VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

MODULÁRNÍ SYSTÉM PRO EVIDENCI A SPRÁVU DIGITÁLNÍCH DŮKAZŮ

MODULAR SYSTEM FOR RECORDING AND MANAGING DIGITAL EVIDENCES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — David Zeman

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. Lukáš Malina, Ph.D.

SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: David Zeman ***ID:*** 231304

Ročník: 3 ***Akademický rok:*** 2023/24

NÁZEV TÉMATU:

Modulární systém pro evidenci a správu digitálních důkazů

POKYNY PRO VYPRACOVÁNÍ:

V rámci práce nastudujte problematiku zabezpečení cloudových služeb a řešení. Analyzujte možné hrozby, zranitelnosti a rizika využívání cloudových služeb s ohledem na privátní a veřejná řešení. Zaměřte se na možnosti bezpečného nastavení a provozování NextCloudu a na bezpečnou správu logů a událostí. Cílem bakalářské práce je návrh a nastavení bezpečnosti cloudového uložiště pro správu a evidenci digitálních důkazů včetně funkční integrace externích modulů pro autentizaci, archivaci a logování událostí.

DOPORUČENÁ LITERATURA:

[1] Menezes, Alfred, Van Oorschot, Paul C. a VANSTONE, Scott A.. Handbook of applied cryptography. Boca Raton: CRC Press, c1997. Discrete mathematics and its applications. ISBN 0-8493-8523-7.

[2] Web eID: electronic ID smart cards on the Web [online]. [cit. 2022-09-06]. Dostupné z: https://web-eid.eu.

Termín zadání: 5.2.2024 ***Termín odevzdání:*** 31.5.2024

Vedoucí práce: doc. Ing. Lukáš Malina, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Tato bakalářská práce pojednává o problematice bezpečnostních hrozeb u cloudových služeb a jejich obecných řešení. Výstupem této práce je dále návrh řešení správy a uchovávání logů. Tato problematika je řešena pomocí softwaru od vydavatele Grafana Labs, mezi které patří Promtail, Loki a Grafana. V úvodu jsou stanoveny základní pojmy pojednávající o cloudových službách a jejich obecné bezpečnosti. Následující kapitola se věnuje Nextcloudu a jeho bezpečnostním prvkům a externím aplikacím, zajišťující bezpečnost Nextcloudu. Další kapitoly jsou seznámením s Dockerem a dále seznámením s logováním a jeho využitím. V závěru je samotný návrh řešení správy a uchovávání logů, řešený již zmíněným softwarem od společnosti Grafana Labs.

KLÍČOVÁ SLOVA

Bezpečnost cloudu, Cloud computing, Nextcloud, Docker, Zpracování logů, Logmanager

ABSTRACT

This bachelor thesis addresses the issue of security threats in cloud services and their general solutions. Furthermore, the output of this thesis is a proposed solution for log management and storage. This issue is addressed by using software from Grafana Labs publisher which include Promtail, Loki and Grafana. In the introduction, the basic concepts dealing with cloud services and their general security are established. The following section discusses Nextcloud and its security features and the external applications that provide Nextcloud security. The next chapters are an introduction to Docker and also an introduction to logging and its usage. Finally, the actual design of the log management and storage solution is addressed by the aforementioned software from Grafana Labs.

KEYWORDS

Cloud security, Cloud computing, Nextcloud, Docker, Log processing, Logmanager

ZEMAN, David. *Modulární systém pro evidenci a správu digitálních důkazů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2024, 55 s. Bakalářská práce. Vedoucí práce: doc. Ing. Lukáš Malina, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	David Zeman
VUT ID autora:	231304
Typ práce:	Bakalářská práce
Akademický rok:	2023/24
Téma závěrečné práce:	Modulární systém pro evidenci a správu digitálních důkazů

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrální práce panu doc. Ing. Lukáš Malina, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Seznam výpisů

Úvod

V současné době, kdy je internet běžně používán téměř pro vše, se ve velké části internetových služeb vyskytuje nějaká forma cloudu, často kvůli jednodušší správě dat a rychlejšímu přístupu. V tomto prostředí se však zvyšuje riziko výskytu kybernetických hrozeb. Útočníci se snaží získat přístup k datům, které jsou v cloudu uloženy, nebo způsobit výpadek služby. Únik dat nebo výpadek služby může mít pro společnosti závažné důsledky. Může vést k finančním ztrátám, poškození dobrého jména nebo dokonce k právním postihům.

Pro lepší přehled nad pokusy o útoky a prevenci před nimi se používají logy (záznamy z aplikace), pomocí kterých můžeme monitorovat dnou aplikaci a události v ní. To vyvolává potřebu kvalitního úložiště a správce logů, který umožňuje pohodlné vyhledávání a filtraci mezi logy.

Tato bakalářská práce se věnuje návrhu a realizaci bezpečnostních opatření nad vybranou instancí Nextcloudu, který je lokálně hostován.

V této práci je dále rozebrána obecná funkcionalita cloudu a jeho bezpečnost, včetně obecných řešení těchto bezpečnostních rizik. Dále představení cloudového úložiště Nextcloud a jeho bezpečnostních prvků, které nabízí, včetně externě vyvinutými aplikacemi, dostupnými pomocí oficiálního Nextcloud App Store.

Práce je také věnována přiblížení fungování Dockeru, na kterém běží přidělená instance Nextcloud a také v této práci navržené řešení správce logů.

Poslední dvě části práce rozebírají nejdříve obecnou problematiku logů včetně jejich využití a typů, způsob logování v Nextcloudu, představení původního řešení zpracování logů a kritéria pro nový návrh řešení správy a uchovávání logů. Na závěr je toto řešení dle kritérií představeno, včetně konfigurace jednotlivých služeb pro správce logů, nastavení webového prostředí pro uživatele a dotazování se nad logy.

1 Cloud computing

V této kapitole se proberou modely cloudového nasazení, servisní modely a obecná bezpečnost a zabezpečení cloudových služeb.

Cloud computing (cloudový výpočet) je model poskytování IT služeb, aplikací, výpočetního výkonu a uložiště dat na dálku. Uživatelé nemusí (ale můžou, dle zvoleného modelu) vlastnit a spravovat vlastní hardware a software. Mezi výhody cloudových služeb patří:

- Efektivnost – uživatelé platí jen za to co potřebují nebo využijí. To může vést k úsporám nákladů.
- Flexibilita – cloudové služby jsou flexibilní a dají se škálovat podle potřeb.
- Dostupnost – tyto služby jsou dostupné z jakéhokoliv místa s přístupem na internet [1].

1.1 Modely cloudového nasazení

Dělení dle modelů nasazení, neboli architektury cloud computingu, je jedním ze základního kategorizování cloudu. Přihlíží se zde jakým způsobem je poskytována infrastruktura a přístup k ní, ale také z pohledu zákazníka míra a způsob sdílení výpočetní ifrastruktury. Existují 4 základní modely dle nasazení [2].

Veřejný cloud

Veřejný cloud je prvním a nejčastěji používaným modelem cloud computingu. Umožňuje uživatelům přístup k nabízeným službám z jakéhokoli místa a kdykoli prostřednictvím internetu. Díky tomu je možné zdroje snadno škálovat podle potřeby. Velkou výhodou veřejného cloudu je vysoká škálovatelnost a schopnost udržet kvalitu služeb i při vysokém vytížení.

Základním principem veřejného cloudu je sdílení výpočetních zdrojů mezi více uživateli. Uživatelé platí provozovateli podle počtu služeb, které používají. To znamená, že na jednom serveru nebo disku mohou být umístěna data od více zákazníků.

Prostředí veřejného cloudu je tvořeno více datovými centry, která jsou propojena mezi sebou. Data uživatelů jsou od sebe důkladně oddělena a izolována. Úložiště je většinou vlastněno a provozováno obchodními, akademickými nebo vládními organizacemi.

Veřejný cloud nabízí širokou škálu služeb, včetně infrastruktury, platformy a aplikací [3].

Privátní cloud

Privátní cloud je vhodný pro velké společnosti a vládní agentury, které vyžadují vysokou úroveň bezpečnosti a dodržování předpisů. V tomto modelu nejsou výpočetní zdroje sdíleny mezi více zákazníky, ale jsou vyhrazeny pouze pro jednoho [5].

Výhodou privátního cloudu je vysoká bezpečnost. Firma má úplnou kontrolu nad infrastrukturou, což jí umožňuje implementovat vlastní bezpečnostní opatření. Data jsou také umístěna na samostatné síti, která je oddělena od jiných sítí. Další výhodou privátního cloudu je vysoká dostupnost. Servery jsou obvykle umístěny v datových centrech poskytovatele, které jsou navrženy tak, aby byly co nejodolnější vůči výpadkům. Smluvní záruky dostupnosti (SLA) zajišťují, že infrastruktura bude v provozu alespoň na určitou dobu [3, 5].

Nevýhodou privátního cloudu je nižší flexibilita než u veřejného cloudu. Výpočetní zdroje nelze škálovat tak snadno a rychle [3].

Hybridní cloud

Hybridní cloud je kombinace veřejného, privátního cloudu, či jiného druhu cloudu. Díky tomu lze kombinovat výhody více modelů. Například citlivá data lze ukládat v privátním cloudu, kde má společnost plnou kontrolu nad bezpečností. Naopak aplikace nebo méně zranitelná data, které nevyžadují tak vysokou úroveň zabezpečení, lze provozovat ve veřejném cloudu, který je flexibilnější a levnější.

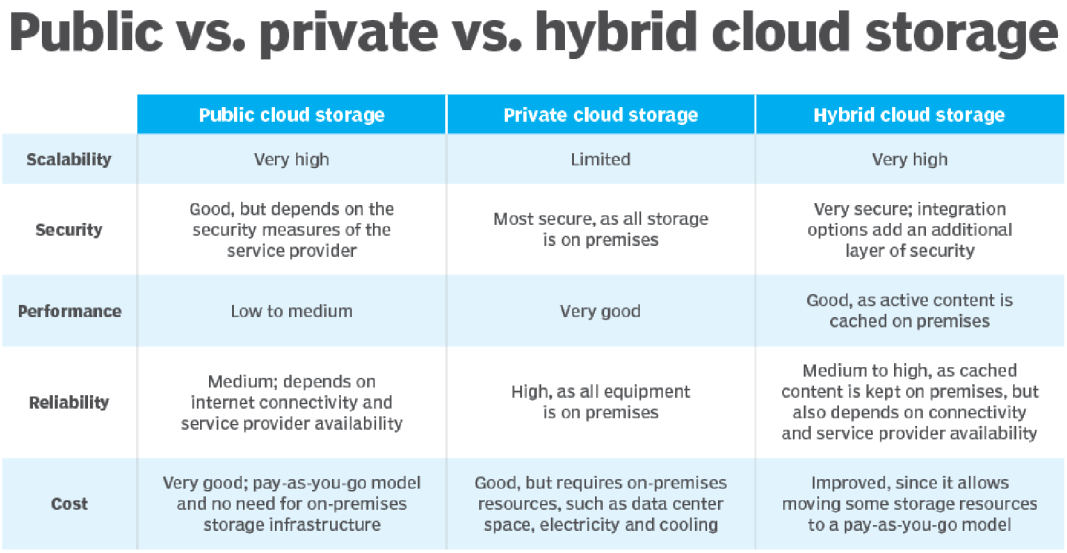
Hybridní cloud také umožňuje škálování výpočetních zdrojů podle potřeby. Pokud se například zvýší poptávka po určité aplikaci, lze výpočetní zdroje pro tuto aplikaci snadno přidat ve veřejném cloudu. To umožňuje společnosti rychle reagovat na změny v poptávce po jejích službách, nebo vyvažovat výkonnové špičky.

Díky tomu, že hybridní cloud kombinuje výhody obou modelů, je vhodným řešením pro mnoho společností. Je vhodný zejména pro společnosti, které potřebují mít kontrolu nad citlivými daty, ale zároveň chtějí využívat flexibilitu a výhody veřejného cloudu [3].

Komunitní cloud

Jedná se o distribuovaný systém, který je vytvořen integrací více služeb z různých cloudů. Je navržen tak, aby vyhovoval specifickým potřebám určité skupiny uživatelů, například odvětví nebo komunity. Tyto komunity může spojovat také stejná bezpečnostní politika, geografická lokace, nebo společné zájmy [4].

Komunitní cloud tak může být vhodný například pro subjekty, kterými je zdravotnický, mediální, či energetický sektor, a to z důvodu, že v těchto oblastech je velké množství generovaných dat, které mohou být částečně zveřejňovány. U zdravotnictví konkrétněji lze například uvažovat o globální platformě pro sdílení znalostí,



Public vs. private vs. hybrid cloud storage

	Public cloud storage	Private cloud storage	Hybrid cloud storage
Scalability	Very high	Limited	Very high
Security	Good, but depends on the security measures of the service provider	Most secure, as all storage is on premises	Very secure; integration options add an additional layer of security
Performance	Low to medium	Very good	Good, as active content is cached on premises
Reliability	Medium; depends on internet connectivity and service provider availability	High, as all equipment is on premises	Medium to high, as cached content is kept on premises, but also depends on connectivity and service provider availability
Cost	Very good; pay-as-you-go model and no need for on-premises storage infrastructure	Good, but requires on-premises resources, such as data center space, electricity and cooling	Improved, since it allows moving some storage resources to a pay-as-you-go model

Obr. 1.1: Porovnání modelů cloudového nasazení [1]

kdy komunitní cloud nabízí jako výhodu nižší náklady. Dalším příkladem vhodnosti tohoto modelu je možnost práce více lidí z různých firem a organizací na stejných projektech [3].

1.2 Servisní modely

Servisní model cloud computingu určuje, co je v rámci cloudové služby nabízeno a jak je poskytována. Odlišnosti mezi těmito modely jsou nejen způsob spravování a čerpání služby, ale také úrovní flexibility a rozdělením odpovědnosti mezi provozovatele a uživatele. Nejvíce cloudových služeb spadá do 4 hlavních modelů, které jsou představeny v této části práce.

IaaS

Infrastruktura jako služba (IaaS) je nejjednodušší model cloudové služby. Dodavatel poskytuje spotřebiteli virtualizované servery, sloužící pro provoz aplikací a platforem, které si spotřebitel následně spravuje sám. Kromě CPU, RAM a systémového disku mohou být virtualizovány i další zdroje, jako jsou síťová připojení nebo datová úložiště. Spotřebitel si na virtuální server může nahrát vlastní operační systém, dle nabídky poskytovatele, která obvykle zahrnuje všechny nejčastěji používané operační systémy. Součástí IaaS mohou být i další služby, jako jsou firewally, monitoring nebo zálohování.

Poplatek za IaaS je účtován podle spotřeby zdrojů spotřebitelem. Výhodou tohoto přístupu je, že poskytovatel se stará o veškerou správu a údržbu hardwaru, zatímco spotřebitel získává přístup k infrastruktuře o vysoké kvalitě [6].

PaaS

Platforma jako služba (PaaS) je pokročilejší model cloudové služby, který poskytuje spotřebiteli výpočetní platformu pro vytváření a provoz aplikací. Platforma nejčastěji zahrnuje operační systém, programové prostředí, databázi a webový server. Navíc jsou vývojářům aplikací k dispozici nástroje, prostředí a programovací jazyky, které jsou podporované poskytovatelem, proto pro vývoj a provoz aplikací není nutné kompletní porozumění HW požadavkům.

Poskytovatel na sebe přebírá povinnost o správu a údržbu platformy, včetně aktualizací a bezpečnostních záplat. Díky této větší odpovědnosti a množství nabízených služeb, je tato varianta cloudu dražší, než IaaS.

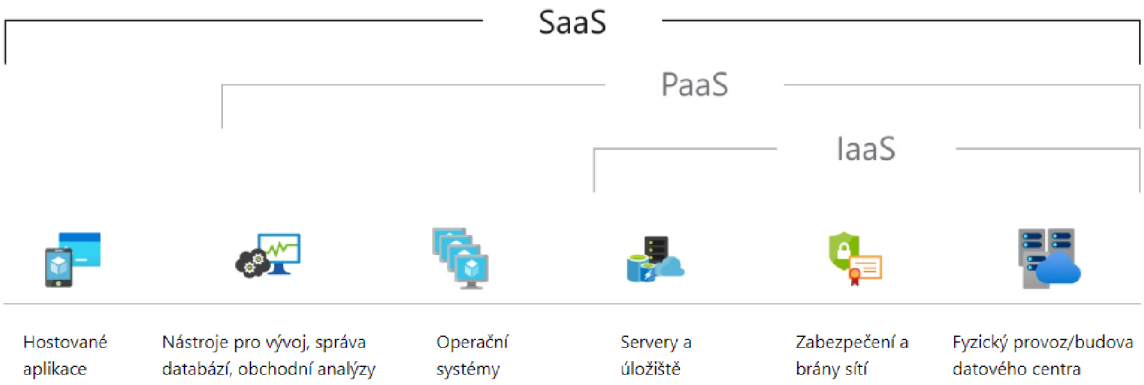
Problémem u tohoto modelu bývá, z důvodu hlubokého propojení s platformou poskytovatele, uzamčení do jednoho PaaS systému (tzv. vendor lock-in). Pokud se uživatel rozhodne přejít na jiného poskytovatele, může být migrace aplikace obtížná [7].

SaaS

Software jako služba (SaaS) je koncept úplného softwarového řešení, který poskytuje spotřebiteli již hotové aplikace, které jsou hostovány v cloudu. Aplikaci spravuje většinou poskytovatel cloudu a spotřebitel ji může na vyžádání využívat nejčastěji prostřednictvím webového rozhraní. SaaS aplikace jsou přístupné bez předchozí instalace, takže uživatelé mohou pracovat odkudkoli. Aplikace jsou také nezávislé na platformě a operačním systému, takže je mohou používat uživatelé s různými zařízeními a systémy.

SaaS je výhodný model pro organizace, které chtějí rychle a snadno začít používat nové aplikace. Uživatelé nemusí investovat do vlastního softwaru nebo infrastruktury, a proto mohou snížit své počáteční náklady.

V tomto modelu zákazníci nepotřebují spravovat ani aplikaci, ani infrastrukturu, která ji podporuje. Všechno je umístěno v datovém centru poskytovatele služeb, který se o vše stará a nese za to odpovědnost. SaaS služby jsou efektivní, snadno konfigurovatelné a velmi flexibilní, díky snadné škálovatelnosti. Snižují náklady na nákup softwaru, licencí a technickou podporu. Mezi příklady SaaS služeb patří portfolio Google Apps (dokumenty, Tabulky, Google Disk a další), Slack, Adobe Creative Cloud a další [8, 6].



Obr. 1.2: Porovnání servisních modelů cloudu z pohledu poskytovaných služeb [9]

1.3 Bezpečnost cloudových služeb

Základní principy zabezpečení dat jsou stejné, ať už jsou data umístěna na lokálním serveru, nebo v cloudovém prostředí. Způsob, jakým se tyto principy uplatňují, se však v cloudu zcela odlišují. Cloudové prostředí představuje nové hrozby a výzvy, proto je nutné k zabezpečení dat přistupovat jiným způsobem. Z tohoto důvodu bylo definováno sedm základních požadavků na bezpečnost v oblasti cloudu [10].

- Dostupnost dat – data jsou dostupná kdykoli je uživatel potřebuje.
- Integrita dat – zabezpečení dat proti neautorizované modifikaci.
- Důvěrnost dat – ochrana informačního obsahu dat, včetně jejich ochrany při přenosu.
- Autorizace – přiřazení správného oprávnění pro přístup do úložiště.
- Autentizace – proces ověřování identity uživatele.
- Soukromí – data uživatele nesmějí být použita k jinému účelu, než jsou určena.
- Odpovědnost – poskytovatel musí být schopen prokázat každou provedenou akci na úložišti.

Bezpečnostní hrozby u cloudu

Podle průzkumu CSA se mezi největší hrozby u claudových systémů řadí [11, 12]:

- Nedostatečná správa práv a přístupů do systému,
- Nezabezpečená rozhraní a API,
- Úniky, ukradení a ztráta dat,
- Systémové zranitelnosti,
- Distributed Denial of Service (DDoS) útoky.

1.3.1 Nedostatečná správa práv a přístupů do systému

Nedostatečná správa přístupu k datům v cloudu může vést k únikům a krádežím dat, které jsou na severech uloženy. Tyto problémy s autentizací vznikají už u uživatelů, kteří nedodržují základní doporučené bezpečnostní postupy, mezi které patří silná hesla, neopakování již použitých hesel nebo používání vícefaktorové (alespoň dvoufaktorové) autentizace přístupu k systému. Pro ideální zabezpečení je také doporučeno používat infrastrukturu veřejných klíčů (PKI). PKI zajišťuje správu kryptografických klíčů, které se používají pro šifrování a dešifrování dat.Kromě správy přístupu je důležitá také správná údržba systému. To zahrnuje pravidelnou aktualizaci databáze s přístupy, zejména odebírání přístupu pro osoby, které již dané práva nepotřebují. Dalším důležitým aspektem je aktivní monitorování dat a vyhodnocování rizik. To může pomoci odhalit potenciální bezpečnostní hrozby a podniknout kroky k jejich odstranění. Špatné nastavení přístupu a identit do systému postihuje zejména modely IaaS a PaaS. Za nedostatečné nastavení práv je vždy odpovědný zákazník. Poskytovatel cloudu pouze poskytuje nástroje pro správu a nastavení přístupu [11, 12].

1.3.2 Nezabezpečená rozhraní a API

Poskytovatel cloudových služeb umožňuje uživatelům komunikovat s těmito službami pomocí softwarových rozhraní (API). API poskytují řadu výhod, včetně možnosti propojování aplikací, správy úložiště a monitorování služeb. Jsou tedy velmi důležitým prvkem bezpečnosti a dostupnosti cloudových služeb, které jsou na nich závislé.

Příkladem služby, která využívá API pro zabezpečení cloudových řešení, je webový aplikační firewall (WAF). WAF používá API k filtrování nechtěného provozu a ochraně aplikací před útoky. Je proto důležité dbát na správné zabezpečení API. To zahrnuje kontrolu z hlediska zranitelností, chyb v kódu a nedostatečné autentizace. Mezi časté chyby v nastavení API patří například chybějící pravidelná aktualizace API verze, nadměrné oprávnění uživatelů nebo často slabá autentizace [12].

1.3.3 Úniky, ukradení a ztráta dat

Únik dat nastává, když někdo, kdo nemá oprávnění, získá přístup k chráněným nebo citlivým datům. Citlivá a chráněná data jsou informace, které nejsou určeny k veřejnému zveřejnění. Mohou zahrnovat osobní údaje, obchodní tajemství nebo finanční informace.

Únik těchto dat může nastat několika způsoby, mezi které patří například cílený útok na aplikaci, špatně nastavenými bezpečnostními postupy a v současné době

často lidskou chybou, kdy zaměstnanec omylem zveřejní data, nebo jsou z něho vylákána. Této hrozbě podléhají všechny tři typy servisních modelů (IaaS, PaaS i SaaS).

Únik dat se řadí mezi jedny z největších hrozeb z důvodu, že data mají velkou obchodní hodnotu. Ztráta dat může mít negativní dopad na pověst společnosti, důvěru zákazníků, ztrátu duševní vlastnictví a může vést k regulačním postihům a finančním ztrátám. Nejlepším způsobem, jak se proti únikům bránit je multifaktorová autentizace, řízení přístupu, šifrování klientských dat silnými klíčy a časté zálohování těchto dat [11].

1.3.4 Systémové zranitelnosti

Jedná se nedostatky vyskytující se potenciálně ve všech cloudových platformách, komponentech, službách cloudu, které mohou být zneužity útočníky pro narušení chodu cloudu. Stejně jako v předchozí kategorii, postihují všechny tři servisní modely cloudových služeb [12].

Mezi hlavní kategorie systémových zranitelností patří:

- Slabé nebo výchozí přihlašovací údaje – útočník může snadno získat přístup do systému, pokud jsou použita nedostatečně silná autentizační data, nebo jsou uložena v nezabezpečeném prostředí.
- Zranitelnosti nulového dne – existují již od vytvoření systému, ale nejsou zatím známy. Jsou velmi nebezpečné, protože proti nim neexistuje efektivní obrana, kromě aktivního vyhledávání a aktualizování systému.
- Chybějící bezpečnostní záplaty – může nastat, pokud je objevena a zveřejněna zranitelnost nulového dne. Taková zranitelnost může být útočníkem snadno zneužitelná, proto by se měly služby pravidelně aktualizovat se záplatami nově objevených zranitelností.
- Chybná konfigurace prvků – tato zranitelnost vzniká, pokud jsou prvky cloudového systému konfigurovány nesprávně. Příkladem může být použití výchozích přihlašovacích údajů nebo nastavení, slabých šifrovacích algoritmů a protokolů, nebo spouštění nepotřebných služeb.

1.3.5 Distributed Denial of Service (DDoS)

Útoky odepření služby (DoS a DDoS) jsou druhy kybernetických útoků, které mají za cíl omezit nebo zcela znepřístupnit službu uživatelům. Útočník k tomu využívá zasílání velkého množství neplatných nebo nesmyslných požadavků na službu. To může vést k vyčerpání zdrojů serveru, na kterém služba běží, a k jejímu výpadku nebo znatelnému omezení výkonu. Existují dvě varianty tohoto útoku. První je DoS, který

útočí pouze z jednoho uzlu (místa).Druhým typem je DDoS a je mnohem nebezpečnější, protože je prováděn z více uzlů, které spolupracují, a proto je útočník schopen vyvinout daleko větší sílu. Toho lze dosáhnou například využitím sítě botů (tzv. botnetu), což jsou pomocí škodlivé aplikace kompromitované zařízení jako osobní počítače, mobilní nebo IoT zařízení a další, které má pod kontrolou útočník.

Ochrana proti těmto útokům je obtížná, ale existují určité kroky, které organizace mohou podniknout, aby snížila jejich riziko. Důležité je správné rozeznání mezi útokem a normálním provozem, popřípadě časová limitace požadavků. Dále se lze bránit použitím firewallů , bezpečné síťové infrastruktury a redundantních linek v této infrastruktuře, DDoS filtrů nebo IPS a IDS systémů [12].

2 Nextcloud

V této kapitole je rozebrán základní popis Nextcloudu a dále pak jeho bezpečnostní prvky rozdělené do dvou skupin.

2.1 Co je Nextcloud

Nextcloud je sada klient-server aplikací s open-source kódem (otevřený zdrojový kód) pro privátní či hybridní cloud. Tento software lze nainstalovat a provozovat na vlastním soukromém serveru a je vhodný jak pro jednotlivce, tak pro malé, střední i velké podniky. Nextcloud umožňuje sdílet a synchronizovat soubory a složky se serverem, podobně jako známá komerční řešení (například Dropbox). Navíc nabízí místní úložiště se silným zabezpečením a možností plné kontroly nad správou. Také umožňuje přehlednou a snadnou manipulaci ze strany uživatele i správce prostřednictvím webového rozhraní, jakož i mobilních a desktopových klientů pro Windows, Mac, Linux, Android a iOS. Podle zvoleného plánu je dostupná podpora migrace souborů i profesionální zákaznický servis. Rozšiřitelnost základní funkcionality je zajištěna pomocí doplňujících aplikací přímo od vývojářů nebo komunity. Také lze vyvinout vlastní aplikaci dle svých potřeb. Nextcloud zabezpečuje přenos souborů pomocí protokolu SSL/TLS a přímo na uložišti pomocí 256-bitového šifrování AES [13, 14].

2.2 Bezpečnostní prvky

V této podkapitole se rozebírají konkrétní bezpečnostní prvky Nextcloudu, které jsou rozděleny do dvou skupin, a to *Nástroje* a *Aplikace*.

2.2.1 Nástroje

Deaktivování nástroje enable preview

Nextcloud umí generovat náhledy běžných typů souborů, jako jsou obrázky nebo textové soubory. Tyto náhledy jsou generovány pomocí knihoven PHP napsaných v jazyce C a mohou být zranitelné vůči některým útokům [15].

Ve výchozím nastavení je pro soubory, které jsou považovány za dostatečně bezpečné, generování náhledů povoleno. Avšak pro lepší zabezpečení je doporučeno generování náhledů zakázat [15].

To se provádí následujícím příkazem, zadaným v našem případě do souboru `nextcloud-config.sh`:

Výpis 2.1: Příkaz pro deaktivaci nástroje enable preview

```
php /var/www/html/occ config:system:set --type boolean
                          --value false enable_previews
```

Deaktivování nástroje debug

Nástroj `debug` je určen pro lokální vývoj a použití v kontrolovaných prostředí, a v produkčních prostředích nebo mimo cílené řešení problémů by neměl být povolen. Pokud je povolen, pak se jím povolují věci, jako např. výpisy kolekcí WebDAV pro celý server [15]. Deaktivování se provádí následujícím příkazem, zadaným v našem případě do souboru `nextcloud-config.sh`:

Výpis 2.2: Příkaz pro deaktivaci nástroje debug

```
php /var/www/html/occ config:system:set --type boolean
                                    --value false debug
```

ClamAV

Jedná se o open-source (licence GPLv2) antivirovou sadu nástrojů. Mezi řadou nástrojů, které poskytuje, patří také flexibilní a škálovatelný daemon, skener pro příkazový řádek a pokročilé nástroje pro automatickou aktualizaci databáze. Jádrem balíku je antivirový engine dostupný ve formě sdílené knihovny [16].

Mezi výhody patří rychlé skenování souborů, ochrana v reálném čase, detekování virů, trojských koní a dalšího malwaru, včetně makrovirů Microsoft Office, mobilního malwaru a dalších hrozeb [16].

2.2.2 Aplikace

Nextcloud poskytuje "Nextcloud App Store", což je knihovna přídavných aplikací (modulů) vyvinutých externími vývojáři (na některých spolupracovali vývojáři z Nextcloudu) na poskytnutém API od Nextcloud. Tyto aplikace rozšiřují funkcionalitu cloudového prostředí například o další bezpečnostní prvky, nástroje na dohled, organizování, hledání a další. Lze si také na poskytnutém API vyvinout vlastní modul a přípádně ho nasdílet pro ostatní uživatele.

Pokud chceme výrazněji konfigurovat instalované aplikace, provádí se tak v `nextcloud-config.sh` pomocí dosazením do obecného příkazu.

Výpis 2.3: Obecný příkaz pro konfiguraci v nextcloud-config.sh

```
php /var/www/html/occ config:app:set <app> <name>
                                    [--value VALUE]
```

Kdy `<app>` je název aplikace (např.`bruteforcesettings`), `<name>` je název atributu (např. `whitelist_1`) a `VALUE` je hodnota daného úkonu, například `1.1.1.1/24`.

Protekce proti Brute Force útokům

Brute Force útok na prolomení hesla je prováděn zkoušením velkého množství různých hesel k přihlášení. Může buďto využívat velký seznam běžně používaných hesel, nebo způsob, kde se využívá slovníku slov anebo testování všech možných kombinací znaků.

Pokud je nástroj `bruteforcesettings` na ochranu proti brute force (hrubé síle) spuštěn, zpomalí opakující se požadavky přicházející z jedné IP adresy na chráněný řadič (controller) se stejným AP na dobu 24 hodin, a to postupně až o 25 vteřin. Jakmile je však přihlášení úspěšné, zpomalení se vyresetuje. Ztěžuje také útoky prostřednictvím formuláře pro obnovení hesla nebo pokusů o nalezení tokenů hesel k aplikací [17].

Aktivování nástroje se provádí, v našem případě, zadáním následujícího příkazu do souboru `nextcloud-config.sh`.

Výpis 2.4: Příkaz pro aktivaci brute force ochrany

```
php /var/www/html/occ app:enable bruteforcesettings
```

Lze také vyjmout IP adresu nebo rozsah z této ochrany, což je užitečné například pro účely testování, nebo v případě false-positive výsledků z důvodu více lidí na jedné IP adrese [17]. Toto vyjmutí se provádí zadáním následujícího příkazu do stejného souboru, jako je aktivace nástroje:

Výpis 2.5: Příkaz pro vyjmutí ip adresy z ochrany

```
php /var/www/html/occ config:app:set bruteForce
                  whitelist_1 --value 1.1.1.1/24
```

Antivirus for files

Díky této aplikaci lze nakonfigurovat Nextcloud server tak, aby automaticky prováděl scan a antivirovou kontrolu nahraných souborů. Využívá se zde open source antivirový engine ClamAV. Ten umí detekovat všechny formy malwaru včetně trojských koní, virů a počítačových červů. Funguje na všech hlavních typech souborů, jako jsou spustitelné nebo komprimované soubory, obrazové soubory, PDF, soubory Flash, soubory pro Windows, Linux i Mac a další. V plánovaných intervalech Démon ClamAV Freshclam aktualizuje automaticky databázi signatur [18].

Nejprve je nutné naistalovat ClamAV, to lze provést v souboru `nextcloud.Dockerfile`:

Výpis 2.6: Příkazy pro instalaci ClamAV enginu

```
RUN apt install -y clamav clamav-daemon
RUN freshclam
```

Dále se instaluje samotný Antivirus pro soubory v `nextcloud-config.sh`:

Výpis 2.7: Příkaz pro instalaci aplikace Antivirus pro soubory

```
php /var/www/html/occ app:enable files_antivirus
```

V případě, že chceme změnit konfiguraci, jako je například mód, ve kterém běží nebo specifikovat, co se má udělat s infikovaným souborem, přidáme příkaz do stejného souboru (`nextcloud-config.sh`) dle vzorové příkazu v úvodu 2.2.2, například pro nastavení reakce při objevení infikovaného souboru na smazání tohoto souboru:

Výpis 2.8: Příkaz pro nastavení činnosti smazání při objevené infikovaného souboru

```
php /var/www/html/occ config:app:set files_antivirus
                    av_infected_action --value delete
```

GeoBlocker

Jedná se o frontend pro geolokační služby, který umožňuje blokování, zpoždění a logování pokusů o přihlášení z vybraných zemí. Nejsou blokovány, odkládány ani zaznamenávány pokusy o přihlášení z IP adres místní sítě. Aktuální verze zobrazuje přihlašovací stránku normálně všem, ale samotné přihlášení je zablokováno z IP adres mimo zvolené země [19].

Jedná se o jednoduché, avšak účinné zabezpečení, neboť většina škodlivého provozu přichází ze zahraničí. Aplikace umožňuje na výběr dva módy výběru států (allowlist a blocklist), kdy mého názoru je pro naše použití vhodnější použití allowlistu a vyselektování pouze České republiky, protože se nepředpokládá používání systému z jiné země. Dále jsou zde různé možnosti, co vše se má logovat (IP adresa, kód země, přihlašovací jméno), co dělat s provozem mimo vybranou zemi (zpoždění pokusů s přihlášením o 30 vteřin nebo zablokování těchto pokusů) a také zde je na výběr použití ze 3 databází (GeoIPLookup, MaxMind GeoLite2, RIR Data). Tyto databáze je nutné stáhnout a doinstalovat separátně do systému, nejsou součástí aplikace.

Příkaz na aktivování aplikace (v souboru `nextcloud-config.sh`):

Výpis 2.9: Příkaz pro aktivování aplikace GeoBlocker

```
php /var/www/html/occ app:enable geoblocker
```

Ransomware protection

Jedná se o jednoduchou prevenci proti nahrátí ransomwaru na server. Pokud je nahráván soubor na server jehož název odpovídá vzoru ransomwaru z integrovaného listu vzorů, je nahrání zablokováno. Pokud je problém s nahráním souboru, u které ověřeno, že není škodlivý, lze přidat vzor tohoto souboru na seznam pro vyloučení ze zablokování.

Bohužel tato aplikace není podporována v námi používané verzi Nextcloudu a z tohoto důvodu nebyla implementována, ale aktivování aplikace by proběhlo obdobně jako dříve zmíněné aplikace [20].

3 Docker

Kapitola je úvodem do Dockeru, včetně rozebrání klíčových komponent Dockeru a jeho výhod.

Docker je platforma pro vývoj, distribuci a běh aplikací v kontejnerizovaném prostředí. To umožňuje oddělit aplikace od infrastruktury pro rychlejší distribuci softwaru, takže lze snadno přenášet aplikace mezi různými prostředími, jako jsou vývojové, testovací a nakonec produkční servery [23].

3.1 Komponenty Dockeru

Docker obsahuje pět hlavních komponentů, a to Docker Engine, Docker Images, Docker Containers, Dockerfile a Docker Hub [23].

Docker Engine

Umožňuje vytvářet, spouštět a spravovat kontejnery. Skládá se ze tří podkomponentů.

1. **Docker Daemon** - služba, která běží na hostitelském systému a spravuje všechny Docker objekty, jako jsou obrazy (Docker Images), kontejnery (Docker Containers), sítě a svazky. Přijímá API požadavky od Docker CLI a komunikuje s dalšími Docker Daemony pro správu Docker služeb [23].
2. **Docker CLI** - jedná se o příkazový řádek, který umožňuje uživatelům komunikovat s Docker Daemonem. Poskytuje řadu příkazu pro vytváření (`docker build`), spouštění (`docker run`), zastavování (`docker stop`) a správu (`docker pull`, `docker push`) kontejnerů a obrazů. Existují ovšem i další příkazy, například příkaz `docker compose [možnosti]`, který je využíván v mojí aplikaci (převzat z původní DECT Nextcloud aplikace), z důvodu, že umožňuje vytvářet a spravovat vícero služeb v Docker kontejnerech najednou [23].
3. **Docker API** - je rozhraní, díky kterému je umožněno aplikacím programově komunikovat s Docker Daemonem. Poskytuje také HTTP API, které lze použít k automatizované integraci nějaké Docker funkcionality do vlastních aplikací a nástrojů [23].

Docker Image

Docker obrazy se vytváří z Dockerfile a jedná se o nezměnitelné šablony, obsahující aplikaci a všechna její prostředí a závislosti. Můžou pocházet přímo ze zdrojových

kódů, anebo se stahují předpřipravené ze společných úložišť, jako je například Docker Hub [23].

Docker Container

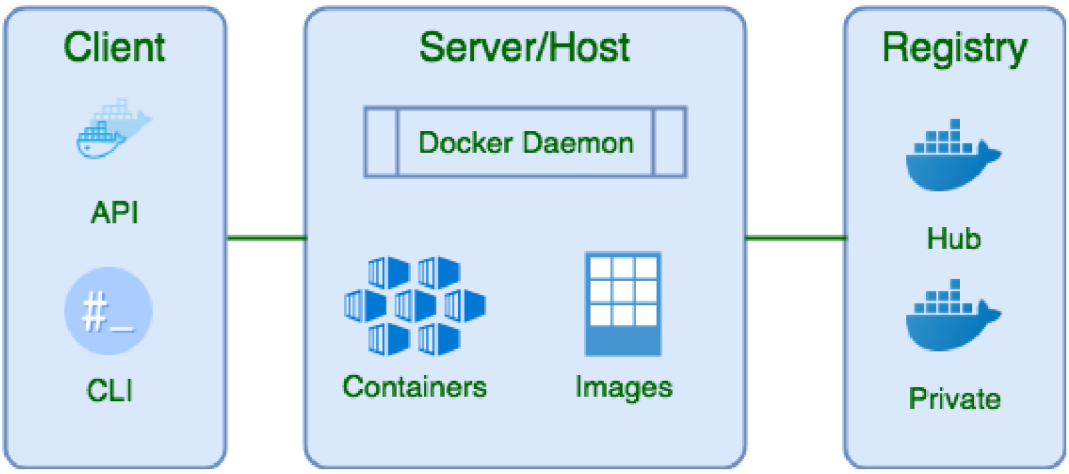
Spustitelné instance Docker obrazů, které jsou izolované jak od hostitelského systému, tak i od sebe navzájem. Každý kontejner obsahuje věci potřebné k běhu aplikace, jako je kód, runtime, knihovny a konfigurace [23].

Dockerfile

Jedná se o textový soubor s příkazy, které definují, jak se vytváří Docker Image. Obsahuje instrukce jako například `FROM`, `RUN`, `COPY`, `CMD`, které specifikují základní obraz, příkazy ke spuštění a kopírování souborů, a výchozí příkaz kontejneru [23].

Docker Hub

Docker Hub je největší veřejné úložiště Docker obrazů, dostupné online, kde mohou uživatelé sdílet a stahovat obrazy. Je nutné si ovšem hlídat věrohodnost vývojáře, protože neznámé obrazy mohou představovat bezpečnostní riziko. Mimo to také Docker Hub nabízí privátní repozitáře pro ukládání vlastních soukromých obrazů [23].

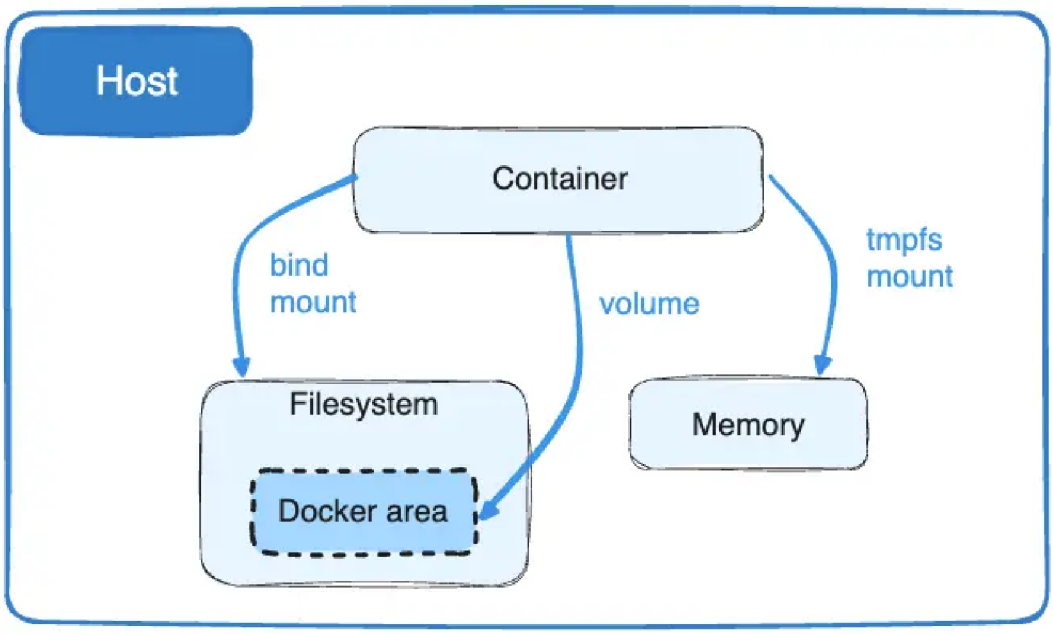


Obr. 3.1: Schéma architektury Dockeru [21]

Doceker Volumes

Docker Volumes, neboli svazky, je preferovaný mechanismus pro uchovávání dat generovaných a používaných kontejnery Docker. Jsou kompletně spravovány systémem

Docker. Často jsou svazky lepší volbou, než uchovávání dat v zapisovatelné vrstvě kontejneru, a to z důvodu, že nezvětšují velikost kontejnerů, které ho používají. Obsah těchto svazků existuje mimo životní cyklus daného kontejneru a může být sdílen mezi více kontejnery [24].



Obr. 3.2: Srovnání místa ukládání dat různých metod v Dockeru [24]

3.2 Výhody Dockeru

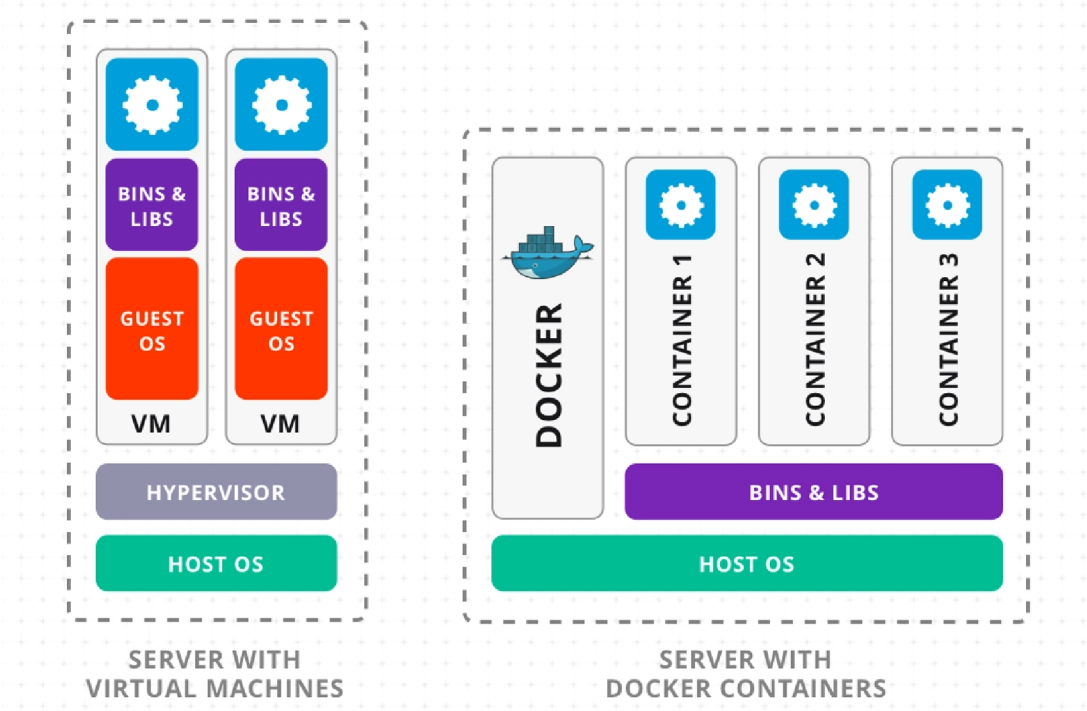
Mezi hlavní výhody Dockeru patří konzistence prostředí, kdy kontejnery zajišťují, že aplikace poběží stejně na jakémkoliv systému, který podporuje Docker.

Dále Efektivní využití zdrojů, jelikož kontejnery sdílejí jádro operačního systému, a to umožňuje vyšší hustotu aplikací na stejném hardware ve srovnání s virtuálními stroji, které hardware daleko více zatěžují.

To souvisí s následující výhodou, kterou je rychlé nasazení (kontejnery mají rychlý proces spuštění) a dobrá škálovatelnost aplikací podle potřeb.

Kontejnery běží izolovaně na sobě i hostitelském systému, díky čemuž se zvyšuje bezpečí a zabraňuje se konfliktům mezi aplikacemi.

Neposlední takovou velkou výhodou je také přenositelnost, kdy kontejnery mohou být snadno přesouvány mezi různými prostředími určenými třeba na vývoj, testování nebo produkci [22].



Obr. 3.3: Srovnání architektury Dockeru a virtuálního stroje [22]

4 Logovaní

V této kapitole je rozebrána problematika logování, dále pak využití a typy typy logů, fungování logů v Nextcloudu, původní řešení zobrazování logů a na závěr kritéria pro správce logů (nové řešení zobrazování a práce s logy).

4.1 Obecné využití logů

Logování slouží k zaznamenávání událostí odehrávajících se v počítačových systémech a aplikacích. Tyto záznamy označujeme jako logy. Můžou obsahovat různé druhy informací, od chybových hlášek a diagnostických zpráv, po bezpečnostní události nebo uživatelské aktivity.

Logy můžou být generovány v různých formátech, včetně textových souborů, specializovaných logovacích systémů nebo databází. Například Nextcloud používá jako svůj výchozí formát JSON. Také mohou být generované na různých úrovních, ale vždy je důležité, aby měli správnou strukturalizaci a byli snadno přístupné pro analýzu.

Z pohledu účelu lze logování rozdělit do několika skupin:

Bezpečnost

Bezpečnostní logy se používají pro záznam přístupů do aplikace, včetně informací kdo se přihlásil, popřípadě pokusil o přihlášení, kdy se to odehrálo a odkud se daný člověk do systému přihlásil. To napomáhá při sledování a analýze přístupů do aplikace [25].

Audit a sledování

Některé odvětví jsou ze zákona povinné uchovávat záznamy pro účely regulace a auditu. To umožňuje ověřit správnost a integritu operací nebo procesů v systému [25].

Monitorování výkonu

Logování se také může využívat při sledování zatížení systému, jako třeba využití pamětí, zatížení procesoru, a nebo využití síťového provozu. Na základě těchto logů pak mohou administrátoři a vývojáři provádět úpravy vedoucí ke zlepšení výkonu systému [25].

Ladění a diagnostika

Logy spadající pod tuto skupinu jsou obzvlášť vhodné pro vývojáře, protože jim poskytují detailní informace o tom, co aplikace provádí v daném čase. Dále zaznamenávají a zobrazují, když se v aplikaci odehraje nějaká chyba, často obsahují proto informaci kde, kdy a proč se tato chyba vyskytla. Poskytují také uživateli další diagnostické údaje, pro lepší nastavení a optimalizaci aplikace [25].

Analýza a reporting

Mohou být užitečné také při vytváření reportů pro manažery a další osoby v rozhodující pozici, kterým poskytují přehled o stavu a provozu systémů a aplikací. Na základě historických dat lze provádět predikce trendů do budoucnosti, nebo identifikovat potenciální problémy dříve, než by k nim došlo. Také je lze analyzovat za účelem vytváření statistik, jako například počet určitých operací, chybovost v systému nebo počet uživatelů používající danou aplikaci [25].

Pohled do historie

Logy jsou zaznamenávány včetně času, kdy se daná událost odehrála. To umožňuje zpětné vyhledávání a sledování toho, co se v systému stalo. Dále lze v případě incidentů nebo problémů dohledat a analyzovat, jaké kroky či události vedly ke konkrétnímu stavu a kdo byl za tyto kroky zodpovědný (kdo je provedl). Logy mohou být ukládány po dlouhou dobu, aby bylo možné v budoucnosti provést již zmíněnou analýzu nebo srovnání s aktuálními logy [25].

4.2 Typy logů

Obecně existuje několik typů logů, které dělíme podle kategorie událostí, které zaznamenávají. Tyto kategorie nejsou konzistentní a každý může mít trochu jiné rozdělení, ať už podrobnější do spoustu kategorií, nebo obecnější, kde jsou podobné kategorie pospojovány. Dle mého názoru můžeme logy rozdělit do následujících hlavních kategorií.

Aplikační logy

Tyto logy obsahují informace specifické pro danou aplikaci, jako jsou interní chyby, stavové změny, operace prováděné uživateli nebo chyby při používání aplikace [26, 27].

Systémové logy (syslog)

Činnosti operačního systému, mezi které patří například spouštění a ukončování služeb, změny konfigurace nebo systémové chyby a varování [26, 27].

Bezpečnostní logy

Tyto logy sledují události, které jsou významné pro bezpečnost systému. Jsou také klíčové pro detekci a analýzu bezpečnostních incidentů. Jejich obsah je tvořen například úspěšnými a neúspěšnými pokusy o přihlášení, změnami hesel a jinými autentizačními událostmi, záznamy s přístupy k citlivým souborům, změnami přístupových oprávnění a dalšími [26, 27].

Auditní logy

Poskytují detailnější informace o všech významných aktivitách v aplikaci. Často jsou potřeba pro zajištění kontroly dodržování předpisů a provádění auditu. Obsahují detailní záznamy o akcích uživatelů, včetně provedených operací, přístupům k datům, změn konfigurací. Dále také umožňují sledovat, kdy kdo provedl jaké změny v datech [26, 27].

4.3 Logování v Nextcloudu

Úrovně logování

Nextcloud umožňuje nastavit 5 úrovní zaznamenávání událostí, které se vypíší do logů, od nejnižší úrovně *DEBUG*, která obsahuje všechny aktivity co se stanou na Nextcloud serveru, až po nejvyšší úroveň *FATAL*, při které se zaznamenávají pouze fatální události serveru. Tyto úrovně jsou následující:

- **0**: DEBUG: Zaznamenána veškerá aktivita, nejvíce detailní úroveň.
- **1**: INFO: Aktivity jako přihlášení uživatelů a činnosti se soubory, dále varování, chyby a fatální chyby.
- **2**: WARN: Do této úrovně se řadí úspěšné operace, které však mají upozornění na případné problémy. Dále opět chyby a fatální chyby.
- **3**: ERROR: Zaznamenány operace, které se nezdařily, ale ostatní operace a služby na serveru dále běží a pokračují, plus fatální chyby.
- **4**: FATAL: Zastavení serveru.

Při instalaci Nextcloud serveru bez další konfigurace je standardně nastavena úroveň logování na WARN (2). Úroveň DEBUG (0) by měla být pouze v případě, když se potýkáme s problémem, který je potřeba diagnostikovat, poté by měla být

úroveň navrácena na některou z vyšších úrovní, a to z důvodu, že úroveň DEBUG vydává mnoho informací a může ovlivnit výkon serveru [28].

Parametry úrovně zaznamenávaní logů se standardně konfigurují příkazem `'loglevel' => 1,` v souboru `config/config.php`. V našem případě, kdy veškerá konfigurace aplikuje již při instalaci serveru pomocí skriptu, uložíme příkaz pod tímto odstavcem do skript souboru `nextcloud-config.sh`, který ji při spuštění zapíše do již zmíněného `config/config.php`.

4.4 Původní řešení

Při převzetí úkolu navrhnout řešení bezpečného ukládání a správy logů bylo pouze zapnuté auditní logování, které se ukládalo do souboru `audit.log`. Auditní logování se zapíná v postinstalačním konfiguračním souboru `nextcloud-config.sh` a to příkazem:

Výpis 4.1: Příkaz pro zapnutí auditního logování

```
php /var/www/html/occ config:system:set --type string
   --value /var/www/html/data/audit.log logfile_audit
```

```
# LOGGING
php /var/www/html/occ config:system:set --type integer --value 1 loglevel
php /var/www/html/occ config:system:set --type string --value /var/www/html/data/audit.log logfile_audit
php /var/www/html/occ config:system:set --type string --value "d.m.Y - H:i:s" logdateformat
php /var/www/html/occ config:system:set --type string --value "admin_audit" log.condition apps 0
```

Obr. 4.1: Původní konfigurace logování

4.5 Kritéria pro správce logů

Hlavním kritériem při výběru vhodného řešení aplikace pro správu logů byl výběr samostatně hostované služby, která by běžela na vlastních serverech. Výhodou tohoto self-hosted řešení je kompletní kontrola nad infrastrukturou a daty, kdy můžeme přizpůsobovat hardwarové a softwarové komponenty, což umožňuje škálovatelnost aplikací včetně optimalizace výkonu. Mezi další výhody patří také zvýšená bezpečnost, která je dána zejména vlastnictvím infrastruktury. To umožňuje lepší kontrolu nad ukládanými daty a při práci s nimi [29].

Dalším kritériem byla přehlednost jednotlivých záznamů, jelikož integrovaná aplikace logmanageru v Nextcloudu (musí se ovšem aktivovat) neumožňuje komplexnější vyhledávání, filtraci a celkový přehled mezi log záznamy.

Při výběru správce logů byla také brána v potaz odbornost personálu, který bude se správcem logů pracovat. Zatímco se základním cloudem Nextcloud budou pracovat i méně zdatní lidé v oboru IT, kteří tolik počítačům a počítačovým aplikacím nerozumí, tak u logmanageru se předpokládá, že s ním budou operovat lidé, kteří mají alespoň nějaký kurz na práci v IT oboru, a to i z důvodu složitosti porozumění datům (záznamům logů), se kterými budou ve styku.

5 Navržené řešení správy a uchovávání logů

V této kapitole je představen návrh řešení správy a uchovávání logů.

Konečné řešení správce logů, které jsem zvolil je postavené na open-source softwaru, jejímž hlavním vývojářem je společnost Grafana Labs. Skládá se ze tří částí (Grafana, Loki, Promtail), které si rozdělují činnosti od sběru logů od Nextcloudu, přes dopravení do správce logů, až po konečné zobrazení uživateli, který nad nimi může vykonávat různé úkony jako například filtraci nebo jednodušší vyhledávání. Je zde využita technologie Docker, viz popsána v kapitole č. 3, kdy každá část má svůj vlastní Docker kontejner.

Obr. 5.1: Schema architektury Grafany

5.1 Architektura souborů pro službu Grafana

Níže je znázorněna architektura nově přidaných složek a souborů, které se týkají služeb Promtail a Loki. Vycházejí ze složky `apps`, kam se přidávají konfigurace pro dodatečné aplikace k Nextcloudu. Složka `apps` je umístěna v root adresáři.

```
apps ..... Již existující složka, která sdružuje přidané aplikace
└── grafana ..... Složka pro grafana aplikaci
    ├── loki ..... Složka pro službu Loki
    │   ├── chunks
    │   ├── index
    │   ├── rules
    │   ├── wal
    │   └── loki-local-config.yaml ..... Konfigurační soubor pro Loki
    ├── prom ..... Složka pro službu Promtail
    │   └── promtail-local-config.yaml ..... Konfigurační soubor pro Promtail
    └── .env ..... Soubor s prostředím (ponechán prázdný)
```

Soubory již existující, u kterých byla provedena změna v obsahu z důvodu přidání služeb Grafana, Loki a Promtail (`docker-compose.yml` a `nextcloud.Dockerfile`)

leží přímo v root adresáři. Soubor `nextcloud-config.sh` je umístěn v adresáři s cestou `./scripts/post-installation/`, a byla do něho doplněna pouze konfigurace na generování logů viz kapitola 5.2.

Nezměněné soubory, které jsou součástí přidělené instance Nextcloud s již exitujícími externími aplikacemi (službami) zde nejsou uvedeny.

5.2 Nextcloud

Logy generuje Nextcloud dle konfigurace, kterou určíme. Pro mé řešení je zapotřebí zapnout jak sběr klasických logů, pojmenovaných Nextcloudem jako `logfile`, tak i auditních logů, pojmenovaných Nextcloudem `logfile_audit`.

Tato konfigurace vedoucí k zapnutí výše zmíněných logů se provádí v souboru `nextcloud-config.sh` (celá cesta k souboru je `./scripts/post-installation/nextcloud-config.sh`). Celou konfiguraci lze vidět na obrázku 5.2. Níže pak jsou rozebrány ty důležité konfigurační příkazy pro správnou funkčnost celého správce logů.

Tímto příkazem je nastavena potřebná úroveň logování, v mém případě je pro správné fungování nutná úroveň 1. Více o úrovních viz podkapitola 4.3.

Výpis 5.1: Příkaz pro nastavení úrovně logování

```
php /var/www/html/occ config:system:set --type integer
                                        --value 1 loglevel
```

Následujícími dvěma příkazy určíme místo uložení souborů s logy. První je zde soubor s auditními logy, nazvaný `audit.log`, následuje soubor s klasickými logy nazvaný `nextcloud.log`.

Výpis 5.2: Příkazy určující místo ulkádání souborů s logy

```
php /var/www/html/occ config:system:set --type string
   --value /var/www/html/data/audit.log logfile_audit
php /var/www/html/occ config:system:set --type string
     --value /var/www/html/data/nextcloud.log logfile
```

Dále obsahuje konfigurační soubor příkaz, který definuje, v jakém formátu budou logy ukládány. V mém případě je to formát `file`, ten odpovídá souboru ve formátu JSON. Opět jsou tyto příkazy dva (téměř identické), jeden pro auditní log, druhý pro klasický log.

Výpis 5.3: Příkazy pro nastavení formátu uložení logů

```
php /var/www/html/occ config:system:set --type string
                              --value file log_type_audit
php /var/www/html/occ config:system:set --type string
                                    --value file log_type
```

Posledním důležitým konfiguračním příkazem, je nastavení správného formátu pro záznam času, kdy se daná událost, která je zaznamenána, odehrála.

Výpis 5.4: Příkaz pro nastavení formátu časového razítka

```
php /var/www/html/occ config:system:set --type string
                --value ''d.m.Y - H:i:s'' logdateformat
```

```
# LOGGING
php /var/www/html/occ config:system:set --type integer --value 1 loglevel
php /var/www/html/occ config:system:set --type string --value /var/www/html/data/audit.log logfile_audit
php /var/www/html/occ config:system:set --type string --value /var/www/html/data/nextcloud.log logfile
php /var/www/html/occ config:system:set --type string --value file log_type_audit
php /var/www/html/occ config:system:set --type string --value file log_type
php /var/www/html/occ config:system:set --type string --value "d.m.Y - H:i:s" logdateformat
php /var/www/html/occ config:system:set --type string --value "admin_audit" log.condition apps 0
```

Obr. 5.2: Mé řešení konfigurace logování

Shrnuto, Nextcloud generuje dva soubory s logy ve formátu JSON, kdy klasický je pojmenovaný nextcloud.log a auditní je pojmenovaný audit.log. Oba tyto soubory jsou pak uloženy v adresáři `/var/www/html/data`, kdy `/var/www/html` je výchozí adresář pro ukládání věcí v přidělené instanci Nextcloudu. Tyto dva soubory jsou pak převzaty pomocí Promtail a je s nimi dále pracováno.

5.3 Promtail

Jedná se o agenta, který odesílá obsah lokálních logů do privátní instance Grafana Loki (můj případ) nebo Grafana Cloud. Obvykle je nasazován na každý stroj, na kterém běží aplikace, které je třeba monitorovat. V případě monitorování přiděleného cloudového úložiště Nextcloud se jedná pouze o jednu instanci (jeden server Nextcloudu). Promtail má tři základní kroky sběru dat:

1. Objevuje cíle
2. Připojuje štítky k log proudům
3. Odesílá je do Grafana Loki

Před odesláním dat do Grafana Loki, Promtail zjistí informace o svém prostředí. Konkrétně to znamená, že se musí objevit aplikace, které generují řádky logů do souborů, které jsou monitorovány [30].

Promtail si na to propůjčuje stejný mechanismus objevování služeb, jako používá i Prometheus (jedná se také o nástroj pro sběr a monitorování logů) [30].

Během objevování služby jsou určena metadata, jako například název podu, název souboru atd., která mohou být připojena k řádku logu jako štítek, pro následnou jednodušší identifikaci při dotazovaní se nad logy v Loki. Pro sofistikovanější filtrování, Promtail umožňuje nastavit štítky nejenom z objevených služeb, ale také na základě obsahu jednotlivých řádků logů [30].

Posledním krokem je nepřetržité sledování (čtení) logů z cílového souboru. Tento krok nastává, pokud je dostupná sada cílů (soubory ke čtení), a dále jsou všechny štítky správně nastaveny. Jakmile je do paměti načteno dostatečné množství dat dle nakonfigurovaného časového rozmezí, jsou hromadně odeslána do Lokiho, jako jedna dávka. Promtail si ukládá pozici posledních načtených logů, a to ve výchozí konfiguraci v souboru pozic umístěném v `/var/log/positions.yaml`, takže zde nedochází k duplicitnímu načtení logů, a to ani v případě, kdy dojde k restartování instance Promtail [30].

5.3.1 Konfigurace a instalace

Konfigurace a instalace služby Promtail se provádí v několika souborech. Mezi tyto soubory patří soubory již existující (`docker-compose.yml`) a soubory nutné nově vytvořit (`promtail-local-config.yaml`).

Konfigurace v docker-compose.yml

V tomto konfiguračním souboru nejprve definujeme, nejlépe na začátku (od 1. řádku), síť s názvem `loki`. Jelikož doposud nebyla využívána v přidělené instanci Nextcloudu žádná síť, je nutné definovat i tento pojem.

Výpis 5.5: Definice nové sítě s názvem loki

```
networks:
  loki:
```

Dále přidáme pod sekci `services` parametry pro vytvoření Docker kontejneru, viz obrázek č.5.3.

První řádek nám udává název kontejneru.

Na druhém řádku je definován Docker image (obraz), který je uložen na Docker Hubu, a který slouží jako předloha pro kontejner. Jedná se o vlastní službu Promtail.

```yaml
promtail:
  image: grafana/promtail:2.9.2
  volumes:
    - nextcloud_server:/var/log/nextcloud/
    - ./apps/grafana/prom/promtail-local-config.yaml:/etc/promtail/config.yml
  command: -config.file=/etc/promtail/config.yml
  networks:
    - loki
```

Obr. 5.3: Definice pro vytvoření Promtail kontejneru

Třetí až pátý řádek definuje Docker Volumes, z toho čtvrtý slouží jako cesta k logům z Nextcloud serveru a pátý slouží k zapsání konfigurace definované před instalací Promtail do již běžící služby Promtail.

Šestý slouží pouze k informování služby Promtail kde hledat configurační soubor.

Poslední sedmý a osmý řádek přiřazují tento kontejner `promtail` do sítě `loki`, kterou jsme si definovali dříve.

Konfigurace v promtail-local-config.yaml

```yaml
server:
  http_listen_port: 0
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
- url: http://dect-nextcloud-docker-loki-1:3100/loki/api/v1/push

scrape_configs:
- job_name: nextcloud
  static_configs:
  - targets:
      - localhost
    labels:
      job: nextcloud
      __path__: /var/log/nextcloud/data/{nextcloud,audit}.log
```

Obr. 5.4: Konfigurační soubor pro Promtail

Na začátku je definován naslouchací port pro http a grpc (0 znamená náhodný port). Dále zde máme cestu k souboru, který uchovává pozici, kde Promtail skončil

a odkud má pokračovat, aby nedocházelo k duplikacím. Sekce `clients` konfiguruje, jak bude Promtail připojen k instanci Loki. Poslední sekce `scrape_configs` definuje, jak budou získávány logy, konkrétně zde máme název, dále cíl, kterým je localhost a potom dva štítky, jeden opět s názvem práce a druhý, který definuje cestu (soubor), ze kterého budou logy brány.

5.4 Loki

Hlavní server, který je zodpovědný za přijímání a ukládání logů, a zpracování dotazů.

Jedná se agregační systém logů inspirovaný Prometheusem, který je horizontálně škálovatelný a vysoce dostupný. Odlišnost Lokiho od Prometheuse tkví v tom, že se zaměřuje na logy namísto metrik a sbírá logy prostřednictvím HTTP API PUSH, místo PULL.

Loki je navržen pro nákladovou efektivnost a vysokou škálovatelnost. Na rozdíl od jiných logovacích systémů neindexuje obsah logů, ale pouze metadata jako sadu štítků pro každý tok logů. Tok (proud) logů je sada logů, které sdílejí stejné štítky. Tyto štítky pomáhají Lokimu najít proud logů v úložišti dat, z toho důvodu pro efektivní provádění dotazů nad logy je klíčem kvalitní sada štítků.

Dosažení efektivity

Základem je, že data logů jsou ukládána ve vysoce komprimovaných blocích, tudíž zabírají méně místa. Dále jsou indexovány pouze sady štítků, v porovnání s jinými nástroji pro agregaci logů je tedy i index Lokiho menší. Využívá také objektového úložiště jako jediného mechanizmu, díky čemu je dosaženo spolehlivosti a stability. To vše vede k efektivní činnosti a od toho odvozených nižších nákladů na provoz.

5.4.1 Konfigurace a instalace

Konfigurace byla inspirována oficiální dokumentací pro Loki [31] a také z webové stránky OKXO [32]. Opět je zde jeden soubor, který byl modifikován a jeden nově vytvořený.

Konfigurace v docker-compose.yml

Stejně jako u Promtail se jedná o modifikaci již existujícího souboru. Slouží k vytvoření kontejneru s názvem `loki` a obsahuje definici Docker obrazu pro poslední verzi Loki, navíc je zde definováno, kdy se může služba restartovat a dále ip adresa včetně portu, na kterém služba naslouchá. Dále zde máme cestu ke konfiguračnímu

souboru, zařazení do sítě `loki` a volumes, které mapují cesty k souborům v Dockeru k cestám uvnitř služby Loki. Tuto konfiguraci pohromadě lze vidět na obr. 5.5.

```yaml
loki:
  image: grafana/loki:latest
  restart: always
  ports:
    - "127.0.0.1:7946:7946"
  command: -config.file=/etc/loki/loki-local-config.yaml
  networks:
    - loki
  volumes:
  - ./apps/grafana/loki/loki-local-config.yaml:/etc/loki/loki-local-config.yaml
  - ./apps/grafana/loki/wal:/tmp/wal
  - ./apps/grafana/loki/rules:/tmp/rules
  - ./apps/grafana/loki/index:/tmp/index
  - ./apps/grafana/loki/chunks:/tmp/chunks
```

Obr. 5.5: Definice pro vytvoření Loki kontejneru

Konfigurace v loki-local-config.yaml

Konfiguraci lze v celku vidět na obrázcích 5.6 a 5.7. Na začátku je vypnuta služba autentizace, protože Loki v základu žádnou nemá a musí se dodatečně implementovat reverse proxy. Dále je definován http naslouchací port a informace ke zdroji, jako ip adresa a trvání chunků (jednotlivých kusů). Následuje `schema_config`, které určuje jakým způsobem jsou ukládána data. Je zde definováno datum, od kdy bude tato konfigurace použita, typ ukládání (`tsdb`), způsob ukládání (`filesystem`), verze schématu a data k indexu (prefix a perioda).

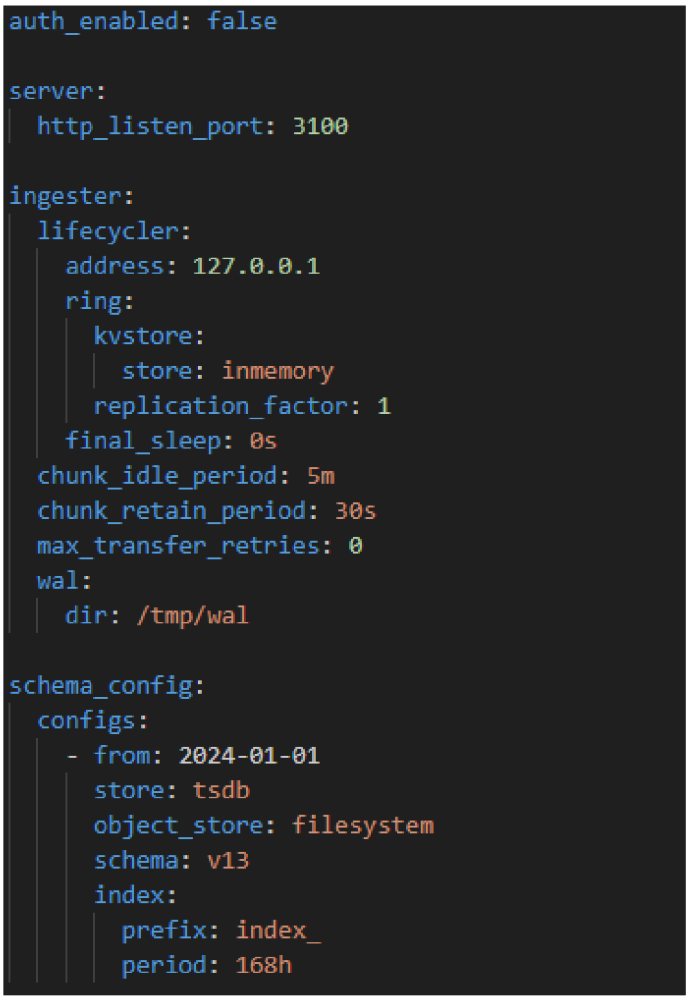
V druhé části konfiguračního souboru (obr. 5.7) lze vidět cesty k složkám pro tsdb a filesystem, dále sekci definující cesty k pravidlům a předpřípravu pro možné budoucí přidání alertmanageru, který upozorňuje na vybrané události pomocí emailu. Konfigurační soubor pak končí definicí limitů, aby nedocházelo k přehlcení systému, jako například maximální velikost proudu 500Mb (`per_stream_rate_limit: 500M`).

Posledním krokem pro správnou funkčnost služby Loki je udělení povolení přístupu k souboru pro loki. Provádí se příkazem 5.6.

Výpis 5.6: Příkaz pro udělení povolení k souboru pro loki

```
chown -R 10001:10001 loki
```

Lokace tohoto souboru (`loki-local-config.yaml`) lze vidět v sekci 5.1.



Obr. 5.6: Konfigurační soubor pro Loki, část 1.

5.5 Grafana

Slouží jako frontend, tedy uživatelské prostředí, pro Loki. Jedná se o open-source software, který umožňuje dotazování, vizualizaci, upozorňování a zkoumání metrik, logů a trasování, nezávisle na jejich uložení.

Vytvoření kontejneru se opět provádí podobně jako u Promtail a Loki (v souboru `docker-compose.yml`), a je zobrazeno na obrázku 5.8. Obsahuje import nejnovějšího Docker obrazu z Docker Hubu, porty, přes které komunikuje, definici, kdy se restartovat, cestu k souboru s prostředím, namapování cesty k souboru s daty a jako poslední přidání do sítě `loki`.

Ve stejném souboru (`docker-compose.yml`) je ještě nutné přidat síť `loki`, tak jak je to u Promtail, Loki a Grafany, i ke kontejnerům Nextcloud (v přiřazené instanci Nextcloud to je `nextcloud_db` a `nextcloud_server`.

Nyní je celá služba včetně podslužeb Nextcloud, Promtail, Loki a Grafana při-

```yaml
storage_config:
  tsdb:
    directory: /tmp/index

  filesystem:
    directory: /tmp/chunks

ruler:
  storage:
    type: local
    local:
      directory: /tmp/rules
  rule_path: /tmp/rules/fake
  # alertmanager_url: http://alertmanager:9093
  ring:
    kvstore:
      store: inmemory
  enable_api: true

limits_config:
  enforce_metric_name: false
  reject_old_samples: false
  reject_old_samples_max_age: 168h
  max_cache_freshness_per_query: 10m
  split_queries_by_interval: 48h
  per_stream_rate_limit: 512M
  cardinality_limit: 200000
  ingestion_rate_mb: 1024
  ingestion_burst_size_mb: 1024
  max_entries_limit_per_query: 1000000
  max_label_value_length: 20480
  max_label_name_length: 10240
  max_label_names_per_series: 300
```

Obr. 5.7: Konfigurační soubor pro Loki, část 2.

pravena ke spuštění příkazem `docker compose up -d`.

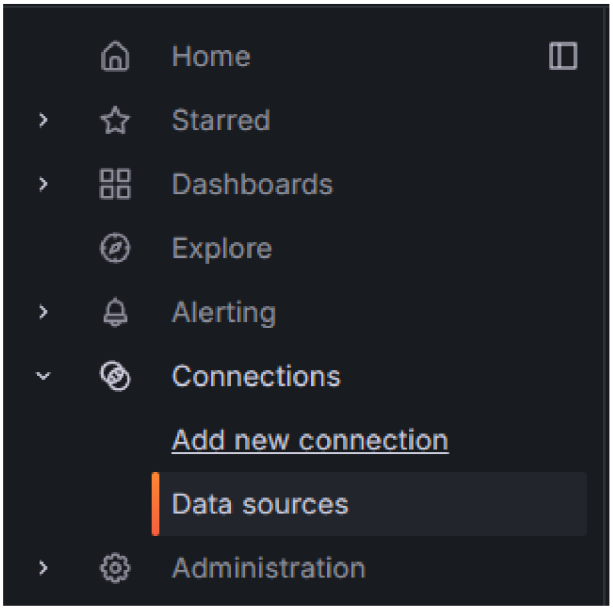
5.6 Nastavení webového prostředí

K webovému prostředí se přistupuje přes url `loclahost:3000`. Výchozími přihlašovacími údaji je admin/admin (jméno/heslo).

Po přihlášení v menu vlevo na stránce rozvineme *Connections* a klikneme na *Data sources*. Menu lze vidět na obr. 5.9. Na nově zobrazené stránce klikneme na *Add new data source* a vybereme v sekci *Logging & document databases* službu *Loki*.

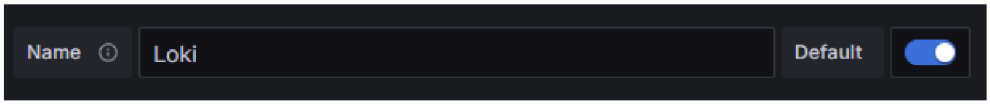
```yaml
grafana:
  image: grafana/grafana:latest
  ports:
    - 127.0.0.1:3000:3000
  restart: always
  env_file:
    - ./apps/grafana/.env
  volumes:
    - grafana-data:/var/lib/grafana
  networks:
    - loki
```

Obr. 5.8: Definice pro vytvoření Grafana kontejneru

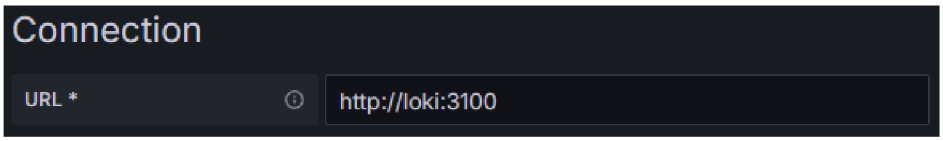


Obr. 5.9: Grafana menu ve webovém prostředí

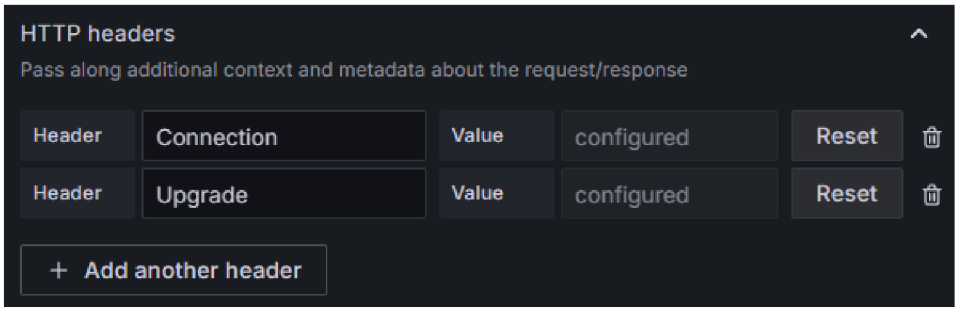
Tím otevřeme okno s nastavením propojení služby Loki. Provedeme nastavení jména, url adresy a HTTP hlaviček viz obrázky 5.10, 5.11 a 5.12. Potvrdíme tlačítkem *Save & test* na spodu stránky.



Obr. 5.10: Nastavení jména



Obr. 5.11: Nastavení url adresy



Obr. 5.12: Nastavení HTTP hlaviček

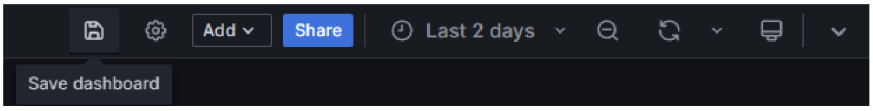
Dalším krokem je přidání dashboardu. Provádí se kliknutím na *Dashboards* v menu (obr. 5.9), následně kliknutím na tlačítko *New -> New dashboard*. Pro náš účel zpracovávání logů z Nextcloudu jsem vybral již předpřipravený dashboard od VoidQuark [33], tím odpadla nutnost vytvářet vlastní od nuly. Dashboard tedy importujeme na následující stránce (po kliknutí na *New dashboard*) kliknutím na *Import dashboard* a do nabízeného pole vepíšeme ID 17821. Toto ID bylo přiděleno stránkou Grafana Dashboards, která slouží jako oficiální úložiště a pro sdílení uživatelských dashboardů. Potvrdím importování tlačítkem *Load*.

Po zobrazení dashboardu zbývá provést konfiguraci štítků, která se provádí ve vrchní části obrazovky. Doplníme údaje viz obr. 5.13. *Datasource: Loki* udává službu, pomocí které jsou logy získávány, *Label name: filename* je název hlavního štítku, *Label Value: All* je výběr konkrétních logovacích souborů (na výběr z All, /var/log/nextcloud/data/admin.log, /var/log/nextcloud/data/nextcloud.log), *Job: nextcloud* je další pojmenování, tentokrát odpovídá štítku *job* v konfiguračním souboru Promtail a *Instance: All* je volitelný štítek pro případ, že bychom měli více Nextcloud instancí a chtěli bychom mezi nimi přepínat.

Obr. 5.13: Nastavení dashboard štítků

Nastavení uložíme v pravém horním rohu dashboardu přes ikonu pro ukládání, viz

obr. 5.14. Při ukládání zaškrtneme políčko *Save current variable values as dashboard default.*



Obr. 5.14: Řádek akcí pro dashboard

5.7 Dotazování se nad logy

Loki využívá na dotazování dotazovací jazyk LoqGL, který byl inspirován dotazovacím jazykem PromQL od Prometheus. Dotazy se chovají jako distribuovaný příkaz pro agregaci zdrojů. V tomto jazyce je využíváno filtrování štítky a operátory.

Dotazování nad logy se používají pro přehledné zobrazení v uživatelském (webovém) prostředí grafany.

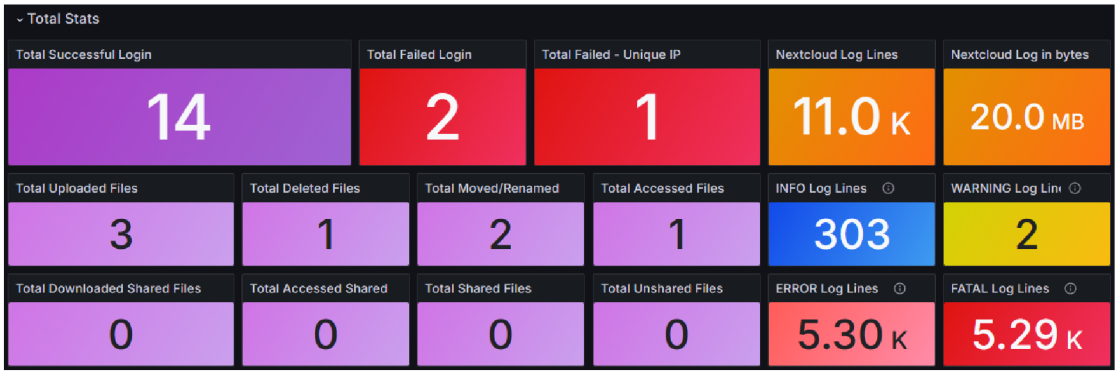
Existují dva typy dotazů:

- Log dotazy, které vracejí obsah řádků logů
- Metrické dotazy, které rozšiřují dotazy nad logy o výpočet hodnot na základě výsledku dotazu

5.8 Testování

Služba pro ukládání a správu logů byla otestována a je funkční. Obrázek 5.15 zobrazuje základní přehlede v uživatelském (webovém) prostředí služby Grafana. Pole, zobrazující počet logů, počet ERROR logů a počet FATAL logů zobrazuje vysoké počty z důvodu deaktivace externích Nextcloud aplikací nepotřebných k mému návrhu řešení. Byly odstraněny (zakomentovány) z důvodu snížení časové náročnosti sestavování Docker kontejnerů v testovacím prostředí.

Další obrázky (výřezy z obrazovky) zobrazující náhledy na uživatelské prostředí lze najít v Elektronické příloze k bakalářské práci [B].



Obr. 5.15: Zobrazení statistik v Grafaně

Závěr

Cílem této bakalářské práce byl návrh a realizace základních bezpečnostních opatření u vybrané instance Nexcloudu, konkrétně řešení DECT. Dále pak návrh řešení bezpečné správy a uchovávání logů.

V první kapitolou je úvod do cloud computingu, srovnání různých modelů cloudového nasazení a servisních modelů. Dále je zde věnována pozornost obecné bezpečnosti cloudových služeb, která je rozebrána a následně představeno obecné řešení těchto zranitelností.

Druhá kapitola je pak ze začátku věnována náhledu na Nextcloud a dále různým bezpečnostním prvkům, které se hodí pro řešení DECT. V podkapitole s bezpečnostními prvky je věnována pozornost jak prvkům integrovaným přímo v Nextcloudu, tak aplikacím, které se doinstalovávají. U těchto prvků je popsána jejich funkčnost a také ukázána integrace do konkrétní instance Nextcloudu.

Třetí kapitola je úvodem do Dockeru, včetně rozebrání klíčových komponent Dockeru a popsání jeho výhod. Tato kapitola byla zařazena do této práce z důvodu, že jak přidělená Nextcloud instance, tak vytvořené řešení logmanageru stojí na Docker řešení.

Ve čtvrté kapitole je rozebrána obecná problematika logování, včetně jeho využití, typů logů, představení původního řešení zpracování logů v přidělené instanci Nextcloud a kritérií pro návrh nového řešení.

V poslední páté kapitole je představen konkrétní návrh řešení správy a uchování logů založený na open-source softwaru od Grafana Labs. Je zde vysvětlena konfigurace všech komponent od generování logů v nextcloudu, přes sběr logů pomocí Promtail, zpracování pomocí Loki a uživatelského prostředí Grafana.

Literatura

[1] CHAI, WESLY. *Definition of cloud storage. Tect Target* [online]. 2021 [cit. 2023-11-11]. Dostupné z: <https://www.techtarget.com/searchstorage/definition/cloud-storage>.

[2] WHAT IS CLOUD COMPUTING? *Microsoft Azure* [online]. 2023 [cit. 2023-11-22]. Dostupné z: <https://azure.microsoft.com/en/resources/cloud-computing-dictionary/what-is-cloud-computing/#cloud-computing-models>.

[3] BUYYA R.;VECCHIOLA C.;SELVIS. T. *Mastering Cloud Computing: Foundations and Applications Programming* [online]. 1.USA: Elsevier, 2013 [cit. 2023-12-03] ISBN978-0-12-411454-8. Dostupné z: <https://ramslaw.files.wordpress.com/2016/07/0124114547cloud.pdf>.

[4] MELL, P.;GRANCE, T. *The NIST Definition of Cloud Computing. NIST* [online]. 2011 [cit. 2023-11-11]. Dostupné z: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>.

[5] WHAT IS A PRIVATE CLOUD? *Microsoft Azure* [online]. 2023 [cit. 2023-11-22]. Dostupné z: <https://azure.microsoft.com/en/resources/cloud-computing-dictionary/what-is-a-private-cloud/>.

[6] STÝBLO, KAREL. *Cloud. historie, definice, modely a praktické využití. Katedra informatiky* [online]. K2 atmitec, 2014 [cit. 2023-12-5]. Dostupné z: <https://katedrainformatiky.cz/Resources/Upload/Home/osobni/radecky/it/pr9.pdf>.

[7] BONIFACE, MICHAEL, a kol. *Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds Fifth International Conference on Internet and Web Applications and Services 2010* [online]. 2010 [cit. 2023-12-5]. s. 155–160. ISBN 978-1-4244-6729-7. Dostupné z: <https://ieeexplore.ieee.org/document/5476775>.

[8] SAAS ANEB PROČ ZVOLIT VÝVOJ WEBOVÉ APLIKACE V CLOUDU? *Rascasone* [online]. 2021 [cit. 2023-12-5]. Dostupné z: <https://www.rascasone.com/cs/blog/vyvoj-webovych-aplikaci-saas>.

[9] Co je SaaS? *Microsoft Azure* [online]. 2023 [cit. 2023-12-5]. Dostupné z: <https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-is-saas/>.

[10] KUMAR, R.; GOYAL, R. *On cloud security requirements, threats, vulnerabilities and countermeasures: A survey. Computer Science Review 33* [online]. 2019. 1-48 [cit. 2023-12-5]. ISSN 15740137 Dostupné z: <https://doi.org/10.1016/j.cosrev.2019.05.002>.

[11] JON-MICHAEL BROOK, a kol. *The Treacherous 12 - Cloud Computing Top Threats in 2016. CSA* [online]. 2016 [cit. 2023-12-5]. Dostupné z: <https://downloads.cloudsecurityalliance.org/assets/research/top-threats/Treacherous-12_Cloud-Computing_Top-Threats.pdf>.

[12] JON-MICHAEL BROOK, a kol. *Top Threats to Cloud Computing Pandemic Eleven. CSA* [online]. 2022 [cit. 2023-12-5]. Dostupné z: <https://cloudsecurityalliance.org/download/artifacts/top-threats-to-cloud-computing-pandemic-eleven/>.

[13] Nextcloud [online]. 2023 [cit. 2023-12-12]. Dostupné z: <https://nextcloud.com/>.

[14] Security and authentication *Nextcloud* [online]. 2023 [cit. 2023-12-12]. Dostupné z: <https://nextcloud.com/secure/>.

[15] Hardening and security guidance *Nextcloud docs* [online]. 2023 [cit. 2023-12-10]. Dostupné z: <https://docs.nextcloud.com/server/latest/admin_manual/installation/harden_server.html>.

[16] ClamAV *ClamAV Documentation* [online]. 2024 [cit. 2023-05-30]. Dostupné z: <https://docs.clamav.net/Introduction.html>.

[17] Brute force protection *Nextcloud docs* [online]. 2023 [cit. 2023-12-10]. Dostupné z: <https://docs.nextcloud.com/server/stable/admin_manual/configuration_server/bruteforce_configuration.html>.

[18] Antivirus scanner *Nextcloud docs* [online]. 2023 [cit. 2023-12-10]. Dostupné z: <https://docs.nextcloud.com/server/latest/admin_manual/configuration_server/antivirus_configuration.html>.

[19] Nextcloud Geoblocker App *HomeITAdmin - github* [online]. 2023 [cit. 2023-12-10]. Dostupné z: <https://github.com/HomeITAdmin/nextcloud_geoblocker/>.

[20] Ransomware Protection App *Nextcloud - github* [online]. 2022 [cit. 2023-12-10]. Dostupné z: <https://github.com/nextcloud/ransomware_protection>.

[21] Docker architecture *O'Reilly* [online]. 2024 [cit. 2024-05-22]. Dostupné z: <https://www.oreilly.com/library/view/learn-openshift/9781788992329/33d025bf-27fa-49b9-99b3-673a20ae6d1e.xhtml>.

[22] Hanwen Zhang, *An Overview of Virtual Machine v.s. Docker v.s. Kubernetes* [online]. 2023 [cit. 2024-05-22]. Dostupné z: <https://hanwenzhang123.medium.com/docker-vs-virtual-machine-vs-kubernetes-overview-389db7de7618>.

[23] Docker Docs *docker.docs* [online]. 2024 [cit. 2024-05-28]. Dostupné z: <https://docs.docker.com/>.

[24] Volumes *docker.docs* [online]. 2024 [cit. 2024-05-28]. Dostupné z: <https://docs.docker.com/storage/volumes/>.

[25] Ephraim Norbert, is Log Analysis: Importance and use Cases [online]. 2024 [cit. 2024-05-30]. Dostupné z: <https://www.linkedin.com/pulse/what-log-analysis-importance-use-cases-ephraim-norbert-14kpe/>.

[26] Arfan Sharif. *LOG FILES EXPLAINED* [online]. 2022 [cit. 2024-05-22]. Dostupné z: <https://www.crowdstrike.com/cybersecurity-101/observability/log-file/>.

[27] Sharath Kumar. *TYPES OF LOGS* [online]. 2024 [cit. 2024-05-22]. Dostupné z: <https://sharath-kumar.medium.com/types-of-logs-5cc6cdb40482>.

[28] Logging *Nextcloud docs* [online]. 2024 [cit. 2024-05-16]. Dostupné z: <https://docs.nextcloud.com/server/latest/admin_manual/configuration_server/logging_configuration.html>.

[29] Výhody a nevýhody self-hosted řešení *MyDreams* [online]. 2024 [cit. 2024-05-16]. Dostupné z: <https://www.mydreams.cz/cz/wiki/7253-vyhody-a-nevyhody-self-hosted-reseni.html>.

[30] Promtail agent *Grafana docs* [online]. 2024 [cit. 2024-05-26]. Dostupné z: <https://grafana.com/docs/loki/latest/send-data/promtail/>.

[31] Loki Configuration *Grafana docs* [online]. 2024 [cit. 2024-05-28]. Dostupné z: <https://grafana.com/docs/loki/latest/configure/examples/configuration-examples/>.

[32] Okko. *Monitor your Nextcloud logs for suspicious activities with Grafana Loki* [online]. 2023 [cit. 2024-05-28]. Dostupné z: <https://okxo.de/monitor-your-nextcloud-logs-for-suspicious-activities/>.

[33] Parsing Nextcloud Audit Logs with Grafana Loki *VoidQuark* [online]. 2023 [cit. 2024-05-28]. Dostupné z: <https://voidquark.com/blog/parsing-nextcloud-audit-logs-with-grafana-loki/#nextcloud-logging-configuration>.

Seznam symbolů a zkratek

SLA	Service-level agreement
CPU	Central processing unit
RAM	Random access memory
HW	Hardware
IaaS	Ifrastructure as a Servise
PaaS	Platform as a Servise
SaaS	Software as a Servise
CSA	Cloud security alliance
API	Application Programming Interface
PKI	Public Key Infrastructure
WAF	Web application firewall
DoS	Denial of Service
DDoS	Distributed Denial of Service
IoT	Internet of Things
IPS	Intrusion Prevention System
IDS	Intrusion Detection System
PDF	Portable Document Format
RIR	Regional Internet Registries
JSON	JavaScript Object Notation

A Návod na instalaci a obsluhu logmanageru

Jedná se o stručný návod na instalaci a obsluhu logmanageru Grafana. Tento návod je přizpůsoben přidělené instanci Nextcloud server využívající Docker (compose).

A.1 Instalace

- Prvním krokem je vytvoření adresáře a souborů, v již existující složce `apps`, viz strom níže. Soubory necháme zpočátku prázdné.

```
apps ..... již existující složka, která sdružuje přidané aplikace
└── grafana ..... složka pro grafana aplikaci
    ├── loki. ..... složka pro službu Loki
    │   ├── chunks. ..... složka
    │   ├── index. ..... složka
    │   ├── rules. ..... složka
    │   ├── wal ..... složka
    │   └── loki-local-config.yaml. ..... konfigurační soubor pro Loki
    ├── prom. ..... složka pro službu Promtail
    │   └── promtail-local-config.yaml ..... konfigurační soubor pro Promtail
    └── .env. ..... soubor s prostředím (ponechán prázdný)
```

- Nahrajeme obsah (můžeme zvolit i možnost přepsat soubor) z elektronické přílohy do souborů `loki-local-config.yaml` a `promtail-local-config.yaml` (soubory se v elektronické příloze jmenují stejně). Slouží jako konfigurační soubory pro Loki a Promtail
- Překopírujeme modifikovaný obsah ze souboru `docker-compose.yml` z elektronické přílohy do stejnojmenného souboru v root adresáři DECT-NEXTCLOUD-DOCKER. Jedná se o obsah řádků 1, 2, 9, 10, 13-47, 73, 74, 115 a 116. Tyto příkazy vytváří kontejnery Grafana, Loki a Promtail, a dále i s kontejnery nextcloud_server a nextcloud_db je zařadí do sítě `loki`
- Překopírujeme řádek 6-13 ze souboru `nextcloud-config.sh` z elektronické přílohy do stejnojmenného souboru (cesta k souboru je `./scripts/post-installation/`). Tyto řádky slouží ke spuštění a konfiguraci generování logů Nextcloudem
- Přidělíme službě loki oprávnění k vytvořené složce loki příkazem zadaným do konzole: `chown -R 10001:10001 loki`
- Spustíme všechny služby příkazem `docker build` a následně `docker compose up -d`

A.1.1 Webové prostředí

- Zadáme do url řádku ve webovém prohlížeči adresu `localhost:3000`, na přihlašovací stránce se přihlásíme výchozím jménem a heslem `admin/admin`
- V levém menu rozrolujeme nabídku `Collections` a zvolíme `Data sources`
- Klikneme na `Add new data source`
- Vybereme službu Loki
- Editujeme položky následovně: `Name: Loki`, `URL*: http://loki:3100`, v sekci HTTP headers `Header: Connection` a `Header: Upgrade`, a potvrdíme nastavení ve spodní části stránky tlačítkem `Save & test`
- V levém menu zvolíme `Dashboards`, klikneme na `New -> New dashboard -> Import dashboard` a do příslušné kolonky vyplníme ID `1782`. Potvrdíme tlačítkem `Load`
- Na vrchu stránky doplníme údaje o štítcích: `Datasource: Loki`, `Label Name: filename`, `Label Value: All`, `Job: nextcloud` a `Instance: All`
- V pravo nahoře uložíme pomocí ikony na uložení, ve vyskočeném okně zaškrtneme (označíme) položku `Save current variable values as dashboard default` a potvrdíme tlačítkem `Save`

A.2 Obsluha

Obsluha je velice jednoduchá. Na stránce `Dashboards` vidíme základní přehled s počtem událostí. Dále dva koláčové grafy, jeden pro úspěšné přihlášení a druhý pro neúspěšné přihlášení. Poslední věcí je přehled nedávných logů, kdy kliknutím na jednotlivé logy si můžeme zobrazit podrobnosti.

Pro další přehledy si musíme rozbalit jednotlivé lišty, které chceme vidět, na spodní části stránky.

U jakéhokoliv přehledu můžeme pomoci stiknutí tří teček (po najetí myší na přehled) rozbalit kartu s činnostmi, jako prozkoumání daných logů patřících k přehledu, zobrazení na celou obrazovku, editaci dotazování nad logy, atd.

Obsluha tvoření dotazů nad logy již není součástí tohoto návodu z důvodu její vyšší složitosti, a je doporučené obsáhlejší proškolení.

B Obsah elektronické přílohy

Elektronická příloha obsahuje upravené a nové zdrojové soubory pro službu logmanagera založeného na softwaru od Grafana Labs. Dále je v těchto souborech také konfigurace bezpečnostních prvků viz kapitola 2.2.

Mezi další soubory elektronické přílohy patří výřezy obrazovky (obrázky) pro náhled do uživatelského prostředí.

Je zde obsažena také digitální kopie této bakalářské práce.

```
David_Zeman_BP_priloha ..... název přílohy práce
├── screenshots_grafana ..... výřezy obrazovky pro náhled uživatelského prostředí
│   ├── dalsi_zobrazeni
│   ├── editace_dotazovani
│   ├── grafy_prihlaseni_recent_logs
│   ├── nahrani_souboru
│   ├── neuspesne_prihlaseni
│   ├── podrobnosti_o_logu
│   ├── prejmenovane_presunute_soubory
│   ├── rozbalene_menu
│   ├── smazani_souboru
│   ├── stitky_grafana_dashboard
│   ├── uspesne_prihlaseni
│   ├── zakladni_prehled
│   └── zobrazene_soubory
├── zdroj ..... zdrojové soubory modifikované/nové
│   ├── docker-compose.yml
│   ├── loki-local-config.yaml
│   ├── nextcloud-config.sh
│   └── promtail-local-config.yaml
└── BP_Zeman_231304.pdf
```