

Anatolii Shokhin


Network monitoring with Zabbix

Bachelor's Thesis
Network monitoring

May 2015



DESCRIPTION

		Date of the bachelor's thesis 28.05.2015
Author(s) Anatolii Shokhin	Degree programme and option Information Technology	
Name of the bachelor's thesis NETWORK MONITORING WITH ZABBIX		
Abstract Network monitoring became an essential part of any network size. It brings monitoring of network components 24/7. This provides clarity and transparency of network infrastructure and performance. With on time alert notification, network administrator may start failure troubleshooting immediately. As a result, end users may not even notice the issue.		
Subject headings, (keywords) Network Monitoring, Zabbix, Agentless monitoring, Agent-based monitoring.		
Pages 71	Language English	URN
Remarks, notes on appendices		
Tutor Matti Koivisto	Employer of the bachelor's thesis Bachelor's thesis assigned by Mikkeli University of Applied Sciences	

CONTENTS

1 INTRODUCTION	1
1.1 My goals of the thesis.....	2
2. NETWORK MONITORING TOOL.....	3
2.1 How was it before?.....	3
2.2 What is network monitoring?	3
2.3 Types of monitoring	4
2.3.1 Network monitoring	4
2.3.2 Route analytics	4
2.3.3 Website monitoring.....	5
2.3.4 Innovations in Monitoring.....	5
2.3.5 Conclusion.....	5
2.4 What are the changes for IT departments?	6
3 CAPABILITIES OF NETWORK MONITORING	6
3.1 Open source vs Proprietary	6
3.2 Agent-based vs Agentless	7
3.2.1 Agent-based.....	7
3.2.2 Agentless	8
3.2.3 Conclusion.....	8
3.3 Auto discovery	9
3.4 Low-level discovery	10
3.5 Trend Prediction	11
3.6 Logical grouping	11
3.7 Conclusion.....	12
4. ZABBIX	12
4.1 Zabbix Overview	12
4.1.1 Zabbix history	13
4.1.2 Zabbix Features	13
4.1.3 Zabbix architecture.....	14
4.2 Advantages of Zabbix	15
4.3 Things to improve in Zabbix	16
4.3.1 Web Interface	16
4.3.2 API	17
4.3.3 Reporting.....	18

4.3.4 Scalability.....	18
4.3.5 Security	19
4.3.6 Conclusion.....	19
4.4 Zabbix on a code level	20
4.4.1 C language in Zabbix	20
4.4.2 PHP in Zabbix	20
4.4.3 SQL in Zabbix.....	21
4.4.4 Advantages of Zabbix architecture	21
4.4.5 The challenges of Zabbix Architecture	22
4.5 Memory Interaction.....	22
4.5.1 Cache.....	22
4.5.2 Bulk operations	23
4.6 Zabbix SIA revenue model	24
4.6.1 Professional Training Program	24
4.6.2 Technical Support	25
4.6.3 Consulting	25
4.6.4 Turn-Key Solution	26
4.6.5 Template building	27
4.6.6 Conclusion.....	28
4.7 Zabbix 2.4.....	28
4.7.1 Optional SNMP bulk.....	28
4.7.2 Flexible filter for Low-Level Discovery	28
4.7.3 Runtime control of log level	28
4.7.4 Node-based Distributed Monitoring is removed.....	29
4.7.5 Minor changes and improvements	29
4.7.6 Upgrading to Zabbix 2.4	30
4.8 Future of Zabbix.....	31
5 PREPARATIONS	31
5.1 Scenario.....	31
5.2 Network Design.....	32
5.3 Capacity Planning	33
5.4 Memory Planning.....	34
5.5.1 History Data Planning	35
5.5.2 Trends Data Planning.....	35
5.5.3 Event Data Planning.....	35

5.5.4 Conclusion.....	36
6 ZABBIX INSTALLATION AND TESTING.....	36
6.1 Installation.....	36
6.1.1 Linux Server Installation.....	37
6.1.2 Installing Zabbix Server.....	38
6.1.3 Log in to Zabbix.....	39
6.1.4 Conclusion.....	40
6.2 Monitoring of Zabbix Server.....	40
6.3 SNMP Monitoring.....	40
6.3.1. Host Creation.....	41
6.3.2 Item Creation.....	42
6.3.3 Creating Graphs.....	43
6.3.4 Conclusion.....	44
6.4 Monitoring with Templates.....	45
6.4.1 Router monitoring.....	46
6.4.2 Router 2 monitoring.....	48
6.4.3 Server monitoring.....	48
6.4.4 Conclusion.....	50
6.5 Agent Monitoring.....	50
6.5.1 Installing Zabbix Agent.....	51
6.5.2 Adding Templates.....	51
6.5.3 Conclusion.....	52
6.6 Auto Discovery.....	52
6.6.1 Network Topology.....	53
6.6.2 Creating Discovery Rule.....	55
6.6.3 Creating Action.....	57
6.6.4 Performing Auto Discovery.....	58
6.6.5 Conclusion.....	62
7 CONCLUSION.....	62
BIBLIOGRAPHY.....	64

1 INTRODUCTION

Nowadays an Information Era has dawned. Data travels with a speed of light. During last 30 years business has radically changed. The main business goal in the 1980s was quality and business reengineering in the 1990s. Nowadays speed is the main concept.

If a retailer or production-based company is able to react to market changes in several hours, rather than days or even weeks, it provides a sufficient advantage among competitors. In addition, company would have a clear vision of customer's expectations. In order to react fast on market changes, a company has to be so organized that within short period of time it will shift to a newer strategy.

Information and Communication Technology (ICT) is playing an essential part in modern enterprise. It helps to organize and even automate some internal processes. ICT helps to structure workload within the company. In addition, it simplifies the communication with external processes such as contacts with suppliers and clients.

Bill Gates (2009) compared a modern enterprise network with a human nervous system. The goal of the natural nervous system is to analyze the environment and inform us about issues that have to be solved primarily and which are not relevant. The same principle could be adopted by modern corporate networks where the principle of the nervous system would allow constant and effective development, fast reaction to emergencies and rapid flow of urgent data. In addition, it would allow the movements of deeply integrated data flows, supplying with it right parts at a right time, which would help a company to perceive market and to easily adapt to any changes. Mr. Gates has described this phenomenon as the "digital nervous system". (Gates 2009.)

Modern enterprise networks, in order to be efficient, must support a wide range of applications, protocols and services. There are four basic characteristics of modern enterprise network architecture: fault tolerance, scalability, quality of service and security. (Cisco 2014.) The inability of these features may paralyze the whole "digital nervous system" of a company and as a result, all internal operations would be stacked.

In order to provide four basic characteristics of network, ICT infrastructure has to be designed and implemented well. However, even in most advanced networks it is impossible to provide

100% availability. Issues and failures will be appearing from time to time. In order to be able to fast react on them, network monitoring is widely used.

The main reason of using network monitoring is to get informed about the problem faster that user would notice. Network monitoring provides rapid diagnose of a current network situation. In case of failure, it would notify network administrator with an alert. For example, there is a failure of email server at 12:30, with a help of monitoring tool, network administrators will be informed about an issue in a minute. Without monitoring administrators would know about it only at 15:00 after a call from an irritated user.

Nowadays there are many network monitoring tools presented on a market. Most of them provide monitoring of network bandwidth. In addition, they allow to collect the performance data of network devices. The differences between motoring tool presented by various vendors could be in additional features such as trend prediction and logical grouping. In addition, diverse companies offering network monitoring are having different model spread wide.

Zabbix is releasing under General Public License version 2 (GPLv2). As a result, Zabbix is offered completely for free. There are no limitations neither in capabilities nor in amount of monitored devices. In addition, Zabbix is based on open source code. That is why, it has a strong community support. (Zabbix 2014.).

As a result Zabbix has been installed in 500'000 network only in 2013 (Zabbix 2014.). In addition, Zabbix has been recently recognized with awards and rankings from authoritative publications. That is why Zabbix has become an important player on a market.

1.1 My goals of the thesis

With my thesis, I am expecting to reach two main goals. My thesis consists of two parts: theoretical and practical. That is why expectations from two parts are diverse.

In theoretical part, I am expecting to develop an understating of key concepts of network monitoring. At the beginning, attention is going to be paid to advantages that are brought by network monitoring. After that, I will develop an understanding of network monitoring features. In addition, I will focus on principles of how network monitoring is working. At the end

of theoretical part, I will analyse Zabbix from technical perspective. Strong sides and drawbacks are going to be outlined by me. I will analyze the future developments of the Zabbix. In addition, I will make a research on a company's processes and revenue model.

In practical part, I will implement knowledge and understanding that I have gained from theoretical part on real environment. I will perform testing of Zabbix on small sized network that is going to consist of routers, switch and a few servers. I will go through different techniques and methods of setting up monitoring, in order to define the optimal one. In addition, all the methods and techniques that are going to be used in practical part could be implemented in real environment on any type of networks, regardless of their size and complexity.

2. NETWORK MONITORING TOOL

In this chapter I will define what network monitoring is. I will explain what changes are brought with networks monitoring for an IT department. In addition, I will outline different types of monitoring.

2.1 How was it before?

A few decades ago, information about network status and infrastructure performance was only collected in one place. The role of the "intellectual analyzer" was on a system administrator. In order to have a full overview of the system this person should be like a pilot inside a jet. He/she had to monitor everything at the same time. It was taking too much time to find out problems and the method of monitoring was just a waste of human resources. That is why, in order to liberate system administrators from routine work network monitoring should be an essential part of modern networks.

2.2 What is network monitoring?

Network monitoring is a system that indicates slow network performance or non-working network devices. The monitoring is based on the analysis of throughput, error rates, packet loss, and latency, availability of routers and switches and response time. If some failure occurs, the network administrator receives a notification about the failure through a warning banner, email, phone and other alarms. (Krock 2012.)

Network monitoring also performs the role of strategic tools in modern enterprises. It helps to optimize the data flow and to detect unreliable equipment. In addition, it verifies the capacity of devices and their conditions such as temperature and utilization rate. As a result, network monitoring helps to maximize network performance and decreases the potential failures of the network. The main advantages of the optimized network for enterprises are decreased infrastructure costs, employee productivity as well as productivity and fast and reliable data flow. (Krock 2012.)

There is a common misunderstanding that network monitoring also provides security audit and prevents from unauthorized access to the network. For this kind of security monitoring Intrusion Prevention Systems (IPS) or Intrusion Detection Systems (IDS) are used. Network monitoring is only used for network utilization and reliability monitoring. (Observium 2013.) Network monitoring supports a wide range of devices like servers, routers, switches and even end devices. In addition, it could be used in any kind of networks such as WLAN, LAN, VPN and even WAN.

2.3 Types of monitoring

Network monitoring tools provide a wide range of scanning and analyzing activities for different types of devices and services. This has been reached by using different types of protocols operating on different OSI layers.

2.3.1 Network monitoring

Network monitoring is used to measure an overall network performance. The measurement is done by comparing the amount of transmitted and received packets. During that process hop count (amount of intermediate devices in order to reach a destination) is measured. In addition, path propagation and network devices delays are measured. That is why, it is possible to estimate packet loss, bandwidth and latency. As a result, this increases the quality of service in the network. (Nowak 2004.)

2.3.2 Route analytics

Another essential part of monitoring is route analytics. It is a set of tools, techniques and algorithms that monitor routing inside a network. It operates on the Network Layer.

Routing analytics passively peers and monitors OSPF, IS-IS, EIGRP and BGP routing protocols. As a result, it receives every update message as all the other routers. In addition, it uses the Dijkstra algorithm that calculates a complete network topology map, including all paths. Moreover, it records a complete routing event history that could be used later for troubleshooting. Overall, route analytics increases the speed and efficiency of the network and also helps to cut the costs and to increase employees' productivity. (Coates 2004.)

2.3.3 Website monitoring

Website motoring provides monitoring of server status. It measures server availability, performance, connectivity, uptime, DNS records, bandwidth and even hardware resources. There are two types of web site monitoring: internal and external. Internal monitoring (inside corporate firewall) is responsible for detecting issues related to internal infrastructure or for designing applications. External monitoring (outside corporate firewall) is responsible for end-to-end monitoring. Web site monitoring is responsible for these Internet protocols: HTTP, HTTPS, FTP, SNMP, STPM, SSH, TELNET, POP3, DNS, SSL, TCP, UDP. (Packet Design Inc. 2015.)

2.3.4 Innovations in Monitoring

Monitoring from the end user perspective is when a robot periodically emulates user's actions. It triggers a special script, as if the user runs through menus and even performs clicks. If the robot is not able to perform some actions, the end user cannot do it either.

One of the latest features that has appeared is monitoring at the code level. This mainly applies to applications in J2EE and .NET. Such modules can determine the delay in system calls, memory leaks and delays in the implementation of SQL queries. (Krock 2012.)

2.3.5 Conclusion

There are many types of monitoring. They have design to perform different types of checks. Their working principals are completely different. In my thesis I will concentrate on network-ing monitoring

2.4 What are the changes for IT departments?

With the usage of the network monitoring tools network administrators are liberated from routine work and can concentrate on more important tasks. They will have a full picture about network processes at the exact moment. In addition, troubleshooting of routine failures could be atomized by special script. There is no doubt that some unexpected failures must still be solved manually. However, it will be considerably easier due to the detailed diagnosis of the problem.

3 CAPABILITIES OF NETWORK MONITORING

When there is a time to choose the network monitoring tool network administrator is facing a big challenge. Right now, there are more than 50 tools provided by different vendors on the market. Most of them provide a wide range of possibilities. However, there are some differences which may play an important role in the network. That is why, in order to choose the best option for a company's own network some key features should be analyzed.

3.1 Open source vs Proprietary

Nowadays on a market two types of monitoring applications could be found. The first one is monitoring tool with an open source code. Usually such systems are released under GPLv2 license. That is why, third-party developers are allowed to make changes on code level. The second type is proprietary monitoring tools. The license restricts making any kind of modification on a code level.

From my perspective, open source network monitoring is bringing considerably more advantages compared to propriety. In my opinion, end user will benefit by amount of capabilities that are brought by open systems.

Despite the fact that network monitoring vendors are trying to include most relevant features for monitoring, it is almost impossible to create one solution which would be optimal for any network. Different networks have different needs. This is where the open source network monitoring tool is bringing its own advantages. If some feature which is relevant for network monitoring is not included in the version out of box, with sufficient skills and knowledge, it can be created by the network administrator or downloaded from the community.

In addition, companies that offer their products with open source code also benefit from this, due to the fact they are experiencing the crowd sourcing effect. When independent developers make some plugins and additional features, most popular of them could be added into the next version of the product. This provides extra flexibility and gives the company information about trends in network monitoring. Overall, open source network monitoring tools are bringing considerable advantages for network monitoring and help expand some features that were not provided by vendors.

3.2 Agent-based vs Agentless

Another decisions for system administrators is to choose if the network monitoring should be agent-based or agentless. There is no one optimal solution for each network. It depends on what level of monitoring is expected to be achieved, on the type of network and even on the budget. That is why it is better to understand the key differences between them in order to choose the best solution for the company's own network.

3.2.1 Agent-based

Agent-based monitoring consists of a piece of software that is called the agent. The agent is an application that is installed locally on servers and other network devices. Its goal is to monitor network performance. If some failure occurs, a warning message is created. In addition, the agent could troubleshoot some failures.

Agent is a lightweight application. However, some of them could heavily consume network resources. As a result, the whole idea of using network monitoring is disappearing. That is why the lightweight agent or the so-called "invisible" agent is gaining popularity nowadays. Its main advantage compared to the traditional one is that there are no footprints on the network performance without any lack in monitoring performance. (Uptime Software Inc. 2014.)

The key benefit of agent-based monitoring is that it provides deeper network analytics. In addition, agent-based monitoring tools could even diagnose hardware performance. It provides also alerting and reporting capabilities. Some failures could be automatically troubleshot. (EG 2013.)

The main disadvantage is that the deployment of such a system is a time consuming process, where many details of the network should be taken into account. In addition, the agents should be updated. The traditional agent-based solution could have a footprint on network performance. It should also be taken into account that agent-based monitoring tools have a considerably higher cost for license. (Uptime Software Inc. 2014.)

3.2.2 Agentless

Agentless is a solution that does not require any separate agent installation. Network analysis is based on direct packet monitoring. It used to monitor network availability and performance. However, it does not provide any detailed information about the failure.

Agentless monitoring is usually based on SNMP (Simple Network Monitoring Protocol) or WMI (Windows Management Instrumentation). It is based on a central management station that monitors all other network devices. However, agent-based solutions provide deeper metrics. (Uptime Software Inc. 2014.)

The main advantage of using agentless monitoring is that there is no agent needed. That is why there is no footprint on network performance. The process of deployment is easier. In addition, there will be no need for regular updating of the agent. In addition, the cost is relatively lower. (EG 2013.) The biggest disadvantage of using agentless monitoring is that there is no deep metrics. In addition, agentless provides neither reporting nor analytical features.

3.2.3 Conclusion

There is no single answer which technique is the best. This is a matter of network type and how complicated the topology is. In addition, it is the matter of the budget.

It is recommended to use agent-based monitoring in large networks with a complicated infrastructure. If a failure occurs, agent-based monitoring tools would alert about the failure. They will also try to solve a problem automatically. If there is a non-trivial failure, network administrators' attention would be needed. An agent-based monitoring tools would provide detailed information on where and which kind of problem has occurred. That is why finding and replacing a failure requires less time. (Uptime Software Inc. 2014.)

Agentless monitoring is more preferable in small-sized networks which consist of few network devices. Only the monitoring for network availability and performance is needed. There is no need for detailed information about metrics and the status of network devices in these types of networks. (Uptime Software Inc. 2014.)

It is recommended to use both agent-based and agentless network monitoring. Nowadays, vendors like Nagios or Zabbix provide both agent-based and agentless capabilities in their solutions. Essential parts of the network, where the availability and performance are prioritized, agent-based monitoring could be used. For the less important parts of the network agentless is more preferable. (Nagios 2015; Zabbix 2015)

As a result, a monitoring solution which combines both agent-based and agentless monitoring would have the advantages of each type. A combined solution would help to prioritize which parts are more important to be monitored in detail. This would decrease the footprint on network performance. In addition, only important information would be reported. That is why network administrators would only have relevant data about network changes.

3.3 Auto discovery

One of the challenges of a network manager is to keep the network management system up to date within all dynamic changes. New devices are added in most networks environments weekly if not daily. To keep track of the constantly changing environments auto discovery is used.

Auto discovery is a feature that allows perform a search of network elements. In addition, it automatically adds new devices and removes the ones that are no longer part of the network. It also performs the discovery of network interfaces, ports and file systems. (Zabbix 2015.)

Auto discovery could be used to figure out current situation in the network. What devices and services are currently on the network? In addition, it helps in security issues. It helps to verify what ports are enabled.

Auto discovery could ping or query every device on a network. If a network has an Intrusion Detection System (IDS), auto discovery may trigger the intrusion alarm. The reason is that

auto discovery could also be used for hacking. An attacker could get the whole picture of the network with all the devices and services. (Tibbo Technology 2014.)

Despite the fact that auto discovery plays an essential part in network monitoring, some tools on the market do not provide this feature. That is why network administrators should pay attention to the presence of auto discovery when choosing the network monitoring tool.

3.4 Low-level discovery

Low-Level Discovery (LLD) is used for monitoring file systems and network interfaces without any need in creating and adding manually each element. Low-level discovery is a dynamic feature that automatically adds and removes elements. It also automatically creates triggers and graphs for file systems, network interfaces and SNMP tables. (Zabbix 2013.)

Before the widespread acceptance of LLD, templates were in use. However, creating a template is a time consuming process. Each template should manually create a trigger for each port or logical disk. In addition, there is a need to specify what the trigger should be aware of. For example, a network has a switch with 24 ports. In order to monitor port status, the template should be created by using SNMPv2 and IF-MIB triggers. Up to 14 elements could be created, of course, depending on network policies. After that, a template should be copied 23 times once for each port. Also, it should be taken into account that networks consist of multiple switches and other devices. That it is why even more time is required to spread triggers across all the network devices. This is a routine and time-consuming process for a network manager. (Zabbix blog 2013.)

Instead of manually creating templates for each port, LLD allows creating prototypes of data elements and triggers prototypes only once. After that LLD automatically discovers ports, file systems and SNMP tables. Several different triggers could be created that meet different needs (Zabbix blog 2013.)

In addition, low-level discovery could be used in order to monitor the workload of CPU cores and physical disks. Moreover, LLD could perform RAID monitoring and could even count the number of users who are using mail services. However, these features are not provided out of the box. In order to implement them network administrators have to manually configure a prototype or download them from the community. Of course, this could be achieved only if the

open-source tool is in use. (Foxnet blog 2013.) Overall, low-level discovery helps to optimize network monitoring and to provide alerting and graphical output of port status, logical disks and SNMP tables.

3.5 Trend Prediction

Some network monitoring tools have a feature called trend prediction. It is used in order to detect a failure before it even occurred. This is done by collecting data about network bandwidth and the status of devices under normal workload. All the information is stored in the SQL database. Further monitoring results are compared to the information that is stored in the database. If some changes between data have been found, network monitoring creates an alert.

As a result, trend prediction allows detecting the problem beforehand, so that network administrators would solve it before end users even notice it. Despite the fact that it brings considerable advantages for system managers and moves network monitoring to a new level, most of the products still do not support this feature. (Helpsystems 2014.)

3.6 Logical grouping

In large networks that consist of many devices it is rather hard to monitor and troubleshoot all the devices during dynamic network monitoring. Logical grouping allows to combine the same kind of devices. As a result, logical grouping makes monitoring of enterprise-level networks significantly easier.

Logical grouping allows to combine the same type of network devices into groups. For each group could be defined what should be monitored and which actions should be done in case of failure. In addition, with a use of logical grouping it is possible to configure unified settings for all members of the group. If one or more members of the group is down or offline, an alert is displayed.

It is possible to create nested groups for large networks. This means that groups could be created inside another group. As a result, the management of network devices inside a big network becomes easier with logical grouping

Logical grouping is also useful for documenting network status. It is possible to create separate documentation for each group. That is why, in case some information about some exact time in the past is needed, it could be got faster and easier due to logical grouping helped with structuring all the data.

To summarize, logical grouping provides additional help for system administrators in monitoring devices' status. Almost all current network-monitoring tools provide logical grouping as an option.

3.7 Conclusion

As it is illustrated in Table 1, there is a list of the most relevant features of network monitoring that have been mentioned above. Each of them has its determination and may even be indirectly related to network monitoring.

TABLE 1, Relevant Features of network monitoring

Feature	Description
License	GPL (Open source code)
Monitoring	Agent-based with Agentless
Network and Port Discovery	Low-Level Discovery with Auto-Discovery
Structures information	Logical Grouping
Prediction of upcoming issues	Trend Prediction

4. ZABBIX

In this chapter I describe the Zabbix network monitoring tool with general facts about Zabbix, its main features and working principles. In addition, I describe the key advantages that distinguish Zabbix from other monitoring tools.

4.1 Zabbix Overview

Zabbix is a network-monitoring tool that performs centralized monitoring of the availability and performance of the network and network devices. If a failure occurs, an alert will notify a network administrator via phone or mail. Zabbix is totally free network monitoring tool. It is

released under the GPLv2 license. There are no limitations in capabilities and number of monitored devices. It is officially allowed to make modifications on the source code level. In addition, Zabbix supports any size of network installation: it could be small-sized network or it could be even enterprise-level architecture. Zabbix team regularly releases improvements and updates.

4.1.1 Zabbix history

Zabbix was created in 1998. It was the corporate project by Alexei Vladishev. At that time, he was a system administrator in a bank. He was responsible for managing databases. In order to automate routine work, Mr. Vladishev created a first prototype of Zabbix. It was based on the Perl script. (Vladishev 2004.)

At that time there were only two players on that market: HP Open View and IBM BMC. However, these solutions were too expensive and too hard to maintain and configure. The first publicly available open source network monitoring tool, Nagios, was only released in 1999.

It took three years to release the first public version of Zabbix. This was Zabbix v1.0 alpha 1 in 2001 that was released under General Public License (GPL). In 2004 Zabbix v1 the first Long Term Support (LTS) version was released. (Zabbix 2004) The latest version of Zabbix for the moment of writing this thesis is 2.4. The next LTS version will be Zabbix 3.0 that will be released in May or June 2015. However, details of the upcoming version and release dates are not yet specified.

4.1.2 Zabbix Features

Zabbix provides vast variety of features and possibilities. I would like to mention the most essential ones in the following section. Details related to the working principles are described in Chapter 3.

Zabbix support agent-based and agentless that are used to monitor network devices such as routers, switches and server. Network devices have to support SNMP protocol. Zabbix is able to monitor device availability and its performance. In addition, Zabbix supports VMware monitoring. This is used to monitor virtual machine statistics. Low Level Discovery rules are

used in order to discover virtual machines and hypervisors. In addition, Zabbix could also perform the monitoring of databases and web services. (Zabbix 2014.)

Moreover, in case of a network or device failure Zabbix is supposed to alert the system administrator. Zabbix supports Low Level Discovery of network devices and is able to group them logically. However, Zabbix does not have Trend Prediction. That is why, Zabbix cannot alert in advance about failures that may happen. However, Zabbix team has announced that they are currently working on integrating trend prediction into Zabbix architecture (Zabbix 2014.)

4.1.3 Zabbix architecture

Zabbix consists of the following components: Zabbix Server, Zabbix Proxy, Zabbix Agent and Web Interface. Each has a pre-defined role in monitoring. In this section, I describe them. Figure 1 illustrates a complete architecture that contains all Zabbix Components.

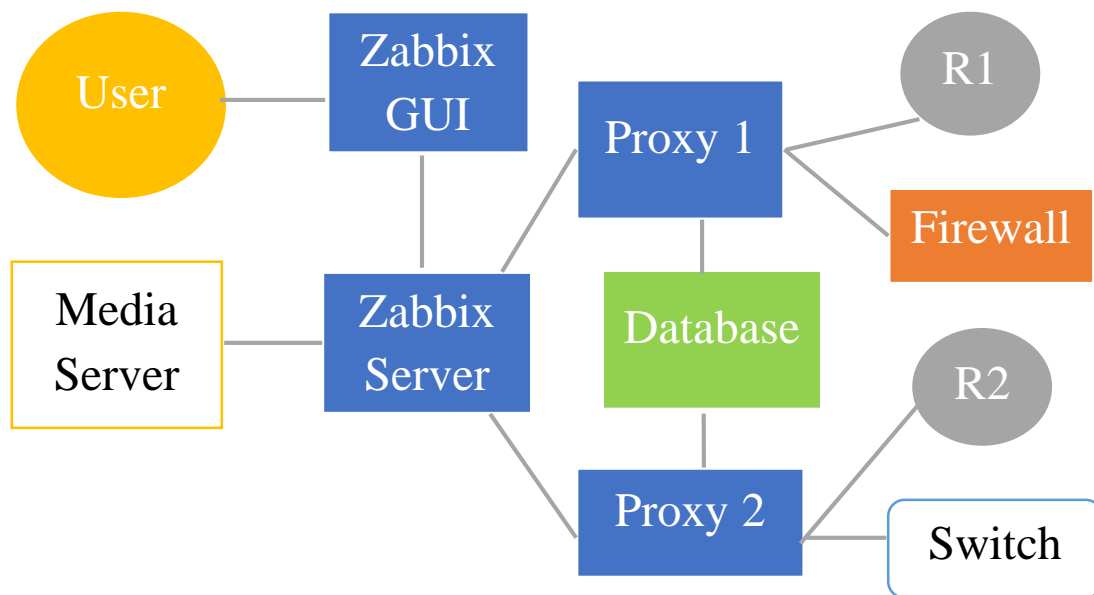


FIGURE 1, Zabbix Components

Zabbix Server is the core part of Zabbix. Its main goal is to perform remote monitoring of the network itself and its components. In addition, it stores configurations, historical and operational data. If an error occurs, Zabbix Server will alert the network manager. (Zabbix 2004.)

Zabbix Proxy collects performance of the data on behalf of the Zabbix Server. On the local level, all the data is collected into the buffer that will be forwarded to Zabbix Server. Proxy is

the solution for centralized remote network monitoring. In addition, Proxy allocates the workload from Zabbix Server. The result is less computational power for Zabbix Server, CPU and memory I/O. (Zabbix 2004.)

Zabbix agent performs local monitoring of the network devices. It monitors resources such as hard drive, memory and CPU statistics. In order to perform resource monitoring, Zabbix agent has to be locally installed on each device. Zabbix agent is highly effective because it makes native system calls. They are aimed at collecting statistical data. (Zabbix 2004.)

Web Interface is a part of Zabbix Server. Usually Web Interface is launched on the same physical server where Zabbix Server is running. Web Interface is not the traditional front-end. All reading and writing operations are directed to the database, bypassing Zabbix Server. This significantly improves Zabbix performance. On the other hand, Zabbix Server is not functional without Web Interface. (Zabbix 2004.)

In Figure 1 there are two additional non Zabbix components illustrated. However, they are playing an essential role in network monitoring. They are Media Server and Database. Media Server is responsible for sending alert notification via Email and SMS while Database is used to store configurational and historical data.

Overall, the combination of these components allows Zabbix to support the three types of monitoring: simple check, Zabbix agent and external check. Simple check verifies the availability of different services such as SMTP or HTTP without additional installations on the host. Zabbix Agent is locally monitoring hardware workload. External check performs remote monitoring through SNMP, TCP and ICMP over IPMI, SSH and Telnet.

I would like to point out that monitoring of the host could be also performed without proxy. In that case all the monitoring data from the host would be collected directly by Zabbix Server. In addition, Zabbix GUI, Zabbix Server, Media Server and Database could be installed in one machine. Such method is essential for small and medium networks.

4.2 Advantages of Zabbix

Compared to Table 1 Zabbix satisfies the requirements of reliable network monitoring tool by 90 percent. It performs both agent-based and agentless monitoring. Such features as Low-

level Discovery, Auto-Discovery and Logical grouping could be found. All the above-mentioned features make Zabbix a foolproof network monitoring tool that completely satisfies the requirements of any size network. However, Zabbix does not support trend prediction. This feature was not included by the Zabbix team because it reduces the overall performance.

Zabbix is a reliable and predictable network monitoring tool. If Zabbix alerts the user about some failures, he/she can be 100 percent confident that such a problem exists. The same principles of reliability are applied to recovery and visualization. (Vladishev 2014.) In addition, one of the main advantages of Zabbix is its scalability as it is applicability for environments of any size. The principal of scalability is applied to performance and usability of the front-end.

However, the capabilities of Zabbix are not limited to IT only. Zabbix as a network monitoring tool could be compared to the brain. It receives the flowing information: input from sensors, data integers, and streaming slope files. Triggers analyze all that data. When an output from triggers is created, the results could be different. It could be mac address of a device, CPU temperature or even an alert or command to launch an automotive script. (Vladishev 2014.)

4.3 Things to improve in Zabbix

Despite the fact that Zabbix provides sufficient network monitoring capabilities, there is still room for improvements. At Zabbix Conference 2014 the founder and CEO of Zabbix, Aleksei Vladishev, outlined the main areas of improvement. Five things that are going to be improved by Zabbix team in the nearest future were mentioned.

4.3.1 Web Interface

The current navigation of the front-end is too complicated. Users for whom Zabbix is new may have some trouble with the Web Interface. Some basic operations may be time-consuming even for experienced users.

Too many clicks are required for basic operations. For example, a network administrator would like to create an item and after that a trigger. First of all, an item should be created. Then the user has to go back and select a trigger. If a connection is lost and the user does not

remember the item key, this will create extra complications. When the item and trigger have been created in order to test them, the user should switch to monitoring. The user should keep in mind what the host for the item was. As a result, a simple process is becoming a nightmare for the system administrator. (Vladishev 2014.)

Another issue is that information is disconnected. For example, an item and its configurations are located in one place and information about monitoring of the latest data could be found in another place. If users would like to see a graph of an item they have to go to another place.

That is why the Zabbix team is currently focused on improving usability. The first thing that they would like to implement is object-centric navigation. It will be based on the principles of one-click away. For example, a user is selecting a host and in one click that person is able to move to the configuration of that host or to reports. In addition, all the information will be interconnected. (Vladishev 2014.)

4.3.2 API

API can be extremely slow, especially when it comes to operations related to template linking. For example, there are 10'000 hosts and a network manager would like to link them into a simple template. It will take approximately 10-20 minutes, depending on the hardware. In addition, it will create too many SQL queries. The number of them could even reach millions.

Another issue is that there is no strict validation and weak error reporting. For example, the user has made a mistake during typing the API call, which has turned into generic error. As a result, the user does not know exactly which error occurred to API.

The Zabbix team is planning to increase the performance of API 10-100x faster. They are planning to reach this result by using algorithms that are more efficient and bulk operations. In addition, the Zabbix team is going to make API "1st class citizen". Currently API is on the front-end part. It is planned to move API to Zabbix server side. As a result, it will bring considerable improvements in performance. (Vladishev 2014.)

At Zabbix Conference 2014 it was announced that there are some issues related to the implementation of the strict validation, error reporting and composability. However, nothing was mentioned about how these issues are going to be solved.

4.3.3 Reporting

Another issue that was announced is the problem with reporting. Information that is collected by Zabbix contains enormous value. It could be events, history data and information about failures. However, there is a need for making improvements for how that information is represented.

Currently Zabbix has limited reporting capabilities as well as no analytics. Users could not create ad-hoc reports. Moreover, Zabbix does not memorize parameters of executed reports. Even in the latest version of Zabbix 2.4 the user has to select the host group and to execute report. (Vladishev 2014.)

4.3.4 Scalability

At the beginning Zabbix works fast and reliably. However, after 1-6 months Zabbix performance dramatically decreases. Zabbix becomes slower as data volume grows.

When network administrator runs `vmstat` it could be seen that I/O is await 50 percent or even up to 100 percent. The reason is that data and indexes do not fit the memory. Tables consist of hundreds kilobytes of history and trends. As a result, the overall performance is decreasing.

In order to scale SQL databases special techniques have to be used, for example database partitioning. However, it is not a part of the Zabbix application. That is why the user has to use some external scripts which may require additional steps. It should be taken into account that not all users are confident to run database partitioning. It is also rather challenging to deliver high availability and redundancy for massive amount of information. In case of terabytes of data, some special techniques should be in use.

That is why horizontal scalability of MySQL is going to be implemented in the next version of Zabbix. In addition, historical and configurational data is going to be stored separately from each other. As a result, it improves Zabbix performance and removes a single point of failure where all the data has been stored. (Vladishev 2014.)

In addition, the Zabbix team is focused on improving front-end performance. Zabbix can store terabytes of data. However, the challenge is to retrieve that data. A good example could be report generating. The Zabbix team is planning to reduce the required time for processing the stored data. (Vladishev 2014.)

4.3.5 Security

The number one feature requested by the community is encryption. Zabbix does not support encryption and authentication out of the box. As a result, users have to implement third-party tools, for example Stunnel and Open VPN. However, they are not appropriately integrated with Zabbix and they are challenging to maintain, especially for large environments.

Alexei Vladishev has announced that encryption and authentication would be integrated to Zabbix. In addition, he has paid attention to security features. Thus, they have to be easily enabled and maintained. The reason is that encryption is a challenging process due to the fact that it has to generate certificates and keys. That is why the Zabbix team would like to implement security features out the box that could be easily implemented for any kind of network. (Vladishev 2014.)

However, the Zabbix team is still not sure about using public encryption keys such as SSL or TLS for agents. The reason for this is that it may bring unwanted footprint on the network. Agent communication is already quite extensive for network performance. If handshaking and encryption related things are added, it will negatively affect the network and the Zabbix server performance. (Vladishev 2014.)

4.3.6 Conclusion

At Zabbix Conference 2014, five main issues to be improved were named. This section provides an overview of how Zabbix will be developed in the nearest future. However, the Zabbix team is not making any specific promises about when and how these improvements are going to be implemented. The reason is that the Zabbix team would like naturally integrate them into existing infrastructure.

4.4 Zabbix on a code level

The first prototype of Zabbix was released as a corporate bank project. It was created on PERL script. However, in order to be the successful commercial product the whole architecture had to be changed.

As a result, the following structure has been chosen. C language was used for all the critical parts such as server side, agent side and later proxy side. For the front-end PHP was chosen. It was used for the visualization and Web Interface. For the database SQL was chosen. (Zabbix 2004.)

This structure appeared in Zabbix 1.0 alpha 1 in 2001. It became a baseline for all the next following versions. Even in the latest version of Zabbix 2.4 this structure is used.

4.4.1 C language in Zabbix

C language is a low-level language. With good programming skills it is possible to create an efficient code. That is why Zabbix is fast and does not require a lot of computational resources. In addition, C language allows creating an application without any dependencies. However, during application development on C language the following features should be taken into account: memory management, logs and shared resources. As a result, this slows down the speed of development.

4.4.2 PHP in Zabbix

PHP is a high-level language. Its main advantage is that it is available for all platforms. In contrast, PHP brings some drawbacks. It is a dynamically typed language. It is common when a variable is defined as an array. However, the next line could be an integer and the next line could turn into an object. As a result, this creates all sorts of problems. In addition, due to the fact that PHP is the interpreted language, an error tends to come up during runtime. In other words, there is no compilation where it is possible to test the application.

4.4.3 SQL in Zabbix

When the first public version was in the process of creation, the choice was between SQL and Round Robin. The biggest disadvantage of Round Robin is that it aggregates information in the database, and there is a need to set up a rule for aggregation in advance. When requirements are changed, there is no more access to original data. That is why SQL was chosen for Zabbix.

SQL is a transactional storage engine. This means that the massive amount of changes could be done, and still the structure will be atomic. As a result, this provides consistency in terms of constraints on database level. The engine itself verifies when data is inconsistent. Moreover, SQL has the API standard. On Zabbix it is possible to run MySQL, PostgreSQL, Oracle, DB2 and SQLite.

On the other hand, it is rather challenging to scale traditional SQL databases. Scaling reading operations is relatively easy. To some extensions, there is a need to make modifications on the application level. However, it is not so easy for extensive operations.

4.4.4 Advantages of Zabbix architecture

The combination of C language, PHP and SQL positively affects Zabbix. As a result, Zabbix is quite small application with almost no dependencies. The usage of C language is the key component of Zabbix performance. In addition, this network monitoring tool requires low resource usage.

The Zabbix architecture provides the separation of functions. For example, data collection is separated from other components like host, items and agents. As a result, it is possible to collect data without affecting other components. Moreover, Zabbix is a multi-process application. Components such as Zabbix Server, Zabbix Proxy and Zabbix Agent scale properly to the number of cores. If the customer's hardware has 32 or even 128 cores, Zabbix will scale it in order to get the full advantage of the customer's hardware.

4.4.5 The challenges of Zabbix Architecture

Despite the fact that the Zabbix architecture provides considerable advantages on overall performance, there are still some drawbacks that have to be mentioned. Two different technologies for the front-end and back-end are used in Zabbix. While PHP is used in front-end, there is C language on the back-end. The result is extra challenges and effects on the speed of development. Sometimes there are situations when the Zabbix team has to create duplicates of the code both on C and on PHP. In addition, code duplication partly affects regressions.

Another issue of the Zabbix architecture is that a front-end PHP is used, which is a dynamically typed and interpreted language. As a result, it causes some additional problems, for example, regressions and unfortunate issues. Theoretically, if PHP is replaced with another language, the whole class of problems will disappear.

4.5 Memory Interaction

Zabbix uses some special techniques in order to get better performance on the memory level. In this section I describe how Zabbix is able to retrieve data bypassing the database. In addition, how several inserts are combined into one bulk operation will be introduced.

4.5.1 Cache

Starting from the second version Zabbix uses the caching technique. Cache is the layer between Zabbix Server or Zabbix Proxy and the database. A good example could be a configuration cache. As it is shown in Figure 2, in order to retrieve data from database, calls are not directed to the database. Configuration data is taken from the cache. (Vladishev 2014.)

In addition, Zabbix has a value cache. It has been introduced in the Zabbix version 2.2. It significantly improves the performance of the trigger calculations. In order to retrieve data for the evaluation of triggers, it is taken directly from the memory, instead of making a direct call to the database.

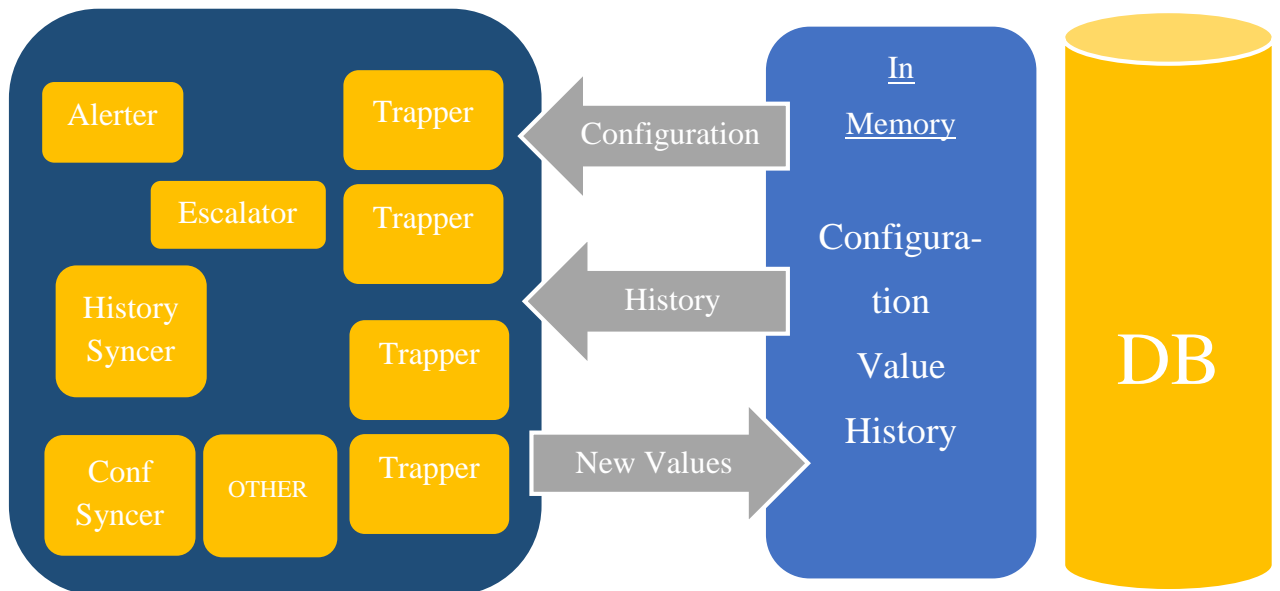


FIGURE 2. Cache Techniques

4.5.2 Bulk operations

Bulk operations is another feature that improves Zabbix performance is history write cache. As it is shown in Figure 3, instead of doing multiple inserts and updates to the database, they are combined as the one bulk operation. However, as caching technique, bulk operations are used only at the back-end. That is why it is only implemented on Zabbix Server and Zabbix proxy. (Vladishev 2014.)

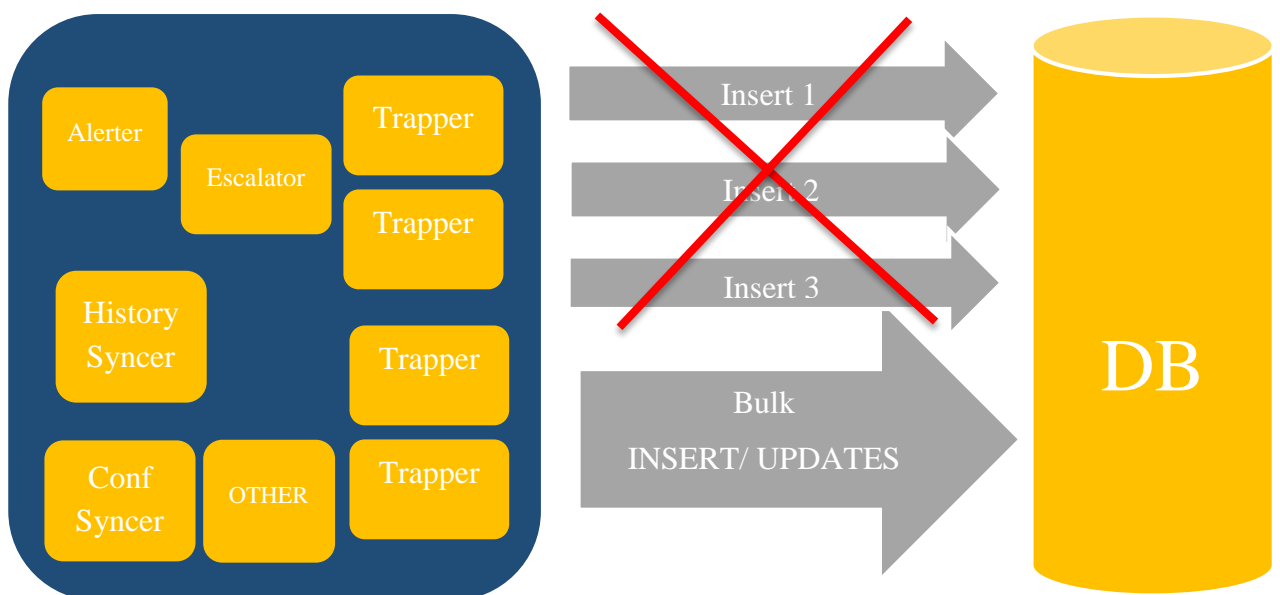


FIGURE 3. Bulk Operations

4.6 Zabbix SIA revenue model

As it was mentioned before, Zabbix is a completely free network monitoring tool with no limitation of devices and capabilities. In this section I analyze how Zabbix SIA is able to make profit.

4.6.1 Professional Training Program

Zabbix professional training program is basically for network administrators for whom Zabbix is completely new. Courses focus on learning Zabbix features and their way of implementation for the company's own network. There are two types of courses: Zabbix Certified Specialist and Zabbix Certified Professional. (Zabbix 2014.)

Zabbix Certified Specialist is aimed at system administrators who are new to Zabbix. The course is focused on extending the knowledge of using Zabbix above standard configurations. Most of the time is spent on solving a real case with Zabbix features. The course length is three days and the price is EUR 1 450. (Zabbix 2014.)

Zabbix Certified Professional is aimed at system administrators who are managing big-sized networks. The course is focused on adapting Zabbix for complicated environments. The course length is two days and the price is EUR 1 250. (Zabbix 2014.)

Courses are scheduled all around the year. They are arranged in different countries and in various languages. It is also possible to make a training location request in order to help the Zabbix team to understand where a better place for the upcoming sessions would be. (Zabbix 2014.)

In addition, it is possible to have training within the company. Zabbix specialist is invited to give a course directly inside companies' offices. The advantage of on-site training is that the course is focused on specific network requirements. In addition, it improves all the IT staff skills related to network monitoring. (Zabbix 2014.)

4.6.2 Technical Support

As networks are different from each other, there is no single solution that perfectly fits for any network requirements out of the box. That is why it is possible to have Zabbix Technical Support. Its aim is to provide Zabbix integration inside the network. In addition, it is possible to order the creation and integration of extra functionalities. (Zabbix Services 2014.)

There are five levels of Technical Support: Bronze, Silver, Gold, Platinum and Enterprise. They are different by limitations regarding to the number of incidents possible to solve. In addition, guaranteed support period also varies. Remote troubleshooting and distributed monitoring with Zabbix Proxy are available with the Gold offer. (Zabbix Services 2014.)

The official Zabbix webpage provides no information regarding the prices for the Technical Support. The cost depends on the level of support. In addition, it can also vary depending on the number of Zabbix Servers and Proxies.

4.6.3 Consulting

Consulting services are aimed at improving and on tight adjustments of Zabbix to the specific network architecture. Consultancy offers the Zabbix migration to the latest version. This service is useful for enterprise-level networks. Consultancy services could be also applied for customer's special case. (Zabbix Services 2014.)

There are several advantages that Zabbix Consultancy is bringing. It is tightly integrated into the enterprise environment in order to get full benefits and network overview. In addition, consultancy is able to solve some issues at short notice. Consultancy could be done in three ways: phone or email, remotely or on-site. (Zabbix Services 2014.)

There are two types of consultancy: standard and premium. Standard consultancy is done remotely on appointment. The price per hour is EUR 200. The minimum order is 15 minutes for EUR 50. Premium is aimed at complicated issues; only a specially named Zabbix consultant would be responsible. The price per hour is EUR 300. The minimum billing for 15 minutes is EUR 75. (Zabbix Services 2014.)

For enterprise-level network there is a possibility to have Prepaid Package. It offers fixed amount of hours for consultancy. The minimum package for 10 hours costs EUR 1 950. The maximum package for 80 hours has a price of EUR 13 200. It is also possible to have on-site consultancy. The price per day is EUR 1 450. However, travel expenses are not included in the price. (Zabbix Services 2014.)

Even though Consultancy and Technical Support may look similar, they have different aims. Consultancy provides advice related to optimizing network monitoring, while Technical Support is focused on troubleshooting issues related to Zabbix.

4.6.4 Turn-Key Solution

Turn-Key Solution allows outsourcing the installation and configuration process of Zabbix. It is aimed at network administrators who concentrate on more prioritized tasks or who are not confident with Zabbix. A certified specialist will not only install Zabbix, but will also configure it for network requirements. In addition, during the installation process network administrator can get advice related to the use of Zabbix. (Zabbix Services 2014.)

There are six solution types of Turn-Key. Zabbix offers Light, Basic, Advanced, Advanced+, Professional and Custom page. Each packet has different solution content and differs by amount of required time and price (Zabbix Services 2014.)

Light and Basic offer a remote Zabbix server installation and basic set ups. The difference between them is that Light has a duration of one day and price of EUR 950, while Basic takes two days and the price is EUR 1 800. (Zabbix Services 2014.)

Advanced and Advanced+ offer remote consultancy as well as Zabbix installation and advanced configuration. Advanced solution will take five days and price of EUR 4 500. Advanced+, in contrast, will last for ten days with the price of EUR 8 500. (Zabbix Services 2014.)

Professional solution has a duration of ten days and a price of EUR 16 500. During the first seven days on-site consultancy, Zabbix installation and advanced configuration will be done. During the last three days Zabbix Certified Specialist will provide on-site training. (Zabbix Services 2014.)

Custom package is based only on customers' needs and requirements. The number of days and price varies depending on the project. In custom package different services such as Technical Support, Template building a Consultancy could be added.

The only requirement for Turn-Key solution is to have virtually or physically installed Zabbix server and the ability to connect it via SSH and HTTPS. In addition, network administrators should define which devices and parameters are expected to be monitored.

4.6.5 Template building

Even though Zabbix offers a vast variety of templates for different needs, it is almost impossible to create one product out of the box which suits perfectly for every network. Of course it is possible to download a template from third-party developers. However, the template for some special requirements may not exist. In order to create them, it is possible to get Zabbix Template Building Services. (Zabbix Services 2014.)

Zabbix developers will create the required template which will be adapted by the network environment and devices. With a created template customers would also get advice on using all of its capabilities. In addition, customers are helping to develop Zabbix. Templates may be added and supported by the next versions of Zabbix. (Zabbix Services 2014.)

The price for one template is EUR 650 per day of design, creation and testing. Duration for creation of one template is 1-3 days. The creation of two templates will take 4-6 days. The price per day is EUR 600. Creation of three or more templates requires 7-9 days. The average price per day is EUR 550. (Zabbix Services 2014.)

There are some requirements before the template creation. The device model should be specified. There is a possibility to create a template only for SNMP devices. In addition, it should be specified what kind of data is required to be monitored. It may require some access to the device or service for the Zabbix team in order to test a template.

4.6.6 Conclusion

To sum up, I would like to point out that even though Zabbix is completely free as a product, services offered by it require extra charge. They are focused on improving Zabbix performance for customers' network requirements. As a result, this increases network clarity and availability. Overall, services offered by Zabbix save customers' time and budget.

4.7 Zabbix 2.4

The latest version of Zabbix 2.4 has been released in September 2014. It brings more than 50 new features and updates. In this section I will introduce the most essential changes which the new version has brought.

4.7.1 Optional SNMP bulk

In Zabbix version 2.2 there has been introduced support of SNMP bulk requests. It brings considerable advantage for enterprise-level network that contains thousands of ports. Instead of making connection for each port, only one connection is needed. Afterwards it is possible to retrieve information using bulk request. However, not all the devices support bulk request. As a result, starting from Zabbix 2.4 SNMP bulk becomes an optional feature. (Zabbix 2014.)

4.7.2 Flexible filter for Low-Level Discovery

For Low-Level Discovery (LLD) it is possible to use multiple filters. For example, network devices could be filtered by interface name, interface speed and interface state. It is possible to specify multiple simple filters and combine them into one complex expression. That is why if there is a need to discover a complex application or process, it is possible to filter it by all attributes which are received by the LLD rule. As a result, extra flexibility is provided. (Zabbix 2014.)

4.7.3 Runtime control of log level

Zabbix version 2.4 enables controlling the debug level of running system. In other words, it is possible to increase or decrease the debug level when Zabbix Server is running. This feature is especially useful, when the support team is connecting to customers' systems in order to

gather some debugging information. In Zabbix 2.0 or 2.2 it was possible to increase the debug level only after restarting Zabbix Server. On Zabbix 2.4 it is possible to increase or decrease debug level during runtime. (Zabbix 2014.)

In addition, it is possible to increase or decrease debug level for a class of processes. As a result, system managers may choose an option where the debug level is changed for all poolers, or all trappers. Moreover, there is a possibility to make changes for individual trapper or pooler. (Zabbix 2014.)

4.7.4 Node-based Distributed Monitoring is removed

Node-based Distributed Monitoring (DM) was used for monitoring the number of Zabbix servers and nodes that are interconnected with each other. However, this setup did not guarantee the consistency of configuration data due to the master-to-master replication of configuration data. If there is a network split, significant changes are done in one node and another node. When communication is back, some conflict may appear, and it was not possible to resolve them in the automatic way. (Zabbix 2014.)

Distributed Monitoring had a reliable performance. It could support up to 100 hosts, depending on number of items and number of checks. However, DM did not scale well enough. The reason is that calculation of configuration changes required a lot of CPU power.

As a result, the Zabbix team decided to remove Node based Distributed Monitoring from Zabbix 2.4. It is recommended to use proxy-based monitoring. If existing distributed monitoring is migrating to Zabbix 2.4 it will be automatically converted to a stand-alone set up. (Zabbix 2014.)

4.7.5 Minor changes and improvements

Zabbix 2.4 has also presented minor changes and updates. However, before upgrading to the latest version, it is crucial to understand which changes are brought within an updated version.

Zabbix can automatically perform the discovery of CPUs. It lists the number of CPUs and their status: online, offline or unknown. An unknown status is shown when a CPU has been discovered, but the status could not be verified. (Zabbix 2014.)

An additional option for XML import has been added. It is called “remove if it’s not an XML”. Suppose there are 100 hosts in XML. When this option is selected, all the hosts would be transferred to the Zabbix configuration. However, hosts that are not in XML will be removed. This option is useful for backups. Before making any changes, it is possible to export host-related configuration to XML. After that some changes could be done. (Zabbix 2014.)

New macros were introduced. User macros `{$MACRO}` could be used for notification and commands. This simple change would be useful for many environments. In addition, a description for host, templates and proxies `{HOST/PROXY.DESCRPTION}` was introduced. The description could be used as macro notification for mapping. Host description could be displayed as a label of the map element. (Zabbix 2014.)

4.7.6 Upgrading to Zabbix 2.4

Before upgrading to the Zabbix version 2.4, network administrators should take into account several facts. There will be limited time support for Zabbix 2.4, as it is not a long-term support (LTS) release. As a result, it will be supported until the next major release plus one month. That is why systems that are running Zabbix 2.4 will be forced to migrate to Zabbix 3.0 in May or June 2015. For installations that require system stability and prefer not to update often enough, the Zabbix team recommends to use Zabbix 2.2. It is an LTS version which will be supported for the next 3 years. (Zabbix 2014.)

Estimated upgrade time depends on the size of configuration data. That is why upgrade time numbers vary from 30 seconds up to 5 minutes. The reason why upgrading will take considerably low amount of time is because historical tables are not affected. In addition, no changes must be done in the database. (Zabbix 2014.)

However, node-based distributed monitoring will be automatically converted to a standalone setup. This process may require extra time, due to the fact that the IDs of historical data should be converted to a newer format. This process requires a lot of computational power. (Zabbix 2014.)

4.8 Future of Zabbix

An original plan of Zabbix was released two years ago. It mentioned that Zabbix 2.4 would be released in May 2014. Next version of Zabbix would be 2.6, which will be released in November 2014. Zabbix 3.0 LST was planned to release in May 2015. (Zabbix 2014.)

However, the release of Zabbix version 2.4 was delayed till September 2014. Zabbix team have presented an adjusted plan, where they have decided to skip version 2.6. That is why the next major release would be Zabbix 3.0 LST in May or June 2015.

At the moment of writing my thesis, on Zabbix official web site it was not mentioned about Zabbix 3.0 release dates. Moreover, there was not any specified information about features and changes in upcoming version.

5 PREPARATIONS

From my practical part, I am expecting to test Zabbix on the real environment. I am aiming to test its monitoring capabilities. In order to be able to do so, there should be done some preparations beforehand.

5.1 Scenario

During practical part, I will monitor routers interfaces as well as memory and CPU usage Zabbix Server. The monitoring will be done by Simple Network Management Protocol (SNMP). I will use SNMP v2 as a standard due to the fact that in testing environment there is no need in security features that SNMP v3 is bringing.

SNMP is a standard internet-protocol that is used for managing devices in IP networks. In the range of devices that supports SNMP includes routers, switches, server, printers and PCs. There are 3 generations of SNMP protocols.

SNMP v1 is 1st generation managements protocol. Version 1 was widely criticized due to lack of security. Authentication was done by community string. Community string is kind of a password, in order to get an access to the router.

SNMP v2 brings improvements in performance, security and confidentiality. GetBulkRequest was implemented in version 2 instead of GetNextRequest. Bulk request allows to get a large number of control data in one request. However, new security system did not get spread wide due to its complexity.

SNMP v3 is the latest version of SNMP protocol. Version 3 brings radical security improvements in remote configuring. Each SNMP version3 messages are getting encrypted. In addition, key concepts of security have been implements such as confidentiality, consistency and authentication.

Internet Engineering Task Force (IETF) concluded that SNMP v3 has the highest maturity level for an RFC. Previous versions are considered historical. However, Zabbix supports all three generations of SNMP protocols.

In addition, during practical part, some additional steps would be done in order to test Zabbix monitoring and alerting capabilities:

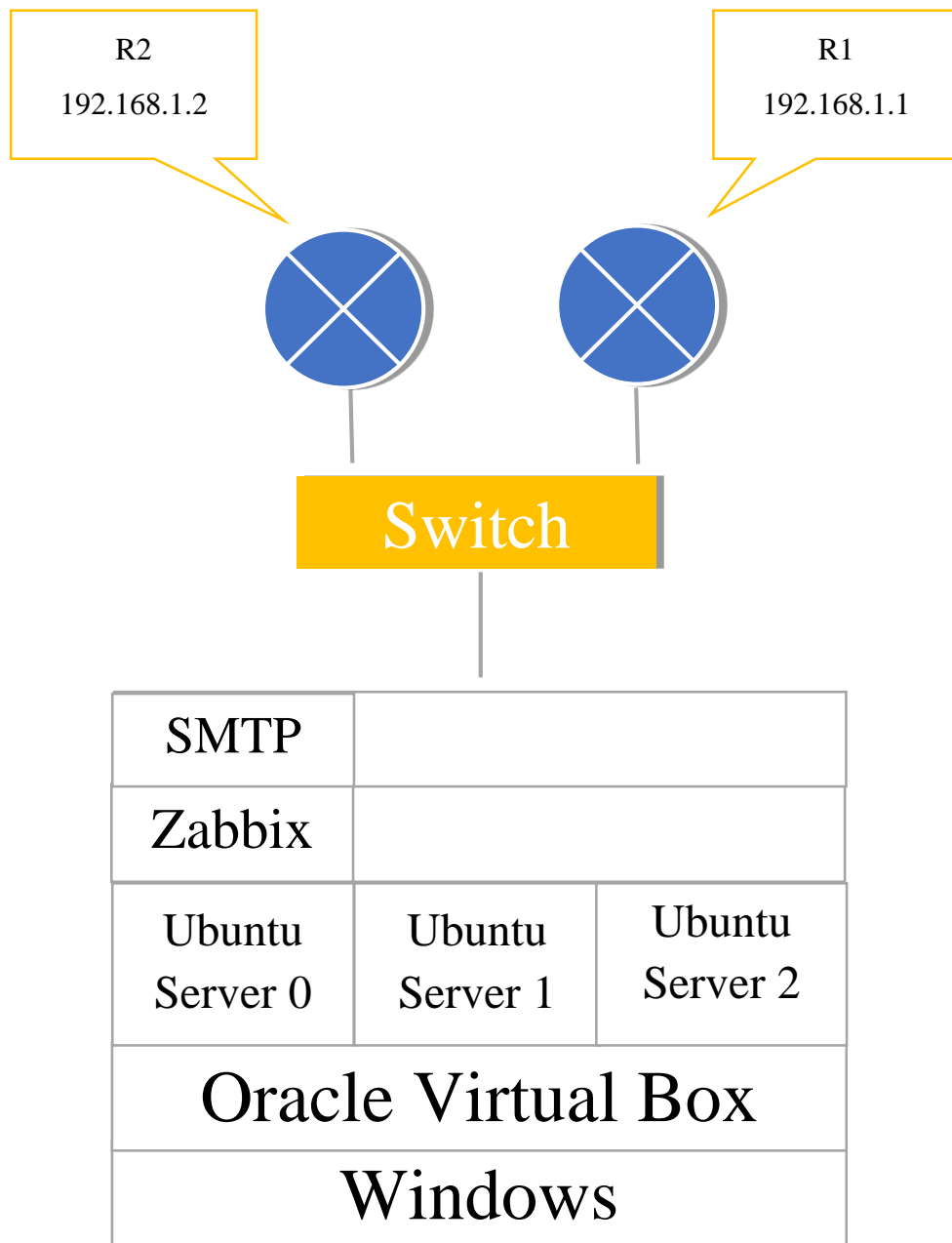
- Interfaces of Linux Server will be monitored by SNMP.
- Zabbix Agent will be locally installed on Windows Server
- The graph of bandwidth usage, RAM and CPU will be created.

5.2 Network Design

First of all, the small-sized network should be created. In Figure 4 it is shown the logical design of the network. This installation consists of PC1, Zabbix Server, one Cisco Catalyst Switch and two Cisco routers. Moreover, two additional virtual machines are going to be the part of the network: Ubuntu Server and Windows Server.

Zabbix Server would be installed on PC1 as the Virtual machine. In addition, Ubuntu Server and Windows servers would be also installed as virtual machines. The connection between PC1 and switch would be done with a straight-through cable. Switch would be interconnected with R1 and R2 also with a straight-through cable.

The last thing left is to activate SNMP server on router 1 and router 2. It could be done by typing commands in Cisco CLI: `#snmp-server community senna RO`. I have named my community as senna.



Name of the server	IP Address
Ubuntu Server 0	192.168.1.10
Ubuntu Server 1	192.168.1.15
Ubuntu Server 2	192.168.1.20

FIGURE 4. Network topology

5.3 Capacity Planning

Before installing Zabbix Server and Ubuntu Server, the capacity planning should be done in advance. The minimum requirements depend on a size of the network and the amount of

monitored devices. After some period of time, the amount of stored data would dramatically increase. As a result, computational resources and amount of required memory have to be planned in advance.

According to the Zabbix documentation, Zabbix Server requires only 128 MB of RAM and 256MB of HDD. However, the amount of hardware capacity depends on the size of the network. In Table 2 there are listed recommended requirements for the networks of different size.

TABLE 2. Zabbix official requirements

Network Size	Platform	CPU/Memory	Database	Monitored Hosts
Small	Ubuntu Linux	Virtual Appliance	SQLite	100
Medium	Ubuntu Linux 64	2 CPU cores/2GB	MySQL In- noDB	500
Large	Ubuntu Linux 64	4 CPU co- res/8GB	RAID10 MySQL In- noDB or Post- greSQL	>1000
Very Lange	RedHat Enterprise	8 CPU cores/16GB	Fast RAID10 MySQL In- noDB or Post- greSQL	>10000

5.4 Memory Planning

In addition, memory size planning should be done in advance. In Table 3 there are some formulas described in order to calculate required disk space for the history, trends and events. For example, 1800 items are used for monitoring. The refresh rate is 60 seconds. The number of new items added to the database is calculated by dividing number of items by seconds. Thus, every minute 30 new values are added.

TABLE 3. Memory planning

Parameter	Disk space required
Zabbix Configurations	10Mb
History	$\text{days} * (\text{items}/3600) * 24 * 3600 * \text{bytes}$
Trends	$\text{days} * (\text{items}/3600) * 24 * 3600 * \text{bytes}$
Events	$\text{days} * \text{events} * 24 * 3600 * \text{bytes}$

5.5.1 History Data Planning

History data is normally stored for the fixed amount of time such as weeks or even several months. For example, there is a need to store history data for 30 day. Every minute 30 new values are received. According to history formula in *Table 3*, $(30 * 24 * 3600) * 30 = 77'760'000$ or 77 Mb of values.

5.5.2 Trends Data Planning

Trend is storing the dynamical changes of history data, thus saves maximum, minimum and average data every hour. The period of one hour could not be customized. Trend is used for creating long period graphs.

One single trend requires 130 bytes, the size depends on database engine. In order to calculate the size of memory required for trends the formula from *Table 3* is used. For 1800 items the formula would be $1800 * (24 * 365) * 128$. As a result, it would require 2 GB of memory in order to store trends data.

5.5.3 Event Data Planning

Each event requires approximately 130 bytes. For example, Zabbix generates event every second. Based on formula from *Table 3*, it is possible to calculate the disk space for one year. Size of one event 130 bytes should be multiplied by $365 * 24 * 3600$. As a result, it would require 4 GB of disk space to store events for one year.

5.5.4 Conclusion

In this chapter, I have described techniques that help to plan memory capacity in advance. For Zabbix Server I decided to have 20 GB of disks space. Such amount of disk space could be enough to store configurational, historical and trend data for several years of active monitoring. I decided to use amount of disk space in order to cut the risks

6 ZABBIX INSTALLATION AND TESTING

In this chapter, I will install and deploy Zabbix Server. I will test Zabbix monitoring capabilities on the network that have been described in Figure 4. For this chapter I have created a roadmap of main steps that are going to be performed during practical part. It is illustrated in Figure 5. Each presented step would have sub-steps.

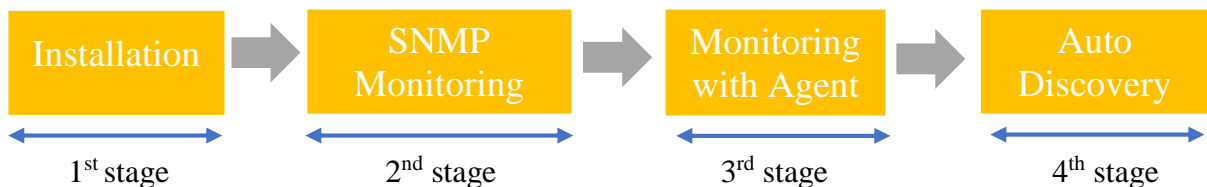


FIGURE 5. Project roadmap

I will start with Zabbix Server installation. Then I will perform SNMP Monitoring of the host. After that, I will complete monitoring with Zabbix Agent on Ubuntu Server 2. In the last step, I will implement auto discovery. However, for the last part, there is a need to make modification in the network topology that is presented in Figure 4.

6.1 Installation

In this section, I will describe the process of Zabbix Server installation. Figure 6 provides a roadmap of installation stages. First there is a need to install Ubuntu Server. After that, on top of it will be installed Zabbix Server. The last stage is going to be login to Zabbix GUI.

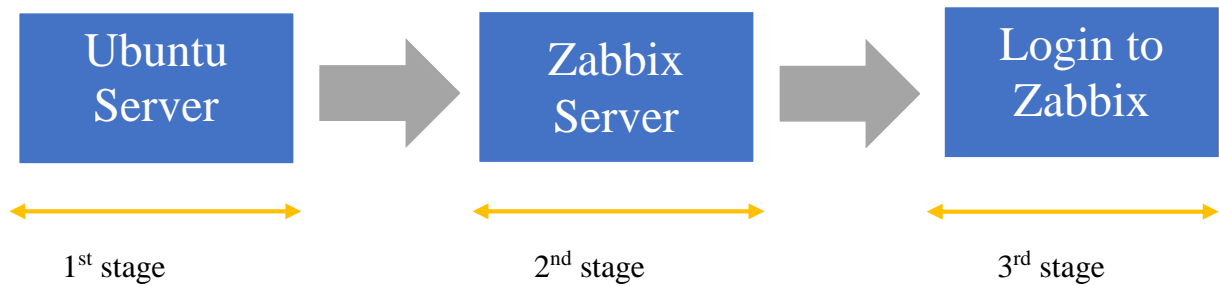


FIGURE 6. Installation roadmap

6.1.1 Linux Server Installation

In this part, I will install Linux Server. I have chosen Ubuntu Server 14.04.2 LTS. It would be installed as a virtual machine in Oracle Virtual box.

During installation process of Ubuntu Server there have to be done some steps that would allow later to install Zabbix Server:

- **No Automatic Updates** due to Zabbix Server requires consistency
- Install **Open SSH Server**. It would allow to connect to Ubuntu Server via Putty

On interface eth1, I have assigned an IP address of Ubuntu Server. The IP address was chosen 192.168.1.10. In addition, this is going to be an address of Zabbix Server. The result could be seen in Figure 7.

```

senna@lotus:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:36:04:0c
          inet addr:172.16.0.208  Bcast:172.16.7.255  Mask:255.255.248.0
          inet6 addr: fe80::a00:27ff:fe36:40c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:110442 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56305 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:147127932 (147.1 MB)  TX bytes:4708451 (4.7 MB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:04:05:fb
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe04:5fb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:594 errors:0 dropped:18 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:148511 (148.5 KB)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:48 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3230 (3.2 KB)  TX bytes:3230 (3.2 KB)
  
```

FIGURE 7. Interfaces of Ubuntu Server

The last step is to install and configure SNMP manager on Ubuntu Server. This would allow Zabbix server to monitor devices by SNMP. In order to be able to do so, the below mentioned list of commands has to be typed in Linux command line.

- *sudo apt-get install libsnmp-mib-compiler-perl*
- *sudo apt-get install snmp-mibs-downloader*
- *sudo apt-get install libsnmp-base*
- *sudo apt-get install libsnmp-dev*
- *sudo apt-get install snmp*
- *sudo apt-get install snmpd*

As a result, the first stage is done. Ubuntu Server with SNMP manager have been installed and configured. The next step is to install and configure Zabbix Server.

6.1.2 Installing Zabbix Server

The second stage is to install Zabbix Server on top of running Linux Server. In order to download and install Zabbix Server below mentioned commands have to be typed in Ubuntu Server's command line. These commands could be found in Zabbix official documentation.

- *sudo wget http://repo.zabbix.com/zabbix/2.4/ubuntu/pool/main/z/zabbix-release/zabbix-release_2.4-1+trusty_all.deb*
- *sudo dpkg -i zabbix-release_2.4-1+trusty_all.deb*
- *sudo apt-get update*

I have installed Zabbix v2.4 package. It includes components such as Zabbix Server, Zabbix front-end (GUI), Zabbix Proxy, Medial Server and Zabbix Database. This is the latest package available at the moment of writing this thesis

Additionally I have installed Zabbix Agent locally on Zabbix Server. It could be done by this command: *sudo apt-get install zabbix-agent zabbix-server-mysql zabbix-frontend-php snmpd php5-mysql php5-curl*. Zabbix Agent would perform local monitoring of processes on Zabbix Server.

Before being able to start to use Zabbix two steps are left. The first one is to configure the time zone according to the region where Zabbix Server is installed. This could be done by this command: `sudo vi /etc/network/interfaces`. In this case, it is *Europe/Helsinki*. The second step is to restart Apache server. This could be done by typing a command: `sudo service apache2 restart`.

As a result, the second step has been done. Zabbix Server has been installed and configured. In addition, Zabbix Agent was installed locally on Zabbix Server. The next step is to log in to Zabbix GUI.

6.1.3 Log in to Zabbix

The last step is to log in to Zabbix front-end (Zabbix GUI). This could be done via browser. In address bar there should be typed an address of own Zabbix Server plus `/zabbix/` part. For this installation the address is `http://192.168.1.10/zabbix/`. After that a welcome banner has appeared.

The next step is checking the test connection and pre-requirement. This is done by Zabbix automatically. The last step is to login to Zabbix. The default username is *Admin* and password *zabbix*. As a result, Zabbix home has appeared. Figure 8 illustrates Zabbix GUI home page.

FIGURE 8. Zabbix main page

6.1.4 Conclusion

The first step of a roadmap presented in Figure 5 is done. I have been able to install and configure Ubuntu Server. On top of it Zabbix Server was installed. I have additionally installed Zabbix Agent locally on Zabbix Server. Finally, I logged in to Zabbix front-end main page.

6.2 Monitoring of Zabbix Server

As it was mentioned in Chapter 6.1.2, Zabbix Agent has been installed locally in Zabbix Server. As a result, in GUI it is possible to monitor Zabbix Server performance. Figure 9 illustrates one of the graphs that has been automatically created. In addition, the process of host, item and trigger installation was done automatically.

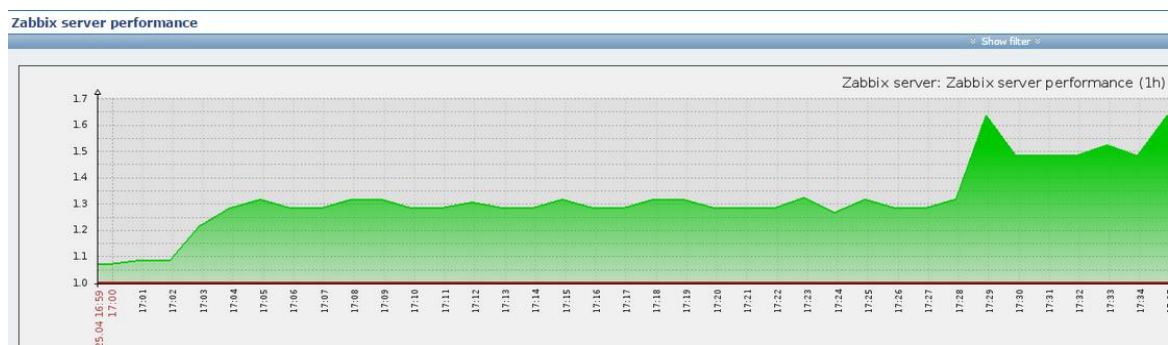


FIGURE 9. Zabbix Server performance

6.3 SNMP Monitoring

In this part, I will perform SNMP monitoring of devices by using Zabbix Server. Figure 10 provides a roadmap of steps that have to be performed before starting monitoring.

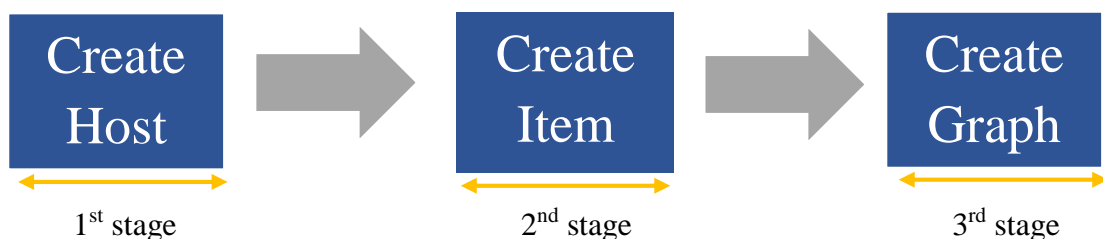


Figure 10. Roadmap of SNMP monitoring

In this chapter, I will create a host, an item and a graph. A host is a network device that would be monitored by Zabbix. An item specifies for Zabbix, what I would like to monitor on a host.

In my case, there will be created several items, in order to monitor CPU usage, memory, interfaces and bandwidth. Graphs will be created in order to provide a graphical output of monitored data.

6.3.1. Host Creation

The first step is to create a host. It is network device or service that is going to be monitored by Zabbix Server. In this chapter I will create a host of Router 1. I will name a host R1. That is why the name R1 is going to be used in context of monitoring and Zabbix environment in general. Router 1 is a name of a network device itself. The name of is going to be used in order to describe it in context of network device.

Figure 11 illustrates the step of creating a new host. I have given named host R1 that represents Router 1. The essential part of host creation is to assign a right interface of a device. R1 is going to be monitored by SNMP, due to the fact that it is impossible to install Zabbix Agent locally on it. In addition, I have to specify the IP address of a device that is going to be created as a host. According to the logical network plan presented in Figure 4, the IP address of Router 1 is *192.168.1.1*

Host name: R1

Visible name:

Groups:

- In groups: Templates
- Other groups: Discovered hosts, Hypervisors, Linux servers, Virtual machines, Zabbix servers

New group:

Agent interfaces	IP address	DNS name	Connect to	Port	Default
SNMP interfaces	192.168.1.1		IP	DNS	161

Use bulk requests:

JMX interfaces: Add

IPMI interfaces: Add

Description:

Monitored by proxy: (no proxy)

Enabled:

Buttons: Update, Clone, Full clone, Delete, Cancel

FIGURE 11. Creating a new host

6.3.2 Item Creation

The second step is to create items for R1. Item collects the data from a host. I will create item that will collect data such as CPU, memory and bandwidth load from host R1 that represents Router 1. In this chapter I will describe in details the process of creation an item for monitoring CPU usage.

For CPU usage monitoring two items have to be created. The first one is CPU idle. It measures the amount of available resources. The second item is CPU usage. It would collect the data from R1 about percent of CPU resources in use.

Figure 12 provides a detailed overview of CPU idle item creation. The monitoring of CPU on router one would be done via SNMPv2. Zabbix also supports SNMPv1 and SNMPv3. The host interface is assigned automatically. The key is created manually and the name of it should be unique.

The next is to assign OID number. It is a unique number that is used to name specific process or parameter. Each process has own unique OID number. Zabbix Server in order to retrieve certain parameter from a device sends OID number of that parameter to that device. It responds to Zabbix already with parameter results. The OID number of CPU idle is *1.3.6.1.4.9.2.1.59.0*. OID numbers could be found in official documentation from manufacturer of a device.

The screenshot displays the Zabbix 'Create item' form. The configuration is as follows:

- Name:** CPUIdle-R1
- Type:** SNMPv2 agent
- Key:** CPUIdleR1
- Host interface:** 192.168.1.1 : 161
- SNMP OID:** 1.3.6.1.4.9.2.1.59.0
- SNMP community:** public
- Port:** (empty)
- Type of information:** Numeric (unsigned)
- Data type:** Decimal
- Units:** (empty)
- Use custom multiplier:** (checked) 1
- Update interval (in sec):** 2
- Flexible intervals:** No flexible intervals defined.
- History storage period (in days):** 90
- Trend storage period (in days):** 365
- Store value:** As is
- Show value:** As is
- Applications:** (empty)
- Populates host inventory field:** -None-
- Description:** (empty text area)
- Enabled:** (checked)

Buttons for 'Add' and 'Cancel' are visible at the bottom.

FIGURE 12. Creating an CPU idle item on R1

I have also created items for CPU load. In addition, memory and bandwidth items were also implemented on R1. Table 4 provides an overview of item names, keys and SNMP OIDs for each parameter. They have been all implemented on a host R1.

Table 4. Item parameters

	ITEM	Key	OID
CPU	CPU Idle	CPUIIdleR1	1.3.6.1.4.1.9.2.1.59.0
	CPU load	CPULoadR1	1.3.6.1.4.1.9.2.1.56.0
Memory	MemoryFree-R1	MemoryFreeR1	1.3.6.1.4.1.9.9.48.1.1.1.6.1
	MemoryUsage-R1	MemoryUsedR1	1.3.6.1.4.1.9.9.48.1.1.1.5.1
Bandwidth	ifInOctets.1-R1	ifInOctets.1R1	1.3.6.1.2.1.2.2.1.10. <u>2</u>
	ifOutOctets.1-R1	ifOutOctets.1R1	1.3.6.1.2.1.2.2.1.16. <u>2</u>

During the process of SNMP OID creation for items Income traffic and Outcome traffic there is no one universal OID. The reason is that each port number has own SMNP OID. Both items have fixed parts. The last digit is a unique OID that varies depending on port.

Command `#show snmp mib ifmib ifindex` for Cisco devices is used in order to retrieve interface index number. Figure 13 provides an output of this command. As my router is connected to the Switch via port `FastInternet0/1`- the index number is 2.

```
Router#show snmp MIB IFMib IFIndex
Jan  2 13:55:48.387: %SYS-5-CONFIG_I: Configured from console by console
Serial0/0/0: Ifindex = 3
Async0/0/1: Ifindex = 8
FastEthernet0/1: Ifindex = 2
VoIP-Null0: Ifindex = 5
Null0: Ifindex = 6
Serial0/0/1: Ifindex = 4
Async0/0/0: Ifindex = 7
FastEthernet0/0: Ifindex = 1
```

FIGURE 13. Port index

6.3.3 Creating Graphs

Zabbix provides graphical output of monitored data. The third step is to create graphs for CPU, memory and bandwidth usage of the Router 1. Figure 14 illustrates the process of the graph creation for CPU usage. In configurations, there could be assigned size and type of the

graph. In addition, the item has to be chosen. For graphical output of CPU performance two items are needed: CPU idle and CPU load. The result can be seen in Figure 15.

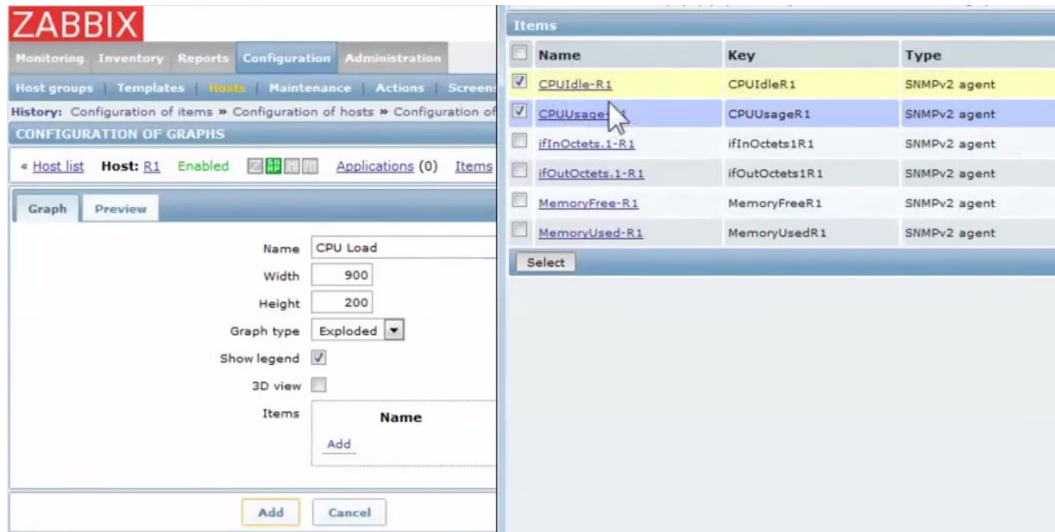


FIGURE 14. Graph Creation

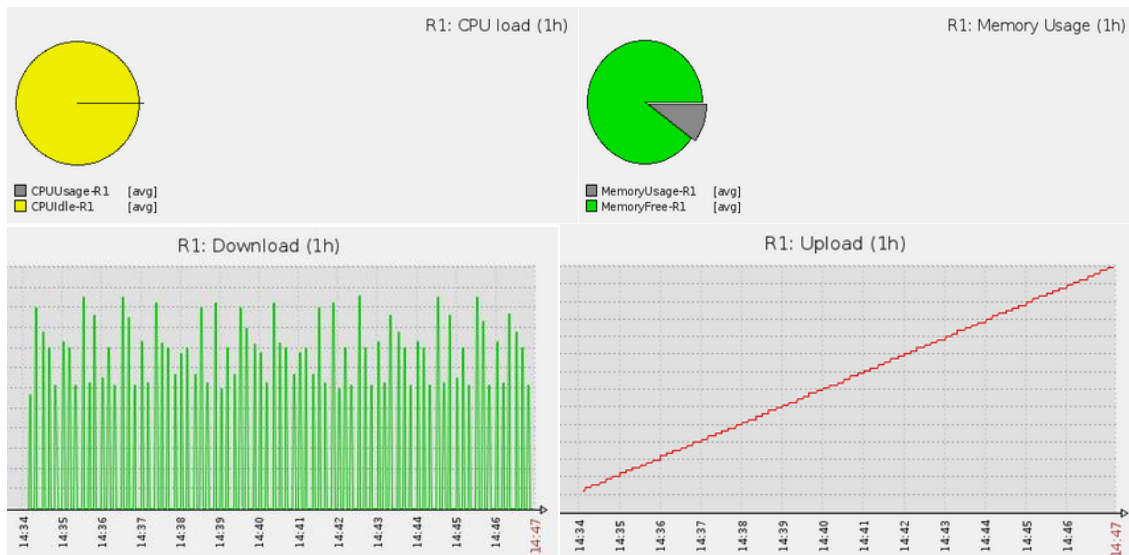


Figure 15. Graphs of R1 performance

6.3.4 Conclusion

I have been able to add a host R1 that is based on Router 1 network device. I have created items, in order to collect specific data from a host. In addition, all the monitored data has a graphical output.

6.4 Monitoring with Templates

In Section 6.3 I have been able to configure monitoring on R1. There I have manually added host created items and graphs. During monitoring no issues have occurred. In addition, the result has been sufficient. Despite the advantages, the configuration of items and graphs is a time consuming process. It should be also taken into account that SNMP OID is unique for different parameters. The same process may have different OID numbers due to different vendors or different type of device. In addition, each interface has a unique OID.

That is why for the manual creation of items it is required a detailed, network documentation. It should at least contain device types, names and statuses. In addition, a network administrator should have a complete list of SNMP OIDs for each device. This may not be a problem for a network containing one 24-port switch. However there is a real issue for a network that has 400 switches, each of them containing 48 ports.

As a result, the whole point of using network monitoring has disappeared. Instead of providing automation and flexibility, a network administrator has to spend huge amount of time only to prepare all the devices for monitoring. That is why templates are used.

A template is a set of entities that can be applied to any host. A template can consist of items, triggers and graphs. In Zabbix by default there is a list of templates. However more templates can be downloaded from official Zabbix web page or even created by a user.

There are two key advantages brought by templates. The first one, as it was mentioned before, templates may contain by default items, trigger and graph. Templates allow to add more items or to edit triggers.

The second advantage is that one template can be used for unlimited number of hosts. That is why assigning of monitoring parameters for hosts is done in one click. In addition, if there is a need to make some changes in monitoring, changes have to be done only on a template. Hosts that are using that template are automatically getting updated. As a result, templates bring flexibility that is expected from network monitoring.

In this section I will configure a template for host R1. A created template will be also used for monitoring processes of the second host- R2 that represents Router 2. In the last step I will

use templates in order to monitor processes on Ubuntu Server 1. Figure 16 provides a roadmap for this section.

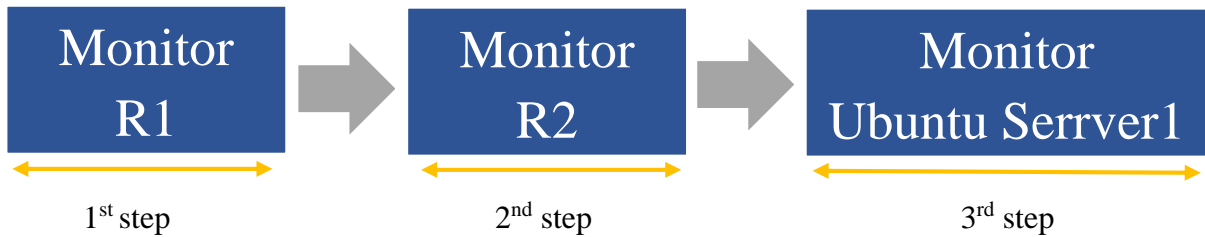


FIGURE 16. Templates roadmap

6.4.1 Router monitoring

In this section I will perform monitoring of R1 using templates. I have chosen Template SNMP Devices. Monitoring of this template is based on SNMP protocol. As I would like to use this template on R1, there is a need to specify SNMP community. For R1 it is *senna*.

Figure 17 illustrates the step when I assigned Template SNMP Devices on R1. As a result, six new items have been automatically created on a R1. In addition, graphs and triggers were also added. However, this template automatically adds discovery and monitoring of interfaces.

I would also like to monitor CPU load and memory usage on R1. There are two ways to do that. The first one is to add these items on Template SNMP Devices. However, this will add limitations to usage of this template for other devices. That is why I will add two more templates on R1.

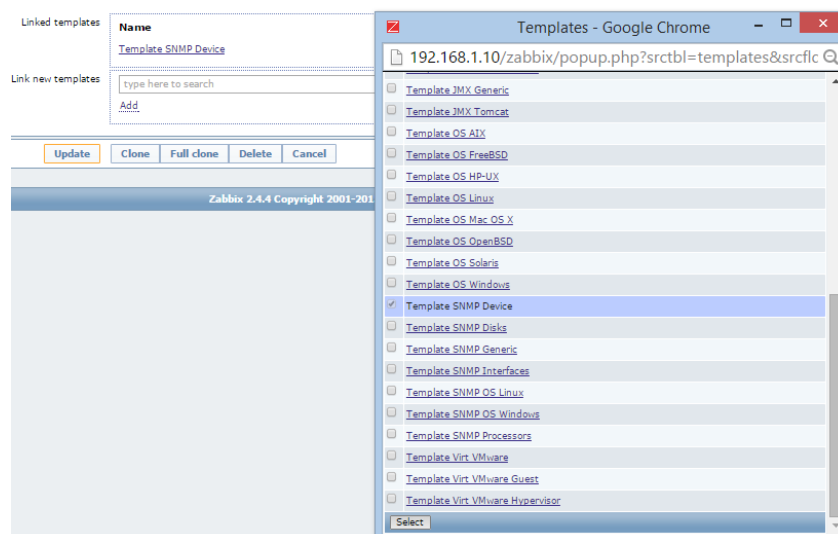


FIGURE 17. Inserting templates

In the list of templates there can be found Template SNMP Processors and Template SNMP Memory. However they do not contain any items. That is why I have to create items for both templates manually.

First of all, I have cloned original templates. Then there is a need to create items the same way as it was done for a host. For CPU performance measurement two items have to be created: CPU Idle and CPU Load. For memory usage items have to be Memory used and Memory free. From Table 4 I have taken SNMP OIDs for required items. It should be taken into account that the name and a key for new created items have to be unique. This should be done in order not correlate with items that have been created in section 6.3.

When new items have been created there is a need to specify SNMP community. This could be done manually in macros tab. There I have to specify that `{SNMP_COMMUNITY}` string is equal to value *senna*.

Based on new item graphs can be created. After that a new template could be assigned on R1. As a result, R1 is monitored by templates. Figure 18 illustrates the graphical output of monitored data on interface *FastEthernet 0/1*.

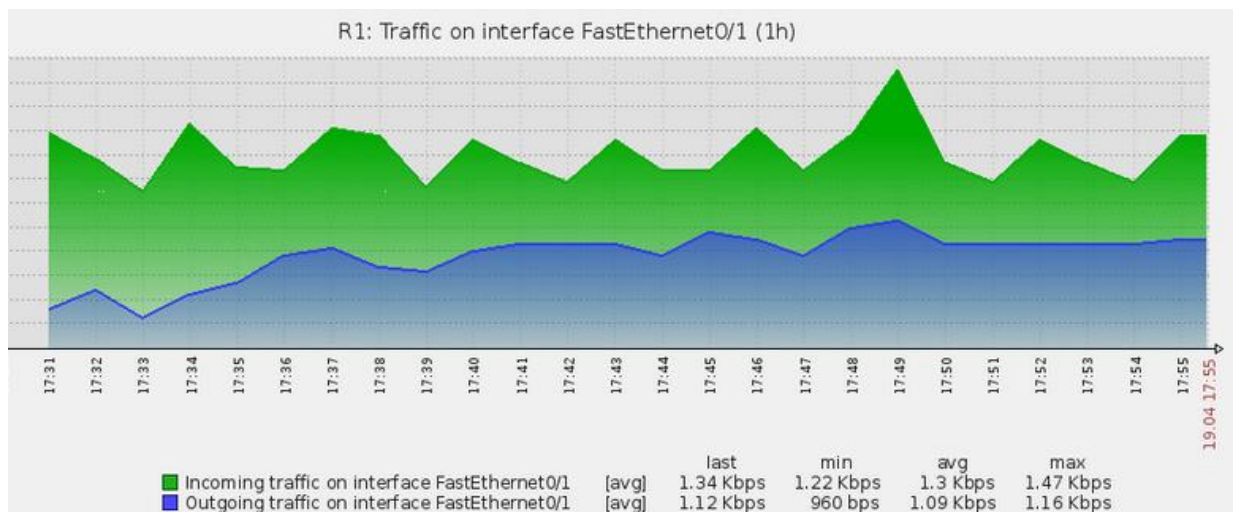


FIGURE 18. Template graph

6.4.2 Router 2 monitoring

In this section I will explain how to migrate templates on Router 2. The first step is to create Router 2 as a host. I have named it R2. The next step is to assign templates that have been used in Chapter 6.5.1 Figure 19 illustrates the step of adding templates for R2

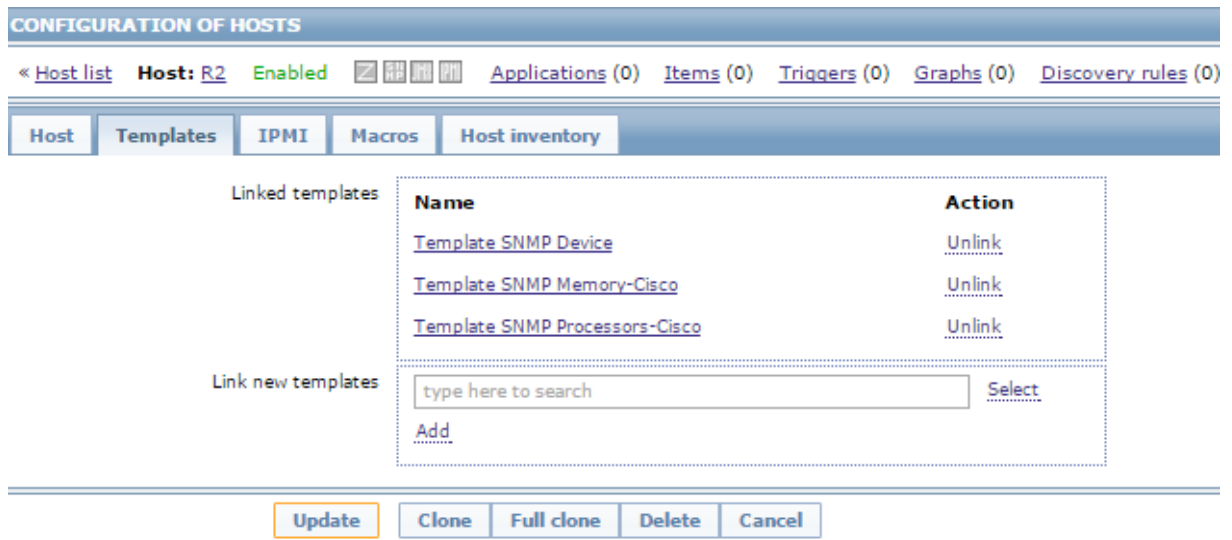


FIGURE 19. Templates on R2

As a result, it required approximately 1 minute to create and assign item on R2, in order to perform monitoring. As a result one set of templates could be used by unlimited number of hosts. The only thing should be taken into account that hosts have to be the same type of devices from the same vendor.

6.4.3 Server monitoring

Templates in Zabbix are not only limited to network devices. Servers could be also monitored by templates. By default, Zabbix offers templates for Linux and Windows based operating systems. In addition, the monitoring could be done two ways: SNMP or Zabbix Agent.

In this part I will monitor Ubuntu Server 1 with SNMP template. At the beginning, some configurations have to be done on Ubuntu Server. In the next step I will assign templates on a host. I would like to point out that Ubuntu Server 1 is installed as a virtual machine.

First I will update a package list of Ubuntu Server. This could be done by this command: `sudo apt-get update`. Next I have to install SNMP Server: `sudo apt-get install snmpd snmp`. This would allow Zabbix Server to monitor Ubuntu Server via SNMP.

When SNMP Server is installed, some configuration changes have to be done. The editing has to be done in configuration file with this command: `sudo vi /etc/snmp/snmpd.conf`. There is a need to comment the string about agent address: `#agentAddress udp:127.0.0.1:161`. This should be done due to the fact that Ubuntu Server is not going to be monitored via Zabbix Agent.

In addition, there is a need to add a line: `agentAddress udp:161,udp6:[::1]:161`. This command specifies port number 161. This is a UDP port number for SNMP. The last line has to be added is SNMP community: `rocommunity senna`. SNMP service has to be restarted: `sudo service snmpd restart`

The next step is to add Linux Server as a host in Zabbix GUI. The name of the host was chosen Linux Server 1. The template to monitor a new host was chosen SNMP OS Linux. In that template I have to specify SNMP community `senna`. This template contains all items that are required, that is why no additional editing has to be done.

Template has automatically created items and graphs of host. As a result, the Linux Server can be monitored. The graphical output of monitored data could be seen in Figure 20.

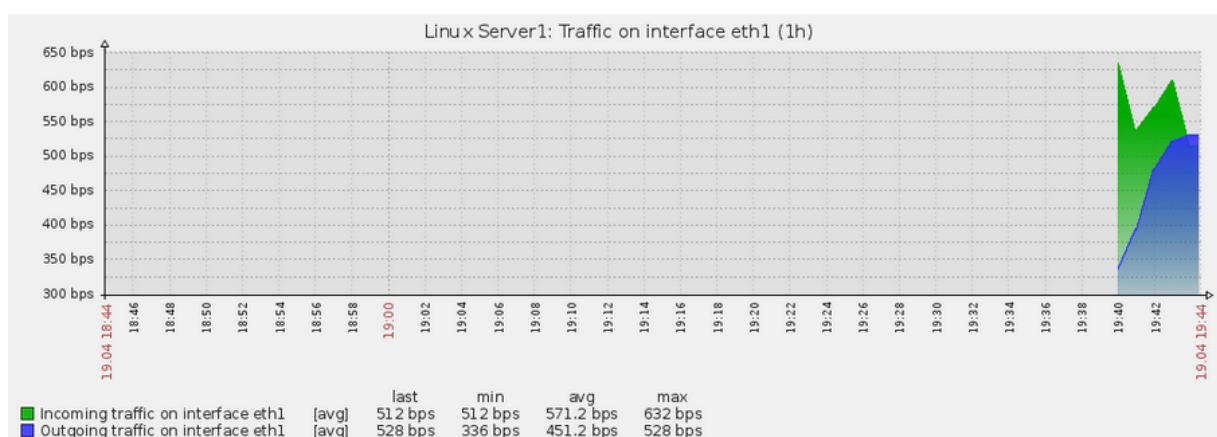


FIGURE 20. Graph of network traffic on Ubuntu Server 1

6.4.4 Conclusion

In this chapter I have been able to use templates in order to monitor processes on R1. The same set of templates has been used on R2. In addition, I have been able to monitor the performance of Ubuntu Server, installed as a virtual machine.

Templates are bringing considerable advantage for medium and enterprise-level network. They are helping to automate such a routine process as creating an item for a host. For a template there is need to configure items only once. After that an item can be easily used by many hosts. In addition, if changes in monitoring are required, they have to be done only once on a template. Hosts that are using this template will be automatically updated, instead of changing items on each host manually.

6.5 Agent Monitoring

Zabbix support Agentless and Agent-based monitoring. In sections 6.3 and 6.5 monitoring of the hosts has been done using agentless method. Zabbix Server has been collecting the data from hosts remotely using SNMP protocol.

In Agent-based monitoring the monitoring is done by an agent. It is software that is locally installed on a host in order to retrieve the monitoring data. Zabbix environment uses naturally Zabbix Agents. In most cases it is used to monitor processes on server.

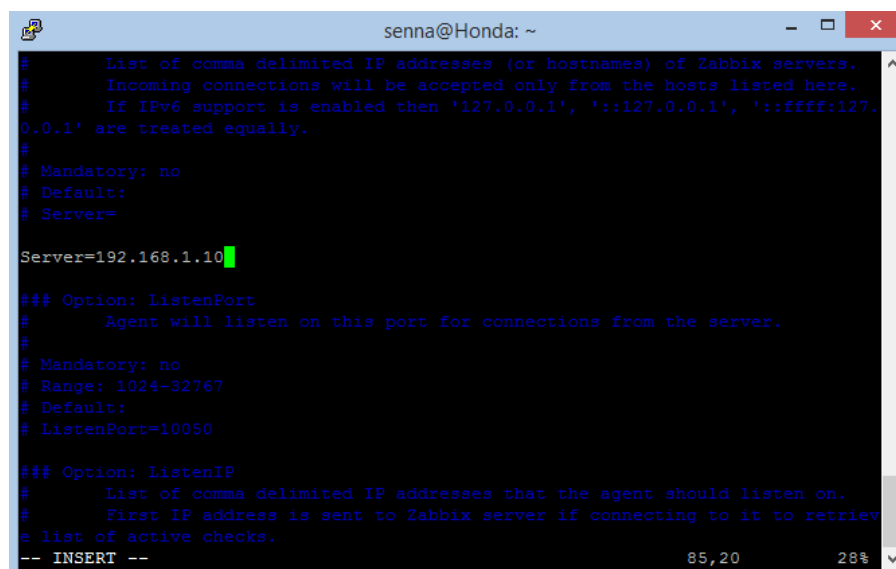
The key advantage of Agent-based monitoring is that it performs more accurate data collection than Agentless. In addition, the agent is a lightweight software that is why it does not consume too much disk space. However the agent may cause an unwanted footprint on the network performance.

In this chapter I will perform monitoring using Zabbix Agent. The host is going to be Ubuntu Server 2. It has been installed as a virtual machine. Figure 4 provides an overview of a network topology.

The first step is to install Zabbix Agent locally on a host. In addition, some configurations have to be done. The second step is to add Ubuntu Server as a host on Zabbix GUI. A template also has to be assigned.

6.5.1 Installing Zabbix Agent

The first step is to install Zabbix Agent locally on Ubuntu Server 2. However before starting to implement this step a package list is recommended to be updated. It could be done by using: `sudo apt-get update`. After that Zabbix Agent can be installed: `sudo apt-get install zabbix-agent`. When installation is done there is need to do configuration. Configuration file has to be edited: `sudo vi /etc/zabbix/zabbix_agentd.conf`. In my case it is `192.168.1.10`. In this step I have specified the address where to send the data that is going to be collected by Zabbix Agent. Figure 21 provides an overview on the process of configuration.



```

senna@Honda: ~
# List of comma delimited IP addresses (or hostnames) of Zabbix servers.
# Incoming connections will be accepted only from the hosts listed here.
# If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.
0.0.1' are treated equally.
#
# Mandatory: no
# Default:
# Server=

Server=192.168.1.10

### Option: ListenPort
# Agent will listen on this port for connections from the server.
#
# Mandatory: no
# Range: 1024-32767
# Default:
# ListenPort=10050

### Option: ListenIP
# List of comma delimited IP addresses that the agent should listen on.
# First IP address is sent to Zabbix server if connecting to it to retriev
e list of active checks.
-- INSERT --
85,20 28%

```

FIGURE 21. Assigning Server's address

If there is need to stop Zabbix Agent on a host, this could be done by this command: `sudo service zabbix-agent stop`. To start it again `sudo service zabbix-agent start`. If there is a need to uninstall Zabbix Agent a command: `sudo apt-get remove zabbix-agent` can be used.

6.5.2 Adding Templates

In the second step I have to add Ubuntu server as a host. I have called it Linux Server 2. In addition, this host is connected to Agent interface. The IP address is `192.168.1.20`.

After that a template can be assigned. I have chosen *Template OS Linux*. This template automatically creates items and graphs. With this template I have been able to monitor CPU load, disk usage and network traffic. Figure 22 illustrates the results.

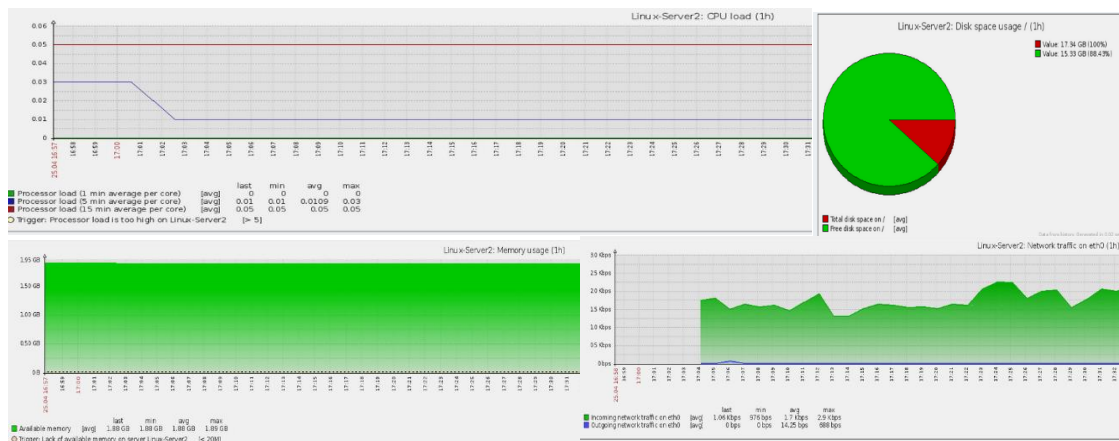


Figure 22. Results of Zabbix Agent monitoring

6.5.3 Conclusion

In this part I have been able to install and configure Zabbix Agent on Ubuntu Server 2. In addition, I have added a host Linux Server 2. There I have added a Template OS Linux. As a result, I have been able to monitor Linux Server 2 using Agent-based monitoring.

6.6 Auto Discovery

In section 6.5 I have been using templates in order to increase the speed of configuring monitoring. Compared to manual creation of items and graphs on each host this brings significant flexibility. However for enterprise-level networks manual assigning templates on each host manually still requires a lot of time. To automate this process, Auto Discovery can be used.

Auto Discovery automatically discovers devices within the network. In addition, discovered devices can be automatically added as hosts and templates may be assigned automatically. Auto Discovery allows to find devices that are monitored both by Agent-Based and Agentless monitoring. As a result, Auto Discovery is widely used in medium and enterprise-level networks.

Before starting to use Auto Discovery some pre-requirements have to be used. It is recommended to group hosts logically. Devices that are monitored by SNMP have to verify the name of the SNMP community. Devices that are monitored by agent-based principle have to verify if the Zabbix Agent is running.

In this chapter, I will perform Auto Discovery. In order to be able to do so, I will create a new network topology and categorize devices by their type. After that discovery rules have to be created for each host group. In order to perform automatic host creation and template assignment an action has to be created. Figure 23 provides an overview for this chapter.



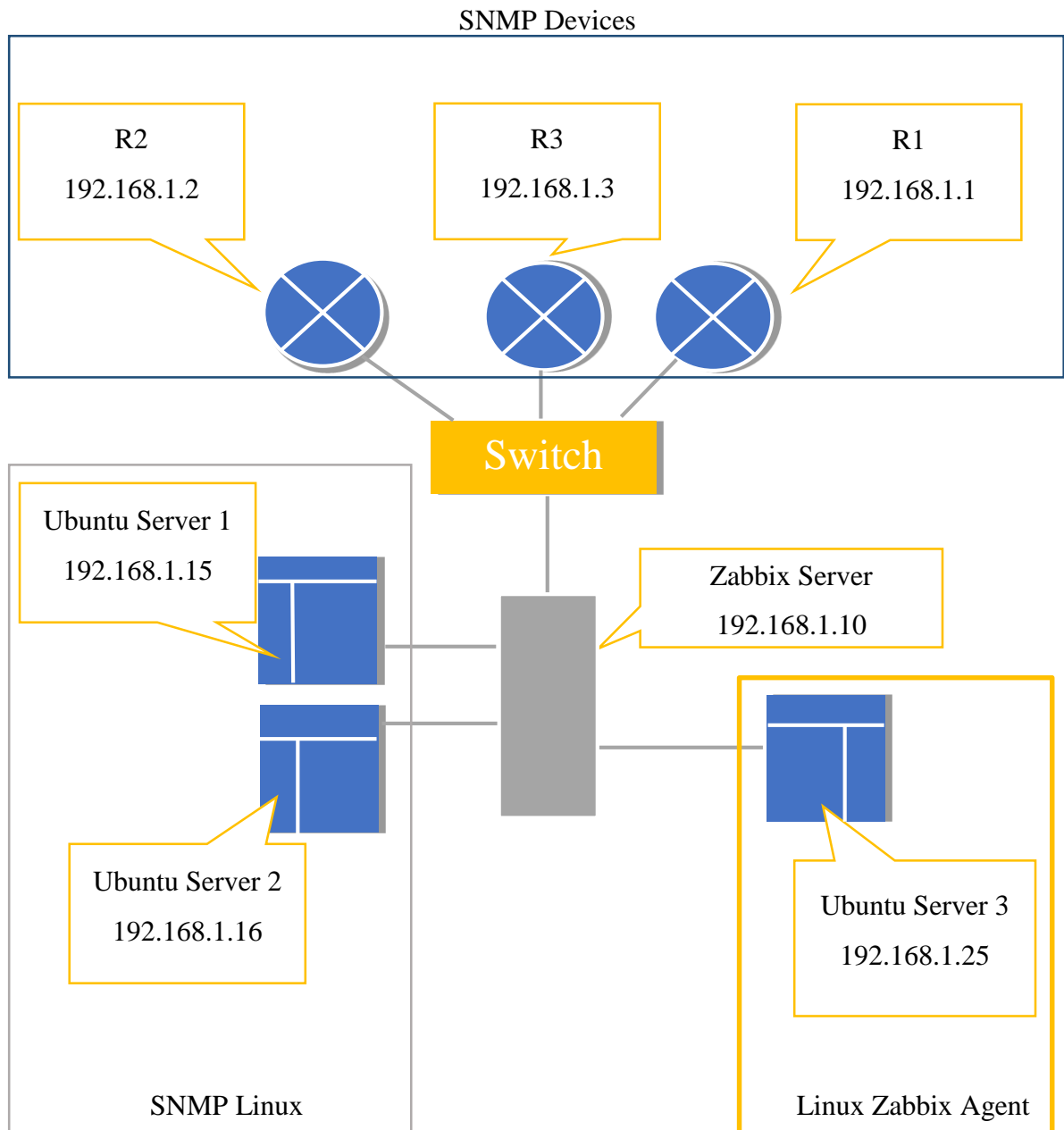
Figure 23. Roadmap for auto discovery

6.6.1 Network Topology

The first step is to create a new network topology. It is illustrated in Figure 24. That logical plan also illustrates the way how devices are grouped.

For enterprise-level networks it is essential to organize devices into groups. This provides clarity and transparency in monitoring and management. It is possible to group devices based on geographical zones. As an alternative, devices could be categorized based on type of devices and their functionality.

I would like to point out that Zabbix Server, Ubuntu Server1-3 are installed as virtual machines on the PC, the same way that it was done in Figure 4. In Figure 24 they are illustrated separately, in order to provide an overview on logical groups.



FUGURE 24. Network topology

Table 5 provides an overview on how devices in the new network are grouped. The table contains information about devices that are included, type of monitoring and IP range of that group. Even though hosts are going to be grouped in Zabbix GUI, it is essential to plan the grouping beforehand

Table 5. Host groups

Name of the host group	Devices	Type of monitoring	IP address Range
SNMP Devices	R1; R2; R3	SNMP	192.168.1.1-4
SNMP Linux	Ubuntu Server 1 Ubuntu Server 2	SNMP	192.168.1.11-20
Linux Zabbix Agent	Ubuntu Server 3	Zabbix Agent	192.168.1.21.-40

In this step, I have activated SNMP Server on devices that are going to be monitored Agentless. The name of SNMP community is *senna*. Zabbix Agent has been installed and configured on Ubuntu Server 3.

In order for Zabbix Server to be able to perform SNMP of device, SNMP Community of that device have to be mentioned on a host. There are two methods that allow to add SNMP Community on hosts. The first one has been described in Chapter 6.3. There I have been manually adding `{SNMP_COMMUNITY}` string. It was done only for one host. However for 300 hosts manual adding SNMP string value is a time consuming process. In addition, the advantages of Auto Discovery have been lost.

The second approach is to add `{SNMP_COMMUNITY}` string into template that is going to be used. As templates could be assigned to unlimited amount of hosts, SNMP community string is going to be added to any host automatically. As a result, there is no need assign it manually.

6.6.2 Creating Discovery Rule

In the second step I will create a discovery rule. Here I specify how devices are going to be discovered. However this step does not provide any automation such as adding templates. This has to be done in Action.

I want to discover all three groups of hosts: SNMP Devices, SNMP Linux and Linux Zabbix Agent. That is why there is a need to create three separate discovery rules. Figure 25 provides an overview on a process of creation a discovery rule for a host group of SNMP devices.

The screenshot shows the 'Discovery rule' configuration window in Zabbix. The 'Name' field is 'Network Devices- Rule'. 'Discovery by proxy' is set to 'No proxy'. The 'IP range' is '192.168.1.1-4'. 'Delay (in sec)' is '10'. Under 'Checks', a 'New' button is visible. A sub-form for adding a check is shown with 'Check type' as 'SNMPv2 agent', 'Port range' as '161', 'SNMP community' as 'senna', and 'SNMP OID' as 'SNMPv2-MIB::sysName.0'. 'Add' and 'Cancel' buttons are present. 'Device uniqueness criteria' is set to 'IP address' and the 'Enabled' checkbox is checked.

FIGURE 25. Creating Discovery rule

IP range allows to limit Discovery Rule within one group of devices. The IP range for SNMP Devices group is *192.168.1.1-4*. The IP Range for other groups can be found in *Table 5*.

Delay means how often Zabbix Server will re-run Discovery process. As a result, Zabbix Server will dynamically update the list of discovered devices. In my case, I have chosen a delay in 10 seconds.

Check is the way how Zabbix Server performs network discovery. In other words, Zabbix Server will find only devices that satisfy check type requirements. It could be SNMP v2 agent or Zabbix Agent. For SNMP Devices the check type is SNMP v2 agent. The SNMP community is *senna*.

A unique SNMP OID number in this case is *SNMPv2-MIB::sysName.0*. In Device Uniqueness criteria I have chosen IP address. This is useful in situations when a host with an IP address from that range already exists. When auto discovery finds this address again, it will ignore it.

For Zabbix Agent based devices the process of creating discovery rule is almost the same. The only difference is that there is a need to specify a key. Key is a parameter that helps to retrieve information from a devices that have Zabbix Agent installed. In my case the key is *system.name*.

6.6.3 Creating Action

The third step according to Figure 23 is to create an action for a discovery. When Zabbix discovers a host, it will check its conditions. If conditions satisfy the requirements, Zabbix Server will perform operations on that host.

Figure 26 illustrates the process of creating conditions for SNMP device host group. Same conditions are applied for SNMP Linux group. The discovery will be performed only if the host is *up*. The service type is SNMPv2 agent.

CONFIGURATION OF ACTIONS

Action Conditions Operations

Type of calculation: And (A and B and C)

Label	Name	Action
A	Discovery status = Up	Remove
B	Service type = SNMPv2 agent	Remove
C	Discovery rule = Network Devices- Rule	Remove

New condition: Discovery rule = [] [Select](#)
[Add](#)

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Figure 26. Conditions

If conditions of the host are matching, Zabbix Server will forward host to operations. Figure 27 illustrates the process of operation creation. When a host is discovered and its conditions match, the host is going to be added to in group of hosts automatically. In addition, a template would be assigned

CONFIGURATION OF ACTIONS

Action Conditions Operations

Action operations:

Details	Action
Add to host groups: SNMP Devices	Edit Remove
Link to templates: AutoDiscovery Template SNMP Device	Edit Remove
New	

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Figure 27. Operations

Table 6 provides an overview on host group's conditions and operations. Conditions would be performed separately for different groups of devices. As a result output would be as it was defined in operations.

Table 6. Alert parameters

	CONDITIONS			OPERATIONS	
	Discovery Rule	Service Type	Discovery Status	Add Host Group	Link to Template
SNMP Devices	SNMPBasedRule	SNMPv2 Agent	Up	SNMP Devices	Template SNMP Device
SNMP Linux	SNMPBasedRule	SNMPv2 Agent	Up	SNMP Linux	Template SNMP Linux
Linux Zabbix Agent	AgentBasedRule	Zabbix agent	Up	Linux Zabbix Agent	Template OS Linux

As a result, when Zabbix Server performs Auto Discovery, it will check host by conditions that have been manually created. If conditions are matching, operation will be done. In my case, the host would be automatically added to one of the three hosts groups and after templates are automatically area added.

6.6.4 Performing Auto Discovery

As the result, all configurations are done. In the fourth and the last step I will perform Auto Discovery. At the beginning I will perform Auto Discovery for SNMP Devices.

In order to start Auto Discovery for SNMP Devices, actions and discovery rules of this group have to be enabled. After short period of time, Zabbix has found three hosts. A bit later I have performed Auto Discovery separately for SNMP Linux and Linux Agent groups. Figure 28 illustrates discovery status which is displayed on the main page.

Discovery status		
Discovery rule	Up	Down
Linux-Rule	2	1
LinuxAgent-rule	1	0
Network Devices- Rule	3	0
Updated: 18:14:05		

Figure 28. Discovery status

I would like to point out that on Linux-Rule discovery status one of the hosts was down. I have found a mistake which was discovered during auto discovery process on Ubuntu Server 2. The problem was that I have assigned a wrong IP address on it. I have changed it to the right one. As a result, Zabbix has discovered Ubuntu Server 2 two times with two different IP addresses. The first one no longer exists and that is why Zabbix marked that address as it is down. Figure 29 reveals details on discovered devices.

Discovered device	Monitored host
192.168.1.15	192.168.1.15
192.168.1.16	192.168.1.16
192.168.1.20	192.168.1.20

FIGURE 29, Discovered devices based on Linux-Rule

In Section 6.6.3 I have created an action that would be activated during auto discovery. Two operations are expected to be done. The first one is that discovered devices are automatically added as hosts and host groups. Figure 30 illustrates the result. However, the only thing that needed to be done manually was renaming the hosts according to logical plan presented in Figure 24.

Hosts
Displaying 1 to 7 of 7 found

<input type="checkbox"/>	Name	Applications	Items	Triggers	Graphs	Discovery	Web	Interface
<input type="checkbox"/>	192.168.1.1	Applications (2)	Items (54)	Triggers (6)	Graphs (6)	Discovery (1)	Web (0)	192.168.1.1: 161
<input type="checkbox"/>	192.168.1.2	Applications (2)	Items (46)	Triggers (5)	Graphs (5)	Discovery (1)	Web (0)	192.168.1.2: 161
<input type="checkbox"/>	192.168.1.3	Applications (2)	Items (54)	Triggers (6)	Graphs (6)	Discovery (1)	Web (0)	192.168.1.3: 161
<input type="checkbox"/>	192.168.1.15	Applications (4)	Items (79)	Triggers (11)	Graphs (11)	Discovery (3)	Web (0)	192.168.1.15: 161
<input type="checkbox"/>	192.168.1.16	Applications (4)	Items (79)	Triggers (11)	Graphs (11)	Discovery (3)	Web (0)	192.168.1.16: 161
<input type="checkbox"/>	192.168.1.25	Applications (10)	Items (32)	Triggers (15)	Graphs (5)	Discovery (2)	Web (0)	192.168.1.25: 10050
<input type="checkbox"/>	Zabbix server	Applications (11)	Items (76)	Triggers (45)	Graphs (14)	Discovery (2)	Web (0)	127.0.0.1: 10050

Export selected Go (0)

FIGURE 30. Automatically added hosts

In Figure 31-36 there are illustrated graphs which represent network bandwidth from perspective of different hosts.

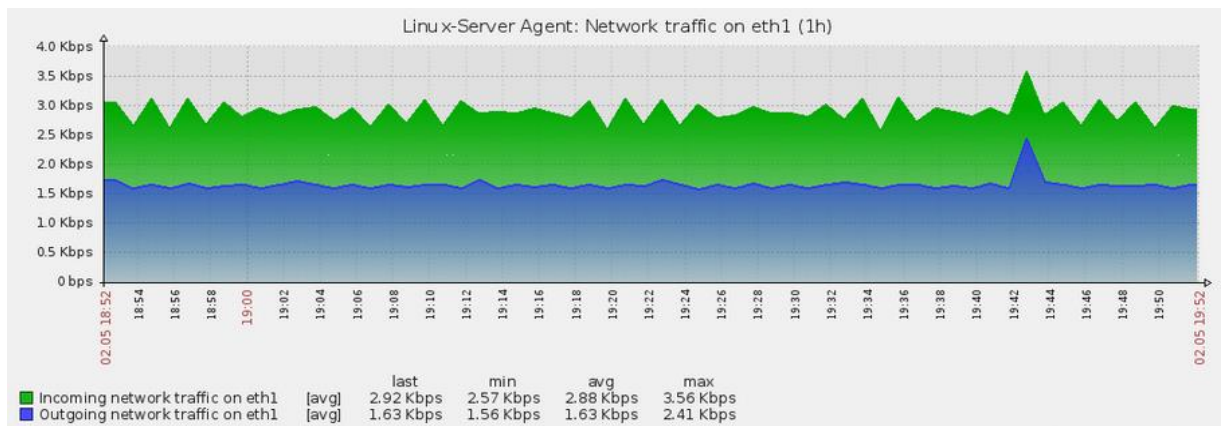


FIGURE 31. Linux-Server Agent Network Performance Graph

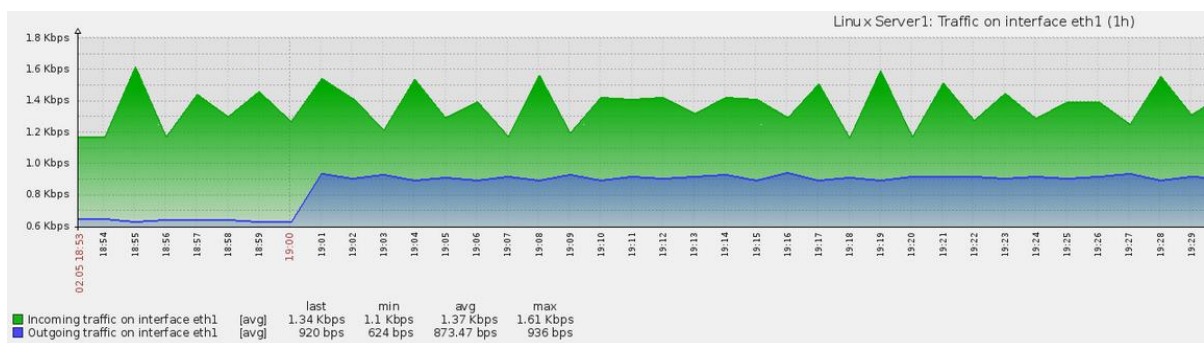


FIGURE 32. Linux Server 1 Network Performance Graph

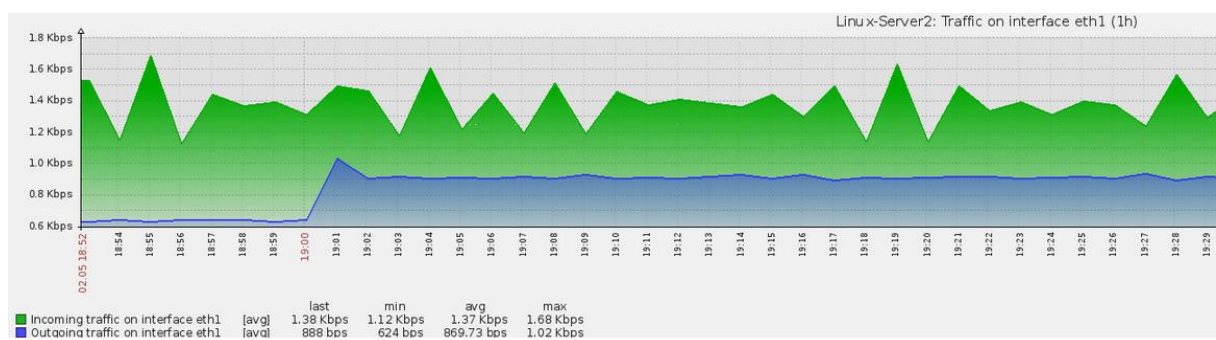


FIGURE 33. Linux Server 2 Network Performance Graph

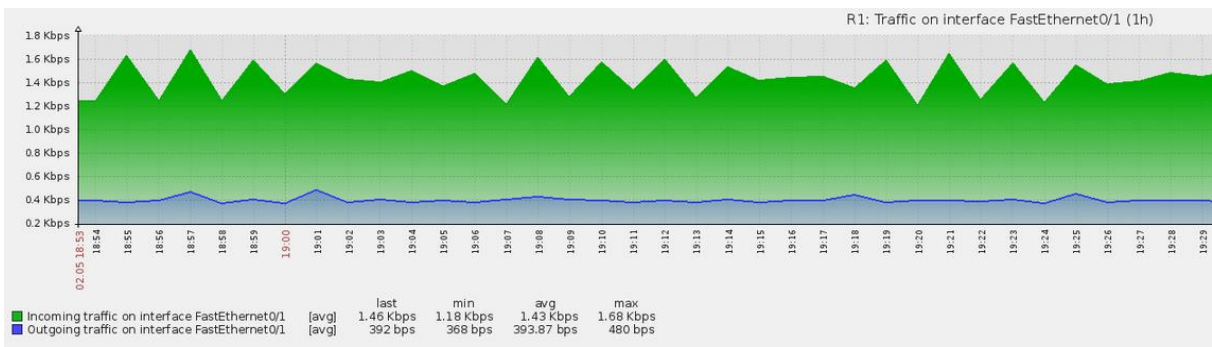


FIGURE 34. R1 Network Performance Graph

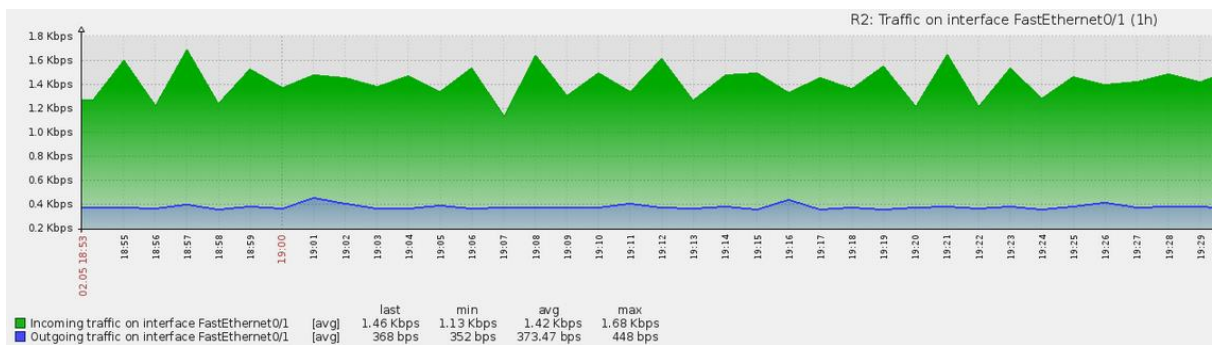


FIGURE 35. R2 Network Performance Graph

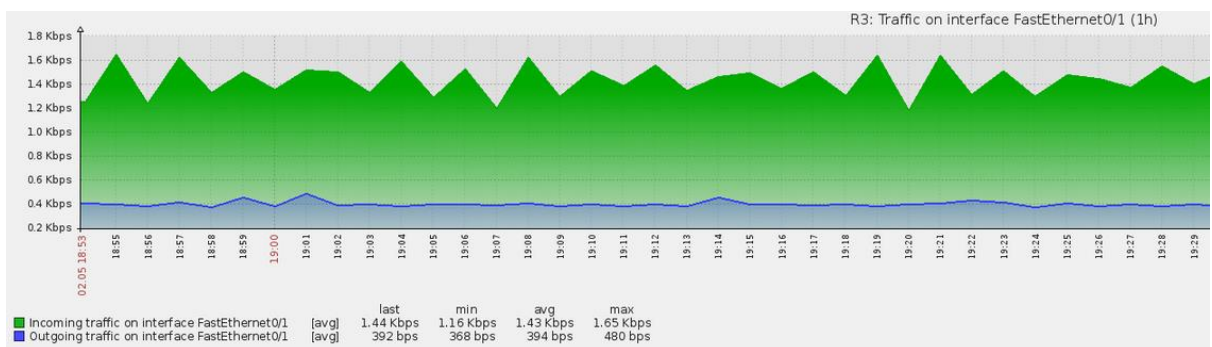


FIGURE 36. R3 Network Performance Graph

After performing auto discovery, it is recommended to disable it. The reason is that auto discovery may cause footprint of the network performance. In addition, during discovery rule creation it is recommended to configure delay on 3600 seconds in order not to decrease the overall network performance.

I would like to point out that I have assigned delay on 10 seconds even though it is recommended to have 3600 seconds. I have done it on purpose in order to speed up the process of auto discovery. In this step, network performance was not an issue that has to be taken into account.

6.6.5 Conclusion

As a result, I have successfully completed Auto Discovery of the network that was described in Figure 24. I have created discovery rules and actions for each of the three main host groups. Network devices within the given network have been successfully detected.

In addition, Zabbix has automatically discovered devices within the network. Every discovered device has been added as the host. On each host a template has been assigned automatically.

I have proved that the auto discovery helps to improve network monitoring in general. It allows to discover network devices automatically. In addition, actions are lowering the network administrator's routine work such as creating host names and host groups. Moreover, templates can be added automatically. Network discovery is particularly relevant for enterprise-level networks that consist of thousands of devices.

7 CONCLUSION

For enterprises that completely depend on ICT, it is required to have the efficient network infrastructure. As it is impossible to provide 100 percent availability, network monitoring helps to reduce impact of the network inability. Network monitoring became an essential part of any network size. It brings monitoring of network components 24/7. This provides clarity and transparency of network infrastructure and performance. With immediate alert notification, network administrator may start failure troubleshooting instantly. As a result, end users may not even notice the issue. Network monitoring is improving performance and availability of the network. This leads to constant workload within the enterprise.

The choice of the network monitoring tool has to be done responsibly. Network administrator should be confided about network structure and has to have clear understanding of network devices and components that are going to be monitored. It should be taken into account that some features may result in declining of the network performance.

I am pleased to announce that I have been able to succeed in all steps of my thesis through putting a lot of effort, patience and attention for the whole process of writing it. In my thesis, I created an understanding of what network monitoring tool is, advantages that are brought and

its key components. Zabbix has been chosen in order to perform monitoring. However, before being able to do so, I made a research about Zabbix architecture, available features and analyzed its future development.

Based on that knowledge, I have performed testing of Zabbix in real environment. I have performed different monitoring techniques in order to outline the most sufficient one for different network size. Even though testing of monitoring has been done in small size network, methods and techniques could be easily scaled up to larger networks.

BIBLIOGRAPHY

Gates, Bill 2009. Business @ the Speed of Thought. Grand Central Publishing.

Crock 2012. Habrahabr. WWW-publication. <http://habrahabr.ru/company/croc/blog/144941/>. Updated 31.05.2012. Referred 04.05.2015

Network Monitoring Software. Spice Works. WWW-publications. <http://www.spice-works.com/free-network-monitoring-management-software/>. Referred 05.02.2015

Network Monitoring and Managemnet. Tibbo Systems. WWW-publications. http://aggregate.tibbo.com/solutions/network_management/network_monitoring.html. Referred 05.02.2015

Zabbix Features, Zabbix. WWW-publications. <http://www.zabbix.com/features.php>. Referred 05.02.2015

Castro, Rui, Coates, Mark, Liang, Gang, Nowak, Robert & Yu, Bin 2004 Network Tomography: Recent Developments. Institute of Mathematical Statistics.

Drogseth, Dennis 2003. HP invests in route analytics with Packet Design. Networkworld. WWW-publication. <http://www.networkworld.com/article/2338253/infrastructure-management/hp-invests-in-route-analytics-with-packet-design.html>. Updated 10.11.2003. Referred 15.02.2015.

O'Donnell, Glenn 2004. Route Analytics Enrich Technology Relationships. META Group, Inc. PDF document. <http://www.glennodonnell.com/documents/d2751-RouteAnalytics.pdf>. Updated 04.02.2004. Referred 16.02.2015

Enhancing Network Monitoring with Route Analytics 2013. Packet Design. PDF document. <http://www.packetdesign.com/resources/white-papers/Enhancing%20Network%20Monitoring%20with%20Route%20Analytics.pdf>. Referred 20.05.2015.

Zabbix True Open Source. Zabbix. WWW-publication. http://www.zabbix.com/true_open_source.php. Referred 05.03.2015

Beal, Vangie. Webopedia. WWW-publication. <http://www.webopedia.com/TERM/G/GPL.html>. Referred 05.03.2015

eG Agentless Monitoring. eG Innovations. WWW-publication. <http://www.eginnovations.com/web/egagentless.htm>. Referred 08.03.2015.

The Truth about Agent vs. Agentless Monitoring. Uptime software. PDF document. <http://www.uptimesoftware.com/pdfs/TruthAboutAgentVsAgentLess.pdf>
Referred 08.03.2015.

Network Monitoring 2015, InterMapper, WWW-publication. <http://www.helpsystems.com/intermapper/network-monitoring>. Updated 30.04.2015. Referred 05.05.2015

Clemm, Alexander, Bansal, Anil 2003. Auto-discovery at the network and service management layer. IEEE. WWW-publication. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=1194192&abstractAccess=no&userType=inst>. Updated 28.03.2003.
Referred 19.03.2015

Auto discovery. Zabbix. WWW-publication. http://www.zabbix.com/auto_discovery.php. Referred 23.03.2015

Grouping And Group Operations. Tibbo Systems. WWW-publication. http://aggregate.tibbo.com/technology/management/grouping_and_group_operations.html. Referred 21.03.2015.

Zabbix. Low Level Discovery. Habrahabr. WWW-publication. <http://habrahabr.ru/company/zabbix/blog/203050/>. Updated 21.11.2013 Referred 15.04.2015

Vladishev, Alexei 2014. 5 Thing to improve in Zabbix. Zabbix conference 2014. Conference in Riga, Latvia. 12.9-13.9.2014. WWW-publication
https://www.youtube.com/watch?v=SwuqNIJb_o. Updated 20.11.2014. Referred 25.03.2015

Vladishev, Alexei 2014. Opening Speech. Zabbix conference 2014. Conference in Riga, Latvia. 12.9-13.9.2014. WWW-publication

<https://www.youtube.com/watch?v=QXmVe3OBLOo> Updates 20.11.2014. Referred 29.03.2014

Zabbix. Services. WWW-publication <http://www.zabbix.com/services.php>. Referred 27.03.2015.

Zabbix. Zabbix Documentation 2.4. WWW-publication <https://www.zabbix.com/documentation/2.4/>. Referred 20.04.2015.