Mendelova univerzita v Brně
Provozně ekonomická fakulta

Monitorovací systém Nagios pro dohled síťových a serverových služeb

Bakalářská práce

Vedoucí práce:
Ing. Jiří Balej

Beťko David

Poděkování

Tímto bych chtěl podekovat vedoucímu práce Ing. Jiří Balej za užitečné rady a trpělivost při konzultacích. Velké poděkování dále patří za nejlepší přednášky a vynikající cvičení v síťových akademií Ing. Martin Pokorný, Ph.D spolu s Ing. Petr Zach, Ph.D, kde mě naučili věci, které jsem využil ve své práci. Mé poděkování si rovněž zaslouží můj externí konzultant s firmy Aliacte s.r.o. Ing. Radim Klabal, bez kterého by tato práce nebyla uskutečněna, dále za jeho rady a vhodné nasměrování k tématu, konzultace a připomínky k práci. V poslední části bych poděkoval své rodině za trpělivost a podporu ve vzdělání.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Monitorovací systém Nagios pro dohled síťových a serverových služeb** vypracoval/a samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací.*

Jsem si vědom/a, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 22. května 2017 _____

Abstract

Beťko, D. Monitoring system Nagios for tracking network and server services. Bachelor thesis. Brno: Mendel University, 2017.

The goal of the bachelor thesis is to configure the monitoring system Nagios in order to cover any defined services in a business environment. The open-source is a key solution of the monitoring system itself. The sample and promotion of the installation and configuration of the Nagios system comes in a practical part of the thesis. Moreover, there is a verification of the Nagios systems functionality on a testing basis, as well.

Keywords

Nagios, Monitoring, network, SNMP protocol, Syslog, server, Zabbix, OPSview, Cacti, Zenoss, OpenNMS, IBM Tivoli, computer network, open-source, plugin

Abstrakt

Beťko, D. Monitorovací systém Nagios pro sledování síťových a serverových služeb Bakalářská práce. Brno: Mendelova univerzita v Brně, 2017.

Bakalářská práce se zabývá konfigurací monitorovacího systému Nagios tak, aby pokrýval veškeré definované služby ve firemním prostředí. Celý monitorovací systém je založen na open-source řešení. V praktické části je ukázka instalace a konfigurace systému Nagios a ověření jeho funkčnosti na bázi testování.

Klíčová slova

Nagios, monitoring, síť, protokol SNMP, Syslog, server, Zabbix, OPSview, Cacti, Zenoss, OpenNMS, IBM Tivoli, počítačová síť, Open-source, plugin.

Obsah

1 Úvod a cíl práce

1.1 Úvod

V dnešní době musí jakákoliv menší nebo větší firma chránit data, se kterými dále pracuje, musí si zabezpečit dostupnost všech poskytovaných služeb, minimalizovat přetížení svojí sítě, detekovat kolize. Dále musí sledovat svoje softwarové a hardwarové prvky, aby se předešlo výpadkům sítě na sledovaných prvcích. V současnosti je monitoringu síťových a aplikačních služeb pro firmu nezbytným systémem. Podstatné je sledování kritických služeb ve firmě, které neohrozí chod. Nasazením monitorovacího nástroje předcházíme kolapsům na síti.

Díky tomu má administrativní zaměstnanec v pozici technika řadu funkcí na získávání podstatných informací. V současné době je na trhu k dispozici velké množství monitorovacích systémů od open-source až po licencované. Na trhu se nabízí volně stažitelné nástroje např. Nagios, Zabbix a licencované systémy poskytují společnosti jako je Cisco, IBM, Microsoft, HP.

1.2 Cíl práce

Hlavním cílem práce je nasazení sledovacího systému pro firmu Aliacte s.r.o a jeho následné testování funkčnosti, konkrétně toho, zda navržený systém Nagios odpovídá požadavkům firmy. Porovnávání monitorovacích systému na základě dohody s administrátory sítě ukáže výhody i nevýhody v rámci firemní dostupnosti. Výsledné řešení monitorovacího systému je důležité nakonfigurovat tak, aby řešení pokrývalo všechny potřebné služby a systémovou notifikaci pomocí emailu. Konfigurované služby budou pro síťové prvky (routery, switche), serverové OS, síťové služby a systémové zdroje. Služby musí vyhovovat daným administrátorům provozované sítě, tak aby vše bylo uživatelsky přívětivé. Dílčím cílem práce je s administrátory sítě analyzovat současný stav firemní sítě a identifikovat hrozby, které mohou nastat při stávajícím stavu. Analýza sítě a současných systémů bude probíhat z důvodu orientování se v infrastruktuře, aby mohl být nasazen monitorovací systém.

1.3 Metodika práce

Ke splnění zadání práce bude důležité uskutečnění následujících kroků jeden za druhým. Analýza současných řešení bakalářských i diplomových prací pro ověření neexistence stejného řešení závěrečných prací, ve kterých se zabývali problematikou monitorování sítě. Z vymezených porovnávaných systémů je potřebné prozkoumat a zkontrolovat, zda neexistuje vhodnější monitorovací systém než Nagios z hlediska ovládání, funkčnosti, instalace, definování hostitelů a konfigurace produktu. U každého porovnávaného produktu bude definován princip monitorování služeb, jeho architekturu a práci s produktem. Monitorované

hodnoty i parametry budou definovány podle potřeb využívání jednotlivých síťových protokolů a služeb ve firmě. Frekvence monitorování i upozorňování bude probíhat u hardwarové složky, operačního systému, síťového prvku a služby v odlišných časových intervalech. Primárním cílem je nasazení a začlenění monitorovacího systému do firemní infrastruktury, jeho konfigurace, nastavení a odladění chyb pro monitorování. V posledním kroku bude provedeno funkční testování varování při výpadku služeb nebo síťových prvků.

2 Rešerše stávajících prací

Tato kapitola se bude zabývat předchozími bakalářskými, popřípadě diplomovými závěrečnými pracemi. Kapitola je přehledem podobných vypracovaných řešení závěrečných prací se zaměření na monitoring. Podobné práce byly vyhledávány z České centrální databáze závěrečných prací (http://theses.cz). Ne všechny vysoké školy využívají centrální databázi, proto bylo nutné použít přímo univerzitní databázi vybraných škol, a to Mendelovy univerzity v Brně (https://is.mendelu.cz/zp), Vysokého učení technického v Brně (https://dspace.vutbr.cz) a Českého vysokého učení technického v Praze (https://dspace.cvut.cz). Bylo vyhledáváno pomocí klíčových slov network monitoring nebo monitoring sítě. Sledované práce byly v rozmezí intervalu tří až devět let. Ve starších pracích je vidět technologický pokrok pro monitorovací systémy z hlediska, jak užívání, tak z hlediska jejich funkčnosti, výkonnosti, optimalizace procesoru, dostupných modulů.

2.1 Analýza stávajících prací

BP – František Vařacha – Monitorovací systém firemní počítačové sítě na bázi open source řešení

Vařacha se ve své bakalářské práci zabývá vylepšením monitorovacího systému v reálné firmě na bázi open-source. Práce je inspirovaná jeho analýzou nových uživatelských požadavků ve firmě a jaké principy sledování použil pro monitoring. Při výběru monitorovacích systému se zaměřil na open-source řešení. Výsledné řešení bylo provedeno pomocí nástroje Nagios. V práci se zabýval testováním toho, zda řešení vyhovuje požadavkům firmy. Provedl i ekonomické zhodnocení instalovaného systému. (Vařacha, 2014)

DP – Bc. Jindřich Matúš – Monitorování stavu rozsáhlých sítí

Matúš se ve své diplomové práci zaměřuje na vytvoření systému pro monitorování stavu rozsáhlého stavu sítě, ve kterém bude ukládat data do databáze. Pro výběr monitorovacích systému se zaměřil na open-source řešení. Zaměřuje se především na služby typu SNMP a SSH, které bude získávat od OS Linux a Routeru Mikrotik. Ve své práci nevyužil možnost otestování monitorovacího systému Cacti, pro správnou konfiguraci a upozorňování uživatele. (Matúš, 2008)

BP- Lukáš Vozdecký – Srovnání systému pro sledování provozu počítačových sítí

Vozdecký se zabývá porovnáváním nejrozšířenějších open-source řešení jako Nagios, Zabbix a Big Sister. Zjišťuje možnost nasazení z hlediska potřeb a časové náročnosti v provozu. Upozorňuje na podstatné rozdíly mezi zvolenými nástroji

skrz efektivitu a v rozdílnost využití aplikace. Ve výsledném procesu zvolil jako nejefektivnější nástroj Zabbix, který se dá konfigurovat jen pomocí webového rozhraní jako druhý nejlépe využitelný nástroj byl podle něj zvolen Nagios, který je vysoce konfigurovatelný a flexibilní pro velké i malé firmy. Nezaměřoval se na konkrétní sledované služby. (Vozdecký, 2007)

DP – Bc. Daniel Eisner – Analýza moderních technologií pro dohled a správu firemní infrastruktury

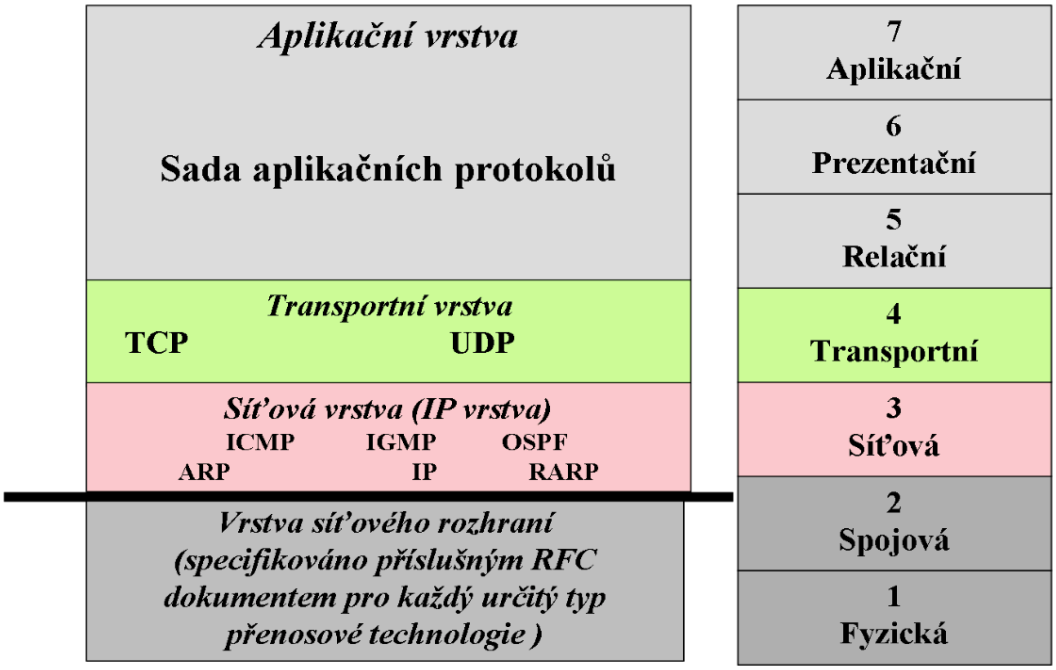
Eisner se ve své práci zabývá problematikou monitorovaní dle technologií s využitím agenta nebo bez využití agenta. Následně vystihuje značné výhody a nevýhody těchto technologií. Dále jsou rozebírány možnosti dotazování pomoci SNMP a ICMP do detailů. Podrobněji se rozebírá monitorovací systém Nagios. Zabývá se jeho prerekvizitami, hardwarovou, softwarovou konfigurací, rozšíření pomocí pluginů. V práci se z důvodu časového vytížení, který uvedl on nezabýval následným testováním. (Eisner, 2015)

2.2 Zhodnocení využití stávajících prací

Porovnávané práce se zabývají nasazením monitorovacího systému na bázi open-source řešení. V práci byly srovnávány kromě nelicencovaných i licencované monitorovací systémy, které přinesly větší rozmanitost práce. V některých pracích se autoři nezaměřili přímo na monitoring vyhrazených síťových služeb a prvků v sítí nebo v práci nespecifikovali stejnou kombinaci monitoringu. Další rozdíl porovnávaných prací byl v tom, že své výsledné řešení autoři neotestovali v různých provozních fází systému, např. při změně zátěže serveru, výpadku služeb nebo celého serveru a nezabývali se návrhem SLA (Service-level agreement)., případně splněním požadovaného SLA.

3 Teoretické možnosti využití ISO/OSI

ISO/OSI umožnuje srovnat strukturu síťových architektur, definuje funkce vytvářející komunikační proces, považujeme ho za základ pro síťové technologie a můžeme ho považovat za základ tvorby struktury sítě a síťové komunikace. V modelu Model TCP/IP lze komunikovat v návaznosti vrstvou následující nebo předcházející a v komunikaci musí být obsaženy všechny nižší vrstvy. Úkolem každé vrstvy je poskytnout informace a služby následující vyšší vrstvě. RM ISO/OSI je tvořen sedmi vrstvami, kde se na každé vrstvě specifikuji protokoly a práce mezi těmito protokoly. Každá vrstva má specifické funkce, které zodpovídají za správný přesun dat v síti a proces musí probíhat ve správném pořadí. (Meinel, 2013)



Obrázek 1: Model TCP/IP

Zdroj: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=590

3.1 Souhrnný popis vrstev

3.1.1 Fyzická vrstva

Fyzická vrstva nám umožňuje přenos jednotlivých bitů komunikačním kanálem, v médiu jako je např. vzduch nebo kabely. Dále zajišťuje synchronizaci a multiplexing v přístupu na médium. Přenášeným bitů nepřiřazuje žádný význam. Specifikace samostatného fyzického média není součástí vrstvy v OSI modelu. Logické spojení lze realizovat jedním fyzickým médiem. (Dostálek, 2003)

3.1.2 Linková vrstva (Spojová)

Úkolem linkové vrstvy je zajistit bezchybný přenos dat mezi přímo propojenými stanicemi. Vytváří se zde rámce, které obsahují vlastní přenášené informace, údaje pro adresování a zabezpečení proti chybám při přenosu. (Dostálek, 2003)

Existují určité protokoly linkové vrstvy a to:

- **Ethernet –** rámce se šíří segmentem lokální sítě, je nezávislý na ostatních rámcích, je dopravován samostatně, nezávisle na ostatních rámcích. Stanice akceptuje nejenom rámce adresované její výlučnou 6B adresou, ale i oběžníky. (Sosinsky, 2010)

3.1.3 Síťová vrstva

Síťová vrstva zajišťuje adresování a směrování dat v síti od zdrojového k cílovému zařízení přes několik mezilehlých zařízení. Síťové pakety jsou na směrovačích předávány mezi fyzickými porty směrovače podle logické adresy cílové sítě. (Dostálek, 2003)

Existují určité protokoly síťové vrstvy a to:

- **ARP** – mapování mezi logickou síťovou adresou (IP) a fyzickou (MAC) adresou, kde hledáme cílovou MAC adresu pomocí IP adresy. Opačným procesem je protokol RARP. (Naugle,2006);
- **IPv4** – používá IP adresu o délce čtyři bajty, která adresuje jednoznačně síťové rozhraní systému. (IPv4, 1981);
- **IPv6 –** používá IP adresu o délce 16 bytů. Typy IPv6 adres máme unicast, multicast a anycast. Linkové adresy platí pouze v lokální síti a poznají se podle prefixu fe80::/10.(IPv6, 1998);
- **DHCP** – protokol pro automatizované přidělování IP adresy a dalších síťových parametrů (IP adresa, maska sítě). (Dostálek, 2003);
- **ICMP –** slouží k signalizaci mimořádných událostí v sítích postavených na bázi TCP/IP. Balí svoje datové pakety do IP-protokolu. ICMP zpráva je generována cílovým i zdrojovým uzlem datagramu, ke kterému se zpráva vztahuje. Jedním z mezilehlých směrovačů, kterými tento IP datagram prochází. (ICMP, 1981).

Na síťové vrstvě se specifikují i směrovací protokoly:

- BGP – navazuje a udržuje komunikace se sousedními směrovači od ISP;
- OSPF – vytváří hierarchické sítě tvořené směrovacími oblastmi, definuje více typů směrovačů generujících více typů OSPF zpráv;
- RIPv1, RIPv2 – umožňuje směrovačům komunikovat mezi sebou a reagovat na změny.

3.1.4 Transportní vrstva

Transportní vrstva je nejnižší vrstva, na kterou se odvolávají síťové aplikace (uživatelské a systémové). Zajišťuje kompletní přenos vlastních dat, kvalitu služby. Zabezpečuje, aby se zpráva dostala k příjemci správně, bez poškození, a v případě chyby, zajistí opakované odeslání. (Dostálek, 2003)

Existují určité protokoly transportní vrstvy a to:

- **TCP –** navazuje a ukončuje spojení, kde pro navázání a ukončení slouží sada příznaků (SYN – používá se pro navázání spojení a FIN – používá se pro ukončení spojení). Při spojení do každého segmentu vkládá pořadové číslo a příjemce pak potvrzuje (acknowledgment), čísluje a sleduje, jestli nebyl tok dat přerušen;
- **UDP –** služba nespojovaná – služba nespolehlivá, kde transport nelze řídit, je velmi efektivní, rychlá. Jsou zde malé provozní režie a pro provoz nepoužívá sekvenční čísla; (Meinel, 2103)
- **Socket** – konkrétní komunikace mezi aplikacemi a uzly. Jsou tvořeny IP adresou uzlu a číslem portu, na kterém naslouchá daná komunikující aplikace. (Socket, 1971)

3.1.5 Relační vrstva

Relační vrstva tvoří spojení mezi aplikacemi, správa session. Komunikuje zde jedna aplikace s druhou a posílá více dat po sobě. Udržuje celé spojení mezi dvěma počítači a úkolem této vrstvy je navázání relací mezi koncovými stanicemi. (Dostálek, 2003)

Prezentační vrstva

Prezentační vrstva poskytuje jednotnou reprezentaci dat a jejich šifrování. Specifikuje způsob, jakým jsou data formátována, prezentována, transformována a kódována. (Dostálek, 2003)

3.1.6 Aplikační vrstva

Aplikační vrstva poskytuje aplikacím přístup k síťovým službám. Obsahuje také protokoly, které implementují síťové aplikace. Využívá služby nižších vrstev. Správce má možnost využít protokol SSH pro bezpečné přihlášení k serveru. Kromě tohoto protokolu máme další protokoly, které budeme využívat, a to např.:

Poštovní aplikace

- SMTP – základní protokol pro výměnu poštovních zpráv;
- POP3 – přístup do mailboxu ze vzdáleného klienta na získání e-mailů;
- IMAP4 – přístup do mailboxu ze vzdáleného klienta, kde máme více funkcí s poštou a umožňuje přístup k poště z různých koncových zařízení.

Další protokoly:

- SNMP – správa síťových prvků, zjišťování stavových informací o zařízeních, nastavení parametrů na síťovém zařízení;
- FTP – autorizovaný přístup do souborového systému hostitelského uzlu, obousměrný přenos, kde vytváří 2 TCP spojení na portech 21 – spojení a řízení a 20 – pro přenos dat. (Dostálek, 2003);
- HTTP – a jeho zabezpečená verze, která je šifrovaná HTTPS, protokol pro přenos dat mezi serverem a klientem schopný přenášet data určená URL adresou. Komunikuje na portu 80 TCP a jeho zašifrovaná verze HTTPS na portu 443; (HTTP, 1999)
- SSH – protokol pro vzdálené a zabezpečené přihlášení, kde neposílá heslo v nezabezpečené formě. Komunikuje na portu 22 protokolu TCP. (Barret, 2003)

4 Monitoring serverových služeb

Existuje mnoho způsobů, jak monitorovat stav operačního systému nebo jednoúčelových zařízení, jako jsou tiskárny, switche, servery nebo routery. Dále lze zjišťovat dostupnost síťových služeb (služby FTP, SSH, HTTP, HTTPS), stavy HW a to nezávisle na použitém OS.

Jednotlivé metody monitoringu je možné kombinovat a maximalizovat, tak aby byla výsledná efektivita co nejlepší. Samotné monitorování se využívá mezi jiným aj k sledování procesů a stavu uvnitř OS (vytížení CPU, paměti RAM, zaplnění disků, služby nesouvisející se sítí).

Při využití monitorovacího systému pro dohled serverů máme dvě možnosti přístupu k informacím. Monitorování s agentem a bez agenta, kdy monitoring s agentem představuje daný prvek, který se na monitoringu podílí aktivně OS. Monitoring bez agenta znamená testování vlastních služeb serveru, kde se data získávají pomoci protokolů např. SNMP.

Na začátku monitorování je třeba naplánovat, jaký výstup je pro danou oblast nejvhodnější. Máme dvě oblasti. Jednou z oblastí jsou události, které můžeme získat pomoci Syslogu nebo SNMP trapů ze serverů či routerů. Druhou oblastí máme hodnoty číselné, které nám okamžitě ukazují stav (CPU, RAM atd.).

Díky softwarovému nástroji Wireshark lze odchytávat dané pakety, monitorovat si provoz v síti pro daný uzel a podrobněji analyzovat daný paket, a to za pomocí protokolu NetFlow tzv. real-time analyzátoru, který zachytává data v reálném čase a vykresluje je do grafu.

Příklad monitorovacích možností serverů, které se mohou využívat

- Dostupnost serverů a jejich služeb;
- aktivní síťové prvky pomocí ICMP;
- bezpečnostní incidenty;
- síťové komunikace/provoz;
- chyba napájení/zdroje, porucha ventilátoru, pevného disku;
- teplota čidla na skříních serveru;
- události na serveru (Syslog);
- vytížení linek – přenos dat;
- vytížení CPU, RAM, zaplnění disku;
- informace o portech switchů, routerů pomocí služby SNMP;
- webové a přenosové služby (HTTP, HTTPS, SSH, FTP, SFTP);
- kontrola e-mail služeb (SMTP, POP3, IMAP);
- jednoduché síťové zařízení jako UPS a tiskárny;
- další možné služby a funkce pomocí pluginů. (Bouška, 2009)

4.1 Síťová vrstva

Pro monitorování dostupnosti stanice je vhodné použít protokol ICMP, který je popsán v dokumentu RFC 792. Monitorovací systém zasílá v pravidelných intervalech ICMP zprávu „Echo Request“, na kterou očekává odpověď dostupnosti stanice „Echo Reply“. Pokud tato zpráva nedorazí ve stanoveném čase a počtu, je monitorovaná stanice považována za nedostupnou případně nespolehlivou.

Kromě monitorování dostupnosti stanice je vhodné monitorovat stav směrovacích tabulek, zvláště pak při používání dynamických protokolů. Směrovací tabulky mohou být naplněny staticky, a to editované ručně administrátorem neměnné nebo dynamicky, pomocí protokolů, kde se vytváří a udržují automaticky. (Bing, 2002)

4.2 Aplikační vrstva

4.2.1 SNMP (Simple Network Managment Protocol)

SNMP protokol je obvykle spojen s řízením routeru, který může být použit pro správu mnoha typů zařízení viz. SNMP Architektura. Jádrem SNMP protokolu je jednoduchý soubor operací (set a get). Pomocí těchto operacích, které umožňují správci změnit informace/hodnoty na zařízení a tím ovlivnit jeho chování (např. máme možnost vypnout jednotlivá zařízení nebo jejich rozhraní). Taky máme možnost kontrolovat rychlost přenosu na těchto rozhraních nebo rychlost zpracování operací na daném zařízení. Popřípadě monitorovat zda-li je síťové rozhraní nebo celé zařízení v provozu. SNMP může monitorovat také teplotu na přepínači a upozorní nás, když je její hodnota mimo stanovených limitů. SNMP lze použít ke správě Unix systémů, systémů Windows, tiskárny, napájecího zdroje a dalšího jakéhokoliv zařízení, kde běží software, který umožňuje vytahování SNMP informací. Vytahování SNMP informací zahrnuje, jak fyzické zařízení, ale také softwary, jako jsou webové servery a databáze. Dále je možnost sledovat celou síť, nejen od jednotlivých zařízení, pomocí vzdáleného monitorování RMON (Remote Monitoring). Může být použit k monitorování nejen provozu sítě, ale také WAN sítě.

Existují určité verze protokolu, kde první protokol SNMPv1**,** počáteční verze protokolu SNMP, je definována v RFC 1157. Bezpečnost je založena na komunitě, které je šifrováno společným heslem (tzv. Community String), které umožní přistup k informacím pro správu daného zařízení. Následující komunity nám umožňuje pouze čtení, čtení se zápisem a trap.

Druhá verze protokolu SNMPv2 stále využívá komunitní řetězec, kde není dobře vyřešena bezpečnost (stejně jako ve verzi jedna je zde heslo – community string – posíláno v plain textu). Jeden z hlavních rozdílů oproti verzi 1 je možnost hromadného stažení více informací naráz.

Poslední nejnovější verze protokolu jeSNMPv3. Jeho hlavním přínosem pro správu sítě je bezpečnost dosažená šifrováním, kontrolou integrity a vylepšením vzdálené komunikace. To přidává podporu silné autentizaci

a soukromé komunikaci mezi spravovanými entity. Inspiruje se předchozími SNMP verzemi a bere si z nich to nejpodstatnější.

SNMP manager nebo managment systém je samostatný subjekt, jenž je zodpovědný za komunikaci s SNMP agentem, který je implementovaný na síťových zařízeních. Klíčové funkce SNMP managera jsou dotazy k agentům a odpovědi od agentů, nastavuje proměnné agentovi a zpracovává asynchronní události od agenta.

SNMP Agent je program, který je zabalen do síťového prvku. Aktivací agenta nám umožňuje sbírat informace z databáze daného prvku a předávat je k dispozici SNMP manažerům. Klíčovou funkcí SNMP agenta je shromažďování informací o řízení a jeho lokálním prostředí, ukládá a získává informace pro řízení, jak je definováno v MIB, a které dále signalizuje události manažera.

SMI (Structure of Management Information)

Poskytuje způsob, jak definovat spravované objekty a jejich chování. Agent má ve svém držení seznam objektů, které sleduje. Jeden takový objekt může být například provozní stav rozhraní routeru. SMI souhrnně definuje informace, které NMS (Network Managment station) používá k určení celkového stavu zařízení, na němž se agent nachází.

MIB (Management information base)

SNMP nám nedefinuje identifikátory jednotlivé proměnné, ale díky MIB nám umožní jejich informace definovat. MIB si můžeme představit jako objekty a strukturu spravovaných dat, která jsou v databázi a která agent sleduje. SMI poskytuje způsob, jak definovat spravované objekty, zatímco MIB přímo definuje samotné objekty. Agent může realizovat více MIB, ale všichni agenti realizují konkrétní MIB s názvem MIB-II (RFC 1213). Hlavním cílem MIB-II je poskytnout obecné TCP / IP informace, definuje nám také statistiky o jednotlivých rozhraních (rychlost odeslaných a přijatých oktetů).

Get Operace

Požadavek GET je iniciován pomocí NMS, jenž pošle požadavek agentovi. Agent obdrží žádost, zpracuje požadavek a pokud úspěšně získá požadované informace odešle GetResponse zpět do NMS.

GET NEXT Operace

Operace GetNext umožňuje vystavit posloupnost příkazů k načtení více hodnot z MIB. Jinými slovy, pro každý objekt MIB chceme získat, oddělené GetNext žádosti a GetResponse, které jsou generovány.

GetBulk Operace

Verze SNMPv2 nám definuje nový provoz getbulk, která nám poskytuje získat větší část sekce tabulky najednou (setříděných podle identifikátoru objektu –

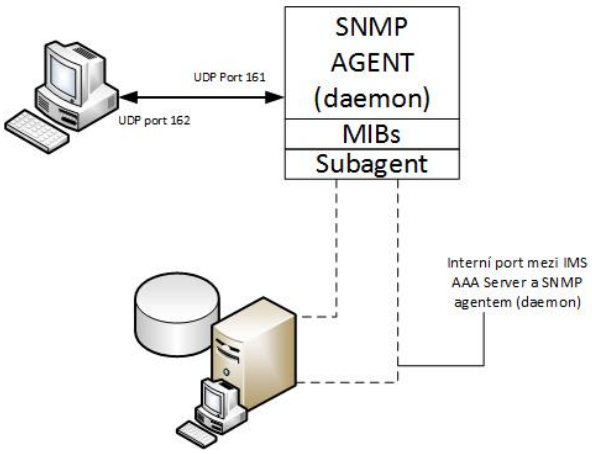
tzv. „lexicografic ordering“). Standardní Get Operace se může pokusit získat více než jeden objekt MIB najednou, ale velikosti zpráv jsou omezeny schopností agenta. Pokud agent nemůže vrátit všechny požadované odpovědi, vrátí chybové hlášení s nulovými daty. Operace getbulk na druhé straně říká, že agent může poslat zpět co největší část odpovědi, pokud to půjde.

Set Operace

Příkaz set se používá ke změně hodnoty spravovaného objektu nebo k vytvoření nového řádku v tabulce. Objekty, které jsou definovány v MIB pro čtení i zápis, nebo jen pro čtení, mohou být pozměněné nebo vytvořené pomocí tohoto příkazu. (mauro)

TRAP

Trap poskytují způsob odeslání oznámení pro agenta o monitorovací stanici, které by měl vědět. Značí nám způsob, jak říci NMS, že nastala chyba. Traps, které agent může generovat, jsou definovány v MIB, které podporuje. Počet traps se může pohybovat v rozmezí od nuly do stovky. (Mauro, 2005)



Obrázek 2: SNMP Architektura

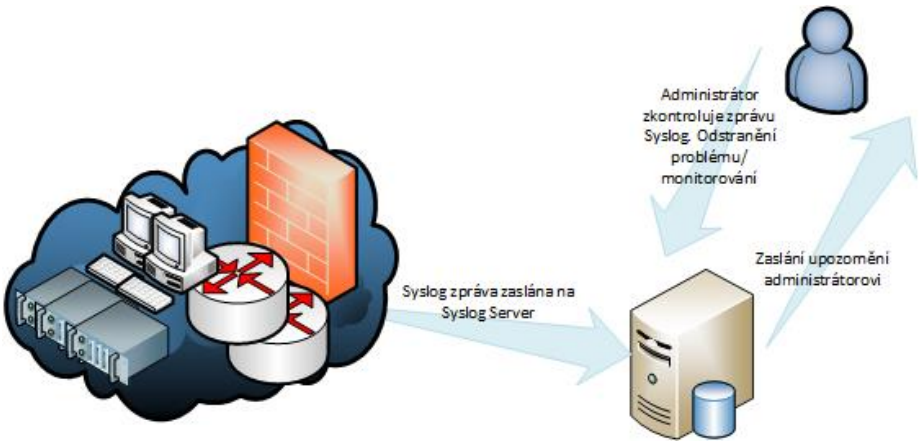
Zdroj: https://www.juniper.net/techpubs/software/aaa_802/imsaaa11/sw-imsaaa-admin/html/SNMP%20Architecture4.gif

4.2.2 Syslog

Linux a UNIX systémy mohou být nastaveny tak, aby se významné události odesílaly prostřednictvím syslogu na známý Syslog server. Syslog protokol je podporovaný širokou škálou zařízení a může být použit pro různé typy událostí. Slouží nám ke koncentrování vzdálených nebo lokálních logů z různých zařízení a jejich aplikací, kde na ně díky tomu můžeme reagovat. Pro lepší přehlednost a vyhledávání je důležité, aby monitorovací systém informoval správce sítě o důležitých syslogových událostech pomocí webového rozhraní na Syslog server.

Ke všem výsledkům na Syslog serveru máme možnost použít metodu filtrování jednotlivých položek daného bloku.

Například směrovače i přepínače mohou odesílat zprávy syslog o uživatelích přihlášení do konzole nebo web managmentu. Servery založené na systému Windows nepodporují Syslog, ale podporují nástroje třetích stran, které nám usnadňují sběr dat a ta dále předávají Syslog serveru. (Leskiw, 2016)



Obrázek 3: Syslog

Zdroj: http://www.networkmanagementsoftware.com/what-is-syslog/

4.2.3 SSH

SSH zkráceně Secure Shell, je softwarové zařízení pro zabezpečení síťového spojení. Protokol SSH prošel několika verzemi SSH-1.3, SSH-1.5 a SSH-2 a je založen na architektuře typu klient/server. Spojení probíhá protokolem TCP. Veškerá komunikace je bezpečně chráněna před narušením, účastníci na obou koncích jsou autentičtí a máme autorizovaný přístup k uživatelským účtům. Při posílání dat se před odesláním zašifrují pomocí SSH protokolu a když dorazí k příslušnému uživateli, jsou automaticky dekódována. Libovolná data poslaná přes takovéto spojení dorazí beze změny a nenaruší je ani přečtení příjemce. SSH klienti komunikují se servery prostřednictvím zašifrovaných síťových připojení tzv. tunelů. Protokol SSH nám zajišťuje autentizaci, šifrování a integrita dat při posílání dat přes síť.

Autentizace znamená ověření identity a určuje identitu uživatele, jestliže se pokusíme přihlásit k nějakému účtu na vzdáleném počítači. SSH pošle žádost o digitální důkaz naší identity, pokud projdeme tímto testem, máme povolení se přihlásit. Každé spojení obsahuje dvě autentizace. Klient ověřuje identitu serveru SSH (autentizace serveru) a server ověřuje identitu uživatele, který požaduje službu.

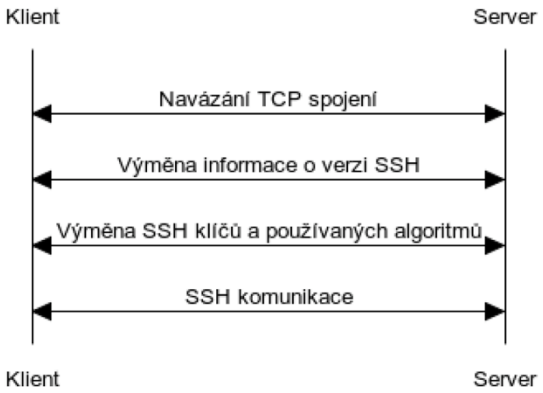
Autorizace probíhá pro autentizaci uživatele, protože není možné přidělit privilegia dříve, než zjistíme, o koho se jedná.

Integrita nám při přenosu dat po síti zaručuje, že data dorazí na místo určení beze změny. Pokud se třetí osoba pokusí zachytit a jakkoliv změnit data, SSH tuto skutečnost pozná. Protokol SSH-2 používá kryptografickou kontrolu integrity, jež zajišťuje, že přenášená data nemohou být změněna. Využívá hashovací algoritmy vycházející z MD5 a SHA-1. SSH1 využívá jen 32 bitovů kontrolu (CRC-32), která je uplatňována na nešifrovaná data v každém paketu.

Šifrování nám zakóduje data tak, že jsou nečitelná pro všechny s výjimkou určitých příjemců. Tímto způsobem jsou data chráněna při přenosu po síti. (Barret, 2003)

Autentizace SSH klienta

Server zrychluje ověřování tím, že sdělí klientovi, které autentizační metody mohou být použity pro pokračování výměny klíčů v daný okamžik. Klient má svobodu vyzkoušet metody uvedené serverem v libovolném pořadí. To dává serveru úplnou kontrolu nad ověřováním procesů v případě nějaké potřeby, ale také poskytuje dostatečnou flexibilitu pro klienta používat metody, které podporuje, nebo které jsou z nabídky serveru pro uživatele nejvhodnější. (barret)



Obrázek 4: SSH navázání komunikace

Zdroj: (Barret, 2016)

5 Výběr monitorovacího systému

Monitoring může využít licencované produkty nebo se nabízí možnost bezplatných řešení tzv. open-source, kde je zapotřebí investovat pouze čas a znalosti. Pro porovnávání monitorovacích systému využijeme komerční i nekomerční. Nabídka produktů na trhu nabízí pro použití širokou škálu. Komerční produkty nabízí vyšší úroveň supportu pro uživatele (např. telefonickou, dokumentovou podporu, která je lépe a podrobně vysvětlená), poskytují kompletní instalaci a uvedení do provozu ve společnosti podle požadavků a potřeb. Nevýhodou těchto produktů pro firmy je, že jsou zpoplatněné skrze licence.

Produkty zdarma nabízí univerzální a širokou možnost konfigurace, ale vyžadují hlubší znalosti pro nastavování a psaní vlastních skriptů. Výhoda je, že si sestavíme komplexní přehled o prostředí, které sledujeme. Nekomerční produkty mají výhodu v tom, že jsou open-source a jsou k dispozici ke stažení na internetu zdarma. Disponují velkými komunitními portály, kde se každý s každým může podělit o své vědomosti a je možné, že zde narazíme na řešení problému, se kterým si nemůžeme poradit.

Víceméně všechny open-source monitorovací systémy poskytují své vlastní stránky, kde si člověk může stáhnout potřebné pluginy, šablony, grafy, skripty apod. Podle své potřeby v síti se pak mohou dále upravovat. Některé nekomerční produkty poskytuji placený support pro instalaci a kompletní nastavení monitorovacího systému v síti. Monitorování pak bude probíhat pomocí nastavení SNMP nebo instalací agentů jak na sledovaném serveru, tak na serveru monitorovacího systému.

Výběr monitorovacího systému pro bakalářskou práci u firmy Aliacte s.r.o. probíhal na základě konzultace s administrátory sítě. Bylo nutné splnit náročnost monitorování serverů a serverových služeb v rámci velikosti firmy, jejich možností, jak z pohledu finančního stavu, tak z hlediska disponování fyzickými zařízeními. Výběr softwarového zařízení probíhal z open-source řešení. Pro výběr nejlepšího řešení pro firmu si porovnáme nejznámější systémy jako je Nagios, Zabbix, Zenoss, Cacti, OpenNMS a dále vezmeme v úvahu komerční řešení od IBM a OPSview.

5.1 Nagios

Charakteristika

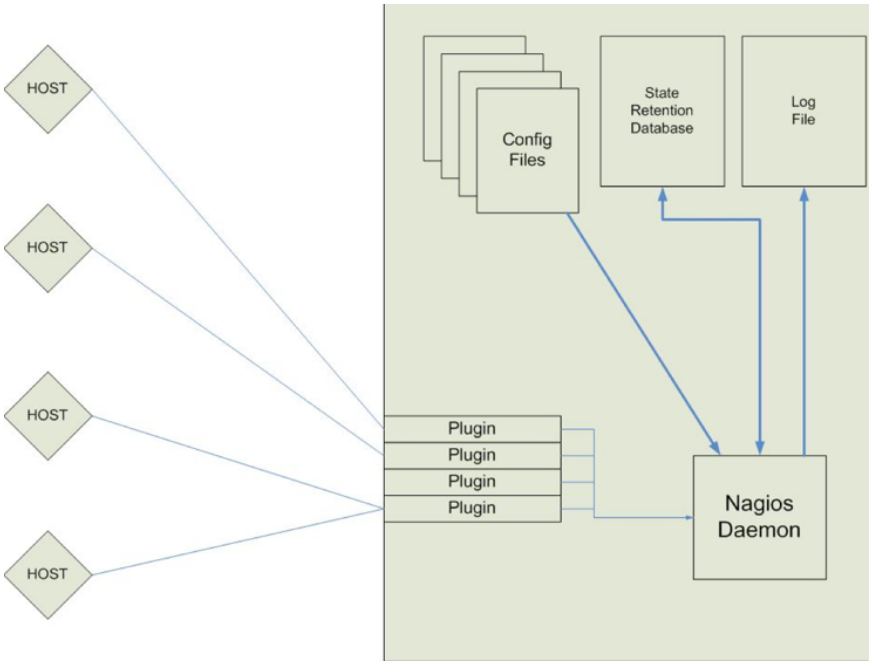
Nagios je jeden z monitorovacích systémů, který umožňuje sledovat počítače, servery a další zařízení v síti. Konstantně kontroluje servisy a procesy všech strojů tak, aby pracovaly bez sebemenších problémů. Přijímá stavy procesů a vykazuje jejich stav přímo na webový server, kde hlásí chyby, pokud dojde k přetížení, či nedostupnosti. Servis je však špatně nakonfigurován nebo systém nepracuje správně.

Monitorovací systém je rozdělen do dvou kategorií – hostitelé a služby. Hostitelé nám představují fyzické nebo virtuální zařízení v síti (servery, routery,

switche, tiskárny atd.). Služby jsou konkrétní funkce jako SSH, FTP, SFTP, SMTP, HTTP. Každá služba je tak spojena s hostitelem, na kterém je spuštěna. Kromě toho lze rozdělit zařízení do hostitelských skupin. Pro kontrolu a oznámení stavů používá čtyři stavy OK, Warning, Critical a Unknown. V poznámce za tímto stavem je jednoduchý popis problému. Podstata monitorování Nagios je založena na pluginech, které jsou volně dostupné na internetu, nebo si je firma může napsat sama dle vlastních potřeb. Portál Nagios obsahuje velké množství, které jsou volné dostupné pro všechny. (Kocjan, 2014)

Architektura

Centrálním mozkem Nagios je jeho daemon tzv. agent (NRPE) viz. Nagios architektura, který se stará o veškeré plánování a spouštění jednotlivých testů pro ověřování dostupnosti serverových služeb s použitím pluginů. Instalace Nagios může být provedena na webovém serveru typu apache, nginx nebo caddy. Definice pro monitorování jsou uloženy v konfiguračních souborech, v nichž si daemon najde podstatné informace pro konkrétní monitorování. Výsledky vyhodnocených testů ze zařízení a jejich služeb mohou být uloženy v log souborech, nebo, pokud použijeme rozšiřující moduly jako relační databázi, lze pro přístup a výpis k jednotlivým výsledkům monitorování použít prohlížeč. (Kocjan, 2014)



Obrázek 5: Nagios architektura

Zdroj: (Ammon, 2007)

Práce s produktem

Práce s Nagiosem je velmi flexibilní, může být nakonfigurován podle představ a potřeb IT infrastruktury. Disponuje mechanismem, který automaticky reaguje na problémy v síti a oznamuje je IT administrátorům pomocí sms či emailů. Rozesílání lze rozdělit do skupin a kontaktů definovaných v Nagiosu. Umožňuje definovat, že určitá služba závisí na jiné službě, a to buď na stejném nebo jiném hostitelském systému. Nagios nabízí ucelený systém definic maker. Jedná se o proměnné, které mohou být uvedeny u všech definic objektů, využívají se uvnitř příkazů v závislosti na hostiteli, servisu. Nagios rovněž nabízí mechanismus pro stanovení časového rámce odstávky. Ten se používá především, je-li potřeba provést údržbu IT infrastruktury serverů a služeb, které sledujeme. Umožňuje sledování všech stavů pomocí webového rozhraní a nabízí možnost podívat se na historii logovacích souborů a událostí, které nastaly. (Barth, 2008)

Konfigurační soubory

Po instalaci máme veškeré konfigurační soubory uvedené na defaultní hodnoty, proto je doporučením, při konfiguraci si udělat zálohu konfiguračních souborů. V níže uvedených konfiguračních souborech byli prováděny změny pro správnou komunikaci a konfiguraci nagiosu pro monitoring.

- **Htpasswd.users -** Seznam uživatelů, kteří mají webový přistup do Nagios a mohou sledovat stavy monitoringu;
- **Cgi.cfg -** Hlavní konfigurační soubor pro nastavování webového přístupu, který obsahuje podstatné cesty k hlavním konfiguračním souborům, což jsou např. html, css a php soubory, při využívání webového prostředí Nagios;
- **Nagios.cfg -** Hlavní konfigurační soubor Nagios – jsou zde definovány další cesty k pluginům, serverům, switchům a logovacímu souboru;
- **Nrpe.cfg -** Soubor, který slouží pro správnou komunikaci NRPE agenta se vzdálenými servery a pluginy s jejich cestou uložení, které může NRPE využívat;
- **Commands.cfg -** Soubor se využívá pro konfiguraci pluginů a spravné navázání komunikace se vzdálenými servery při definování hostů;
- **Contacts.cfg -** Konfigurační soubor slouží pro vytvoření kontaktů, kterým se můžou odesílat různorodá upozornění o služby při výpadku serverů;
- **Templates.cfg -** V tomto souboru se vytváří se šablony pro linuxové, windowsové servery, hosty. S jejích pomocí si ulehčíme práci v dalších konfiguračních souborech.

Obrázek 6: Nagios dashboard

Zdroj: vlastní zhotovení screenshotu z provozního systému Nagios pořízený dne 20.4.2017

5.2 Zabbix

Charakteristika

Zabbix lze definovat jako distribuovaný monitorovací systém s centrálním webovým rozhraním, na kterém můžeme řídit téměř vše. Mezi jeho hlavní rysy patří:

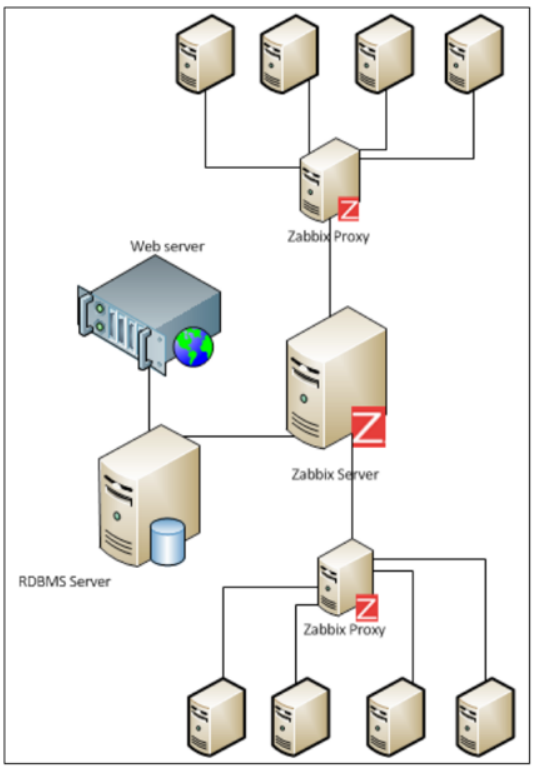
- Centralizované webové rozhraní, kde se provádí téměř všechny konfigurační úlohy;
- Všechna data ukládá do relační databáze;
- Server lze provozovat na většině unixových operačních systémů;
- Má velmi flexibilní konfiguraci a nastavení;
- Lze jim snadno integrovat s jinými systémy, přizpůsobitelný;
- Může být monitorován pomocí SNMP v1, 2 a 3.

Jeho základním cílem je sběr dat, kde jsou shromážděné údaje zpracovány a uloženy pro případné budoucí srovnávání. Data se porovnávají pomocí tzv. triggerů. Zabbix se skládá ze čtyř částí – server, proxy, databáze a webový prohlížeč. Každý z nich má své specifické vlastnosti. (Vacche, Lee, 2015)

Architektura

Zabbix architektura by měla být nakonfigurována na stejném fyzickém nebo virtuálním serveru, a to webový server apache nebo nginx, Zabbix server

a RDBMS (relational database management system) server. Významnou roli při rozsáhlé firmě má Zabbix agent a Zabbix proxy.



Obrázek 7: Zabbix architektura

Zdroj: (Vacche, Lee, 2013)

Zabbix Server

Zabbix Server je ústředním prvkem, ke kterému agenti hlásí svoji dostupnost a integritu, k němuž poskytují informace, data a statistiky. Tento server je centrálním úložištěm, v němž jsou všechna nastavení statistická a operační data uložena.

Zabbix agent

Zabbix agent je nasazen na sledování vzdálených cílů, aby se aktivně monitorovali jeho zdroje a aplikace a oznamoval shromážděná data Zabbix serveru.

Zabbix proxy

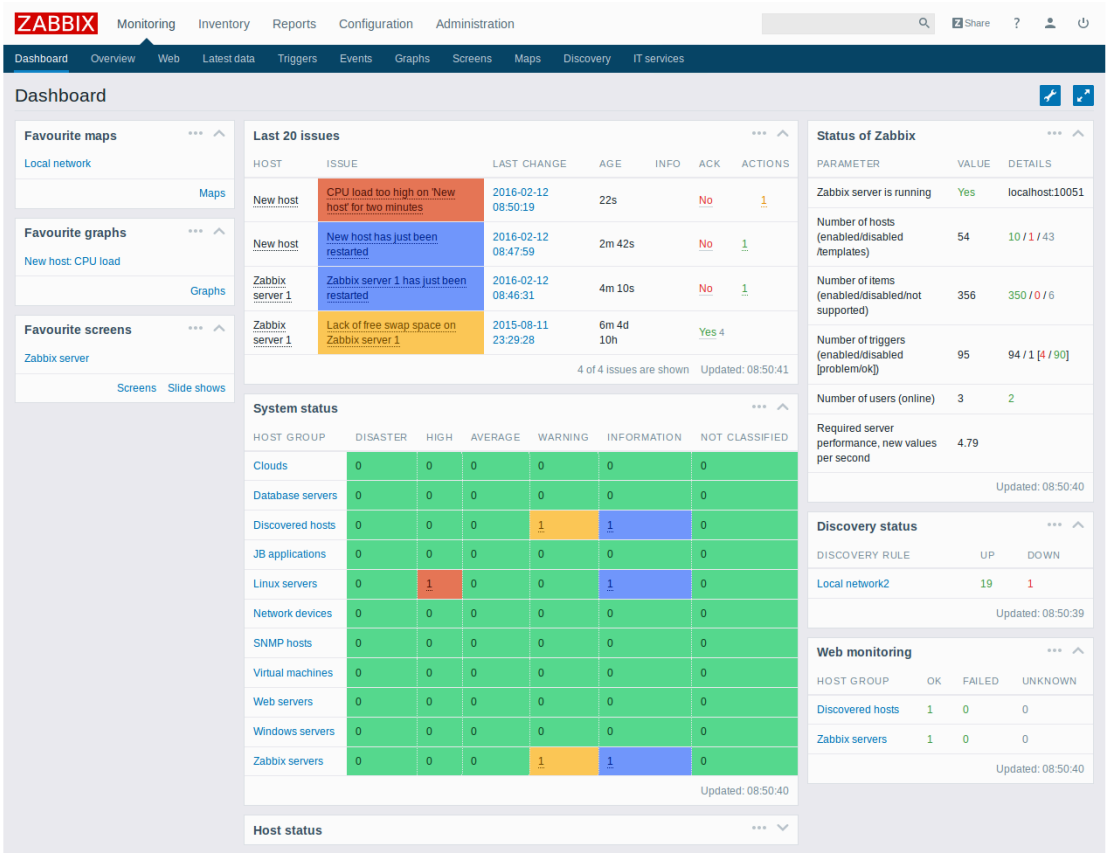
Zabbix proxy může shromažďovat údaje o výkonu a dostupnosti jménem Zabbix server. Proxy je volitelnou součástí nasazení, nicméně může být velmi prospěšné pro distribuci zatížení jednoho Zabbix serveru. Spolu se serverem tvoří hlavní složku – řídí všechny pravidla (collections, triggery, upozornění atd.).

Webový server

Webový server apache nebo nginx slouží ke snadnému přístupu odkudkoliv při použití jakékoliv platformy. Obvykle běží na stejném fyzickém počítači jako server, zobrazí se nám zde všechna data získaná serverem.

Práce s produktem

Zabbix dashboard poskytuje přehled o celkových sledovaných stavech, systémech, o stavu Zabbix serveru a poskytuje seznam všech problémů, které se nedávno uskutečnily, a také síťovou mapu IT infrastruktury. Zabbix poskytuje pluginy pro sledování tzv. templates, které jsou napsány v jazyku XML.Na internetu jsou volně dostupné šablony pro sledování různých zařízení, včetně serverů, operačních systémů, služeb, hardwarových stavů, síťových zařízení, UPS, webových stránek a dalších složek IT infrastruktury. Vše vyžaduje konfiguraci, ale prvotní je nainstalovaní Zabbixu na server pomocí distribučních balíčků nebo zdrojových kódů. (Vacche, Lee, 2013)



Obrázek 8: Zabbix dashboard

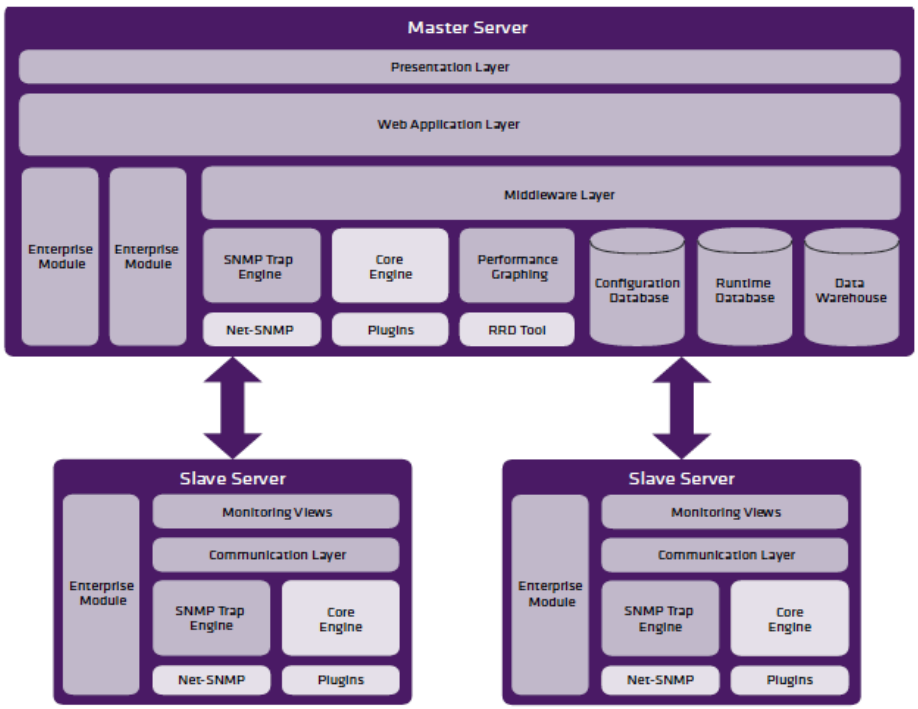
Zdroj: http://www.zabbix.com/img/screenshots/3.0/monitoring/Dashboard.png

5.3 OPSview

Charakteristika

OPSview je platforma pro monitoring IT infrastruktury, která poskytuje vylepšenou kontrolu. Systém lze snadno nainstalovat a nakonfigurovat a přináší vylepšené funkce ve vysoce škálovatelném komerčně podporovaném systému. Oproti Nagiosu nevyžaduje vysoce specializované znalosti nebo další investice do drahých speciálních hardwarů. Nevyžaduje ani integrační dovednosti skrze instalace, konfigurace a případnému upgradu systému. Jako veškeré monitorovací systémy dovede sledovat hostitele a jejich služby i pohotovostní problémy uživatelů. Komerční verze OPSview nabízí vstupní body pro všechny typy a velikosti podniků. Podniky mohou získat OPSview, který závisí na celkovém počtu sledovaných hostů. Počáteční cena začíná na 900 € pří 50 sledovaných hostech. Maximální počet sledovaných hostů je 300 a to při ceně 5640 €. Dané řešení je možné považovat za cenové dostupnou záležitostí pro střední firmy, pro malé podniky je však cenově nedostupnou záležitostí. Na svém portálu nabízí tzv, Web-Based Training na školení svého produktu, za který se jednorázově zaplatí 615 €. (Jackgckoros, 2014)

Architektura



Obrázek 9: OPSview architektura

Zdroj: (OPSview, 2012)

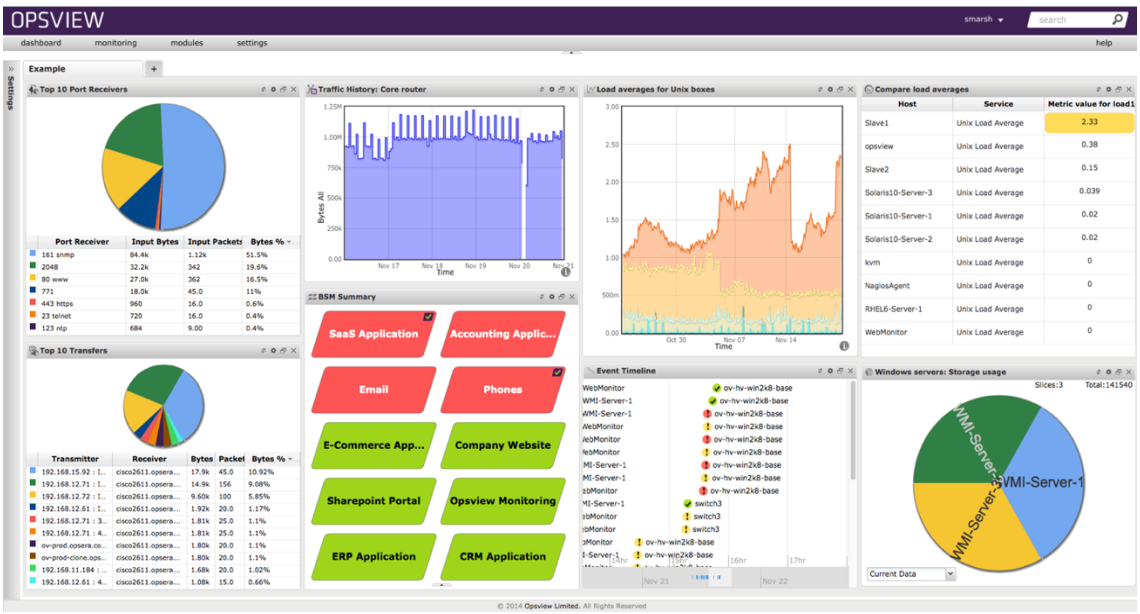
Prezentační služba OPSview kombinuje dynamické dashboardy s funkcí drag & drop, výsledky z monitorování a uživatelsky lehkou konfiguraci. Vše je dostupné pomocí webového rozhraní. Webová aplikační vrstva je podporována frameworkem typu Catalyst MVC, který je napsán pomocí jazyku Perl. Vrstva poskytuje rozhraní pro programovaní aplikací (API), monitorovaní dat a konfiguraci objektů. Middleware vrstva obsahuje framework, který nám poskytuje lepší flexibilitu a distribuování monitorovacích schopností. Nabízí několik důležitých funkcí:

- Zprostředkovává komunikaci mezi moduly a komponenty;
- Vytváří, ověřuje a distribuuje konfigurace ve formátu, který je srozumitelný pro každou složku;
- Zajišťuje správu mezi Master a Slave servery.

Slave servery mohou být organizovány do clusterů, které poskytují automatické vyrovnávání zatížení a při výpadku serveru jsou schopni pokračovat v monitorování autonomně. Komunikace mezi Master a Slave je zřízená přes šifrovaný tunel službou SSH. Jádro je 100% kompatibilní a postavené na Nagios, tím se zajišťuje snadná aktualizace softwaru a přístup k široké škále dostupných pluginů. Využívá agenta od Nagiosu tzv. check_nrpe agent. (OPSview, 2012)

Práce s produktem

V OPSview dashboard OPSview dashboardlze rychle konfigurovat identické objekty pomocí funkce klonování, pomáhá udržovat kontrolu nad složitými konfiguracemi, kde místo více služeb pro jednotlivé hostitele, může vytvořit jeden tzv. check a využít ho pro každý prvek, který je zapotřebí sledovat stejným způsobem. Disponuje vlastností samostatně najít zařízení, které vykazuje SNMP trapy a tím si samostatně přidá a nakonfiguruje zařízení do sledovacích možností. Poskytuje varovné upozorňování při kritických stavech pomocí emailu a sms zpráv. Předdefinované šablony pro mnoho typů hardwarů, operačních systémů, databází a aplikací jsou zahrnuty přímo. Veškeré předdefinované šablony můžeme upravovat podle vlastních potřeb. (Why we made Opsview Atom, 2015)



Obrázek 10: OPSview dashboard

Zdroj: https://www.opsview.com/sites/default/files/opsview%20atom%20dash.png

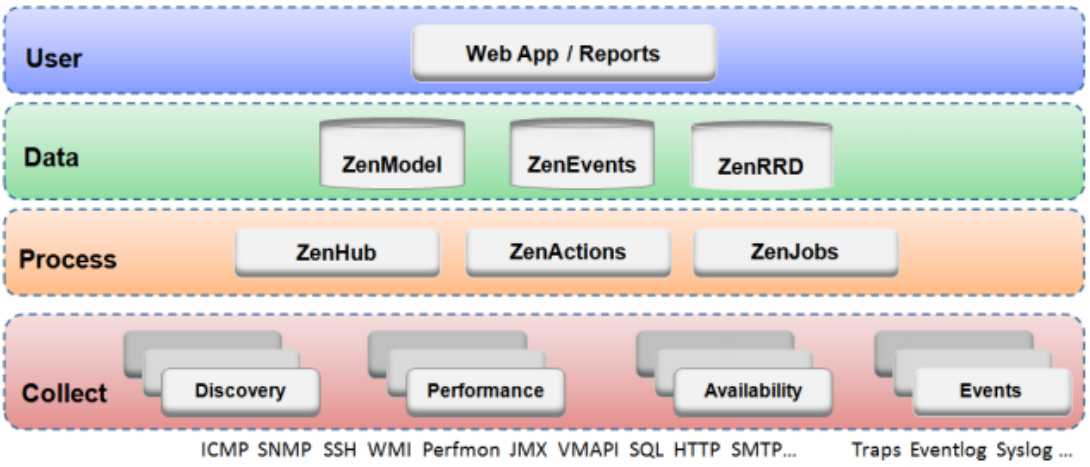
5.4 Zenoss

Charakteristika

Hlavní funkcí jádra pro správu řízení Zenossu je databáze pro správu konfigurace tzv. CMDB (configuration managment database), která ukládá infrastrukturu prostředí a historii změn. Jádro zenossu podporuje přidávání aktiva v CMDB jedno po druhém, nebo poskytuje možnost automatického odhalení aktivních zařízení, která jsou sledována pomocí služeb SNMP, SSH nebo skenování portů. Jádro nám umožňuje uspořádat a filtrovat výpis aktivních prvků definovaných uživatelem pomocí místa, skupin a systémů. Jádro nám poskytuje sběr dat ze vzdálených zařízení pomocí agenta. (Badger, 2011)

Architektura

Zenoss ukládá svoje data do MySQL databáze. Jádro Zenossu je rozděleno do 4 vrstev, které spolu navzájem komunikují.



Obrázek 11: Zenoss architektura

Zdroj: http://docs.huihoo.com/zenoss/admin-guide/2.4.2/resources/architecture.png

User layer

User layer webového aplikačního prostředí připravuje webový portál pro jeho správu. Pomocí uživatelského prostředí lze přistupovat a spravovat klíčové komponenty a funkce. Z webového prostředí lze sledovat a upravovat:

- Celkový stav podniku pomocí dashboard utiliy;
- Sledování systémových zpráv, uživatelů, událostí;
- Úprava sledovaných zařízení v síti;
- Vytvářet a spouštět reporty na zařízení.

Uživatelská vrstva komunikuje s datovou vrstvou a překládá informace k zobrazení.

Data layer

Data layer seskupuje informace o konfiguraci a sběru dat Zenoss core a ukládá je do tří samostatných databází:

- ZenRRD – využívá RRDtool, který ukládá data do RRD (Round Robin Database) databáze. RRD soubory jsou uloženy lokálně na každý kolektor, kde nám ze zápisu nevyplývají žádná omezení při vytváření nového kolektoru do databáze;
- ZenModel – slouží jako základní konfigurační model, který zahrnuje všechny zařízení včetně jejich komponentů a umístění;
- ZenEvents – ukládá všechna data od jednotlivých eventů do databáze MySQL.

Process layer

Process layer zahrnuje procesy řídící komunikaci mezi sbíráním a ukládáním dat. V pozadí zde běží procesy vykonávané uživatelem:

- ZenActions;
- ZenJobs.

ZenHub střídavě ukládá data na příslušné místo do databáze MySQL. Zenoss core generuje události, pokud byla překročena stanovená hodnota jako je např. vysoké využití pamětí, CPU. Události spouští jednotlivé akce jako je např. odeslání emailu požadovaným administrátorům. RRD databáze se liší od MySQL v tom, že je kruhová, což znamená, že velikost databáze se zvyšuje v průběhu času.

Collect layer

Vrstva obsahuje daemony, kteří shromažďují informace od zařízení v síti. Daemoni vykonávají funkce pro modelování, sledování a řízení událostí. Modelový systém využívá služby ke shromažďování informací ze vzdálených zařízení pomocí SNMP, SSH a WMI (Windows Management Instrumentation). Informace o dostupnosti a stavu služeb jsou vráceny prostřednictvím ZenHub do daemonu Events, kde můžeme generovat upozornění, oznámení nebo spustit vlastní skript pro upozornění. Primární sbírání informací probíhá pomocí protokolu SNMP, avšak informace lze také shromažďovat pomocí skenování portů nebo pluginů.

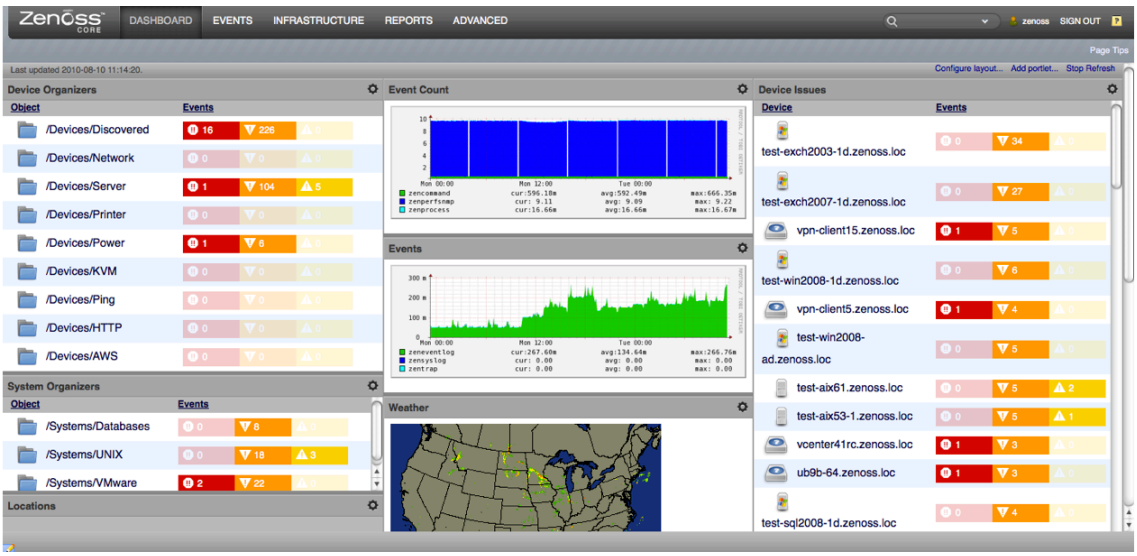
Tabulka 1: Popis zenoss daemonu

Device daemon	**Popis**
Zenmodeler	Dotazuje se zařízení přes SSH / Telnet, SNMP, a skenuje porty, když přidáváme nové zařízení. Pokaždé, když zenmodeler běží na zařízení, porovná svá zjištění s existujícími konfiguracemi a aktualizuje.
Zendisc	Spouští request k vyhledání nového zařízení nebo sítě
Performance daemon	**Popis**
Zenperfsnmp	Ukládá sesbíraná data v RRD souborech, tak RRDtool může poskytnout vykreslení grafu o zařízení v rozdílném časovém trvání např. hodinové, denní, měsíční, roční.
Zencomand	Poskytuje způsob, jak spouštět vlastní skripty a zásuvné moduly třetích stran, včetně od Nagiosu a Cacti pluginů uvnitř Zenossu.
Zenprocess	Monitoruje procesy v systémech Linux, UNIX a Windows.
Zenping	Posílá ping na zařízení a poskytuje zprávu o aktivitě zařízení, zda je dostupné.
Zenstatus	Testuje TCP porty a hlásí, zda jsou služby na portech aktivní.
Event daemon	**Popis**
Zensyslog	Vytváří eventy ze syslog messages.
Zeneventlog	Vytváří eventy z windows zařízení a jejich logů.
Zentrap	Vytváří eventy z protokolu SNMP. Pokud se vyskytne problém na monitorovaném zařízení, generuje se SNMP trap, aby se upozornil Zenoss o problému.

Zdroj: (Zenoss core administration, 2014)

Práce s produktem

Veškerá konfigurace probíhá přes webové prostředí Zenoss dashboard, kde lze organizovat zařízení podle třídy, lokace, systému a skupiny. Další typ rozdělení je pomocí služby sledování a to SNMP, WMI, zenoss plugins, ICMP, portů, skriptů a dokáže oznamovat všechny nadefinované chybné stavy pomocí emailu daným uživatelům nebo skupinám uživatelů. Lze používat Nagios nebo Cacti pluginy. Zenoss podporuje postupné přidávání jednotlivých uživatelů do CMDB nebo je možné použít auto-discovery pomocí routovací tabulky. (Zenoss core administration, 2014)



Obrázek 12: Zenoss dashboard

Zdroj: https://upload.wikimedia.org/wikipedia/commons/3/38/Zenoss_Core_Dashboard.png

5.5 Cacti

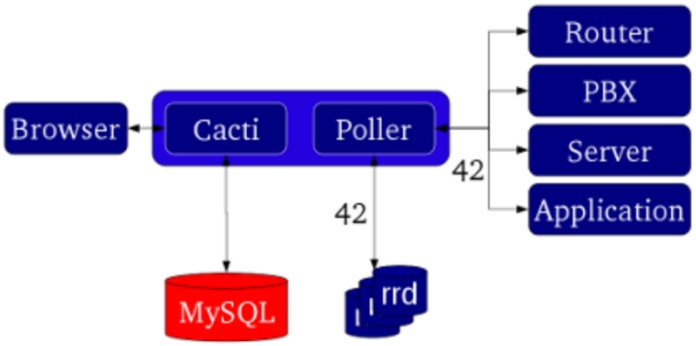
Charakteristika

Cacti je open source síťový monitorovací a grafický nástroj napsaný v PHP/MySQL. Používá RRDtool engine pro ukládání dat a generování grafického výstupu, kde pravidelně sbírá data přes Net-SNMP. U Cacti není nutné používat pouze SNMP, lze si napsat své vlastní skripty v Perlu nebo Shellu. Považuje se za jeden z nejlepších RRDtool front-end monitorovacích nástrojů. Operace jsou rozděleny do tří rozdílných úloh a to sbírání, uložení a prezentace dat. Sbírání dat probíhá skrze tzv. pooler, který pomocí SNMP sbírá informace o zařízeních v síti. Obsahuje vestavěnou funkci grafů, díky čemuž je schopen vykreslovat grafy na základě sběrů dat pomocí SNMP. Je možné mít v grafu více položek, přidávat různé legendy a nastavovat automatická měřítka.

Architektura

Všechna data se ukládají do databáze pomocí nastavení časového intervalu za pomoci aplikace Pooler viz Cacti architektura, která je nastavena jako tzv. plánovač operačního systému pro sběr dat. RRDtool je systém, který je schopen ukládat informace od rozdílných zařízení, poskytuje velmi výkonný systém pro zaznamenávání dat a grafů. Je také známy jako databázový nástroj round-robin, jehož řešení je open-source. Všechny shromážděné informace prostřednictvím RRDtool jsou ukládány do souborů s názvem RRA. Pro sběr těchto informací do RRA potřebujeme určité parametry s cílem, proto kdykoliv chceme přidat nové zařízení nebo vytvořit graf musíme zadat parametry ručně. Umožňuje

to administrátorovi analyzovat shromážděná data ze všech typů datových zdrojů, které jsou schopné odpovídat na dotazy SNMP. Nabízí několik typů přepínačů pro přístup a manipulaci s tzv. rdd soubory.

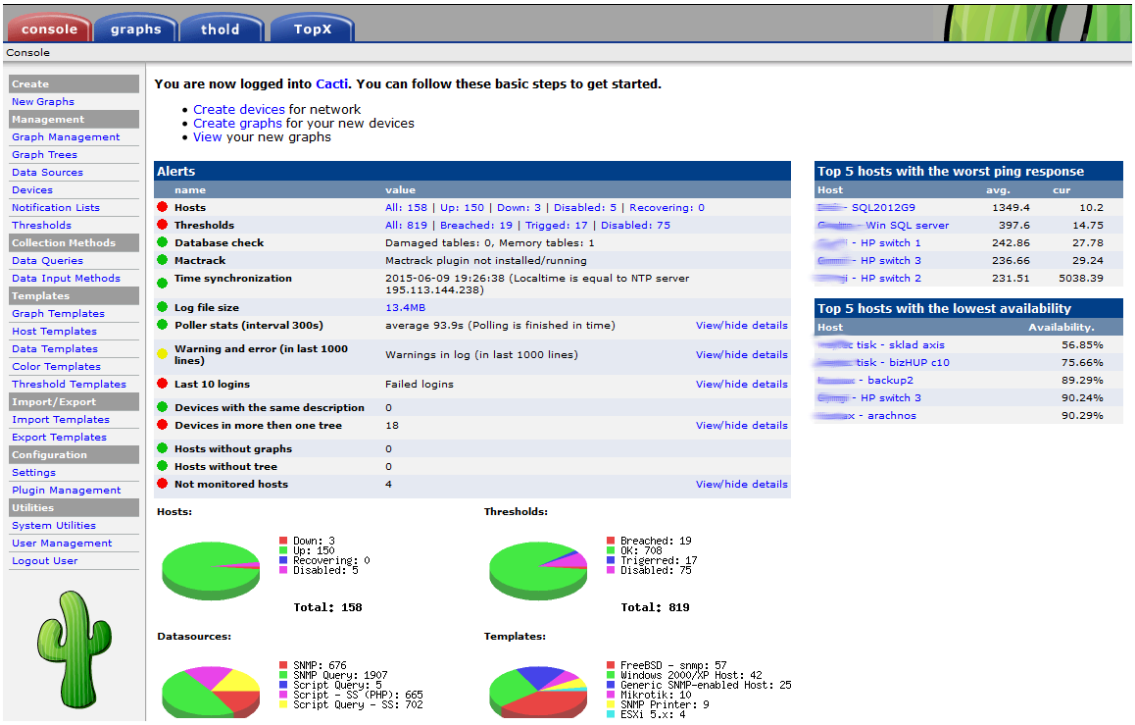


Obrázek 13: Cacti architektura

Zdroj: http://docs.cacti.net/manual:088:2_basics.0_principles_of_operation

Práce s produktem

Cacti zpracovává data jako např. šířku pásma v síti, zatížení CPU, serverů či pokojovou teplotu. Dále umožňuje sledování zařízení, jako jsou routery, UPS apod. Pro rychlejší a uživatelsky přijatelnější verzi je možné použít šablony, které lze importovat a exportovat pomocí Cacti dashboard. Cacti poskytuje na své wiki vlastní šablony pro více druhů a typů zařízení, jež lze stáhnout zdarma. Rozlišují se dva způsoby rozdělení pro hosta a grafy. Šablona host znamená související kolekci šablony grafů a datových dotazů, které souvisí s konkrétním typem hosta. Grafy jsou použity k vizualizaci dat, která jsou shromážděna a která lze je použít pro více typu systému a zařízení. Pro přístup do administrace je možnost přidávat nové uživatele a omezovat je skrze jejich práva. (Kundu, 2009)



Obrázek 14: Cacti dashboard

Zdroj: http://docs.cacti.net/_media/userplugin:intropage_1.png

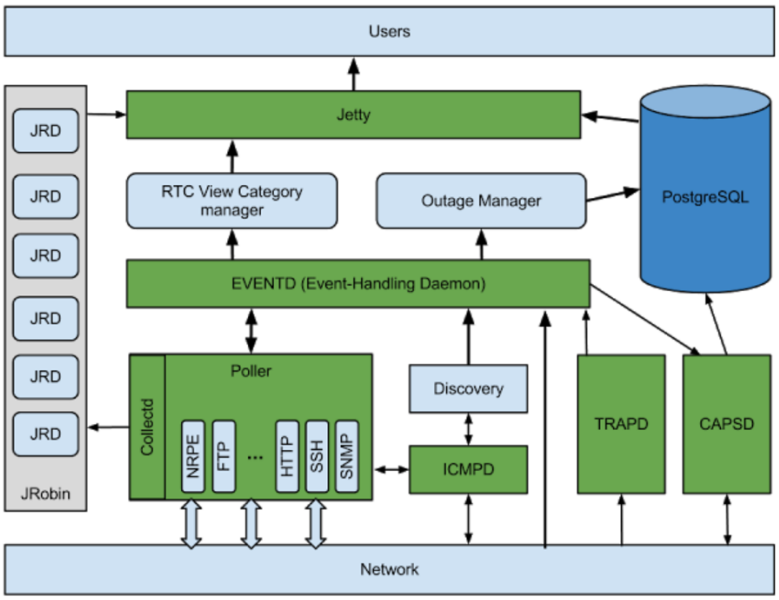
5.6 OpenNMS

Charakteristika

OpenNMS je platforma využívaná pro správu a sledování síťových aplikací a služeb sestavená pomocí programového jazyku Java. Vykonává všechny funkce pro správu sítě, které jsou definované v ITU včetně jejich zásad pro telekomunikační sítě včetně zprávy poruch, konfigurace, účetnictví a administrace neboli často zkráceně FCAPS. Účinně dokáže monitorovat všechny zařízení v síti a jejich údaje mohou být shromažďovány pomocí SNMP. Všechna data jsou uložena, poskytují statistiky využití zařízení, které mohou být dále vykreslovány pomocí grafů na webovém rozhraní.

Architektura

Pro OpenNMS existují dva způsoby, jak sbírat a ukládat data. Prvním z nich je sběr prostřednictvím dotazování přes tvz. pooling., který vykonává testy pro dostupné služby jako ICMP, DNS, FTP, HTTP, SSH. Druhý způsob sběru dat je pomocí SNMP, kdy jsou data ukládána pomocí RRDtool.

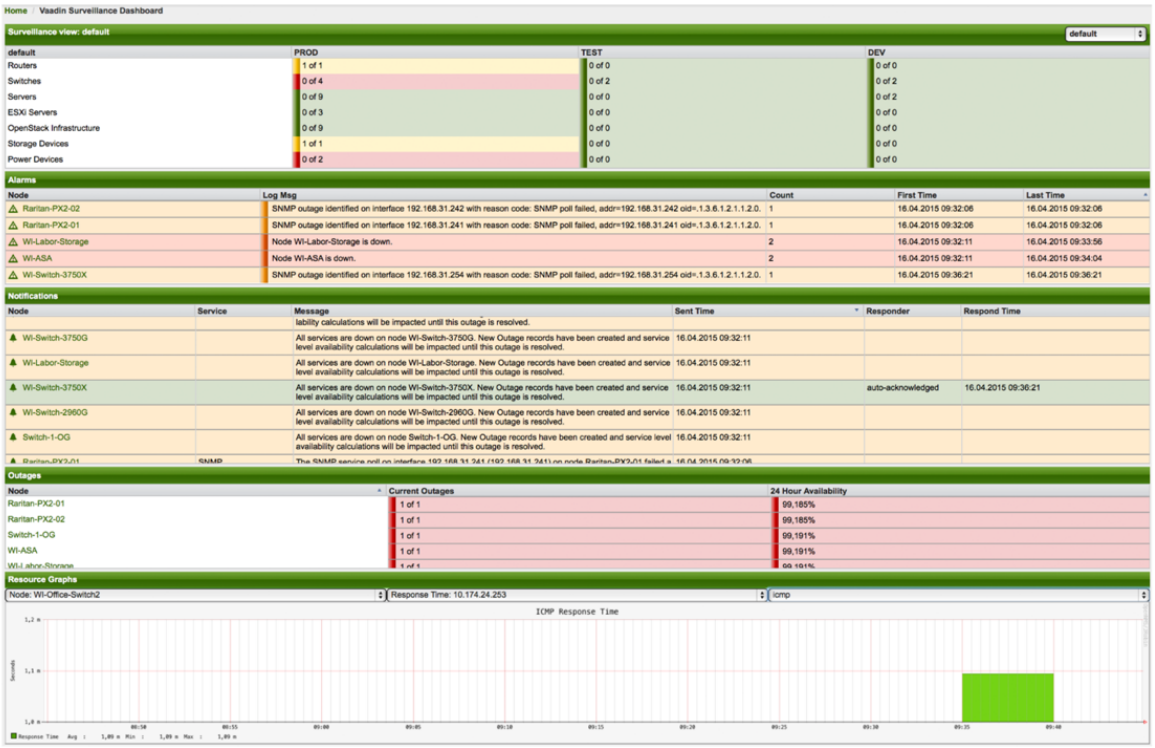


Obrázek 15: OpenNMS architektura

Zdroj: http://oopsmonk.blogspot.cz/2013/02/opennms-architecture-introduction.html

Práce s produktem

OpenNMS dashboard umožňuje plnou kontrolu většiny potřebných funkcí, které umožňuje základní nastavení. Eventy s negativním důsledky jsou na síti monitorovány, lze je identifikovat, zaznamenávat nebo mohou být i předvídány. Dojde-li k chybě jsou prostřednictvím předdefinovaných akcí oznámeny. Mohou být zaslány přímo z opennms několika způsoby a to prostřednictvím sms, emailů či tiketů. Poskytuje způsob, který umožňuje lépe zaznamenat a aktualizovat uzly a síťové prvky v síti. Pro zjištění údajů o výkonnosti prvků a zařízení v síti se využívá SNMP a JMX (Java managment extension) kompatibilní software. Data jsou ukládána do databáze pro vykreslování grafů na webovém rozhraní. Autentizace a autorizace uživatelů může probíhat pomocí lokálních uživatelů nebo LDAP a RADIUS serveru. (Docu-overview, 2017)



Obrázek 16: OpenNMS dashboard

Zdroj: http://docs.opennms.org/OpenNMS/snapshot/develop/documentation/guide-user/

5.7 IBM Tivoli V6.2

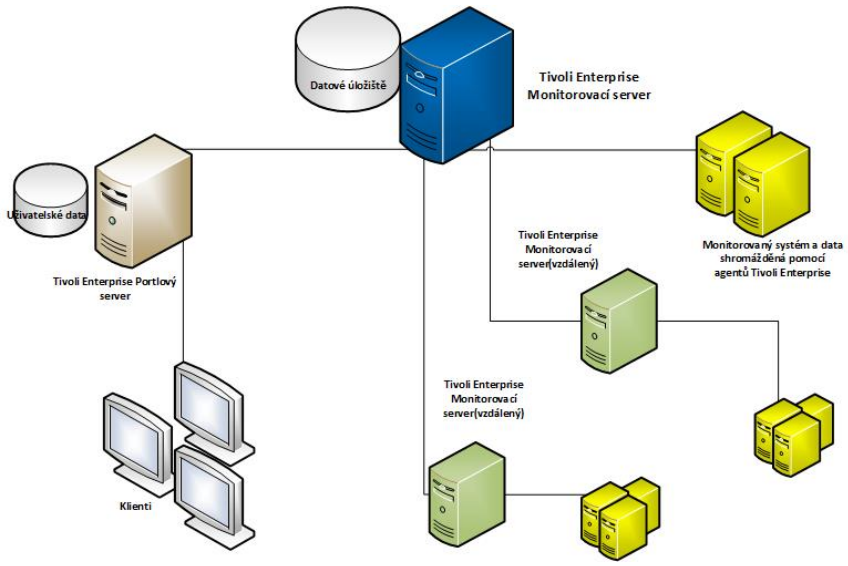
Charakteristika

IBM Tivoli monitoring V6.2 je produkt pro monitoring výkonu a dostupnosti distribuovaných operačních systémů a aplikací v síti. Zabezpečuje ukládání firemního majetku a disponuje systémem se softwarem pro správu sítě, který umožňuje snížit složitost technologií, prostřednictvím integrace procesů IT. Díky automatizaci podnikových služeb, která je známá jako Service Management firmy může být urychlen tok zpracování dat, a to umožňuje uvolnění technologické zdrojů pro další projekty. Service Management zahrnuje řídící procesy a taktiku, zaměřuje se na vývoj, nasazení a správu služeb. To pomáhá snižovat IT provozní náklady na automatizaci procesů a účinněji spravovat jejich systém dodržování.

Architektura

Tivoli Enterprise Monitoring Server je klíčovou komponentou, která obsahuje architektonické prvky, které jsou na sobě vzájemně závislé. Funguje jako sběratel a kontrolní bod pro výstrahy situací, kde jsou nastavené politiky skrze agenty připojené k serveru. S jejich pomocí se sbírají data o výkonnosti a dostupnosti monitorovacích serverů. Agenti jsou zodpovědní za shromažďování údajů

a distribuci atributů k monitorovacím serverům. Tivoli Enterprise Portal server je úložiště pro všechny grafické prezentace z údajů. Poskytuje základní vyhledávání, manipulaci, analýzu a reformátování dat přes webové rozhraní.



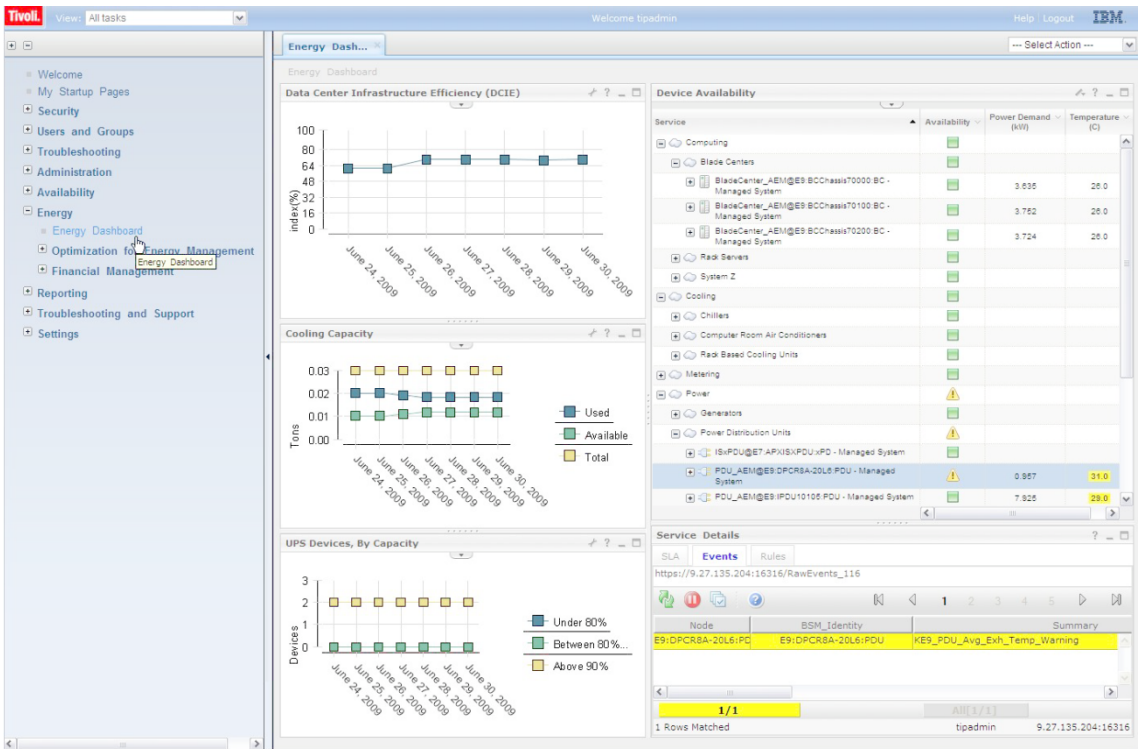
Obrázek 17: IBM Tivoli

Zdroj: (Gucer, Altaf, Anderson, et al., 2008)

Práce s produktem

Monitorování aplikací a serverů jsou silná a mohou generovat velké množství informací v různých vizualizovaných formátech. Všechny potřeby jsou pro úplný přehled zahrnuty v následujících kategorií.

Definování a určení sběru dat se provádí pomocí kategorie reports, kde definujeme služby a síťové prvky pro sledování. Kategorie alert se využívá na druhy událostí k zasílání uživatelům. Poslední kategorie actioin se užívá k definování akcí, jež budou prováděny automaticky. Ke každému datovému souboru je vytvořen inventář, který obsahuje časové údaje, kdy mají být data shromážděna a v jakých intervalech. Existují intervaly pro vzorkování krátkých i delších situací. Intervaly delších situací mohou mít významný dopad na režii, ale krátké situace by měly být vyhrazeny pro velmi kritické aplikace a jejich nejkritičtější výstrahy. (Gucer, Altaf, Anderson, et al., 2008)



Obrázek 18: IBM Tivoli dashboard

Zdroj: https://www.ibm.com/support/knowledgecenter/SSSPFK_4.2.1.3/images/bsmc_eng_dashboard.jpg

6 Praktické řešení

Výběr a porovnávání monitorovacího systému probíhalo na základě konzultace s administrátory sítě. Monitorovací systém musel splňovat následující potřebné vlastnosti viz tabulka Přehled monitorovacích systémů, ostatní vlastnosti monitorovacího systému nebyli podstatné při výběru. Výsledný monitorovací systém Nagios, který splňoval podmínky nasazení, byl následně nakonfigurován. Monitoring sítě umožnil sledovat předpokládaný rozsah i úroveň služeb od poskytovatele internetového připojení a také případné postihy za jejich nedodržení. Cílem SLA (Service Level Agreement) je zadefinovat míru kvality poskytovaných služeb a může taky zahrnovat způsoby sankcí. Penalizování poskytovatele nemá mít za cíl ušetřit na nákladech, ale preventivně motivovat poskytovatele, aby předcházel tomu a nedocházelo k výpadku poskytovaných služeb.

Tabulka 2: Přehled monitorovacích systémů

Produkt	Licence	Určení	OS	Agent	Plugins	Triggers/ Alerts	Syslog
Nagios	GNU GPL	Síťový, systémový, Aplikační	Windows, Linux	Ano	Ano	Ano	Ano
IBM Tivoli	Komerční	Síťový, systémový	Windows, Linux, Unix(AIX)	Ano	Ano	Ano	Ano
OPS view	Komerční	Síťový, aplikační	Linux, Solaris	Ano	Ano	Ano	Ano
Zenoss	GNU GPLv2	Síťový, systémový	Linux, VMware, OS X	Ano	Ano	Ano	Ano
Cacti	GNU GPL	Síťový	OS X, Windows, Linux	Ano	Ano	Ano	Ano
Zabbix	GNU GPLv2	Síťový, systémový, Aplikační	OS X, Windows, Linux	Ano	Ano	Ano	Ano
Open NMS	GNU GPLv3	Síťový, systémový, Aplikační	OS X, Windows, Linux	Ano	Ano	Ano	Ano

Zdroj: vlastní kompletace tabulky

6.1 Popis firemního prostředí a analýza

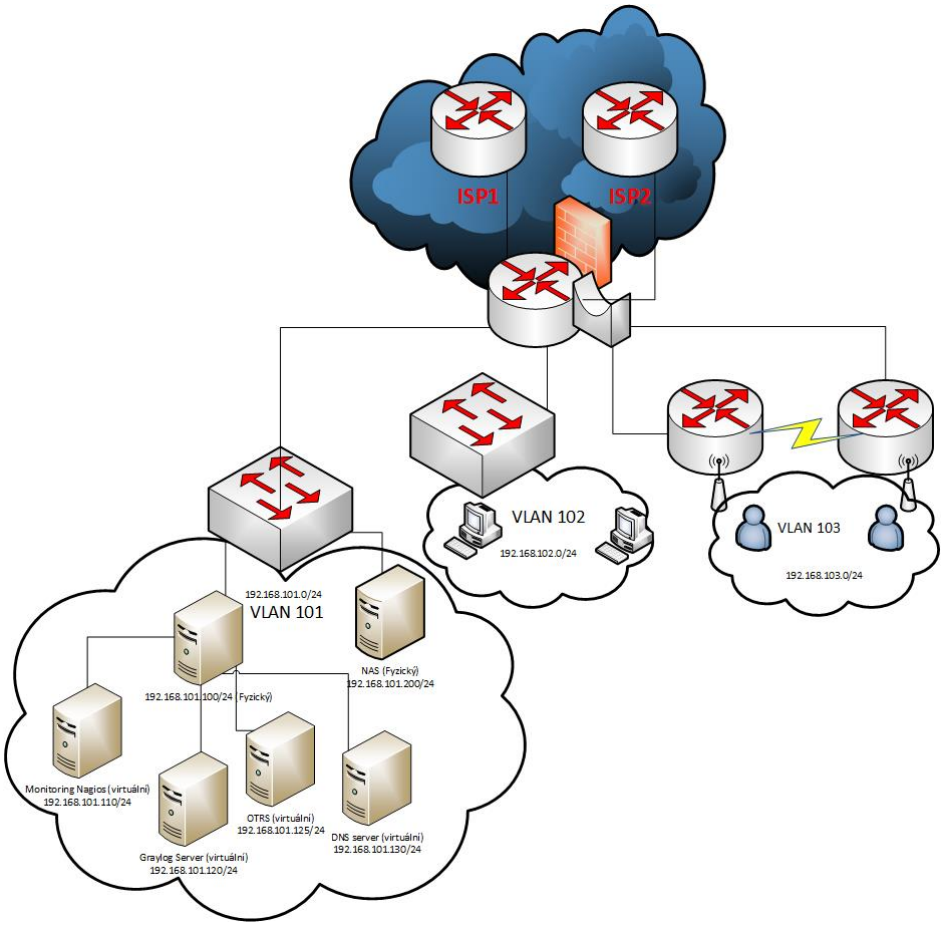
Analýza stávajícího stavu

- Firma Aliacte s.r.o.;
- Zabývá se správou infrastruktury a vývojem aplikací, řízení oběhu produktů a výrobků;
- Využívá dva rozdílné poskytovatele internetu UPC a Netdatacom s veřejnými adresami, pro případ výpadku sítě na lince;

- Firma využívá fyzickou infrastrukturu a HW prostředky:
  - Fyzický server Fujitsu primergy RX 300 S6;
  - Router Mikrotik Cloud Core 1016 12G;
  - Router pro wifi RB951-2HnD;
  - Switch Mikrotik Cloud Core 226-24G, HP 2530-24G;
  - UPS PAC smarat 100;
- Firma využívá i cloudovou infrastrukturu:
  - Veřejný CLOUD pro provozování aplikačního, databázového prostředí na provoz služeb;
  - Veřejný cloud office systémů Mircrosoft Office 365;
  - Interní cloud provozovaný na zařízeních firmy, virtualizovaný a sloužící k testování OS, DB, aplikací, ukládání a zálohování dat, provoz monitoring systémů;
  - Veřejný SaaS cloud určený na provoz verzovacího systému typu GitHUB;
- Firemní popis síťové infrastruktury:
  - Striktně jsou odděleny demilitarizované zóny pro provoz serverů a publikaci aplikací jak směrem do vnitřních sítí, tak do internetu;
  - Nasazené VLANy na oddělení serverových aplikací a uživatelských úrovní;
  - Vzdálený přistup pomocí VPN;
  - Zařízení v jednotlivých VLAN:
    - Zařízení Switch Mikrotik Cloud Core 226-24G 102 – připojení počítačů – 192.168.102.10-192.168.102.200/24;
    - 2x Router RB 951-2HnD 103 – wifi – 192.168.103.10-192.168.103.200/24;
    - Switch HP 2530-24G 101 – serverová část bude řešena statickými IP, ale pro případ bude nastaven i DHCP 192.168.101.10-192.168.101.50/24.

Spuštěné služby na jednotlivých serverech

- Nagios Server – apache2, php7, nagios, mysql, ssh, snmpd, samba, rsync, rsyslog, bind9;
- Graylog Server – apache2, ntp, rsync, rsyslog, samba, ssh, xinetd, bind9;
- OTRS – apache2, mysql, rsync, rsyslog, samba, ssh, xinetd, bind9;
- DNS server – apache2, rsync, rsyslog, ssh, bind9.



Obrázek 19: Topologie sítě

Zdroj: vlastní kompletace

6.2 Popis požadavků monitoringu

1) Záznam a výstup HW monitoringu:

Monitorované hodnoty budou zaznamenávány do centrální databáze a výstup bude prezentován na webovém rozhraní systému Nagios. V případě poruchy bude zjištěná informace odeslána na zadaný email, resp. pomocí „sms“ na mobilní telefon. Lze nastavit pravidla pro zasílání např. podle času poruchy.

2) Rozsah monitorování na serverech

Na serveru budou monitorovány tyto parametry

Hardware (pokud to daná kombinace HW a OS umožní):

- Chyba napájení/zdroje;
- Porucha ventilátoru;
- Porucha pevného disku;

- Teplota čidla na skříni serveru;
- Využití úložného prostoru na disku;
- Zatížení procesoru;
- Počet Cron procesů;
- Využití paměti;
- Počet přihlášených uživatelů;
- Využití swap space;
- Celkový počet procesů;
- Stav zálohovaného počítače.

Operační systém

- Souborový systém – volné místo;
- Dostupnost serveru na síti.

Aplikační služby/aplikace

- Kontrola stavu mariadb databáze.

Monitoring síťových služeb

- Dostupnost služby – ssh, sftp, http.

Frekvence monitorování

- Hardware 1x za 5 minut;
- Operační systém 1x za 10 minut;
- Síťové služby 1x za 5 minut;
- Síťové prvky 1x 5 min.

Report chyb

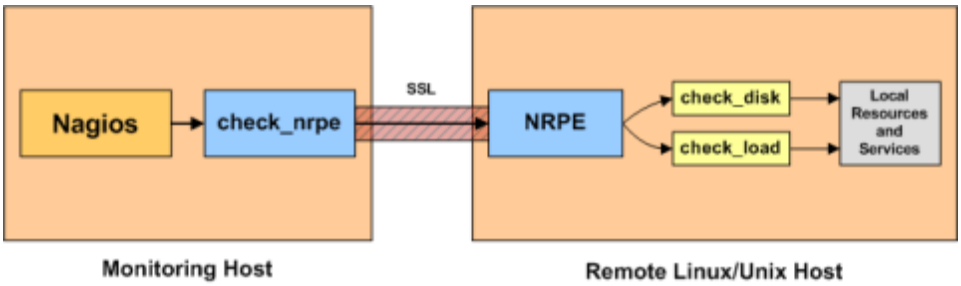
- Chyby budou zaznamenávány zasíláním emailů do tiketového open-source systému OTRS pro přehlednější výpis;
- Pro výpis logovacích chyb ze serverů budou využity open-source systém graylog;

Rozsah monitorování síťových prvků s využití SNMP informací

- CPU zatížení;
- Počet procesů;
- Síťové rozhraní IO, statusy jednotlivých portů;
- Provoz na síti;
- Využití paměti.

6.3 Návrh řešení

Monitoring funguje na hierarchii klient/server, což vyžadovalo na sledovaných serverech nainstalovat NRPE (Nagios Remote Plugin Executor) agenta. Doplněk NRPE je navržen tak, aby umožňoval spouštění pluginů Nagios na vzdálených hostech. Komunikace Nagios a NRPE agenta je navazována vždy při volání pluginu „check_nrpe“, který dá pokyn vzdálenému stroji, který lokálně spustí příslušné pluginy. Výsledky pak vrací zpět procesu Nagios, jenž je vyhodnotí. Nagios střídavě prověřuje dostupnost vzdálených hostů a jejich síťové služby (ping, ssh). Interval, timeouty a rozklad zatěžování systému je možné konfigurovat. Odchytávání SNMP ze síťových prvků se provádí pomocí pluginu check_snmp, který přijímá tzv. snmp-traps. Odchytávat lze všechny informace, které daný síťový prvek dokáže odesílat.

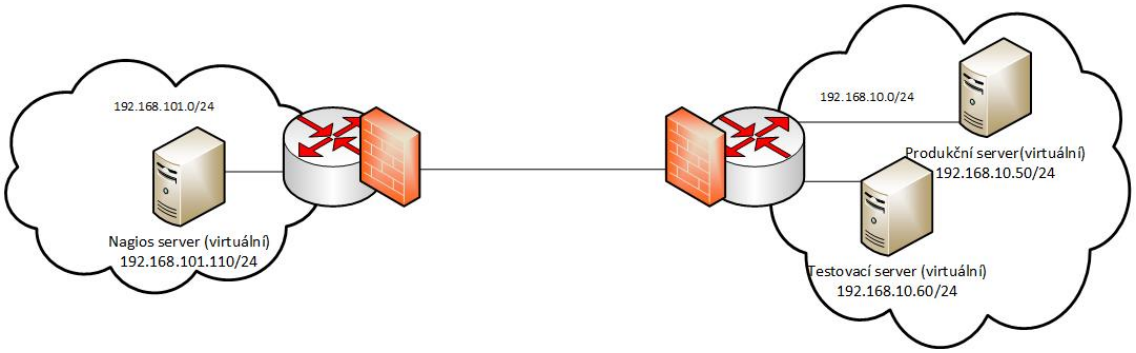


Obrázek 20: NRPE

Zdroj: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/images/nrpe.png

6.3.1 Architektura řešení

Přístup k monitorovaným datům na vzdálený server bude uskutečněn z privátní sítě přes veřejnou síť. Vzhledem k bezpečnostním opatřením a případným hrozbám skrz odchytáváni paketů je do budoucna podstatné opatření sledovat server pomocí VPN spojení mezi routery. Vzdálený server běží v O2 cloudu s využitím NGFW (řeší – produkt – O2 Next generation firewall). O2 cloud je postaný na VMware virtualizaci. Monitorované servery budou produkční a testovací, viz obr. Architektura monitoringu. Na obou serverech běží projekt https://www.prymum.cz . Servery jsou postaveny na open-source řešení Ubuntu-server 14.04 LTS. Na sledovaných serverech běží služby typu apache2, php7, rsync, rsyslog, snmpd, ssh, xinetd a databáze mongodb k uchovávání informací. Celé prostředí běží v jedné LAN síti 192.168.10.0/24. Adresa je z bezpečnostních důvodů změněna. Podrobnějšími logovacími informacemi o serveru si budeme sbírat data do Graylogu pomocí syslog-ng, kde bude probíhat analýza logovacích dat pro zlepšení funkcí na serveru.



Obrázek 21: Architektura monitoringu

Zdroj: vlastní kompletace

6.4 Implementace řešení

Implementace řešení probíhala na virtuálním stroji postaveném na linuxovém řešení od Ubuntu 16.04 LTS. Virtuální prostředí bylo řešeno open-source softwarem operačním systémem zvaným XenServer. Instalace probíhala na fyzickém stroji Fujitsu. Nastavení sítě bylo řešeno nastavením statické privátní IP adresy v souboru:/etc/network/interfaces

```
auto eth0
iface eth0 inet static
address 192.168.101.110
netmask 255.255.255.0
gateway 192.168.101.1
broadcast 192.168.101.255
dns-search dns.aliacte.local
dns-nameservers 192.168.101.130
```

Pro správné načtení a uložení IP adresy staticky je potřeba restartovat service networking.

6.4.1 Instalace balíčků

K instalaci Nagios je důležité na serveru doinstalovat následující balíčky, jinak nebude fungovat správně pracovat. Všechny balíčky jsou potřebné, ale některé balíčky slouží jen pro práci na instalaci, jiné poskytují software pro samotné fungování aplikací.

```
apt-get install wget build-essential
apache2 apache2-utils libgd2-xpm-dev php7.0 php7.0-gd
libapache2-mod-php7.0 openssl perl make wget
libgd2-xpm-dev libperl-dev libssl-dev daemon unzip
openssl xinetd sendmail libnagios-object-perl
```

```
librrds-perl rrdtool iptables-persistant
libgd-graph-perl perl*
```

6.4.2 Instalace Nagios serveru

Instalace Nagios probíhala pomocí instalačních balíčků ze stránek https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.2.0.tar.gz. Byl vytvořen uživatel nagios i skupina nagcmd, s nimiž se provádí instalace a přidělují se práva ke komunikaci s Nagios.

```
useradd nagios
groupadd nagcmd
usermod -a -G nagcmd nagios
usermod -a -G nagcmd www-data
```

Poslední řádek příkazu usermod přidával do skupiny nagcmd účet www-data, pod nímž běží webový server apache.

Další kroky instalace probíhali následovným způsobem.

```
./configure --with-command-group=nagcmd --with-httpd-co
nf=/etc/apache2/conf-enabled
make all
make install
make install-commandmode
make install-config
make install-exfoliation
make install-webconf
a2enmod rewrite
a2enmod cgi
```

A2enmod je skript pro webový server apache, který povoluje zadanou konfigujraci. Po dokončení instalace si ověříme správnou konfiguraci souboru nagios.cfg pomocí:

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

Restartujeme-li webový server apache a nevyskytuje-li se žádný problém při ověřování pokračujeme dále v instalaci Nagios pluginů, které si stáhneme na stránce http://nagios-plugins.org/download/nagios-plugins-2.1.2.tar.gz . K instalaci a konfiguraci pluginů využijeme již vytvořeného uživatele nagios.

```
./configure --with-nagios-user=nagios --with-openssl
make
make install
```

Po instalaci je nutné přidat na pluginy vlastnictví uživatele nagios i pomocí rekurzivního způsobu.

```
chown nagios.nagios /usr/local/nagios
chown –R nagios.nagios /usr/local/nagios/libexec
chown nagios.nagios /usr/local/nagios/plugins
chown –R nagios.nagios /usr/lib/nagios/plugins
```

Tím se docílí toho, že Nagios bude správně komunikovat s nainstalovanými pluginy pod právy uživatele nagios a vykonávat svou práci. Při přidávání nových pluginů je nutné opětovně přidat vlastnictví pro nově přidané pluginy. Pro sledování vzdálených serverů je potřebné nainstalovat na serveru NRPE pro komunikaci se vzdálenými servery. Instalace bude probíhat pomocí instalačního balíčku, který si stáhneme na stránce https://github.com/NagiosEnterprises/nrpe/archive/3.0.tar.gz . Instalaci provedeme pomocí následujících příkazů.

```
./configure --enable-command-args --with-nagios-user=na
gios --with-ssl=/usr/bin/openssl --enable-ssl --with-ss
l-lib=/usr/lib/x86_64-linux-gnu
make all
make install
make install-plugin
make install-inetd
make install-init
make install-xinetd
make install-daemon-config
```

Pro správné využívání funkci NRPE je potřeba nainstalovat daemona xinetd, který se bude dále konfigurovat pro správnou vzdálenou komunikaci. Daemon xinetd naslouchá příchozím požadavkům v síti a spouští službu NRPE.

Monitoring v rámci firmy nevyžadoval těžkou konfiguraci firewallových pravidel pro ochranu, bylo nutné na každém serveru povolit ve firewall pravidlu port 5666 pro NRPE agenta, aby se mohla navázat komunikace na odesílání dat pomocí pluginu. S důrazem na bezpečnost je dobré do budoucnost využít zdrojové a cílové adresy.

```
iptables -A INPUT -p tcp --dport 5666 -j ACCEPT
```

V případě restartu serveru Nagios je potřeba uložit firewall pravidlo.

```
iptables-save > /etc/iptables/rules.v4
```

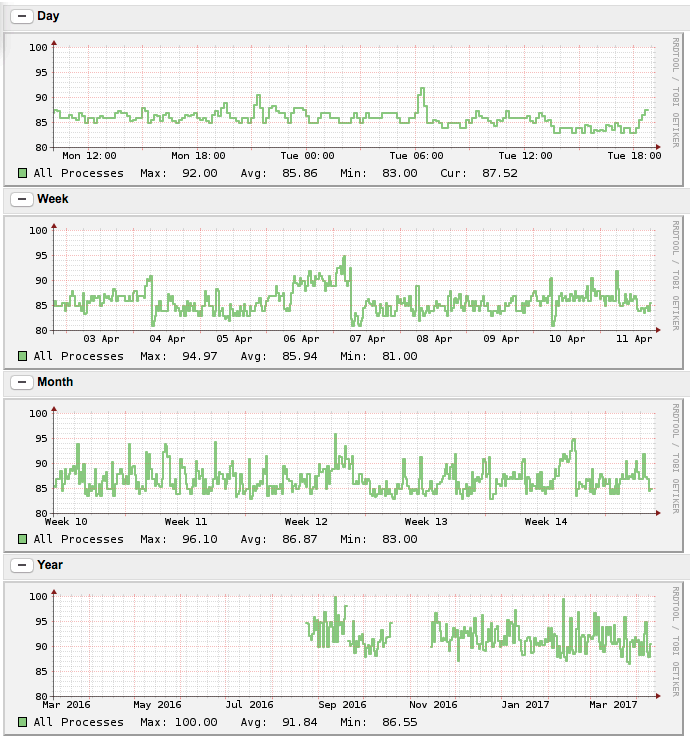
Systém Nagios nepodporuje funkci vykreslování grafů v základní instalace. Služba musela být doinstalována. Instalace probíhala pomocí balíčku, který byl dostupný na stránkách

http://downloads.sourceforge.net/project/nagiosgraph/nagiosgraph/1.5.2/nagiosgraph-1.5.2.tar.gz . Důležité je si zkontrolovat nainstalované balíčky pomocí skriptu ve složce nagiosgraph, kterou jsme si také stáhli.

```
./install.pl --check-prereq
```

V případě, že chybí balíčky, je nutné si je nainstalovat. K nainstalování grafů využijeme automatický skript.

```
./install.pl --layout overlay --prefix
/usr/local/nagios
```



Obrázek 22: Graf Nagios CPU produkčního serveru

Zdroj: vlastní kompletace

6.4.3 Úprava výchozí konfigurace

Pro vzdálenou komunikaci skrze plugin check_nrpe, bylo nutné přidat definici do souboru command.cfg, jinak by pluginy pomocí check_nrpe nefungovaly:

```
define command{
command_name check_nrpe
command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c
$ARG1$}
```

Důležitá konfigurace je daemona xinetd a jeho souboru nrpe, pomocí kterého komunikuje plugin check_nrpe. Z důvodu bezpečnosti je adresa X.X.X.X adresa serveru Nagios.

```
service nrpe{
disable      = yes
socket_type       = stream
port              = @npre_port@
wait              = no
user              = nagios
group             = nagios
server            = /usr/local/nagios/bin/nrpe
server_args  = -c
/usr/local/nagios/etc/nrpe.cfg --inetd
only_from         = 127.0.0.1 X.X.X.X}
```

Správná funkčnost check nrpe se ověřuje způsobem */usr/local/nagios/libexec/check_nrpe -H localhost,* pokud se ukáže verze nainstalovaného NRPE, vše je v pořádku.

K zasílání emailu bylo nutné na serveru nainstalovat Sendmail k zasílání pošty přes protokol SMTP a definovat si potřebné proměnné v souboru resource.cfg. Sendmail je specializovaný program, který zajišťuje jednoduchý způsob zasílání emailu. Program je open source. Využitelná konfigurace proměnných pro Nagios:

```
$USER5$=nagios@gmail.com
$USER6$=nagios@firma.cz
$USER7$=smtp.gmail.com
$USER9$=nagios@gmail.com
$USER10$=password pro nagios@gmail.com
```

Z důvodu bezpečnosti jsou uváděné hodnoty zaměněny. Pro správné zasílání emailu, pokud došlo k problému a výpadku na serveru, bylo nutné definovat v souboru command.cfg jednotlivé typy oznámení hosta a služby:

```
define command{
command_name notify-host-by-email
command_line /usr/bin/printf "%b" "***** Nagios
*****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost:
$HOSTNAME$\nState: $HOSTSTATE$\nAddress:
$HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time:
$LONGDATETIME$\n" |
/usr/local/bin/sendEmail -s $USER7$ -xu $USER9$ -xp
$USER10$ -t $USER6$ -f $USER5$ -l
/var/log/sendEmail -u "** $NOTIFICATIONTYPE$ Host
Alert: $HOSTNAME$ is $HOSTSTATE$ **" -m "***** Nagios
*****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost:
```

```
$HOSTNAME$\nState: $HOSTSTATE$\nAddress:
$HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time:
$LONGDATETIME$\n"
}
define command{
command_name notify-service-by-email
command_line /usr/bin/printf "%b" "***** Nagios
*****\n\nNotification Type:
$NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost:
$HOSTALIAS$\nAddress: $HOSTADDRESS$\nState:
$SERVICESTATE$\n\nDate/Time:
$LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$"
| /usr/local/bin/sendEmail -s $USER7$ -xu $USER9$ -xp
$USER10$ -t $USER6$ -f $USER5$ -l
/var/log/sendEmail -u "** $NOTIFICATIONTYPE$ Service
Alert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$
**" -m "***** Nagios *****\n\nNotification Type:
$NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost:
$HOSTALIAS$\nAddress: $HOSTADDRESS$\nState:
$SERVICESTATE$\n\nDate/Time:
$LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$"}
```

(Anismaj, 2015)

Pro vykreslování grafu na webovém prostředí bylo nutné definovat v command.cfg a templates.cfg podstatné konfigurační věci:

```
define command {
command_name process-service-perfdata-for-nagiosgraph
command_line /usr/local/nagios/libexec/insert.pl
}
define service {
name graphed-service
action_url
/nagios/cgi-bin/show.cgi?host=$HOSTNAME$&service=$SERVI
CEDESC$' onMouseOver='showGraphPopup(this)'
onMouseOut='hideGraphPopup()'
rel='/nagios/cgi-bin/showgraph.cgi?host=$HOSTNAME$&serv
ice=$SERVICEDESC$&period=week&rrdopts=-w+450+-j
register 0}
```

(Sharma, 2013)

6.4.4 Instalace NRPE a pluginů na monitorovaných serverech

Navázání vzájemné komunikace formou NRPE a funkce pluginů pro Nagios musí nainstalováno na monitorovacích serverech. Instalace je provedena stejným způsobem jako u Nagios. Významnou roli zde bude hrát daemon xinetd a jeho správná konfigurace IP adres. Špatná konfigurace daemona může způsobit

problém v navázání komunikace. Z bezpečnostních důvodu nebude IP adresa bude označena jako X.X.X.X.

```
service nrpe
{
disable      = yes
socket_type       = stream
port              = @npre_port@
wait              = no
user              = nagios
group             = nagios
server       =/usr/local/nagios/bin/nrpe
server_args  =-c/usr/local/nagios/etc/nrpe.cfg --inetd
only_from         = 127.0.0.1 X.X.X.X}
```

Kontrolu správné konfigurace je možné provést tímhle způsobem.

```
/usr/local/nagios/libexec/check_nrpe -H X.X.X.X
```

Úspěšné navázání komunikace poznáme výpisem verze nrpe.

6.4.5 Nastavení monitorovaných služeb Nagios

Nainstalovaná služba Nagios plugins už disponuje základními pluginy k okamžitému využití. Více pluginů lze zdarma volně stáhnout na stránkách https://exchange.nagios.org/directory/Plugins. Konfigurace pro monitorování serverů a jejich služeb se implementuje v souboru nrpe.cfg. V části allowed_host definujeme všechny IP adresy vzdálených serverů. Command argumenty na konci souboru v nrpe.cfg slouží k definování jednotlivých pluginů a jejich cestou uložení na serveru. Při vkládání nových pluginů na server je důležité přidat cestu a název s možnými parametry pluginu:

```
command[check_users]=/usr/local/nagios/libexec/check_us
ers -w 5 -c 10
command[check_load]=/usr/local/nagios/libexec/check_loa
d -w 15,10,5 -c 30,25,20
```

Názvy pluginů v hranatých závorkách se využívají ke konfigurování jednotlivých služeb pro monitoring hostů. Sledování vzdálených serverů se provádí vytvořením konfiguračního souboru nazevserveru.cfg, který už je však nutné správně nakonfigurovat. Např.:

```
define host{
use                                  linux-server
host_name                            server
address                              x.x.x.x
max_check_attempts                        5
```

```
check_period                    24x7
notification_interval           30
notification_period             24x7
}
define service {
use
   generic-service,graphed-service
host_name                   server
service_description         PING
check_command
   check_ping!100.0,20%!500.0,60%
}
```

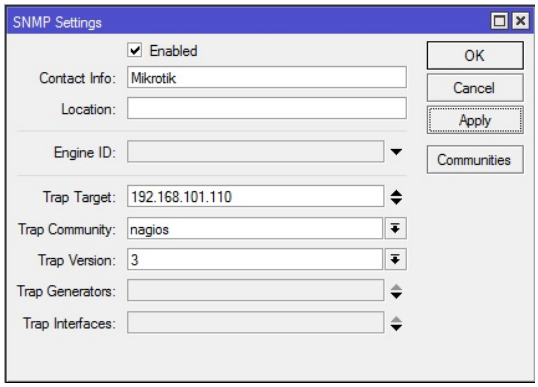
Nagios podporuje vytvoření vlastních period pro sledování, tedy na každý den odlišné časové intervaly, v souboru timeperiods.cfg. Tato schopnost je velmi užitečná, aby byl Nagios v pohotovosti, tak jak každý administrátor potřebuje uzná za vhodné. Konfigurace pro sledování v rozdílných intervalech je v defaultním stavu, takže se sleduje „24 hodin, 7 dní v týdnu“.

6.4.6 Nastavení monitoringu síťových zařízení pomocí SNMP

Monitoring síťových prvků pomocí protokolu snmp s Nagios nám umožňuje plugin check_snmp. Nastavení pluginu se provede podle návodu https://www.monitoring-plugins.org/doc/man/check_snmp.html a konfigurace se provádí v souboru switch.cfg. Bylo nutné definovat plugin check_snmp:

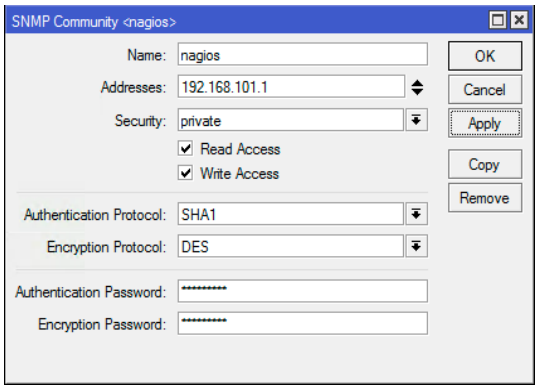
```
define command{
command_name check_snmp
command_line $USER1$/check_snmp -H $HOSTADDRESS$ -c
$ARG1$}
```

Dodatečné pluginy na využívání SNMP můžeme stáhnout ze stránek https://exchange.nagios.org/directory/Plugins/Network-Protocols/SNMP zdarma. V případě, že neexistuje požadovaný plugin, je potřebné si jej do budoucna naprogramovat. Na sledovaných síťových zařízeních od mikrotiku si povolíme službu snmpv2 nebo snmpv3 pro bezpečnější monitoring.



Obrázek 23: SNMP nastavení na Mikrotiku

Zdroj: vlastní kompletace



Obrázek 24: Nastavení SNMP komunity

Zdroj: vlastní kompletace

6.4.7 Nastavení limitů(triggers)

Konfigurování triggerů o notifikaci lze provádět vytvářením šablon pro kontakty, servery, hosty, síťové prvky a pro jejich způsob prováděného upozornění. Nagios disponuje širokým spektrem skrze nakonfigurování svých šablon. Způsoby nastavení lze provádět v souboru templates.cfg. Konfigurace generic-service se využívá u jednotlivých pluginů následovně:

```
define service{
name                    generic-service         ; Jméno šablony
active_checks_enabled                   1       ; Aktivní kontrola povolena
passive_checks_enabled                  1       ; Pasivní kontrola povolena
notifications_enabled                   1       ; Povolení upozornění
event_handler_enabled                   1       ; Služba, která obsluhuje události
                                                        je povolena
process_perf_data                       1       ; Povolení zpracovaní dat o výkonu
check_period                            24x7    ; Sledování každý den
max_check_attempts                      3       ; Opakuje kontrolu pokud je stav jiný než OK
normal_check_interval                   10      ; Kontrola služby každých 10 minut
retry_check_interval                    2       ; Kontrola stavu každé 2 minuty
contact_groups                  admins          ; Upozornění se zašle uživatelům ve skupině
                                                        admin
```

```
notification_options       w,u,c          ; Upozornění ve stavu warning,unknown,
                                                critical
notification_interval      60             ; Opakování o nedostupnosti každou hodinu
notification_period        24x7           ; Upozornění se zašle 24/7
}
```

Konfigurace linuxového serveru pro hosta vypadá následovně:

```
define host{
name                linux-server   ; Název pro hosta
use                 generic-host   ; Pro dědění hodnot z jiné šablony
check_period                24x7           ; Nastavení na denní kontrolu
check_interval              5              ; Aktivní kontrola každých 5 minut
retry_interval              1              ; Naplánování opakování v minutách
max_check_attempts          10             ; Počet kontrol serveru
check_command       check-host-alive       ; Výchozí příkaz pro kontrolu systémů Linux
notification_period    workhours           ; Kontrola probíhá jen v pracovní dny
notification_interval       120            ; Odesílání oznámení každé 2h
notification_options        d,u,r          ; Oznámení pro stavy down, up, recovery
contact_groups              admins         ; Oznámení pro skupinu admin
}
```

Více možností pro nastavení upozornění na jednotlivé triggery je možné se dočíst zde: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/objectdefinitions.html.

6.5 Testování

Testování monitorovacího systému bude probíhat na nově vytvořeném virtuálním serveru, aby nebylo narušeno monitorovací prostředí i infrastruktura z důvodu SLA. U testování bude kladen důraz na rychlost odpovědi, testovány budou jen parametry konektivity dále služba daemon xinetd, NRPE, SSH, HTTP a testování přetížení CPU s pamětí. Testování může mít zkreslené údaje a proto, že probíhá ve firemní síti. Výsledné grafy u testovaného serveru nebudou obsahovat podstatné údaje z důvodu malého sběru dat o serveru.

6.5.1 Testování konektivity

Testování proběhne simulací vypnutí serveru a to proto, že server je virtuální a nelze vytáhnout UPT kabel. Vypnutí serveru bylo provedeno v čase 15:30. Za 22 sekund nám Nagios, viz Výpadek serveru oznámil na webovém prostředí, že server je nedostupný. Ostatní služby byli uváděny ve stavu OK, ale po čase 6 minut a 20 sekund se staly stejně nedostupné. Email o nedostupnosti serveru nám přišel na Tiketovací systém OTRS. Nastavení splňuje očekávané požadavky na upozornění. V případě výpadu konektivity jsme byli upozorněni na logovací systém graylog, viz Logovací upozornění.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
TestovaciServer	CPU Load	OK	04-26-2017 15:45:29	0d 0h 54m 50s	1/3	OK - load average: 0.04, 0.13, 0.07
	Cron procs	OK	04-26-2017 15:46:36	0d 0h 53m 43s	1/3	PROCS OK: 1 process with args 'cron'
	Current Users	OK	04-26-2017 15:47:43	0d 0h 52m 36s	1/3	USERS OK - 0 users currently logged in
	Free Space Disk	OK	04-26-2017 15:48:49	0d 0h 51m 29s	1/3	DISK OK - free space: /var/tmp 11813 MB (78% inode=87%):
	PING	CRITICAL	04-26-2017 15:49:57	0d 0h 0m 22s	1/3	PING CRITICAL - Packet loss = 100%
	SSH	OK	04-26-2017 15:43:03	0d 0h 7m 16s	1/3	SSH OK - OpenSSH_7.2p2 Ubuntu-4ubuntu2.1 (protocol 2.0)
	Swap Usage	OK	04-26-2017 15:42:10	0d 0h 48m 9s	1/3	SWAP OK - 100% free (4217 MB out of 4217 MB)
	Total Processes	OK	04-26-2017 15:47:17	0d 0h 3m 2s	1/3	PROCS OK: 136 processes
	Zombie Processes	OK	04-26-2017 15:47:27	0d 0h 52m 52s	1/3	PROCS OK: 0 processes with STATE = Z

Obrázek 25: Výpadek serveru

Zdroj: vlastní kompletace

Timestamp	source	application_name	connection_id	connection_requests	facility	http_referer	http_user_agent	response_bytes
2017-04-26 13:50:37.113	nagiosAli	nagios			user-level			
HOST ALERT: TestovaciServer;DOWN;SOFT;1;PING CRITICAL - Packet loss = 100%								

Obrázek 26: Logovací upozornění

Zdroj: vlastní kompletace

6.5.2 Spolehlivost dostupnosti daemonů, NRPE

Cílem testování spolehlivosti daemona xinetd a služby NRPE je ověření komunikace Nagios a vzdáleného serveru NRPE. Pokud nebude daemon a NRPE spuštěn, není možné sledovat vzdálený server. Testování probíhalo vypnutím daemona i NRPE příkazem service nrpe stop && service xinetd stop. Přibližně za dvě minuty a třicet sekund nám odpověděl první plugin Swap Usage, že je adresa s portem nedostupná. Po čase pěti minut a třiceti sekund už zareagovalo více pluginů viz Nagios – testování NRPE, xinetd.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
TestovaciServer	CPU Load	CRITICAL	04-26-2017 16:41:29	0d 0h 0m 25s	1/3	(No output on stdout) stderr: connect to address 10.202.60.249 port 5666: Connection refused
	Current Users	CRITICAL	04-26-2017 16:41:43	0d 0h 0m 11s	1/3	(No output on stdout) stderr: connect to address 10.202.60.249 port 5666: Connection refused
	Free Space Disk	OK	04-26-2017 16:32:49	0d 0h 39m 5s	1/3	DISK OK - free space: /var/tmp 11812 MB (78% inode=87%):
	HTTP	OK	04-26-2017 16:38:48	0d 0h 13m 6s	1/3	HTTP OK: HTTP/1.1 200 OK - 11595 bytes in 0.002 second response time
	PING	OK	04-26-2017 16:33:57	0d 0h 37m 57s	1/3	PING OK - Packet loss = 0%, RTA = 0.33 ms
	SSH	OK	04-26-2017 16:37:03	0d 0h 34m 51s	1/3	SSH OK - OpenSSH_7.2p2 Ubuntu-4ubuntu2.1 (protocol 2.0)
	Swap Usage	CRITICAL	04-26-2017 16:40:10	0d 0h 5m 44s	3/3	(No output on stdout) stderr: connect to address 10.202.60.249 port 5666: Connection refused
	Total Processes	CRITICAL	04-26-2017 16:41:17	0d 0h 0m 37s	1/3	(No output on stdout) stderr: connect to address 10.202.60.249 port 5666: Connection refused
	Zombie Processes	CRITICAL	04-26-2017 16:41:27	0d 0h 0m 27s	1/3	(No output on stdout) stderr: connect to address 10.202.60.249 port 5666: Connection refused

Obrázek 27: Nagios – testování NRPE, xinetd

Zdroj: vlastní kompletace

6.5.3 Testování služeb

Cílem testování služby budou SSH a HTTP. Služba HTTP je velmi podstatná z důvodu jejího největšího využívání na produkčním serveru. Uživatelé a firmy, které využívají projekt https://www.prymum.cz, k němu přistupují pouze přes webové rozhraní. Proto je velmi podstatnou službou. Služby se vypnou příkazem service apache2 stop && service ssh stop. Služba SSH byla na webovém prostředí oznámena za dvě minuty a dvacet sekund a služba HTTP byla oznámena za tři minuty a padesát sekund, viz obr. Výpadek služeb.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
TestovaciServer	CPU Load	OK	04-26-2017 17:25:29	0d 0h 33m 26s	1/3	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	04-26-2017 17:25:43	0d 0h 33m 12s	1/3	USERS OK - 1 users currently logged in
	Free Space Disk	OK	04-26-2017 17:26:49	0d 0h 32m 6s	1/3	DISK OK - free space: /var/tmp 11812 MB (78% inode=87%):
	HTTP	CRITICAL	04-26-2017 17:28:48	0d 0h 0m 7s	1/3	connect to address 10.202.60.249 and port 80: Spojení odmítnuto
	PING	OK	04-26-2017 17:23:57	0d 1h 24m 58s	1/3	PING OK - Packet loss = 0%, RTA = 0.27 ms
	SSH	CRITICAL	04-26-2017 17:27:03	0d 0h 1m 52s	1/3	connect to address 10.202.60.249 and port 22: Spojení odmítnuto
	Swap Usage	OK	04-26-2017 17:20:10	0d 0h 28m 45s	1/3	SWAP OK - 100% free (4217 MB out of 4217 MB)
	Total Processes	OK	04-26-2017 17:25:17	0d 0h 33m 38s	1/3	PROCS OK: 138 processes
	Zombie Processes	OK	04-26-2017 17:25:27	0d 0h 33m 28s	1/3	PROCS OK: 0 processes with STATE = Z

Obrázek 28: Nagios – Výpadek služeb

Zdroj: vlastní kompletace

6.5.4 Performance testování

Performance testování znamená zatížení CPU, paměti a disku. Pro testování byl využit nástroj sysbench a následně byli naprogramovány jednoduché skripty pro opakované zatížení serveru. Na zatížení serveru disku byla vygenerována data pomocí příkazu dd if=/dev/zero of=filename bs=1024 count=2GB. Výsledky testování proběhly, dle požadavků administrátorů sítě.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
TestovaciServer	CPU Load	WARNING	04-28-2017 17:25:26	0d 0h 15m 42s	3/3	WARNING - load average: 10.05, 7.67, 12.14
	Current Users	OK	04-28-2017 17:29:42	0d 22h 51m 26s	1/3	USERS OK - 5 users currently logged in
	Free Space Disk	CRITICAL	04-28-2017 17:26:46	0d 0h 26m 22s	3/3	DISK CRITICAL - free space: /var/tmp 62 MB (0% inode=86%):
	HTTP	OK	04-28-2017 17:30:40	0d 0h 0m 28s	1/3	HTTP OK: HTTP/1.1 200 OK - 11595 bytes in 1,741 second response time
	Memory Usage	CRITICAL	04-28-2017 17:29:54	0d 0h 3m 14s	2/3	[MEMORY] Total: 3950 MB - Used: 3902 MB - 98% [SWAP] Total: 4217 MB - Used: 1568 MB - 37%
	PING	OK	04-28-2017 17:27:44	0d 22h 53m 16s	1/3	PING OK - Packet loss = 0%, RTA = 0.30 ms
	SSH	OK	04-28-2017 17:26:56	0d 0h 4m 12s	1/3	SSH OK - OpenSSH_7.2p2 Ubuntu-4ubuntu2.1 (protocol 2.0)
	Swap Usage	OK	04-28-2017 17:24:03	0d 22h 46m 58s	1/3	SWAP OK - 71% free (2977 MB out of 4217 MB)
	Total Processes	CRITICAL	04-28-2017 17:25:12	0d 0h 5m 56s	3/3	CHECK_NRPE STATE CRITICAL: Socket timeout after 10 seconds.
	Zombie Processes	OK	04-28-2017 17:21:23	0d 0h 39m 45s	1/3	PROCS OK: 0 processes with STATE = Z

Obrázek 29: Nagios – performance zatížení

Zdroj: vlastní kompletace

Skripty pro zatěžování serveru

```
#!/bin/bash
for each in 1 2 4 8 16; do
 sysbench --test=cpu --cpu-max-prime=8000000 --num-threads=$each run;
done
```

Obrázek 30: Zatížení CPU

Zdroj: vlastní kompletace

```
#!/bin/bash
for each in 1 2 4 8 16 32 64 128; do
sysbench --test=memory --num-threads=20 --memory-block-size=4G --memory-scope=gl
obal --memory-total-size=4G --memory-oper=read run; done
```

Obrázek 31: Zatížení pamětí

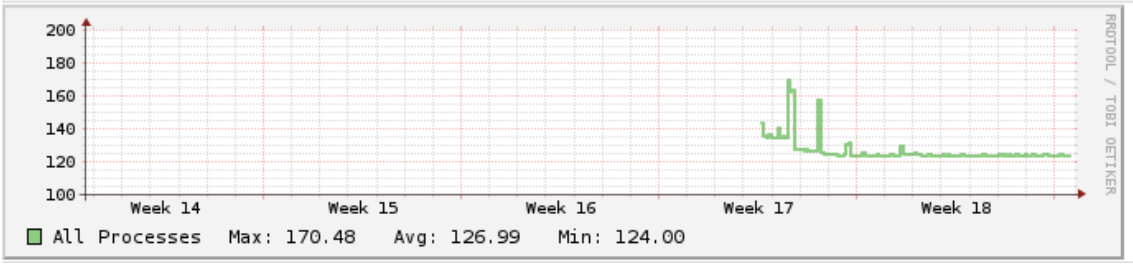
Zdroj: vlastní kompletace

```
#!/bin/bash
for each in 1 4 8 16 32 64; do sysbench --test=fileio --file-total-size=20G --fi
le-test-mode=rndwr --max-time=480 --max-requests=10 --file-block-size=16K --file
-num=128 --num-threads=$each run; sleep 10; done;
```

Obrázek 32: Zatížení disku

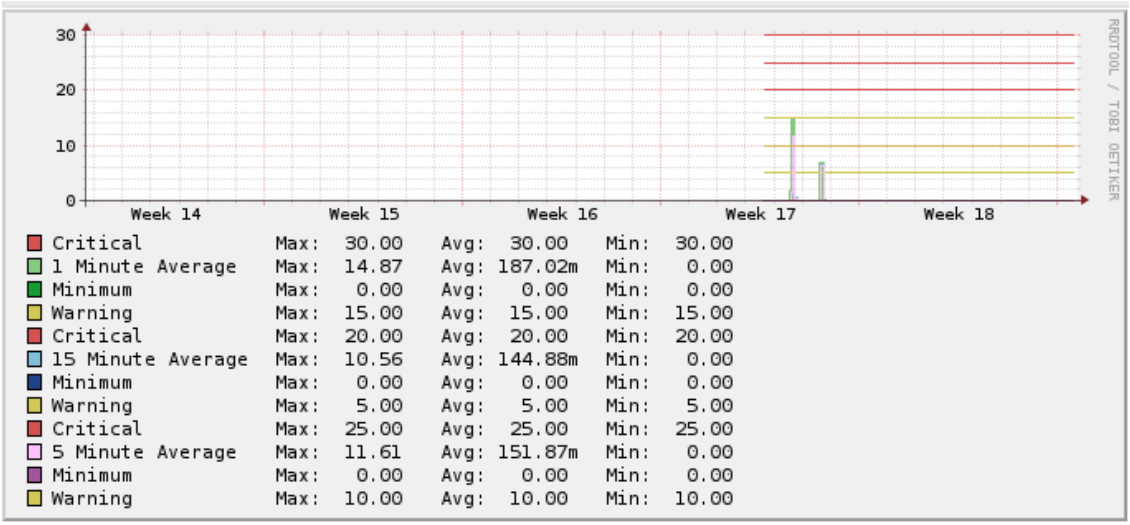
Zdroj: vlastní kompletace

Výsledné grafy znázorňují na testovaném serveru u některých jednotlivých služeb:



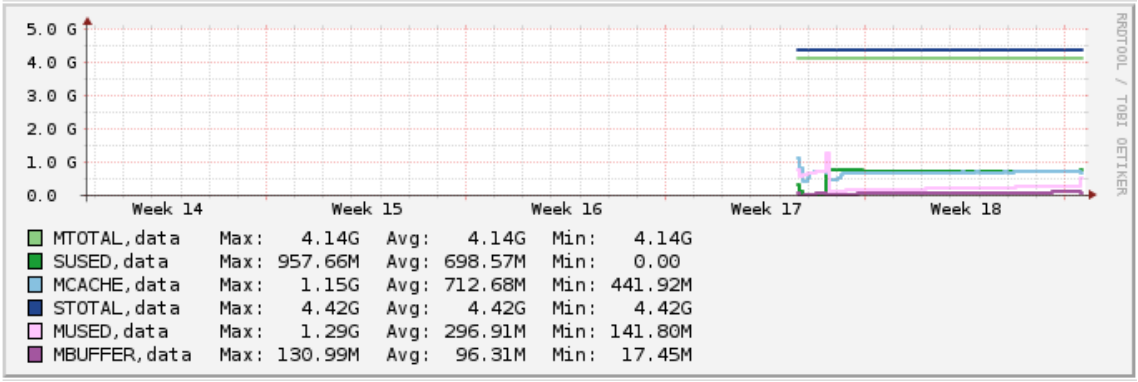
Obrázek 33: Celkový počet běžících procesů

Zdroj: vlastní kompletace



Obrázek 34: Zatížení CPU

Zdroj: vlastní kompletace



Obrázek 35: Zatížení paměti

Zdroj: vlastní kompletace

6.5.5 Souhrn výsledků testování

Všechny testy proběhly s ohledem na požadované výsledky administrátorů sítě. Do budoucna však bude nutné otestovat funkčnost databáze mariadb na monitorovacích serverech. Testování služeb SFTP nemůže být provedeno proto, že daná služba není na produkčních serverech v současné době využívána. Na serverech totiž probíhají firemní úpravy a nemůže být narušen běh serverů. Sledování služby SFTP bylo navrhnuto kvůli importu pluginů do budoucnosti. Stejný problém nastává při testování databáze mariadb na jejímž produkčním i testovacím serveru je testování přísně zakázáno, aby v případě poruchy nebyla narušena infrastruktura. Testování na novém virtuálním serveru mariadb by nemělo žádný význam z důvodu, protože nejsou potřebné údaje v databázi na testování mariadb.

7 Ekonomické zhodnocení navrhovaného řešení

V následující kapitole bude nastíněn finanční náklady na provoz využívaných serverů se zahrnutím problematiky SLA.

7.1 Počáteční investice a náklady

Vzhledem k tomu, že ve výsledné práci se vyskytují jen open-source programy, neměla firma velké počáteční investice. Disponovala už jedním fyzickým serverem, v němž bylo nutné nainstalovat virtuální prostředí a postavit na něm monitoring.

Náklady na nasazení monitorovacího systému se vztahují jen na osobu, která bude daný systém implementovat. Výše finančních nákladu se může lišit v závislosti na zkušenostech a odborných znalostech osoby. Vezmeme-li v úvahu, že systém bude instalovat průměrně zkušený člověk, kterému se zaplatí 200 Kč na hodinu, instalace a konfigurace systému v maximálním možném čase může trvat průměrně 40 hodin práce, takže počáteční investice na práci člověka činí 8000 Kč.

7.2 Provoz a údržba

Cena na provoz a údržbu monitorovacího systému postačí správce systémů se zkušenostmi 0 až 3 roky praxe nebo junior administrátor s méně zkušenostmi. Osoba, která bude sledovat výpadky serverů a služby SLA, může být i student informatiky. Systém nevyžaduje každodenní kontrolu a analýzu stavů. Je-li studentovi zaplaceno za odvedenou práci v časovém rozpětí cca 20 hodin týdně 100 Kč na hodinu, celková investice na provoz a údržbu může firmě přinést průměrné náklady 8000 Kč na měsíc.

7.3 Úspory

Výsledné monitorovací řešení může po nastavení sledovat provoz na přepínači, díky monitorování technických parametrů SLA jako je např. latence, ztrátovost paketů atd. Díky tomu je vhodné sledovat a spočítat i netechnické parametry v případě výpadku sítě od poskytovatele a to např. celková doba trvání opravy konektivity, nedostupnost služeb, apod. Pomocí nakonfigurování Nagios je možné sledovat provoz internetu a u poskytovatele uplatnit slevu v procentech, která je uvedená ve smlouvě.

Další ztráty na nákladech můžou vzniknout při upozornění na životnost disků. V extrémních případech je totiž možné přijít o všechna data na serveru.

7.4 Shrnutí

S neustálým zvyšováním nároků na spolehlivost IT systému stoupají i nároky na kontrolu jejich funkčnosti. Přínos monitorovacího systému umožnil dohled nad prací serverových, síťových služeb a poskytovatelem internetových služeb na sjednané podmínky SLA. Údržba a provoz Nagios nevyžaduje velkou pozornost. Stačí zaměstnat studenta IT, aby získal zkušenosti z praxe a vyzkoušel si pracovat se serverem. V případě výskytu chyby Nagiosu, lze na googlu vyhledat množství rad a vyřešených bugů na nejrůznějších zájmových diskuzních fórech a webech od unix.stackexchange.com až po stackoverflow, ubuntu forum atd. Zavedení monitorovacího systému přineslo firmě Aliacte s.r.o. sledování sítě a systémů, zlepšilo stabilitu celé informační struktury, některé problémy jsou dokonce identifikovány s předstihem.

8 Závěr

Bakalářská práce se zabývá navrhováním a implementováním monitorovacího systému, který tvoří základ každé firmy, jež se zabývá problematikou ICT. Nasazení monitorovacího systému umožňuje předcházet kolizím na síti, výpadkům serverů a jejich síťových a aplikačních služeb, které jsou pro chod firmy nezbytně nutné. To by mělo do budoucna ušetřit náklady na provoz a zajistit přehled o dostupnosti služeb. Postup instalace a konfigurace najde uplatnění v každé firmě či organizaci, kde je nezbytné mít pod kontrolou své servery a software na nich.

Dosažení cíle práce vyžadovalo výběr a porovnání monitorovacích systému a jejich následnou konfiguraci. Výsledná instalace se skládala s open-source řešení. Požadovaný výstup monitoringu se skládal z analýzy síťové infrastruktury, serverů, které firma využívá. Bylo nutné určit služby běžící na serveru a podle toho sestavit výsledný výstup monitoringu. Definování probíhalo diskuzí s administrátory sítě a vedením ICT firmy. Instalace probíhala na virtuálním prostředí, kde bylo nutné uvést monitorovací systém do chodu s ohledem na požadavky firmy.

V implementaci Nagios proběhlo vše bez problémů, až na drobné nedostatky s pluginy, které je nutné do budoucna odstranit. Bylo dosaženo sledování předem určených parametrů skrze pluginy, kde nebyl problém s nastavením a konfigurací. Problém nastal v nedostatku volně stažitelných pluginů, proto je do budoucna dobré napsat si vlastní pluginy pro vlastní potřeby a nedostatky. Problém nastal v monitoringu vzdálené sítě bez VPN tunelu, tudíž řešení je možné v současné době identifikovat pomocí paketů.

Nagios byl rozšířen o vykreslování grafů jednotlivých služeb a zasílání emailu pří výpadku monitorované služby. Testování probíhalo simulací možných problému, které se mohou v praxi uskutečnit. Výsledné upozornění Nagios uskutečnil v relativně krátkém časovém rozpětí. Výsledky testu byly překvapující a byli odeslány emailem na tiketovací systém OTRS.

Všechny testované požadavky byly splněny s výjimkou testování databáze mariadb, která běží na produkčním serveru, kam nebyl umožněn přístup. Monitorovací systém je nasazen a je plně funkční, avšak jak již bylo zmíněno do budoucna je nutné vyřešit problém s VPN a konfiguraci vlastních skriptů.

9 Literatura

BARTH, WOLFGANG.NAGIOS: SYSTEM AND NETWORK MONITORING. 2ND ED. MUNICH:OPEN SOURCE PRESS, 2008, 719 P. ISBN 1593271794.

DOSTÁLEK, LIBOR.VELKÝ PRŮVODCE PROTOKOLY TCP/IP: BEZPEČNOST. 2. AKTUALIZ.VYD. PRAHA: COMPUTER PRESS, 2003, XVI, 571 S. ISBN 80-7226-849-X.

KOCJAN, W. Learning Nagios 4. Birmingham: Packt Publishing, 2014. 400 s.ISBN 978-1-78328-864-9.

NAUGLE, G. MATTHEW. 2006. ILLUSTRATED TCP/IP. WILEY COMPUTER PUBLISHING. 846 S.ISBN 047-119-65-68.

BING, BENNY A PASCAL LORENZ. NETWORKS: THE PROCEEDINGS OF THE JOINT INTERNATIONAL CONFERENCE ON WIRELESS LANS AND HOME NETWORKS (ICWLHN 2002) AND NETWORKING (ICN 2002): ATLANTA, USA, 26-29 AUGUST 2002. RIVER EDGE, N.J.: WORLD SCIENTIFIC, C2002. ISBN 9812381279.

SOSINSKY, BARRIE A. MISTROVSTVÍ – POČÍTAČOVÉ SÍTĚ: [VŠE, CO POTŘEBUJETE VĚDĚT O SPRÁVĚ SÍTÍ]. VYD. 1. BRNO: COMPUTER PRESS, 2010. ISBN 978-80-251-3363-7.

MAURO, DOUGLAS R A KEVIN J SCHMIDT. ESSENTIAL SNMP. 2ND ED. SEBASTOPOL, CA: O'REILLY, 2005. ISBN 0596008406.

MEINEL, CHRISTOPH A HARALD SACK. INTERNETWORKING: TECHNOLOGICAL FOUNDATIONS AND APPLICATIONS. SPRINGER SCIENCE & BUSINESS MEDIA, 2013. ISBN 9783642353918.

BARRETT, DANIEL J. SSH: KOMPLETNÍ PRŮVODCE. 1. VYD. BRNO: COMPUTER PRESS, 2003. ISBN 807226852X.

KUNDU, Dinangkur a S.M. IBRAHIM LAVLU. *Cacti 0.8 network monitoring: monitor your network with ease!*. Birmingham, UK: Packt Publishing, 2009. ISBN 9781847195968.

BADGER, Michael. *Zenoss Core 3.x Network and System Monitoring: a step-by-step guide to configuring, using, and adapting this free open source network monitoring system*. Birmingham: Packt Pub, 2011. ISBN 9781849511582.

ANDREA DALLE VACCHE a STEFANO KEWAN LEE. *Zabbix Network Monitoring Essentials*. Birmingham: Packt Pub, 2015. ISBN 9781784399764

ANDREA DALLE VACCHE a STEFANO KEWAN LEE. *Mastering Zabbix monitor your large IT environment efficiently with Zabbix*. Birmingham: Packt Pub, 2013. ISBN 9781783283491.

OPSVIEW. *Opsview Enterprise Architecture Whitepaper* [online]. 2012 [cit. 2017-05-16]. Dostupné z: https://www.opsview.com/sites/default/files/EnterpriseArchitecture.pdf

AMMON, Tom. *System Monitoring With Nagios* [online]. 2007 [cit. 2017-05-16]. Dostupné z: http://www.macos.utah.edu/documentation/administration/nagios/mainColumnParagraphs/010/document/2007.08.03-univ_of_utah-nagios.pdf

PR NEWSWIRE. LATEST OPSVIEW RELEASE BRINGS BUSINESS SERVICE MONITORING TO GLOBAL USER BASE. PR NEWSWIRE US[ONLINE]. 2014 [CIT. 2017-02-26].

Zenoss Core Administration [online]. 2014, 203 [cit. 2017-05-09]. Dostupné z: https://www.zenoss.com/sites/default/files/Zenoss_Core_Administration_02-022014-4.2-v08.pdf

GUCER, Vasfi, Naeem ALTAF, Erik D ANDERSON, et al. *IBM Tivoli Monitoring: Implementation and Performance Optimization for Large Scale Environments* [online]. 2008. [cit. 2017-05-09]. Dostupné z: http://www.redbooks.ibm.com/abstracts/sg247443.html?Open

IPV4. HTTPS://TOOLS.IETF.ORG [ONLINE]. 1981 [CIT. 2017-03-19]. DOSTUPNÉ Z: HTTPS://TOOLS.IETF.ORG/HTML/RFC791

IPV6. HTTPS://TOOLS.IETF.ORG [ONLINE]. 1998 [CIT. 2017-03-19]. DOSTUPNÉ Z: HTTPS://TOOLS.IETF.ORG/HTML/RFC2460

SOCKET. HTTPS://TOOLS.IETF.ORG [ONLINE]. 1971 [CIT. 2017-03-19]. DOSTUPNÉ Z: HTTPS://TOOLS.IETF.ORG/HTML/RFC147

HTTP. HTTPS://TOOLS.IETF.ORG [ONLINE]. 1999 [CIT. 2017-03-19]. DOSTUPNÉ Z: HTTP://TOOLS.IETF.ORG/HTML/RFC2616

ICMP. HTTPS://TOOLS.IETF.ORG [ONLINE]. 1981 [CIT. 2017-03-19]. DOSTUPNÉ Z: HTTPS://TOOLS.IETF.ORG/HTML/RFC792

BOUŠKA, PETR. ZAČÍNÁME S MONITORINGEM SÍTĚ. IN: SAMURAJ-CZ [ONLINE]. 2009[CIT. 2016-01-12]. DOSTUPNÉ Z: http://www.samuraj-cz.com/clanek/zaciname-smonitoringem-site.

Why we made Opsview Atom. *Opsviewe* [online]. 2015 [cit. 2017-05-09]. Dostupné z: https://www.opsview.com/resources/blog/why-we-made-opsview-atom

ANISMAJ. *How To Send Alerts From Nagios Core Using Gmail And Yahoo* [online]. 2015 [cit. 2017-05-10]. Dostupné z: https://www.unixmen.com/send-alerts-nagios-core-using-gmail-yahoo/

LESKIW, Aaron. *Understanding Syslog: Servers, Messages & Security* [online]. 2016 [cit. 2017-05-08]. Dostupné z: http://www.networkmanagementsoftware.com/what-is-syslog/

JACKGCKOROS. *Opsview Gaining an Edge with Opsview Open Source Monitoring* [online]. 2014 [cit. 2017-05-09]. Dostupné z: https://dogfoodrecipescihuw.wordpress.com/2014/01/17/opsview/

SHARMA, SACHIN. *NAGIOSGRAPH – GRAPHS IN NAGIOS ON CENTOS/RHEL 6.3* [online]. 2013 [cit. 2017-05-10]. Dostupné z: https://sachinsharm.wordpress.com/2013/08/07/nagiosgraph-graphs-in-nagios-on-centosrhel-6-3/

Docu-overview [online]. 2017 [cit. 2017-05-09]. Dostupné z: https://wiki.opennms.org/wiki/Docu-overview

VOZDECKÝ, LUKÁŠ. SROVNÁNÍ SYSTÉMŮ PRO SLEDOVÁNÍ PROVOZU POČÍTAČOVÝCH SÍTÍ [ONLINE]. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. FAKULTA INFORMAČNÍCH TECHNOLOGIÍ, 2007 [CIT. 2017-05-08]. DOSTUPNÉ Z: HTTP://HDL.HANDLE.NET/11012/53022. BAKALÁŘSKÁ PRÁCE. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. FAKULTA INFORMAČNÍCH TECHNOLOGIÍ. ÚSTAV INFORMAČNÍCH SYSTÉMŮ. VEDOUCÍ PRÁCE RUDOLF ČEJKA.

VAŘACHA, FRANTIŠEK. MONITOROVACÍ SYSTÉM FIREMNÍ POČÍTAČOVÉ SÍTĚ NA BÁZI OPEN SOURCE ŘEŠENÍ [ONLINE]. MENDELOVA UNIVERZITA V BRNĚ, PROVOZNĚ EKONOMICKÁ FAKULTA, 2014 [CIT. 2017-05-08]. DOSTUPNÉ Z: HTTPS://IS.MENDELU.CZ/ZP/INDEX.PL?PODROBNOSTI_ZP=47023;ZPET=;PREHLED=VYHLEDAVANI;VZOREK_ZP=MONITORING%20VARACHA;DOHLEDAT=DOHLEDAT;KDE=NAZEV;KDE=AUTOR;KDE=KLIC_SLOVA;STAV_FILTR=BEZ;TYP=1;TYP=2;TYP=3;TYP=6;TYP=8;TYP=7;FAKULTA=14;FAKULTA=23;FAKULTA=220;FAKULTA=38;FAKULTA=2;FAKULTA=79;FAKULTA=60;OBDOBI=2017;OBDOBI=2016;OBDOBI=2015;OBDOBI=2014;JAZYK=1;JAZYK=3;JAZYK=2;JAZYK=29;JAZYK=182;JAZYK=22;JAZYK=23;JAZYK=4;JAZYK=-1;LANG=CZ. MENDELOVA UNIVERZITA, PROVOZNĚ EKONOMICKÁ FAKULTA. VEDOUCÍ PRÁCE ING. MARTIN POKORNÝ, PH.D.

MATŮŠŮ, JINDŘICH. MONITOROVÁNÍ STAVU ROZSÁHLÝCH SÍTÍ [ONLINE]. UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ. FAKULTA APLIKOVANÉ INFORMATIKY, 2008 [CIT. 2017-05-08]. DOSTUPNÉ Z: HTTP://THESES.CZ/ID/KUWOBT?INFO=1;ISSHLRET=ROZSÁHLÉ%3B;ZPET=%2FVYHLEDAVANI%2F%3FSEARCH%3DMONITOROVÁN%C3%AD%20STAVU%20ROZSÁHLÝCH%20S%C3%ADT%C3%AD%26START%3D1. UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ. FAKULTA APLIKOVANÉ INFORMATIKY. VEDOUCÍ PRÁCE ING. TOMÁŠ DULÍK.

EISNER, DANIEL. ANALÝZA MODERNÍCH TECHNOLOGIÍ PRO DOHLED A SPRÁVU FIREMNÍ INFRASTRUKTURY [ONLINE]. ČESKÉ VYSOKE UČENÍ TECHNICKÉ, INFORMATIKA, 2015 [CIT. 2017-05-08]. DOSTUPNÉ Z: HTTPS://DSPACE.CVUT.CZ/HANDLE/10467/62755. ČESKÉ VYSOKE UČENÍ TECHNICKÉ, INFORMATIKA. VEDOUCÍ PRÁCE NÁPLAVA PAVEL.

Přílohy

10 Seznam obrázků a tabulek

Seznam obrázků

Seznam tabulek

A Ukázka výsledné práce