



LAUREA
UNIVERSITY OF APPLIED SCIENCES
Together we are stronger

Implementing a Network Monitoring Tool in a Company IT Infrastructure

Perttu, Valtteri

2016 Laurea



Laurea University of Applied Sciences

Implementing a Network Monitor Tool in a Company IT Infrastructure

Valtteri Perttu
Degree Programme in Business
Information Technology
Bachelor's Thesis
December, 2016

Perttu, Valteri

Implementing a Network Monitoring Tool in a Company IT Infrastructure

Year	2016	Pages	46
------	------	-------	----

Zabbix is an open-source network monitoring tool with the possibility to monitor a company's IT infrastructure and alert administrators when problems occur on the level of hardware, application or process.

The thesis was commissioned by a Finnish retail company. The goal of the thesis was to automate the hardware, application and process monitoring by implementing Zabbix in the company IT infrastructure and to provide a report about its possibilities.

In the thesis report a list of Zabbix features and protocols are presented and an explanation is given of the kind of monitoring solutions that were implemented. A how-to guide was composed, so the company could refer to this thesis when changes to the system are needed to perform, such as adding new hosts to monitoring or implement features that had not previously been used.

When starting the project, the basic knowledge of the software was already acquired, hence some of the methods used were based on the experiences gained previously. Other research data comes from Zabbix guideline books and online sources.

With the help of Zabbix the company can identify and react to the problems immediately when they occur and take preventive measures to minimize the impact on business continuity. New development ideas for making the services highly available were found during the research and they could be implemented to Zabbix in the future.

Keywords: Zabbix, Network, Monitoring

Terminology

- API - application programming interface, allows communication between different applications.
- BMC - Baseboard management controller, provides the features to the IPMI protocol and it is usually integrated to the motherboard.
- GPL - General Public License, widely used among open source projects
- HTTP - Hypertext Transfer Protocol, a communication protocol web browsers and web servers use to transfer data.
- IPMI - Intelligent Platform Management Interface, computer subsystem that allows independent monitoring of operations.
- NMS - Network Monitoring System, system to monitor IT infrastructure and notify administrators when problems occur.
- OSI model - The Open Systems Interconnection model, dividing computer networking protocols into different layers. Each layer uses functions from the layers below and provide functions to the upper layers.
- PHP - Hypertext Preprocessor, coding language used in web server environments and creating dynamic web pages
- PSK - Pre-shared key, encryption method in cryptography.
- SLA - Service Level Agreement, describes the services agreed level of downtime.
- SNMP - Simple Network Management Protocol, TCP/IP protocol that offers possibility to monitor IP based network devices.
- SOA - Service Oriented Architecture, software design method where functions and processes are designed to work independently and flexibly.
- TLS - Transport Layer Security also known as SSL. Encryption protocol to protects IP-communication between internet applications, e.g. https protocol.

Table of Contents

1	Introduction	6
1.1	Network Monitoring System	7
2	Zabbix	8
2.1	Why Zabbix?	8
2.2	Architecture	9
2.3	Features and protocols	11
3	Setting up Zabbix	20
3.1	Preparations	20
3.2	Zabbix server installation	20
4	Zabbix Agent Installation for Windows	23
4.1	Installing Zabbix agent	24
5	Zabbix frontend configuration	25
5.1	Adding a new user	25
5.2	Server host groups	25
5.3	Adding new host	26
5.4	Adding item and trigger	29
5.5	Configuring actions	31
6	Current state of monitoring	34
6.1	Future implementations	36
7	Conclusion	37
	References	38
	Figures	42
	Appendices	43

1 Introduction

It is difficult to know what is going on inside a company's network and servers without any monitoring. Are all the services running? Are there any performance issues within servers? Are the problems caused by Internet Service Provider (ISP) or is it within the IT-infrastructures connections? If these questions are being asked, then centralized network monitoring tool might be the answer.

The definition of network monitoring can be different depending on what aspect it is looking from. One definition from article, Nash and Behr (2009), displays it in a way from where it is simple to understand the basic approach of network monitoring; "Network monitoring is like a visit to a cardiologist. You're combining experience, judgement and technology to chart a system's performance. Your doctor is watching for danger signs as blood flows through vessels, valves and chambers of the heart, while your network monitoring systems are tracking data moving along cables and through servers, switches, connections and routers." In addition, network monitoring is also related to monitor applications, running processes and hardware statuses.

This thesis is focused on implementing network monitoring to a large Finnish company's IT-infrastructure with open source monitoring platform, Zabbix. The company needs to know what is going on inside their systems and automate as much hardware, application and process monitoring as possible.

Zabbix was the chosen monitoring platform because the company had previously used it, therefore they had the basic understanding of its capabilities. The previous system, Zabbix 2.0, was outdated and un-administrated, which resulted to lots of false alerts and some of the monitored hosts were not existing anymore. Before deciding whether to upgrade to a newer version or use the old already deployed version, the differences were analyzed. The conclusion was evident for deploying a new, Zabbix 3.0, system and start the configuration from the beginning. (Appendix 1: Upgrade notes)

In order to achieve the best results, the understanding of how Zabbix functions and its possibilities need to be identified. This is done through introducing the Zabbix architecture, the key features and protocols behind the frontend.

To demonstrate the administration of the system, a how-to guide is composed and a production server is used as an example. The purpose is to utilize network, application and hardware monitoring to company's production and acceptance servers. To successfully configure and set up the monitoring platform and add the requested hosts, monitored items and configure

email alerts to the person in charge of the monitored system. Each system was introduced one-by-one and given specific details of the necessary items they wanted to monitor. Zabbix is meant for IT administrators who wish to be notified of problems immediately when they occur. This helps the company to react faster to problems, identify the cause and prevent them from causing any problems for the business. Utilizing monitoring tools increases the availability and business continuity on offered services. Therefore, the end-user doesn't necessarily even notice there has been a break in the provided service. This thesis was done in collaboration with the software departments IT manager. The company can use this thesis as a reference when some changes or new implementations to the system are needed to perform.

1.1 Network Monitoring System

Network monitoring has quickly become a trend. Because the networks have grown and more data is being transferred, the use of network monitoring has become a valuable addition. In the 1990s, there were mostly proprietary monitoring solutions on the market and most of them were vendor locked. The reason why vendor free products were not available was because International Telecommunication Union (ITU) kept a private "blue book" which held all the telecoms standards. Carl Malamud also referred as "Savior of the internet" realized that ITU had no legal rights to hold copyrights over the material, and published them available to the public. This made it possible to vendors and admins to start using the same standards and it opened the network monitoring system (NMS) market for developers. A decade later, open-source solutions began to come to the market. (Birch 2016)

The concept of network monitoring consists of three main subjects: security, performance and reliability. System administrations are generally interested in all these areas. By the help of wide range of NMS available, such as Zabbix, all those aspects can be acknowledged. As an example, it is possible to see what ports are open in routers, has somebody change the root password even though it should have not been altered, is the root cause of poor performance because of low disk space or high processor load or to see whether the SLA of an offered services are within the boundaries. (Zabbix security 2016)

The qualities of NMS offer companies to save money, since only one administrator is required to monitor the network in a single location, save time with an easy and direct access to the monitored data and to increase productivity by identifying the issues immediately when they occur. (Technopedia 2016)

2 Zabbix

Alexei Vladishev, the creator of Zabbix, started developing Zabbix as a hobby but soon it turned into a full-time job. “I hoped to find a way how to automate certain administration and monitoring tasks. The goal was to deliver nearly 100% uptime and be informed immediately about any issues in production environments doing financial transaction processing. I realized very soon that standard commercial solutions won't be able to deliver what I wanted, for a reasonable budget. My expectations were pretty low at that moment: real-time graphing, pro-active alerting and access to historical data.” (Vladishev 2011)

Zabbix was announced in 1998 and the first public release was in 2001. The first stable version was released in 2004 and from that point it has been under continuous development. A new stable version of Zabbix is released in every 1,5 years. Zabbix offers variety of different features out of the box, such as auto-discovery, trend prediction, support for multiple operating systems, agent-based and agent-less monitoring. It is released under the GPLv2 license, which allows end-users possibility to modify any configuration settings. Even though there are no commercial based product available from Zabbix, the company behind Zabbix, Zabbix SIA, offers customer training and support. The price for training starts from 1500€ and there are various technical support packages available from basic online support to custom Zabbix on-site builds. (Zabbix product 2016)

Zabbix does have some disadvantages such as the learning curve is high and proper configuration involves a lot of “trial-and-error” approach. Structure of the official documentation is hard to follow and information is chattered through different pages. But on the other hand, the flexibility and highly customizable possibilities outcome the negative sides. The user can modify the base code and develop their own solutions to problems. There is also a strong community to provide user support, templates and custom scripts to users. Zabbix offers possibility to monitor everything from server hardware, routers, printers, logs, to running services and a way to execute remote commands. The rule of thumb is that if some item has an output, it can be monitored with Zabbix.

2.1 Why Zabbix?

Zabbix was tested to provide the needed features by another company inside the same corporation. It was coming along with the Enterprise service bus (ESB) system, which delivers messages through it from system to system. At first it was only being used to monitor the ESB system, but since monitoring was required in other systems as well, Zabbix was expanded.

Including the fact that with the help of Zabbix the company knows what is going on in their IT-infrastructure, some of the daily tasks which were previously done manually could be automated. For example, making sure certain programs have been executed successfully by monitoring the log file or composing a simple script which will tell the modification time of the files in some directory. Also with the help of Zabbix, previously outsourced point of sale system monitoring could be taken back to internal monitoring and therefore save money. The ultimate goal with Zabbix is to achieve as much proactive monitoring as possible. Declaring triggers to create an alert when problems are about to arise and using remote commands to react to the most common situations that could be fixed by automatically run some script or restart the services that are required for the service to function properly.

2.2 Architecture

In the Figure 1, all the components and the way they are linked to each other are presented. Zabbix can be allocated to three main components: Zabbix server, Database and Zabbix frontend.

Zabbix server supports only UNIX platform. It has been tested to be the only operating system which has the needed performance, fault tolerance and flexibility. It is the central process of Zabbix software. The server is in charge of polling and trapping of the gathered data; it performs trigger calculations and sends notifications to users when events occur. (Zabbix requirements 2015)

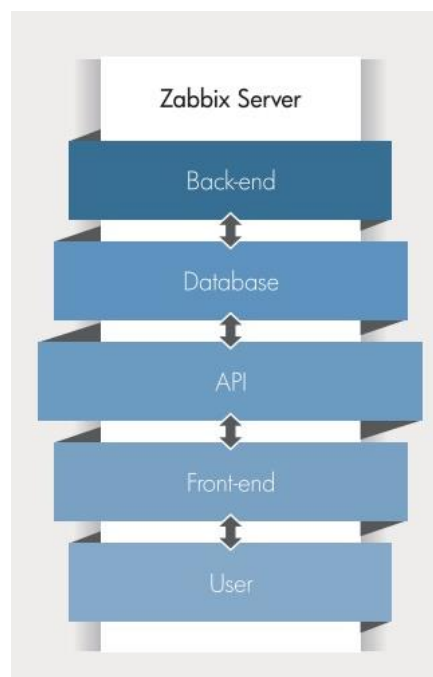


Figure 1: Zabbix Architecture (Brochure 2016)

The hardware requirements grow depending on the size of the monitored network and the chosen database engine. But even with large installations, the requirements should be easily met with modern technology.

Name	Platform	CPU/Memory	Database	Monitored hosts
<i>Small</i>	CentOS	Virtual Appliance	MySQL InnoDB	100
<i>Medium</i>	CentOS	2 CPU cores/2GB	MySQL InnoDB	500
<i>Large</i>	RedHat Enterprise Linux	4 CPU cores/8GB	RAID10 MySQL InnoDB or PostgreSQL	>1000
<i>Very large</i>	RedHat Enterprise Linux	8 CPU cores/16GB	Fast RAID10 MySQL InnoDB or PostgreSQL	>10000

Figure 2: Zabbix hardware requirements (Zabbix requirements 2015)

Zabbix has a support for most of the popular database engines such as MySQL, PostgreSQL, SQLite and IBM DB2. The database is used for keeping the records of historical and configuration data. This means it is possible to view the gathered data and see how that specific item has acted in the past. To backup Zabbix, the database should be backed up accordingly.

Database size depends on three variables: history, trends and events. History keeps the information on each collected value and trends keep the average data values on hourly basis. Trends are more important for long-term period, since storing trends requires less resources. For a short period, history values are preferred. There is a built-in housekeeper in Zabbix, it deletes every value from database that is older than the value they've been set on. Example when history value is seven days, the housekeeper will delete values older than the configured period. Same applies in trends and events.

New values polled per second depends on the size on the monitored network. As an example, if the total number of monitored items is 1200 with the refresh rate of 60 seconds, 20 new values per second are stored in the Zabbix database. One numeric value is approximately 90 bytes. If seven days of history and a years' worth of trend values are stored, the needed disk space is 1,1Gb for history and 0,95 Gb for trend data. Events are incidents that happens, such as a trigger changes its status from OK to PROBLEM or when a new host is detected. Each event requires about 170 bytes of disk space. It is hard to predict how many events are generated, but in the worst-case scenario Zabbix generates one event per second. Storing the event data for one month it requires 0,45 Gb of disk space. The database size will grow depending on the gathered data. The total required disk space can be calculated as: Configuration + History + Trends + Events. Fixed size for configuration data is 10Mb, in this case the database size should be 10Mb + 1,1Gb + 0,95Gb + 0,45Gb =2,5Gb.

The web frontend utilizes PHP coding and is functioning on top of Apache web server. After the server installation, the managing and configuration is mostly done through the frontend. Zabbix frontend communicates with database through API. With the API, it is also possible to integrate Zabbix with third party software, such as a company's ticketing system to automatically create tickets when an alert occurs. (Zabbix requirements 2015)

The screenshot displays the Zabbix 3.0 web frontend dashboard. At the top, there is a navigation bar with the ZABBIX logo and menu items: Monitoring, Inventory, Reports, Configuration, and Administration. Below this is a secondary navigation bar with: Dashboard, Overview, Web, Latest data, Triggers, Events, Graphs, Screens, Maps, Discovery, and IT services. The main content area is titled 'Dashboard' and is divided into several sections:

- Favourite graphs:** Shows a single graph titled 'Zabbix server: Zabbix server performance' with a 'Graphs' button below it.
- Favourite screens:** Displays 'No screens added.' with 'Screens' and 'Slide shows' buttons.
- Favourite maps:** A section for map favorites.
- Status of Zabbix:** A table providing system health metrics.

PARAMETER	VALUE	DETAILS
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled/templates)	114	57 / 8 / 49
Number of items (enabled/disabled/not supported)	6560	3890 / 2651 / 19
Number of triggers (enabled/disabled [problem/ok])	1846	1771 / 75 [25 / 1746]
Number of users (online)	13	1
Required server performance, new values per second	42.48	
- Last 20 issues:** A section for recent alerts.
- System status:** A section for overall system health.
- Host status:** A section for individual host monitoring.
- Web monitoring:** A section for web service monitoring.

Figure 3: Zabbix 3.0 web frontend

2.3 Features and protocols

Zabbix supports many different methods and protocols, such as SNMP, IPMI, low-level-discovery and auto-discovery. For understanding how each feature relates to each other, some basic concepts need to be explained at first.

The monitoring process starts by defining a host which is the machine data is being gathered, it can be server, router, switch or any IP-based network device. From that host a specific set of data that is collected are called items, as an example item key 'cpu.util' collects values of the hosts processor utilization percentage.

Based on that item, a trigger can be set up. Trigger is the entity that creates an alert when the specific threshold has become true, such as when free hard disk space goes below 10Gb. Trigger functions define on what basis the alert is created, whether the average values are over the set threshold or the previous value is different than the last. All the supported trigger functions are found from the Zabbix official manual. Mathematical calculation can also be conducted between triggers, for example if division between two triggers is over the threshold, an alert is created. Without any triggers, it would be impossible to analyze what is wrong with the IT infrastructure, due the large amount of data available.

With actions, an email can be send to different administrators when a trigger has become true. It is also possible to configure actions to send alerts via SMS or instant messaging services like Jabber. Alerts can be sent only for those who are concerned by the information. Actions also have many different useful functions; manage agent discovery, network discovery and execute remote commands. In order to achieve pro-active monitoring a remote command can be configured to act accordingly when a certain event happens. For example, if a service that needs to be running constantly changes it status to 'stopped', then the corresponding trigger creates an alert and based on that alert a remote command is executed to automatically start the service again. The whole process can be described as a chain reaction, when incident X happens it creates incident Y, which creates another set of events and so on.

Agent and Agent-less monitoring

Zabbix supports both agent and agent-less monitoring. The chosen method depends on the level of details one wishes to acquire and the device base of the infrastructure.

Agent is an external software developed in C-language, that collects data locally from the host machine and then reports the findings to central processing system, which in this case is the Zabbix server. The agent communicates with server using JSON (JavaScript Object Notation) based communication protocol. Zabbix agent's greatest benefits are efficiency and diversity. There are wide range of monitoring possibilities to choose from. Including the built-in items, Zabbix agent can collect values from any output whether it is coming from Windows performance metrics, an output of a script or a command. It is recommended to use agent based monitoring, since it resorts to local machines resources. Zabbix agent has a support for Windows and Linux operating systems. (Zabbix agent 2015)

It is not always plausible or sensible to install agent to the host machine, such as in routers or network switches, this is when agent-less options come in. Simple Network Management Protocol (SNMP), is the most common way to monitor devices without the need of installing any external software. SNMP works in an application layer of an OSI model and collects device information over IP-connection. With SNMP, it is possible to retrieve data from various sources,

such as CPU, hard disk, log, temperature and bandwidth statuses. SNMP-agent is required for monitoring, but difference is, it's already installed on most of the network devices, it just needs to be enabled and configured. (Ellingwood 2014) Zabbix server also have a feature called simple checks. With simple checks, it is possible to send ping over networks, monitor TCP/UDP ports and monitor VMware virtual machines. Zabbix server is in charge of processing and polling the data, therefore these items can be used to any computer in the network regardless whether a Zabbix agent is found or not. (Zabbix simple checks 2015)

Passive and Active agent

Zabbix agent can be configured working in both passive and active mode. Passive agent works in a way that Zabbix server asks for data to monitor with a certain frequency and the agent sends the collected values back to the server. The more efficient method is to use active agent; it retrieves a list of items to monitor from Zabbix server and process them locally independently and sends the values to the server. This is especially convenient when monitoring log files and searching for a specific string to appear. Agent only sends the current lines where the string is attached, this saves bandwidth and server resources. (Zabbix agent 2015)

IPMI

Zabbix has a built-in support for IPMI, Intelligent Platform Management Interface. IPMI runs on the Baseboard Management Controller (BMC), which is a computer sub-system that can be found on most of the modern server motherboards. The protocol works on a hardware level and it provides low-level access to a system, overriding operating system controls. By utilizing IPMI it is possible to give simple remote commands such as booting or shutting down the machine, gaining access to BIOS settings and monitor hardware level data. If not secured accordingly a hostile entity can cause considerable damage to the systems. (US-CERT 2013)

Trend prediction

Trend prediction was introduced in Zabbix 3.0. Trend prediction, also referred as predictive trigger functions, calculates using algorithms and historical values to forecast what could happen in the near future. By getting an alert of a problem that may happen in a few days, actions can be taken to prevent the whole incident. One of the most common usage of trend prediction is when monitoring hard disk space. Zabbix forecasts when disk space is about to run out and gives an alert of it, for example in the previous day. (Zabbix predictive 2015)

Zabbix Proxy

Proxy is meant for collecting values on behalf of the Zabbix server. This eases the load on Zabbix server, since proxy is in charge of receiving collected data. By using proxy, it is also simple to build centralized and distributed monitoring. A proxy can be recommended if the monitored network has branches in several different locations, because only one firewall rule needs to be declared for Zabbix. Implementing proxy to remote locations with unreliable connections confirms that no data is lost in case of network failure, since Zabbix proxy buffers the collected values locally in a separate database and then reports to the assigned Zabbix server. Depending on the network architecture, proxy is not always needed or recommended, for example if the host are all in same subnet or in same location, the usage of proxy becomes unnecessary. (Zabbix proxy 2015)

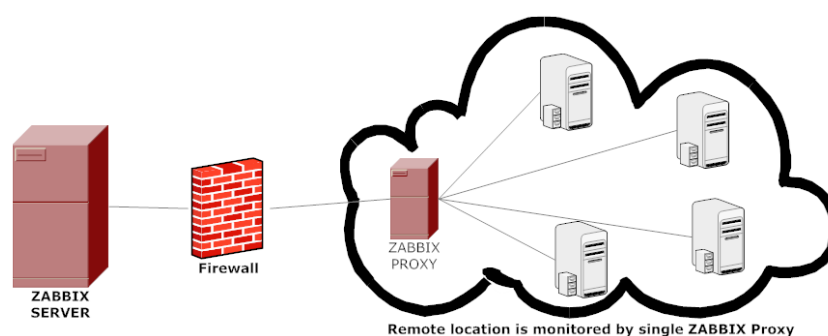


Figure 4: Proxy concept (Zabbix proxy 2015)

Templates

Template is a set of prebuild objects which can be assigned to multiple hosts. A template can consist of items, triggers, graphs, applications, screens, low-level discovery rules and web scenarios. Usually hosts have fairly similar set of the basic monitoring specifications. Each host can be added to their own template group, for example, adding every Windows based machine to Windows template. Assigning templates to hosts saves time when managing the objects e.g. when one monitored item is needed to be applied to multiple host, it is only required to add the item to a template and it will be automatically added to every host within that template. Zabbix comes with various ready-made templates that gives a good starting point to monitoring. Usually the triggers are too sensitive and some items are not required so the templates should be modified accordingly to fit the monitored network. In addition to Zabbix official templates, a community made templates are also available. (Zabbix template 2015)

Graphs, Screens

Because of the massive amounts of data Zabbix is collecting, it becomes inefficient to look the data by the numbers. Zabbix provides a way to visually represent the data through graphs. Any unusual behavior can be noticed from several graphs available; simple graph, ad-hoc graph and custom graph.

Simple graphs are provided straight from the gathered items without any configuration. They represent the gathered data visually and easily by just few clicks. The time period can be selected directly from the graph menu. The white background presents the working days while non-working days are displayed in grey. (Zabbix graph 2015)

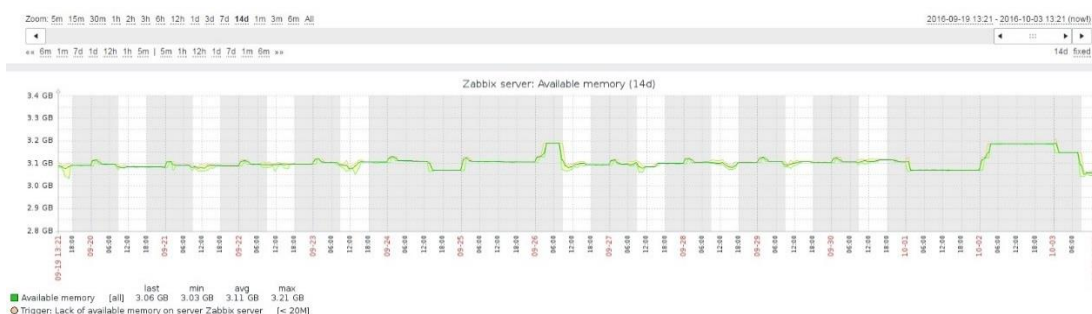


Figure 5: Simple graph

While simple graphs display one meter of an item. With ad-hoc graphs it is possible quickly to compare multiple items at the same time. One difficulty with ad-hoc graphs is that if the items have a different scale, one items graph becomes nearly unreadable. (Zabbix graph 2015)

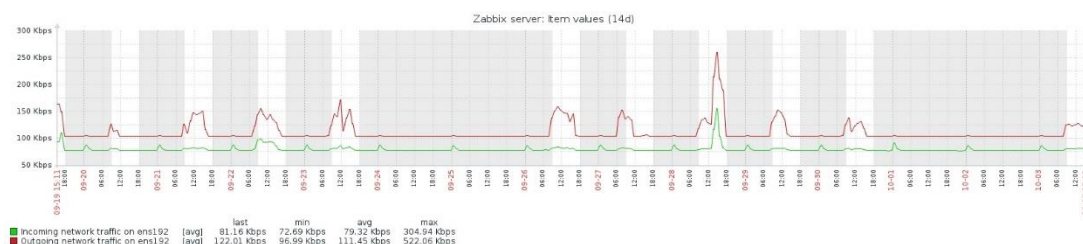


Figure 6: Ad-hoc graph

Custom graph offers customizable objects and different combinations in graph style. Graphs can be assembled from any items available and add multiple values to a one graph. Unlike simple and ad-hoc graphs, custom graphs are saved, hence it is possible to assign the most used graph to a template and deliver it easily to different hosts if necessary. (Zabbix graph 2015)

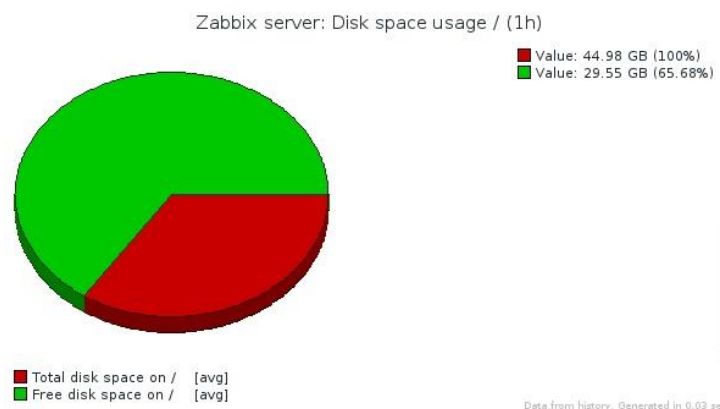


Figure 7: Zabbix custom graph

Screen is a multiple set of different custom graphs in one display. Screen can consist graphs from different hosts and different items e.g. one logical groups CPU and hard disk read/write performance from several different machines. With a single glance, it is possible to see how particular objects handle and if there are any correlations between them. (Zabbix screens 2015)

Web monitoring

By defining web scenarios, it is possible to monitor web sites response time, download speed, incoming error messages and number of failed steps. A web scenario can consist of one or more http requests. The steps are pre-defined and Zabbix executes them in the numeric order. It's possible to execute simulated logins or simulate mouse clicks on the page.

Web monitoring utilizes cURL protocol which retrieves the sites HTML code and inputs the variable to the wanted fields e.g. Log in details. cURL doesn't support JavaScript so if a site's functions are designed to work only with JavaScript, the simulated clicks won't work. This is good to keep in mind when starting to experiment with web monitoring. Http GET and POST methods are supported. Zabbix server must be configured with cURL support if web monitoring is used. (Zabbix web 2015)

Database monitoring

Zabbix supports database monitoring through ODBC monitoring. ODBC is an API that can access database management systems (DBMS). Any database with ODBC support can be queried with Zabbix, such as MySQL, PostgreSQL, Oracle and Microsoft SQL Server. Some Microsoft SQL meters can be queried straight from the windows performance meters without the need of

configuring ODBC connection. ODBC driver passes the queries to the DBMS and then sends the data back to Zabbix, hence the queries are done indirectly from the database. The queries itself are defined in the database item creation step in the web frontend. This method allows monitoring to be more efficient when querying from multiple databases. Zabbix supports unixODBC and iODBC, which are the most commonly used open source ODBC API implementations. (Zabbix database 2015)

Command execution

To extend the monitoring capabilities, remote command and script execution are valuable auxiliaries. These methods are useful when the wanted monitoring output cannot be executed by the help of available items. Remote commands can either be executed locally by the agent or the Zabbix server can oversee the execution. If a specific command or script is wanted to apply only to few hosts, it makes more sense to use the agent to execute the command locally. To achieve this, a User parameter is required to add in to the agent's configuration file. Such parameter can be set to execute a script from the machine or a command through the available command-line terminal which in Windows environment is usually PowerShell or a batch command and in Unix environment a bash command. (User parameter 2015)

As a real-life example a PowerShell script was created to retrieve the list of files in a folder and in the morning, the folder needed to be empty. On a Unix machine a bash script was executed to monitor file modification times. If the oldest file was over half an hour old, Zabbix would create an alert. When this happened, it meant something in the system wasn't functioning properly. Zabbix can't handle empty values coming from the host so it is mandatory for the command to have some value or a string printed in the output.

Server scripts are called external scripts and can be can executed on any host available. The scripts are stored in the Zabbix server. Therefore, a running agent is not needed on the monitored host, but since the scripts are run by the server they can decrease the server performance and should not be overused. (External 2015)

Auto-discovery

To add new hosts without manually adding them one by one in the frontend, it is possible to use auto-discovery function. The function automatically discovers previously unknown hosts trying to contact Zabbix server. In large installations, this greatly reduces the time it takes to apply monitoring. The agent auto-discovery is done by defining an action and selecting the

auto registration event source. From there it is possible to manage the discovery settings depending what operating systems are added. Any discovered host can be automatically added to a specific host group and template. Auto-discovery can also be used to discover network devices within a certain IP range, such as SNMP enabled switches and routers. (Vacche and Lee 2013, 246)

Low-level discovery

As host discovery can be automated, with the help of Low-level discovery (Lld), so can item, trigger and graph creation. Out of the box, Zabbix has a support for six discovery rules: Network interfaces, Filesystems, SNMP objects, CPUs, Windows services, SQL queries and more can be created by the user. These items are included in the predefined templates that come with Zabbix installation. Zabbix will automatically look such data as available hard disks and network adapters. Because of Lld, the Zabbix administrator does not have to manually add every hard disk volume or network interface of the host machine. (Vacche and Lee 2013, 248)

Encryption

Zabbix 3.0 offers Transport Layer Security (TLS) encryption, previously known as Secure Sockets Layer (SSL), between Zabbix server and agent. To encrypt communication, both server and agent need to be compiled with TLS support. Encryption can be configured to individual components, meaning that host X's communication can be encrypted while host Y's might not be. There are two encryption methods available, pre-shared key (PSK) and certificate-based. PSK consist of two identifiers; a non-secret PSK identity string like "PSK ID 001" and a secret PSK string value which is a complex string of hexadecimal digits e.g. "e651fb0y918d23d21b4f641851f5f570ad79c390931302e5391d09327ba434e9". Before establishing PSK encrypted connection, the server checks the PSK values configured for that agent in database and when agent is receiving the connection it compares if the values are same, the connection is enabled. (Zabbix PSK 2015)

Certification-based encryption supports public or in-house certificate authors (CA). The certificate needs to be signed by a trusted CA. The verification steps are similar to PSK-based encryption. When establishing connection between the server and agent they both check each other's certifications and private keys, if they match and are validated the connection is made. (Olups 2016)

IT Services

IT services are meant to display the high-level business availability of the infrastructure. In business-level view, the interest is not on low-level details such as CPU utilization or in memory consumption. The business is only interested in the availability of those specific services that are needed to maintain business continuity and whether the Service-level agreement (SLA) is within the agreed boundaries. Zabbix offers IT services to provide this information. To configure this, the business processes which have an effect on availability, need to be identified and add triggers for them. When the triggers are set, they can be individually configured to display each services SLA. Figure 8 provides an example of the IT service display. (Vacche and Lee 2013,205)

SERVICE	STATUS	REASON	PROBLEM TIME	SLA / ACCEPTABLE SLA
root				
▼ Production services				
▶ ERP	OK		<div style="width: 100%; height: 10px; background-color: green;"></div> 0.0000	100.0000 / 99.0500
▶ ESB	OK		<div style="width: 100%; height: 10px; background-color: green;"></div> 0.0000	100.0000 / 99.9000
▶ Production	OK		<div style="width: 100%; height: 10px; background-color: green;"></div> 0.0000	100.0000 / 99.0500
▶ Reporting	OK		<div style="width: 100%; height: 10px; background-color: green;"></div> 0.0180	99.9820 / 99.0500

Figure 8: IT services

Network map

By configuring a network map of the company's IT-infrastructure, it helps to display the topology visually. When a trigger on a host activates, it is highlighted on the map from where it is easy to identify the problems and dependencies. Maps are highly customizable and the level of details is up to the user to decide. (Zabbix map 2015)

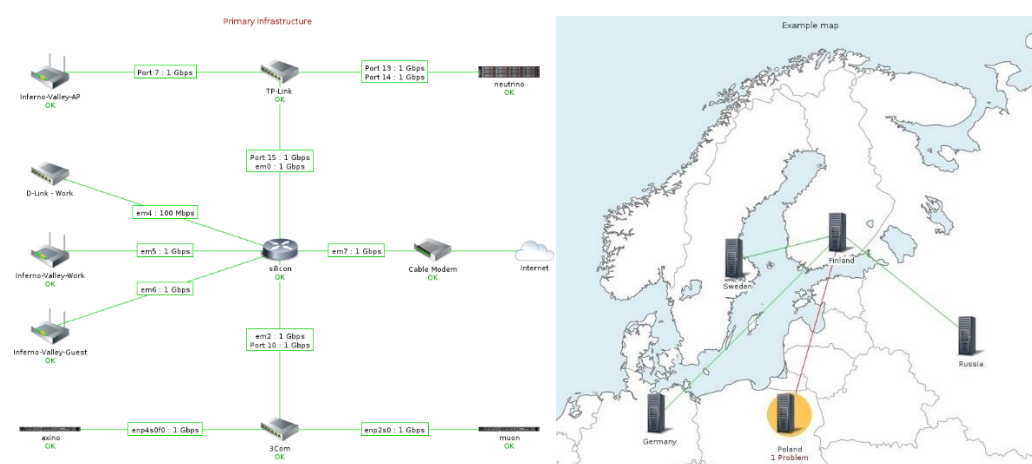


Figure 9: Example maps

3 Setting up Zabbix

In the following part, the Zabbix server 3.0 and agent installation procedure is demonstrated. The procedure is described, so the company can refer to this chapter when new systems are deployed to Zabbix or if the Zabbix server itself needs to be re-installed. For upgrading Zabbix to a newer version, the details can be found from Appendix 2.

No real-life IP addresses or host names are used. The practical part will also consist the different configuration steps to achieve fully functional monitoring platform. This includes adding host in the frontend and add suitable items, triggers, actions and templates to that specific host. The example host machine will be a real production Windows server, but the implementation will not show any specific details from that server due the fact that company's policy forbid that.

3.1 Preparations

Zabbix 3.0 is installed to a Red Hat Enterprise Linux 7 server, with dual core CPU, 4 Gb of memory and 50 Gb of hard disk space. The initial steps to configure Zabbix server successfully is to install web server, database server and PHP. The chosen database engine is MySQL, due to its reliability and popularity.

Currently Zabbix 3.0 supports following versions of the objects:

- Apache web server 1.3.12 or later
- MySQL 5.0.3 or later
- PHP 5.4.0 - 5.6.25, version 7 is not yet supported.

3.2 Zabbix server installation

To start the installation process, the package files should be retrieved. Zabbix server package files are available at Zabbix official repository, repo.zabbix.com. Version 3.0 is used in this implementation. This package contains yum configuration files. Yum is open source package handler software for RPM (Red Hat Package Manager) based Linux distributions. If Zabbix server is needed to configure with support for Jabber message alerting or to be able to ping hosts with simple checks or encrypt communication with TLS, the additional packages are in the non-supported directory in the official repository. To start the installation, add the configuration package to yum repository.

```
# rpm -ivh http://repo.zabbix.com/zabbix/3.0/rhel/7/x86_64/zabbix-release-3.0-1.el7.noarch.rpm
```

Run the command to install Zabbix server with MySQL database support, front-end web interface and Zabbix agent for internal monitoring

```
# yum install zabbix-server-mysql zabbix-web-mysql zabbix-agent
```

Editing PHP configuration for Zabbix frontend

Zabbix creates an apache configuration file to “/etc/httpd/conf.d/zabbix.conf” directory. Only value that needs to be changed is the time zone parameter. Since the installation is done in Finland the “Europe/Helsinki” is the correct value.

```
php_value max_execution_time 300
php_value memory_limit 128M
php_value post_max_size 16M
php_value upload_max_filesize 2M
php_value max_input_time 300
php_value always_populate_raw_post_data -1
php_value date.timezone Europe/Helsinki
```

After modifying the configuration file restart the apache web server.

```
# systemctl start httpd
```

MySQL database

The next step after apache installation is to create a Zabbix MySQL database. Create the Zabbix database and MySQL user with commands:

```
shell> mysql -uroot -p<your root password>
mysql> create database zabbix character set utf8 collate utf8_bin;
mysql> grant all privileges on zabbix.* to zabbix@localhost identified by '<new password for zabbix user on zabbix database>';
mysql> quit;
```

After MySQL creation, import initial schema and data. They are located in /usr/share/doc/zabbix-server-mysql-3.0.0.

```
# cd /usr/share/doc/zabbix-server-mysql-3.0.0
# zcat create.sql.gz | mysql -uroot zabbix
```

Edit database configuration in `zabbix_server.conf`.

```
# vi /etc/zabbix/zabbix_server.conf
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=zabbix
```

There is no need to change other server configuration settings at the beginning. The configurations may need some tuning after Zabbix has been running for a while.

When all the previous steps are done, the Start Zabbix server process can be started with a command:

```
# systemctl start zabbix-server
```

4 Zabbix Agent Installation for Windows

The Agent is the main component in charge of collecting values from the hosts. The Agent can be downloaded from Zabbix official homepage and is available for Windows and Unix operating systems. Once the file has been downloaded, it needs to be extracted to a host machine. The actual agent configuration is done through Zabbix agent configuration file, which is found from the newly extracted folder. Open the file in a text editor and configure the following settings. These settings will get you started and it is possible to modify them at any given point in the future. After the values are in place save the modifications and close the editor.

```
#LogFile directory
LogFile=D:\Zabbix\zabbix_agentd.log

# Allow remote commands for reactive monitoring
EnableRemoteCommands=1

# Log all remotely executed commands
LogRemoteCommands=1

# Zabbix Server (IP address for passive checks)
Server='Zabbix server IP'

#Server active (IP address for active checks)
ServerActive='Zabbix server IP'

#Hostname
Hostname= full name of the host computer
```

Zabbix agent saves all the execution data to a log file and the diversity depends on the “debug” level defined in the agent configuration file. Debug 3 is the default value and should be increased to display more data. When debugging the connection, the log file should be investigated. Passive agent uses port 10050 to communicate with the Zabbix server, and active agent port 10051. The inbound and outbound rules should be added to firewall configuration. An example of this happened when configuring monitoring to a point of sale server. It was noticed that the agent cannot communicate with Zabbix server, even though the ports were open in a third-party firewall the company was using. The reason turned out to be Windows firewall. In the other production servers Windows firewall was disabled, so at first it was not suspected being the cause of this. Once the firewall rules were added to the Windows firewall, the connection was established immediately.

4.1 Installing Zabbix agent

To install the agent on a Windows machine, open a command prompt with admin privileges and run the following commands. The path depends on where the agent was extracted, in this case it is: D:\Zabbix.

```
D:\Zabbix\zabbix_agentd.exe -i -c D:\Zabbix\zabbix_agentd.conf
```

```
D:\Zabbix\zabbix_agentd.exe -s -c D:\Zabbix\zabbix_agentd.conf
```

- i = installs zabbix agent
- s = starts zabbix agent
- c = configures the correct path for Zabbix agent configuration file

To check if the Zabbix agent service has successfully started, go to windows services and search Zabbix Agent from the list. The status should be now changed to 'Started'. From the services the agent can also be quickly restarted if needed, for example when changing the agent's configuration parameters.

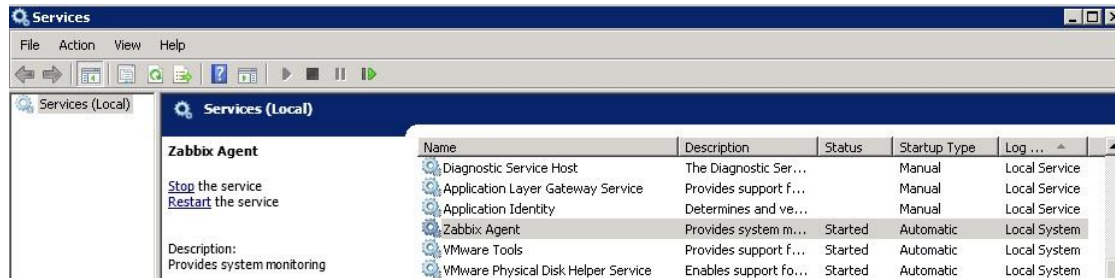


Figure 10: Agent service

5 Zabbix frontend configuration

Now that Zabbix server and agent have been configured and installed, it's time to start using Zabbix frontend. Frontend URL is: `http://<zabbix-server-ip>/zabbix`. When logging in for the first time use default Zabbix username and password (Admin/zabbix)

5.1 Adding a new user

The first thing to do is to create a new user, the preset Zabbix admin user should be deactivated or removed. To create a new user, go to administration tab > users > create user. From the user window, username, password, media and permissions are configured. The media tab, where the users email address is specified, is where Zabbix sends the alerts when a trigger is activated.

Zabbix user permissions are administrated through user groups. The user groups define the user rights they have and to what hosts. To be able to receive alert messages that user should have access to the corresponding host. This case it would be ideal to have all the rights to all the hosts, hence the user group is Zabbix administrators and from the permission tab Zabbix super admin needs to be selected.

5.2 Server host groups

Before it is time to add the host in the frontend, a host group needs to be created. Using host groups makes the host management more user friendly. Every system cluster should be in its own group e.g. ERP host in ERP host group and point of sale host in POS group. In this example the windows server group is used. In Zabbix web interface go to > Configuration > Host groups > Create host group > Windows servers. Host group can also be assigned and created when creating new hosts.

5.3 Adding new host

When installing Zabbix agent, the agent was successfully configured and installed to a monitored machine and it is ready to establish a connection to the Zabbix server.

From the Configuration > Hosts > Create host tab, a new host can be created.

- Enter your FQDN host name, this needs to be the same as in agent configuration file
- Visible name is the one displayed in Zabbix GUI
- Add group
- Add host IP
- Template tab allows user to add template to the new host. Since the machine is Windows environment, a Windows server template is chosen.
- Once everything is configured, clicking the add button makes the host go online

The screenshot displays the Zabbix web interface for creating a new host. The top navigation bar includes 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. The 'Configuration' menu is expanded to show 'Hosts', 'Maintenance', 'Actions', 'Discovery', and 'IT services'. The 'Hosts' page has sub-tabs for 'Host', 'Templates', 'IPMI', 'Macros', 'Host inventory', and 'Encryption'. The 'Host' tab is active, showing a form with the following fields and options:

- Host name:** Windows server
- Visible name:** Windows server
- Groups:** A list box containing 'Windows servers'. To the right, a list of 'Other groups' includes: Optos, Profix, Routers, SAP, Switches, Templates, TEST, Virtual machines, WEB services, and Zabbix servers.
- New group:** An empty text input field.
- Agent interfaces:** A table with columns for IP ADDRESS, DNS NAME, CONNECT TO, PORT, and DEFAULT. The first row contains 'xxxxxx.xxx', an empty field, 'IP', 'DNS', '10050', and a 'Remove' button. An 'Add' button is below the table.
- SNMP interfaces:** An 'Add' button.
- JMX interfaces:** An 'Add' button.
- IPMI interfaces:** An 'Add' button.
- Description:** Test server
- Monitored by proxy:** (no proxy) (dropdown menu)
- Enabled:**
- Buttons:** Add, Cancel

Figure 11: Host creation menu

Figure 12 displays the items that are included in Windows template. The template consists of the basic items and triggers that are needed for monitoring the hardware statistic, memory, hard disk, CPU, network bandwidth and other general information.

NAME ▲	TRIGGERS	KEY	INTERVAL	HISTORY	TRENDS
Template App Zabbix Agent: Agent ping	Triggers 1	agent.ping	1m	7d	90d
Template Windows Server: Average disk read queue length		perf_counter[234_Total]1402]	1m	7d	90d
Template Windows Server: Average disk write queue length		perf_counter[234_Total]1404]	1m	7d	90d
Template Windows Server: CPU utilization	Triggers 2	system.cpu.util	2m	7d	90d
Template Windows Server: File read bytes per second		perf_counter[2116]	1m	7d	90d
Template Windows Server: File write bytes per second		perf_counter[2118]	1m	7d	90d
Mounted filesystem discovery: Free disk space on C:	Triggers 4	vfs.fs.size[C:free]	1m	7d	365d
Mounted filesystem discovery: Free disk space on C: (percentage)	Triggers 6	vfs.fs.size[C:prfree]	1m	7d	365d
Template Windows Server: Free memory	Triggers 1	vm.memory.size[free]	2m	7d	90d
Template Windows Server: Free swap space	Triggers 1	system.swap.size[free]	2m	7d	90d
Template App Zabbix Agent: Host name of zabbix_agentd running	Triggers 1	agent.hostname	1h	7d	
Network interface discovery: Incoming network traffic on Intel(R) Ethernet Connection (3) I218-LM		net.if.in[Intel(R) Ethernet Connection (3) I218-LM]	5m	7d	365d
Template Windows Server: Number of processes	Triggers 1	proc.num[]	1m	7d	90d
Template Windows Server: Number of threads		perf_counter[21250]	1m	7d	90d
Network interface discovery: Outgoing network traffic on Intel(R) Ethernet Connection (3) I218-LM		net.if.out[Intel(R) Ethernet Connection (3) I218-LM]	5m	7d	365d
Template Windows Server: Processor load (1 min average)	Triggers 1	system.cpu.load[percpu.avg 1]	1m	7d	90d
Template Windows Server: Processor load (5 min average)		system.cpu.load[percpu.avg 5]	1m	7d	90d
Template Windows Server: Processor load (15 min average)		system.cpu.load[percpu.avg 15]	1m	7d	90d
Template Windows Server: System information	Triggers 1	system.uname	1h	7d	
Template Windows Server: System uptime	Triggers 1	system.uptime	1m	7d	90d
Mounted filesystem discovery: Total disk space on C:		vfs.fs.size[C:total]	1h	7d	365d
Template Windows Server: Total memory		vm.memory.size[total]	1h	7d	90d
Template Windows Server: Total swap space		system.swap.size[total]	1h	7d	90d
Mounted filesystem discovery: Used disk space on C:		vfs.fs.size[C:used]	1m	7d	365d
Template App Zabbix Agent: Version of zabbix_agentd running	Triggers 1	agent.version	1h	7d	

Figure 12: Template items

Figure 13 shows the templated triggers, which have been added to all windows hosts. They have been modified from the original template, since the triggers in the premade template were too sensitive. That caused a flapping phenomenon, triggers going on/off multiple times in a short period. Instead hard disk trigger to alert immediately when the space is under 20% it was changed to create an alert when hard disk space is under 20% for 15 minutes, same method was applied to all hard disk triggers. Similar modifications were implemented to other triggers as well.

Zabbix also supports hysteresis function to disable flapping. The concept is that once a trigger goes to problem state, it does not recover until the second parameter has come true. This was implemented in the CPU trigger, to create an alert when the average CPU utilization is over 95% for 15 minutes and to recover when the average CPU utilization has been below 75% for 10 minutes. The purpose behind all this is to prevent false alarms. Once a Zabbix alert has been received it should be treated as a real warning that some target might need some attention.

High	Template Windows Server: CPU util over 95% for 15min on (HOST.NAME)	((TRIGGER.VALUE)=0 and system.cpu.util.avg(15m))>95) or ((TRIGGER.VALUE)=1 and system.cpu.util.avg(10m))>75)
Average	Template Windows Server: Host information was changed on (HOST.NAME)	system.uname.diff(0)>0
Information	Template App Zabbix Agent: Host name of zabbix_agentd was changed on (HOST.NAME)	agent.hostname.diff(0)>0
Average	Template Windows Server: Lack of free memory on server (HOST.NAME)	vm.memory.size[free].last(0)<10000
Average	Template Windows Server: Lack of free swap space on (HOST.NAME)	system.swap.size[free].avg(10m)<100000
Average	Template Windows Server: Processor load is too high on (HOST.NAME)	system.cpu.load[percpu,avg1].min(15m)>5
Average	Template Windows Server: Too many processes on (HOST.NAME)	proc.num[] avg(5m)>300
Information	Template App Zabbix Agent: Version of zabbix_agentd was changed on (HOST.NAME)	agent.version.diff(0)>0
Average	Template App Zabbix Agent: Zabbix agent on (HOST.NAME) is unreachable for 5 minutes	agent.ping.nodata(5m)=1
Warning	Mounted filesystem discovery: (HOST.NAME): Free disk space is less than 1% on volume C:	vfs.fs.size[C:,free].last(0)<1
Warning	Mounted filesystem discovery: (HOST.NAME): Free disk space is less than 1% on volume C:	vfs.fs.size[C:,free].last(0)<1 and vfs.fs.size[C:,free].last(0)>=0
High	Mounted filesystem discovery: (HOST.NAME): Free disk space is less than 5% on volume C:	vfs.fs.size[C:,free].max(10m)<5 and vfs.fs.size[C:,free].last(0)>=0
Warning	Mounted filesystem discovery: (HOST.NAME): Free disk space is less than 5% on volume C:	vfs.fs.size[C:,free].last(0)<5
Warning	Mounted filesystem discovery: (HOST.NAME): Free disk space is less than 20% on volume C:	vfs.fs.size[C:,free].max(15m)<20 and vfs.fs.size[C:,free].last(0)>=0
Average	Template Windows Server: (HOST.NAME) has just been restarted	system.uptime.change(0)=0

Figure 13: Template triggers

To check if the connection is established, go to the host tab where is a list of all the added hosts. If the ZBX text on the right has turned green, the received data can be seen from the latest data tab.



Figure 14: Newly added host

The latest data tab shows all the data that is currently being gathered from the host. The items are arranged in applications, which are defined in the item creation step. Here a user can create simple and ad-hoc graphs of the items and view the incoming data in real time.

Item Name	Last Update	Current Value	Delta	Graph
CPU (4 Items)				
Processor load (15 min average)	2016-10-19 16:22:40	0.03		Graph
Processor load (5 min average)	2016-10-19 16:22:42	0.04		Graph
Processor load (1 min average)	2016-10-19 16:22:41	0.0042	-0.04	Graph
CPU utilization	2016-10-19 16:22:28	12.26 %	-16.19 %	Graph
Filesystems (8 Items)				
Used disk space on C:	2016-10-19 16:22:04	70.17 GB	-64 KB	Graph
Total disk space on C:	2016-10-19 15:35:03	238.47 GB		Graph
Free disk space on C: (percentage)	2016-10-19 16:22:02	70.57 %		Graph
Free disk space on C:	2016-10-19 16:22:01	168.3 GB	+64 KB	Graph
File write bytes per second	2016-10-19 16:22:37	365.95 KBps	+89.47 KBps	Graph
File read bytes per second	2016-10-19 16:22:36	392.57 KBps	-6.26 KBps	Graph
Average disk write queue length	2016-10-19 16:22:35	0		Graph
Average disk read queue length	2016-10-19 16:22:34	0		Graph
General (2 Items)				
Memory (4 Items)				
Total swap space	2016-10-19 15:34:44	7.88 GB		Graph
Total memory	2016-10-19 15:34:48	7.88 GB		Graph
Free swap space	2016-10-19 16:22:43	7.86 GB	-14.46 MB	Graph
Free memory	2016-10-19 16:22:47	4.28 GB	-134.53 MB	Graph
Network interfaces (6 Items)				
Outgoing network traffic on Intel(R) Ethernet Connection (3) I218-LM-WFP LightWeight Filter-0000	2016-10-19 16:22:12	5.81 Kbps	+1.58 Kbps	Graph
Outgoing network traffic on Intel(R) Ethernet Connection (3) I218-LM-QoS Packet Scheduler-0000	2016-10-19 16:22:10	5.78 Kbps	+1.56 Kbps	Graph
Outgoing network traffic on Intel(R) Ethernet Connection (3) I218-LM	2016-10-19 16:22:13	5.34 Kbps	+672 bps	Graph
Incoming network traffic on Intel(R) Ethernet Connection (3) I218-LM-WFP LightWeight Filter-0000	2016-10-19 16:21:25	37.75 Kbps	+32.01 Kbps	Graph

Figure 15: Latest data

5.4 Adding item and trigger

Item creation

Items are the ones that gather the actual data from the host. In this example, the monitored item will be “AdminService 10.2B” service’s state.

- Go to Configuration > Hosts > Select host > Items > Create item
- Name: something that describes the item which is polled
- Type: identifies the item type, whether it is passive/active check, SNMP, database query etc.
- Key: defines the gathered data (see all supported key functions - https://www.zabbix.com/documentation/3.0/manual/config/items/itemtypes/zabbix_agent)
- Type of information: depending on what kind of values are expected to receive, a correct parameter should be chosen from the drop-down menu. If the received value is numbers a numeric presentation should be chosen and for string values text or log.
- Update interval determines the polling frequency, how often a new value is received
- With custom intervals, it is possible to make the item poll only in a certain time. Custom intervals only function with passive checks.

Name: AdminService for OpenEdge 10.2B service state

Type: Zabbix agent

Key: service_state[AdminService10.2B]

Host interface: 172.16.0.65 : 10050

Type of information: Numeric (unsigned)

Data type: Decimal

Units:

Use custom multiplier: 1

Update interval (in sec): 60

TYPE	INTERVAL	PERIOD	ACTION
Flexible	Scheduling	50	1-7,00:00-24:00 <input type="button" value="Remove"/>

History storage period (in days): 90

Trend storage period (in days): 365

Store value: As is

Show value: Windows service state

New application:

Applications:

- None-
- CPU
- Custom Items**
- Filesystems
- General
- Memory
- Network interfaces
- OS
- Performance
- Processes

Figure 16: Item creation

Trigger creation

Once the item is successfully polling data and the values can be seen in the latest data tab, a corresponding trigger can be set. In figure 17, the trigger expression can be divided to four parameters which are always required:

- “Windowserver” is the hostname to where the trigger is created
- “service_state[AdminService10.2B]” is the item key, was created in previous paragraph
- “last(0)” is the trigger function, other available function can be found from Zabbix manual (<https://www.zabbix.com/documentation/3.0/manual/appendix/triggers/functions>)
- “<>0” is the value when trigger actives.

When the last received value of the service state is something else than 0, the trigger creates an alert. In Windows services, 0 equals that service is in “running” state.

Description should clearly tell what the trigger does, incase another user needs to find out the meaning of the trigger. Depending how severe the incident is, a correct severity should be chosen.

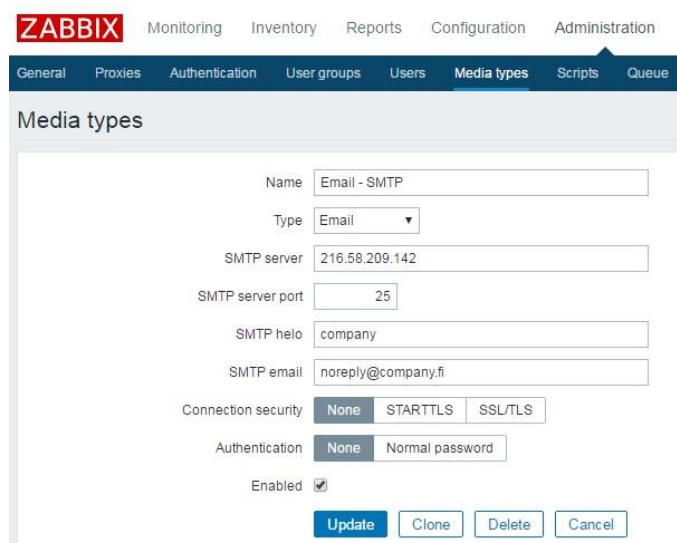
The screenshot displays the Zabbix Trigger configuration page. At the top, there are two tabs: "Trigger" (selected) and "Dependencies". The form contains the following fields and options:

- Name:** AdminService for Openedge 10.2B is not running
- Expression:** {Windosserver:service_state[AdminService10.2B].last(0)}<>0. An "Add" button is located to the right of the expression field. Below the field is a link for "Expression constructor".
- Multiple PROBLEM events generation:** A checkbox that is currently unchecked.
- Description:** Alerts when AdminService stops running.
- URL:** An empty text input field.
- Severity:** A set of radio buttons with options: Not classified, Information, Warning, Average (selected), High, and Disaster.
- Enabled:** A checked checkbox.
- Buttons:** Update, Clone, Delete, and Cancel.

Figure 17: Trigger creation

5.5 Configuring actions

Before Zabbix can send alerts via email, a media type must be declared. SMTP-email server forwards the alert messages, therefore a correct information is needed to configure in the fields. An example of configuration can be seen in figure 18. After the media type is declared, a user can start creating actions.



The screenshot shows the Zabbix web interface for configuring a media type. The navigation bar includes Monitoring, Inventory, Reports, Configuration, and Administration. The 'Media types' tab is selected. The form fields are as follows:

Field	Value
Name	Email - SMTP
Type	Email
SMTP server	216.58.209.142
SMTP server port	25
SMTP helo	company
SMTP email	noreply@company.fi
Connection security	None
Authentication	None
Enabled	<input checked="" type="checkbox"/>

Buttons: Update, Clone, Delete, Cancel

Figure 18: Media configuration

As stated earlier, actions oversee sending alerts to user's email, auto discovering hosts and executing remote command when certain trigger creates an alert. This example shows the creation of a basic action for email alerts. Actions are managed in frontend: Configuration > Actions > Create action. Action form is displayed in figure 19. The message box shows the details of the alert, which will be sent to user's email. When the trigger recovers to OK state, Zabbix can be configured to send an email of that as well.

Figure 19: Action form

Conditions are set of rules that define which alerts are sent to the user's email. The available conditions can be applied from the "New conditions drop-down menu" and the combinations are flexible. Different scenarios can be applied with the use "Type of calculation". In the example Figure 20, every trigger that goes to problem state in host "Windows server" are processed and sent to user's email which is defined in operations tab.

LABEL	NAME	ACTION
A	Maintenance status not in <i>maintenance</i>	Remove
B	Trigger value = <i>PROBLEM</i>	Remove
C	Host = <i>Windows server</i>	Remove

Figure 20: Conditions

From the operations tab select "Operation type" to "Send message", add the user to who the email will be sent and select "Send only to" the configured SMTP media source. It is important to specify actions in a way that nobody gets duplicate alerts to their email, merely because it can confuse and annoy the users if they receive multiple messages of the same alert.

Action Conditions **Operations**

Default operation step duration (minimum 60 seconds)

Action operations

STEPS	DETAILS	START IN	DURATION (SEC)	ACTION
1	1			

Operation details

Steps -

Step duration (minimum 60 seconds, 0 - use action default)

Operation type

Send to User groups

USER GROUP	ACTION
Add	

Send to Users

USER	ACTION
vpe (Valtteri Perttu)	Remove
Add	

Send only to

Default message

LABEL	NAME	ACTION
New		

Conditions [Add](#) [Cancel](#)

Figure 21: Operations

When the action configuration process is done and a trigger that fulfills the criteria's goes to problem state, an email alert will be sent to the corresponding user. Similar email is received when the trigger returns to OK state.

PROBLEM: AdminService for Openedge 10.2B is not running



Trigger: AdminService for Openedge 10.2B is not running
Trigger status: PROBLEM
Trigger severity: Average
Trigger URL:

Item values:

- AdminService for Openedge 10.2B service state Windows server: Stopped (6)

Original event ID: 63768

Figure 22: Email alert

6 Current state of monitoring

During this eight-month period, the most crucial systems are added in monitoring. Currently 56 hosts and 3502 items are monitored. The Zabbix database uses about 10Gb of hard disk space for storing history, trend and event data. The monitored IT infrastructure consists mostly of Windows servers, but there is also a Unix based host and several network switches. The triggers have been tuned specifically for each host. The only way to adjust the triggers, is to get familiar with the environment by tracking alerts and analyzing graphs.

Status of Zabbix		
PARAMETER	VALUE	DETAILS
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled/templates)	113	56 / 9 / 48
Number of items (enabled/disabled/not supported)	6168	3502 / 2662 / 4
Number of triggers (enabled/disabled [problem/ok])	1457	1365 / 92 [13 / 1352]
Number of users (online)	13	1
Required server performance, new values per second	47.49	

Figure 23: Status

To all hosts, a template with the standard monitoring was added. The standard items are CPU, memory, disk space, network bandwidth and some general information about the host. For every production host, there is a similar acceptance server with the same monitoring items and triggers. In addition to the standard metrics, in the following systems specific monitoring was implemented.

The PROD system take care of the product orders coming from the customers. It is essential for the business that there are no problems with the ordering. To identify the most common malfunctions, the required services to keep the system functioning are monitored, TCP port monitoring to see if a specific application responds to a query, file size monitoring to observe if the application files grow too large and a custom check to monitor the connection between PROD4 and PROD5 server. The custom check is achieved with a remote command and it is required to identify the time when a connection is lost between the servers.

The Enterprise service bus (ESB) is a Service Oriented Architecture (SOA) system that transfers the messages to a unified format and delivers them to different systems depending on the message type and the parameters. Daily, some messages get stuck in the message queues and needs human attention. Without needing to manually check the queue size and count, the queues are monitored. Also, the applications logfile is monitored to look for strings 'warning' and 'severe'. If those key strings are found, Zabbix sends an alert.

The Enterprise Resource Planning (ERP) system, is the business management software and the base of all operations. It is running on the Microsoft SQL database.

The ERP services that are required for the system to function and to observe the performance of the SQL server some SQL queries are monitored. The SQL performance data is obtained from the Windows performance counters.

The reporting servers creates reports for different business needs. There are multiple applications running on different time schedule, some daily and some only once a month. The log files are monitored to find out if the applications have been successfully executed. To identify the execution, a specific string is searched to appear in the log.

The Point of sale (POS) system is where all store cash transactions are done. Every night a script runs that sends warehouse and cash transaction files data between the POS and the ERP system and the log file is monitored to see if some data was unsuccessfully transferred. The process that handles the script is also monitored, to see if it is delayed or stuck.

A bash script was composed to monitor file modification times and with the help of Zabbix, run that script to identify if there are some problems with the POS system. In POS system, there are ESB adapters installed and their log file was monitored to see if the adapters are functioning.

The Opt system, is a service that provides customers more detailed way to discover what kind of additional services they might need. There is a service that needs to be running constantly and it is monitored and set a remote command to automatically start the service when it is not running anymore.

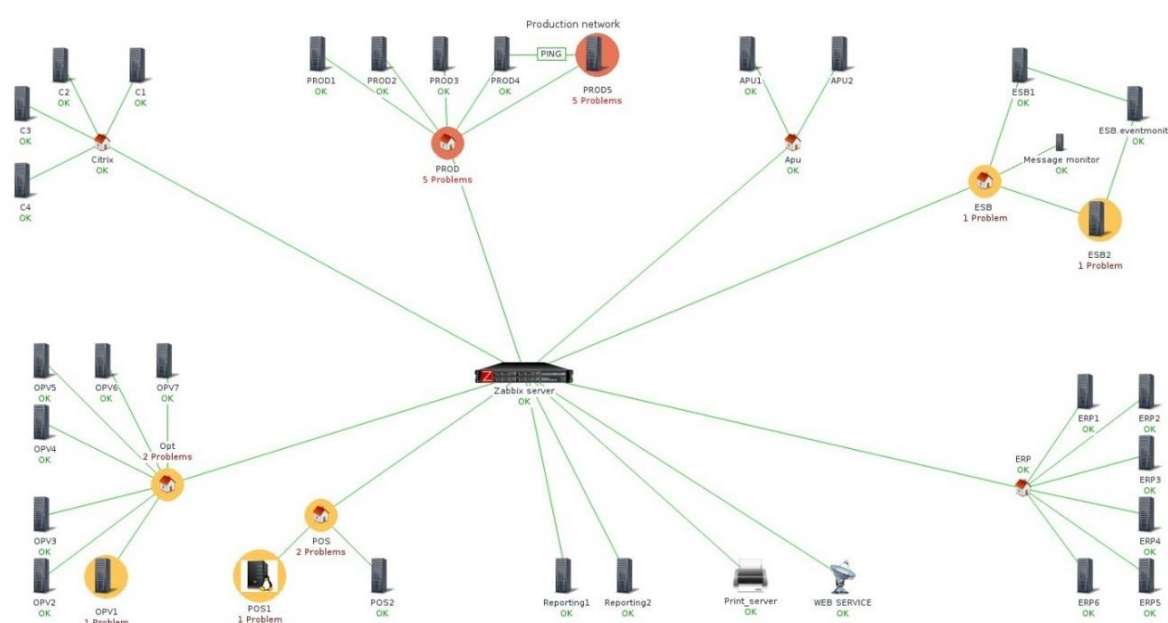


Figure 24: Production hosts

6.1 Future implementations

Here are a few ideas what could still be implemented to Zabbix. Some of these have already been discussed and are waiting for more information.

- Monitoring the SLA's by configuring IT-services is an upcoming task. Business processes need to be specified a bit different and more detailed way in order to have a clear understanding of the relations between business processes and technical incidents and therefore the monitoring the status of business processes cannot yet be implemented to Zabbix.
- The integration to the ServiceNow ticketing system could be done through ServiceNow API. A ready-made python script can be found from the Zabbix share community site. Zabbix would automatically create a ticket to ServiceNow when an alert that needs attention occurs. This is beneficial to track the time used in solving the problems. (Github 2015)
- Failover plan if Zabbix server crashes. At the moment, there is no automatically configured failover. A clustered solution is one possibility and, since the previous Zabbix server is still available it could be used. The idea is that when Zabbix becomes unavailable the switchover would happen and the secondary Zabbix server would start to poll the same items. More on this topic can be found from a book Mastering Zabbix. (Vacche and Lee 2013, 81)
- Monitoring database performance by configuring ODBC connection to production databases. The databases in use are Solid, Firebird, Progress and Microsoft SQL.
- Monitoring if backups are executed successfully. The backups are already monitored by a 3rd party vendor, but it would be suitable to add the backup status monitoring to Zabbix as well. One way to achieve this is to use the Windows task scheduler. A community made template can be found from the Zabbix wiki. (Zabbix wiki 2014)
- To implement more automatic corrective measures to the production servers. The idea is to automatically run a command or script for the most common problems to minimize the downtime of offered services. For example, run a script that will free disk space when it is about run out or to try to start some service when Zabbix notices they are not running anymore. This has already been tested on one system and the next step is to identify the situations where this could be exploited more.

7 Conclusion

As the main goals of the thesis were to introduce Zabbix and then implement it to a company's IT infrastructure, the company can now continue developing the monitoring based on the information given out on this thesis. They know more of Zabbix and what can be achieved when using it. The information is useful for the company, especially when they are adding new systems to monitoring or updating the existing ones.

For the company, Zabbix provides a reliable method to monitor the hardware, application and process statuses. Utilizing Zabbix, the company ensures that the problems are noted and identified immediately. By achieving proactive monitoring by preventing the incidents, the company's services are highly available which makes it unlikely that business continuity would be endangered.

The thesis was a continuation to the previous work done at the work placement. When starting to write this thesis, knowledge of Zabbix and its basic capabilities were already obtained. Because of that, the most important aspects of the software were easier to identify and describe. Even still a lot of new information was found about different network protocols and features. During the implementations, some minor problems occurred occasionally, but they were resolved with the help of Zabbix forum and IRC channel.

To sum up Zabbix in a few words, it is very customizable and can monitor a wide range of objects from the box and the users can develop their own solutions to obtain data by using remote commands, scripts or even custom agents. If the users are willing to use some time to learn how Zabbix functions and accept that there is somewhat a high learning curve to configure everything properly, it is a great open-source platform for multiple different scenarios.

References

- Birch, S. 2016. The history of Network Management: An infographic. Iris, 31 March. Accessed 25 September 2016.
<https://www.irisns.com/the-history-of-network-management-an-infographic/>
- Brochure 2016. Zabbix general brochure. Accessed 16 September 2016.
http://www.zabbix.com/files/Brochures/General_Brochure_3.2.pdf
- Ellingwood, J. 2014. An Introduction to SNMP (Simple Network Management Protocol). Digitalocean, 18 August. Accessed 1 October 2016.
<https://www.digitalocean.com/community/tutorials/an-introduction-to-snmp-simple-network-management-protocol>
- External. 2015. External scripts. Accessed 16 October 2016.
<https://www.zabbix.com/documentation/3.0/manual/config/items/itemtypes/external>
- Github. 2015. ServiceNow tickets from Zabbix. Accessed 20 October 2016.
<https://github.com/monitoringartist/zabbix-script-servicenow>
- Nash, K and Behr, A. 2009. Network Monitoring Definitions and Solutions. CIO, 5 June. Accessed 15 September 2016.
<http://www.cio.com/article/2438133/networking/network-monitoring-definition-and-solutions.html>
- Olups, R. 2016. Zabbix Network Monitoring: Second edition. Accessed 15 October 2016.
<https://www.packtpub.com/books/content/encrypting-zabbix-traffic>
- Technopedia. 2016. Network Management System (NMS). Accessed 15 September 2016.
<https://www.techopedia.com/definition/11988/network-management-system-nms>
- Upgrade. 2016. Zabbix Upgrade Procedure from 2.4 to 3.0 on CentOS 7. Techbylooney, 25 March. Accessed 28 October 2016.
<http://techbylooney.com/tutorials/linux-software-tutorials/zabbix-upgrade-procedure-from-2-4-to-3-0-on-centos-7>

US-CERT. 2013. Risks of Using Intelligent Platform Management Interface (IPMI).
26 July. Accessed 10 October 2016.
<https://www.us-cert.gov/ncas/alerts/TA13-207A>

User parameter. 2015. userparameters.
Accessed 15 October 2016.
<https://www.zabbix.com/documentation/3.0/manual/config/items/userparameters>

Vacche and Lee. 2013. Mastering Zabbix: Monitor your large IT environment efficiently with Zabbix. Accessed 10 October 2016
<http://site.ebrary.com/nelli.laurea.fi/lib/laurea/reader.action?docID=10822192>

Vladishev, A. 2011. Zabbix: Interview of Alexei Vladishev.
Monitoring-fr, 2011. Accessed 20 September 2016.
http://www.monitoring-fr.org/wp-content/uploads/2011/06/Interview_AlexeiVladishev_EN1.pdf

Zabbix agent. 2015. concepts/agent.
Accessed 1 October 2016.
<https://www.zabbix.com/documentation/3.0/manual/concepts/agent>

Zabbix database. 2015. ODBC checks.
Accessed 7 October 2016.
https://www.zabbix.com/documentation/3.0/manual/config/items/itemtypes/odbc_checks

Zabbix graph. 2015. Graphs.
Accessed 5 October 2016.
<https://www.zabbix.com/documentation/3.0/manual/config/visualisation/graphs/>

Zabbix map. 2015. Map.
Accessed 16 October 2016.
<https://www.zabbix.com/documentation/3.0/manual/config/visualisation/maps/map>

Zabbix PSK. 2015. Using pre-shared keys.
Accessed 9 October 2016.
https://www.zabbix.com/documentation/3.0/manual/encryption/using_pre_shared_keys

Zabbix product. 2015. Product Overview.
Zabbix, 2015. Accessed 15 September 2016
<http://www.zabbix.com/product>

Zabbix proxy. 2015. Proxy.
Accessed 1 October 2016.
<https://www.zabbix.com/documentation/3.0/manual/concepts/proxy>

Zabbix requirements. 2015. Installation/Requirements.
Accessed 29 September 2016.
<https://www.zabbix.com/documentation/3.0/manual/installation/requirements>

Zabbix screens. 2015. Screens.
Accessed 5 October 2016.
<https://www.zabbix.com/documentation/3.0/manual/config/visualisation/screens>

Zabbix security. 2016. Security Monitoring.
Accessed 15 September 2016.
http://www.zabbix.com/security_monitoring.php

Zabbix server. 2015. concepts/server.
Accessed 25 September 2016.
<https://www.zabbix.com/documentation/3.0/manual/concepts/server>

Zabbix simple checks. 2015. Simple checks.
Accessed 1 October 2018
https://www.zabbix.com/documentation/3.0/manual/config/items/itemtypes/simple_checks

Zabbix template. 2015. Templates.
Accessed 5 October 2016.
<https://www.zabbix.com/documentation/3.0/manual/config/templates>

Zabbix web. 2015. Web monitoring.
Accessed 5 October 2016.
https://www.zabbix.com/documentation/3.0/manual/web_monitoring

Zabbix wiki. 2014. Monitoring Windows 2008, 2008R2, 2012 Server backups.
Accessed 20 October 2016.
https://www.zabbix.org/wiki/Monitoring_Windows_2008,_2008R2,_2012_Server_backups

Figures

Figure 1: Zabbix Architecture (Brochure 2016).....	9
Figure 2: Zabbix hardware requirements (Zabbix requirements 2015).....	10
Figure 3: Zabbix 3.0 web frontend	11
Figure 4: Proxy concept (Zabbix proxy 2015)	14
Figure 5: Simple graph	15
Figure 6: Ad-hoc graph.....	15
Figure 7: Zabbix custom graph	16
Figure 8: IT services	19
Figure 9: Example maps	19
Figure 10: Agent service	24
Figure 11: Host creation menu	26
Figure 12: Template items	27
Figure 13: Template triggers	28
Figure 14: Newly added host.....	28
Figure 15: Latest data.....	28
Figure 16: Item creation.....	29
Figure 17: Trigger creation.....	30
Figure 18: Media configuration	31
Figure 19: Action.....	32
Figure 20: Conditions.....	32
Figure 21: Operations	33
Figure 22: Email alert	33
Figure 23: Status	34
Figure 24: Production hosts	35

Appendices

Appendix 1: Upgrade notes.....	44
Appendix 2: Upgrade procedure Zabbix 3.0→3.2	46

Appendix 1: Upgrade notes

As stated in the introduction an analysis of whether the system should be upgraded or not was formed. The previous system was Zabbix version 2.0 and the implemented was 3.0. A new version 3.2 was released at the end of making this thesis and for possible future upgrading its new features are also displayed.

Zabbix 2.2.x

- Virtual machine monitoring with LLD (low-level discovery)
- Notifications on unsupported items, unknown triggers
- Support of internal checks for proxies
- Official database monitoring
- Automatic database upgrade
- Network map improvements
- Housekeeper tasks can be enabled/disabled on a per-task basis
- SNMP v3 support
- Editable discovery checks
- Frontend improvements

Zabbix 2.4.x

- Graphs for several items
- New trigger operators
- Web monitoring improvements
- SSL verification and authentication
- New filter options
- More details about unsupported agent item
- Frontend improvements
- VMware monitoring improvements

Zabbix 3.0

- New frontend design
- Encryption support
- Predictive trigger functions (forecasts)
- Item checking at specific times
- Performance improvements
- Graph improvements
- VMware monitoring improvements
- Zabbix agent crash log on Windows platforms

Zabbix 3.2

- Event correlation
- Event tags
- Close problems manually
- Improved trigger hysteresis
- Recovery operations
- New filters
- LLD item, trigger, graph modifications
- VMware monitoring improvements

Appendix 2: Upgrade procedure Zabbix 3.0→3.2

1. Stopping the Zabbix server
`service zabbix-server stop`
2. Backup MySQL database
`mkdir /zabbixbackup`
`mysqldump -uzabbix -pzabbix zabbix > zabbixbackup/zabbix.sql`
3. Backup zabbix directory
`zip -r /zabbixbackup/zabbix.conf.zip /etc/zabbix`
4. Copy zabbix.conf file
`cp /etc/httpd/conf.d/zabbix.conf /zabbixbackup`
5. Backup Zabbix binaries
`zip -r /zabbixbackup/zabbixbinaries.zip /usr/share/doc/zabbix*`
6. Delete Zabbix 3.0 repository
`rpm -e zabbix-release-3.0*`
7. Install new repository configuration file and clean yum cache
`rpm -ivh http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-3.2-1.el7.noarch.rpm`
`yum clean all`
8. Upgrading
`yum -y upgrade zabbix-server-mysql zabbix-web-mysql zabbix server zabbix agent`
9. Start server
`service zabbix-server start`

Zabbix support automatic database upgrade from Zabbix 2.2→, hence the database should be automatically upgraded. Database upgrading can take up to two hours depending on the database size. After reviewing the configuration files, Zabbix frontend should be now accessible. (Upgrade 2016)