Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Porovnání free monitoring nástrojů včetně PoC (Proof of Concept)

Bakalářská práce

Autor: Viktor Pešek
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. Ing. Vladimír Bureš, Ph.D., MBA

Odborný konzultant: Ing. Karel Schejbal, Production Manager, Unicorn Systems a.s.

Hradec Králové duben 2019

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

……

V Hradci Králové dne 19.4.2019 Viktor Pešek

Poděkování:

Děkuji vedoucímu bakalářské práce doc. Ing. PhD. Vladimíru Burešovi za metodické vedení práce. Dále děkuji Ing. Karlu Schejbalovi za odbornou konzultaci a podporu při zpracování bakalářské práce.

Anotace

Tato bakalářská práce se zabývá problematikou monitorování aplikací pomocí free monitorovacích nástrojů. Cílem této práce je popsat doménu monitorování, představit používané postupy, najít vhodné nástroje a porovnat přístupy, které používají při implementaci monitorování. Dále je v této práci věnován prostor k prozkoumání možností integrace odesílání notifikací do Instant Messaging aplikací. Pro tuto implementaci je vybrána aplikace Telegram, která poskytuje API pro posílání zpráv mimo uživatelské rozhraní aplikace. V bakalářské práci jsou porovnávány nástroje Nagios a Zabbix. U nástrojů je zaměřeno na porovnání instalace, konfigurace a uživatelské přívětivosti obou nástrojů. Poslední kapitoly bakalářské práce jsou věnovány porovnání nástrojů.

Annotation

Title: Comparison of monitoring tools including PoC (Proof of Concept)

This bachelor thesis deals with application monitoring with free monitoring tools. The aim of this work is to describe the monitoring domain, introduce the procedures used, find the appropriate tools and compare the approaches they use to implement monitoring. Furthermore, the work is devoted to exploring the possibilities of integrating the sending of notifications to Instant Messaging applications. The Telegram application is provided for this implementation, which provides an API for sending messages outside the application's user interface. The bachelor thesis compares Nagios and Zabbix tools. The tools are aimed at comparing the installation, configuration, and user-friendliness of both tools. The last chapters of the thesis are devoted to the comparison of tools.

Obsah

1 Úvod ... 1

2 Teoretická část ... 2

2.1 Monitorování ... 2

2.1.1 Historie ... 2

2.1.2 Monitorovací systém ... 3

2.2 Přístupy k monitorování ... 4

2.2.1 Manuální iniciované uživatelem ... 4

2.2.2 Reaktivní (pasivní) ... 5

2.2.3 Proaktivní (aktivní) ... 5

2.3 Metriky ... 6

2.3.1 Meřítko (gauge) ... 7

2.3.2 Počítadlo (counter) ... 8

2.3.3 Časomíra (timer) ... 8

2.3.4 Agregace metrik ... 8

2.4 Alerting ... 9

2.4.1 Alert ... 10

2.4.2 Identifikace problému ... 10

2.4.3 Rozdělení alertů podle metody zasílání ... 11

2.5 Vlastnosti nástrojů ... 12

2.5.1 Klíčové vlastnosti nástrojů ... 12

2.5.2 Rozšiřující vlastnosti monitorovacích nástrojů ... 14

3 Praktická část ... 16

3.1 Požadavky na monitorovací nástroje ... 16

3.1.1 Oblíbenost nástroje ... 16

3.1.2 Free licence ... 16

3.1.3 Webové rozhraní ... 16
3.1.4 Vytváření šablon a skupin pro služby a klientské stanice ... 17
3.1.5 Využití agentů ... 17
3.1.6 Stáří poslední verze nástroje ... 17
3.2 Kontroly pro monitorovaní ... 17
3.2.1 Obecná kontrola společná pro všechny servery ... 17
3.2.2 Kontrola databáze ... 18
3.3 Vybrané nástroje ... 18
3.3.1 Nagios ... 19
3.3.2 Zabbix ... 30
3.4 Porovnání nástrojů ... 40
3.4.1 Instalace ... 41
3.4.2 Konfigurace ... 41
3.4.3 Uživatelská přívětivost ... 41
3.4.4 Dokumentace ... 42
3.4.5 Rozsah funkcionalit ... 42
3.4.6 Celkové zhodnocení nástroje Nagios ... 42
3.4.7 Celkové zhodnocení nástroje Zabbix ... 43
4 Závěr ... 45
5 Seznam literatury ... 46

Seznam obrázků

Obrázek 1: Graf využití procesoru ... 7
Obrázek 2: Příklad využití měřítka. ... 7
Obrázek 3: Počítadlo a jeho využití ... 8
Obrázek 4: Využití časomíry ... 8
Obrázek 5: Agregované metriky ... 9

Obrázek 6 Webové rozhraní Nagios Core ... 22
Obrázek 7 NRPE ... 24
Obrázek 8 Nagios adresářová struktura ... 26
Obrázek 9 Nagios Telegram ... 30
Obrázek 10 Přihlášení Zabbix webové rozhraní ... 32
Obrázek 11 Zabbix adresářová struktura ... 34
Obrázek 12 Vytvoření klientské stanice ... 35
Obrázek 13 Zobrazení alertů agregovaně podle skupin ... 35
Obrázek 14 Přidání šablony ke klientské stanici ... 36
Obrázek 15 Hlavička stránky při konfiguraci klientské stanice ... 36
Obrázek 16 Konfigurace alertů (Triggers) ... 36
Obrázek 17 Vytvoření šablony ... 37
Obrázek 18 Logický výraz pro využití paměti nad 95% ... 38
Obrázek 19 Přidání nové notifikace ... 39
Obrázek 20 Přidání akce pro odesílání notifikací ... 39
Obrázek 21 Vložení identifikátoru skupiny pro uživatele ... 39
Obrázek 22 Telegram ukázka posílání zpráv ... 40

Seznam tabulek

Tabulka 1 Požadavky na nástroje ... 18
Tabulka 2 Porovnání nástrojů ... 44

Seznam konfigurací

Konfigurace 1 Nastavení SELinux ... 19
Konfigurace 2 Zdrojové soubory a přidání uživatele nagios ... 20
Konfigurace 3 Instalace démona ... 20
Konfigurace 4 Instalace webového serveru a výchozí konfigurace pro Nagios ... 20
Konfigurace 5 Nastavení Firewall pravidel ... 21
Konfigurace 6 Vytvoření uživatele pro webové rozhraní ... 21
Konfigurace 7 Příkazy pro kontrolu konfigurace ... 21
Konfigurace 8 Spuštění služeb ... 21

Konfigurace 9 Prerekvizity pro Nagios Plugins 22
Konfigurace 10 Stažení a instalace Nagios Plugins 23
Konfigurace 11 Stažení NRPE 24
Konfigurace 12 Instalace služby NRPE 25
Konfigurace 13 Nastavení Firewallu a konfiguračního souboru 25
Konfigurace 14 Založení klientské stanice 27
Konfigurace 15 Založení skupiny klientských stanic 27
Konfigurace 16 Založení metriky 28
Konfigurace 17 Příkaz Memory 28
Konfigurace 18 Stažení skriptu pro Nagios Telegram 29
Konfigurace 19 Nastavení příkazu pro posílání zpráv Nagios 30
Konfigurace 20 Instalace MySQL a Zabbix serveru 31
Konfigurace 21 Nastavení časového pásma pro Zabbix 32
Konfigurace 22 Instalace Zabbix agenta 33
Konfigurace 23 Parametry skriptu pro Zabbix Telegram 38

1 Úvod

Tato bakalářská práce se zabývá porovnáním nástrojů pro monitorování aplikací, které jsou provozovány na linuxových serverech. Zásadními otázkami jsou: Jaké nástroje jsou vhodné pro monitorování aplikaci? Jaké jsou jejich vlastnosti? Co je potřeba pro konfiguraci těchto nástrojů? Jaké jsou rozdíly mezi nástroji?

Monitorování je proces používaný při provozování aplikací. Jedná se o sledování stavů zařízení, aplikací a dalších nástrojů v infrastruktuře aplikace. Monitorování je používáno pro informování administrátorů o aktuálním stavu prostředí (infrastruktury).

Bakalářská práce je rozdělena na praktickou a teoretickou část. Teoretická část se zabývá popisem monitorování. Je v ní obsažen popis jednotlivých části monitorování a monitorování obecně. V teoretické části jsou dále popsány jednotlivé principy, používané při monitorování. Ve druhé části jsou vybrány 2 nástroje, použité pro implementaci monitorování. U každého nástroje je popsán průběh instalace a konfigurace. Na konci druhé části bakalářské práce jsou nástroje porovnány podle stanovených parametrů.

2 Teoretická část

2.1 Monitorování

Podle Tsaie [1] je princip monitorování v sledování stavu a dostupnosti jednotlivých IP zařízení v síti. Monitorování je proces analýzy serverů z hlediska dostupnosti, výkonu, zabezpečení a dalších procesů souvisejících s provozem [2]. Z hlediska monitorování lze sledovat celou řadu věcí. Ligus [3] uvádí, že monitorování se stalo zastřešujícím termínem, jehož význam silně závisí na kontextu, a jedná se o proces poznávání systému.
Monitorování pomáhá administrátorům mít přehled o kritických místech v síti [1]. Turnbull [4] dodává, že monitorování je proces udržování dohledu nad existencí změn stavu a datového toku v sytému.

Hlavním přínosem je přehled o všech zařízeních v síti. Při správném nastavení upozornění může administrátor vědět o možných chybách dlouho předtím, než ve skutečnosti v systému nastanou. To administrátorům dává čas na zmírnění, nebo úplné odvrácení hrozby výpadku. [1]

2.1.1 Historie

První myšlenka, která souvisela s monitorováním síťových zařízení, vznikla na přelomu 80. a 90. let minulého století. V průběhu 90. let byl ve většině operačních systémů implementován systém, který je v dnešní době znám, jako například Správce úloh v operačních systémech od společnosti Microsoft. [5]

Koncem 90. let byla většina sledovacích systémů primárně určena pro monitorování lokálních sítí. Postupným vývojem a rozšířením internetu, hlavně ve 21. století, se zjistilo, že systémy, které jsou dostupné online, a systémy, které jsou dostupné jen v rámci jedné lokální sítě (LAN), se chovají odlišně. [5]

To vedlo ke vzniku nové generace monitorovacích nástrojů, která si uměla poradit s protokoly používanými v rámci internetu. Příklady nástrojů této generace jsou například: Nagios, Zabbix, Cacti a další. Tyto nástroje podporovaly

většinu tehdy používaných platform (Linux, Windows, …) a obsahovaly jednoduché webové rozhraní určené pro správu a nastavení daného software. [5]

Podle Churchmana [5] se v dnešní době pohled na monitorování změnil. Přínos už není jen ve sledování a nahrávání dat. Churchman [5] dále uvádí: [1]*„Monitoring je agregace dat, monitoring je filtrování, monitoring je analýza, monitoring je rozhodování, a hlavně monitoring je akce.*“ (překlad autor). Tyto systémy umožňují získat pro uživatele zajímavé informace, které mohou pomoci při provozování online služeb.

2.1.2 Monitorovací systém

Ligus [3] uvádí, že monitorovací systém je množina softwarových komponentů, které nahrávají, ukládají a zobrazují monitorovaná data. Většina monitorovacích systému sdílí stejnou architekturu a jsou si velmi podobné. Podle Turnbulla [4] techniky používané při monitorování informačních systémů protínají zpracování v reálném čase, statistiku a analýzu dat.

Dále Turnbull [4] uvádí, že monitorovacím systémem nazýváme sadu softwarových komponentů, používaných pro sběr dat, jejich zpracování a následnou prezentaci. Celý proces začíná sbíráním vstupních dat. Agenti shromažďují data a starají se o vyhodnocování vstupů do monitorovacího systému.

Agent je proces, který neustále nahrává data a posílá je do monitorovacího systému. Systém ukládá vstupní data do tzv. metrik a vyhodnocuje možné porušení nastavených podmínek, kdy se mají odesílat notifikace administrátorovi o případné chybě v systému. [3]

Ligus [3] dále rozděluje proces monitorování na tři části:

1) **Sběr dat** – Data o systému jsou sbírána pomocí agentů ze serverů, databází a síťových zařízení. Zdrojová data jsou logy nebo systémová měření jednotlivých zařízení. Agenti seskupí vstupy do metrik a přidají jim další vlastnosti, například časové razítko, kdy byl vstup zaznamenán. Vstupy

[1] Monitoring is data agregation, monitoring is filtering, monitoring is analytics, monitoring is decision-making, and monitoring is action.

jsou dále posílány monitorovacímu systému a ukládány v databázi metrik. [3]

2) **Shromažďování a ukládání dat** – Příchozí vstupní data jsou seskupena podle jejich vlastností a následně uložena v příslušných metrikách. Datové vstupy jsou získávány z metrik a následně jsou předávány jeden po druhém ke kontrole podmínek výskytu chybných nebo anomálních stavů v systému. Když je nějaká z těchto podmínek splněna, odešle se administrátorovi zpráva o chybném stavu v systému. [3]
3) **Vizualizace a prezentace** – Administrátor může generovat grafy vybraných časových intervalů pro kontrolu systému. Graf by měl po zásahu do systému a odstranění poruchy dát administrátorovi přesnou zpětnou vazbu do jaké míry nápravná opatření pomohla. [3]

Monitorovací systém poskytuje referenční bod pro všechny operátory. Jeho výhody jsou nejvíce patrné ve vyspělých organizacích, kde infrastrukturní týmy, vývojáři a systémoví inženýři mají možnost volně komunikovat, vyměňovat pozorování a přenést odpovědnost. Mít jediný bod pro všechny týmy zvyšuje účinnost detekce chyb v systému. [3]

2.2 Přístupy k monitorování

Turnbull [4] rozlišuje několik přístupů k monitorování:

- Manuální iniciované uživatelem, žádné monitorování
- Reaktivní (pasivní)
- Proaktivní (aktivní)

2.2.1 Manuální iniciované uživatelem

Tento typ monitorování je pouze iniciativou zaměstnance a je prováděn manuálně. Většinou se jedná o procházení kontrolních seznamů, vytváření jednoduchých skriptů a další neautomatizované procesy. Manuální monitorování se často stává pouze akcí na chybnou činnost systému v minulosti. Například, pokud v minulosti docházelo k častému výpadku zařízení, administrátor jej v budoucnu častěji kontroluje, aby předešel výpadku. Tento způsob monitorování

přináší velmi malou, nebo žádnou přidanou hodnotu při hodnocení kvality služeb. Využívá se nejčastěji v malých firmách, kde není vyhrazené personální obsazení pro IT. [4]

2.2.2 Reaktivní (pasivní)

Reaktivní monitorování je podle Turnbulla [4] většinou už automatické, s pozůstatky manuálního monitorování některých komponentů. Monitorovací nástroj upozorní administrátora až ve chvíli, kdy se systém dostane do chybového stavu [6]. Sledování je založeno na jednoduchých prahových hodnotách tzv. **thresholdech** a upozornění jsou odesílána přes email, SMS atp. Společnost Rapid7 [7] uvádí, že reaktivní monitorování ukazuje pouze, jak sledované zařízení pracuje v nynějších podmínkách, ale neumí poskytnout náhled, jak by zařízení mohlo pracovat v budoucnu.

Rapid7 [7] uvádí příklad, kdy je součástí systému databázový server, který se přibližuje stavu přetížení. Tým administrátorů o tom nemůže do poslední chvíle vědět. Jedinou možností při reaktivním monitorování je neustále sledovat graf zatížení a v pravou chvíli zasáhnout. Což je v prostředí mnoha zařízení přinejmenším nepředstavitelné. [7]

Tento typ přístupu k monitorování je typický pro malé až střední podniky s vyčleněným servisním týmem. Také se vyznačuje velkou mírou nevyřízených upozornění na chybné stavy a stavy ohrožující funkčnost systémů. Kontroly upozornění jsou obvykle posledním krokem před nasazováním nové verze aplikace nebo infrastruktury. [4]

2.2.3 Proaktivní (aktivní)

Posledním přístupem k monitorování je podle Turnbulla [4] proaktivní monitorování. Podle Lawrence [8] tento přístup k monitorování umožňuje servisním týmům pochopit, jak služby pracují spolu s identifikací možných rizik 7 dní v týdnu, 24 hodin denně. Společnost Rapid7 [7] uvádí, že proaktivní monitorování je nejefektivnějším přístupem k monitorování.

Proaktivní monitorování je přístup, který využívá algoritmy inteligentního rozpoznávání pro získání aktuálních dat a jejich další využívání pro předvídání

budoucích stavů. Proaktivní monitorování může být aplikováno na aplikace, počítačové sítě, počítače nebo datová centra.[7, 8]

Tento přístup monitorování dává podle Lawrence [8] možnost upozornit servisní tým na nenadálou situaci v sytému, i když jsou mimo své pracoviště. Proaktivní monitorování aplikací tedy umožňuje IT týmům odhalit příčinu chyby daleko dříve, než ovlivní funkčnost aplikace nebo sytému. Monitorovací systémy mohou také poskytovat informace o výkonu webové stránky, databáze a analytických nástrojů.

Turnbull [4] dále dodává, že proaktivní monitorování poskytuje data, která měří kvalitu služeb a poskytuje IT oddělením data pomáhající k odůvodnění rozpočtů, nákladů projektů a vytváření nových projektů. Tento přístup je velmi typický pro webově-orientované společnosti.

Monitorování je stále do značné míry řízeno, ale odpovědnost za zajištění nových služeb může být delegována na vývojáře aplikací. Software vyvíjený firmou není považován za kompletní nebo připravený k nasazení bez provedeného monitorování. [4]

2.3 Metriky

Monitorování se podle Liguse [3] skládá z nahrávání a následné analýzy velkého množství číselných vstupů, nesoucích informaci o současném stavu zařízení. Tyto vstupy mají své další vlastnosti. Jako například čas, kdy byl vstup zaznamenán v systému. Další vlastností může být jednotka, kterou číselný záznam reprezentuje. Tato skupina údajů se nazývá **metrika** (z angl. metric).

Metrika je podle Liguse [3] datová struktura přizpůsobená k uložení a následnému získání číselných hodnot. Turnbull [4] tvrdí, že metrika je nejlehčeji pochopitelná část jakékoli monitorovací architerktury. Podle Ellingwooda [9] jsou metriky základními hodnotami pro pochopení historických trendů, měření změn ve výkonu nebo chybovosti. Příklad metriky může být zaznamenávání návštěvnosti webových stránek. Podle webu Stackify je měření výkonu důležité pro všechny typy aplikací [10].

Jednou z nejpoužívanějších metod na vizualizaci metrik je graf s osami x, y. V grafu reprezentuje osa Y jednotlivé naměřené metriky a osa X je určená pro čas.

V praxi se zobrazuje graf jen pro daný časový interval. Nejčastěji od 5 minut do 1 hodiny. Limit ale není stanoven. V následujícím obrázku je vidět graf využití procesoru během posledních 24 hodin. [4]

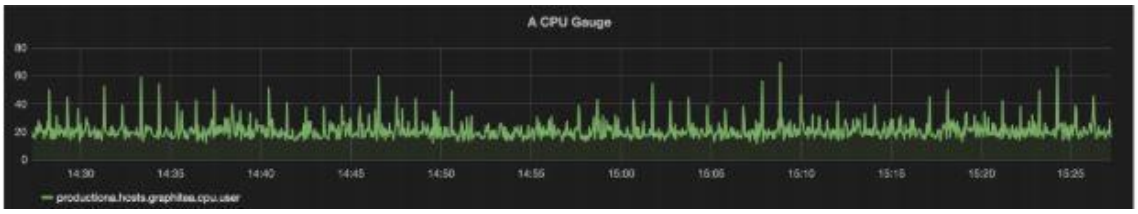
Obrázek 1: Graf využití procesoru.
Zdroj: [4]

Podle typu dat, která metrika sbírá, Turnbull [4] rozlišuje několik typů metrik:

a) Meřítko (gauge)
b) Počítadlo (counter)
c) Časoměřič (timer)

2.3.1 Meřítko (gauge)

První, a podle Turnbulla [4] nejpoužívanější metrikou je meřítko. Předpokladem pro použití je, že se hodnoty v průběhu času mění. Metrika je tedy hodnotou v bodě času. Příkladem může být využití disku nebo počet návštěvníků na stránce v jednu chvíli [4]. V obrázku níže je zobrazena reprezentace měřítka. V tomto obrázku je možné vidět využití procesoru za poslední hodinu a půl.



Obrázek 2: Příklad využití měřítka.
Zdroj: [4]

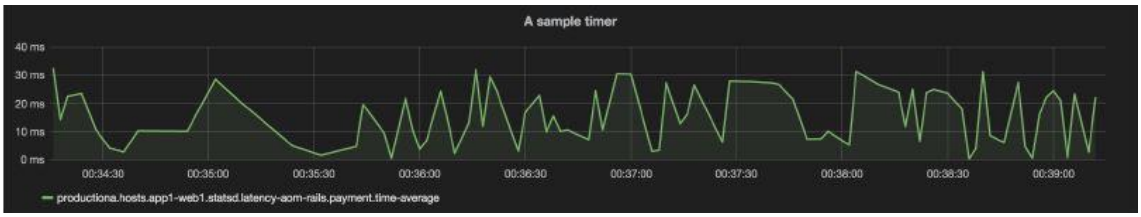
2.3.2 Počítadlo (counter)

Dalším typem metriky je počítadlo. Tento typ metriky je reprezentován číselnými záznamy, které během času narůstají. Občas může dojít k vynulování počítadla. Příkladem může být návštěvnost webových stránek nebo počet provedených nákupů v internetovém obchodu [4]. Obrázek níže reprezentuje příklad počítadla.

Obrázek 3: Počítadlo a jeho využití
Zdroj: [4]

2.3.3 Časomíra (timer)

Jako poslední typ metriky Turnbull [4] uvádí časomíru. Tato metrika reprezentuje dobu, po kterou byla vykonávána nějaká systémová procedura nebo metoda. Časomíra může začít při zavolání dané metody a skončit po jejím dokončení. Na obrázku je zobrazeno využití časomíry pro vykonání metody pro platbu.



Obrázek 4: Využití časomíry
Zdroj: [4]

2.3.4 Agregace metrik

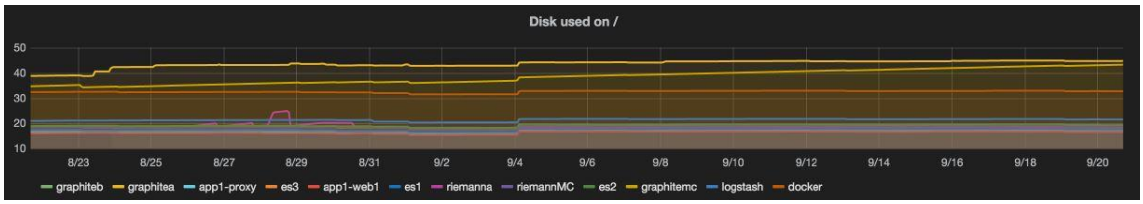
Podle Turnbulla [4] většinou nestačí zobrazit pouze jednu metriku. Místo toho se na metriky aplikují různé matematické operace a metriky se zobrazují najednou do jednoho grafu.

Turnbull [4] dále uvádí příklady matematických operací nad metrikami.

- Sčítání – sečtení všech metrik za daný časový interval

- Průměr – průměrná hodnota za časový interval
- Medián – střední hodnota v posloupnosti čísel
- Míra změn – ukazuje stupeň změny dat z časového intervalu

Metriky jsou velice užitečné, jelikož poskytují přehled o chování a stavu systému, zejména pokud se agregují neboli zpracovávají souhrnně. Je jednodušší některé chyby vidět v souhrnné metrice, nežli je složitě vyhledávat v jednotlivých metrikách. V následujícím obrázku je zobrazeno využití disku na všech monitorovaných serverech po dobu 30 dní, což urychluje práci a není tedy nutno kontrolovat vytížení v jednotlivých grafech pro každý server zvlášť. [4, 9]



Obrázek 5: Agregované metriky
Zdroj: [4]

2.4 Alerting

Ligus [3] uvádí alerting jako schopnost monitorovacího systému upozornit administrátora o událostech, které reprezentují změnu stavu v systému. Upozornění se nazývá **alert**. Alert je jednoduchá zpráva, doručována nejčastěji pomocí emailu nebo SMS. Alert je doručován určitému pracovníkovi, který se stará o jeho další vyřešení.

Alerting je část monitorovacího systému, která je zodpovědná za akce na základě změn hodnot v již zmíněných **metrikách**. Zatímco monitorovací systém je velice užitečný pro interpretaci a možnosti mít přehled o systému, kompletní monitorovací systém dovoluje administrátorům odejít od systému. Alerty nám umožňují předem definovat situace, které by se měly aktivně sledovat. Systém tedy v čase sleduje a reaguje na měnící se podmínky. [9]

2.4.1 Alert

Definice alertu podle Liguse: [3] [2]*„Alert je upozorňující zpráva informující o změně stavu v systému typicky oznamující potencionální problém.“* Alert přináší způsob, kterým lze informovat uživatele o vážném problému uvnitř systému. Slouží také pro ujištění, že po odstranění chyby systém pracuje správně [11].

Podle naléhavosti akce rozdělujeme alerty do několika skupin. Některé potřebují okamžitou akci a jiné pouze upozorňují na potenciální nutnost zásahu v budoucnosti [12].

I. **Alerty jako záznamy** – Tyto alerty vyžadují velmi nízkou součinnost operátora nebo administrátora. Příkladem může být například nižší odezva z internetové stránky, která je z pohledu uživatele zanedbatelná, ale vyšší než obvykle. [12]
II. **Alerty jako upozornění** – Další úroveň alertů pro problémy, které vyžadují zásah do systému. Zásah do systému však není nutný hned po doručení alertu. Obvykle se jedná o docházející místo na disku. Tento problém lze řešit v průběhu několika dní po doručení alertu. [12]
III. **Alerty jako „pages“** – Název vznikl od zařízení pager, tento typ vyžaduje okamžitou reakci administrátora. Jde o cokoliv, co by ohrožovalo funkčnost sytému. Rozdíl mezi alertem potřebujícím urgentní zásah a tím, který se dá odložit, by měl být definován v interních pravidlech firmy, pro kterou se monitorování provádí. [12]

2.4.2 Identifikace problému

Ligus [3] rozděluje potencionální příčiny chyb v systému do čtyř skupin:

- **Nedostupnost zdroje** – Nejčastější problém vyskytující se v produkčním prostředí. Hlavním příznakem je částečná nedostupnost nebo zvýšená odezva. Prevencí těmto problémům je včasná detekce a zásah. [3]

[2] A notification message informing about a change of state, typically signifying a potential problem.

- **Softwarové problémy** – Softwarové chyby jsou přítomny všude. Jejich odstranění se neprovede samo, mělo by s nimi být jednáno jako se součástí denní rutiny. Když monitorovací systém nahlásí operátorovi softwarovou chybu, měl by být operátor oprávněn učinit opravu. Proto servisní týmy potřebují mít přístup k softwaru, aby mohly provádět rollbacky a případně roll-forward. [3]
- **Chyba v konfiguraci** – Chyby v konfiguraci jsou velice těžko programově odhalitelné. Příkladem viditelně špatné konfigurace může být, když webový prohlížeč označuje veškeré webové stránky, jako stránky se škodlivým obsahem. Konfigurace by měly být uchovávány ve verzích, aby bylo možné nahrát starší verzi při nefunkčnosti nové verze. [3]
- **Selhání hardwaru** – Tento typ selhání nenastává frekventovaně, ale mívá velký dopad a bývá těžko řešitelný. Hardwarové selhání může způsobit značné snížení výkonu nebo úplné odstavení zařízení. Vznik hardwarového selhání může být sledován skrze zprávy o chybách v operačním systému. Často ale dochází k nenadálému selhání. [3]

2.4.3 Rozdělení alertů podle metody zasílání

Po vyskytnutí chyby v systému se musí odeslat alert administrátorovi, aby věděl o chybném stavu a měl možnost, co nejdříve zareagovat a pokusit se chybu opravit. Podle typu doručení se rozdělují alerty do několika skupin. [3]

- **Email** – Nejčastější metoda pro zasílání alertů. Výhody jsou v rychlosti doručení, spolehlivosti a širokém použití v běžném životě. [3]
- **SMS** – Administrátor obdrží textovou zprávu s číslem chyby. Výhoda doručování pomocí SMS je v nízké ceně, rychlosti doručení a spolehlivosti. Oproti emailu je u SMS lepší pokrytí signálu. [3]
- **Telefonické upozornění** – Upozornění telefonicky vyžaduje automatický systém volání. Dále potřebuje rychlé potvrzení stavu alertu. Většinou existuje sada klíčových slov, které se používají. Největší výhodou oproti výše zmíněným metodám je eliminace času potřebného pro potvrzení doručení alertu. Přijmutí automatového hovoru trvá zhruba od 15 sekund

do 1 minuty. Přijmutí několika SMS zpráv najednou je ale pro administrátora příjemnější než vyřizování několika telefonátů. [3]

- **Instant messaging** – Pro zrychlení doručení zprávy o anomálii se v poslední době začíná využívat Instant messaging. Monitorovací nástroj tak může zaslat do skupinové konverzace zprávu o anomálii a celková reakční doba se zkrátí. [4]
- **Další typy** – Dalším typem může být například upozornění pomocí zvuku nebo blikajícího světla. Tato metoda se moc nepoužívá, jelikož administrátorovi nedává vědět, jaká chyba nastala. Jedná se pouze o metodu přitáhnutí pozornosti administrátora, že něco není v pořádku. [3]

Důležité je používat více způsobů doručování informací o chybách najednou. Například začít s upozorněním v systému, poté přejít k odeslání emailu, a nakonec zvolit metodu odesláním SMS zprávy. Poslání emailu je nejméně násilná forma zaslání informace o alertu a nikoho nevzbudí uprostřed noci. Proto se na běžné chyby neohrožující přímo chod aplikace používají převážně emaily. [12, 13]

2.5 Vlastnosti nástrojů

Bernard Golden [14] ve svém článku představuje klíčové vlastnosti monitorovacích nástrojů. Hlavním úkolem všech nástrojů je snižovat dobu, po kterou je aplikace nebo služba nedostupná. Tím nástroje pomáhají administrátorům při jejich práci a šetří peníze vynaložené na řešení problémů.

2.5.1 Klíčové vlastnosti nástrojů

Mezi základní vlastnosti, které by měl obsahovat každý nástroj podle typu monitorovaného systému, ke kterému je určen podle [14] patří:

1. **Klíčové vlastnosti nástrojů pro monitorování fyzických nebo virtuálních serverů –** Nástroj, který se stará o monitorování serverů by měl podle [14] pokrývat tyto oblasti:
    a. **Využití CPU –** Využití procesoru v procentech.
    b. **Využití disku –** Kolik je využito z lokálního úložiště, případně jakého výkonu dosahují jednotlivé disky.

c. **Zbývající místo na disku –** Nástroj by měl umožňovat zjistit, kolik procent z kapacity disku je již využito a kolik procent zbývá.
d. **Operační paměť –** Kolik operační paměti je využito.

2. **Klíčové vlastnosti nástrojů pro monitorování sítě –** Monitorování stavu sítě by se mělo zaměřit na tyto tři klíčové statistiky počítačové sítě [14]:
    a. **Aktuální použití šířky pásma –** Toto sledování se týká využití příchozí i odchozí šířky pásma serveru.
    b. **Počet odeslaných paketů –** Tímto je zajištěna celková kontrola šířky pásma. V případě, že je spotřeba šířky pásma vysoká, ale počet odeslaných paketů je nízký, obvykle to znamená problém, který je potřeba řešit.
    c. **Detekce chybných paketů –** Je důležité vědět, kolik chybných paketů se v síti vyskytuje, protože to může poukazovat na problém uvnitř sítě.
3. **Klíčové vlastnosti nástrojů pro monitorování sdíleného úložiště –** Tento typ monitoringu je zaměřen na sdílené úložiště a poskytování přístupu k datům. Proto se monitorování podle [14] potřebuje zaměřit na tyto aspekty:
    a. **Dostupnost diskového pole –** Souborové servery jsou sdílená zařízení a poskytují data mnoha aplikacím. Sledování dostupnosti je velice důležité k zajištění bezproblémového chodu serveru.
    b. **Stav jednotlivých svazků diskového pole –** Disková pole jsou složena ze spousty svazků. Monitorování dílčích svazků a logických oddílů může včas odhalit potencionální problémy.
    c. **Kapacita diskového pole –** Vyčerpání úložného prostoru je běžným problémem při nedostupnosti aplikace. Sledování použitého a volného místa v diskovém poli je základním požadavkem na monitorovací systém.

Všechny zmíněné kategorie se zaměřují jen na sledování chování hardwarových prvků systému, stejně důležitou součástí monitorovacího systému je schopnost monitorovat softwarovou část aplikace [14].

Podle Goldena [14] zahrnuje monitorování softwaru následující požadavky:

1. Monitorování softwarových komponentů, které tvoří infrastrukturu společnosti. Při používání virtualizace je nutné sledovat chování HyperVisoru.
2. Schopnost monitorovat produkty třetích stran. Jako například monitorování relačních databází (MySQL, Oracle, …)
3. Sada nástrojů, umožňující sledování vlastních aplikací a dalších metrik. Příkladem může být průměrná doba odezvy funkcí aplikace, běžící procesy, stavy front atp.

2.5.2 Rozšiřující vlastnosti monitorovacích nástrojů

Klíčové vlastnosti, které byly zmíněny v minulé podkapitole, postačí na sledování aplikací malého rozsahu. V praxi ale topologie aplikací vyžaduje rozsáhlejší funkcionality. Proto [14] uvádí následující seznam funkcionalit:

1. **Agregované/oddělené zobrazení výkonu –** Aplikační vrstva se může skládat z více různých virtuálních strojů, kdy všechny zastávají stejnou funkci. Je užitečné vidět všechny virtuální stroje, avšak při výskytu potencionálního problému, je lepší vidět jednotlivé stroje odděleně. [14]
2. **Logování –** Aplikace se skládají z desítek až stovek softwarových částí. Sledování problémů s výkonem je náročné. Logování všech záznamů z komponentů může pomoci při hledání hlavní příčiny výskytu chyby v systému. [14]
3. **Alerting –** I když jsou alerty obsaženy v základních funkcích monitorovacího systému, nástroje s rozšířenou funkcionalitou mohou definovat thresholdy a logiku alertů. Jeden druh problému upozorní na chybu v síťové skupině, zatímco jiný problém bude detekován ve skupině serverové. [14]

4. **Konfigurovatelný dashboard –** Grafický výstup je velice užitečný pro rychlé zkontrolování stavu systému administrátorem. Konfigurovatelný dashboard umožňuje definovat jiný pohled pro rozdílná oddělení ve společnosti podle jejich oprávnění v systému. [14]
5. **Uložení metrik v operační paměti –** Díky obrovskému množství dat z webových aplikací a potřebě okamžitě reagovat na problémy s výkonem, může načítání dat z disku trvat nepřijatelně dlouho. Pro rychlejší odezvu nabízejí některé nástroje uložení metrik v operační paměti. [14]

3 Praktická část

3.1 Požadavky na monitorovací nástroje

V následující kapitole autor představí požadavky na monitorovací nástroje, které považuje jako důležité při vybírání monitorovacího nástroje.

3.1.1 Oblíbenost nástroje

Autor provedl průzkum článků porovnávajících monitorovací nástroje. Oblíbenost a používanost nástroje je jedním z nejdůležitějších parametrů při výběru. Není ideální se začít učit nástroj, který není ostatními administrátory používaný, a tudíž nikde nelze najít pomoc při řešení problémů. Společnost ITTSystems [15] zveřejnila článek, ve kterém rozebírá téma top open source nástrojů pro monitorování systému na platformách Windows a Linux. V tomto článku doporučuje nástroje Nagios, Zabbix, Incinga, Libre NMS a Pandora FMS. Dalším článkem od DNSstuff [16] jsou doporučovány podobné nástroje. Andrew Tabona v květnu 2018 publikoval článek, kde popisoval top 20 nástrojů pro implementaci monitorování [17]. Ke konci roku 2018 James Hayden vydal na svém blogu příspěvek s 20 top monitorovacími nástroji [18]. Autor na základě těchto článků vybral pro účely této bakalářské práce nástroje Zabbix a Nagios Core. Oba nástroje jsou open-source a dostupné pod licencí GPL.

3.1.2 Free licence

Požadavkem společnosti Unicorn, která specifikovala zadání této práce, bylo, že používané nástroje nebudou mít placenou licenci. Autor se tedy zaměřil pouze na nástroje s bezplatnou licencí.

3.1.3 Webové rozhraní

Používaný nástroj by měl obsahovat alespoň jednoduché webové rozhraní pro zkontrolování stavu všech klientských stanic a stavů nastavených kontrol u klientských stanic.

3.1.4 Vytváření šablon a skupin pro služby a klientské stanice

Pro rychlejší konfiguraci klientských stanic a monitorovaných služeb je výhodou možnost šablonování. Nástroj umožňuje přidávat klientské stanice do skupin a těmto skupinám nastavit požadované vlastnosti na jednom místě. Tím se ušetří čas potřebný na konfiguraci.

3.1.5 Využití agentů

Jedním ze způsobů monitorovaní je použití tzv. „agentů“. O agentech již bylo zmíněno v teoretické části. Na klientské stanici je nainstalovaný malý „program“, se kterým komunikuje monitorovací server a požaduje po agentovi provedení určité akce na klientské stanici a vrácení zpět serveru. Poté už server vyhodnotí, zda je vrácená hodnota v normálních hodnotách, nebo zdali je potřeba upozornit administrátora.

3.1.6 Stáří poslední verze nástroje

Poslední verze nástroje by neměla být starší než 1 rok. S rychlostí, jak se vyvíjí nové technologie by byla ztráta času pracovat s nástrojem, který je zastaralý, případně je jeho vývoj už nadobro ukončen.

3.2 Kontroly pro monitorovaní

Monitorovaná aplikace, kterou vyvíjí a dodává firma Unicorn pracuje převážně s databází. Většina sledovaných metrik byla zaměřena na kontrolu stavu databázových schémat. Kontroly aplikace jsou rozděleny do skupin. První skupina je společná pro všechny servery a jedná se o kontrolu hardwaru serveru a kontrolu operačního systému. Ve druhé skupině se autor zaměřil na kontroly klíčových schémat a tabulek v databázi, aby byl zajištěn hladký chod aplikace.

3.2.1 Obecná kontrola společná pro všechny servery

Podkapitola obsahuje seznam metrik sledovaných skrze všechny servery v infrastruktuře aplikace.

- Využití CPU
- Využití operační paměti

- Zaplnění místa na oddílech serveru
- Celkový počet běžících procesů
- Využití SWAP operační paměti

3.2.2 Kontrola databáze

Jak už bylo zmíněno výše, velký důraz byl kladen na kontrolu databáze. Níže je seznam kontrol prováděných v databázích

- Velikost databázového schéma gm_midterm
- Velikost databázového schéma gm_operational
- Počet aktuálně běžících procesů
- Počet čekajících procesů
- Kontrola, zda běží služba MySQL

3.3 Vybrané nástroje

V tabulce níže je souhrn požadavků kladených na nástroje a informace, zda nástroje plní a případně v jaké míře.

Nástroj	**Webové rozhraní**	**Vytváření šablon**	**Použití agentů**	**Poslední vezre**
Nagios Core	Ano, pouze pro kontrolu klientských stanic	Ano	Ano	V 4.4.3 15. 01. 2019
Zabbix	Ano, použito pro veškerou správu serveru a nastavení potřebných služeb	Ano	Ano	V 4.0 1. 10. 2018

Tabulka 1 Požadavky na nástroje
Zdroj: vlastní zpracování

3.3.1 Nagios

Monitorovací systém Nagios je dostupný na trhu již od roku 1996. Je jedním z nejstarších nástrojů vůbec. Poslední verze (4.4.3) je z ledna 2019. V roce 2016 překonal Nagios 7 500 000 stažení z portálu SourceForge [19]. Nagios Core je založen na architektuře klient/server. Pro funkčnost je zapotřebí nainstalovat serverovou část a na každém monitorovaném serveru (klientské stancici) poté nainstalovat NRPE (Nagios Remote Plugin Executor). Instalací všeho potřebného se zabývá následující kapitola.

3.3.1.1 Instalace serveru Nagios Core

Tato podkapitola se zabývá instalací nástroje Nagios Core ve verzi 4.4.3. Autor postupoval podle návodu[3] na oficiálních stránkách výrobce nástroje. Oficiální návod obsahuje části pro odlišné distribuce operačního systému Linux. Monitorovací server poskytnutý pro vypracování této bakalářské práce používá operační systém Linux v distribuci Red Hat Enterprise Linux. Autor tedy použil část pro tuto distribuci.

V prvním kroku je třeba nastavit Security-Enhanced Linux. To lze udělat více způsoby. Jedním je úplné vypnutí nebo nastavení do permissive módu. Dále je potřeba nainstalovat balíčky potřebné k instalaci nástroje. Textové pole níže obsahuje všechny příkazy k provedení akcí.

```
# Vypnutí SELinux
sed -i 's/SELINUX=.*/SELINUX=disabled/g' /etc/selinux/config
# Odlišný způsob vypnutí SELinux
setenforce 0
# Instalace potřebných služeb
yum install -y gcc glibc glibc-common wget unzip httpd php gd
gd-devel perl postfix
```

Konfigurace 1 Nastavení SELinux

Podle návodu je potřeba stáhnout zdrojové soubory Nagios Core, a poté je ručně zkompilovat na serveru. Nástroj Nagios Core je psán v jazyce C. Po zkompilování

[3] https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source-96.html#RHEL

zdrojových souborů je nutné vytvořit na serveru uživatele a skupinu nagios. Do skupiny nagios také přidat uživatele apache. Po dokončení předchozích kroků lze překročit k instalaci všech souborů nástroje Nagios Core. V tabulce níže jsou vypsány potřebné příkazy pro provedení zmiňovaných akcí.

```
# Stažení zdrojových souborů
cd /tmp
wget -O nagioscore.tar.gz
https://github.com/NagiosEnterprises/nagioscore/archive/nagios-
4.4.3.tar.gz
tar xzf nagioscore.tar.gz
# Kompilace
/tmp/nagioscore-nagios-4.4.3/
./configure
make allace zdrojových souborů
# Vytvoření uživatele a skupiny „nagios“
make install-groups-users
# Přidání uživatele apache do skupiny nagios
usermod -a -G nagios apache
# Instalace souborů
make install
```

Konfigurace 2 Zdrojové soubory a přidání uživatele nagios

Příkazy níže instalují službu nagios včetně souborů pro démona a nastavení spouštění služby httpd po startu systému.

```
# Instalace démona
make install-daemoninit
# Zapnutí služby nagios.service po startu systému
systemctl enable httpd.service
```

Konfigurace 3 Instalace démona

Po dokončení instalace je potřeba nainstalovat soubor s externími příkazy, vzorovou konfiguraci a také konfigurační soubory pro webový server Apache. Nagios potřebuje jakékoli konfigurační soubory, aby mohl být spuštěn. Proto je nutné instalovat výše zmíněné soubory.

```
make install-commandmode
make install-config
make install-webconf
```

Konfigurace 4 Instalace webového serveru a výchozí konfigurace pro Nagios

Po dokončení veškeré instalace přichází na řadu konfigurace Firewall pro správnou funkčnost webového serveru. To znamená přidat firewall pravidlo pro

přijímání paketů na port 80. Na serveru byla použita pro správu firewallových pravidel služba iptables.

```
iptables -I INPUT -p tcp --destination-port 80 -j ACCEPT
service iptables save
ip6tables -I INPUT -p tcp --destination-port 80 -j ACCEPT
service ip6tables save
```

Konfigurace 5 Nastavení Firewall pravidel

Nagios obsahuje jednoduché webové rozhraní pro správu a kontrolu všech nastavených klientských stanic a jejich definovaných metrik. Pro přihlášení do webového rozhraní je třeba vytvořit soubor ve složce /usr/local/nagios/etc. Příkazem níže se vytvoří uživatel *nagiosadmin* a po zadání příkazu je uživatel vyzván k zadání hesla pro uživatele *nagiosadmin*.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Konfigurace 6 Vytvoření uživatele pro webové rozhraní

Pro ověření správnosti konfigurace lze použít příkaz na kontrolu konfiguračních souborů před spuštěním serveru. Použitím příkazu níže se na terminál vypíše počet nastavených klientských stanic, objektů a konfiguračních souborů. V případě výskytu chyby v konfiguraci, která vede k chybě při zapnutí Nagios Core serveru, je toto velmi užitečný nástroj na hledání chyb v konfiguračních souborech, kterých se při konfiguraci používá velké množství. Tento nástroj při nalezení chyby vypíše soubor, ve kterém se chyba vyskytuje včetně čísla řádku. Umí i detekovat závislosti při konfiguraci. Což může méně zkušeným uživatelům ušetřit čas při opravě konfigurace.

```
/usr/local/nagios/bin/nagios -v
/usr/local/nagios/etc/nagios.cfg
```

Konfigurace 7 Příkazy pro kontrolu konfigurace

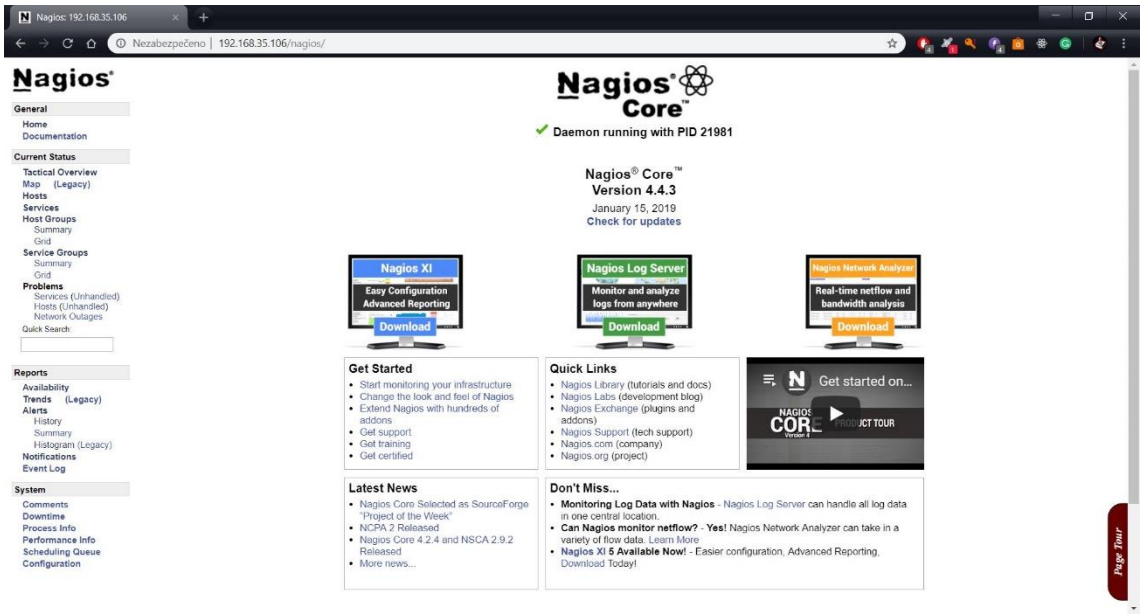
Nyní lze přejít k finálnímu testu funkčnosti Nagios Core. Příkazy níže je třeba spustit službu httpd.service a nagios.service.

```
systemctl start httpd.service
systemctl start nagios.service
```

Konfigurace 8 Spuštění služeb

Nyní je potřeba pouze přistoupit k webovému rozhraní přes prohlížeč a otestovat funkčnost webového rozhraní. Po zadání URL adresy do adresního řádku v prohlížeči ve tvaru <IP_ADRESA_SERVERU>/nagios/ a zadání uživatelského

jména a hesla, které bylo nastaveno o 3 kroky dříve, se zobrazí hlavní strana webového rozhraní pro Nagios Core.



Obrázek 6 Webové rozhraní Nagios Core
Zdroj: vlastní zpracování

Dosavadní instalace se týkala pouze jádra monitorovacího nástroje Nagios Core. V tuto chvíli Nagios neobsahuje žádné pluginy pro monitorování. Ty se musí doinstalovat zvlášť. Příkazy níže se nainstalují potřebné soubory a služby, které doplňky Nagios vyžadují ke své instalaci.

```
cd /tmp
wget https://dl.fedoraproject.org/pub/epel/epel-release-
latest-7.noarch.rpm
rpm -ihv epel-release-latest-7.noarch.rpm
subscription-manager repos --enable=rhel-7-server-optional-
rpms
yum install -y gcc glibc glibc-common make gettext automake
autoconf wget openssl-devel net-snmp net-snmp-utils
yum install -y perl-Net-SNMP
```

Konfigurace 9 Prerekvizity pro Nagios Plugins

Po úspěšném instalování lze přejít ke stažení zdrojových souborů a k samotné instalaci pluginů. Některé z pluginů vyžadují instalaci dalších balíčků pro svou funkčnost. Při monitorování bude využíváno sledování výstupů z relační databáze

MySQL, je proto nutno nainstalovat balíčky pro tuto databázi. Soubory .rpm je potřeba stáhnout z oficiálního repozitáře[4] MySQL. Použití parametrů –with-nagios-user a –with-nagios-group se postará o změnu vlastníka a skupiny u všech nově vytvořených pluginů pro Nagios. Tím se dá předejít problémům s právy k souborům v budoucnosti.

```
# Instalace funkcí k monitorování MySQL databáze
yum install mysql-community-devel-5.7.21-1.el7.x86_64.rpm
yum install mysql-community-libs-5.7.21-1.el7.x86_64.rpm
# Stažení zdrojových souborů
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz
https://github.com/nagios-plugins/nagios-
plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
# Instalace Nagios Plugins
cd /tmp/nagios-plugins-release-2.2.1/
./configure --with-nagios-user=nagios --with-nagios-
group=nagios --enable-perl-modules
make
make install
```

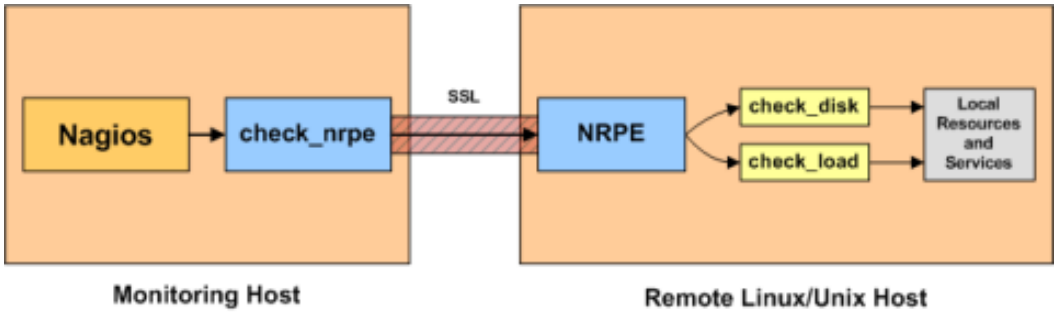
Konfigurace 10 Stažení a instalace Nagios Plugins

Tímto je instalace monitorovacího serveru ukončena a lze přejít k instalaci NRPE na všechny klientské stanice.

3.3.1.2 NRPE

Jak již bylo zmíněno dříve, Nagios používá na svých klientských stanicích službu NRPE (Nagios Remote Plugin Executor). Tato služba se využívá na stanicích s operačním systémem Linux. Na obrázku níže je vidět schéma komunikace mezi serverem a klientskou stanicí. Nagios Server přímo nevykonává akce na klientovi, ale v intervalech se dotazuje NRPE o provedení akce a vrácení výsledku zpět. Nagios Server komunikuje s NRPE na klientské stanici pomocí pluginu check_nrpe a portu 5666. Plugin je umístěn v **/usr/local/nagios/libexec**.

[4] https://repo.mysql.com/yum/mysql-5.7-community/el/7/x86_64/



Obrázek 7 NRPE
Zdroj: https://bit.ly/2EzEmNr

3.3.1.3 Instalace NRPE

Jak již bylo zmíněno dříve, na všech monitorovacích stanicích je třeba nainstalovat službu NRPE, včetně balíčku Nagois Plugins, aby služba NRPE mohla využívat těchto pluginů. Níže je probrána instalace služby NRPE. Autor postupval podle oficiálního návodu[5].

Prvním krokem je instalalace prerekvizit a stažení zdrojových souborů stejně jako u Nagios Core serveru.

```
# Instalace prerekvizit
yum install -y gcc glibc glibc-common openssl openssl-devel
perl wget
# Stažení zdrojových souborů
cd /tmp
wget --no-check-certificate -O nrpe.tar.gz
https://github.com/NagiosEnterprises/nrpe/archive/nrpe-
3.2.1.tar.gz
tar xzf nrpe.tar.gz
```

Konfigurace 11 Stažení NRPE

Dalším krokem je kompilace zdrojových souborů, vytvoření uživatele a skupiny nagios, instalace binárních souborů, instalace konfiguračních souborů pro NRPE, přidání NRPE do */etc/services* a instalace souborů služby.

```
# Kompilace
cd /tmp/nrpe-nrpe-3.2.1/
./configure --enable-command-args
```

[5] https://support.nagios.com/kb/article/nrpe-how-to-install-nrpe-v3-from-source-515.html#RHEL

```
make all
# Vytvoření uživatele a skupiny
make install-groups-users
# Instalace binárních souborů a konfiguračních souborů
make install
make install-config
# Upravení /etc/services
echo >> /etc/services
echo '# Nagios services' >> /etc/services
echo 'nrpe    5666/tcp' >> /etc/services
# Instalace souborů pro službu a nastavení spuštění po startu
systému
make install-init
systemctl enable nrpe.service
```

Konfigurace 12 Instalace služby NRPE

Posledním krokem instalace NRPE je přidání pravidel do IPTABLES (firewall) a úprava konfiguračního souboru v /usr/local/nagios/etc/nrpe.cfg, kde se nastaví IP adresa Nagios Core serveru. Parametr *dont_blame_nrpe=1* umožní rozsáhlejší konfiguraci NRPE, protože je možno posílat přes check_nrpe v argumentech příkazy, které provádějí jednotlivé kontroly na serverech.

```
# Nastavení IPTABLES
iptables -I INPUT -p tcp --destination-port 5666 -j ACCEPT
service iptables save
ip6tables -I INPUT -p tcp --destination-port 5666 -j ACCEPT
service ip6tables save
# Úprava konfiguračního souboru /usr/local/nagios/etc/nrpe.cfg
allowed_hosts=127.0.0.1,<IP_ADRESA_NAGIOS_SERVERU>
dont_blame_nrpe=1
```

Konfigurace 13 Nastavení Firewallu a konfiguračního souboru

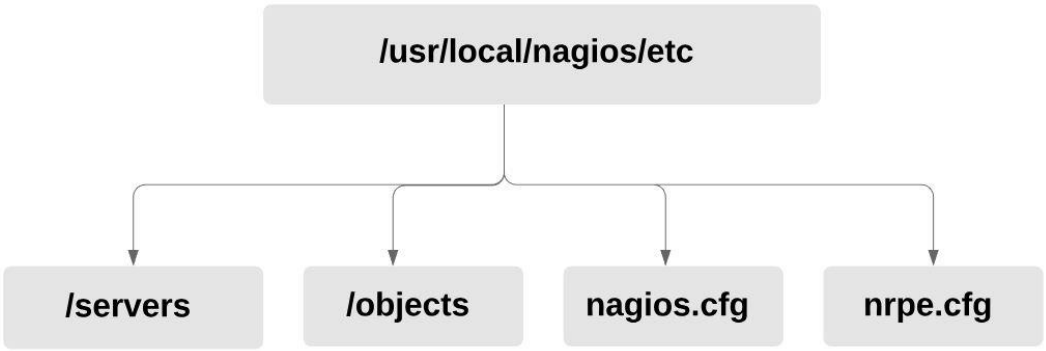
Nyní je instalace NRPE kompletní a lze přejít k instalaci Nagios Plugins stejně jako u monitorovacího serveru. Na databázových serverech je potřeba při instalaci pluginů také doinstalovat balíčky *mysql-community-devel* a *mysql-community-libs*.

3.3.1.4 Konfigurace

Tato kapitola se zabývá konfigurací monitorovacího nástroje Nagios. Ten používá několik hlavních konfiguračních souborů. Při konfiguraci autor používal online dostupnou dokumentaci nástroje Nagios. Pro server je použit soubor *nagios.cfg* v */usr/local/nagios/etc/* na Nagios serveru. V něm se konfigurují složky,

které nagios používá a ve kterých hledá další konfigurační soubory. Dále se pro NRPE používá soubor *nrpe.cfg* v */usr/local/nagios/etc/* na klientské stanici. V tomto souboru se konfigurují příkazy, které se poté volají přes plugin *check_nrpe* z monitorovacího serveru. [20]

Dalším důležitým konfiguračním souborem je *commands.cfg,* umístěný ve složce *objects*. Tento soubor se, jak již název napovídá, používá pro konfiguraci příkazů například pro notifikace nebo přidání příkazů pro Nagios Plugins. Do složky *servers* se ukládají soubory s konfiguracemi hostitelských stanic a sledovaných služeb na těchto stanicích. Nagios používá konfigurace pomocí definování objektů. Pro lepší představení je níže diagram popisující adresářovou strukturu nástroje Nagios.



Obrázek 8 Nagios adresářová struktura
Zdroj: vlastní zpracování

3.3.1.5 Založení klientské stanice

Jak již bylo zmíněno, konfigurace služeb, klientských stanic, kontaktů, skupin a všeho dalšího probíhá přes definování objektů. V ukázce níže je vidět, jak lze nakonfigurovat klientskou stanici Aplikačního serveru. Vlastnost *use* znamená použití šablony linux-server, která je uložena v souboru */objects/templates.cfg*. Direktiva *max_check_attempts* určuje kolikrát se Nagios server dotáže agenta po doručení jiného stavu než je OK. Podobně direktiva *notification_interval* nastavuje prodlevu mezi prvním odesláním zprávy o problému a dalším odesláním notifikace, pokud problém stále přetrvává. Po upravení konfiguračních souborů na Nagios serveru nebo NRPE je nutné dané zařízení restartovat.

```
define host {
```

```
    use                     linux-server
    host_name               app_server
    address                 192.168.35.60
    max_check_attempts           5
    check_period     24x7
    notification_interval  30
}
```

Konfigurace 14 Založení klientské stanice

3.3.1.6 Založení skupiny a definování metrik pro skupiny

Monitorované metriky na klientských stanicích se zakládají jako tzv. „services“ (služby, metriky). Opět se používá objektová definice. Služby se zakládají ve stejném souboru jako klientské stanice. Tím je docíleno soudržnosti konfigurace. Tímto způsobem konfigurace by se musely společné metriky definovat pro každý server zvlášť v jeho vlastním konfiguračním souboru. Což by nebyl až takový problém. Problém by nastal v případě nutnosti změnit některou vlastnost. To by musel administrátor v každém souboru měnit stejné nastavení, což je dosti nepohodlné řešení.

Nagios proto obsahuje logické členění klientských stanic do skupin a disponuje možností přiřazovat definované metriky těmto skupinám. V textovém poli níže je ukázáno, jak se vytváří skupiny klientských stanic. Konfigurace probíhá v souboru */servers/host_groups.cfg*.

```
define hostgroup {
    hostgroup_name      all
    alias               all
    members             app_server, db1_server, db2_server
}
```

Konfigurace 15 Založení skupiny klientských stanic

Po vytvoření skupiny serverů lze přejít k definici samotných metrik. Pro dodržení soudržnosti konfigurace se tyto metriky nastavují v souboru */servers/all-servers.cfg*. Příklad jedné z metrik je ukázán v textovém poli níže. Direktiva *check_command* určuje, jaký příkaz bude zavolán pro kontrolu stavu metriky. V tuto chvíli se již využívá NRPE agenta. Zavoláním příkazu *check_nrpe* a předáním paramteru *Memory*, který je definován v *nrpe.cfg* souboru na klientské stanici. Direktiva *contact_group* definuje, jakým skupinám budou zasílány notifikace při změně stavu této metriky.

```
define service {
    use                     generic-service
    hostgroup_name          all
    service_description     Memory usage
    check_command           check_nrpe!Memory
    contact_groups          admin_email, admin_telegram
}
```

Konfigurace 16 Založení metriky

Metrika „Memory usage“ se odkazuje na příkaz Memory. Níže je ukázano, jak se příkazy vytvářejí. Stejná konfigurace se musí provést pro všechny další metriky.

```
command[Memory]=/usr/local/nagios/libexec/check_mem -w 85 -c
95
```

Konfigurace 17 Příkaz Memory

3.3.1.7 Integrace IM do Nagios Core nástroje

Pro notifikaci administrátora se využívá několika způsobů, jak už bylo řečeno v teoretické části. Autor prostudoval několik možných řešení, jak docílit požadovaného chování.

Jednou možností bylo použití aplikace WhatsApp. Po několik let existovala možnost, jak posílat z linuxového terminálu zprávy do WhatsApp. Toto bylo umožněno díky skriptům, které simulovaly odesílání zpráv, jako by byly odeslány ze zařízení Nokia S40. Tato možnost byla bohužel k 31. 1. 2019 zrušena. WhatsApp tehdy zrušil podporu zařízení Nokia S40.

Alternativní řešení je integrace platformy Telegram za použití skriptů psaných v jazyce Python. Autor postupoval podle návodu[6]. Po nainstalování Python na server, nainstalování potřebných knihoven a vytvoření BOTa pro Telegram, je zapotřebí stáhnout skript pro odesílání zpráv.

```
yum -y install python python-dateutil python-argparse python-
readline python-devel libevent-devel python-pip python-dev
build-essential
pip install twx.botapi

wget -O /usr/local/bin/nagios_telegram.py
```

[6] https://pommi.nethuis.nl/nagios-notifications-via-telegram/

```
https://raw.githubusercontent.com/pommi/telegram_nagios/master/telegram_nagios.py
chmod 755 /usr/local/bin/nagios_telegram.py
```

Konfigurace 18 Stažení skriptu pro Nagios Telegram

V souboru */objects/commands.cfg* je nutno přidat příkazy pro odesílání zpráv. Poté v */objects/contacts.cfg* doplnit do kontaktů direktivu pager, do které se nastaví identifikátor skupinového chatu nebo uživatele v aplikaci Telegram. Identifikační číslo uživatele nebo skupiny se získá zadáním URL adresy https://api.telegram.org/<BOT_KEY>/getUpdates do prohlížeče. Ve výstupu lze vidět JSON s parametry *chat* a poté *id*.

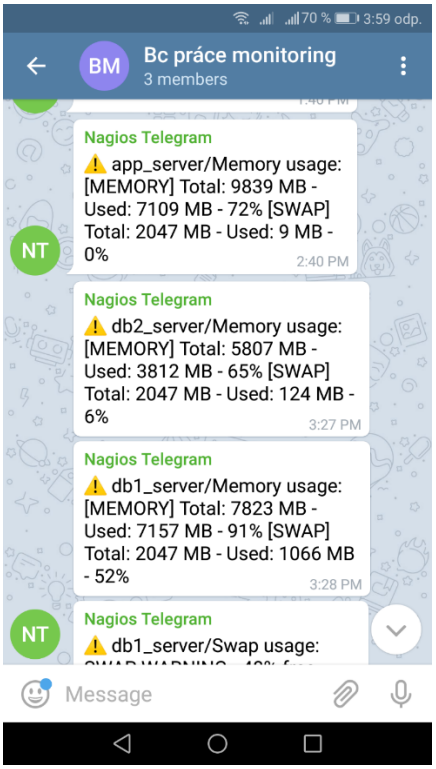
```
# Část výstupu z URL
"chat":{"id":-387682497

define command {
  command_name     notify-host-by-telegram
  command_line
/usr/local/nagios/libexec/telegram_nagios.py --token
681005111:AAHmDDM8FYlPcwR395a0AVsd0MdupvaXseA --object_type
host --contact "$CONTACTPAGER$" --notificationtype
"$NOTIFICATIONTYPE$" --hoststate "$HOSTSTATE$" --hostname
"$HOSTNAME$" --hostaddress "$HOSTADDRESS$" --output
"$HOSTOUTPUT$"
}
define command {
  command_name     notify-service-by-telegram
  command_line
/usr/local/nagios/libexec/telegram_nagios.py --token
681005111:AAHmDDM8FYlPcwR395a0AVsd0MdupvaXseA --object_type
service --contact "$CONTACTPAGER$" --notificationtype
"$NOTIFICATIONTYPE$" --servicestate "$SERVICESTATE$" --
hostname "$HOSTNAME$" --servicedesc "$SERVICEDESC$" --output
"$SERVICEOUTPUT$"
}
define contact {
    contact_name     admin_telegram
    use              generic-contact
    alias       Admin to get messages to Telegram
    pager       -387682497
    service_notification_commands     notify-service-by-
telegram
```

```
    host_notification_commands          notify-host-by-telegram
}
```

Konfigurace 19 Nastavení příkazu pro posílání zpráv Nagios

Na obrázku níže je vidět výsledná zpráva, která je odeslána do skupinové konverzace v Telegramu.



Obrázek 9 Nagios Telegram
Zdroj: vlastní zpracování

3.3.2 Zabbix

Monitorovací nástroj Zabbix vznikl v roce 1998 a od roku 2001 je nabízen pod licencí GPL. Verze použitá v této bakalářské práci je 4.0.1 (10/2018). Během zpracovávání této bakalářské práce byla vydána nová verze 4.2.0 (03/2019). Ke konfiguraci všech klientských stanic, reportů a notifikací se přistupuje přes webové rozhraní. Nástroj vytvořil Alexei Vladishev. Dále je vyvíjen a spravován firmou Zabbix SIA [15].

3.3.2.1 Instalace Zabbix serveru

Při instalaci autor postupoval podle oficiálního návodu[7] na stránkách výrobce nástroje Zabbix. Pro instalaci bylo využito souborů RPM (RPM Package Management).

Zabbix využívá ukládání konfiguraci v relační databázi. Oficiálně jsou podporovány MySQL a PostgreSQL. V prvním kroku je tedy potřeba stáhnout a nainstalovat MySQL server. Ke stažení balíčků bylo použito oficiálních zdrojů[8]. Po úspěšné instalaci lze přistoupit k přidání repozitáře pro Zabbix a k instalaci Zabbix serveru a webového rozhraní. Pro Zabbix server je dále nutno importovat výchozí databázi.

```
yum install mysql-community-server-8.0.15-1.el7.x86_64.rpm
rpm -ivh
https://repo.zabbix.com/zabbix/4.0/rhel/7/x86_64/zabbix-
release-4.0-1.el7.noarch.rpm
yum-config-manager --enable rhel-7-server-optional-rpms
yum install zabbix-server-mysql
yum install zabbix-web-mysql
# Import databáze
zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql
-uzabbix -p zabbix
```

Konfigurace 20 Instalace MySQL a Zabbix serveru

Tímto je instalace Zabbix serveru hotová a nyní je potřeba nastavit v souboru */etc/zabbix/zabbix_server.conf* parametry se jménem a heslem pro připojení do databáze.

```
vi /etc/zabbix/zabbix_server.conf
  DBHost=localhost
  DBName=zabbix
  DBUser=zabbix
  DBPassword=<password>
```

[7] https://www.zabbix.com/documentation/4.0/manual/installation/install_from_packages/rhel_centos

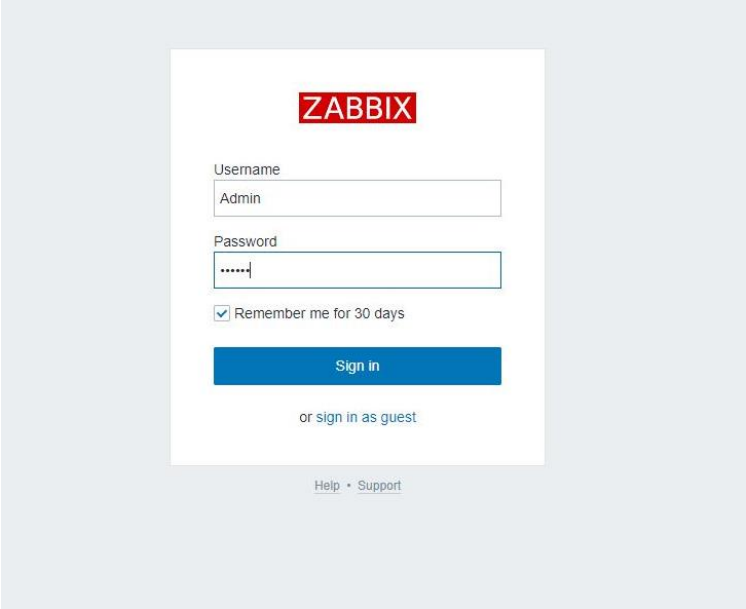
[8] https://dev.mysql.com/downloads/mysql/

Pro webové rozhraní je třeba nastavit v souboru */etc/httpd/conf.d/zabbix.conf* parametr *php_value date.timezone*. Další konfigurace probíhá již přes grafické rozhraní.

```
php_value date.timezone Europe/Prague
# Start zabbix serveru
systemctl start zabbix-server
```

Konfigurace 21 Nastavení časového pásma pro Zabbix

Ve webovém rozhraní je potřeba projít několika kroky instalace grafického rozhraní pro správu nástroje Zabbix. Prvním je kontrola všech instalovaných prerekvizit, nastavení údajů pro připojení do Zabbix databáze, vyplnění dalších informací o instalaci a instalace samotná. Po úspěšné instalaci se zobrazí přihlašovací stránka do Zabbix webového rozhraní. Výchozí uživatelské jméno a heslo je: *Admin, zabbix*.



Obrázek 10 Přihlášení Zabbix webové rozhraní
Zdroj: vlastní zpracování

3.3.2.2 Zabbix agent

Zabbix agent je software instalovaný na všech klientských stanicích. Stará se o sledování využití prostředků serverů (procesoru, paměti, disků, operačního systému atp). Zabbix server komunikuje s agentem skrze port 10050. Konfigurační soubor je umístěn v */etc/zabbix/zabbix_agent.conf*.

3.3.2.3 Instalace Zabbix agenta

Instalace je podobná serveru. Na začátku je třeba přidat repozitář pro Zabbix. Poté stačí nainstalovat a spustit službu zabbix-agent.

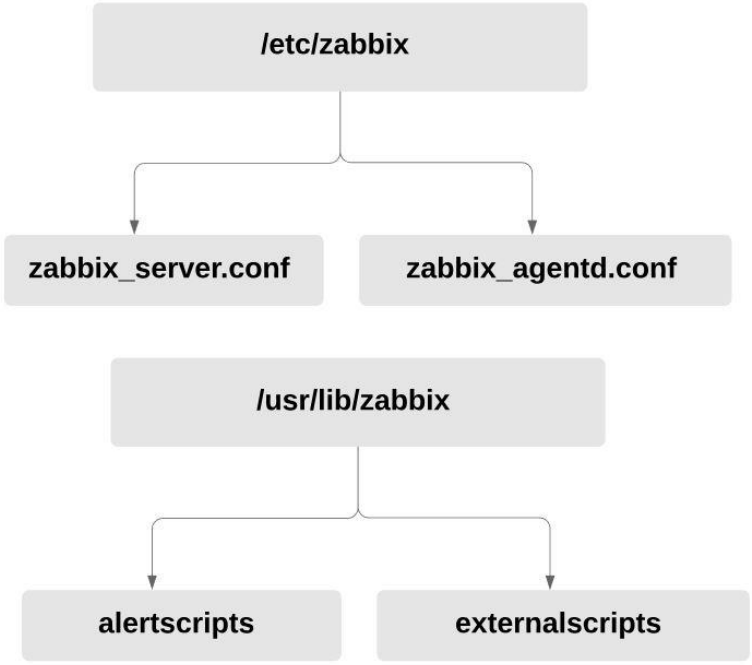
```
rpm -ivh
https://repo.zabbix.com/zabbix/4.0/rhel/7/x86_64/zabbix-
release-4.0-1.el7.noarch.rpm
yum install zabbix-agent
systemctl start zabbix-agent
```

Konfigurace 22 Instalace Zabbix agenta

Pro správnou funkčnost Zabbix agenta je potřeba přidat parametr Server=<IP_ADRESA_ZABBIX_SERVERU> do konfiguračního souboru zabbix_agent.conf ve složce /etc/zabbix/.

3.3.2.4 Konfigurace

Většina konfigurace probíhá přes uživatelsky přívětivé grafické rozhraní. Podrobnosti o nastavení Zabbix nástroje, včetně ukázek konfigurací, jsou dostupné v online dokumentaci [21]. Podle této dokumentace bylo postupováno při vytváření této bakalářské práce. Pro konfiguraci serveru je využit soubor zabbix_server.conf v adresáři /etc/zabbix. Pro agenta je využit soubor zabbix_agentd.conf ve stejné složce. Pro lepší představení je níže obrázek adresářové struktury konfiguračních a jiných souborů, které Zabbix využívá.



Obrázek 11 Zabbix adresářová struktura
Zdroj: vlastní zpracování

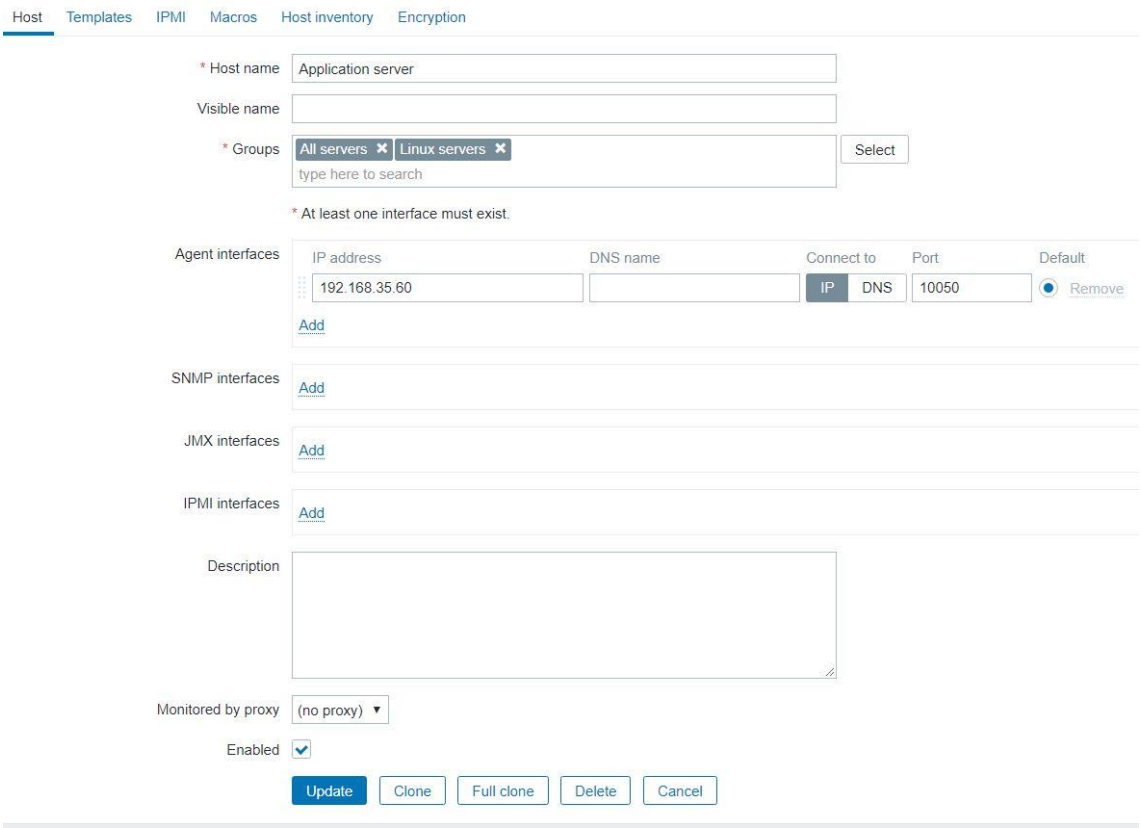
3.3.2.5 Vytvoření klientské stanice a přidání šablony

Zabbix obsahuje po instalaci širokou škálu přednastavených šablon pro monitorování různých zařízeních na různých platformách. Jednotlivé šablony jsou přehledně rozdělené do skupin. Zabbix také podporuje vytváření skupin pro klientské stanice. Bohužel díky databázové architektuře[9], přidělení šablony ke skupině stanic neovlivní stanice samotné a šablona nebude aplikována. Šablony se tedy musejí manuálně přidat ke každé stanici na záložce *Templates* při vytváření klientské stanice. Vytvoření klientské stanice se provádí v *Configuration -> Hosts -> Create Host.* Na této stránce se vyplní podrobnosti o zařízení. Název, IP adresa, popis a další.

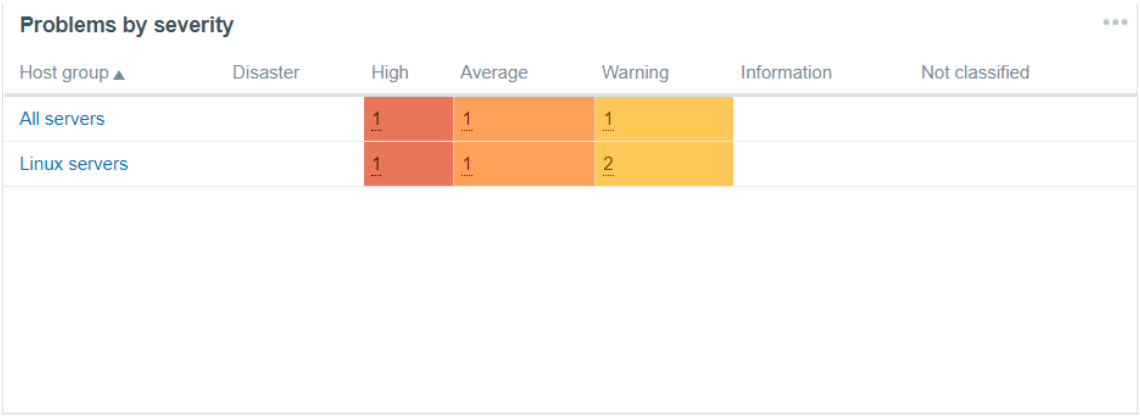
Povinným parametrem je i skupina, do které bude zařízení přidáno. Hlavním přínosem při další konfiguraci je v tom, že se administrátor nemusí probírat nekonečnými seznamy všech zařízení. Seznamy si zobrazí pouze pro určitou skupinu, a tím si ulehčí práci při hledání daného zařízení [22]. Problém může

[9] http://www.zabbixbook.com/2016/09/22/added-a-template-to-group-nothing-happened/

nastat při přidání jedné klientské stanice do více skupin, jelikož některé oznámení na hlavní stránce se zobrazují za celou skupinu zároveň [22]. V obrázku níže je ukázáno vytvoření klientské stanice aplikačního serveru. Další obrázek ukazuje popisovaný problém při přidání jednoho zařízení do více skupin. Administrátor vidí alerty pro skupiny *Linux servers* a *All servers*. To je způsobeno tím, že jeden server má definované obě skupiny, a tudíž se zobrazuje „dvakrát“.

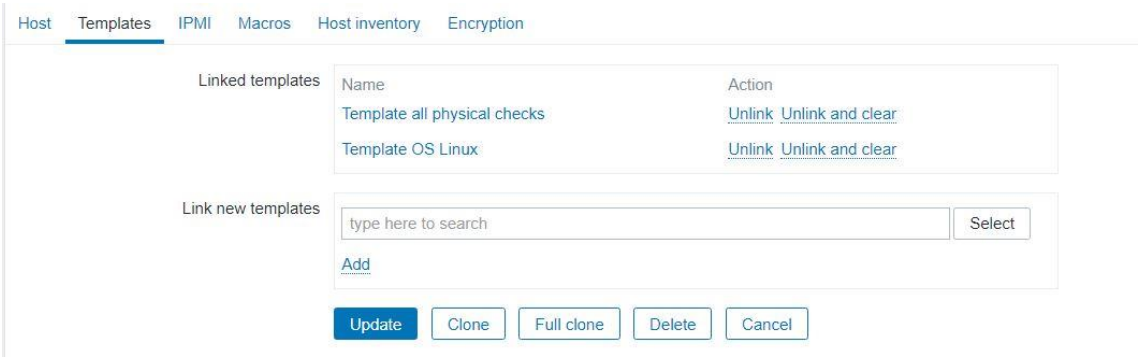


Obrázek 12 Vytvoření klientské stanice
Zdroj: vlastní zpracování



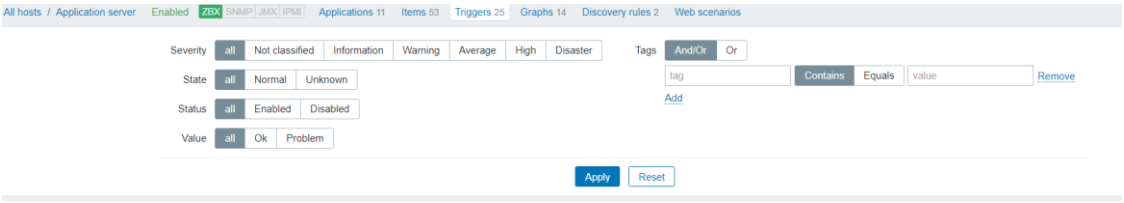
Obrázek 13 Zobrazení alertů agregovaně podle skupin
Zdroj: vlastní zpracování

Pro připojení šablony se musí daná šablona vybrat na záložce *Templates.* V obrázku níže je ukázáno kde se šablony přidávají.



Obrázek 14 Přidání šablony ke klientské stanici
Zdroj: vlastní zpracování

Každá klientská stanice se skládá z několika částí. První jsou informace o zařízení. Následují aplikace (Applications). Jednotlivé metriky jsou rozděleny podle toho, co monitorují. Dalšími parametry jsou metriky (Items), alerty (Triggers) a posledními parametry jsou grafy (Graphs). V obrázcích níže je ukázáno grafické rozhraní při konfiguraci jednotlivých klientských stanic.



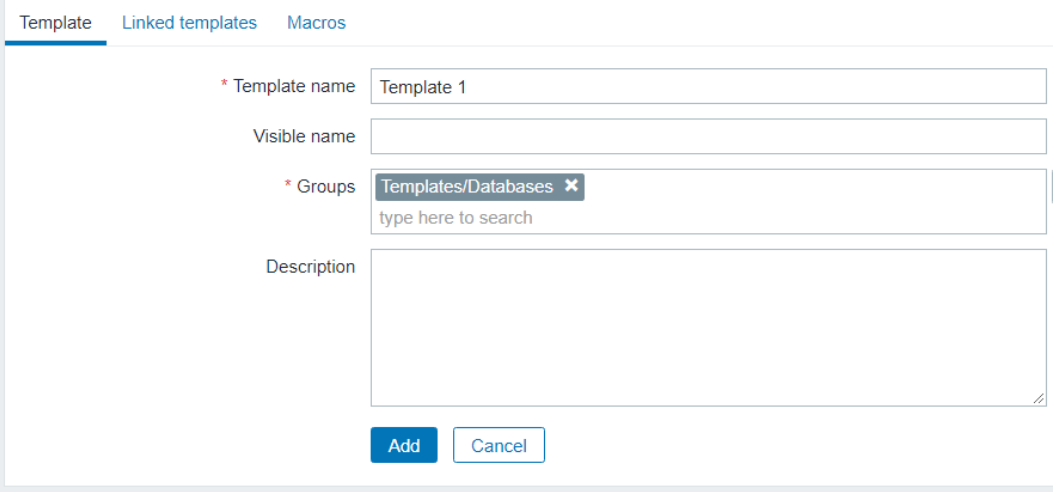
Obrázek 15 Hlavička stránky při konfiguraci klientské stanice
Zdroj: vlastní zpracování

Obrázek 16 Konfigurace alertů (Triggers)
Zdroj: vlastní zpracování

3.3.2.6 Vytvoření šablony

Užitečnou funkcionalitou je vytváření šablon pro urychlení konfigurace metrik, které jsou společné pro více klientských stanic. Obecně je doporučováno použít šablony i při konfiguraci metrik, které se nejsou až tolik časté. [21]

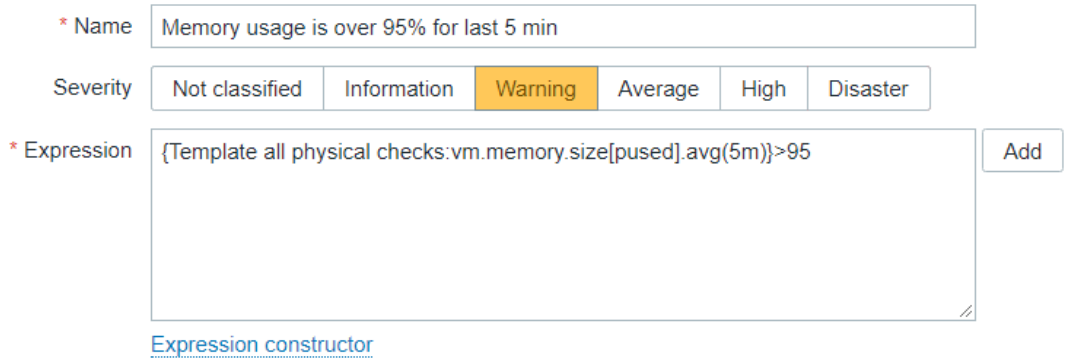
Šablony se vytvářejí na stránce *Configuration -> Templates -> Create Template.* Šablona se konfiguruje podobně jako klientská stanice. Na záložce *Linked Templates* lze vybrat již existující šablonu a přidat její vlastnosti do vytvářené šablony.



Obrázek 17 Vytvoření šablony
Zdroj: vlastní zpracování

3.3.2.7 Logické výrazy při vytváření alertů (Triggers)

Zabbix umožňuje při vytváření alertů definovat logické výrazy a použít funkce pro přepočet hodnoty vrácené agentem. Zabbix umožňuje použít mnoho různých funkcí např. průměrnou hodnotu, součet, porovnání hodnoty s regulárním výrazem, porovnání, zda se hodnota shoduje s předešlou hodnotou a další. V obrázku níže je ukázán logický výraz pro definovaní alertu, který bude porovnávat hodnotu využití operační paměti a upozorní uživatele, pokud je po dobu 5 minut průměrné využití operační paměti nad 95 %.



Obrázek 18 Logický výraz pro využití paměti nad 95%
Zdroj: vlastní zpracování

3.3.2.8 Integrace IM do procesu monitorování

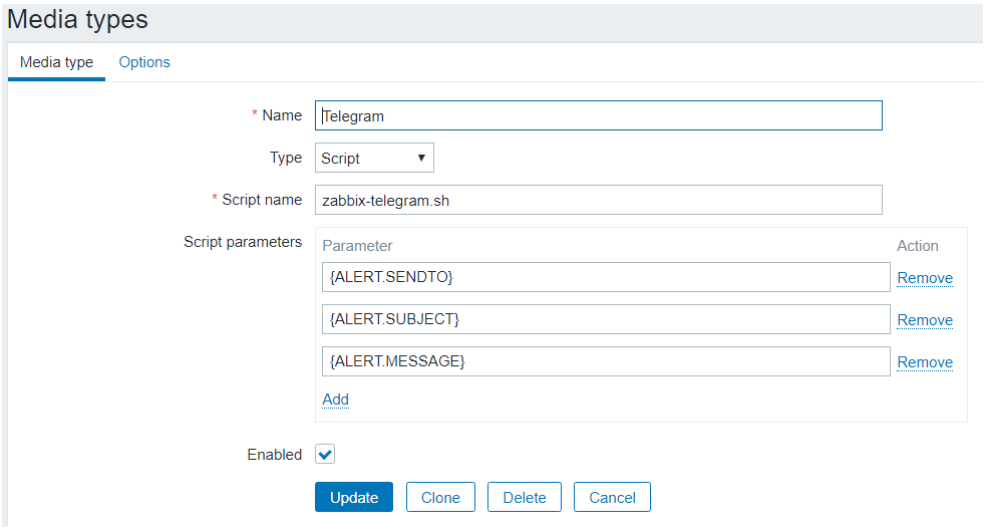
Pro Zabbix existuje opět několik řešení, jak zprovoznit posílání notifikací do aplikace Telegram. Pro potřeby této bakalářské práce byl použit skript[10] psaný v Linux BASH. Pro funkčnost bylo třeba vytvořit nového Bota. Proces byl stejný jako při vytváření Bota pro Nagios. Následovalo přidání do skupiny a vyplnění několika proměnných ve skriptu.

```
ZBX_URL=<URL_ADRESA_ZABBIX_SERVERU>
USERNAME=uživatelské jméno
PASSWORD=heslo
BOT_TOKEN=<BOT_TOKEN>
SEND_GRAPH=1
SEND_MESSAGE=1
ZABBIXVERSION34=1
```

Konfigurace 23 Parametry skriptu pro Zabbix Telegram

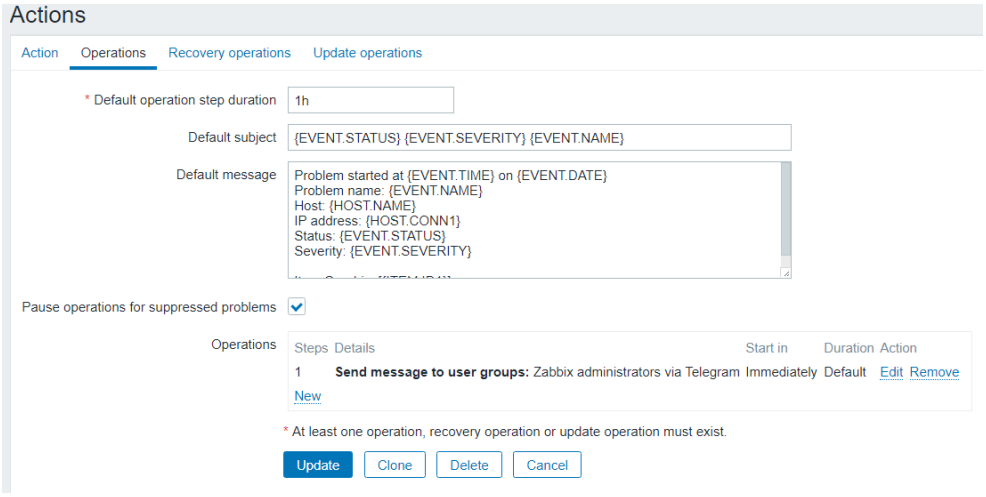
Po vyplnění parametrů se skript musel vložit do složky */usr/lib/zabbix/alertscripts* a přidat v grafickém rozhraní jako nový typ notifikace. V obrázku níže je ukázka přidání nového typu notifikace.

[10] https://git.cdp.li/polcape/zabbix/tree/master/telegram-notify



Obrázek 19 Přidání nové notifikace
Zdroj: vlastní zpracování

Nyní je třeba přidat text zprávy, která se bude odesílat uživateli do skupinového chatu a také uživateli nastavit identifikační číslo skupiny, do které se budou zprávy odesílat. Jak se konfigurace provede, je ukázáno v obrázcích níže.



Obrázek 20 Přidání akce pro odesílání notifikací
Zdroj: vlastní zpracování

Obrázek 21 Vložení identifikátoru skupiny pro uživatele
Zdroj: vlastní zpracování

Po poslání notifikace do telegramu přijdou 3 zprávy. První s předmětem problému, druhá s textem zprávy a třetí je obrázek grafu o daném problému za

stanovený časový interval. Tato doba se dá ručně ve skriptu nastavit. Snímek obrazovky níže ukazuje, jak vypadají zprávy doručené do aplikace Telegram.

Obrázek 22 Telegram ukázka posílání zpráv
Zdroj: vlastní zpracování

3.4 Porovnání nástrojů

V následující kapitole budou vybrané nástroje porovnány. Porovnání je rozděleno do několika částí:

- Instalace
- Konfigurace
- Uživatelská přívětivost
- Dokumentace
- Rozsah funkcionalit
- Celkové zhodnocení nástrojů

3.4.1 Instalace

Nagios byl instalován přes zdrojové soubory. Instalace přes zdrojové soubory je dosti zdlouhavá a uživatelsky nepřívětivá. Pro Nagios nejspíš také existují RPM balíčky. Bohužel tyto balíčky nebyly zmíněny v dokumentaci.

Zabbix byl instalován přes RPM balíčky. Instalace přes balíčky je mnohem jednodušší. Instalace Zabbix nástroje byla rozšířena o nutnost instalování MySQL serveru, pro ukládání konfigurace do relační databáze.

3.4.2 Konfigurace

Nagios používá ke konfiguraci textové soubory. V těchto souborech se poté definují objekty, které reprezentují jednotlivé metriky, klientské stanice atp. I při konfiguraci pár klientských stanic byla konfigurace velmi zdlouhavá. Užitečnou funkcionalitou je možnost přidání metrik přímo ke skupině klientských stanic. Tím lze snížit čas, potřebný ke konfiguraci monitorování.

Zabbix využívá ke konfiguraci webové rozhraní a k ukládání dat používá relační databázi. Při potřebě konfigurovat velké množství klientských stanic, se doba, potřebná pro konfiguraci, výrazně prodlouží.

3.4.3 Uživatelská přívětivost

Nagios je orientován spíše na textové soubory. Poskytované grafické rozhraní je „zastaralé“ a poskytuje velmi omezený počet akcí, kterými lze nástroj konfigurovat. Spíše slouží jen pro zobrazení konfigurace a změny se musí provést opět přes textové soubory. Nagios umí nativně pracovat s eskalací problému. Pro upozornění na anomálii v systému existují dvě úrovně: WARNING, CRITICAL. Pokud by systém zjistil, že se požadovaná metrika nachází v CRITICAL úrovni, pošle pouze varování pro tuto úroveň. Je totiž nadřazená úrovni WARNING.

Zabbix používá grafické rozhraní pro většinu konfigurace. Grafické rozhraní je moderní a snadno se v něm orientuje. Zabbix používá více úrovní eskalace problému než Nagios. Konkrétní jsou: Not Classfied, Information, Warning, Average, High, Disaster. Zabbix nativně neumožňuje automatickou eskalaci. Pro docílení tohoto chování je nutno definovat závislosti mezi jednotlivými triggery. V konfiguraci triggeru A ho lze propojit s triggerem B. Nyní je trigger A závislý na

triggeru B. Pokud je trigger B ve stavu PROBLEM, je trigger A ignorován. Tímto je redukováno množství odesílaných notifikací. Celkově je Zabbix uživatelsky velmi přívětivý.

3.4.4 Dokumentace

Při práci s nástroji bylo použito online dokumentací. Oba nástroje poskytují dostatečnou dokumentaci pro provedení celé konfigurace. Dokumentace k nástroji Zabbix je až příliš podrobně rozdělená a byl problém se v ní občas orientovat. Při potížích nebylo problém najít řešení na oficiálních fórech. Vše potřebné bylo v dokumentacích dostatečně vysvětleno.

3.4.5 Rozsah funkcionalit

Nagios po instalaci neobsahuje skoro žádné metody pro monitorování. Je potřeba instalace balíčku Nagios Plugins. I tento balíček však nepřináší veškeré potřebné nástroje pro implementaci monitorování. Například neobsahuje funkcionalitu pro monitorování využití operační paměti. Bylo nutno tedy stáhnout skript pro implementaci této funkce. Skript byl stažen ze stránky Exchange.nagios.com[11], která obsahuje celou řadu pluginů pro konfiguraci monitorování. Toto je velká nevýhoda oproti Zabbixu.

Zabbix umožňuje tzv. auto-discovery. Tato funkcionalita umožňuje automatické nastavení metrik, triggerů, grafů a dalších konfigurací u přidané klientské stanice. Nejčastěji se této funkce používá pro automatické zjištění diskových oddílů a síťových rozhraní serveru.

3.4.6 Celkové zhodnocení nástroje Nagios

Nagios je na trhu již dlouhou dobu. Stále je však považován za „otce“ většiny monitorovacích nástrojů [16]. Nagios poskytuje základní mechanismy potřebné k implementaci monitorování. Instalace nástroje Nagios je složitější a vyžaduje lepší orientaci v operačním systému Linux. Po instalaci všech potřebných pluginů

[11] https://exchange.nagios.org/directory/Plugins/System-Metrics/Memory/Check-mem-%28by-Nestor%40Toronto%29/details

se z něj stává použitelný nástroj. Neuškodilo by mu lepší grafické rozhraní a modernější vzhled. Lepší vzhled, podpora grafů a další funkce jsou dostupné v placené verzi Nagios XI. Výhoda Nagios nástroje je v možnosti automatizace konfigurace. Pomocí skriptů lze přenést konfigurace mezi servery bez nutnosti psaní společné konfigurace. Aby Nagios poskytoval stejné možnosti monitorování, je nutno instalovat velké množství pluginů. Toto je oproti Zabbixu méně uživatelsky přívětivé.

3.4.7 Celkové zhodnocení nástroje Zabbix

Zabbix je o něco málo mladší než Nagios. V základu nabízí oproti nástroji Nagios více možností, jak monitorovat klientské stanice. Jak bylo zmíněno dříve, nativně bohužel nepodporuje eskalaci problémů. Velkou výhodou je ale možnost odesílání obrázků s grafy do aplikace Telegram. Při konfiguraci rozsáhlé infrastruktury je u Zabbix nástroje nevýhodou delší doba potřebná na konfiguraci. Všechna nastavení je potřeba naklikat v grafickém rozhraní. Naopak při monitorování často se měnící infrastruktury, je vhodnější použít nástroj Zabbix. Díky funkcionalitě automatického nastavení metrik se dokáže přizpůsobovat změnám v prostředí.

Zabbix také umí pracovat s funkcemi pro vypočítání průměrné hodnoty za určitý čas. Tím lze eliminovat „zbytečné“ odeslání notifikace při krátkodobém vytížení klientské stanice. Pro naprostého začátečníka ve správě operačního systému Linux je lepší volbou Zabbix, kvůli jednoduchosti instalace a konfigurace. V tabulce níže je shrnuto porovnání obou nástrojů.

Nagios	Zabbix
Klady + eskalace problémů + přehledná dokumentace	**Klady** + webové rozhraní + automatické nastavení některých metrik
Zápory - omezená použitelnost webového rozhraní - nutnost manuálního doinstalování potřebných pluginů	**Zápory** - při více klientských stanicích dlouhá konfigurace - orientace v dokumentaci

Tabulka 2 Porovnání nástrojů
Zdroj: vlastní zpracování

4 Závěr

Cílem bakalářské práce bylo najít a porovnat nástroje pro monitorování aplikací, které jsou provozovány na linuxových serverech.

V teoretické části byla popsána definice monitorování. Byly představeny klíčové principy, které se při monitorování používají. Hlavním přínosem monitorování je informovanost administrátorů o stavu zařízení v infrastruktuře. Pomocí popsaných mechanismů lze snížit reakční dobu potřebnou při havárii v systému nebo prostředí. Díky včasnému doručení notifikace lze vědět o možné havárii delší dobu předtím, než by k ní skutečně mělo dojít.

V praktické části byly deklarovány požadavky na nástroje. Podle definovaných požadavků byly vybrány 2 nástroje. Konkrétně nástroje Nagios a Zabbix. Při porovnávání nástrojů byl kladen důraz na použitelnost nástroje a složitost konfigurace. Byla také prozkoumána možnost odesílání notifikací do aplikace Telegram. Pro odesílání zpráv existují i další alternativy, jako například aplikace Slack. Pro potřeby této bakalářské práce byla vybrána aplikace Telegram. Na závěru bakalářské práce byly celkově zhodnoceny oba porovnávané nástroje.

5 Seznam literatury

[1] TSAI, Peter. What is it and why is it amazing? *Spiceworks* [online]. 2016. Dostupné z: https://community.spiceworks.com/networking/articles/2553-server-monitoring-software-what-is-it-and-why-is-it-amazing

[2] Server Monitoring. *Techopedia* [online]. Dostupné z: https://www.techopedia.com/definition/29993/server-monitoring

[3] LIGUS, Slawek. *Effective monitoring and alerting*. Farnham: O'Reilly, 2013. ISBN 978-1-4493-3352-2.

[4] TURNBULL, James. *The art of monitoring* [online]. 2016 [vid. 2019-02-02]. ISBN 978-0-9888202-4-1. Dostupné z: http://proquest.safaribooksonline.com/9780988820241

[5] CHURCHMAN, Michael. The History of Monitoring Tools. *Sumologic* [online]. 2018. Dostupné z: https://www.sumologic.com/blog/devops/monitoring-tools-history/

[6] Active Vs reactive monitoring. *Olaohs* [online]. 2017. Dostupné z: https://olaohs.wordpress.com/2017/06/02/active-vs-reactive-monitoring/

[7] Active vs. Passive Server Monitoring. *Rapid7* [online]. 2017. Dostupné z: https://blog.rapid7.com/2017/05/24/active-vs-passive-server-monitoring/

[8] LAWRENCE, Michael. What is Proactive Monitoring? *Smallbusiness* [online]. Dostupné z: http://smallbusiness.chron.com/proactive-monitoring-73438.html

[9] ELLINGWOOD, Justin. An Introduction to Metrics, Monitoring, and Alerting. *DigitalOcean* [online]. 2017. Dostupné z: https://www.digitalocean.com/community/tutorials/an-introduction-to-metrics-monitoring-and-alerting

[10] 8 Key Application Performance Metrics & How to Measure Them. *Stackify* [online]. 2017. Dostupné z: https://stackify.com/application-performance-metrics/

[11] Monitoring Alerts. *InterSystems* [online]. Dostupné z: https://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=EMONITOR_alerts#EMONITOR_alerts_intro

[12] LÊ-QUÔC, Alexis. Monitoring 101: Alerting on what matters. *Datadog* [online]. 2015. Dostupné z: https://www.datadoghq.com/blog/monitoring-101-alerting/

[13] LEONI, Ken. Server Monitoring and Alerts - Getting past common obstacles. *Heroix* [online]. 2017. Dostupné z: https://blog.heroix.com/blog/server-monitoring-and-alerts-getting-past-common-obstacles

[14] GOLDEN, Bernard. *The requisite and advanced features of performace monitoring software* [online]. 2013. Dostupné z: https://searchitoperations.techtarget.com/tip/The-requisite-and-advanced-features-of-performance-monitoring-software

[15] Best Open Source Network Monitoring Tools and Software (Linux/Windows). *ITTSystems* [online]. 2019. Dostupné z: https://www.ittsystems.com/best-open-source-network-monitoring-tools/

[16] Top FREE Network Monitoring Tools. *DNSstuff* [online]. 2019. Dostupné z: https://www.dnsstuff.com/free-network-monitoring-software

[17] TABONA, Andrew. The top 20 free Network Monitoring and Analysis Tools for sysadmins. *TechTalk* [online]. 2018. Dostupné z: https://techtalk.gfi.com/the-top-20-free-network-monitoring-and-analysis-tools-for-sys-admins/

[18] HAYDEN, James. 20 Top Server Monitoring & Application Performance Monitoring Solutions. *Hayden James* [online]. 2018. Dostupné z: https://haydenjames.io/20-top-server-monitoring-application-performance-monitoring-apm-solutions/

[19] History of Nagios. *Nagios* [online]. 2016. Dostupné z: https://www.nagios.org/about/history/

[20] Nagios documentation. *Nagios* [online]. 2019. Dostupné z: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/toc.html

[21] Zabbix documentation. *Zabbix* [online]. 2018. Dostupné z: https://www.zabbix.com/documentation/4.0/start

[22] VACCHE, Andrea Dalle a Stefano Kewan LEE. *Zabbix network monitoring essentials your one-step solution to efficient network monitoring with Zabbix*. Birmingham, UK: Packt Publishing, 2015. ISBN 978-1-78439-408-0.

Oskenované zadání práce

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2018/2019

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Pešek Viktor	Prokopova 324, Česká Třebová - Parník	I1600588

TÉMA ČESKY:

Porovnání free monitoring nástrojů včetně PoC (Proof of Concept)

TÉMA ANGLICKY:

Comparison of free monitoring tools including PoC (Proof of Concept)

VEDOUCÍ PRÁCE:

doc. Ing. Vladimír Bureš, Ph.D., MBA - KIT

ZÁSADY PRO VYPRACOVÁNÍ:

Cílem této bakalářské práce je porovnat free nástroje a jejich výhody a nevýhody podle typu monitorovaného systému. Vytvoření PoC pro vybrané nástroje. V rámci PoC prakticky ověřit použitelnost, uživatelskou přívětivost, administraci a robustnost vybraných nástrojů.

SEZNAM DOPORUČENÉ LITERATURY:

CHURCHMAN, Michael. The History of Monitoring Tools. Sumologic [online]. 2018 Dostupné z: https://www.sumologic.com/blog/devops/monitoring-tools-history/
LIGUS, Slawek. Effective monitoring and alerting. Farnham: O'Reilly, 2013. ISBN 978-1-4493335-2-2.
TURNBULL, James. The art of monitoring. New York: Turnbull Press, 2016. ISBN 978-0-9888202-4-1
ELLINGWOOD, Justing. An Introduction to Metrics, Monitoring, and Alerting [online]. 2017 Dostupné z: https://www.digitalocean.com/community/tutorials/an-introduction-to-metrics-monitoring-and-alerting

Podpis studenta: ..... **Datum:** .....

Podpis vedoucího práce: ..... **Datum:** .....