)
- send: configuration, tasks
- receive: historical data, auto registration data, discovery data, host availability data, tasks

Zabbix proxy (active)
- send: historical data, auto registration data, discovery data, host availability data, tasks

database
data gathering processes don't use DB or history cache to write metrics

everything is sent to "preprocessing manager"
Data pre-processing

- converts/transforms values according to defined pre-processing steps
  - trim
  - regex
  - json-path
  - xml-path
  - multiply
  - other...

### Option: StartPreprocessors=3
Data pre-processing

**preprocessing worker**
- sends value, pre-processing steps
- get result

**configuration cache**
- get pre-processing steps
- get internal items
- get dependent items

**history cache**
- store value/error

**data gathering**
- queue values to pre-process

**preprocessing manager**
- queue dependent items
Data pre-processing

- All incoming values get to preprocessing manager queue (itemid and value)
- uses preprocessing workers to apply steps
- re-queues dependent items with the received master item value
- Incoming data rule: fist in – first out
- can handle >100 000 NVPS (up to 300k)

### Option: StartPreprocessors=3

More details: https://www.zabbix.com/documentation/4.0/manual/appendix/items/preprocessing
- get up to 1000 values from history cache (uses index)
- flushes pre-processed values to database
- calculates and updates trends
- updates host inventory
- calculates triggers including those with time based functions (uses value cache)
- generates, correlates and flushes events
- processes actions and creates escalations
- creates IT service alarms
- exports data
History syncer
Incident management

- processes escalations
- executes remote commands
- sends notifications
Incident management

- **host**
  - execute command via ssh/telnet
- **shell**
  - execute alert script
  - execute command in shell
- **SMTP server**
  - send email
  - send alert
  - send alert get result
- **GSM modem**
  - send SMS
- **alerner**
  - custom alert script
- **alert manager**
  - get message alerts
  - get database error notification targets
  - update alerts
  - get alert
- **escator**
  - read escalations
  - update escalations
  - create alerts
  - create remote tasks
  - resolve macros
- **ipmi manager**
  - send ipmi command
  - resolve value macros
- **value cache**
- **configuration cache**
  - get item/host configuration data
  - resolve macros
- **external application**
- processes maintenance
- puts hosts in and out of maintenance
- handles event suppression
- reads tasks from DB
- handles manual problem closing
- expires remote commands
- processes remote command results
- performs problem acknowledge updates
- Handles ‘check now’ task by re-queuing item in configuration cache
Housekeeper

- removes old data
- settings: Administration -> General -> Housekeeper
- historical data (history, trends, events) are not immediately deleted.
- By default Housekeeper runs every hour

### Option: HousekeepingFrequency=1

/usr/sbin/zabbix_server: housekeeper [deleted 105573 hist/trends, 2304 items/triggers, 885 events, 0 sessions, 0 alarms, 0 audit items in 36.626682 sec, idle for 1 hour
- Listens on TCP/10051 (by default)
- data gathering
- proxy management tasks
- performs auto registration
- will create a new hosts directly in database (check all conditions, generate new event and process it).
- will execute frontend commands/scripts (e.g. ping, traceroute, etc).
- Queue (calculates by using configuration cache) – uses items next check to understand add it or not
Discoverer

- performs network discovery.
- works with DB directly (i.e. gets rules and creates hosts)

### Option: StartDiscoverers=1
DB down!

Be ready for that!

Administration -> General -> other:

User group for database down message

Zabbix administrators

Alerter caches users and will send “DB Down” message via all configured media types
Stop & main process

[root@zabbix ~]# systemctl stop zabbix-server

2. syncing history data
3. syncing trend data

1. stop other processes
Want to know more?

Documentation

<https://www.zabbix.com/documentation>

Look in to Zabbix sources

debug level increase

[root@zabbix ~]# zabbix_server -R log_level_increase=<process name>,<NR>

search the internet of course 😊
Questions?

Edmunds Vesmanis
Team ZABBIX

ZABBIX 2019 Conference
BENELUX
Thank you!

Edmunds Vesmanis
Team ZABBIX