EXTENDED OUT OF THE BOX MONITORING WITH

AGENT 2

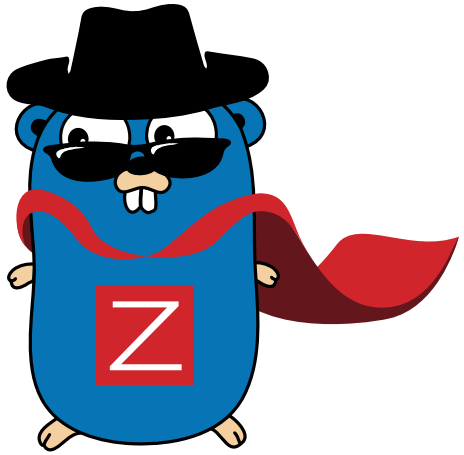
all our microphones are muted

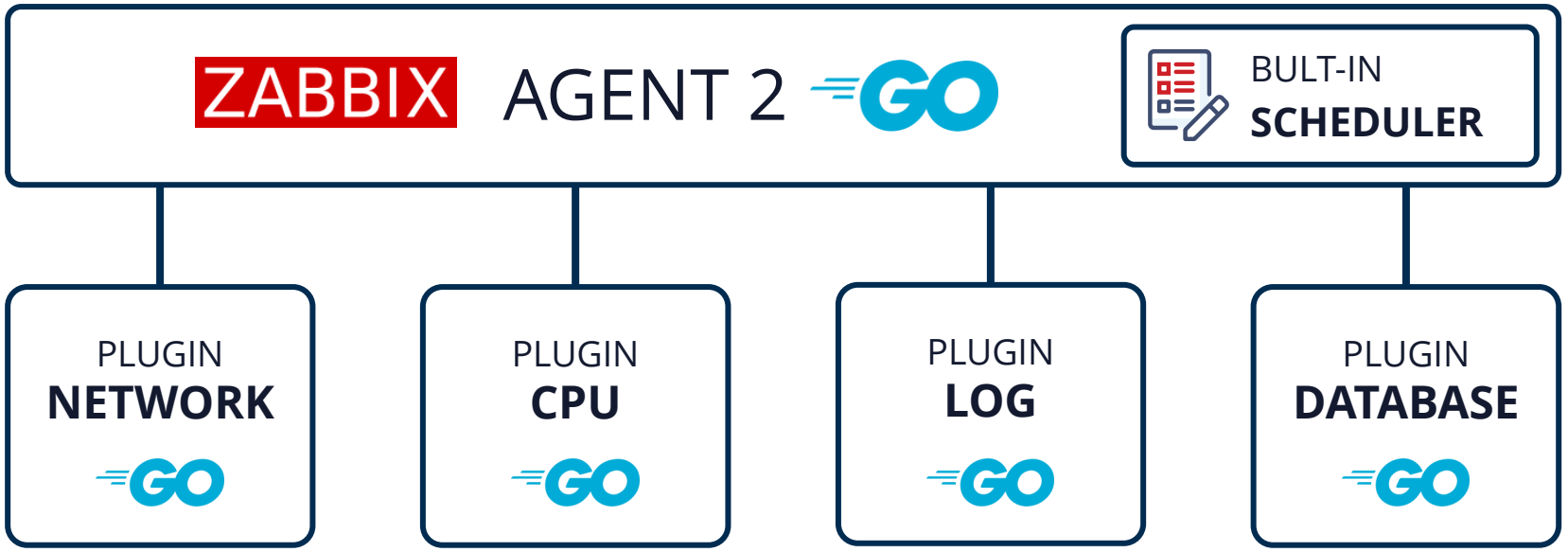
ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause

initMAX

AGENT 2 OVERVIEW





- scheduled/flexible intervals
- written in Go
- less TCP connections
- older configuration file support
- easily extendable
- out-of-the-box systemd monitoring

1

New Functionality

initMAX

Agent 2

NEW FUNCTIONALITY

Zabbix Agent 2 is written in Go (Golang)

Drop-in replacement for Zabbix agent

- Supports all previous functionality (same item keys)
- Supports old configuration file format

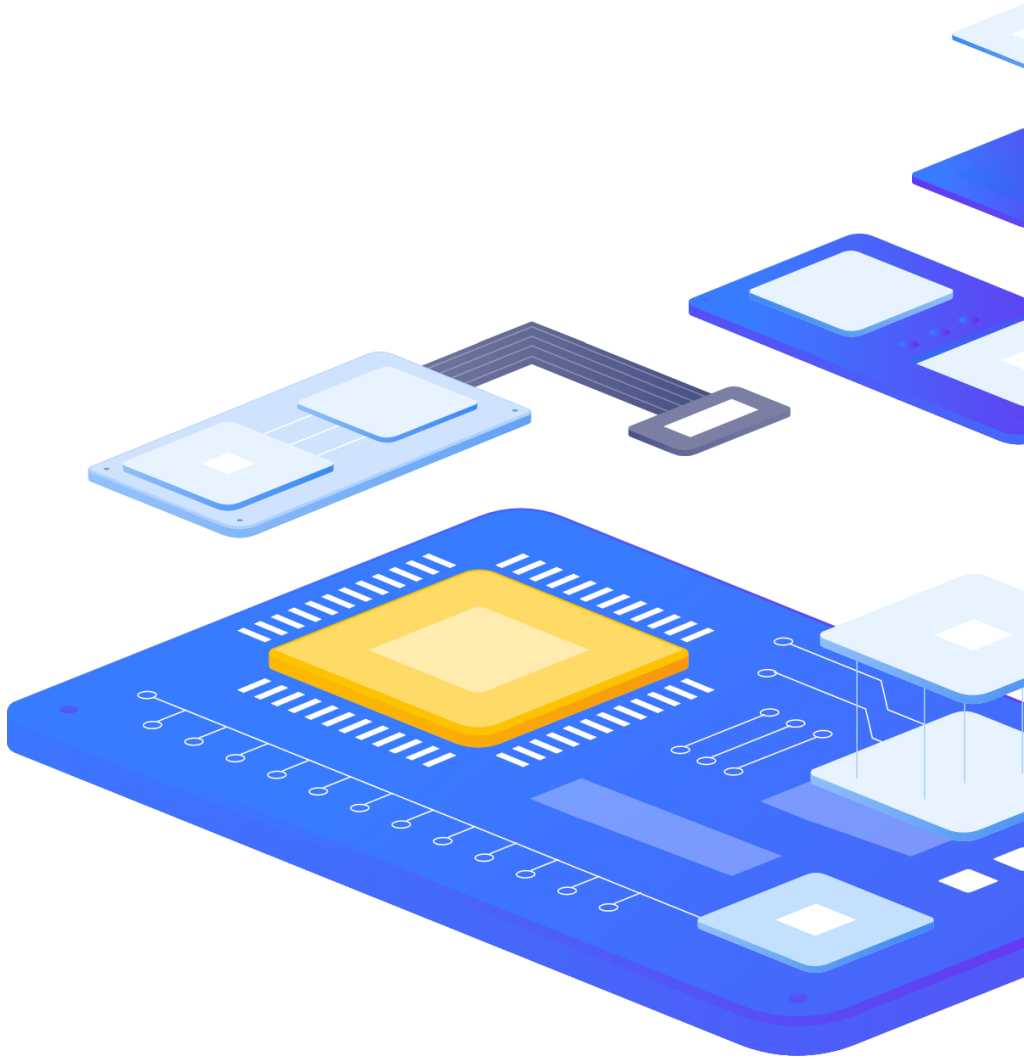
Standard items

Type: Zabbix agent (active)

Key	Name
agent.hostname	Agent host name. Returns string
agent.ping	Agent availability check. Returns nothing - unavailable; 1 - available
agent.version	Version of Zabbix agent. Returns string
eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]	Event log monitoring. Returns log
kernel.maxfiles	Maximum number of opened files supported by OS. Returns integer
kernel.maxproc	Maximum number of processes supported by OS. Returns integer
log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>]	Log file monitoring. Returns log

Has been developed to

- Reduce the number of TCP connections
- Be easily extendable with plugins



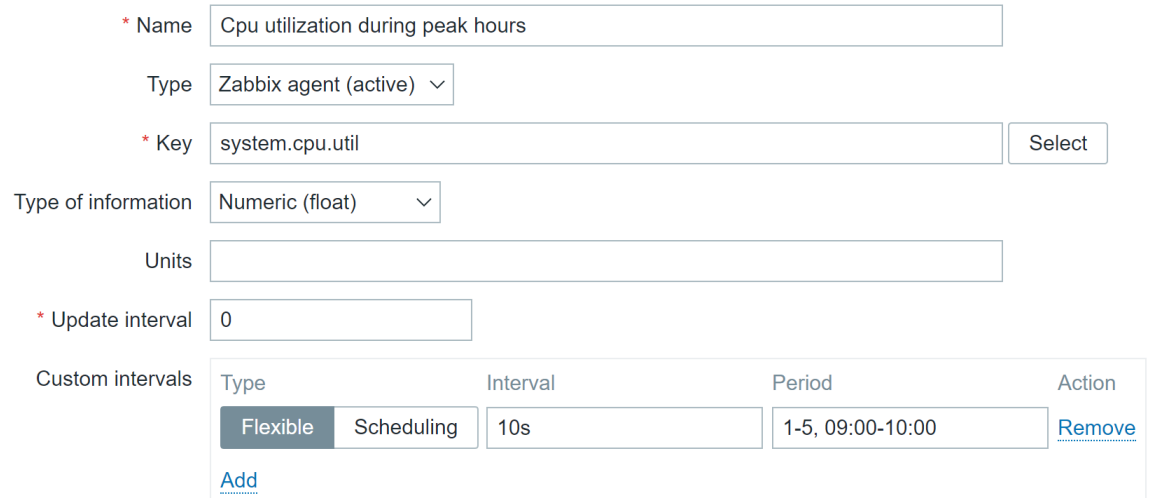
initMAX

Agent 2

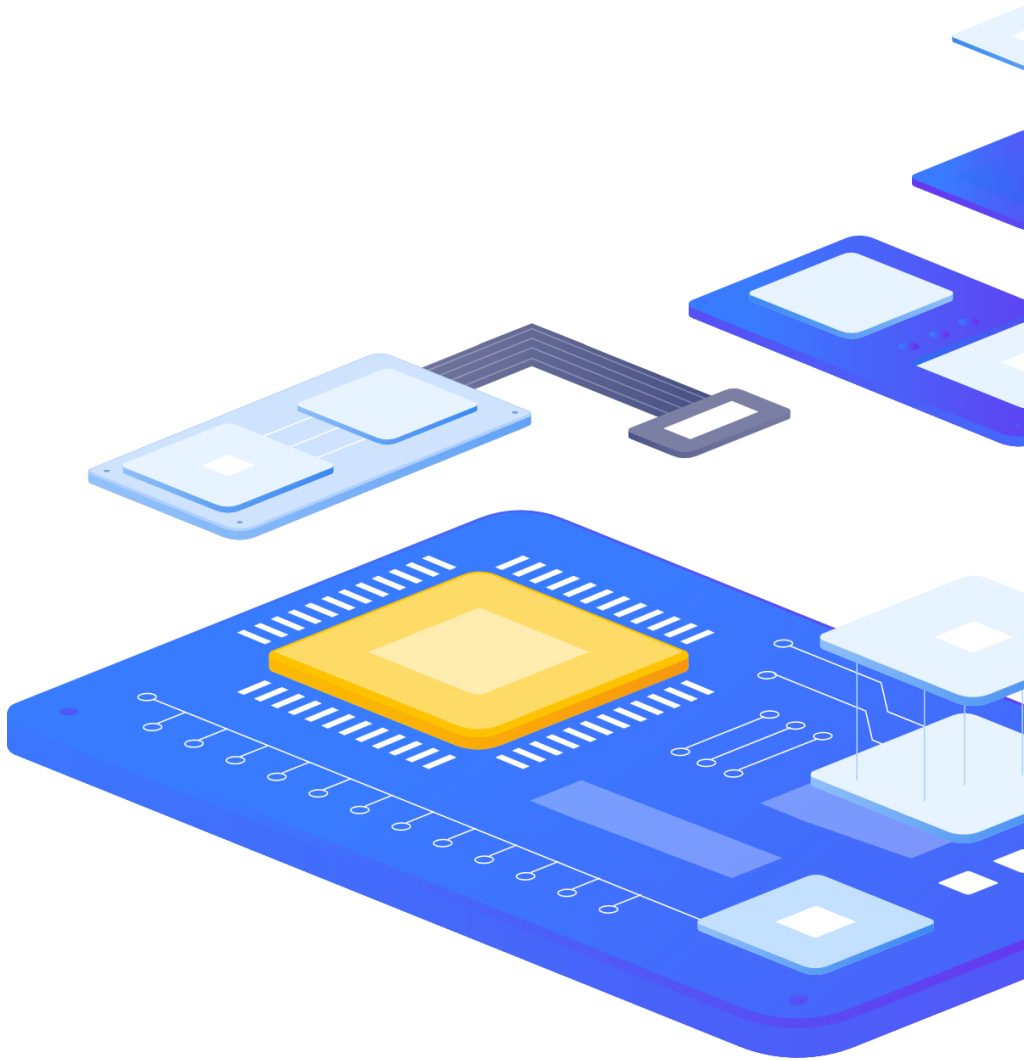
NEW FUNCTIONALITY

Improved active checks

- Active checks support scheduled/flexible intervals
- Parallel execution of multiple active checks for each ServerActive



- Limited system support



Agent 2

AGENT 2 BUFFER

Two types of buffer are supported by Agent2 in active mode

- Memory buffer
- Buffer using SQLite engine

Persistent buffer is supported both on Unix-like systems and Windows hosts

Configurable via Agent 2 configuration file

- Enable persistent buffer:

```
EnablePersistentBuffer=1
```

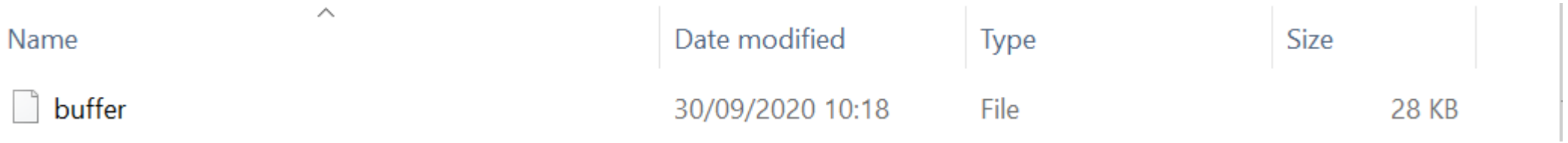
- Specify the location of the buffer file:

```
PersistentBufferFile=C:\temp\buffer
```

Agent 2

AGENT 2 BUFFER

The buffer file is created on agent startup in your specified location



Additionally, persistent buffer period can be configured

- Default is 1 hour

```
PersistentBufferPeriod=1h
```

Remember that persistent buffer is used only for active checks

If persistent buffer is disabled, then in memory buffer is used

Agent 2

AGENT 2 BUFFER

We can always take a look at what data is stored in the persistent buffer

```
sqlite> select * from data_1;
9|1601470098|31578|-1|-1|-1|491974656||-1|-1|-1|1601470098|461886
10|1601470099|31579|-1|-1|-1|47.336478||-1|-1|-1|1601470099|683697
11|1601470100|31580|-1|-1|-1|1039331328||-1|-1|-1|1601470100|605187
12|1601470155|31575|-1|-1|-1|859828224||-1|-1|-1|1601470155|973576
13|1601470156|31576|-1|-1|-1|100.000000||-1|-1|-1|1601470156|1253467
```

- Using persistent buffer adds a layer of redundancy
- If server for some reason is unreachable, the data is still stored in the persistent buffer
- The data is still preserved after the agent restart, as opposed to in memory buffer

Agent 2

ZABBIX 6.0 NEWS - agent/agent 2

agent.variant

- New Item for instaled variant since agent version 5.0.18

6.0 – HA settings

Count matching TCP/UDP sockets

- net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

Additional file information such as file owner and file permissions

- vfs.dir.get[dir,<regex_incl>,<regex_excl>,<types_incl>,…….]
- Parameter changes:
  - vfs.file.cksum[file,<mode>], vfs.file.get[file], vfs.file.owner[file,<ownertype>,<resulttype>]

Collect agent host metadata as a metric

- agent.hostmetadata

Supports loading stand-alone plugins without having to recompile the Zabbix Agent2.

2

Installation

Agent 2

AGENT 2 INSTALLATION

Install Zabbix Agent2

```
# dnf install https://repo.zabbix.com/zabbix/5.2/rhel/8/x86_64/zabbix-release-5.2-
1.el8.noarch.rpm
# dnf install zabbix-agent2
```

Start Zabbix Agent2

```
# systemctl start zabbix-agent2
```

Enable auto start

```
# systemctl enable zabbix-agent2
```

Agent 2

AGENT 2 INSTALLATION

Konfigurace

```
Server=<IP/DNS> for passive agent checks
```

```
ServerActive=<IP/DNS> for active agent checks
Hostname=Name of host for active checks
```

```
# systemctl restart zabbix-agent2
```

Agent 2

AGENT 2 INSTALLATION on windows – command line

- Available for download as archive or MSI install
- To install a single instance of Zabbix agent with the default configuration file c:\zabbix_agentd.conf:

```
zabbix_agent2.exe --install
```

- If you wish to use a configuration file other than c:\zabbix_agentd.conf, you should use the following command for service installation:

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

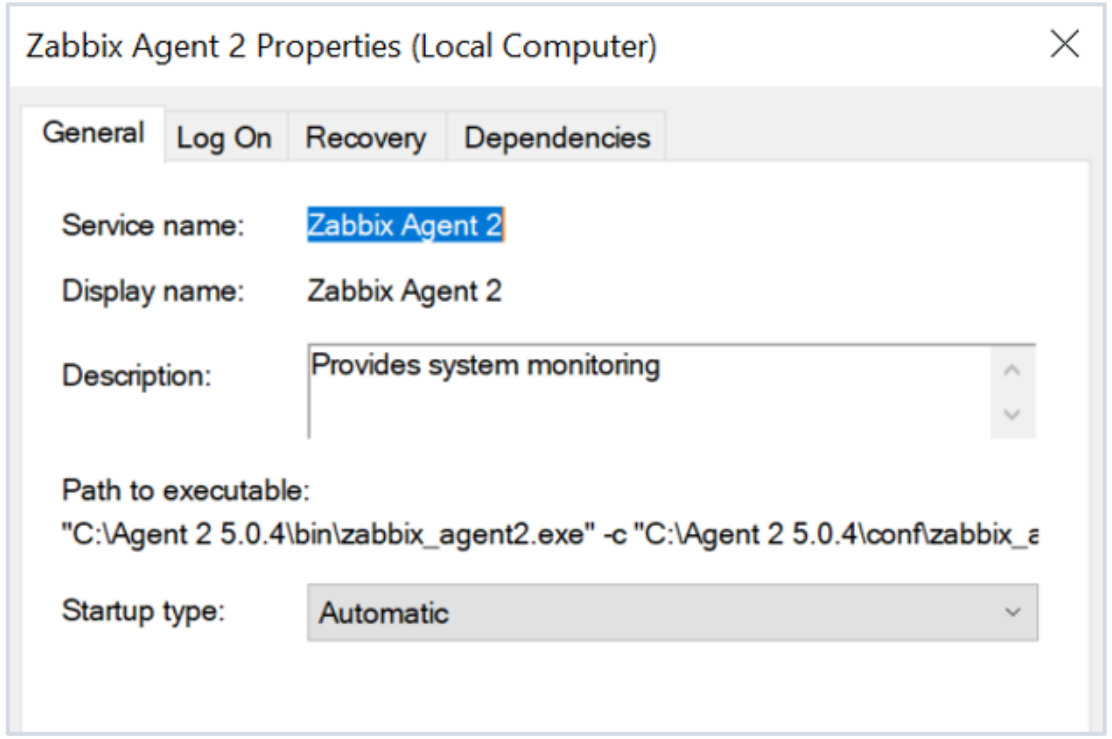
```
"C:\Agent 2 6.0.4\bin\zabbix_agent2.exe" --install --config "C:\Agent 2
6.0.4\conf\zabbix_agent2.conf"
```

Agent 2

AGENT 2 INSTALLATION on windows – command line

Once installed, runs as a service under Local System account

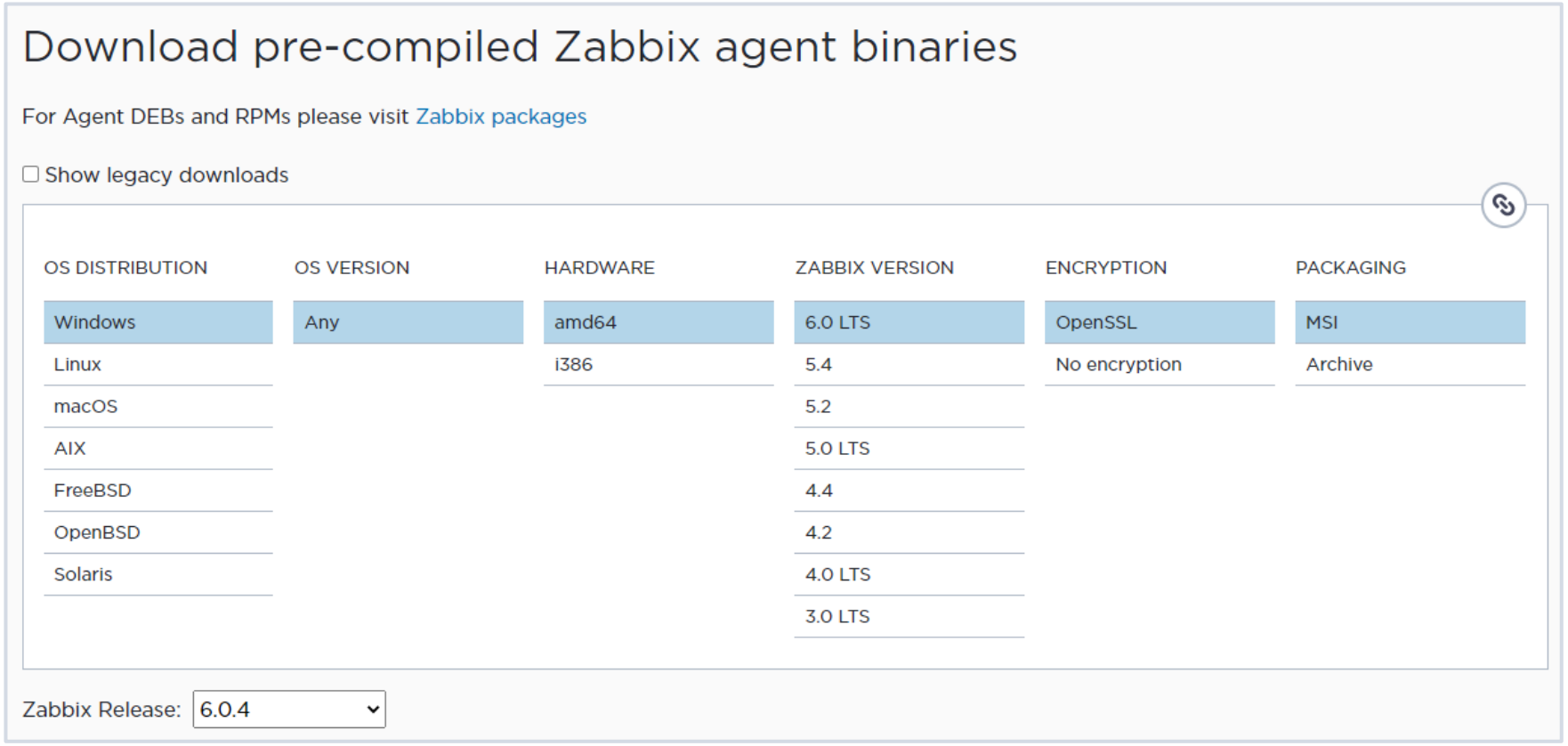
```
zabbix_agent2 [20268]: 'Zabbix Agent 2' installed succesfully
```



Agent 2

AGENT 2 INSTALLATION on windows – MSI

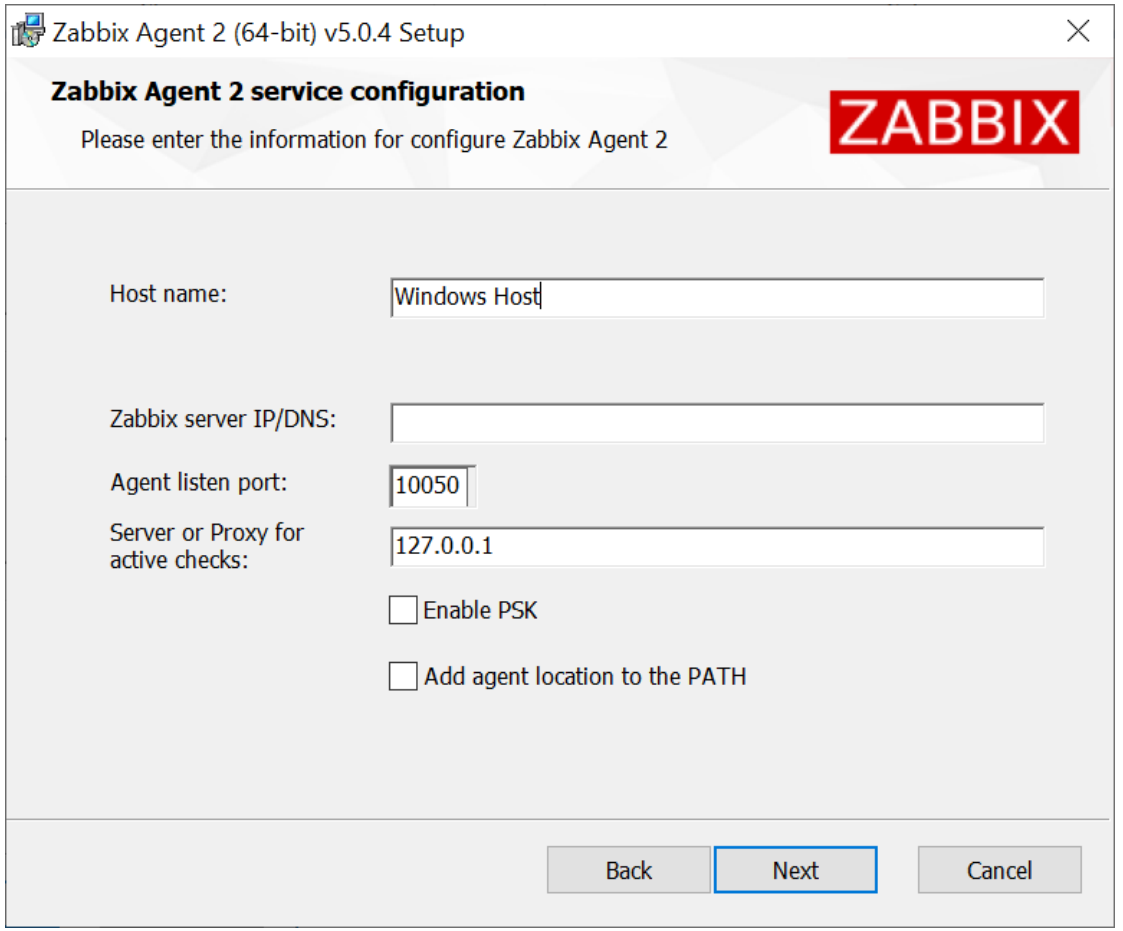
MSI Installer can be downloaded from Zabbix website



initMAX

AGENT 2 INSTALLATION on windows – MSI

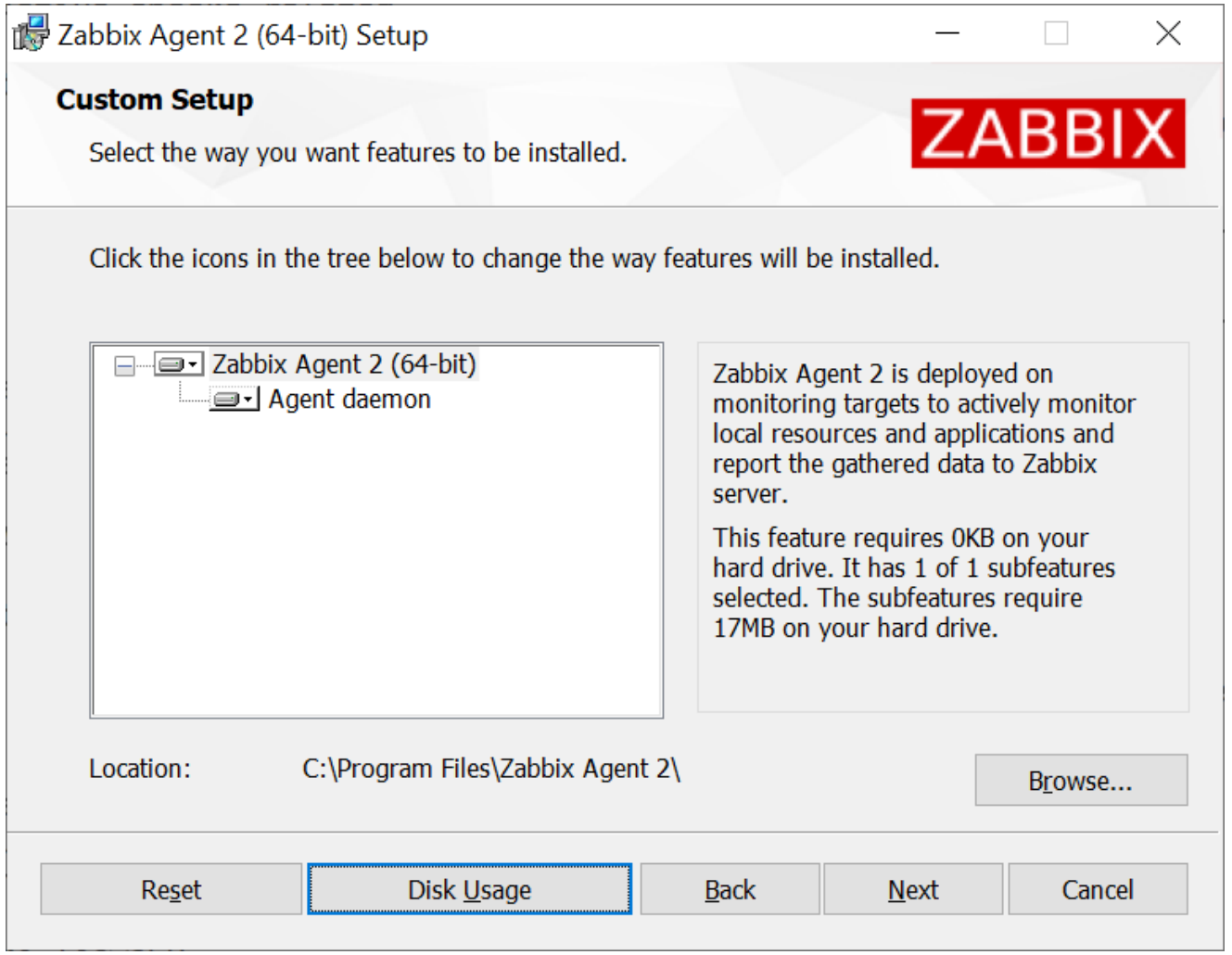
- Host name – Hostname parameter for active agent checks
- Zabbix server IP/DNS – Server parameter for passive checks
- Agent listen port - port on which the agent will listen for passive check requests
- Server or Proxy for active checks – ServerActive parameter for sending active agent datax



initMAX

AGENT 2 INSTALLATION on windows – MSI

- Location – Agent 2 install location
- Disk Usage – disk space usage statistics
- Agent 2 takes up ~17MB of disk space



3

COMMAND LINE OPTIONS

initMAX

AGENT 2 COMMAND LINE OPTIONS

```
# zabbix_agent2 -R help
Remote control interface, available commands:
        loglevel increase - Increase log level
        loglevel decrease - Decrease log level
        metrics - List available metrics
        version - Display Agent version
        help - Display this help message
```

- Other C agent parameters such as – p (print), – t (test) etc. are also fully supported by Agent 2

```
#zabbix_agent2 -t agent.hostname
agent.hostname                                [s|LinuxHost]
```

Agent 2

AGENT 2 PLUGIN STATUS

Available metric list is divided into plugin sections

- Displays per plugin statistics

```
# zabbix_agent2 -R metrics
[NetIf]
active: true
capacity: 0/100
tasks: 2
net.if.in: Returns incoming traffic statistics on network interface.
net.if.out: Returns outgoing traffic statistics on network interface.
net.if.total: Returns sum of incoming and outgoing traffic statistics on network
interface.
```

Agent 2

AGENT 2 PLUGIN STATUS

Status of running agent can be viewed via web browser

- Status page will be accessible from anywhere
- Can be set to any unused port
- No default port setting
- Agent restart is still required after making configuration changes

```
### Option: StatusPort
#       Agent will listen on this port for HTTP status requests.
StatusPort=10080
```

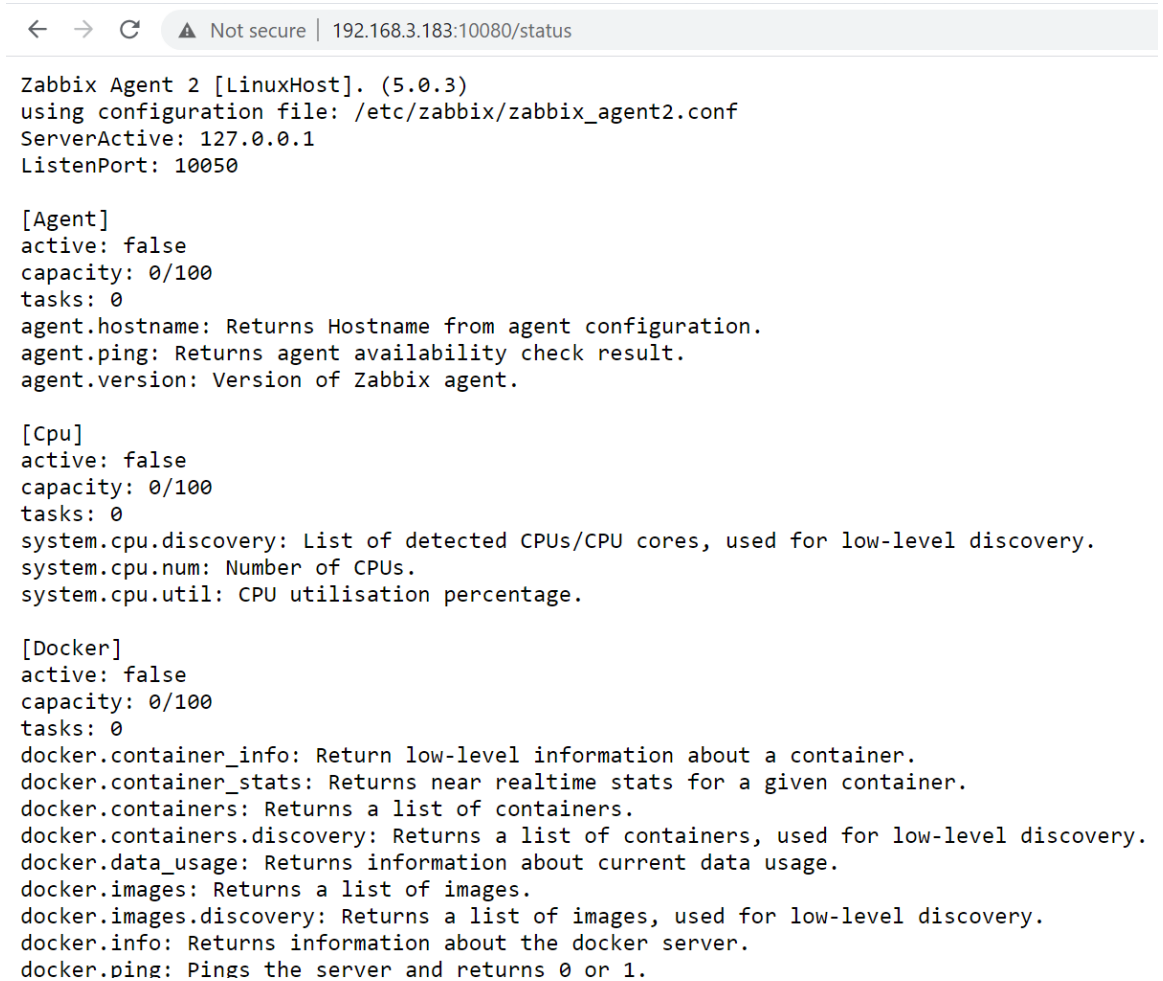
Agent 2

AGENT 2 PLUGIN STATUS

In browser navigate to:

- http://<IP/DNS>:10080/status

Same as from the command line



Agent 2

AGENT 2 PLUGIN CONFIGURATION

- Each Agent 2 item belongs to a specific plugin
- Plugins can be individually configured by changing plugin parameters
- All plugins are configured using "Plugins.*" parameter in "zabbix_agent2.conf"

C Agent MaxLines configuration (plugin-less):

```
MaxLinesPerSecond=20
```

«Go Agent MaxLines configuration – Via Log Plugin:

```
Plugins.Log.MaxLinesPerSecond=20
```

initMAX

AGENT 2 PLUGINS

All of Zabbix agent 2 items utilize some sort of a plugin written by Zabbix developers

Plugins provide an option to extend the monitoring capabilities of Zabbix

- Written in Go programming language and supported for Zabbix agent 2 only
- Alternative to "loadable modules" (written in C)

Each plugin can be configured by editing the plugin specific parameters in Zabbix agent 2 configuration file

- Syntax: Plugins.<PluginName>.<Parameter>=<Value>

4

NAMED SESSIONS

Agent 2

AGENT 2 NAMED SESSIONS

Named sessions

- Represent an additional level of plugin parameters
- Can be applied only to specific plugins
  - Mysql
  - Postgres
  - Redis
  - Memcached
  - More as they are released in the future
- Mostly related to Agent 2 DB monitoring plugins
- Used to define separate sets of authentication parameters
  - URI - Uniform Resource Identifier
  - User – Username which is used for obtaining the metrics from the resource
  - Password – Password for the specified user

Agent 2

AGENT 2 NAMED SESSIONs

Example:

- Monitoring of two sessions "MySQL1" and "MySQL2“

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=<UsernameForMySQL1>
Plugins.Mysql.Sessions.MySQL1.Password=<PasswordForMySQL1>
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://127.0.0.1:3307
Plugins.Mysql.Sessions.MySQL2.User=<UsernameForMySQL2>
Plugins.Mysql.Sessions.MySQL2.Password=<PasswordForMySQL2>
```

- We can now use the defined session names – "MySQL1" and "MySQL2" as connString parameters in our mysql agent 2 item keys

Agent 2

AGENT 2 NAMED SESSIONS

Example keys:

- mysql.db.size[connString,username,password,dbName]
- mysql.ping[connString,username,password]
- mysql.db.discovery[connString,username,password]

Using named session names in key parameters:

- mysql.db.size[MySQL1,,,productionDB]
- Note that we still need to specify the DB name "productionDB" directly in the key

Parameters also can be specified directly if needed:

- mysql.db.size[tcp://localhost,root,P445W0RD,productionDB]

4

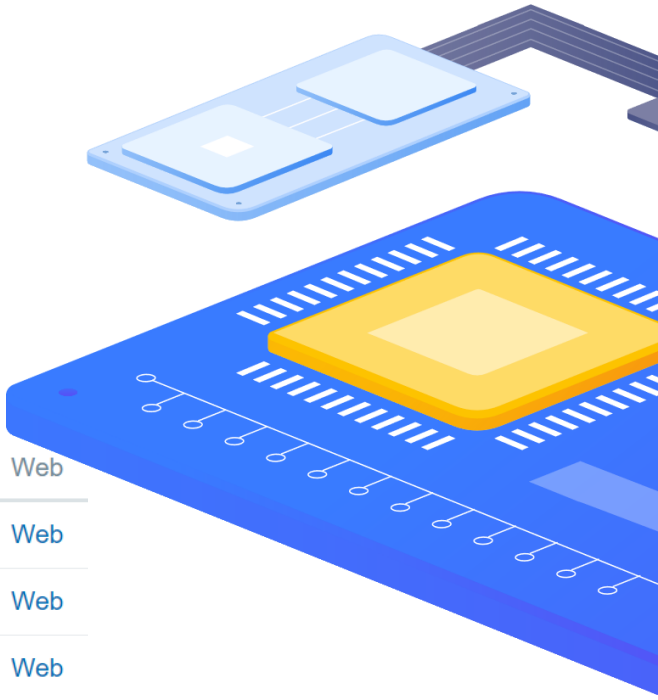
TEMPLATES

Agent 2

AGENT 2 TEMPLATES

Integrated plugins and templates are available out of the box

- DB MySQL
- DB PostgreSQL
- DB Redis
- App Docker
- App Memcached
- More templates will be published in the future

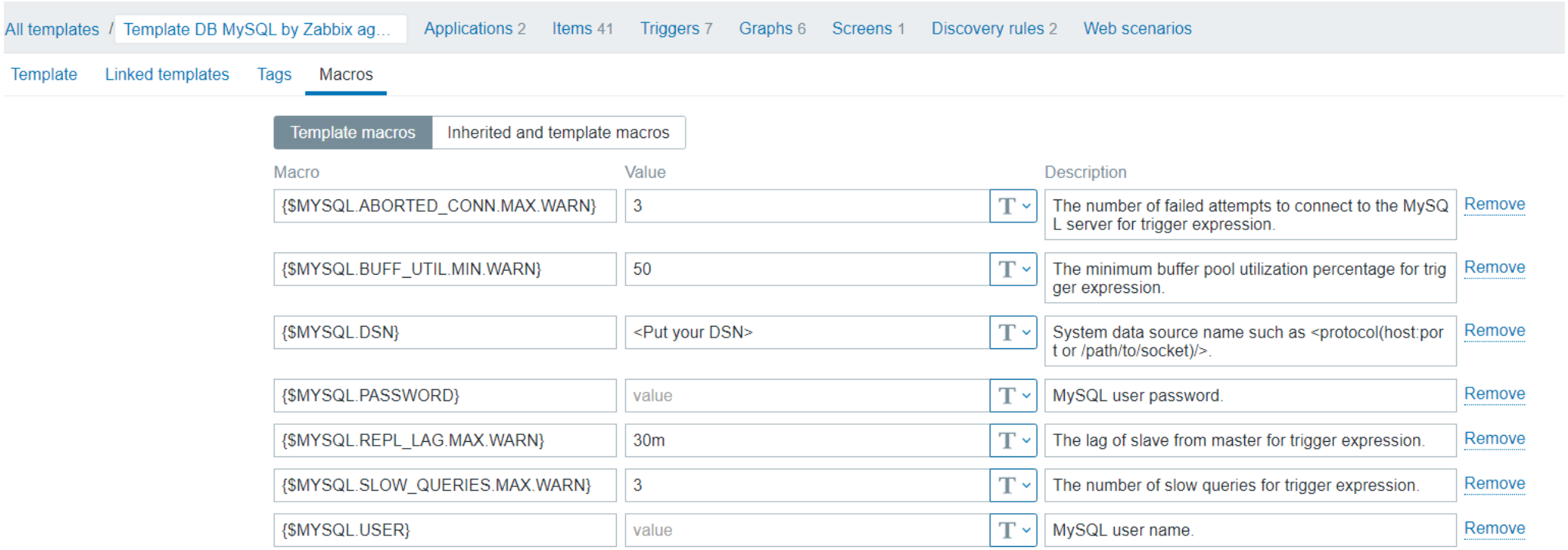


Name ▲	Hosts	Applications	Items	Triggers	Graphs	Screens	Discovery	Web
Template App Docker	Hosts 1	Applications 2	Items 44	Triggers 3	Graphs 5	Screens 1	Discovery 2	Web
Template App Memcached	Hosts	Applications 2	Items 26	Triggers 8	Graphs 6	Screens	Discovery	Web
Template DB MySQL by Zabbix agent 2	Hosts	Applications 2	Items 41	Triggers 7	Graphs 6	Screens 1	Discovery 2	Web
Template DB PostgreSQL	Hosts	Applications 2	Items 40	Triggers 11	Graphs 8	Screens 2	Discovery 1	Web
Template DB Redis	Hosts	Applications 2	Items 64	Triggers 13	Graphs 12	Screens 2	Discovery 7	Web

Agent 2

AGENT 2 TEMPLATES

- Zabbix agent 2 out of box templates can be further configured by customizing User Macros or Plugin parameters
- Basic configuration can be done by simply adjusting the user macros



Agent 2

AGENT 2 TEMPLATES

Deeper customization can be achieved by configuring the plugin itself

```
### Option: Plugins
#       A plugin can have one or more plugin specific configuration parameters in format:
#     Plugins.<PluginName>.<Parameter1>=<value1>
#     Plugins.<PluginName>.<Parameter2>=<value2>
```

```
### Option: Plugins.Redis.Uri
#       Connection string. Can be overwritten by the first parameter of an item's key.
#
# Mandatory: no
# Range: Must matches the URI format.
# Default:
# Plugins.Redis.Uri=tcp://localhost:6379
```

Agent 2

AGENT 2 Exclusive keys

Zabbix agent 2 has many new built-in keys

For Docker

- docker.data_usage - Information about current data usage
- docker.containers.discovery - A list of containers. Used for low-level discovery

For MySQL

- mysql.get_status_variables – get mysql variable data
- mysql.db.discovery - Result of the “show databases” SQL query in LLD JSON format

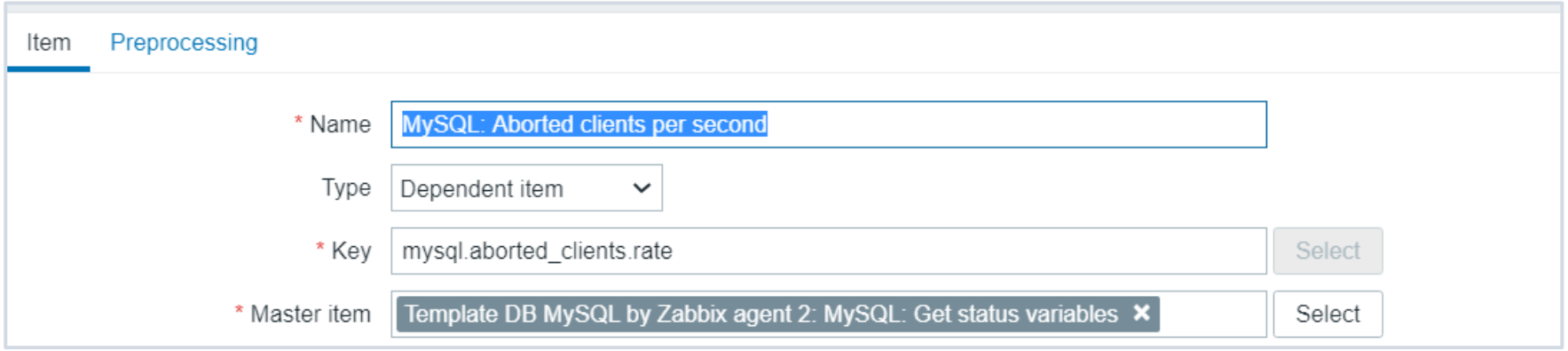
For PostgreSQL

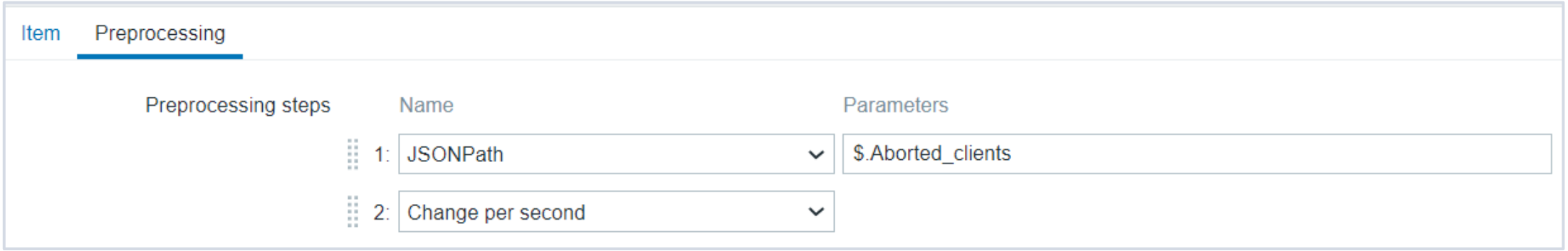
- pgsql.dbstat - Collects statistics per database
- pgsql.db.discovery - List of the PostgreSQL databases

initMAX

AGENT 2 Exclusive keys

- Many more Agent 2 keys are available
- Many of those keys are used as master items
- Dependent items preprocess the obtained values and provide more granular and detailed data overview

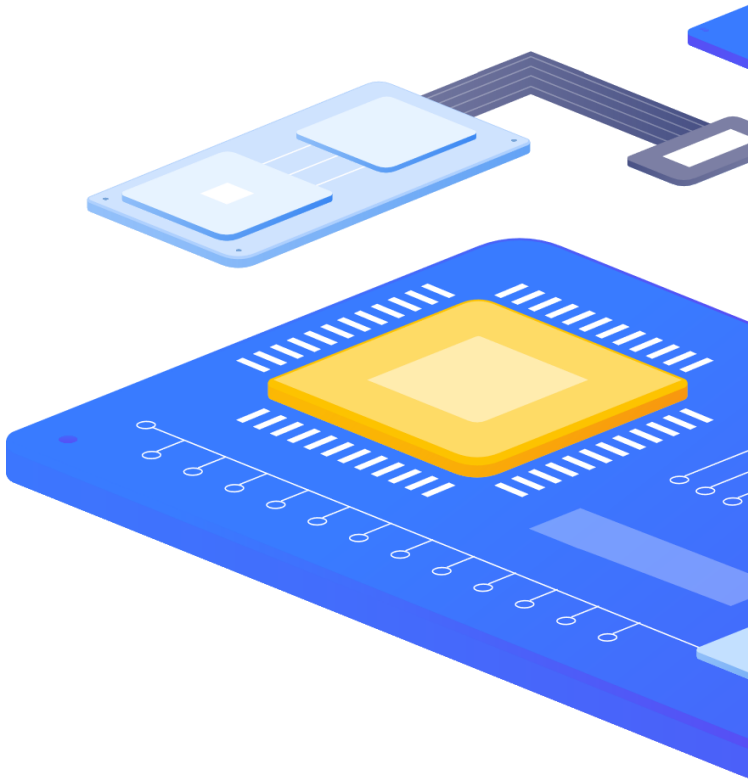




Agent 2

AGENT 2 Master/dependent items - Docker

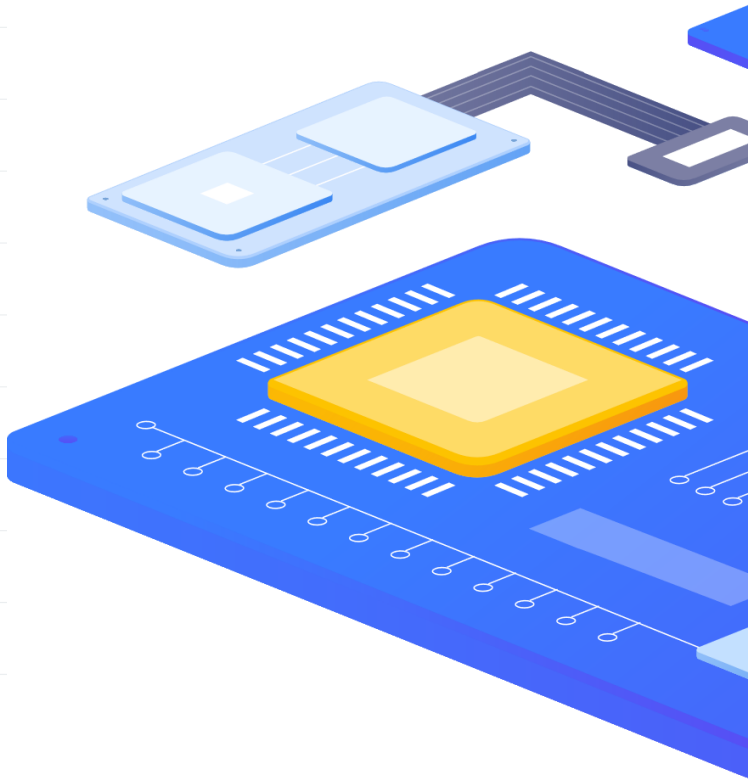
Wizard	Name ▲	Triggers	Key	Interval	History	Trends	Type	Applications	Status	Ir
	Docker: Get info: Docker: Architecture		docker.architecture		7d		Dependent item	Docker	Enabled	
	Docker: Get info: Docker: Cgroup driver		docker.cgroup_driver		7d		Dependent item	Docker	Enabled	
	Docker: Get info: Docker: Containers paused		docker.containers.paused		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get info: Docker: Containers running		docker.containers.running		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get data_usage: Docker: Containers size		docker.containers_size		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get info: Docker: Containers stopped		docker.containers.stopped		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get info: Docker: Containers total		docker.containers.total		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get info: Docker: CPU CFS Period enabled		docker.cpu_cfs_period.enabled		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get info: Docker: CPU CFS Quota enabled		docker.cpu_cfs_quota.enabled		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get info: Docker: CPU Set enabled		docker.cpu_set.enabled		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get info: Docker: CPU Shares enabled		docker.cpu_shares.enabled		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get info: Docker: Debug enabled		docker.debug.enabled		7d	365d	Dependent item	Docker	Enabled	
	Docker: Get info: Docker: Default runtime		docker.default_runtime		7d		Dependent item	Docker	Enabled	
	Docker: Get info: Docker: Docker root dir		docker.root_dir		7d		Dependent item	Docker	Enabled	
	Docker: Get containers		docker.containers	1m	0		Zabbix agent	Zabbix raw items	Enabled	
	Docker: Get data_usage		docker.data_usage	1m	0		Zabbix agent	Zabbix raw items	Enabled	
	Docker: Get images		docker.images	1m	0		Zabbix agent	Zabbix raw items	Enabled	
	Docker: Get info		docker.info	1m	0		Zabbix agent	Zabbix raw items	Enabled	
	Docker: Get info: Docker: Goroutines		docker.goroutines		7d	365d	Dependent item	Docker	Enabled	



Agent 2

AGENT 2 Master/dependent items - PostgreSQL

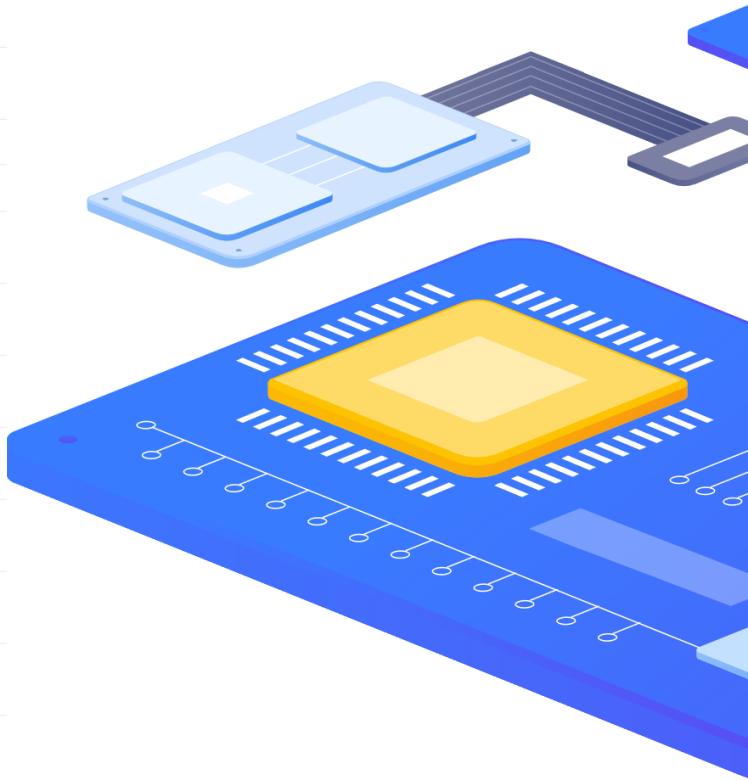
Wizard	Name ▲	Triggers	Key	Interval	History	Trends	Type	Applications	Status
	PostgreSQL: Bgwriter: Bgwriter: Buffers allocated		pgsql.bgwriter.buffers_alloc		90d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Bgwriter: Bgwriter: Buffers written directly by a backend		pgsql.bgwriter.buffers_backend		90d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Bgwriter: Bgwriter: Number of bgwriter stopped		pgsql.bgwriter.maxwritten_clean		90d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Bgwriter: Bgwriter: Times a backend execute its own fsync		pgsql.bgwriter.buffers_backend_fsync		90d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Bgwriter: Checkpoint: buffers background written		pgsql.bgwriter.buffers_clean		90d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Bgwriter: Checkpoint: Buffers checkpoints written		pgsql.bgwriter.buffers_checkpoint		90d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Bgwriter: Checkpoint: By timeout		pgsql.bgwriter.checkpoints_timed		90d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Bgwriter: Checkpoint: Requested		pgsql.bgwriter.checkpoints_req		90d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Get connections sum: Connections sum: Active		pgsql.connections.active		7d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Get connections sum: Connections sum: fastpath function call		pgsql.connections.fastpath_function_call		7d	365d	Dependent item	PostgreSQL	Enabled
	PostgreSQL: Get connections sum: Connections sum: Idle		pgsql.connections.idle		7d	365d	Dependent item	PostgreSQL	Enabled



Agent 2

AGENT 2 Master/dependent items - MySQL

Wizard	Name ▲	Triggers	Key	Interval	History	Trends	Type	Applications	Status
•••	MySQL: Get status variables: MySQL: Aborted clients per second		mysql.aborted_clients.rate		7d	365d	Dependent item	MySQL	Enabled
•••	MySQL: Get status variables: MySQL: Aborted connections per second	Triggers 1	mysql.aborted_connects.rate		7d	365d	Dependent item	MySQL	Enabled
•••	MySQL: Buffer pool efficiency		mysql.buffer_pool_efficiency	1m	7d	365d	Calculated	MySQL	Enabled
•••	MySQL: Buffer pool utilization	Triggers 1	mysql.buffer_pool_utilization	1m	7d	365d	Calculated	MySQL	Enabled
•••	MySQL: Get status variables: MySQL: Bytes received		mysql.bytes_received.rate		7d	365d	Dependent item	MySQL	Enabled
•••	MySQL: Get status variables: MySQL: Bytes sent		mysql.bytes_sent.rate		7d	365d	Dependent item	MySQL	Enabled
•••	MySQL: Get status variables: MySQL: Command Delete per second		mysql.com_delete.rate		7d	365d	Dependent item	MySQL	Enabled
•••	MySQL: Get status variables: MySQL: Command Insert per second		mysql.com_insert.rate		7d	365d	Dependent item	MySQL	Enabled
•••	MySQL: Get status variables: MySQL: Command Select per second		mysql.com_select.rate		7d	365d	Dependent item	MySQL	Enabled
•••	MySQL: Get status variables: MySQL: Command Update per second		mysql.com_update.rate		7d	365d	Dependent item	MySQL	Enabled
•••	MySQL: Get status variables: MySQL: Connection errors accept per second		mysql.connection_errors_accept.rate		7d	365d	Dependent item	MySQL	Enabled
•••	MySQL: Get status variables: MySQL: Connection errors internal per second		mysql.connection_errors_internal.rate		7d	365d	Dependent item	MySQL	Enabled



Agent 2

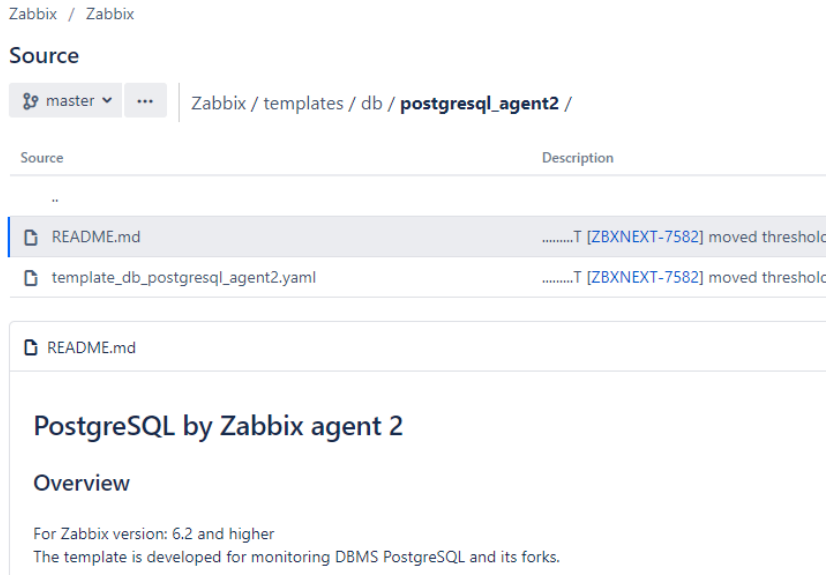
AGENT 2 TEMPLATES

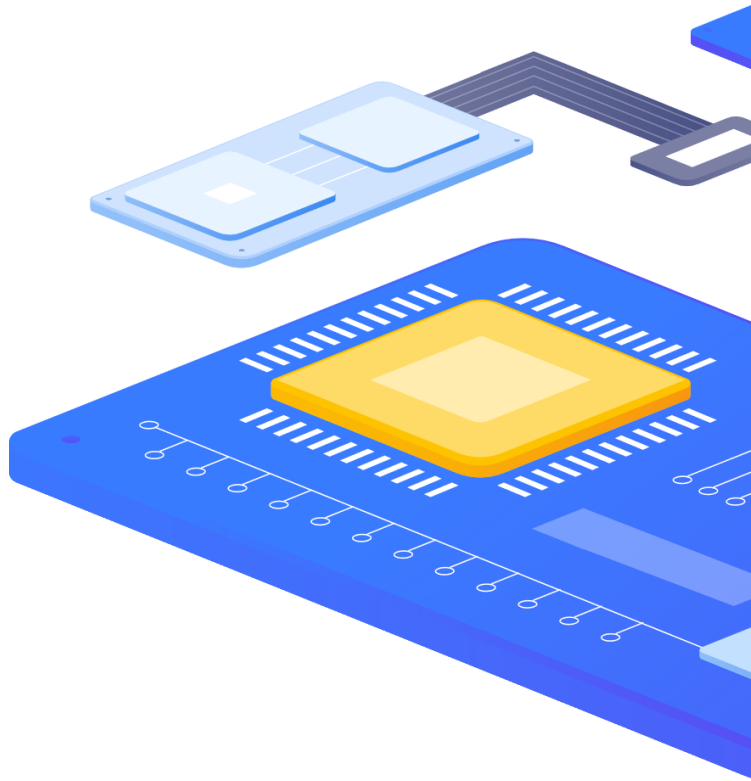
Documentation

- https://www.zabbix.com/documentation/current/en/manual/config/items/itemtypes/zabbix_agent/zabbix_agent2

GIT

- https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates





4

DEVELOPING PLUGINS

Agent 2

DEVELOPING an AGENT 2 PLUGIN

The Go agent features multiple plugin interfaces which can be used for different types of tasks

- Exporter interface
- Watcher interface
- Collector interface
- Runner interface
- Configurator interface

A Go agent plugin must implement one or several plugin interfaces.

ZABBIX BLOG

https://blog.zabbix.com/developing-plugins-for-zabbix-agent-2/9682/

Agent 2

DEVELOPING AN AGENT 2 plugin

Exporter is a very simple interface that polls metrics and returns a value, several values, an error, or nothing at all. It accepts a preparsed key, its parameters and context

```
type Exporter interface {
Export(key string, params []string, context ContextProvider) (result interface{}, err
error)
}
```

Agent 2

DEVELOPING AN AGENT 2 plugin

With Watcher you can implement a metric polling process without using Scheduler. This interface is mostly used to wait for data and upon receiving it send the results to the server, e.g. log file monitoring.

```
type Watcher interface {
Watch(requests []*Request, context ContextProvider)
}
```

Agent 2

DEVELOPING AN AGENT 2 plugin

Collector is used for plugins that need to collect data regularly. However, it can’t return data, so you’ll need Exporter for that.

```
type Collector interface {
Collect() error
Period() int
}
```

- The main use case for this interface is when we need to collect data often and store it in cache until Zabbix server requests it.

Agent 2

DEVELOPING AN AGENT 2 plugin

Runner provides a way to perform initialization when a plugin is activated (the Start() function) and deinitialization when it is stopped (the Stop() function).

```
type Runner interface {
    Start()
    Stop()
}
```

- With this interface a plugin can, for example, start or stop a background thread, release unused resources, close connections, etc.
- Activates plugins when there are metrics ready to be processed for passive checks or a task is assigned to it for active checks.

Agent 2

DEVELOPING AN AGENT 2 plugin

Configurator serves for configuring plugins.

```
type Configurator interface {

    Configure(globalOptions *GlobalOptions, privateOptions interface{})

    Validate(privateOptions interface{}) error

}
```

- Configure() loads configuration parameters in a structure defined by the developer.
- Validate() checks the configuration file for errors. If it finds any, the agent won't start, and we'll get an error notification.

initMAX

DEVELOPING AN AGENT 2 plugin

Agent 2 is constantly in development. We have many new features planned for future releases:

- Building loadable plugins as dynamic libraries
  - Removes the need to rebuild the agent to add a new plugin
- Updating agent configuration in runtime
  - No need restart the agent when changing configuration
- New out of the box plugins to use with Agent 2
  - Support for new applications and systems
- Ongoing optimizations, bug fixes and other new features

FAQ

Should I use the C agent or the Go agent?

- Use the Go agent if you need the extra monitoring features or want to implement your own custom plugin
- systemctl won't report back the agent status on startup

Will you still support the C agent?

- Yes, of course! C agent will still get updates, fixes, etc. We have no plans to deprecate it.

Can I use both agents in my environment?

- Yes! You can use the Go agent in environments where you need the new features and keep the C agent on all other hosts.

5

Demonstration

initMAX

Questions?

Agent 2

CONTACT US:

Phone:	+420 800 244 442
Web:	https://www.initmax.cz
Email:	tomas.hermanek@initmax.cz
LinkedIn:	https://www.linkedin.com/company/initmax
Twitter:	https://twitter.com/initmax
Tomáš Heřmánek:	+420 732 447 184