3 Proxy

Overview

Zabbix proxy is a process that may collect monitoring data from one or more monitored devices and send the information to the Zabbix server, essentially working on behalf of the server. All collected data is buffered locally and then transferred to the Zabbix server the proxy belongs to.

Deploying a proxy is optional, but may be very beneficial to distribute the load of a single Zabbix server. If only proxies collect data, processing on the server becomes less CPU and disk I/O hungry.

A Zabbix proxy is the ideal solution for centralized monitoring of remote locations, branches and networks with no local administrators.

Zabbix proxy requires a separate database.

Note that databases supported with Zabbix proxy are SQLite, MySQL and PostgreSQL. Using Oracle or IBM DB2 is at your own risk and may contain some limitations as, for example, in return values of low-level discovery rules.

See also: Using proxies in a distributed environment

Running proxy

If installed as package

Zabbix proxy runs as a daemon process. The proxy can be started by executing:

```shell
shell> service zabbix-proxy start
```

This will work on most of GNU/Linux systems. On other systems you may need to run:

```shell
shell> /etc/init.d/zabbix-proxy start
```

Similarly, for stopping/restarting/viewing status of Zabbix proxy, use the following commands:

```shell
shell> service zabbix-proxy stop
shell> service zabbix-proxy restart
shell> service zabbix-proxy status
```

Start up manually

If the above does not work you have to start it manually. Find the path to the zabbix_proxy binary and execute:

```shell
shell> zabbix_proxy
```
You can use the following command line parameters with Zabbix proxy:

- `--config <file>`  
  path to the configuration file

- `--runtime-control <option>`  
  perform administrative functions

- `--help`  
  give this help

- `--version`  
  display version number

Runtime control is not supported on OpenBSD and NetBSD.

Examples of running Zabbix proxy with command line parameters:

```shell
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf
shell> zabbix_proxy --help
shell> zabbix_proxy -V
```

### Runtime control

Runtime control options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>config_cache_reload</td>
<td>Reload configuration cache. Ignored if cache is being currently loaded.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Active Zabbix proxy will connect to the Zabbix server and request configuration data.</td>
<td></td>
</tr>
<tr>
<td>housekeeper_execute</td>
<td>Start the housekeeping procedure. Ignored if the housekeeping procedure is currently in progress.</td>
<td></td>
</tr>
<tr>
<td>log_level_increase[=&lt;target&gt;]</td>
<td>Increase log level, affects all processes if target is not specified.</td>
<td>process type - All processes of specified type (e.g., poller)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See all proxy process types.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>process type,N - Process type and number (e.g., poller,3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pid - Process identifier (1 to 65535). For larger values specify target as 'process type,N'.</td>
</tr>
<tr>
<td>log_level_decrease[=&lt;target&gt;]</td>
<td>Decrease log level, affects all processes if target is not specified.</td>
<td></td>
</tr>
</tbody>
</table>

Example of using runtime control to reload the proxy configuration cache:

```shell
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R config_cache_reload
```

Example of using runtime control to trigger execution of housekeeper
Examples of using runtime control to change log level:

Increase log level of all processes:
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
log_level_increase

Increase log level of second poller process:
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
log_level_increase=poller,2

Increase log level of process with PID 1234:
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
log_level_increase=1234

Decrease log level of all http poller processes:
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R
log_level_decrease="http poller"

Process user

Zabbix proxy is designed to run as a non-root user. It will run as whatever non-root user it is started
as. So you can run proxy as any non-root user without any issues.

If you will try to run it as ‘root’, it will switch to a hardcoded ‘zabbix’ user, which must be present on
your system. You can only run proxy as ‘root’ if you modify the ‘AllowRoot’ parameter in the proxy
configuration file accordingly.

Configuration file

See the configuration file options for details on configuring zabbix_proxy.

Proxy process types

- configuration syncer - process for managing in-memory cache of configuration data
- data sender - proxy data sender
- discoverer - process for discovery of devices
- heartbeat sender - proxy heartbeat sender
- history syncer - history DB writer
- housekeeper - process for removal of old historical data
- http poller - web monitoring poller
- icmp pinger - poller for icmpping checks
- ipmi manager - IPMI poller manager
- ipmi poller - poller for IPMI checks
- java poller - poller for Java checks
- poller - normal poller for passive checks
- self-monitoring - process for collecting internal server statistics
- snmp trapper - trapper for SNMP traps
- task manager - process for remote execution of tasks requested by other components (e.g. close problem, acknowledge problem, check item value now, remote command functionality)
- trapper - trapper for active checks, traps, proxy communication
- unreachable poller - poller for unreachable devices
- vmware collector - VMware data collector responsible for data gathering from VMware services

The proxy log file can be used to observe these process types.

Various types of Zabbix proxy processes can be monitored using the `zabbix[process,<type>,<mode>,<state>]` internal item.

**Supported platforms**

Zabbix proxy runs on the same list of supported platforms as Zabbix server.

**Locale**

Note that the proxy requires a UTF-8 locale so that some textual items can be interpreted correctly. Most modern Unix-like systems have a UTF-8 locale as default, however, there are some systems where that may need to be set specifically.

From: https://www.zabbix.com/documentation/4.2/ - Zabbix Documentation 4.2

Permanent link: https://www.zabbix.com/documentation/4.2/manual/concepts/proxy

Last update: 2018/11/08 13:46