PROJEKT

Multitenantné operačné centrum kybernetickej bezpečnosti riešené
ako otvorená cloudová služba s prvkami strojového učenia

Previazané s balíkom KPB1 – Návrh virt. riešenia

Previazané s výstupom
V3 – Popis vybraného riešenia spĺňajúceho stanovené kritéria

typ – dokumentácia

Analýza I.

Analýza nástrojov na monitorovanie siete pre potreby NOC centra

OBSAH

1 Úvod 4
1.1 Metodika 4
1.2 Prínos projektu 4
2 Teoretická analýza 5
2.1 Network operation center (NOC) / SOC 5
2.2 Operatívne procesy v NOC 5
2.2.1 Význam a prínos NOC 6
2.3 Monitoring 6
2.3.1 Typy monitoringu 6
2.3.2 Spôsoby získavania údajov 7
2.3.3 Automatizácia a integrácia 8
2.4 Zabbix 8
2.4.1 Kľúčové vlastnosti Zabbixu 8
2.4.2 Architektúra Zabbixu 9
2.4.3 Špecifické funkcie Zabbixu 9
2.4.4 Licencovanie a komunita 9
2.5 LibreNMS 9
2.5.1 Kľúčové vlastnosti LibreNMS 9
2.5.2 Architektúra LibreNMS 10
2.5.3 Rozdiely oproti Zabbix 10
2.5.4 Licencovanie a komunita 10
2.6 Prometheus 10
2.6.1 Kľúčové vlastnosti Prometheusu 10
2.6.2 Architektúra Prometheus 11
2.6.3 Špecifické funkcie Prometheusu 11
2.6.4 Integrácia s inými systémami 11
2.6.5 Licencovanie a komunita 12
3 Základná inštalácia a konfigurácia nástrojov 13
3.1 Zabbix 13
3.1.1 Nastavenie notifikácií 14
3.2 LibreNMS 15
3.3 Prometheus 16
4 Porovnávacia analýza 20
4.1.1 Bodovanie 21
5 Záver 24
6 Zdroje 25

1 ÚVOD

Moderné IT infraštruktúry predstavujú komplexné siete zariadení, aplikácií a služieb, ktoré musia fungovať s maximálnou spoľahlivosťou a efektivitou. Aby sa zabezpečila ich kontinuita a rýchle riešenie problémov, je nevyhnutné zaviesť systémy na monitorovanie ich činnosti. Monitorovanie umožňuje organizáciám nielen detegovať potenciálne problémy v reálnom čase, ale tiež analyzovať historické dáta na zlepšenie výkonu a prevádzky.

V tomto projekte sa zameriavame na vytvorenie a implementáciu Security Operations Center (SOC) alebo Network Operations Center (NOC). Tento projekt reaguje na potrebu monitorovania a zabezpečenia IT infraštruktúry katedry, s perspektívou rozšírenia na celú fakultu.

Cieľom je vybudovať plne funkčné operačné centrum, ktoré bude monitorovať kľúčové komponenty siete a poskytovať pravidelné reporty o stave infraštruktúry. Centrum bude navyše schopné detegovať bezpečnostné incidenty a poskytovať odporúčania na ich riešenie.

1.1 Metodika

Realizácia projektu zahŕňa nasledovné kroky:

- Analýza požiadaviek: Na začiatku projektu boli definované požiadavky pre NOC, vrátane špecifikácií monitorovaných komponentov, bezpečnostných funkcií a škálovateľnosti riešenia.
- Spísanie kritérií: Na porovnanie monitorovacích nástrojov (Zabbix, LibreNMS, Prometheus) boli vytvorené kritériá s pridelenými váhami.
- Testovanie nástrojov: Každý nástroj bol najskôr inštalovaný lokálne pomocou Dockeru na testovacie účely. Následne boli tieto nástroje nasadené v prostredí OpenStack na univerzitnom cloude.
- Porovnanie a vyhodnotenie: Nástroje boli porovnávané na základe definovaných kritérií, ktorým boli pridelené váhy, a ich bodové hodnotenia sa použili na vybratie najvhodnejšieho riešenia.

1.2 Prínos projektu

Implementácia NOC na KIS/FRI prinesie zlepšenie v oblasti:

- Monitorovania siete a zariadení,
- Prediktívnej analýzy na detekciu potenciálnych problémov,
- Poskytovania pravidelných reportov o stave siete.

V nasledujúcich kapitolách sa podrobne zameriame na teoretický základ monitorovania, charakteristiky jednotlivých monitorovacích nástrojov a analýzu ich použitia v tomto projekte.

2 TEORETICKÁ ANALÝZA

2.1 Network operation center (NOC) / SOC

Network Operations Center (NOC) je kľúčovým prvkom zabezpečenia efektívnej správy IT infraštruktúry. Ide o centrálny bod monitorovania, ktorý zaisťuje dohľad nad sieťami, servermi, aplikáciami a ďalšími IT systémami [1]. Úlohou NOC je minimalizovať výpadky, zvyšovať dostupnosť a zabezpečovať plynulý chod celej infraštruktúry organizácie [3].

Základnou myšlienkou NOC je konsolidácia údajov zo všetkých sledovaných komponentov do jedného operačného centra. To umožňuje sledovať stav infraštruktúry v reálnom čase, identifikovať potenciálne problémy a reagovať na incidenty ešte predtým, než ovplyvnia užívateľov [1][4].

NOC je navrhnuté tak, aby reagovalo nielen na aktuálne problémy, ale aj analyzovalo dlhodobé trendy, ktoré môžu poukázať na potrebu optimalizácie alebo rozšírenia infraštruktúry [6].

Fungovanie NOC vyžaduje kombináciu funkčných systémov a kvalifikovaného personálu. Tieto dva aspekty sa navzájom dopĺňajú, aby zabezpečili maximálnu efektivitu a spoľahlivosť [3].

1. **Technológie v NOC:** Technologické zázemie NOC zahŕňa:
   - Monitorovacie nástroje: Softvéry ako Zabbix, LibreNMS a Prometheus umožňujú sledovanie výkonnosti systémov, analýzu metrík a generovanie výstrah [5][6].
   - Databázy a analytické systémy: Tieto riešenia zhromažďujú a analyzujú údaje na identifikáciu trendov a predikciu problémov [4].
   - Komunikačné nástroje: Integrované systémy umožňujú rýchlu výmenu informácií medzi členmi tímu a automatizované upozornenia [7].
   - Záložná infraštruktúra: Servery, UPS zariadenia a redundantné sieťové komponenty minimalizujú riziko výpadkov [1].

2. **Tím odborníkov:** NOC je zvyčajne obsluhované tímom, ktorý zahŕňa:
   - Operátorov: Prvotný kontakt pri monitorovaní a základnej analýze problémov [6].
   - Analytikov: Špecialistov na detailnú diagnostiku a návrh riešení [5].
   - Technických inžinierov: Ľudia, ktorí implementujú nápravné opatrenia a optimalizácie [7].

2.2 Operatívne procesy v NOC

Prevádzka NOC zahŕňa široké spektrum činností, ktoré spolupracujú na zabezpečení spoľahlivosti IT systémov :

1. Monitorovanie v reálnom čase: Neustále sledovanie výkonu systémov a siete. Automatizované nástroje generujú upozornenia pri detekcii rôznych odchýlok. [3].
2. Analýza historických dát: Zber a spracovanie dlhodobých údajov na identifikáciu trendov a predikciu možných problémov. Táto analýza pomáha predchádzať výpadkom a zlepšovať výkon infraštruktúry [4][6].
3. Riešenie incidentov: Pri detekcii problému sa spustí proces správy incidentu, ktorý zahŕňa diagnostiku, nápravu a dokumentáciu. Tento proces je podporený ticketing systémami, ktoré sledujú stav riešenia problému [5].
4. Plánovanie kapacít: Na základe analýzy údajov NOC plánuje rozšírenie zdrojov, aby vyhovovalo rastúcim potrebám organizácie [6].
5. Sledovanie SLA: Zabezpečenie dodržiavania dohodnutých úrovní služieb (Service Level Agreements) a vypracovanie pravidelných reportov pre manažment a klientov [7].

2.2.1 Význam a prínos NOC

Nasadenie NOC prináša organizáciám množstvo výhod, medzi ktoré patria:

- Zvýšená dostupnosť: NOC minimalizuje výpadky a zabezpečuje nepretržitú prevádzku IT systémov [5].
- Vyššia bezpečnosť: Vďaka pokročilému monitorovaniu a rýchlej reakcii na incidenty NOC pomáha chrániť infraštruktúru pred kybernetickými hrozbami [4].
- Efektívna správa zdrojov: Automatizácia procesov umožňuje optimalizáciu nákladov a zdrojov [7].
- Podpora rozhodovania: Analýza dát z NOC poskytuje manažmentu dôležité informácie pre strategické plánovanie [3][6].

2.3 Monitoring

Monitoring predstavuje systematické sledovanie a vyhodnocovanie IT infraštruktúry s cieľom zabezpečiť jej optimálny výkon, dostupnosť a bezpečnosť. Je kľúčovým prvkom prevencie problémov a zvyšovania efektivity prevádzky IT systémov [1].

Hlavnou úlohou monitoringu je zhromažďovať a analyzovať údaje o rôznych komponentoch infraštruktúry vrátane sietí, serverov, aplikácií a cloudových prostredí. Tieto informácie pomáhajú identifikovať potenciálne problémy, odhaľovať slabé miesta a prijímať rozhodnutia na základe údajov [3][4].

2.3.1 Typy monitoringu

Monitoring sa môže zameriavať na rôzne aspekty IT prostredia, čo umožňuje flexibilné prispôsobenie podľa potrieb organizácie:

1. **Sieťový monitoring:** Sieťový monitoring zahŕňa sledovanie priepustnosti, latencie, paketových strát a dostupnosti sieťových zariadení, ako sú prepínače, smerovače a firewally. Používa protokoly ako SNMP a nástroje ako Paessler PRTG, ktoré umožňujú podrobné sledovanie prevádzky siete a identifikáciu úzkych miest [2][5].
2. **Serverový monitoring:** Tento typ monitorovania sleduje stav a zaťaženie serverov, vrátane vyťaženia CPU, pamäte, diskového priestoru a teploty hardvéru. Nástroje ako Zabbix a Nagios poskytujú podrobné informácie o výkone serverov a umožňujú predchádzať ich preťaženiu [3][6].
3. **Aplikačný monitoring:** Sledovanie výkonu aplikácií zahŕňa dobu odozvy, chybové hlásenia, dostupnosť a využitie zdrojov aplikácií. Nástroje ako Dynatrace poskytujú detailný pohľad na výkon aplikácií a pomáhajú identifikovať problémy na úrovni kódu, čo zlepšuje užívateľskú skúsenosť [4][7].
4. **Cloudový monitoring:** Cloudový monitoring je zameraný na sledovanie výkonnosti a dostupnosti cloudových služieb, vrátane virtuálnych serverov, kontajnerov a databáz. Nástroje ako Prometheus a Grafana umožňujú podrobné sledovanie cloudových systémov s dôrazom na škálovateľnosť a dostupnosť [5][8].
5. **Bezpečnostný monitoring:** Bezpečnostný monitoring identifikuje a sleduje hrozby, neoprávnené prístupy a podozrivé aktivity. Poskytuje kľúčové informácie pre prevenciu kybernetických útokov a ochranu citlivých údajov [6][9].

2.3.2 Spôsoby získavania údajov

- Protokol SNMP (Simple Network Management Protocol): SNMP je jedným z najpoužívanejších protokolov pre zber údajov o sieťových zariadeniach. Umožňuje sledovanie metrík, ako sú šírka pásma, stav portov a stav zariadení, a poskytuje ich v reálnom čase [1][7].
- API-based monitoring (rozhrania API): Mnohé moderné aplikácie a systémy poskytujú API rozhrania, ktoré umožňujú priamy prístup k metrikám a špecifickým údajom. Tento spôsob je často využívaný v cloudových prostrediach a moderných aplikáciách [8].
- NetFlow/sFlow/jFlow: Tieto protokoly umožňujú sledovanie sieťovej prevádzky. Poskytujú detailné údaje o zdrojoch, cieľoch a objeme prenosu v sieti. Sú ideálne na analýzu sieťových tokov [5][9].
- Syslog: Systémové správy a udalosti môžu byť odosielané do centralizovaného logovacieho servera, kde sú analyzované na účely diagnostiky a bezpečnosti [3].
- WMI (Windows Management Instrumentation): Používa sa na monitorovanie zariadení so systémom Windows. Umožňuje získať informácie o procesoch, výkonových metrikách a konfiguráciách systému [4].
- Logy: Logy poskytujú detailné informácie o aktivitách v systéme, vrátane chýb, výstrah a udalostí. Ich analýza je kľúčová pri diagnostike problémov a zlepšovaní bezpečnosti [4][9].

- Agenti: Softvérové agenti nainštalovaní na sledovaných zariadeniach zhromažďujú údaje o výkonnosti a stave systému. Sú schopní poskytovať detailné metriky, ktoré zlepšujú presnosť monitorovania [3].
- Packet capture (zachytávanie paketov): Nástroje ako Wireshark umožňujú sledovanie detailných informácií o sieťovej prevádzke. Táto metóda je užitočná pri diagnostike a analýze bezpečnostných problémov [6].

2.3.3 Automatizácia a integrácia

Moderné monitorovacie systémy kladú dôraz na automatizáciu a jednoduchú integráciu s inými nástrojmi, čo zvyšuje efektivitu a spoľahlivosť:

- Automatické výstrahy: Systémy automaticky upozornia na prekročenie prahových hodnôt alebo výskyt anomálií, čo umožňuje rýchlu reakciu [5][6].
- Self-healing: Niektoré pokročilé nástroje dokážu automaticky vykonávať nápravné opatrenia, ako je reštartovanie služieb alebo aplikácia opráv.
- Integrácia s inými platformami: Nástroje ako Prometheus sa integrujú s Grafanou na vizualizáciu údajov alebo s ticketovacími systémami na správu incidentov [7][8].

2.4 Zabbix

Zabbix je open-source nástroj navrhnutý pre monitorovanie IT infraštruktúry, zahŕňajúcej siete, servery, virtuálne stroje, cloudové platformy a aplikácie. S jeho schopnosťou zberať, ukladať a analyzovať obrovské objemy dát je Zabbix vhodný pre organizácie, ktoré potrebujú robustné riešenie pre sledovanie výkonnosti a dostupnosti [11].

2.4.1 Kľúčové vlastnosti Zabbixu

Reálne časové monitorovanie: Zabbix umožňuje sledovať milióny metrík v reálnom čase zo serverov, sieťových zariadení a aplikácií. Tento proces je podporený vysokou frekvenciou zberu dát a ich okamžitým spracovaním [12].

Flexibilný notifikačný mechanizmus: Upozornenia môžu byť zasielané prostredníctvom e-mailu, SMS, webhookov alebo integrácie s externými aplikáciami, ako je Slack alebo PagerDuty. Zabbix umožňuje nastaviť podmienky na generovanie upozornení pre takmer každý scenár [13].

Automatické objavovanie: Automatická detekcia sieťových zariadení a služieb minimalizuje čas potrebný na konfiguráciu monitorovania. Tento proces zahrňa detekciu nových zariadení na základe pravidiel, čo zjednodušuje rozšírenie monitorovacieho prostredia [11].

Distribuované monitorovanie: Zabbix podporuje centralizovanú administráciu prostredníctvom webového rozhrania, čo umožňuje monitorovať vzdialené lokality bez potreby fyzickej prítomnosti [14].

Podpora rôznych databáz: Rôznorodosť podpory databáz vrátane MySQL, PostgreSQL, Oracle, IBM DB2 a SQLite poskytuje flexibilitu pri voľbe infraštruktúry a jej integrácie [12][15].

2.4.2 Architektúra Zabbixu

Zabbix je navrhnutý ako modulárna platforma, ktorá obsahuje nasledujúce hlavné komponenty:

1. **Zabbix Server:** Hlavná súčasť zodpovedná za spracovanie dát, vykonávanie kalkulácií, generovanie upozornení a ich distribúcia [12].
2. **Zabbix Agenti:** Klientské programy nainštalované na sledovaných zariadeniach zhromažďujú údaje o výkone a stave zariadenia. Agent poskytuje podporu pre operačné systémy ako Linux, Windows a UNIX [14].
3. **Proxy:** Proxy servery sprostredkúvajú komunikáciu medzi agentmi a hlavným serverom, čo zvyšuje efektivitu pri monitorovaní rozsiahlych a geograficky rozdelených prostredí [13].
4. **Databáza:** Centálne úložisko pre konfiguračné údaje, historické metriky a logy [15].
5. **Webové rozhranie:** Front-end aplikácia pre konfiguráciu, prístup k dátam a vizualizáciu [11].

2.4.3 Špecifické funkcie Zabbixu

- **Podpora viacerých protokolov:** Okrem vlastných agentov podporuje monitorovanie prostredníctvom SNMP, IPMI, JMX, SSH a Telnet, čo zvyšuje univerzálnosť a prispôsobivosť [12][13].
- **Škálovateľnosť:** Schopnosť zvládnuť milióny metrík za sekundu robí Zabbix vhodným pre monitorovanie veľkých a dynamických prostredí [14].
- **Bezpečnosť:** Podporuje šifrovanú komunikáciu medzi komponentmi a pokročilú autentifikáciu, čo minimalizuje riziká narušenia [15].
- **Vizualizácia dát:** Dashboardy, mapy a grafy poskytujú prehľadnú a rýchlu analýzu kľúčových metrík [11].

2.4.4 Licencovanie a komunita

Zabbix je distribuovaný pod licenciou GNU Affero General Public License (AGPL) version 3. Zdrojový kód je dostupný verejne, čo umožňuje komunite prispievať k jeho vývoju. Rozsiahla dokumentácia a aktívne fórum poskytujú vysokú podporu[12][13].

2.5 LibreNMS

LibreNMS je open-source systém pre monitorovanie sietí, navrhnutý na automatickú detekciu a sledovanie sieťových zariadení a služieb. Bol vytvorený ako fork projektu Observium a od roku 2013 sa aktívne rozvíja s dôrazom na komunitný vývoj a flexibilitu [16][17].

2.5.1 Kľúčové vlastnosti LibreNMS

- **Automatické objavovanie zariadení:** LibreNMS podporuje protokoly ako CDP, FDP, LLDP, OSPF, BGP, SNMP a ARP na detekciu zariadení, čo znižuje manuálnu prácu pri nasadení a rozšírení monitorovacieho systému [17].

- **Flexibilný systém upozornení:** Poskytuje prispôsobiteľný mechanizmus upozornení prostredníctvom e-mailu, IRC, Slack a ďalších komunikačných kanálov. To zabezpečuje, že administrátori sú okamžite informovaní o problémoch [18].
- **Rozhranie API:** LibreNMS poskytuje API, ktoré umožňuje automatizáciu, integráciu s inými nástrojmi a prístup k monitorovacím dátam [16].
- **Systém účtovania:** Nástroj ponúka možnosť generovať účty za šírku pásma podľa využitia, čo je užitočné pre poskytovateľov internetových služieb [17].
- **Automatické aktualizácie:** LibreNMS sa pravidelne aktualizuje, čo zaisťuje prístup k novým funkciám a opravám bezpečnostných zraniteľností bez potreby manuálneho zásahu [18].

2.5.2 Architektúra LibreNMS

LibreNMS sa skladá z nasledovných komponentov:

1. **Hlavný server:** Spracováva dáta zozbierané od zariadení a zabezpečuje ich ukladanie do databázy [16].
2. **Webové rozhranie:** Poskytuje prístup k vizualizácii dát, konfigurácii a analýze výkonu [17].
3. **Databáza:** Ukladá konfiguračné údaje a zozbierané metriky. LibreNMS je kompatibilný s databázami MySQL a MariaDB [18].

2.5.3 Rozdiely oproti Zabbix

- LibreNMS má natívnu podporu pre účtovanie šírky pásma, čo Zabbix nemá ako natívnu funkciu [17].
- LibreNMS je viac zameraný na sieťové monitorovanie, zatiaľ čo Zabbix pokrýva širší záber vrátane aplikácií a serverov [16].

2.5.4 Licencovanie a komunita

LibreNMS je dostupný pod licenciou GPLv3 a jeho vývoj je posúvaný komunitou. Komunita však nie je natoľko aktívna ako pre Zabbix.

2.6 Prometheus

Prometheus je open-source nástroj na monitorovanie a alarmovanie, špeciálne navrhnutý pre moderné cloudové a kontajnerované prostredia. Jeho architektúra a funkcie ho odlišujú od tradičných riešení, pričom sa zameriava na flexibilitu, vysoký výkon a ľahkú integráciu s ďalšími systémami [19][20].

2.6.1 Kľúčové vlastnosti Prometheusu

- **Pull-based model:** Prometheus zbiera metriky prostredníctvom HTTP endpointov. Tento model eliminuje potrebu agentov a zvyšuje kontrolu nad dátami. Monitorované služby musia byť schopné poskytovať metriky vo formáte, ktorý Prometheus dokáže spracovať [20].
- **Viacrozmerné dáta:** Metriky sú ukladané vo forme časových radov identifikovaných unikátnymi štítkami (labels), čo umožňuje pokročilú analýzu dát a vysokú úroveň prispôsobenia [21].

- **Dotazovací jazyk PromQL:** Prometheus používa vlastný dotazovací jazyk PromQL, ktorý umožňuje výkonné analýzy, agregácie a výpočty priamo nad časovými radmi. Tento jazyk je základom pre tvorbu alarmov a vizualizácií [22].
- **Autonómnosť:** Prometheus funguje ako samostatná jednotka bez závislosti na externých systémoch, čo zvyšuje jeho spoľahlivosť a odolnosť voči výpadkom [23].
- **Alertmanager:** Súčasťou Promethea je Alertmanager, ktorý spracováva výstrahy a umožňuje ich distribúciu cez e-mail, Slack, PagerDuty a ďalšie platformy. [24]

2.6.2 Architektúra Prometheus

Hlavé komponenty prometheus:

1. **Prometheus Server:** Tento komponent zodpovedá za zber metriky, ukladanie časových radov a spracovanie dotazov PromQL [20].
2. **Exporter**: Exportéry sú špeciálne moduly, ktoré konvertujú metriky z rôznych systémov a aplikácií do formátu čitateľného pre Prometheus. Príklady zahŕňajú Node Exporter (monitorovanie OS) a Blackbox Exporter (testovanie dostupnosti) [21].
3. **Alertmanager**: Riadi všetky generované výstrahy a ponúka možnosti ako zoskupovanie a umlčovanie upozornení [24].
4. **Pushgateway**: Umožňuje krátkodobo žijúcim aplikáciám, ktoré nemajú vlastné endpointy, posielať metriky do Promethea [23].
5. **Webové rozhranie**: Poskytuje základné možnosti dotazovania a vizualizácie dát, hoci väčšina používateľov volí na tento účel Grafanu [19].

2.6.3 Špecifické funkcie Prometheusu

- **Optimalizácia pre kontajnery a mikroservisy**: Prometheus je ideálny pre dynamické prostredia. Jeho integrácia s Kubernetes umožňuje automatické objavovanie služieb a sledovanie ich stavu [22].
- **História a uchovávanie dát**: Dáta sú uchovávané lokálne na disku, čo umožňuje vysokú rýchlosť dotazovania. Pre dlhodobé uchovávanie je možné integrovať nástroje ako Thanos alebo Cortex [23].
- **Flexibilné alertovanie**: Kombinácia PromQL a Alertmanagera umožňuje vytvárať komplexné pravidlá pre generovanie výstrah na základe časových radov [24].

2.6.4 Integrácia s inými systémami

- **Grafana**: Prometheus je plne kompatibilný s Grafanou, ktorá slúži na vizualizáciu dát a tvorbu pokročilých dashboardov. [19].
- **Kubernetes**: Prometheus podporuje natívnu integráciu s Kubernetes, vrátane automatického objavovania podov, služieb a ďalších objektov. Táto funkcia je kľúčová pre monitorovanie moderných cloudových prostredí [22].
- **Thanos**: Thanos rozširuje možnosti Prometheusu o dlhodobé uchovávanie dát, horizontálne škálovanie a dostupnosť dát v distribuovaných systémoch [23].

- **CI/CD nástroje**: Prometheus môže byť integrovaný do CI/CD pipeline na monitorovanie a alertovanie počas nasadzovania aplikácií, čím zabezpečuje vyššiu stabilitu produkčných prostredí [24].

2.6.5 Licencovanie a komunita

Prometheus je distribuovaný pod licenciou Apache 2.0, čo poskytuje veľkú flexibilitu pri jeho nasadení a úpravách. Aktívna komunita a členstvo v CNCF (Cloud Native Computing Foundation) zabezpečujú jeho neustály vývoj [19].

3 ZÁKLADNÁ INŠTALÁCIA A KONFIGURÁCIA NÁSTROJOV

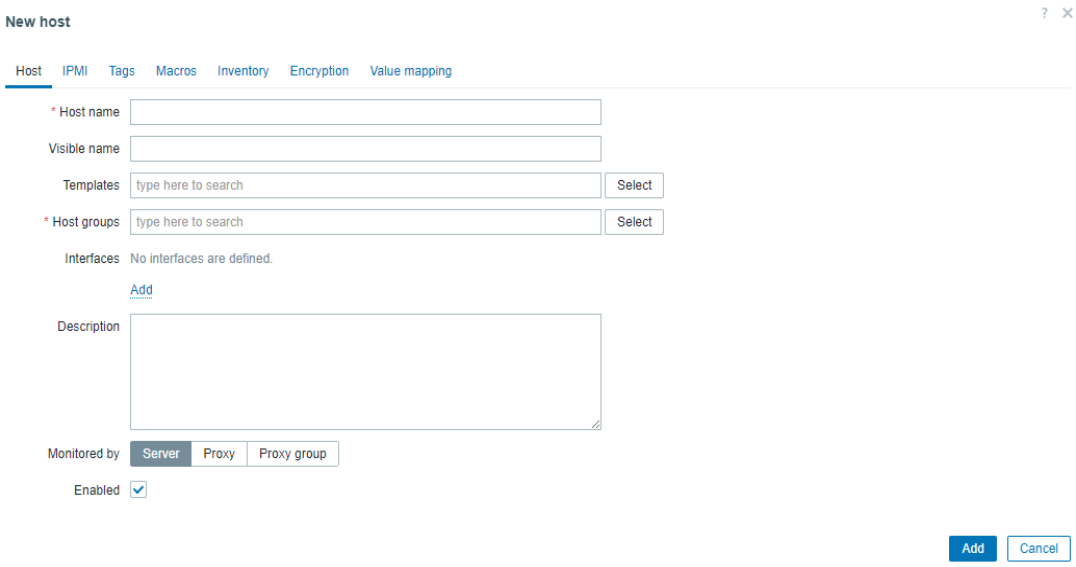
Na inštaláciu jednotlivých nástrojov, bol využitý docker compose, čo umožňuje spúšťať všetky potrebné kontajnere súčasne.

3.1 Zabbix

Zabbix Server kontajner obsahuje hlavnú logiku Zabbixu. Premenné prostredia definujú pripojenie k databáze („DB_SERVER_HOST“) a prihlasovacie údaje („MYSQL_DATABASE“, „MYSQL_USER“, „MYSQL_PASSWORD“). Komponent „depends_on“ zabezpečuje, že Zabbix Server začne až po spustení MySQL Servera.

Zabbix Web Interface kontajner poskytuje webové rozhranie na správu a konfiguráciu Zabbixu. Premenné prostredia zdieľajú rovnaké nastavenia databázy ako Zabbix Server. Port 8080 je zmapovaný na hostiteľský systém, čo umožňuje prístup k rozhraniu cez prehliadač.

MySQL Server je databázový kontajner, ktorý uchováva všetky dáta Zabbixu vrátane konfigurácie a historických metrík. Objem („mysql-data“) zabezpečuje, že dáta pretrvávajú aj po reštarte alebo odstránení kontajnera.



Obrázok 1: Pridanie zariadenia na monitorovanie pomocou Zabbix

```yaml
default_version: '3.8'
services:
  # Zabbix Server
  zabbix-server:
    image: zabbix/zabbix-server-mysql:latest
    environment:
      DB_SERVER_HOST: mysql-server
      MYSQL_DATABASE: zabbix
      MYSQL_USER: zabbix
      MYSQL_PASSWORD: zabbix_pass
      MYSQL_ROOT_PASSWORD: root_pass
    depends_on:
      - mysql-server

  # Zabbix Web Interface
  zabbix-web:
    image: zabbix/zabbix-web-nginx-mysql:latest
    environment:
      DB_SERVER_HOST: mysql-server
      MYSQL_DATABASE: zabbix
      MYSQL_USER: zabbix
      MYSQL_PASSWORD: zabbix_pass
      MYSQL_ROOT_PASSWORD: root_pass
    ports:
      - "8080:8080"
    depends_on:
      - zabbix-server

  # MySQL Server
  mysql-server:
    image: mysql:latest
    environment:
      MYSQL_DATABASE: zabbix
      MYSQL_USER: zabbix
      MYSQL_PASSWORD: zabbix_pass
      MYSQL_ROOT_PASSWORD: root_pass
    volumes:
      - mysql-data:/var/lib/mysql
volumes:
  mysql-data:
```

Obrázok 2: Ukážka zabbix .yml súboru

3.1.1 Nastavenie notifikácií

Je potrebné prejsť do sekcie „Administration -> Media types“ a pridať požadovaný komunikačný kanál, napríklad e-mail. Po pridaní média je potrebné nakonfigurovať konkrétne nastavenia, ako sú prihlasovacie údaje alebo URL webhooku.

Následne je potrebné prejsť do časti „Configuration -> Actions“, kde sa vytvorí nová akcia na odosielanie upozornení. Táto akcia musí obsahovať podmienky, ktoré určujú, kedy sa upozornenie odošle (napríklad pri výskyte konkrétneho problému alebo prekročení určitého prahu metriky).

Akcia ďalej obsahuje kroky, kde sa špecifikuje, ktorému používateľovi alebo skupine používateľov budú upozornenia zasielané a prostredníctvom ktorého komunikačného kanála. Týmto spôsobom je možné presne prispôsobiť systém notifikácií požiadavkám monitorovanej infraštruktúry.

3.2 LibreNMS

```yaml
version: "3.8"
services:
  db:
    image: mariadb:10.5
    container_name: librenms-db
    environment:
      MYSQL_ROOT_PASSWORD: root_password
      MYSQL_DATABASE: librenms
      MYSQL_USER: librenms
      MYSQL_PASSWORD: librenms_password
    volumes:
      - db-data:/var/lib/mysql
    restart: unless-stopped

  librenms:
    image: librenms/librenms:latest
    container_name: librenms-app
    depends_on:
      - db
    environment:
      DB_HOST: db
      DB_NAME: librenms
      DB_USER: librenms
      DB_PASSWORD: librenms_password
      TZ: UTC
      BASE_URL: http://localhost
    ports:
      - "8000:8000"
    volumes:
      - librenms-data:/data
    restart: unless-stopped

  memcached:
    image: memcached:latest
    container_name: librenms-memcached
    restart: unless-stopped

  redis:
    image: redis:latest
    container_name: librenms-redis
    restart: unless-stopped

volumes:
  db-data:
  librenms-data:
```

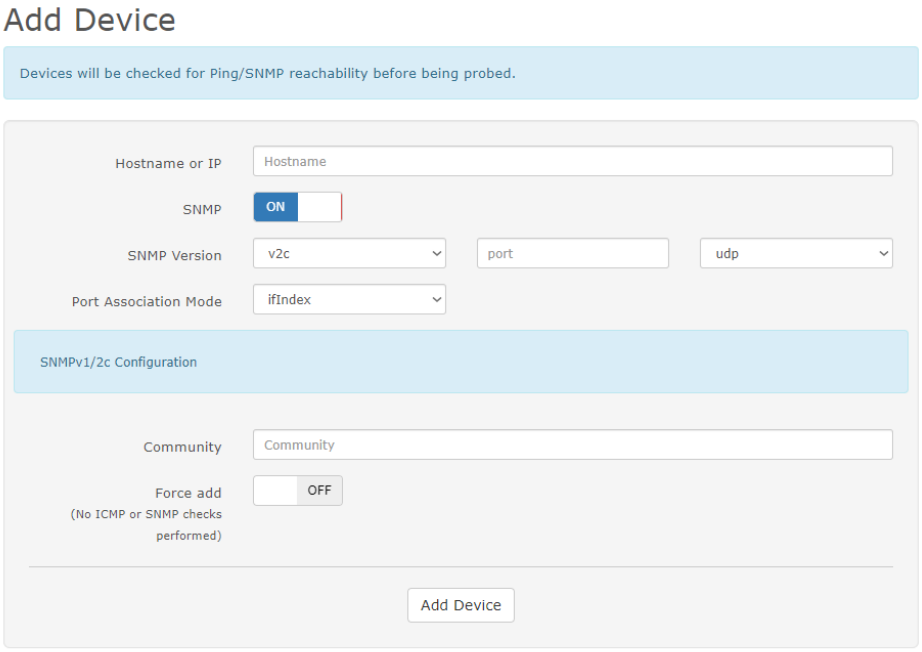
Obrázok 3: Ukážka librenms .yml súboru

Konfigurácia obsahuje databázový kontajner, aplikačný kontajner a podporné služby, ako sú Memcached a Redis.

Databázový kontajner používa obraz mariadb:10.5. Environmentálne premenné MYSQL_ROOT_PASSWORD, MYSQL_DATABASE, MYSQL_USER a MYSQL_PASSWORD nastavujú heslo pre root prístup, názov databázy a prihlasovacie údaje pre LibreNMS. Úložisko databázy je perzistentné a mapované na db-data:/var/lib/mysql.

Aplikačný kontajner využíva obraz librenms/librenms:latest a závisí od databázového kontajnera. Premenné DB_HOST, DB_NAME, DB_USER a DB_PASSWORD definujú pripojenie k databáze. Premenná TZ nastavuje časovú

zónu a BASE_URL definuje adresu pre prístup k aplikácii. Dáta aplikácie sú uložené v librenms-data:/data, pričom port 8000 je zmapovaný na hostiteľský systém.



Obrázok 4: Pridanie zariadenia na monitorovanie pomocou LibreNMS

3.3 Prometheus

Táto sekcia popisuje nastavenie Prometheus a Grafana pomocou Docker Compose. Konfigurácia zahŕňa hlavné komponenty potrebné na monitorovanie, vrátane Prometheus servera, SNMP Exportera pre zber SNMP metrík a Grafana pre vizualizáciu dát.

```yaml
version: "3.8"
services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    ports:
      - "9090:9090"
    volumes:
      - prometheus-config:/etc/prometheus
    command:
      - --config.file=/etc/prometheus/prometheus.yml
    restart: unless-stopped

  snmp-exporter:
    image: prom/snmp-exporter:latest
    container_name: snmp-exporter
    ports:
      - "9116:9116"
    volumes:
      - snmp-exporter-config:/etc/snmp_exporter
    restart: unless-stopped

  grafana:
    image: grafana/grafana:latest
    container_name: grafana
    ports:
      - "3000:3000"
    environment:
      GF_SECURITY_ADMIN_USER: admin
      GF_SECURITY_ADMIN_PASSWORD: admin
    volumes:
      - grafana-data:/var/lib/grafana
    restart: unless-stopped

volumes:
  prometheus-config:
  snmp-exporter-config:
  grafana-data:
```

Obrázok 5: Ukážka konfiguracieprometheus, snmp exporter a grafana

Táto konfigurácia zahŕňa tri hlavné komponenty: Prometheus, SNMP Exporter a Grafana, ktoré spolupracujú na zbere, spracovaní a vizualizácii dát.

Kontajner Prometheus využíva obraz prom/prometheus:latest a je pomenovaný ako prometheus. Tento kontajner pracuje na porte 9090, ktorý je zmapovaný na hostiteľský systém. Úložisko pre konfiguračné súbory je mapované na prometheus-config:/etc/prometheus. Politika reštartovania unless-stopped zaručuje, že kontajner bude automaticky spustený, ak dôjde k jeho zlyhaniu.

SNMP Exporter je ďalší dôležitý komponent tejto konfigurácie. Používa obraz prom/snmp-exporter:lates. Tento kontajner je zodpovedný za zber dát zo zariadení prostredníctvom SNMP protokolu a pracuje na porte 9116.

Kontajner Grafany beží na porte 3000, a poskytuje používateľské rozhranie pre tvorbu a správu dashboardov. Prihlasovacie údaje pre administrátora sú definované premennými GF_SECURITY_ADMIN_USER a GF_SECURITY_ADMIN_PASSWORD. Rovnako ako ostatné kontajnery, aj tento využíva politiku reštartovania unless-stopped.

Zdieľané objemy v tejto konfigurácii zabezpečujú perzistentné uchovávanie dát a konfigurácií. Konfiguračné súbory Promethea sú uložené v prometheus-

config, nastavenia SNMP Exportera v snmp-exporter-config a dáta Grafany v grafana-data.

Prometheus je dostupný na adrese http://<IP_adresa>:9090, SNMP Exporter na porte 9116 a Grafana na http://<IP_adresa>:3000. Na správu monitorovania je potrebné pridať IP adresy alebo hostname zariadení do konfiguračného súboru prometheus.yml, čo umožní SNMP Exporteru zbierať dáta z týchto zariadení. Po správnom nastavení je možné zobraziť zozbierané dáta v Grafane, kde sa dá vytvoriť prehľadný a prispôsobiteľný dashboard pre monitorovanie infraštruktúry.

```yaml
# prometheus.yml
global:
  scrape_interval: 15s  # Interval zberu dát

scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "snmp_router"
    static_configs:
      - targets: ["158.193.154.59"]  # IP adresa SNMP zariadenia
    metrics_path: /snmp
    params:
      module: [if_mib]  # SNMP modul pre rozhrania
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: snmp-exporter:9116  # Endpoint SNMP Exportera

# snmp.yml
modules:
  if_mib:
    walk:
      - 1.3.6.1.2.1.1  # System group (uptime, názov systému)
      - 1.3.6.1.2.1.2  # Interfaces (stav rozhraní, traffic)
    version: 2
    auth:
      community: public
```

Obrázok 6: Ukážka prometheus .yml a snmp.yml

Konfiguračný súbor prometheus.yml obsahuje základné nastavenia na zber dát pre Prometheus. Parameter scrape_interval je nastavený na 15 sekúnd, čo určuje interval medzi jednotlivými cyklami zberu dát. Sekcia scrape_configs obsahuje definíciu cieľov monitorovania. Prvý cieľ, označený ako prometheus, monitoruje samotný Prometheus server na adrese localhost:9090. Druhý cieľ, s názvom snmp_router, je zameraný na monitorovanie zariadenia s IP adresou 158.193.154.59 prostredníctvom SNMP. Zber dát je špecifikovaný pomocou cesty /snmp a využíva parametre modulu if_mib, ktorý je navrhnutý na sledovanie sieťových rozhraní.

V sekcii params je definovaný modul SNMP if_mib, ktorý zahŕňa sledovanie systémových informácií a stavov rozhraní. Sekcia relabel_configs umožňuje prispôsobenie názvov metrík a ich premapovanie na jednotlivé metriky Promethea. Cieľ SNMP Exportera je definovaný na adrese snmp-exporter:9116, čo umožňuje integráciu medzi Prometheom a SNMP Exporterom.

Konfiguračný súbor snmp.yml obsahuje nastavenia pre SNMP Exporter. Modul if_mib obsahuje zoznam OID, ktoré určujú, aké dáta budú zbierané. OID

1.3.6.1.2.1.1 je používaný na monitorovanie systémových informácií, ako je uptime a názov systému, zatiaľ čo OID 1.3.6.1.2.1.2 monitoruje stav sieťových rozhraní a prenesené dáta. SNMP verzia je nastavená na verziu 2 a autentifikácia využíva komunitu public.

4 POROVNÁVACIA ANALÝZA

Na začiatok bolo potrebné vytvoriť zoznam kritérií, slúžiacich na vzájomné porovnávanie skúmaných nástrojov. Každé s kritérií ma pridelenú váhu na škále 1-10, ktorá hovorí o dôležitosti konkrétneho kritéria.

V analýze má každý nástroj pridelený počet bodov v rozmedzí 1-10 pre dané kritérium. Počet bodov je následne vynásobený váhou daného kritéria, čo zabezpečuje, že dôležitejšie kritéria majú väčší vplyv na finálny výsledok. Následne sa body za všetky kritéria spočítajú, čo udáva výsledný počet bodov. Pre lepšiu prehľadnosť výsledkov sú finálne body pre všetky nástroje vynásobené x0,01.

Kritérium	Váha	Zabbix	LibreNMS	Prometheus
Hardvérové požiadavky	8	9	8	7
Výkonnosť	6	8	7	9
Bezpečnostné funkcie	5	6	7	8
Podpora protokolov SNMP verzií	7	10	10	10
Automatizácia a tvorba skriptov	5	8	8	9
Možnosť nastavenia monitorovania	9	8	7	9
Upozornenia	9	8	7	9
História monitorovaných údajov a ukladanie dát	6	7	6	8
Škálovateľnosť	7	8	7	9
Podpora API	6	8	7	9
Integrácia s inými nástrojmi	7	7	6	9
Podpora redundancie a vysoká dostupnosť	5	6	5	7
Vizualizácia a reportovanie	9	8	7	9
Používateľské rozhranie	7	7	8	8
Kompatibilita	8	9	7	8
Podpora a komunita	8	9	7	9

Jednoduchosť konfigurácie	7	8	8	8
Náklady	7	6	7	8
Jednoduchosť inštalácie	4	7	7	9
Upgrade systémov	7	8	9	9
Licencia	10	10	10	10
Dokumentácia	7	8	7	9
Aktuálnosť riešenia	10	8	7	9
AAA voči LDAP/AD	5	5	4	6
Inštalácia do VM/Kubernetes	2	5	4	6
Spolu		15,28	14,16	16,88

4.1.1 Bodovanie

- Hardvérové požiadavky: Zabbix dosiahol vysoké skóre v tejto kategórii, pretože jeho hardvérové požiadavky sú optimalizované pre nasadenie vo väčších IT infraštruktúrach, pričom ponúka robustnosť a flexibilitu aj na menších systémoch. LibreNMS a Prometheus majú vyššie nároky pri škálovaní vo veľkých prostrediach.
- Výkonnosť: Prometheus spracováva veľké objemy dát s nízkou latenciou, čo ho predurčuje na moderné prostredia. Zabbix a LibreNMS sú menej efektívne pri intenzívnom monitorovaní.
- Bezpečnostné funkcie: Prometheus vyniká podporou TLS a flexibilnou správou prístupov. Zabbix a LibreNMS ponúkajú základné bezpečnostné možnosti.
- Podpora protokolov SNMP verzií: Všetky tri nástroje podporujú SNMPv1, SNMPv2c a SNMPv3, pričom umožňujú šifrovanú komunikáciu a autentifikáciu.
- Automatizácia a tvorba skriptov: Prometheus exceluje v tejto oblasti vďaka svojej podpore moderných nástrojov ako Grafana a jednoduchému integrácii s nástrojmi na automatizáciu. Zabbix a LibreNMS poskytujú základné možnosti skriptovania, ale nie sú tak rozšíriteľné.
- Možnosť nastavenia monitorovania: Prometheus a Zabbix umožňujú široké prispôsobenie monitorovania, zatiaľ čo LibreNMS má obmedzenejšie možnosti konfigurácie.

- Upozornenia: Všetky tri nástroje ponúkajú dobré možnosti nastavenia upozornení, ale Prometheus je vďaka flexibilným pravidlám upozornení a integráciám s inými nástrojmi mierne lepší.
- História monitorovaných údajov a ukladanie dát: Prometheus efektívne ukladá časové rady dát. Zabbix a LibreNMS uchovávajú historické údaje, ale s vyššími nárokmi na zdroje.
- Škálovateľnosť: Prometheus je optimalizovaný na jednoduché škálovanie v moderných prostrediach. Zabbix a LibreNMS vyžadujú zložitejšiu konfiguráciu.
- Podpora API: Prometheus ponúka moderné API s jednoduchou integráciou. Zabbix a LibreNMS majú menej pokročilé, ale stále funkčné API.
- Integrácia s inými nástrojmi: Prometheus podporuje širokú škálu integrácií, najmä s nástrojmi ako Grafanou. Zabbix a LibreNMS sú menej flexibilné.
- Podpora redundancie a vysoká dostupnosť: Prometheus ponúka základnú podporu redundancie, ale Zabbix aj LibreNMS zaostávajú kvôli zložitejšej implementácii týchto funkcií.
- Vizualizácia a reportovanie: Prometheus s Grafanou vyníká vo vizualizáciách. Zabbix ponúka kvalitné reporty, LibreNMS mierne zaostáva.
- Používateľské rozhranie: LibreNMS má jednoduché a intuitívne rozhranie, čo mu dáva výhodu oproti Zabbixu a Prometheusu, ktoré môžu byť náročnejšie na používanie.
- Kompatibilita: Zabbix je kompatibilný s Linuxom, Windows a Unixom, čo ho robí ideálnym pre rôzne prostredia. Prometheus je optimalizovaný pre kontajnerizované a cloudové technológie, ako Kubernetes a Docker. LibreNMS sa zameriava primárne na Linux, s obmedzenou podporou iných operačných systémov.
- Podpora a komunita: Zabbix a Prometheus majú silnú komunitu a rozsiahlu dokumentáciu. LibreNMS poskytuje menšiu, ale užitočnú podporu.
- Jednoduchosť konfigurácie: Prometheus umožňuje jednoduché prispôsobenie cez YAML konfiguračné súbory, čo uľahčuje nastavovanie metrik a upozornení. LibreNMS má intuitívne používateľské rozhranie, ktoré uľahčuje konfiguráciu základných funkcií. Zabbix ponúka robustnejšie možnosti konfigurácie, ale vyžaduje viac manuálnych krokov a zložitejšiu správu.
- Náklady: LibreNMS je bezplatný nástroj a má nižšie náklady na nasadenie a prevádzku. Prometheus a Zabbix majú vyššie náklady na údržbu vo väčších prostrediach.
- Jednoduchosť inštalácie: LibreNMS aj Zabbix majú rýchle a jednoduché nasadenie, hlavne cez kontajnery. Prometheus je optimalizovaný pre kontajnerizované prostredia, ale vyžaduje základné znalosti.
- Upgrade systémov: Prometheus a LibreNMS podporujú jednoduché a často automatizované upgrady. Zabbix ponúka spoľahlivé aktualizácie každých 6 mesiacov, ale vyžaduje viac manuálnej práce v komplexných prostrediach.

- Licencia: Všetky tri nástroje majú plnú open-source licenciu.
- Dokumentácia: Prometheus ponúka najaktuálnejšiu a najrozsiahlejšiu dokumentáciu, ktorá uľahčuje jeho používanie. Zabbix a LibreNMS tiež poskytujú kvalitné zdroje, ale menej detailné.
- Aktuálnosť riešenia: Prometheus je najaktuálnejší vďaka modernému vývoju a častej aktualizácii. Zabbix a LibreNMS sú stabilné, ale menej inovatívne.
- AAA voči LDAP/AD: Prometheus ponúka základnú podporu cez integráciu s LDAP na overenie prístupov, ktorá je jednoduchá na konfiguráciu, ale menej rozšírená na pokročilé scenáre. Zabbix umožňuje integráciu s LDAP a Active Directory s podrobnejším nastavením prístupových práv, avšak konfigurácia môže byť zložitejšia. LibreNMS podporuje LDAP/AD pre základné autentifikačné funkcie.
- Inštalácia do VM/Kubernetes: Prometheus je navrhnutý pre jednoduché nasadenie v moderných kontajnerovaných prostrediach, ako sú Kubernetes a Docker. Zabbix podporuje nasadenie do VM a kontajnerov, ale vyžaduje viac manuálnych krokov a konfigurácie. LibreNMS je tiež možné nasadiť do VM a kontajnerov, avšak má obmedzená a menej zdokumentovanú podporu.

ZÁVER

Na základe analýzy a bodového hodnotenia je možné odporučiť kombináciu nástrojov Zabbix a Prometheus. Táto voľba je podložená detailnou analýzou vlastností nástrojov a ich schopností pokryť široké spektrum monitorovacích potrieb.

Zabbix vyniká prostrediach, kde je potrebná detailná kontrola nad tradičnými sieťovými zariadeniami. Podpora protokolov ako SNMP, IPMI či JMX, robustné vizualizácie a upozornenia umožňujú rýchlu detekciu problémov. Vďaka silnej komunite a dokumentácii je ľahko implementovateľný a podpora redundancie a škálovateľnosti zaručuje stabilitu pri rozšírení infraštruktúry.

Prometheus je ideálny pre dynamické a cloudové prostredia. Schopnosť spracovávať veľké množstvo metrík s nízkou latenciou, integrácia s Grafanou pre pokročilé vizualizácie a flexibilné API z neho robia moderný nástroj na rýchle adaptovanie sa na zmeny infraštruktúry. Pre katedru, ktorá plánuje nasadenie cloudových technológií, je Prometheus nevyhnutný.

Zabbix bude optimálny pre monitorovanie tradičných zariadení a aplikácií, zatiaľ čo Prometheus bude využitý na monitorovanie dynamických a cloudových prvkov infraštruktúry. Táto kombinácia poskytne komplexné riešenie monitorovania, ktoré pokryje všetky potreby katedry, od stability a spoľahlivosti až po flexibilitu a škálovateľnosť. Návrh zároveň zabezpečuje, že katedra bude pripravená na budúce rozšírenie infraštruktúry a integráciu ďalších technológii.

ZDROJE

1. "The Comprehensive Guide to SNMP" – LogicMonitor. Dostupné na: https://www.logicmonitor.com/resource/the-comprehensive-guide-to-snmp
2. Dirk Schrader - " Understanding the 4 Types of Network Monitoring Tools and Comparing Available Solutions. Dostupné na: https://blog.netwrix.com/2023/12/27/network-monitoring-tools/
3. "NOC Tools and Software" – INOC. Dostupné na: https://www.inoc.com/blog/noc-tools-and-software
4. "Important Roles in the NOC" – Extnoc. Dostupné na: https://www.extnoc.com/network-operations-center/what-are-the-important-roles-in-the-noc/
5. "Enterprise Network Management Guide" – INOC. Dostupné na: https://www.inoc.com/enterprise-network-management-guide
6. "Network Operations Center Overview" – Extnoc. Dostupné na: https://www.extnoc.com/network-operations-center/
7. "Best Practices for Effective NOC Operations" – BMC. Dostupné na: https://www.bmc.com/blogs/noc-network-operations-center/
8. " What is network monitoring?. – IBM. Dostupné na: https://www.ibm.com/think/topics/network-monitoring
9. Grafana and Prometheus Integration" – Grafana Labs. Dostupné na: https://grafana.com/docs/grafana/latest/getting-started/prometheus/
10. "Zabbix Architecture and Key Features" – Hawatel. Dostupné na: https://hawatel.com/en/blog/introduction-to-zabbix-architecture-and-key-features/
11. "Zabbix Overview" – Initmax. Dostupné na: https://www.initmax.cz/wp-content/uploads/2022/03/zabbix-overview-en.pdf
12. "Zabbix Documentation" – Zabbix.com. Dostupné na: https://www.zabbix.com/documentation
13. "Comprehensive Guide to Zabbix" – Zabbix Solutions. Dostupné na: https://www.zabbix.com/solutions
14. "Zabbix Features and Advantages" – Tech Resources. Dostupné na: https://www.techresources.com/zabbix-features
15. "LibreNMS Overview" – LibreNMS Documentation. Dostupné na: https://docs.librenms.org/
16. "Getting Started with LibreNMS" – LibreNMS Community. Dostupné na: https://community.librenms.org/
17. "LibreNMS Features and Comparison" – Network Monitoring Tools. Dostupné na: https://www.networkmonitoringtools.org/librenms
18. "Introduction to Prometheus" – Prometheus Documentation. Dostupné na: https://prometheus.io/docs/introduction/overview/
19. "PromQL Guide" – Prometheus Documentation. Dostupné na: https://prometheus.io/docs/prometheus/latest/querying/basics/
20. "Monitoring with Prometheus and Grafana" – Grafana Labs. Dostupné na: https://grafana.com/docs/grafana/latest/getting-started/prometheus/
21. "Kubernetes Integration" – Prometheus Documentation. Dostupné na: https://prometheus.io/docs/prometheus/latest/getting_started/
22. "Thanos and Long-term Storage" – Thanos Documentation. Dostupné na: https://thanos.io/
23. "Alertmanager" – Prometheus Documentation. Dostupné na: https://prometheus.io/docs/alerting/latest/alertmanager/