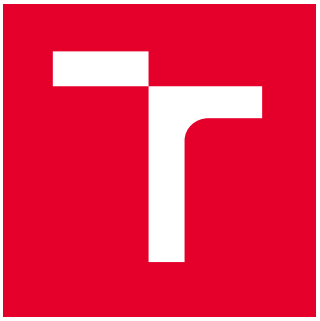


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

CLOUDOVÝ DOHLEDOVÝ SYSTÉM PRO IT INFRASTRUKTURY

CLOUD MONITORING SYSTEM FOR IT INFRASTRUCTURES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Bc. Daniel Kunčický**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Marek Sikora**

SUPERVISOR

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Daniel Kunčický ***ID:*** 230612

Ročník: 2 ***Akademický rok:*** 2024/25

NÁZEV TÉMATU:

Cloudový dohledový systém pro IT infrastruktury

POKYNY PRO VYPRACOVÁNÍ:

Cílem této diplomové práce je inovovat existující systém pro monitoring síťových zařízení na základě automatizovaného vyhodnocování emailových notifikací. Prvním úkolem práce bude oprava implementačních chyb a chyb návrhu existujícího systému, inovace grafického rozhraní a implementace nových funkcí (mj. přidání souhrnné statistiky, zpřehlednění oznámení o problémech, analýza notifikací na základě obsahu, systém výjimek detekce problémů atd.). Dalším úkolem bude přidání podpory protokolu IMAP za účelem zvýšení univerzálnosti a užitné hodnoty tohoto řešení. Navazujícím úkolem bude analýza cloudových řešení a převod celého systému na vhodnou cloudovou platformu. Posledním úkolem bude vytvoření instalačního skriptu, či obrazu disku, pomocí kterého bude možné jedním krokem celý systém nainstalovat a uvést do provozu.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce a konzultanta.

Termín zadání: 10.2.2025 ***Termín odevzdání:*** 27.5.2025

Vedoucí práce: Ing. Marek Sikora
Konzultant: Ing. Jan Sikora

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

ABSTRAKT

Tato práce se zabývá úpravou dohledového systému, který automatizovaně vyhodnocuje e-mailové notifikace od monitorovaných zařízení. O výsledku vyhodnocení informuje správce formou zprávy. Cílem práce je přidat nové funkce a následně celý systém převést na cloudovou platformu. Pro zvýšení užitné hodnoty a odstranění závislosti na jednom dodavateli e-mail serveru byla přidána podpora pro protokol IMAP.

KLÍČOVÁ SLOVA

cloudový dohledový systém, vyhodnocení dle obsahu, instalační skript, přehled problémů, grafické rozhraní, vyhodnocení dochvilnosti

ABSTRACT

This thesis focuses on modifying a monitoring system that automatically evaluates e-mail notifications received from monitored devices. Administrator is informed about the evaluation result by message. The aim of the work is to add new features and then migrate the whole system to a cloud platform. To increase functional value and eliminate dependency on one e-mail server provider, support for IMAP protocol has been added.

KEYWORDS

cloud monitoring system, content-based evaluation, installation script, problems summary, graphical interface, timeliness evaluation

KUNČICKÝ, Daniel. *Cloudový dohledový systém pro IT infrastruktury*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2025. Vedoucí práce: Ing. Marek Sikora

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Daniel Kunčický
VUT ID autora:	230612
Typ práce:	Diplomová práce
Akademický rok:	2024/25
Téma závěrečné práce:	Cloudový dohledový systém pro IT infrastruktury

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 diplomové práce panu Ing. Marku Sikorovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Úvod

Práce se zabývá inovací existujícího dohledového systému a implementací nových funkcionalit. Dohledový systém vyhodnocuje přijaté e-mailové zprávy, které zasílají monitorovaná zařízení. Zprávy označí příslušnými štítky dle vzoru a přesune je do archivní lokace. Nezpracované zprávy představují problémy, které správce řeší ručně. Obsluha aplikace probíhá pomocí grafického rozhraní, které nabízí kompletní správu aplikace. Například tvorbu a úpravu plánovaného spouštění.

Cílem práce je přidat nové funkce a to tak, aby zůstala zachována původní funkčnost. Nejprve se musí opravit chyby v návrhu. Notifikace pro správce budou zpřehledněny přidáním systému šablon. Způsob vyhodnocení zpráv bude upraven. Přidá se detekce výjimek a vyhodnocení zprávy na základě obsahu. Pro zvýšení univerzálnosti aplikace bude implementován protokol IMAP.

První kapitola popisuje koncept dohledových systémů a jejich využití. Co a proč se monitoruje a jak sběr dat technicky zajistit. Představuje dohledový systém mail-notifier, který je předmětem této práce.

Druhá kapitola se zaměřuje na vyhodnocení počátečních požadavků. Tedy předem známých faktů, které je třeba brát v potaz při návrhu nových funkcí. Diskutuje možná bezpečností rizika, která vznikají při použití externích knihoven programovacího jazyka Python. Nabízí pohled na mitigaci rizik. Dále řeší udržitelnost aplikace.

Třetí kapitola se zabývá výběrem vhodné cloudové platformy. Zjišťuje parametry jednotlivých platforem. Analyzuje způsoby provozu vlastních aplikací v cloudu. Dále řeší nákladovou stránku. S tím souvisí výběr vhodného tarifu.

Čtvrtá kapitola analyzuje architektura stávající aplikace a mapuje všechny známé nedostatky původního řešení, které vyplynuly z praxe. Hlavní moduly aplikace jsou podrobněji popsány a jejich funkce graficky vysvětlena.

Pátá kapitola se věnuje návrhu úprav jednotlivých modulů. Ty jsou navrženy na základě analýzy, která proběhla v kapitole čtvrté. Řeší zařazení nových funkcí do určených modulů aplikace. Popisuje potřebné změny ve struktuře modulů.

Šestá kapitola řeší implementaci inovací v rámci serverové části aplikace. Detailněji se zaměřuje na jednotlivé moduly a jejich funkci. Představuje nový šablonový systém notifikací. S tím souvisí nový koncept konfiguračních souborů.

Sedmá kapitola se věnuje realizaci grafického rozhraní. U existujících nástrojů bylo nutné provést úpravy, aby zůstala zachována funkčnost. Popisuje vytvoření nových funkcí. Například přehled problémů, který zobrazuje stav sledovaných složek.

Poslední kapitola řeší zprovoznění systému. Nabízí využití vlastního serveru nebo spuštění na cloudové platformě. Pro automatickou instalaci byl vytvořen skript, který na jedno kliknutí (spuštění) zajistí instalaci celého dohledového systému.

1 Dohledové systémy

Tato kapitola stručně pojednává o dohledových systémech a jejich vlastnostech. K čemu a proč se vůbec používají je také součástí. Poslední část představuje dohledový systém, který je součástí práce (bude inovován).

1.1 Využití dohledového systému

Dohledové systémy se stávají nezbytnou součástí vybavení IT oddělení. Počet zařízení připojených k síti se s moderní dobou razantně navyšuje dokonce i v domácnostech. Zařízení mohou zasílat notifikace o svém stavu. Ve větších firmách je není možné uhlídat ručně. Navíc je to nákladné [1]. Od domácích uživatelů zase nelze očekávat, že budou trávit čas vyhodnocováním došlých notifikací. Už nyní je notifikací všude obrovské množství. A stále to narůstá a narůstat bude. To pak vede k situaci, kdy jsou přehlíženy nahlášené problémy. Hlášené problémy mají určitou závažnost. Je nutné tuto závažnost také vyhodnocovat a podle nastavených kritérií provést připravenou akci a reakci. Nízká závažnost bude zřejmě značit problém, který není nutné bezprostředně řešit. Naopak problém s vysokou závažností bude pravděpodobně nutné řešit neprodleně. Příkladem může být informace o docházejícím místě na síťovém úložišti. Této informaci je možné přiřadit nízkou závažnost. Pouze upozorňuje, že bude nutné rozšířit diskové pole, aby v budoucnosti nedošlo k totálnímu zaplnění. Naopak za vysokou závažnost lze považovat upozornění na kritickou aktualizaci operačního systému, která má opravit útočníky zneužívanou bezpečností chybu [2]. Systém bude nezbytné aktualizovat ideálně ihned, aby se zabránilo potencionálnímu průniku útočníka a napadení zařízení. Pokud by reakce nebyla rychlá, může dojít ke škodám ať už na zařízení nebo uložených datech.

1.2 Komerční řešení

Komerční řešení mohou obsahovat mnoho funkcí, které ale nebudou nikdy využity. Tím narůstá náročnost správy takového řešení. Studování mnohostránkového návodu není úplně jednoduché. Ve výsledku se tak může stát, že by pokročilé nastavení takového software bylo velmi nákladné. Nikdo zřejmě není dlouhodobě schopen pojmout takové množství informací. Instalace vlastními silami pak představuje riziko. Pokud odejde pracovník, který to zprovoznil a nastavil, pak se nemusí najít nikdo, kdo tomu ještě rozumí. Případná oprava problému, který vznikne je náročná. A externí firmy si zase účtují vysoké poplatky. Pro menší firmu to není nákladově únosné.

Zabezpečení takových produktů je také otázkou. Jedná se o velmi komplexní produkty s mnoha funkcemi. Instalace agenta na každé zařízení představuje potenciální riziko. Když agent obsahuje chybu, je zneužitelná po celé interní síti [3]. Stále častěji se prosazuje pouze cloudový provoz s měsíčním poplatkem. Najít produkt, který se zakoupí pouze jednorázově začíná být složité Nevýhodou cloudového přístupu je výpadek spojení do internetu. V takovém případě agent sice stále funguje, ale nemá kam zaslat report. Není jasné, jestli obsahuje nějakou mezipaměť, kam si uloží výsledky k pozdějšímu odeslání.

Naopak provoz na vlastním serveru představuje náklady na pořízení serveru a jeho následnou správu. Není ovlivněn výpadkem připojení k internetu. Licenční politika však bývá podobná, jako u cloudového řešení. Jsou poskytovány aktualizace software a podpora ze měsíční poplatek [4].

Další problém představuje výběr produktu. Webové stránky dodavatelů obsahují spoustu prohlášení, ale získat zkušební verzi bez kontaktování obchodního oddělení je téměř nemožné. Je nutné zadat kontaktní údaje. Obchodní oddělení pak obtěžuje se svými nabídkami.

1.3 Co se monitoruje

Data která lze z monitorovaných zařízení získat jsou různého charakteru. Lze sbírat vytížení systémových prostředků. Tedy procesoru, operační pamětí, zatížení sítě a další. Nasbíraná data se vyhodnocují a vyhledávají se odchylky od normálu [5]. Pokud bezdůvodně vzroste vytížení procesoru, může to znamenat útok. Také lze identifikovat zařízení, které již svými parametry nepostačuje nárokům v důsledku rozvoje sítě. Problém může způsobit i chybná konfigurace. Pokud po její aktualizaci nebude dostupná část sítě, dá se předpokládat viník.

Monitorují se i různé stavové informace. Třeba ohledně zálohování dat. To se provádí v předem nastavený čas. Pokud notifikace nedorazí, znamená to problém.

1.4 Techniky sběru dat

Aby bylo co monitorovat, musí se nejdříve nějakým způsobem získat data. Technicky je třeba zajistit sběr dat ze zařízení různých výrobců, které nejsou navzájem kompatibilní. Zařízení obvykle obsahují nějakou funkci, která zasílá status zařízení [6]. Ale nelze přidat vlastní preferovaný způsob. Musí se využít nabízené varianty.

1.4.1 Emailové zprávy

Zařízení v pravidelných intervalech zasílá e-mail, který obsahuje informaci o stavu. Ten tak lze zobrazit v běžném e-mail klientu bez dalšího software. Notifikace jsou v čitelném formátu, většinou neobsahují technický detail [7].

1.4.2 SNMP

Protokol SNMP (Simple Network Management Protocol) se používá k monitorování síťových zařízení. Je standardizován a tak ho podporují různé druhy zařízení [8]. Celkem existují tři verze, nejnovější je SNMPv3. Starší verze neřeší zabezpečení dat, provoz není šifrován a dá se odposlechnout. S tím je nutné při návrhu sítě počítat [9]. Na lokální síti to nemusí představovat problém.

1.4.3 Agent

Na zařízení se naistaluje software (agent), který provádí monitorování. Sbírá data, která poté zasílá na centrální server. V případě výskytu problému může být instruován, aby provedl akci. Třeba vypnul zaseknutý proces. Může obsahovat mezipaměť, kam zapisuje data v případě výpadku spojení serverem. Po obnovení spojení data zašle. Velikost paměti je omezena, výpadek nesmí být příliš dlouhý.

Agent musí být kompatibilní s provozovaným zařízením, nelze jej použít všude. Není standardizován, takže není univerzální. Navíc zatěžuje systém, protože je neustále spuštěn. Musí se provádět jeho aktualizace [6].

1.4.4 NetFlow

NetFlow je protokol vyvinutý firmou Cisco. Používá se k monitorování síťového provozu. Díky němu lze zjistit zatížení sítě. Zařízení sbírá provozní data, která agreguje a poté zašle na server. Tam se data vyhodnotí a případně zobrazí. Záleží na zvoleném výrobci. Systémů s podporou NetFlow je více [10].

1.5 Mailnotifier v1.5

Mailnotifier je dohledový systém, který byl vytvořen v rámci bakalářské práce [11]. Periodicky provádí automatické vyhodnocení e-mailových zpráv, které přicházejí od monitorovaných zařízení. Po spuštění se přihlásí do e-mailové schránky. Nejprve vyhledá vzorové zprávy, které jsou označeny štítkem „_TEMPLATE“. Jakmile dokončí vyhledání vzorů, přejde na vyhodnocení nepřečtených zpráv. Zprávy načte a zkouší je přiřadit k jednomu ze vzorů. Porovnává se shoda odesílatele a předmětu

přijaté zprávy se vzorem. Pokud byla nalezena shoda, načte ze vzoru štítek dochvilnosti, například „10-minutly“. Tento štítek pak na hodnocenou zprávu nastaví.

Poté se vyhodnotí dochvilnost zprávy, jestli přišla v čas. Pokud byla zpožděna, oznámí to prostřednictvím notifikačního e-mailu správci. Po vyhodnocení dochvilnosti přesune označené zprávy do archivní lokace „_název složky“. Výsledkem je, že ve sledované složce zůstane pouze vzor a nejnovější přijatá zpráva. Ostatní jsou označeny jako přečtené a přesunuty.

Pokud ve složce zůstane zpráva, která neodpovídá žádnému vzoru, zůstane nepřečtená. Při pohledu na počet nepřečtených zpráv ve složce je možné ihned identifikovat, jestli přišla nějaká chybová zpráva. A to bez využití externích nástrojů. V ideálním případě bude v monitorované složce nula nepřečtených zpráv.

1.5.1 Obsluha systému

Mailnotifier je rozdělen na dva celky. První se stará o vyhodnocování a má pouze konzolový výstup. Vypisuje do něj stav zpracování. Uživatel se s ním běžně nesetká, protože při automatickém spouštění není zobrazen.

Druhá část je grafické rozhraní, přes které probíhá správa systému. Lze přes něj vytvářet a editovat úlohy. Dále provést kompletní konfiguraci. Obsahuje také funkci pro vyhledání a export zájmových zpráv do textového souboru.

1.5.2 Systémové informace

Systém byl vytvořen v jazyce Python 3.7 a je distribuován ve dvou formátech. První formát jsou soubory „.exe“, které jsou určeny k provozu na OS Microsoft Windows bez další instalace. Druhý formát jsou „.py“ soubory, které jsou určeny pro OS Linux. Vyžadují nainstalované externí knihovny. Zprovoznění je tak složitější.

Aplikace podporuje připojení výhradně k e-mailovému serveru Exchange Online od společnosti Microsoft. Přístup k serveru je proveden přes API MS Graph. Jiné servery nejsou podporovány.

2 Vyhodnocení počátečních požadavků

Před začátkem prací na inovaci dohledového systému je nezbytné vyhodnotit počáteční požadavky. Ty vznikají nejen na základě zadání, ale i podle reálných zkušeností z provozu. Teprve po jejich vyhodnocení bude možné pokračovat s návrhem řešení a implementací. Jinak by mohlo dojít k vytvoření nefunkčního celku.

2.1 Bezpečnost použitých knihoven

Řešený dohledový systém (viz kapitola 1.5) využívá programovací jazyk Python 3 (viz kapitola 1.5.2) a proto bude využíván i nadále. V rámci implementace je použita řada externích knihoven. Jejich využití snižuje časovou náročnost na vývoj již vyvinutých funkcí. Místo opětovného vývoje se použijí funkce z vhodné knihovny.

Stahují se z veřejného repozitáře PyPI a to představuje určitou míru rizika. Pokud útočník pozmění kód takové knihovny, nemusí se na to dlouhou dobu přijít. Repozitář PyPI se snaží zabraňovat takové kompromitaci, ale v tom velkém množství se snadno něco přehlédne [12]. Navíc sofistikovanost útočníků stále narůstá. Některé kampaně jsou velmi obtížně detekovatelné.

2.1.1 Známé způsoby útoků

V minulosti bylo zaznamenáno mnoho pokusů o kompromitaci programátorů (jejich pracovních stanic) využívajících nějaké externí knihovny. Známých způsobů útoků existuje hned několik a další přibývají.

Některé útoky spoléhají na chybu uživatele. Útočník vytvoří knihovnu s podobným názvem jako má originál. Překlepové názvy jsou tímto velmi nebezpečné. Nemusí jít pouze o překlepy. Útočník vezme původní název a přidá k němu nějakou příponu. Pokud byl původní název KeyTool, tak útočník zvolí obdobný název, třeba KeyTool-sdk. Některé knihovny mohou používat v názvu číslo verze. Útočník tak pouze číslo zvedne o jedna (KeyTool1/KeyTool2) [13]. Pojmenování knihoven není nijak unifikováno. A tím vznikají tyto problémy, na které není řešení.

Další útok spočívá v napadení publikujícího účtu vývojáře, který se stará o vývoj a údržbu své knihovny. To se stává hlavně u účtů, které delší dobu nikdo nepoužil. Velmit tomu napomáhá chybějící multifaktorová autentizace [14].

Dokonce je nutné počítat se záškodnickými úmysly autora. Ten z počátku vyvíjí zcela nezávadné řešení. Ale z důvodu složité životní situace a špatného psychického stavu vydá závadnou aktualizaci. Poslední kategorií jsou knihovny vydané se špatnými úmysly od počátku. Obvykle je cílem odcizení přístupových tokenů ke cloud službám. K těm útočník získá neautorizovaný přístup [15].

Dalším zdrojem útoků jsou na první pohled užitečné blogové stránky a články, které zcela náhodou nabízí odpověď na hledaný problém. Jako přílohu nabídnou ke stažení zaheslovaný soubor, který obsahuje škodlivý kód. Praktika zaheslování se využívá k vyřazení antivirového software, který se nachází na platformě hostující soubor. Nelze jej jednoduše zkontrolovat. Ztěžuje se tím jeho detekce a včasné odstranění. Taková praxe byla sledována i na serverech s převážně multimediálním obsahem. Ve videu je opět k nalezení vyobrazené řešení hledaného problému [16].

2.1.2 Následky útoků

Následky útoků jsou závažné. Záleží na popularitě knihovny a počtu uživatelů, kteří si závadnou knihovnu/aktualizaci stáhnou. Velké nebezpečí představuje takzvaný supply chain attack [17]. Tedy napadení dodavatele/autora knihovny. Ten chce dodávat nezávadný produkt. Avšak také využije dalších knihoven jako externí závislost. Jakmile dojde k napadení byť jediné závislosti v řetězci, jsou automaticky napadeni všichni uživatelé dodavatele, jestliže aktualizují na novou závadnou verzi.

Následkem jsou odcizené přístupové údaje, které má uživatel uložené. Také dochází k instalaci sledovacího software, která průběžně zasílá na servery útočníka citlivé informace. Nelze vyloučit ani destruktivní útoky.

2.1.3 Obrana

Žádná stoprocentní ochrana neexistuje. Jediným řešením je nepoužívat žádné knihovny. To ale stále nevylučuje jiné způsoby útoku. Obrana se tak musí zaměřit na snížení rizika do akceptovatelné úrovně. Obecně se lze držet doporučení využívat známé a prověřené knihovny. Riziko útoku sice stále nebude plně eliminováno, ale je sníženo na přijatelnou úroveň. Naopak nelze doporučit instalaci knihoven bez reputace.
Ani z počátku výborná reputace však není zárukou bezpečnosti.

2.2 Udržitelnost aplikace

Aplikace by měla být navržena s určitým ohledem na dlouhodobější udržitelnost. Tím je myšleno, že se obejde bez zásadní údržby a vydrží v provozu po mnoho let. Vyměňovat dohledové systémy totiž není jednoduchá ani levná záležitost. Dalším vývojem technologií může dojít k postupné nedostupnosti některých funkcí. Jedním z řady důvodů mohou být změny ve využívaném API. Může být vydána nová verze. Ta po určité době zcela nahradí původní verzi API a ve výsledku aplikace přestane zcela fungovat. Tento problém má efektivně řešit přidání podpory pro protokol IMAP. Ten je podporován mnoha servery a pravděpodobnost jeho budoucí

nedostupnosti je nižší než u API. Pokud by přece jenom nastala situace kdy vyjde novější verze IMAP, tak stále bude možné provozovat vlastní server.

Před samotným návrhem a další implementací je nutné vyhodnotit podstatná rizika, která by vybraná řešení mohla představovat. Mělo by být vybráno takové řešení, které z dlouhodobého hlediska představuje nejnižší, nebo alespoň značně sníženou, úroveň rizika. Některé funkce sice vypadají z počátku líbivě, ale s ohledem na udržitelnost musí dojít k rozhodnutí co vše a jak udělat Má být nalezen kompromis. Nalezení rovnováhy mezi použitelností a bezpečností bude součástí návrhu jednotlivých navrhovaných řešení.

Řešená aplikace není vytvářena komerčním stylem. To znamená, že po jejím dokončení velmi pravděpodobně ustane další rozvoj. Respektive se aktualizace aplikace nemusí dostat ke všem uživatelům, kteří si ji stáhnou z veřejně přístupného úložiště. A právě proto s tím návrh většiny částí musí počítat.

2.2.1 Uživatelské rozhraní

Veřejně přístupné uživatelské rozhraní přímo ze sítě internet (i mimo lokální server/stanici) může značit velké riziko. V průběhu let se velmi pravděpodobně objeví nové zranitelnosti, které ale už nebude mít kdo opravit. Uživatelské rozhraní aplikace je řešeno s využitím prvků webové aplikace. To má své výhody v podobě multiplatformního využití. Stinnou stránkou je nutnost provozu nějaké formy webového serveru. Pokud by měl být server přístupný i mimo lokální server/stanici, přináší to nutnost řešit jeho časté aktualizace. Pokud by však přes všechna další bezpečnostní opatření byla chyba v samotné aplikaci, nelze to bez zásahu programátora řešit. S ohledem na zmíněné problémy tak bylo riziko provozu takového serveru vyhodnoceno jako příliš vysoké a navíc neopodstatněné. Lokální dostupnost je dostačující. Chyby se sice stále vyskytovat budou, ale riziko jejich zneužití již bude velmi omezeno na přijatelnou úroveň.

2.2.2 Synchronizace dat

Nově bude aplikace rozdělena na serverovou a klientskou část. To vychází z požadavku převodu na cloudovou platformu. Za běžných okolností spolu nemusí vůbec nijak komunikovat. Serverová část vyhodnocuje zprávy zcela nezávisle na klientovi a pracuje s externím zdrojem dat. Což v tomto případě je přímo e-mailový server, který je provozován odborníky.

Klientská část však musí podporovat úpravy konfiguračních souborů. A všechny tyto změny je nutné nějakým způsobem synchronizovat na serverovou část. S ohledem na udržitelnost musí být navrhnut takový způsob, který neohrozí fungování serveru. Nabízená řešení budou dále v rámci této práce popsána.

3 Cloudové platformy

Výběr vhodné cloudové platformy není jednoduchý úkol. Platforem existuje celá řada. Rozdíly přitom nemusí být zásadního charakteru. Na provoz aplikace, která je předmětem této práce, není třeba žádného výkonného systému. Při vlastním šetření bylo prokázáno, že postačí jedno virtuální jádro procesoru a 512 MB operační paměti RAM. To odpovídá těm nejlevnějším tarifům.

3.1 Náklady

Nákladová efektivnost provozu cloudové IT infrastruktury bývá pro uživatele stěžejní. Úspor lze dosáhnout i výběrem fyzického umístění datacentra. Větší poskytovatelé obvykle nabízí více lokalit s rozdílnou cenovou hladinou [18]. Přitom se jedná o zcela totožný produkt/službu. Nejprve se však musí vyhodnotit regulační požadavky. Tedy jestli systém zpracovávající potenciálně citlivá data může být provozován mimo jurisdikci EU [19]. Teprve podle toho lze vybírat vhodnou lokalitu.

Lepších cenových nabídek lze dosáhnout přes registrované partnery, kteří služby pouze přeprodávají [20]. Od původního poskytovatele cloudové platformy přebírají závazek podpory. Při vzniku problému se tak uživatel obrací na svého partnera, který s ním problém vyřeší. Otevírá se tak šance vyjednat lepší cenové podmínky. Ty záleží i na objemu objednaných služeb. Čím více služeb si zákazník objedná, tím nižší jednotkovou cenu může získat [21].

3.2 Varianty integrace aplikací

Existuje více strategií převodu vlastní aplikace do cloudového prostředí. První volba využívá běžného virtuálního počítače (VM). Zásadní výhodou je přenositelnost celého virtuálního počítače ke konkurenci. Přenos spočívá v exportu úložiště VM z původní platformy a importu na nové platformě. Toho lze využít i při přesunu na vlastní server. Druhá varianta využívá serverless přístupu. Tento přístup vyniká tím, že se správce stará pouze o svoji aplikaci. Nemusí se zajímat o správu systému jako takového [22]. To zajistí poskytovatel služby. Nevýhodou je složitá migrace jinam.

Dalším relevantním požadavkem může být garantovaná dostupnost služby. Při výpadku cloudu nelze nic dělat. Služba prostě nefunguje a zbývá jen čekat, než to poskytovatel vyřeší [23]. Problém způsobí i výpadek připojení k internetu [24]. Služba sice běží, ale není dostupná.

3.3 Zabezpečení dat

Přechod do cloudu je často prezentován jako moderní koncept IT infrastruktury [25]. Dochází k nahrazování vlastních serverů a převodu aplikací na cloudovou variantu. A to za měsíční poplatek. To znamená svěřit svá data cizí firmě, která cloud provozuje. S tím se pojí možná rizika. Nelze nezávisle ověřit, jak poskytovatel data ukládá, co s nimi dělá a kdo k nim má přístup. Nebylo by to poprvé, co zaměstnanec firmy ukradl data [26]. Nemusí jít ani o záměr. Vychytralost útočníků je velká, do cloudu se prostě nabourají a data odcizí [27]. Na druhou stranu je nutné vyhodnotit, jestli má interní IT oddělení dostatečné znalosti ke správě vlastních systémů. V některých případech se ukazuje, že rizika cloudu převáží základní chyby, kterých se dopouští právě interní IT [28].

3.4 Výběr vhodné platformy

Jak již bylo zmíněno na začátku kapitoly 3, platforem existuje nespočet a nelze je srovnat všechny. Níže vybrané platformy jsou mix různých společností, které nabízejí službu požadovaného typu. Tedy nabízí virtuální počítač, nebo serverless variantu s podporou aplikací napsaných v jazyce Python.

3.4.1 Microsoft Azure

Microsoft Azure je komplexní cloudová platforma, která nabízí mnoho služeb. Ať už jde o virtuální počítače, disková úložiště, různé databáze a mnoho dalšího [22]. Do výběru byla zařazena z důvodu existujícího napojení stávající aplikace na API MS Graph. To se využívá pro přístup k serverům Exchange Online, které jsou součástí infrastruktury Microsoft Azure. Zadavatel je tedy již seznámen s prostředím a správou této platformy. Na druhou stranu je potřeba zmínit, že se jedná opravdu o velmi komplexní platformu. Konfigurace je rozsáhlá a vyžaduje určitou znalost prostředí. Pro nové uživatele nemusí být plně přehledná. Rychlost načítání konfiguračního rozhraní není blesková, což pro někoho může být mírně frustrující. Rozhraní se také v průběhu času vyvíjí a dochází k přeskupení ovládacích prvků. Návody, které popisují postup zprovoznění aplikace na této platformě tak mohou brzy zastarat.

První zvažovanou možností je využití virtuálního počítače. K dispozici je jako VM s operačním systémem Windows nebo Linux. Z Licenčních důvodů (je zdarma) bude zvolen OS Linux, který má navíc nižší požadavky na HW. Také to zajišťuje snadnou přenositelnost na jinou platformu. Není využito žádné specifické vlastnosti Azure. Proto byla ke srovnání zařazena tato varianta. Výběr vhodného tarifu byl

proveden přes konfigurátor, který na základě požadavků zákazníka doporučí nejvhodnější tarif s optimálními parametry [29].

Druhá zvažovaná možnost je Azure Functions. Jedná se o serverless řešení, které podporuje provoz aplikací vytvářených (mimo jiné) v jazyce Python. Nabízí plánované spouštění úloh [30]. Výhodou je, že se do vytvořené Azure aplikace vloží pouze kód. Není nutné řešit aktualizace operačního systému a podobně. Ale vytváří se tím vendor lock-in, tedy složitější přesun na jinou cloudovou platformu [31]. Při přesunu by totiž muselo dojít i k úpravám kódu aplikace, což je nákladné. Tato varianta proto byla zavrhnuta. Výhodou je tarif, který nabízí nulovou spotřebu finančních prostředků při nepoužívání aplikace. Platí se na bázi vykonaných akcí [30]. Což ale nemusí vyhovovat všem. Někdy je třeba znát fixní měsíční náklady předem.

3.4.2 Google Cloud

Google Cloud je další z řady komplexních cloudových platforem. Nabízí virtuální počítače, úložiště, databáze a mnoho dalších produktů. Pro účely vyzkoušení této cloudové platformy nabízí Google kredit 300$ [32]. Což postačuje na plnohodnotné otestování provozu aplikace.

3.4.3 Python Anywhere

Python Anywhere je příklad serverless řešení. Specializuje se primárně na webové aplikace. Ale ani provoz jiného typu aplikace není vyloučen. Nabízí přímou editaci kódu přes webový prohlížeč. Takže je možné komfortně provádět úpravy kódu i bez instalace lokálního vývojového prostředí. Součástí je také bohatá knihovna předinstalovaných knihoven, které tak není třeba ručně instalovat. Nabízí automatické spouštění Python skriptů a vyhovuje tak architektuře dohledové aplikace. Samotná platforma Python Anywhere pro svůj provoz využívá Amazon AWS cloud. Přidaná hodnota tak spočívá v grafickém prostředí, přes které probíhá správa a konfigurace celé platformy [33].

3.4.4 Clever Cloud

Clever Cloud nabízí serverless řešení s vysokou dostupností. Což se hodí pro aplikace vyžadující nepřetržitý provoz. Podporuje automatické škálování pro případy nenadálé potřeby zvýšit výpočetní výkon. Platba probíhá s využitím kreditů. Aplikaci je možné spouštět pouze dle potřeby. Když neběží, nespotřebuje žádné kredity [34]. Podporuje jazyk Python od zastaralé verze 2 až po nejnovější verze 3.

3.4.5 Forpsi Cloud

Forpsi Cloud nabízí především privátní virtuální servery VPS, ale také i další služby jako zálohovací a výpočetní servery. Pro potřeby této práce byl vybrán nejlevnější produkt Cloud VPS, který splňuje výkonové požadavky. V rámci produktu není poskytována správa operačního sytému. Tu si musí správce zajistit sám [35].

3.4.6 OVHcloud

Platforma OVHcloud nabízí nepřeberné množství služeb. Dle tvrzení na stránkách firmy neprodávají uživatelská data a mají vysoký standard v oblasti zabezpečení dat. Jako vhodná služba pro srovnání byl vybrán produkt VPS. Tento produkt vyžaduje vlastní správu operačního systému [36].

3.4.7 Srovnání

Na základě výše uvedeného textu byla sestavena srovnávací tabulka 3.1, která uvádí parametry cloudových služeb včetně cenového srovnání. Ceny se mohou v budoucnu změnit a tak je platnost cen uvedena k roku 2025. Z funkčního pohledu jsou všechny cloudové platformy srovnatelné a dostatečné. Na všech by bylo možné dohledový systém spustit. Typem platformy VM se myslí virtuální počítač. Označení SV znamená použití serverless řešení.

Platforma	Typ	CPU	RAM	Disk	Cena/měsíc	Název
Microsoft Azure	VM	1x	512 MB	30 GB	123Kč	B1ls
Google Cloud	VM	1x	900 MB	30 GB	122Kč	f1-micro
Python Anywhere	SV	-	-	1 GB	125Kč	Hacker
Clever Cloud	SV	1x	256 MB	-	113Kč	pico
Forpsi Cloud	VM	1x	1 GB	20 GB	61Kč	VPS O1I1
OVHcloud	VM	1x	2 GB	20 GB	94Kč	VPS Starter

Tab. 3.1: Srovnání parametrů cloudových platforem

4 Analýza stávajícího řešení

Dohledový systém byl vytvořen s cílem uspořit čas správce IT infrastruktury. Hlavním požadavkem byla jednoduchá konfigurace a správa. Stávající verze v1.3/v1.5 toto stále splňuje a byla v tomto ohledu funkční. Automatizovaně prochází přijaté zprávy, které následně zpracuje a přesune do archivní lokace pro možnou budoucí analýzu. Správce dohledového systému se může věnovat mimořádným a důležitým situacím. Ty mohou značit nějaký problém. Nemusí se zabývat procházením nekonečné fronty méně významných zpráv. Avšak nejedná se o úplně bezvýznamné zprávy. Většina přijatých zpráv pouze informuje, že zařízení funguj správně. Není možné je ignorovat a vůbec nevyhodnocovat či rovnou mazat.

Poslední testování před zahájením dalších prací v roce 2024 potvrdilo, že aplikace byla i po letech stále funkční. Při provozování stávající verze v praxi se však ukázalo, že existují určitá místa ke zlepšení. Například nebylo možné na zprávu přiřadit více štítků se zachováním kompletní funkčnosti aplikace. Jedná se o chybu v návrhu, respektive chybné vyhodnocení požadavku.

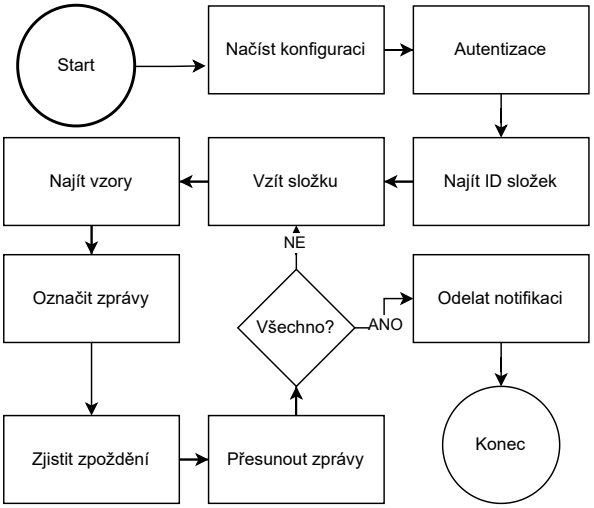
4.1 Architektura aplikace

Z pohledu uživatele je aplikace rozdělena na dvě části (viz obrázek 7.4), které spolu spolupracují. Jsou určeny k provozu na jednom zařízení. První část se stará o vyhodnocování zpráv a všechny s tím související kroky. Druhá část řeší obsluhu grafického rozhraní.

Načítání dat pro grafickou část probíhá přes vyhodnocovací část. Tím vzniká problém, že lze aplikaci konfigurovat a provozovat pouze na vybrané stanici. Jistě by šlo využít síťového disku, kdy jsou potřebné soubory sdíleny. Ale samotná aplikace toto prozatím neřešila. Navíc to zvyšuje nároky na správce, který musí být erudovaný v oblasti architektury aplikace.

Interní architektura aplikace využívá moduly a dílčí funkce. Ty na sebe vždy nějakým způsobem navazují a je nezbytné dodržovat přesné pořadí spouštění (zobrazeno na obrázku 4.1). Tento problém řeší samotná aplikace a běžného uživatele toto nemusí zajímat.

Využívají se centrální seznamy jako úložiště dočasných dat. Jsou k dispozici pro vybrané moduly. Ukládají se do nich jednotlivé mezivýsledky zpracování zpráv. Také obsahují vybrané součásti z konfigurace, které se týkají vyhodnocovací části. Jsou využity pro přenos těchto dat. Výhodou je dostupnost dat pro více modulů, jsou na jednom místě. Nevýhodou je nejasnost, kde přesně dochází k tvorbě seznamů. Není to úplně přehledné.

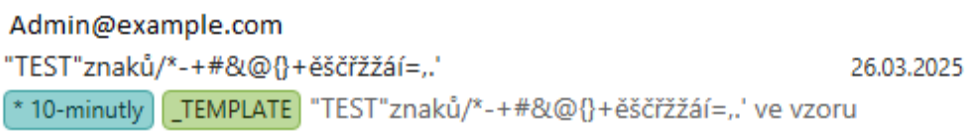


Obr. 4.1: Zjednodušený diagram vyhodnocovací části

4.2 Vyhodnocení zpráv

Zprávy se nachází v předem vybrané složce. Využije se integrovaných filtrů e-mailového řešení pro zatřídění zpráv například podle názvu firmy. Takto zatříděné zprávy jsou poté vstupem do modulu vyhodnocení zpráv. Správce nejdříve určí vzory pro obvykle docházející zprávy. Každý vzor následně označí k tomu určeným štítkem, jak je vidět na obrázku 4.2. Štítky mají několik funkcí. Jsou využívány správcem pro označení nějakého stavu. Dále se používají k definici rozestupu zpráv a také nastavení povoleného zpoždění.

Modul obdrží seznam složek, které má vyhodnotit. Vyhodnocení probíhá jednotlivě pro každou složku samostatně. V každé složce dojde k vyhledání všech vzorů. Jejich archivace se z důvodu možných změn uživatelem neřeší. Následuje vyhodnocení zpráv, kdy se prochází všechny nezpracované zprávy. Pokud zpráva souhlasí se vzorem, je dle něj označena příslušným štítkem a poté přesunuta do archivu. Mezitím probíhá vyhodnocení dochvilnosti zpráv. Modul vyhodnocení také posílá data do modulu notifikací. Sbírají se počty zpracovaných zpráv, jaké vzory byly zpracovány a případné problémy vzniklé při běhu aplikace.



Obr. 4.2: Zpráva označená jako vzor

4.3 Hlídání dochvilnosti

Stěžejní částí stávajícího dohledového systému je vyhodnocení dochvilnosti přijatých zpráv. Běžně mají všechny zprávy chodit v předem stanovený čas. A ve většině případů tak opravdu přicházejí. To značí, že monitorované zařízení funguje správně a provádí všechny nakonfigurované akce. Jestliže by došlo k zásadnímu zpoždění, může to značit nějaký závažný problém. Pokud záloha databáze normálně trvá deset minut a najednou trvá hodinu, měl by to správce prověřit. Z technických důvodů může dojít k nepatrnému zpoždění přijmu zprávy v řádech minut. Z tohoto důvodu je aplikace vybavena nastavitelnou funkcí přijatelného zpoždění. Správce určí časové okno, ve kterém musí zpráva přijít. Pokud zpráva přijde v daném časovém okně, je zpracována běžným způsobem (označena štítkem). Pokud přijde později, je o tom správce informován prostřednictvím notifikace. Problém nastane, pokud zpráva vůbec nepřijde. V tom případě se žádná notifikace nezasílá. Správce se tak nemusí dozvědět, že nějaké zařízení nefunguje vůbec, případně bylo vypnuto. Odpovídá to původnímu návrhu, kde se vyhodnocují pouze přijaté zprávy.

Samotné hlídání dochvilnosti zpráv funguje pouze pokud je aplikace spuštěna. Správce proto musí předem vyhodnotit, jak často bude aplikaci spouštět. Nejedná se o systém, který něco řeší v reálném čase. Aplikaci je možné spouštět každou minutu, což tento problém řeší. Běžně však není taková intenzita spouštění potřebná.

4.4 Připojení na server pomocí API

Stávající řešení podporuje připojení na e-mail servery Exchange Online od společnosti Microsoft v rámci produktu Microsoft 365 Copilot. Veškerá komunikace s těmito servery je řešena pomocí API Microsoft Graph. Využití proprietárního API vytváří omezení. Aplikace je funkční pouze u jediného cloudového poskytovatele. Není možné napojení na konkurenční produkty.

Čitelné přihlašovací údaje nejsou v konfiguračních souborech ukládány. Používá se token, který je složen z více než třiceti znaků. Nahrazuje tak běžně používané přihlašovací údaje typu jméno a heslo. Token má omezenou platnost po kterou ho je možné využívat. Informace o platnosti tokenu je součástí odpovědi od serveru [37]. Aplikace token získá samostatně po interakci s uživatelem. Ten musí provést přihlášení přes grafické rozhraní.

4.4.1 Získání ID složek

Každá složka na e-mailovém serveru obsahuje pole „Zobrazovaný název“ a „ID“. Pro přístup ke složce se používá jedinečný identifikátor (ID), uživatel však v konfiguraci zadává „Zobrazovaný název“. Využívané API nepodporuje vyhledání identifikátoru složky podle zobrazovaného názvu. Vždy je nutné procházet stromovou strukturu složek a vyhledat v ní požadované identifikátory podsložek [38]. Identifikátor složky aplikace využívá k výpisu zpráv složky. Z důvodu nejistoty stálosti identifikátoru složek nebylo jejich kešování původně implementováno. Při provozu v praxi se ukazuje, že identifikátor je neměnný. Dalším problémem bylo potenciální chování uživatele. Pokud by totiž složku/podsložku přesunul jinam, ke změně identifikátoru dojít může. Proto nedocházelo k ukládání výsledků hledání.

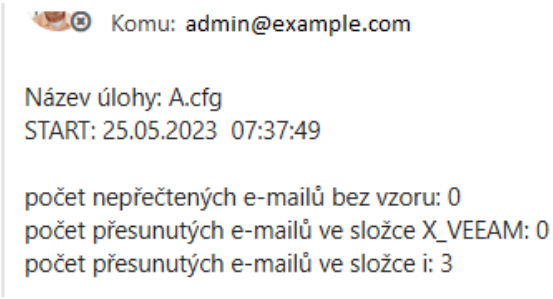
Neustálé vyhledávání složek (jejich ID) ve schránce je časově velmi náročné. Při běžném provozu zdržuje a prodlužuje dobu běhu aplikace. Zbytečně dochází ke zvýšenému počtu dotazů na API rozhraní. Počet dotazů závisí na velikosti schránky. Čím větší schránka, tím delší čas potřebný k nalezení všech potřebných identifikátorů. Tento problém je částečně řešen pomocí konfigurace. Správce může zakázat prohledání vybraných složek a tím rozsah prohledávání razantně omezit.

4.5 Notifikace

Notifikace se zasílají po každém úspěšném běhu aplikace. Pokud nastane neočekávaný problém, aplikace se pokusí zaslat nouzovou notifikaci. Musí však mít platný přístupový token, který je využit k přístupu na server. Obsahem notifikací jsou výsledky vyhodnocování zpráv. Tedy jaké vzory aplikace zpracovala, kolik zpráv bylo přiřazeno k nějakému vzoru a také kolik neodpovídá žádnému vzoru. Pokud při běhu došlo k nějaké chybě, je to také zaznamenáno.

Informace k odeslání se sbírají v modulu „notifications“. Ten po skončení běhu všech ostatních částí (viz obrázek 4.1) zajistí odeslání notifikace. Zasílá je na předem nakonfigurované e-mailové adresy, které jsou pro všechny úlohy stejné. Do předmětu notifikace je možné přidat vlastní označení. Pro každou úlohu lze nastavit jiné. Pokud není nastaveno, použije se globální hodnota, případně žádná.

Pokud se sejde vyšší počet událostí, které je třeba správci předat, stávají se notifikace méně přehledné. Nedochází ke shrnutí nejdůležitějšího obsahu. Výhodou je dohledatelnost všech událostí. Nevýhodou je možné přehlédnutí významnějších zpráv. Nedochází k žádnému řazení na základě významu zpráv. Ty jsou nyní řazeny chronologicky tak, jak je posbíral modul „notifications“. Výrazným nedostatkem je také absence editace podoby notifikace. Ta využívá jednu předem určenou formu, jako na obrázku 4.3, pro všechny úlohy. Bez zásahu přímo v kódu nelze nijak upravit.



Obr. 4.3: Příklad notifikace (částečný)

4.6 Provoz aplikace

Stávající aplikaci je možné provozovat přímo na vyhrazeném serveru. Nebo ve virtualizační platformě. Tam je následně instalován virtualizovaný operační systém, který slouží pouze k provozu dohledového systému. Zprovoznění aplikace je zcela v gesci administrátora systému. Na výběr jsou dvě varianty instalace. První možnost vyžaduje instalaci prerekvizit software Python, který aplikace využívá. Druhá možnost je přizpůsobena pro operační systém Windows. Jedná se kompletní řešení zabalené do jednoho spustitelného souboru formátu exe. Nevyžaduje žádnou instalaci podružného software, ani pokročilou znalost použitého operačního systému.

Aplikace využívá vestavěný plánovač úloh v operačním systému Windows. Pro jiné operační systémy nebyla v původním návrhu vytvořena podpora plánovaného spouštění úloh. Provoz na jiných platformách není vyloučen. Pouze to vyžaduje technickou zdatnost správce systému, který si aplikaci zprovozňuje.

Požadavky potřebné k provozu aplikace jsou takřka nulové. Provoz je možný i na nejnižších a nejlevnějších tarifech různých hostingových poskytovatelů. Serverová část disponuje pouze konzolovým výstupem. Ani ten ale není při použití plánovače vidět. Neprovádí se žádné výpočetně náročné operace. Ani spotřeba operační paměti není vysoká. Data se načítají v blocích a po zpracování nejsou v paměti zachována.

4.7 Způsob konfigurace

Nejstarší verze aplikace bylo možné konfigurovat pouze ručně a to přímým zásahem do konfiguračních souborů. Přehlednost takového řešení nebyla vysoká. Správce dohledového systému musel pomocí návodu vytvářet konfigurační soubory a na nic nezapomenout. Proto byla v pozdějších verzích přidána podpora konfigurace do grafického rozhraní. Tam je přehlednost vyšší a jsou dostupné návody. Ruční zásahy do konfigurace jsou stále možné.

```
[SETUSER]
taskFolder = md
do_scan = _TEMP,Doručená pošta,SERVIS
do_not_scan = Archiv,a,SYN,_DORUCENE,MNUSER
```

Obr. 4.4: Příklad konfiguračního souboru

Všechny konfigurační soubory jsou uloženy v určené adresářové struktuře. Pro přehlednost se používají podsložky. Například podsložka „watchdog“ obsahuje definiční soubory akceptovatelného zpoždění. Využívá se čitelných textových souborů, které lze upravit běžnými textovými editory. Jako oddělovací prvek hodnot se používá středník nebo čárka. Nynější uspořádání hodnot je omezující, jestliže potřebujeme uložit hodnotu se středníkem. Samotná aplikace takové hodnoty ukládat nepotřebuje. Uživatel v určitých případech ano. Při lepším uspořádání hodnot by to nepředstavovalo žádný problém. Některé hodnoty totiž středník nesmí obsahovat vůbec (třeba odesílatel e-mailu).

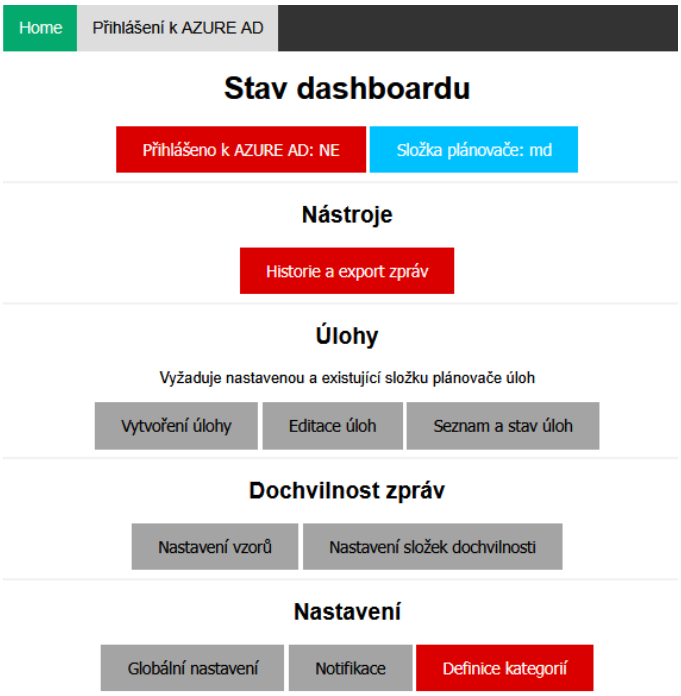
Konfigurační soubory musí být umístěny na stejném zařízení, na kterém probíhá spouštění aplikace. Dílčí změny v konfiguraci je proto nutné přenášet na server, kde je aplikace provozována. Synchronizace mezi více zařízení není řešena. Konfigurační soubory jsou primárně vytvářeny společně s tvorbou úloh, kde se také nastavuje individuální nastavení specifické pro danou úlohu. Individuální hodnoty mají vždy přednost před hodnotou globální. Globální hodnoty, platné pro všechny úlohy, se vytváří v samostatné sekci hlavního konfiguračního souboru. Sekce jsou rozděleny na dílčí konfigurační prvky podobně jako na obrázku 4.4.

Pro správnou funkci vyhodnocovací části aplikace je nutné definovat význam e-mailových štítků. Štítky se používají k označování zpráv, definici vzoru a určení rozestupů mezi zprávami. V rámci konfigurace je možné štítky zapnout nebo vypnout. Vypnuté štítky nemají na funkci aplikace žádný vliv přestože se u nějaké zprávy mohou vyskytovat.

4.8 Grafické rozhraní

Po spuštění grafického rozhraní se zobrazí úvodní stránka (obrázek 4.5) se všemi právě dostupnými funkcemi. Některé funkce mají technologické prerekvizity a proto nemusí být z počátku zpřístupněny (značeno červeně). Pokud jsou všechny podmínky splněny, automaticky se zpřístupní (značeno šedě).

Rozhraní využívá lokální webové stránky. Tento přístup je vhodný s ohledem na výbornou kompatibilitu mezi platformami, kde lze aplikaci provozovat. Při spuštění grafického rozhraní se otevře webový prohlížeč s úvodní stránkou. Pokud máme



Obr. 4.5: Přehledová stránka

prohlížeč již spuštěný, otevře se nová karta. Přístup ke stránce je povolen pouze z lokálního počítače a proto zcela záměrně nevyžaduje přihlašovací údaje.

Navigace mezi stránkami je poměrně jednoduchá. Vždy se zobrazuje lišta, která informuje kde se uživatel zrovna nachází. Na liště je vždy zobrazeno návratové tlačítko, které uživatele přesměruje zpět na úvodní stránku. Uživatel se tak nemůže nikdy ztratit v hlubokém stromu dostupných nastavení. Nástroje jsou rozděleny na tematické sekce podle určení. Nastavení je sdruženo ve spodní části stránky, není očekáván častý přístup. Užitečné nástroje a správa úloh jsou ve vrchní části stránky, ty se používají nejčastěji.

4.8.1 Přihlášení

Přihlášení k e-mailové schránce je prvním krokem, který uživatel provádí. Jinak nedojde ke zpřístupnění všech funkcí stránky. Uživatel prochází několika podstránkami a plní požadované kroky. Využívá se přístupu „Device Code flow“ [39], tedy aplikace v posledním kroku vygeneruje kód zařízení. Uživatel kód zkopíruje a přihlášení provede přímo na stránkách Microsoftu. Autor aplikace k heslu přístup nezíská.

Výstupem přihlašovacího procesu je token, který si aplikace ukládá do vyhrazeného souboru. Platnost tokenu je však omezena [37]. Může se tedy stát, že již zpřístupněné funkce budou opět znepřístupněny. V tom případě uživatel opakuje proces přihlášení. Poté se uloží nový token, který nahradí ten neplatný.

4.8.2 Správa úloh

Plánované spouštění aplikace je zajištěno přes integrovaný nástroj „plánovač úloh“ v OS Microsoft Windows. Grafické rozhraní podporuje veškerou nezbytnou správu takové úlohy. K dispozici je přehled vytvořených úloh, kde se zobrazuje stav a další očekávané spuštění. Dále tvorba a editace úloh. Úlohy není nutné vytvářet přes grafické rozhraní. Pokud správce dohledového systému nemá důvěru k aplikaci a není jí ochoten předat heslo od svého účtu, nemusí. Stačí úlohu vytvořit ručně. V přehledu se přesto zobrazí. Heslo je vyžadováno pro plánované spouštění aplikace bez přihlášeného uživatele [40]. Třeba po restartu počítače.

Sekce tvorby úlohy obsahuje všechna potřebná pole, které musí uživatel vyplnit. Některé informace si aplikace samostatně zjistí ze systému. Zároveň dochází k vytvoření a uložení všech potřebných konfiguračních souborů.

Pokud uživatel udělal při tvorbě úlohy chybu, lze ji opravit přes grafické rozhraní. Jestliže byla úloha vytvořena ručně, může vzniknout problém s její úpravou. Způsobuje to chyba v kódování textu. Proto je doporučeno úlohy vytvářet přes grafické rozhraní a až poté je ručně upravit se zadáním hesla uživatele systému.

4.8.3 Export historie

Tato funkce se využívá, pokud nastal nějaký problém na monitorovaném zařízení. Správce potřebuje zjistit, kdy k problému došlo. Zvolí se požadovaná složka, vzor ve složce a poté časový rozsah, který je požadován. Rozhraní načte tato data přímo z API za pomoci serverové části aplikace (dodá podpůrné informace). Jakmile je obdrží, jsou zobrazeny v přehledové tabulce. U tabulky se nachází tlačítko export, které obsah tabulky uloží do textového souboru. Na každém řádku je vždy pouze jedna zpráva s nejdůležitějšími informacemi.

Vzhledem k využívání serverové části aplikace trvá načtení dat delší dobu. Je to z důvodu opětovného vyhledávání identifikátoru složky. Bez něj není možné zprávy získat. Využívá se stejného procesu jako při vyhodnocování zpráv. Jediný rozdíl je, že se zprávy neoznačí a nepřesunou.

Obr. 4.6: Zobrazení historie zpráv

5 Návrh úprav a nové funkce

Z analýzy původního řešení vyplývají nedostatky, které je nezbytné reflektovat. Zároveň musí dojít k vyhodnocení kam přidat nové funkce podle zadání. To částečně závisí na původně navržené struktuře aplikace, která bude v opodstatněných případech přepracována. Zároveň by nemělo docházet k ovlivnění původních funkcí aplikace. Všechny stávající funkcionality mají být zachovány minimálně ve stejné nebo lepší kvalitě. Zpětnou kompatibilitu konfiguračních souborů nelze zaručit.

Tato kapitola se snaží odhalit a navrhnout řešení na všechny problémy, které jsou známé ještě před implementací. Při reálné implementaci se zcela určitě objeví další neočekávané problémy. Jejich řešení ale ze zjevných důvodů tato kapitola neřeší.

5.1 Úprava struktury

Aplikace bude převedena na cloudovou platformu. Tím vyvstává nutnost upravit celkovou strukturu aplikace. Aplikace bude rozdělena na dvě zcela oddělené části (viz nový koncept na obrázku 7.4). Existující propojení (popsáno v kapitole 4.1) se musí zrušit, respektive plnohodnotně nahradit. Cílem je, aby klientská část pracovala zcela samostatně. Úpravy jednotlivých modulů se řeší samostatně. Každý modul má definované úkoly, které plní. Výstup se nebude měnit.

5.2 Vyhodnocení na základě obsahu

Dosud probíhalo vyhodnocení zpráv na základě shody předmětu a odesílatele se vzorem (zobrazen na obrázku 4.2). Některá zařízení však zasílají předmět zprávy stále stejný. Další informace o stavu se nachází přímo v těle zprávy (podobně jako na obrázku 6.2). Takže se musí brát ohled i na tuto část. V prvním kroku se upraví modul vyhledání vzorů, který také doposud pracoval pouze s předměty zpráv.

Funkcionalita bude zařazena do existujících funkcí v modulu „CORE“. Původní způsob vyhodnocení zůstane plně zachován. Ke každému vzoru bude přidán identifikátor vzoru. Ten určí, o jaký druh vzoru se jedná. Označení „[body,nazev_sablony]“ představuje vzor, kde se bude hodnotit tělo zprávy. Je následován názvem šablony, která definuje co v těle zprávy hledat. Pokud bude obsahem „[subject,-]“, aplikuje se původní funkce, která porovnává jen odesílatele a předmět zprávy. Pro účely vyhodnocení dochvilnosti zpráv se budou všechny vzory ukládat do úložiště vzorů.

Samotné získání obsahu z těla zprávy není úplně triviální. Zprávy mají různé formáty. Ať už textové nebo s využitím HTML. Součástí zpráv bývá časový údaj a různé další proměnné hodnoty. Nelze tak aplikovat jednoduché porovnání obsahu

zprávy se vzorem. Pro spolehlivé přiřazení k relevantnímu vzoru se použije řádkového vyhodnocení.

Systém bude pracovat s klíčovými slovy. Za toto slovo některá zařízení vkládají detail stavu (viz obrázek 6.2). Klíčových slov se ve zprávě může nacházet více, musí se vyhodnotit všechna. Využije se zjednodušeného systému šablon. Na vstupu bude šablona s předpokládaným obsahem. Ten se srovná s právě přijatou zprávou a získají se požadované informace. S tím souvisí potřeba vzory nějakým způsobem ukládat pro další použití. Stávající systém ukládání vzorů není dostatečný. Proto bude vybrán nový způsob ukládání, který reflektuje požadavky. Tím dojde k odstranění nedostatku původního návrhu. Ten spočívá v nemožnosti využít některé znaky v předmětu zprávy (například středník).

5.3 Modul dochvilnosti

Dochvilnost se dosud vyhodnocovala jen u zpracovaných zpráv (popsáno v kapitole 4.3). Že zpráva zcela chybí (nikdy nepřišla) modul nepoznal. Analýzou stávajícího modulu dochvilnosti bylo zjištěno, že chybějící zprávy detekovat umí. Jen nedocházelo ke sběru chybějících zpráv, což vyřeší funkce „timeout“. Primárně tak dojde k optimalizaci a konsolidaci původního kódu. Nynější logika je velmi obsáhlá a v některých případech zbytečně složitá. Cílem je snížení komplexity kódu.

5.3.1 Timeout zpráv

Funkce „timeout“ bude reagovat na požadavek sbírat chybějící zprávy. To znamená, že pokud zpráva vůbec nepřišla, uživatel se o tom prostřednictvím notifikace dozví. Budou využity prvky stávající konfigurace, které se využívají pro výpočet akceptovatelného zpoždění. Zásah do konfiguračních souborů není nutný. Výsledkem vyhodnocení dochvilnosti mohou být celkem čtyři stavy. První značí, že zpráva přišla v očekávaný čas. Druhý stav informuje o zpožděné zprávě. Třetí stav je nový, a to chybějící zpráva. Ta se následně uloží pro další využití. Poslední stav označuje situaci, která vznikne špatnou koordinací plánovaného spouštění aplikace a očekávaného přijetí zprávy. Tedy pokud zpráva dosud nedošla, ale stále běží limit akceptovatelného zpoždění. Tato informace je méně důležitá a zobrazí se pouze jako varování. Při dalším běhu aplikace pak bude reagováno i na možnost, že zpráva nakonec nepřišla.

5.4 Synchronizace dat

Přechod na cloudovou platformu vytvořil nový problém. Konfigurační soubory se musí nějakým způsobem kopírovat mezi klientskou a serverovou částí. Lze zvolit hned několik způsobů přenosu dat. Každý přístup má určité technické výhody a nevýhody. Je nutné brát ohled na to, že aplikace je spouštěna periodicky. Neběží tedy neustále a s tím musí synchronizační funkce počítat. Také není jisté, jestli bude v cloudu dostupná veřejná IPv4/IPv6 adresa. Bez ní by přímé spojení nefungovalo.

5.4.1 Přímé HTTPS spojení

Aplikace může provozovat veřejně přístupné HTTPS rozhraní, kam se budou data zasílat. Po přijetí požadavku se zapíše obsah do určeného konfiguračního souboru. Ověření uživatele by probíhalo s využitím již dostupného tokenu.

Toto řešení nicméně není dlouhodobě udržitelné. Vyžaduje poměrně častou údržbu jak samotného serveru, tak potenciálně i samotné aplikace. Nelze vyloučit existenci chyb, které může útočník využit k napadení systému. Zvyšuje to nároky na správce. Další problém je, že by služba poskytující HTTPS server musela běžet neustále, což se neslučuje s principem funkce aplikace. Riziko takového řešení bylo určeno jako neúměrně vysoké.

5.4.2 Terminálové spojení

Terminálové spojení bude existovat minimálně pro správu cloudového prostředí. Nabízí se tedy využít toto spojení i pro automatickou synchronizaci dat. Změny se mohou přenášet pomocí příkazů. Nejdříve se naváže spojení. Poté se ověří aktuálnost konfigurace a následně by došlo k samotnému přenosu změn. Přenos přes příkazy by vyžadoval vytvoření nové funkce, která by obstarávala jejich obsluhu.

Toto řešení je poměrně bezpečné. Stále existuje problém bezpečného uložení přístupových údajů k serveru. Tam by postačovalo založit separátní uživatelský účet s minimálním oprávněním. Přesto zbývá vyřešit problém kompatibility mezi platformami. Jde o záložní řešení.

5.4.3 Servisní zpráva

Poslední zvažovanou možností je využití servisních zpráv. Aplikace má již nyní přístup do e-mailové schránky. Využil by se tak existující omezený přístup. K e-mailové zprávě je možné přikládat přílohy. Právě přes přílohy by přenos souborů probíhal. Došlo by k vytvoření servisní složky, která bude vyhrazena právě a jen pro tyto účely. Uživatel přes grafické rozhraní na své klientské stanici upraví konfigurační soubor.

Ten se následně nahraje formou přílohy do servisní složky. Jakmile se spustí aplikace na serveru, proběhne nejprve ověření změn. Pokud změna existuje, aplikace si soubor stáhne k sobě na lokální disk. Ideálně se také ověří integrita souboru. Přijetím nevalidního souboru by došlo k odstávce serveru.

Mírnou úpravou by se dalo dosáhnout cíle, kdy jsou všechny potřebné konfigurační soubory dostupné pouze v servisní složce e-mail schránky. Na lokálním disku serveru by se tak vyskytoval pouze omezený počet informací. Nevýhodou pak je nutnost neustálého načítání konfigurace ze schránky.

Tato volba se jeví jako nejvíce vhodná. Riziko bylo vyhodnoceno jako nízké. Využívají se již dostupné zdroje dat. A navíc jde konfiguraci zobrazit a zkontrolovat přímo přes e-mailového klienta. Proto tato volba vítězí a bude využita.

5.5 Souhrnná statistika

Statistika bude nabízet základní přehled o funkci aplikace. Tedy kolik bylo celkově zpracováno zpráv během každého běhu. Statistika bude doplněna vhodně zvoleným grafickým znázorněním, které sleduje trendy. Zároveň nesmí docházet k velkému nárůstu objemu dat, která budou pro účely statistiky ukládána.

Sbírání dat bude probíhat zároveň s běžným provozem serverové části aplikace. Obsahem dat má být pouze číselná hodnota doplněná časovou hodnotou. Třeba jeden kalendářní den. Pokud počet uložených kalendářních dnů přesáhne nastavenou hodnotu, původní data se vymažou.

5.6 Konsolidace notifikací

Notifikace pro správce jsou nyní jednolité a při velkém počtu událostí také nepřehledné. Proto se navrhuje vytvořit šablonový systém. Uživatel si bude moci podle návodu upravovat šablonu pro každou úlohu zvlášť. Ve výsledku je tak počet šablon neomezený. Šablona může obsahovat uživatelský popis. Ten bude možné doplnit pomocí předem definovaných proměnných hodnot, které se při zpracování šablony aplikací nahradí za data z provozu.

Šablonový systém nebude povinný. Součástí aplikace bude výchozí šablona, která obsahuje stejnou strukturu notifikace, jako má stávající řešení. Pokud se uživatel rozhodne, že je pro něj výchozí šablona postačující, nemusí nic dělat. Automaticky se použije globální šablona. Efektivně se tím vyřeší také problém překladu notifikací do dalších jazyků. Uživatel si vytvoří šablonu přesně s tím jazykem, který požaduje. Také může vynechat pro něj nezajímavé informace.

Zprávy se stejným zněním, které aplikace při běhu nasbírá, se zobrazí pouze jednou. Pokud to bude vhodné, také se doplní číselnou hodnotou, která bude zobrazovat četnost výskytu upozornění. Nejdříve však proběhne revize zasílaných hlášek. Zbytečné budou odstraněny. To nebude platit pro chybové hlášky, které mají za cíl informovat o nekritické chybě. Nekritická chyba například označuje, že se nezdařilo vyhodnotit dochvilnost konkrétní složky. Pro ostatní složky chyba nenastane.
Pokud vyhodnocení dochvilnosti selže pro dvě složky ze tří, chybová zpráva se zašle dvakrát. Obsahuje totiž název problémové složky.

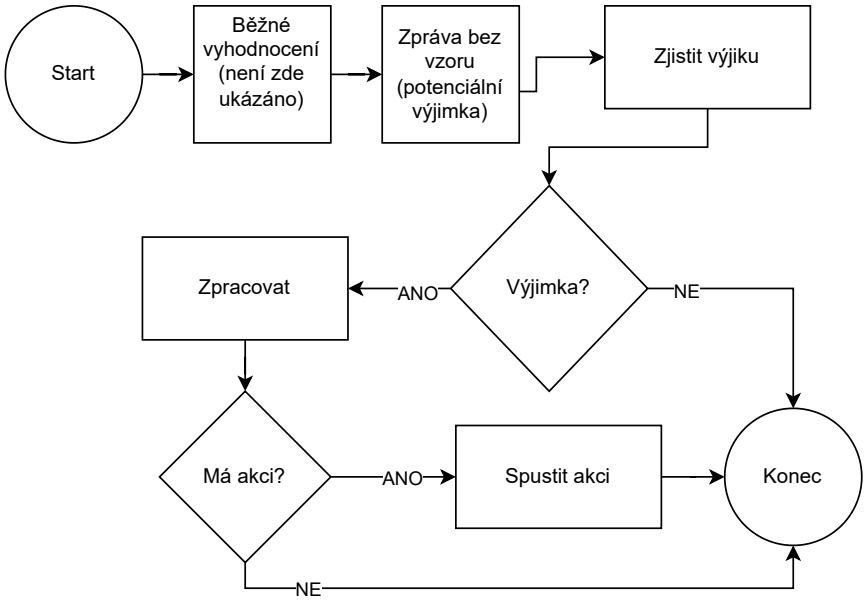
5.7 Systém výjimek

Pokud monitorované zařízení zasílá chybové zprávy, které neodpovídají žádnému vzoru, zůstávají záměrně nezpracovány. Správce musí každou takto nezpracovanou zprávu ručně vyhodnotit. Přitom se může jednat o méně významnou informaci, na kterou není nutné reagovat ihned. Prioritu řešení a závažnost zprávy tak správce rozhoduje vždy znovu. Takové zprávy však mohou přicházet poměrně často.
Například každý měsíc přijde několik zpráv informujících o nově dostupné verzi systému k instalaci. Nejedná se o natolik závažný problém, aby správce musel neprodleně reagovat. Stává se z toho rutinní činnost.

Kromě vyhodnocení na základě vzorů tak bude přidána funkce detekce výjimek. Jedná se vlastně o jakési sekundární vyhodnocení nezpracovaných zpráv. Definice výjimky bude na globální úrovni. To znamená, že nebude nutné definovat výjimku pro každou sledovanou složku zvlášť. Tím bude dosaženo jednotné konfigurace výjimek na jednom místě. Nicméně i tak bude možné omezit působení výjimky na konkrétní úlohy. Tím se omezí časová náročnost vyhodnocení výjimek.

Systém detekce bude fungovat následujícím způsobem (viz obrázek 5.1). Nejprve proběhne běžné vyhodnocení. Pokud by měla zůstat nějaká nezpracovaná zpráva, bude podrobena analýze na základě definice globálních výjimek. Pokud se bude shodovat s některou definicí výjimky, bude podle ní zpracována. To znamená, že bude minimálně označena štítkem pro pozdější analýzu.

Součástí budou čtyři definované úrovně závažnosti. Jejich znění půjde upravit. První je nízká úroveň, na kterou není třeba bezprostředně reagovat. Střední, která značí závažnější problém. Předposlední je vysoká, která značí závažný stav k okamžitému řešení. A poslední je kritická úroveň, která značí vysoce závažný stav ohrožující funkci zařízení nebo aplikací. Na každou vyhovující zprávu bude možné uplatnit pouze jednu úroveň závažnosti.



Obr. 5.1: Návrh systému výjimek

5.7.1 Akce

Ke každé výjimce bude možné přiřadit akci, respektive její název. V rámci návrhu bude připraveno několik experimentálních akcí. První akcí bude odeslání mimořádné notifikace pro nastaveného příjemce. Další akce umožní spustit skript, který si uživatel vytvoří a uloží na předem stanovené místo. Zvolení akce bude volitelné, nebude nutné ji nastavit. To se hodí především u nezávažných zpráv.

Při využívání vlastních akcí se meze nekladou. Správce si může zajistit napojení například na SMS bránu. Není vyloučeno využití i různých komunikačních platforem, které nabízejí napojení přes API. O závažných zprávách se tak může dozvědět prakticky okamžitě. Ale to již není součástí této práce.

5.8 Grafické rozhraní

Grafické rozhraní má působit přehledně. Je to totiž část, se kterou se uživatel nejčastěji setká. Obsahuje jak konfiguraci, tak různé podpůrné funkce a nástroje. V rámci úprav dojde k zakomponování nových funkcí a informací. Systém přihlašování se musí upravit, aby reflektoval nové spojení přes IMAP. S tím jak budou přibývat nové funkce se změní i design samotné úvodní stránky. Závislost grafického rozhraní na serverové části bude odstraněna. Proběhne nahrazení využitých externích funkcionalit integrovanými moduly. S tím se pojí tvorba nových zdrojů dat.

5.8.1 Přehled chyb

Nejvýznamnější novou funkcí bude přehled chyb. Na jedno kliknutí zobrazí přehledovou tabulku. Obsahem tabulky bude informace ohledně stavu zpráv v jednotlivých složkách. Tedy jestli složka neobsahuje nějakou výjimku či nezpracovanou zprávu. Tyto informace se souhrnně zobrazí číslem. Po kliknutí na číslo se zobrazí detailní výpis zpráv, kterých se problém týká. Pokud správce nazná, že situaci vyřešil, může stisknout tlačítko vyřešit. To způsobí označení zprávy štítkem vyřešeno. V důsledku se pak takto označená zpráva skryje a nebude již zobrazena v přehledu problémů. Nelze ji přímo smazat, protože by došlo k narušení požadavku na archivaci a pozdější dohledání zpráv.

5.9 Podpora IMAP

Z analýzy stávajícího řešení v kapitole 4.4 vyplývá, že je aplikace závislá pouze na jednom dodavateli. Proto bude přidána podpora pro protokol IMAP, který je podporován mnoha servery. Před implementací bude nutné prověřit, jestli budou splněny všechny požadavky kompatibility. Pokud splněny nebudou, nelze IMAP podporovat.

Zásadní až klíčový je požadavek na podporu tvorby uživatelských štítků. Ty jsou důležitou součástí samotného vyhodnocovacího procesu. Tuto podporu bude nutné předem zjistit. Dle dokumentu RFC9051 [41] se jedná o volitelně implementovanou funkci. Může se tedy stát, že používaný IMAP server tuto funkci vůbec neumí.

5.9.1 Zjištění podpory na serveru

Jestli zvolený server funkci podporuje nebo ne lze zjistit minimálně dvěma způsoby. První možnost je ručně přes příkazový řádek. Uživatel si nainstaluje potřebné programy a se znalostí příkazů na serveru podporu zjistí. Že server funkci podporuje značí symbol hvězdičky, který se vypíše u seznamu již vytvořených štítků. Každý server nějaké štítky obsahuje. Pokud ne, musí si je uživatel vytvořit [42]. Znalostí příkazů IMAP pravděpodobně nedisponuje téměř žádný běžný správce. Proto bude přidán modul, který automaticky podporu vlastních štítků ověří. Jestliže funkce podporována nebude, informuje správce, že vybraný server nelze použít.

Na základě testování v rámci příprav bylo ověřeno, že podpora je zajištěna například na serverech firmy Seznam.cz a tak budou tyto servery v rámci práce využity. To stejné nelze prohlásit o serverech firmy Microsoft s produktem Exchange Online. Tam podpora nebyla zjištěna a tak protokol IMAP konkrétně zde nebude vůbec fungovat [43].

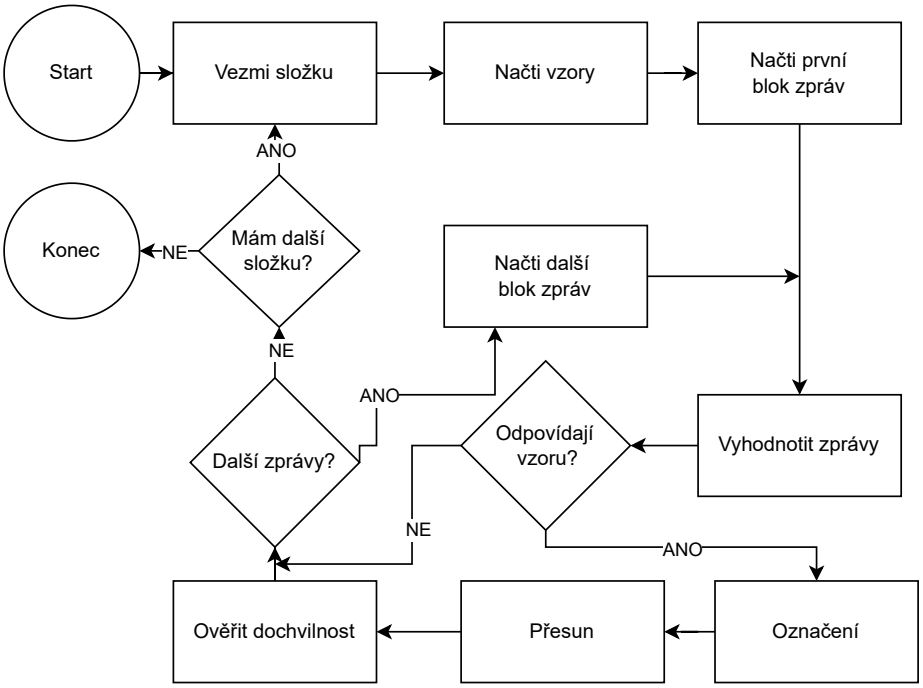
```
(XLIST,IMAP4REV1,AUTH=PLAIN,CHILDREN,SORT,I18NLEVEL=1,UIDPLUS,ID,MOVE,UNSELECT)
{PERMANENTFLAGS:(\\Seen,\\Answered,\\Flagged,\\Deleted,$Forwarded,\\*)}
```

Obr. 5.2: Výpis ze serveru IMAP (Seznam.cz)

5.9.2 Adaptace původních modulů

Všechny původní moduly bude nutné kompletně projít a stávající kód upravit. Jednotlivé úpravy vyplynou až při implementaci. Není možné je předem odhalit, projekt je rozsáhlý. V co největší míře se využije současné logiky průchodu zprávy.

Výstupem modulů je obvykle seznam s výsledky, se kterými se dále pracuje. Tam, kde to situace dovolí, se zařadí principiálně stejný modul. Pouze se přizpůsobí na protokol IMAP. Ve výsledku modul provádí rozdílné činnosti. Avšak výsledek bude ve stejném formátu, jako produkují moduly určené pro API. Z důvodu rozdílné funkce protokolu IMAP oproti API dojde k vytvoření zcela nových modulů, které to vyřeší. Některé moduly se nebudou řešit vůbec. Protokol IMAP má totiž jinak strukturované dotazy a odpovědi. Například je možné vypsat seznam všech složek a podsložek jedním dotazem na server [41]. Zatímco modul pro API musí zasílat řádově desítky dotazů.



Obr. 5.3: Stávající průchod vyhodnocení zprávy

6 Inovace serverové části

Úpravy serverové části se zaměřují na přizpůsobení modulů k provozu v cloudovém prostředí. Došlo k úpravě všech zasažených modulů. Zároveň bylo přidáno několik nových funkcí, například vyhodnocení zprávy dle jejího obsahu. Tato kapitola postupně většinu změn podrobněji popisuje.

6.1 Modul konfigurace

Modul konfigurace obstarává většinu práce s konfiguračními soubory. Jde o jejich načítání a ukládání. Pokud je povolena synchronizace, stahuje nové či upravené soubory ze servisní složky na lokální disk. Také zajišťuje odstranění nepotřebných souborů. Ty vznikají především po odstranění úlohy.

Původní koncept konfiguračních souborů nebyl optimální. Neumožňoval využít některé znaky pro uživatelské hodnoty. Došlo tak k jejich přepracování. Nyní se využívají dva formáty. První formát je textový. Používá se pro konfigurační soubory úloh. Je rozdělen do sekcí a na každém řádku se může nacházet právě jedna hodnota. Výhodou je výborná čitelnost souboru uživatelem. Ten může provádět ruční úpravy. Jako druhý formát byl zvolen JSON (příklad na obrázku 6.1), čímž byl odstraněn problém se zakázanými znaky (použití středníku bylo dříve zakázáno). Nevýhodou je nutnost dodržet platný formát souboru. Ruční uživatelský zásah tak není triviální. Proto se tento typ používá především pro interní účely aplikace. Není u nich předpoklad ručního zásahu. Je použit například pro zaznamenání vzorů zpráv. Obsluhu a aktualizaci souboru pak zajišťuje grafické rozhraní.

```
{
    "i": {
        "enabled": "ON",
        "templates": {
            "T1": {
                "sender": "admin@example.com",
                "subject": "\"TEST\"znaků",
                "categories": "['* 10-minutly']",
                "timeout": "0",
                "expected": "0"
            }
        }
    }
}
```

Obr. 6.1: Konfigurační soubor typu JSON

6.1.1 Zabezpečení přihlašovacích údajů

Pro přihlášení ke zdroji dat (e-mailová schránka) jsou vyžadovány přihlašovací údaje. Tyto údaje musí být nějakým způsobem uloženy na lokálním úložišti. S tím vzniká riziko odcizení těchto údajů při kompromitaci serveru. Případná nepozornost uživatele při zálohování složky s aplikací je další rizikový faktor. Proto byla implementována alespoň základní ochrana před zneužitím. Pokud aplikace využívá API, je na lokálním disku uložen pouze speciální token. Ten má na serveru nastaveno omezené oprávnění. Nelze tak získat plnou kontrolu nad účtem. Heslo k účtu se aplikace nikdy nedozví a nemůže tak uniknout. Proces přihlášení probíhá přímo přes servery Microsoft. Pro variantu IMAP bylo zvoleno kompromisní řešení, jelikož nelze garantovat podporu tokenů všemi poskytovateli. Heslo pro IMAP server je tak uloženo šifrovaně. Cílem této ochrany je znemožnit okamžitý přístup k heslu. Útočník musí vyvinout určitou snahu, aby heslo extrahoval.

6.2 Systém vyhodnocení zpráv

Systém vyhodnocení zpráv byl upraven, aby umožňoval přidání nových funkcí. Většina modulu se však stále zakládá na původním řešení, aby byl zachován původní princip vyhodnocení zpráv. Původně se pracovalo pouze s předmětem zprávy, štítky a odesílatelem. Pro nové funkce došlo k rozšíření načítaných údajů ještě o tělo zprávy. Princip vyhodnocení zpráv je následující. Nejprve se vyhledají vzory ve zvolené složce. Následně se načtou ostatní zprávy. Pokud zpráva odpovídá nějakému ze vzorů, je dle něj označena. Zároveň je zařazena do seznamu k přesunutí do archivní lokace. Pokud neodpovídá žádnému vzoru, zašle se na detekci výjimky.

Funkce ukládání vzorů byla odstraněna a zcela přesunuta do klientské části. Tam se k nim uloží informace o akceptovatelném zpoždění. Vyhledání vzorů však bylo zachováno a používá se v kombinaci s uloženými vzory. Systém vyhledání vzorů byl upraven tak, aby se snížila zátěž na API. Nově se využívá filtru, který načte pouze zprávy, které obsahují štítek označující vzor. Pouze takto označené zprávy 4.2 jsou načteny a po další kontrole uloženy do dočasného seznamu pro další použití. Jakmile dojde k vyhodnocení složky, dočasné seznamy jsou vymazány.

6.2.1 Detekce výjimky

Systém detekce výjimek je spuštěn pokud ve složce zůstala nějaká nezpracovaná zpráva. Pouze v takovém případě se načtou vzory výjimek. Každý vzor výjimky obsahuje pole odesílatel, předmět, text k detekci, závažnost a akci. Odesílatele není nutné vyplnit, jedná se o volitelný parametr.

Funkce má na vstupu zprávu, kterou má zpracovat. Porovná ji s šablonami. Pokud je v předmětu zprávy text „aktualizace k dispozici“, tak stačí zadat pouze „aktualizace“, aby platila shoda se vzorem. Nebo lze zvolit druhý mód, ve kterém je vyžadována přesná shoda. Následně je dle vzoru označena štítkem závažnosti. Štítek slouží pouze pro přehled a vyhledání zpráv se stejnou závažností.

Ke každému vzoru výjimky je možné přiřadit akci, která se spustí ihned po detekci výjimky. Celkem jsou předem připraveny dvě akce. První slouží k okamžitému zaslání notifikace na určený kontakt. Druhá umožňuje spustit uživatelský skript. Dá se tak definovat vlastní řešení jakékoliv situace.

6.2.2 Zpracování na základě obsahu

Zpracování na základě obsahu těla zprávy představuje možnost monitorovat další zařízení od různých výrobců. Nestačí jednoduché srovnání odesílatele a vzoru. Tato vlastnost je sice stále přítomna, ale je doplněna novou funkcí.

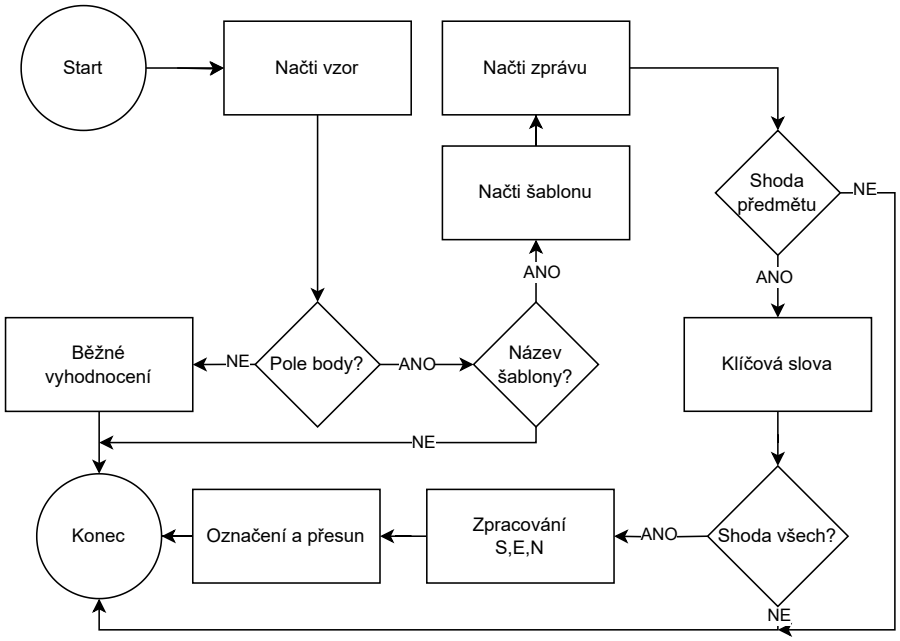
Prvním krokem tak bude standardní vyhodnocení zprávy. Porovná se shoda odesílatele a předmětu se vzorem. Pokud vzor obsahuje štítek s názvem „body“, značí to zprávu u které chceme vyhodnotit obsah. V takovém případě se na vzoru zprávy musí nacházet další štítek „body_název šablony“. Ten slouží jako jedinečný identifikátor šablony. Je velmi vhodné vyhradit pro tento způsob vyhodnocení unikátní předmět zpráv, jinak se prodlouží čas vyhodnocení. Načítání těla vyhodnocované zprávy totiž probíhá jednotlivě. Zatímco předmět zprávy a odesílatel je načítán předem, jelikož je objem dat dostatečně malý, aby se dal uložit do dočasného listu. To samé se nedá říct o těle zprávy, které zabírá místa více.

```
Notifikace stavu

Název NAS: Firemni-NAS1-AD
Závažnost: info
Datum/Čas: 2025/01/20 12:12:12

Název aplikace: Zálohování
Kategorie: PC
Zpráva: Zálohování PC bylo úspěšně dokončeno [Snímek]

2025 Firma, Inc.
```

Obr. 6.2: Příklad zprávy od monitorovaného zařízení

Druhým krokem je řádkové vyhodnocení těla zprávy. Šablona obsahuje informaci, co se má na každém řádku nacházet. Porovnává se tedy obsah řádku. Respektive minimálně jeho začátek. Jak je vidět na příkladu zprávy (obrázek 6.2), na každém řádku je klíčové slovo oddělené dvojtečkou. Za ní se nachází zájmové hodnoty. V šabloně se definuje, že na prvním řádku se vyskytuje „Název NAS“. Prvně se ověří, jestli řádek obsahuje toto klíčové slovo. Pokud ano, je extrahována hodnota za oddělovacím znakem. Případně je k dispozici možnost zadat přesný počet znaků, od kterého se má zájmová hodnota extrahovat. To se může hodit pro odstranění mezer.

Posledním krokem je zpracování extrahovaných hodnot. Ke každému klíčovému slovu se v šabloně nastavuje příznak. První je „S“, který značí shodu. Tedy extrahovaná hodnota se rovná hodnotě v šabloně dle klíčového slova. Druhá hodnota „E“ je zkratkou pro export. Tato hodnota je pak zaslána do modulu notifikací (všechny takto získané hodnoty). To se hodí například pro časovou hodnotu nebo pro v příkladu na obrázku 6.2 označenou hodnotu „Zpráva:“. Obsahuje detailní informaci o výsledku zálohování počítače, kterou potřebujeme vyhodnotit. Hodnota „N“ značí nedělej nic.



Obr. 6.3: Diagram vyhodnocení dle obsahu

6.2.3 Přesun zpráv

Zprávy k přesunu jsou v průběhu vyhodnocování shromážděny v dočasném listu „ListOfToBeMoved“. Ke každé zprávě je uložen její jedinečný identifikátor. Název

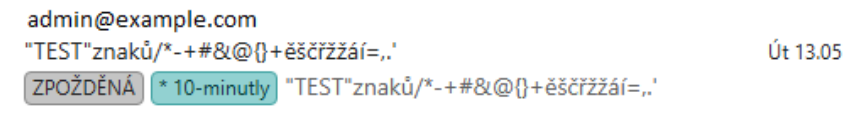
archivní lokace kombinuje podtržítko a název složky. Avšak archivační složky se nacházejí v centrální složce, která je pro tento účel určena. Proto se ke každé zprávě musí uložit ještě jedinečný identifikátor cílové lokace.

Jakmile proběhne označení zpráv a vyhodnocení dochvilnosti, je spuštěn proces přesunu. Postupně prochází list a zadává příkazy k přesunu. V případě přístupu přes API jsou požadavky k přesunu sdruženy do skupin po dvaceti zprávách. Nebo do vyčerpání zpráv. Následně je zaslán tento skupinový požadavek na API, které přesuny zpracuje. Poté se ověří, jestli nebyla ohlášena chyba API. Pokud ano, tak se chyba zaznamená v notifikačním modulu. Nicméně přesun ostatních zpráv není přerušen. Pokračuje se až do vyprázdnění listu. V případě přístupu přes IMAP se do funkce přesunu zpráv zašle celý list s identifikátory zpráv. Server jej zpracuje.

6.2.4 Dochvilnost zpráv

Hodnocení dochvilnosti zpráv bylo přepracováno. Celkový objem kódu byl snížen. Pracuje na principu časových slotů, kam je možné přiřadit pouze nejbližší starší zprávu. Nikdy ne tu předcházející slotu. Sloty se generují od nejstarší zprávy, která se nachází ve složce. Je zde předpoklad, že předcházející chyby jsou vyřešeny minulou notifikací. Konec generování slotů je pak aktuální datum a čas. Rozestupy časových slotů jsou určeny štítkem, který je na vzorové zprávě (obrázek 4.2). Nastavení štítků probíhá přes grafické rozhraní. Na jednom vzoru může být libovolný počet štítků. Ale pouze jeden, který definuje rozestup zpráv. To je určeno příznakem „zapnuto/vypnuto“. Pokud vzor obsahuje více zapnutých štítků, je vyhodnocení dochvilnosti pro konkrétní vzor přeskočeno. Ostatní vzory se vyhodnotí.

Následně probíhá přiřazování přijatých zpráv ke slotům. Podle počtu přiřazených zpráv se postupuje dále. Pokud není přiřazena žádná zpráva, znamená to, že nedorazila. Pokud je přiřazena pravě jedna, tak se ověří jestli přišla včas. Případně jestli byla zpožděna. U každého vzoru je možné definovat akceptovatelné zpoždění zprávy. V tom případě je k času slotu tento čas přičten. Jestliže se uživateli povedlo spustit vyhodnocení právě v době, kdy ještě probíhá akceptovatelné zpoždění, tak je na to v notifikaci upozorněn. Avšak nedojde k chybnému vyhodnocení zprávy do stavu „chybějící“. Jiné stavy jsou označeny jako chybové. Správce je o chybách informován prostřednictvím notifikace.



Obr. 6.4: Vyhodnocená dochvilnost zprávy

6.3 Modul notifikací

Modul notifikací byl značně inovován. Nově nabízí systém šablon a další menší vylepšení. Zvýšila se přehlednost notifikací a rovněž došlo k rozšíření konfigurace. Notifikace je možné dokonce i zcela vypnout. V tom případě se nebudou zasílat.

6.3.1 Systém šablon

Systém šablon byl implementován tak, aby to bylo pro uživatele co nejvíce jednoduché a srozumitelné. Tělové (body) šablony se ukládají jako běžný textový soubor. K editaci není třeba žádných specializovaných nástrojů. Postačí integrovaný textový editor, který je běžnou součástí operačních systémů. Výhodou je jednoduchost a čitelnost zvoleného řešení. Tak jak je šablona vytvořena (obsah jednotlivých řádků), tak se nakonec zobrazí příjemci notifikace.

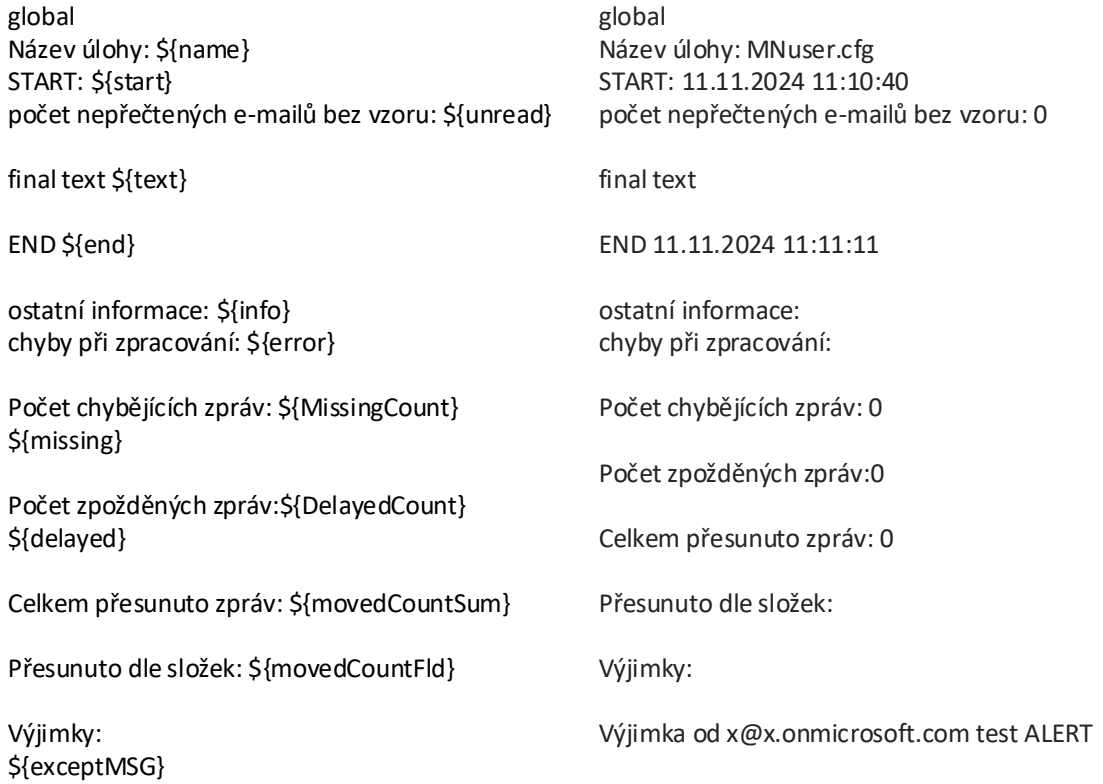
Uživatel si pouze napíše svůj text, který vhodně doplní předem připravenými proměnnými. Proměnné hodnoty jsou bezprostředně před odesláním notifikace nahrazeny daty, které dosud aplikace nasbírala. Každá proměnná je označena jedinečným názvem. Pokud chce uživatel proměnnou použít, musí zapsat její název do zástupné kolonky ve formátu ${název} a právě místo kolonky se ve výsledku zobrazí obsah proměnné. Na ochranu před chybou uživatele je také myšleno. Pokud by napsal do kolonky neexistující název, substituce hodnoty se neprovede.

Příklad šablony je viditelný na obrázku 6.5, kde lze vidět šablonu a pak zpracovaný stav. Jedná se pouze o testovací zobrazení a tak jsou některé sekce prázdné.

Pro zvýšení přehlednosti byl přidán šablonový systém i pro předmět notifikace. Je tak na první pohled zřejmé, zda vyhodnocení proběhlo v pořádku nebo ne. Záleží, co si tam uživatel navolí. Na rozdíl od tělové šablony se u předmětové šablony nepoužívá soubor. Vzhledem malému rozsahu předmětu je to zbytečné. Proto je předmětová šablona součástí jiného konfiguračního souboru. Mezi tělovou a předmětovou šablonou jinak není žádný rozdíl, pracují na stejném principu.
Pro každou úlohu si uživatel může navolit použití lokální šablony, která nahradí tu globální. To se velmi hodí pro uživatele s více monitorovanými lokacemi. Pro každou lokaci si do předmětu může nastavit jinou předponu, či úplně jiné uspořádání informací. Proměnné hodnoty nemusí využít.

6.3.2 Proměnné

Na základě zkušeností z provozu bylo předem vybráno několik hodnot, které je nutné v notifikacích zobrazovat. Například se jedná o výpis chyb, datum a čas, název úlohy a další. Všechny dynamické hodnoty, které původní aplikace vkládala do notifikace jsou převedeny na proměnné. Každá proměnná má unikátní název.



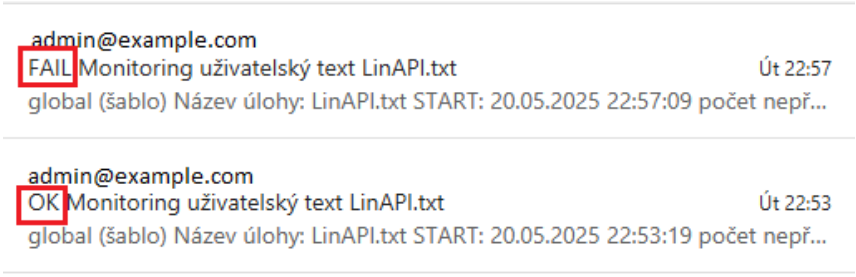
Obr. 6.5: Šablona a výsledek zpracování

Byla také přidána zcela nová hodnota pojmenována jako „FAILOK“. Ta slouží k jednoznačné informaci, jestli při průběhu vyhodnocování došlo k nějakému problému nebo ne (viz obrázek 6.6). Tato proměnná je vhodná především do předmětu notifikace, kde je ihned viditelná. Pokud nenastal žádný problém, obsahuje hodnotu „OK“. Jestliže došlo ke zpoždění zprávy, nebo nastal méně významný problém, je obsahem hodnota „warning“. A poslední stav „FAIL“ značí závažný problém. Ten nastane, pokud chybí nějaká zpráva od monitorovaného zařízení. Případně se vyskytla závažná chyba při běhu aplikace. Nebo také výskyt výjimky.

Všechny tři stavy proměnné „FAILOK“ jsou součástí konfiguračního souboru. Uživatel si tedy může zvolit jejich znění. Význam ale zůstane stejný, jak již bylo popsáno výše. Jde o globální nastavení. Lokální nastavení (specifické pro každou úlohu) není podporováno z důvodu neopodstatněnosti.

6.3.3 Adresáti

Ke každé úloze je nyní možné zvolit odlišné adresáty. Existuje zde rozlišení na globální a lokální adresáty. Globální adresáti se nastavují jako výchozí hodnota, která se použije, pokud uživatel nezvolil použití lokálních adresátů. Globální adresát se



Obr. 6.6: Příklad proměnné FAILOK

aplikuje na všechny úlohy. Zatímco lokální adresát pouze na individuální úlohu. K tomu je určen přepínač, kterým se volí mezi globálním a lokálním adresátem.

Jako ochrana před chybou uživatele se také kontroluje, jestli položka lokální adresát obsahuje nějakou hodnotu. Pokud ne, bude využit adresát globální. Nemělo by tedy docházet k situacím, že se notifikace neodešle z důvodu nevyplnění lokálního adresáta. Pokud však není vyplněn ani globální adresát, notifikace se jednoduše neodešle. Aplikace totiž neví, kam by měla notifikaci poslat. Průběh vyhodnocování by to však ovlivňovat nemělo. Odesílání notifikací je totiž jedna z posledních akcí, které aplikace před svým ukončením provádí.

6.3.4 Systémové hlášky

Systémové hlášky obsahují informace, které mohou být závislé na názvu složky. Například kolik zpráv bylo zpracováno ve složce „Servery“ a v několika dalších složkách. Dosud jsou hlášky uváděny pouze jedním jazykem. Proto došlo k přesunu těchto hlášek do konfiguračního souboru. Uživatel si je může volitelně přepsat vlastní hodnotou. Oproti šabloně, která se používá pro tělo notifikace, je zde omezený počet dostupných proměnných. Nejpoužívanější proměnné jsou většinou dvě. První je název složky a druhá je například počet/součet provedených akcí. Při editaci hlášek tak musí uživatel využít návodu. Tam jsou dostupné proměnné popsány.

Při obecných chybách se vypíšou hlášky moderního charakteru. To znamená, že obsahují informaci, která nevede k identifikaci problému. Typickým příkladem moderní hlášky je „Nastala chyba“. Kde chyba nastala a proč se uživatel bez detailní znalosti aplikace nedozví. Většinou jde o situace, které za běžného provozu nemají nastat. Tyto hlášky se často vyskytují u populárního software nejmenované společnosti, která je nicméně v této práci několikrát zmíněna. Nešlo tak nevyužít této ultimátní příležitosti vložit do aplikace moderní prvky. Aplikace tak obsahuje i hlášky jako „Jejda, něco se nepovedlo“. Jestli a jak to ovlivní psychický stav uživatele lze pouze spekulovat.

6.3.5 Tvorba textu k odeslání

Princip zpracování informací k odeslání zůstává shodný s verzí v1.5. Notifikační modul při provozu sbírá chybové a informační zprávy. Jakmile jsou dokončeny všechny potřebné akce (ukončení vyhodnocování), je spuštěn modul notifikací. Ten nejprve vytvoří slovník se základním obsahem. Ten je následně naplněn nasbíranými informacemi, které jsou doplněny šablonovým systémem. Předmět a tělo notifikace tak obsahuje uživatelem požadované informace v daném formátu.

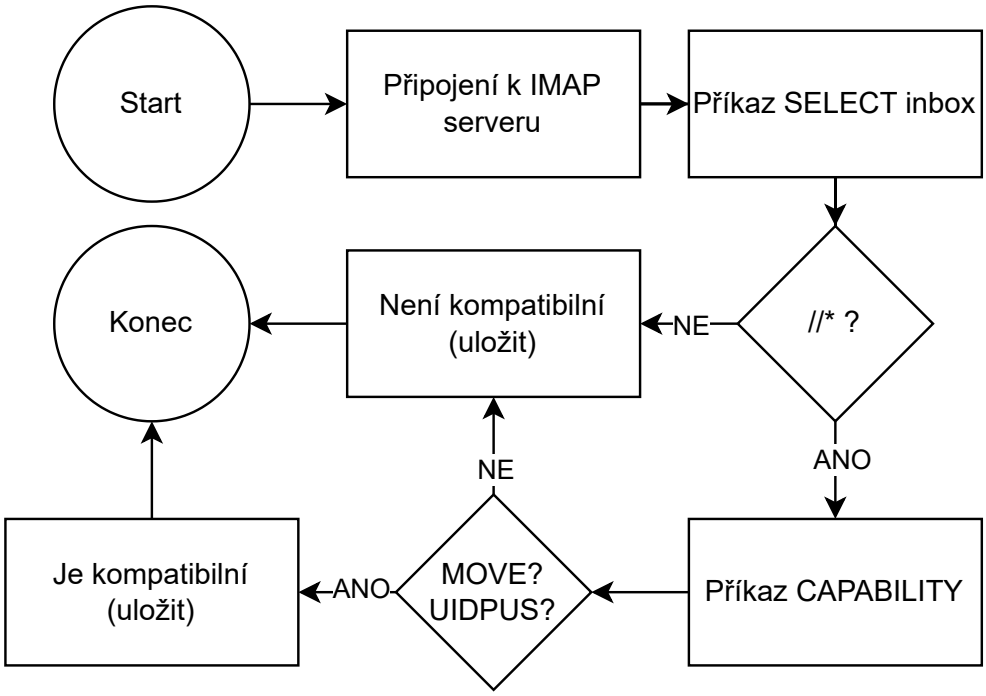
Slovník, který byl vytvořen v předchozím kroku je univerzální. Obsahuje všechny potřebné parametry jako je příjemce, tělo a předmět zprávy. Lze jej využít pro všechny implementované způsoby odeslání. Proto je využíván pro odeslání notifikace přes API a nyní bude využit i pro odeslání přes protokol IMAP. Obsahuje také nepotřebné hodnoty a ty jednoduše nebudou pro IMAP využity. Odeslání notifikace samozřejmě proběhne přes protokol SMTP. Ale pro zachování archivu odeslaných zpráv se notifikace zároveň uloží do složky odeslané. A to přes protokol IMAP.

6.4 Implementace IMAP

Protokol IMAP (Internet Message Acces Protocol) byl implementován jako alternativa k přístupu přes API. Používá se pro přístup k e-mailové schránce na serveru. Je definován standardem RFC9051 [41]. Poskytují ho mnozí výrobci e-mailových řešení. To na jednu stranu vytváří možnost kooperace mezi externím e-mailovým klientem a serverem. Ale na druhou stranu to přináší implementační chyby a nedostatky na straně dodavatelů e-mail serveru. Každý to implementuje po svém a v různém rozsahu. Základní funkčnost je zajištěna vždy. Nicméně pokročilé funkce obvykle nefungují. To přináší komplikace v podobě nutnosti dodělávat požadované funkce vlastními silami. Mohou vznikat další chyby.

6.4.1 Kontrola kompatibility

Jak již bylo zmíněno v kapitole 5.9.1, ne každý IMAP server bude s aplikací kompatibilní. Tím je myšleno, že nemusí mít dodavatelem implementovány potřebné příkazy. Je zajištěna pouze základní funkčnost získávání a nahrávání e-mailů do schránky. Pro provoz aplikace je nezbytná podpora tvorby vlastních štítků, které se používají jako označení zpracovaných zpráv. Dalším potřebný příkaz je MOVE, který zajistí přesun zprávy do archivní lokace. Tento příkaz je případně možné nahradit vlastní implementací. Nejprve se původní zpráva zkopíruje do cílové lokace. Poté se původní zpráva odstraní. Poslední požadovanou funkcí je podpora UIDPLUS. Jde o unikátní identifikátor zprávy v celé e-mailové schránce. Zjednodušuje tak následnou práci se zprávami. Standardní identifikátor zprávy je platný pouze ve vybrané složce [41].



Obr. 6.7: Test kompatibility IMAP serveru

Z pohledu uživatele je téměř nemožné předem ověřit, jestli zvažovaný IMAP server podporuje potřebné příkazy. Proto byla implementována funkce, která kompatibilitu IMAP serveru ověří. Podpora tvorby vlastních štítků je zjišťována příkazem SELECT, kterým se vybírá složka ve schránce. Server na tento příkaz odpovídá mimo jiné parametrem PERMANENTFLAGS. Pokud tento parametr obsahuje znak hvězdičky, znamená to, že lze vytvářet vlastní štítky a server tak splňuje první část kontroly. Podporu příkazu MOVE lze zjistit dotazem CAPABILITY. Pokud se v odpovědi nachází MOVE, je server kompatibilní. Ve stejné odpovědi se může nacházet i parametr UIDPLUS.

Výsledek kontroly kompatibility je ukládán do konfiguračního souboru společně s adresou serveru. Kontrola se tak provádí pouze při nastavení nového serveru. Aplikace pak při každém spuštění kontroluje, jestli byl test kompatibility proveden či nikoliv. Kontrola se primárně provádí tlačítkem z klientské části. Serverová část pak pracuje pouze s výsledkem kontroly, pokud již existuje. Jestliže IMAP server kontrolou neprošel, aplikace se ukončí a neprovede žádné akce.

7 Inovace grafického rozhraní

V rámci přechodu na cloudovou platformu bylo nezbytné upravit všechny relevantní části grafického rozhraní. Největším hendikepem byl původní zdroj dat, který se využíval k načítání informací. Zdroj dat byl zajištěn skrz vyhodnocovací část aplikace. K načítání nových dat docházelo pouze při spuštění aplikace. Kompletní konfiguraci úlohy nešlo dodělat najednou. Prvně se musela vytvořit samotná úloha a poté spustit vyhodnocení. Tím se nasbírala potřebná data (vzory zpráv) a teprve poté bylo umožněno konfiguraci úlohy kompletně dokončit.

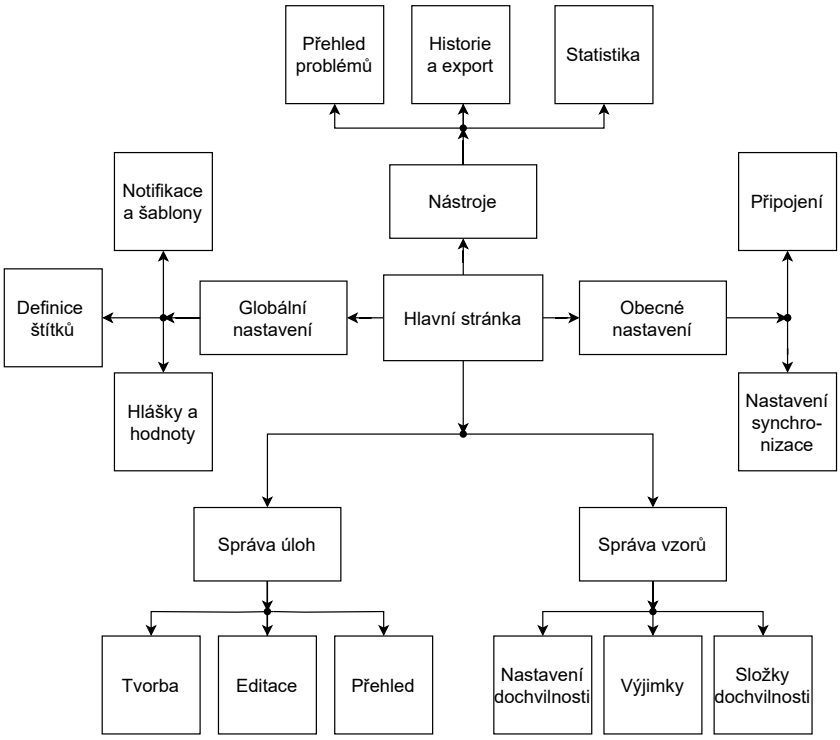
Grafické rozhraní je vytvářeno v podobě lokální webové stránky. Ve výchozím nastavení je dostupná přes http://localhost:5555. Nebude přístupná po síti. Stránky jsou vytvářeny v HTML s prvky JavaScriptu. Ten se používá hlavně pro vyskakovací notifikace a různé skrývání a odkrývání prvků stránky. Byla snaha využít JavaScript co nejméně, aby bylo zachováno co nejvíce funkcionalit stránky i přes jeho zákaz. Používá se také k tvorbě grafů statistiky.

Pro vytváření prvků stránek se využívá šablonového systému Jinja2. Systém umožňuje definovat generování tabulek a zobrazovat části stránky na základě vstupní proměnné [44]. Obsah HTML souboru se vygeneruje v aplikaci, do prohlížeče přichází kompletní. Neprobíhá žádné dynamické načítání přes JavaScript, jak je dnes moderní. Některé stránky totiž obsahují tolik JavaScriptu, že je zpomaleno načítání stránek. Objevují se zbytečné přechodové animace [45]. Přitom stejný web by šel udělat mnohem jednodušeji a bez výpočetně náročných skriptů.

7.1 Rozvržení funkcí

Po spuštění grafického rozhraní se uživateli zobrazí hlavní/úvodní stránka. Ta obsahuje odkazy na další podstránky, kde se nachází další funkce aplikace. Funkce jsou sdruženy do sekcí podle svého zaměření. Na obrázku 7.2 je vidět navržená struktura prvků. Na základě této struktury uživatel vždy ví, kde se nachází. Aktuální lokace se zobrazuje v horní liště jako zvýrazněný prvek (viz obrázek 7.1). Ten je následován dalším zvýrazněným prvkem, který signalizuje právě otevřenou stránku. Na každé stránce se nachází tlačítko „Hlavní stránka“. V grafickém rozhraní se tak nedá ztratit, vždy je možnost návratu.

Obr. 7.1: Orientační lišta



Obr. 7.2: Struktura prvků grafického rozhraní

7.2 Systém zobrazení chyb

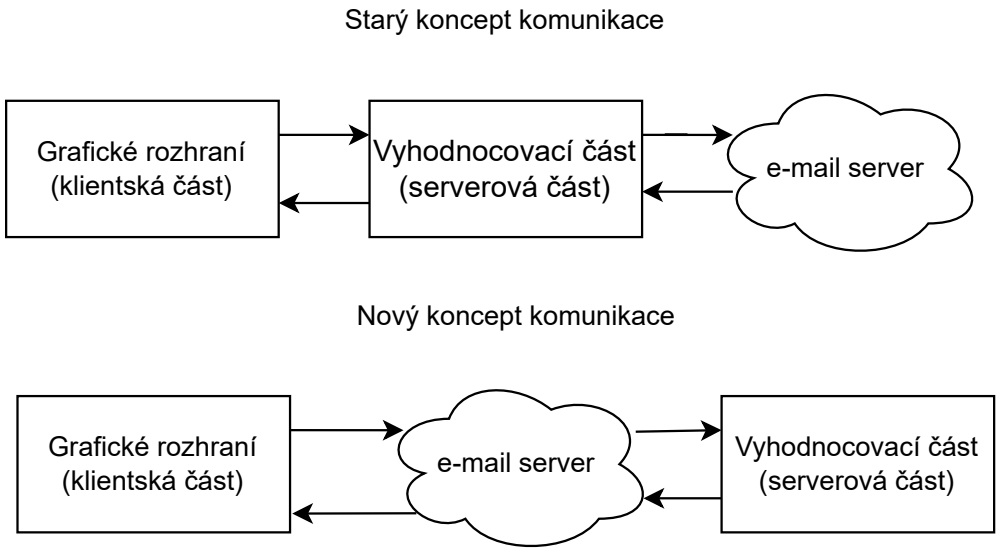
Při akcích, které uživatel provádí (ukládání konfigurace), mohou vzniknout chybové stavy. Proto byl vytvořen systém zobrazení chyb. Způsoby zobrazení chyby existují dva. První je formou oznámení, které na uživatele vyskočí po provedené akci. Nemusí obsahovat pouze chybu. Často obsahuje informaci o částečném úspěchu/neúspěchu funkce. Druhý způsob uživatele přesměruje na chybovou stránku (viz obrázek 7.3), kde je červenou barvou chyba zvýrazněna. Uživateli se zobrazí důvod vzniku chyby, případně kde chyba vznikla. Ve spodní části se nachází tlačítko „Jít zpět“. To uživatele vhodně přesměruje na předchozí stránku. Pokud to není možné (povaha chyby to znemožňuje), dochází k přesměrování na hlavní stránku.

7.3 Zdroj dat

Koncept získávání dat ke konfiguraci úloh, anebo k zobrazení uživateli, byl kompletně přepracován. Grafické rozhraní si nyní data získává samostatně přímo ze zdroje. Tím je myšleno přímé dotazování na API MS Graph nebo protokol IMAP. Zcela tím odpadá provádění mezikroků při konfiguraci úloh a bylo dosaženo osamostatnění grafického rozhraní. Nyní již nevyžaduje žádný přímý kontakt s vyhodno-

Obr. 7.3: Chybová stránka

covací částí (nový koncept na obrázku 7.4). Jakmile uživatel otevře funkci/nástroj, který data vyžaduje, dojde k načtení aktuálních dat. Celkově touto úpravou došlo ke zvýšení komfortu při používání aplikace. A hlavně se data načtou za pár sekund. Dříve by to trvalo podstatně déle. Toho je docíleno zvýšeným využitím kešovaných informací. Například je nyní ukládán identifikátor složek, který se dříve vyhledával při každém požadavku na data. Došlo k optimalizaci dotazů na API. Nově jsou dotazy seskupovány do skupin po maximálně dvaceti dotazech. Ty jsou odeslány najednou. Odpověď se také vrátí jenom jedna. Velmi to zrychluje načítání dat, jelikož jeden dotaz na API (skupinový nebo samostatný) trvá několik milisekund. Což se při desítkách dotazů nasčítá do vysokých hodnot (sekundy).



Obr. 7.4: Napojení zdroje dat

7.4 Přehled problémů

Uživatel je s problémy primárně seznámen přes notifikační e-mail, který obdrží po každém spuštění aplikace. Jako problém se označuje každá situace, kdy e-mailová zpráva od monitorovaného zařízení přijde jinak, než měla. Zpráva může být označena jako zpožděná, nepřečtená, chybějící nebo výjimka. Notifikace se však postupně nahromadí, což vytváří nutnost analyzovat problém v několika notifikačních zprávách. Stránka „Přehled problémů“ tuto situaci efektivně řeší. Na jednom místě zobrazí v přehledové tabulce všechny úlohy a jejich složky (obrázek 7.5). Na první pohled je vidět, jestli někde vznikl problém. To je signalizováno číselnou hodnotou a pro rychlejší identifikaci i barevně odlišeno. Zelená barva značí, že je vše v pořádku. Zatímco červená barva označuje problém.

Pokud nelze načíst stav složky, je to znázorněno šedou barvou a tlačítko je deaktivováno. Nenačtený identifikátor složky je označen písmenem „x“. Jiná chyba pak znakem „-“. Ani na jednu takto označenou hodnotu nelze kliknout.

Úloha	Složka	Nepřečtených	Zpožděných	Chybějících	Výjimek
CT.txt	i	0	5	0	0
LinAPI.txt	i	0	5	1158	0
LinIMAP.txt	iIMAP	x	x	x	x

Obr. 7.5: Přehled problémů

Po kliknutí na červeně zabarvenou číselnou hodnotu se zobrazí detail problému. Respektive se vypíšou zprávy, které jsou jako problémové označeny. Jakmile správce považuje problém za vyřešený, může vybrané zprávy označit a stisknout tlačítko vyřešit. Tímto se na označené zprávy nastaví štítek „vyřešeno“ a již se v přehledu problémů nezobrazí. Po kliknutí na šipku v prvním řádku tabulky se data seřadí.

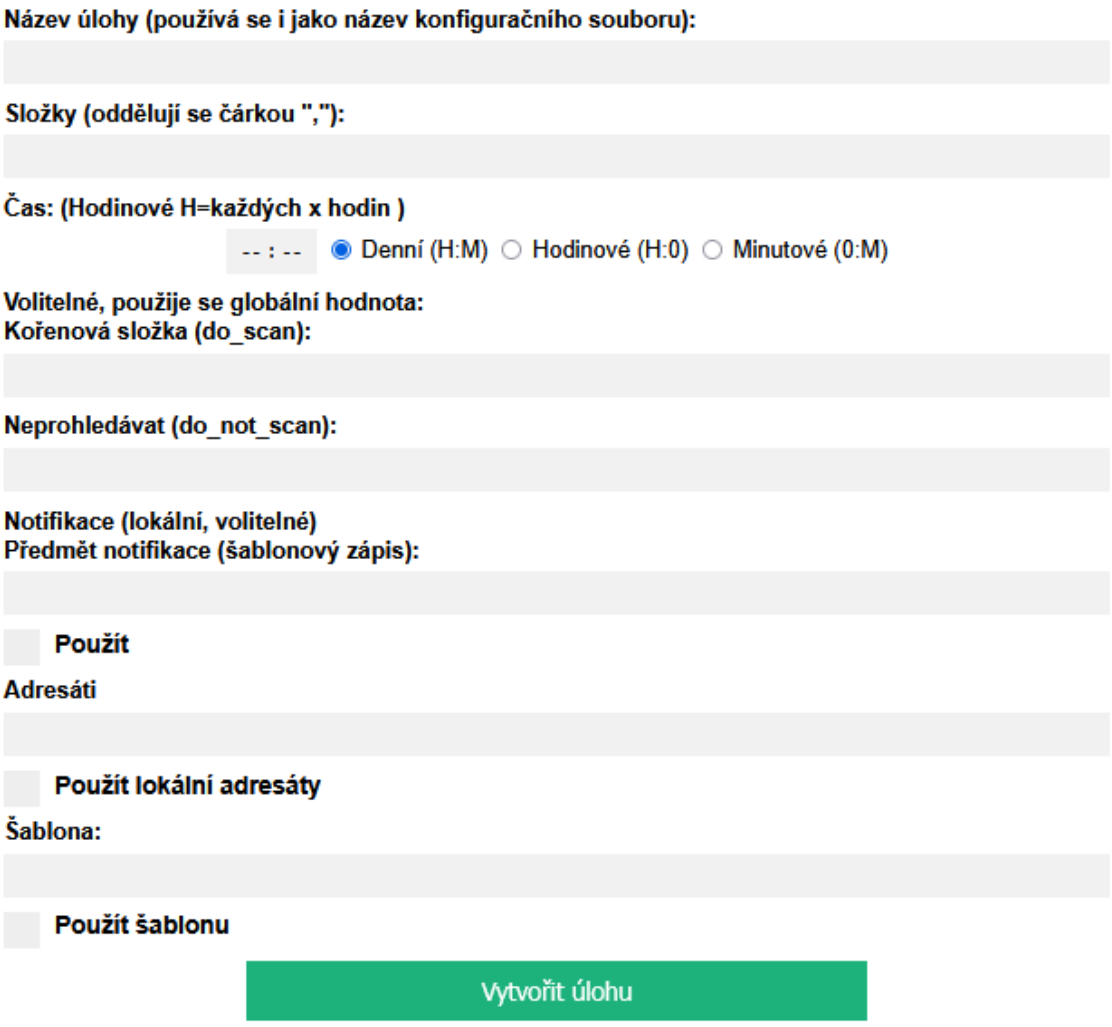
↑ Přišlo ↓	↑ Odesílatel ↓	↑ Předmět ↓	Flagy	Tlačítko	Vybrat
13-05-2025 22:20:39	J @ onmicrosoft.com	"TEST"znaků/*-+#&@{}+ěščřžžáí=,.'	['* 10-minutly', 'ZPOŽDĚNÁ']	Vyřešit	☐
12-05-2025 11:06:25	J @ onmicrosoft.com	"TEST"znaků/*-+#&@{}+ěščřžžáí=,.'	['* 10-minutly', 'ZPOŽDĚNÁ']	Vyřešit	☐
09-05-2025 22:41:54	J @ onmicrosoft.com	"TEST"znaků/*-+#&@{}+ěščřžžáí=,.'	['* 10-minutly', 'ZPOŽDĚNÁ']	Vyřešit	☐
09-05-2025 22:26:59	J @ onmicrosoft.com	"TEST"znaků/*-+#&@{}+ěščřžžáí=,.'	['* 10-minutly', 'ZPOŽDĚNÁ']	Vyřešit	☐
09-05-2025 22:16:19	J @ onmicrosoft.com	"TEST"znaků/*-+#&@{}+ěščřžžáí=,.'	['* 10-minutly', 'ZPOŽDĚNÁ']	Vyřešit	☐

Obr. 7.6: Detail problému

7.5 Tvorba a správa úloh

Správa úloh byla rozšířena na OS Linux, který je využíván při převodu na cloud. Stávající stránka „Tvorba úloh“ byla přizpůsobena, aby šlo definovat časy plánovaného spouštění pro crontab. Na výběr jsou tři druhy spouštění. První je denní, spouští se každý den v nastavený čas. Zadávají se obě hodnoty (H a M). Další je hodinové spuštění. První hodnota „H“ definuje počet hodin, po kterých se aplikace spustí. Druhá hodnota „M“ pak zpřesňuje čas. Pokud uživatel nastaví 02:03, bude úloha spouštěna každé dvě hodiny a tři minuty (00:03, 02:03). Poslední je minutové spouštění. To bere v potaz jen hodnotu „M“, do hodnoty „H“ se zadá nula.

Po kliknutí na „Vytvořit úlohu“ se nejprve vytvoří konfigurační soubory. Ty jsou uloženy na lokální disk klientské části. V případě úspěchu se úloha a konfigurační soubory přenesou do servisní složky e-mialové schránky. Tam čekají, až si je serverová část stáhne a uloží na svůj disk.

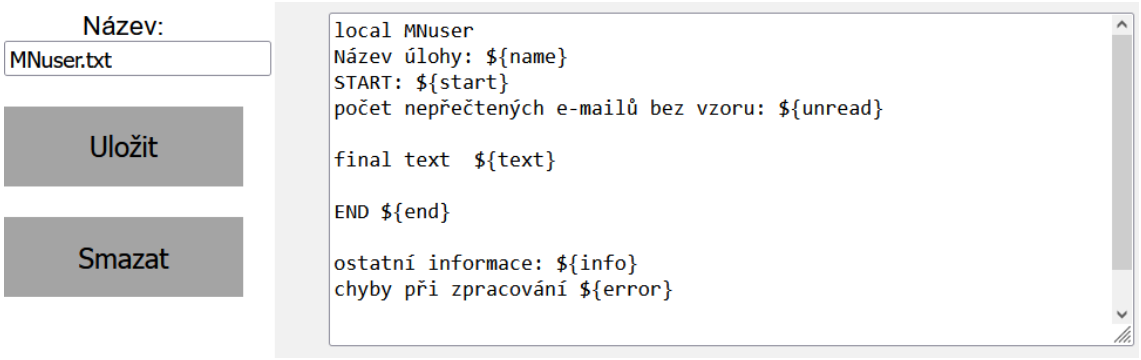


Obr. 7.7: Vytvoření úlohy

Při úpravě úlohy se soubory opět nahrají do servisní složky. Soubory jsou nahrávány jako příloha e-mailu. Předmět zprávy pak obsahuje název souboru, který se musí shodovat s názvem přílohy. Podmínkou je povolená funkce synchronizace. Pokud není zapnuta, dojde pouze k vytvoření konfiguračních souborů na lokální disk. Nejsou přenášeny na server.

7.6 Editor šablon

Aplikace nově přidává systém šablon do notifikačních zpráv. Pro zvýšení uživatelského zážitku byl do grafického rozhraní přidán editor šablon. Přes něj se dá provádět kompletní správa všech existujících šablon. Po otevření editoru je nejprve zobrazena úvodní stránka se seznamem existujících šablon. Ty lze upravit odstranit nebo vytvořit. Způsob editace šablony je jednoduchý. Tak jak uživatel napíše text do šablony, tak se zobrazí v notifikaci. V pravé části stránky se nachází návod, který popisuje všechny dostupné proměnné hodnoty a jak je vložit.



Obr. 7.8: Editor notifikačních šablon

7.7 Zadávání výjimek

Zadávání výjimek je společné pro všechny úlohy (globální). Po otevření stránky se zobrazí přehledová tabulka (obrázek 7.9), která obsahuje všechny vytvořené výjimky. Na první pohled je možné zkontrolovat nastavení všech výjimek. V pravé části tabulky se nachází editační tlačítko, kterým probíhá úprava šablony výjimky. V levé části (viz obrázek 7.10) jsou situovány ovládací prvky, které směřují do dalších konfiguračních podstránek.

V sekci „Závažnost“ se nastavuje znění štítků, které označují závažnost. Úrovně závažnosti existují celkem čtyři (popsáno v kapitole 5.7). Sekce „Akce“ pak nabízí nastavení akcí, které se spouští při detekci výjimky. Ke každé akci je možné uložit

Název	Výraz	Typ	Závažnost	Akce	Označení	Odesílatel	Povolené úlohy		
testINT	vyr	výraz kdekoliv	low	žádná	štítek1	sender@sender.sender	MNuser.txt	Editovat	Smazat
asdsa	dasd	výraz kdekoliv		žádná				Editovat	Smazat

Obr. 7.9: Přehled výjimek (přiblížené)

čtyři hodnoty. Pokud je akce spuštěna, má je k dispozici. Do první hodnoty se ukládá třeba adresát, kterému je zaslána mimořádná notifikace. Uživatel si tam dá co potřebuje, jedná se o přípravu k tvorbě vlastních akcí. Připravené akce jsou pojaty jako experimentální a nemusí nutně fungovat. Zaručeně bude fungovat pouze „Odeslání mimořádné notifikace“. Tato akce byla otestována a je funkční.

Při zakládání nové výjimky je nutné zadat název, hledaný výraz a způsob zpracování výrazu. Ostatní parametry jsou plně volitelné a nemusí se vyplnit. Šablony výjimek jsou ukládány ve vyhrazeném souboru typu JSON. Ruční editace tak není doporučena. Mohlo by dojít k poškození souboru, respektive rozhození formátu.

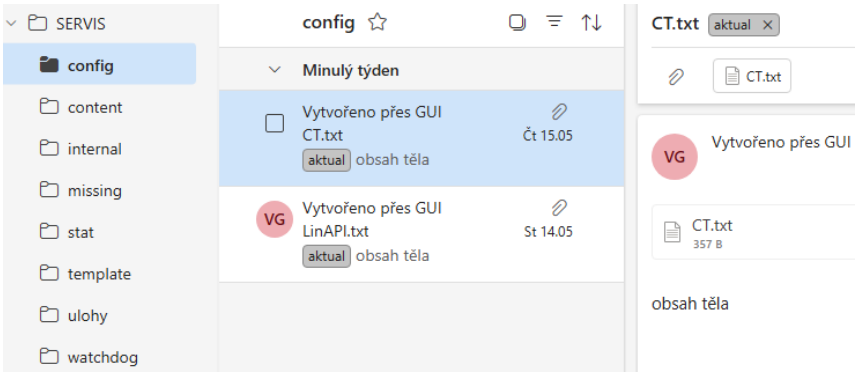
Obr. 7.10: Založení nové výjimky

7.8 Synchronizace dat

Vytváření konfiguračních souborů probíhá výhradně přes grafické rozhraní. To je spouštěno na počítači uživatele a tak se soubory ukládají pouze na lokální disk. Pro přenos do serverové části v cloudu proto byla vytvořena synchronizační funkce. Synchronizaci lze vypnout nebo zapnout na stránce „Nastavení synchronizace“. Stejná stránka obsahuje dvě tlačítka. První stáhne všechny soubory ze servisní složky (obrázek 7.11) na lokální disk uživatele. Druhé tlačítko naopak nahraje obsah složek z lokálního disku do servisní složky e-mailu. Tyto tlačítka jsou využívána pro ruční synchronizaci konfiguračních souborů. Respektive ke sjednocení stavu lokálního disku a vzdáleného úložiště (cloudové prostředí).

Samotná synchronizace je pojata částečně jednostranně. Grafické rozhraní soubory primárně nahrává. Při vytvoření úlohy přes grafické rozhraní dojde k nahrání všech konfiguračních souborů a definičního e-mailu úlohy. Stahují se pouze soubory s chybějícími zprávami a statistikou. Obsah složek tak nemusí být na všech místech úplně stejný. Opravit to lze ruční synchronizací viz text výše.

Serverová část zase soubory primárně stahuje. Po spuštění ověří existenci nových souborů, které stáhne a případně zavede novou úlohu do plánovače úloh. Nahrává pouze soubor s chybějícími zprávami a případně se statistikou.



Obr. 7.11: Servisní složka a servisní zpráva

Servisní zprávy jsou označeny provozními štítky, které určují, co se zprávou dělat. Štítek „new“ nebo „updated“ značí nový nebo upravený soubor. Při příští kontrole servisní složky bude stáhnut. Další štítek „aktual“ označuje zpracovanou zprávu a tak bude přeskočena. Pokud je třeba soubor smazat, nastaví se štítek „delete“. Vymazání provádí serverová část, která zároveň smaže soubor ze svého disku. Pro testovací účely ještě existuje štítek „deleted“. Ten označuje zprávu, která měla být smazána, ale pro účel testu byla zachována. Zprávy bez štítku jsou ignorovány.

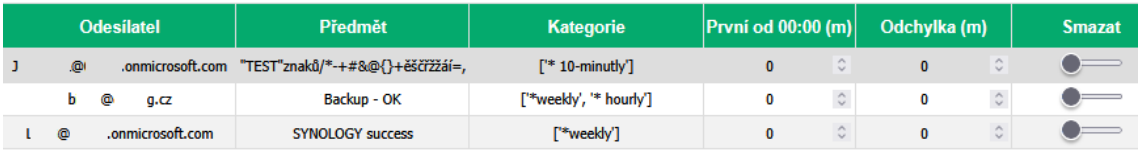
7.9 Definice kategorií

Kategorie neboli štítky, se používají pro označení zpráv a další interní funkce. Definují chování aplikace. Kategorie „* 10-minutly“, která je vidět na obrázku 7.12, představuje nastavení rozestupu zprávy. V tomto případě má zpráva chodit každých deset minut (00:00, 00:10). Aby byl štítek využit modulem dochvilnosti, musí být zapnutý. Zároveň na zprávě může být jen jeden zapnutý štítek. Jak je vidět na obrázku 7.12, štítek „ZPOŽDĚNÁ“ je vypnut a tak může být na zprávě (obrázek 6.4) nastaven. V případě výskytu dvou a více zapnutých štítků na zprávě se dochvilnost nevyhodnotí. Uživatel proto musí počátečnímu nastavení věnovat čas.

Obr. 7.12: Definice kategorií

7.10 Nastavení vzorů

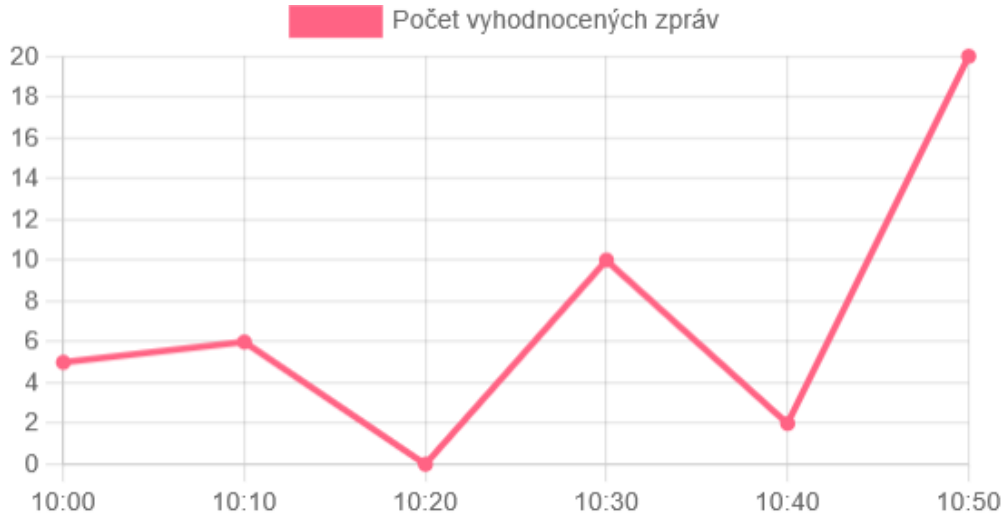
Jedná se o další součást konfigurace vyhodnocení dochvilnosti zpráv. Vzory se načtou ze zdroje dat a pak se zobrazí v tabulce na obrázku 7.13. Parametr „První od 00:00(m)“ vyjadřuje, za jak dlouho po půlnoci přijde první zpráva od monitorovaného zařízení. Jestliže první zpráva přijde v 00:01, nastaví se „1“. Pro zprávy, které chodí jednou denně se nastavuje čas očekávaného přijetí (19:43 => 1943). Z důvodu limitace systému se od času ubírají počáteční nuly. Pokud zprávy chodí každý den v 00:05, nastavím „5“.



Obr. 7.13: Nastavení vzorů

7.11 Statistika

Modul statistiky zobrazuje sesbírané hodnoty v podobě grafu. Zobrazuje se počet vyhodnocených zpráv po každém spuštění aplikace. Ta je totiž spouštěna periodicky, a při každém běhu uloží potřebná data. Na první pohled je vidět, kolik zpráv bylo v každém běhu vyhodnoceno. Je možné zvolit různé časové období a porovnat, jestli nedošlo k nárustu či poklesu počtu zpráv. Ukládá se historie posledních tisíc spuštění. Starší data se smažou.



Obr. 7.14: Graf vyhodnocených zpráv dle spuštění

7.12 Spuštění

Jak již bylo uvedeno v úvodu kapitoly 7, grafické rozhraní je vytvořeno formou lokální webové stránky (není dostupná mimo místní počítač). Ve výchozím nastavení je dostupné přes adresu http://localhost:5555.

Pro OS Windows je nachystán spustitelný exe soubor. Po otevření souboru se zobrazí konzolové okno, které slouží jako web server. Vypisuje různé stavové informace. Jakmile se web server spustí, je automaticky otevřen webový prohlížeč s grafickým rozhraním. Není třeba žádné instalace. Jakmile je okno s konzolí zavřeno, vypne se i web server. Záměrně neběží na pozadí.

Pro OS Linux slouží „.py“ soubory. Vyžaduje se instalace externích knihoven. Zprovoznění není triviální, vyžaduje zkušenosti. Návod k zprovoznění na OS Linux nebyl vytvořen. Zkušený uživatel by si měl poradit, není to určeno pro začátečníky. Místo exe souboru se spustí py soubor. Názvy jsou mezi OS sjednoceny.

8 Zprovoznění aplikace

Aplikace byla navržena jako multiplatformní. Lze ji spustit na všech systémech, které podporují jazyk Python verze 3.7 a vyšší. Způsob zprovoznění závisí na vybrané platformě. Tato kapitola popisuje pouze některé. Nicméně uvedené postupy lze přizpůsobit na všechny. Primárně bude aplikace distribuována formou „py“ souborů. Ty jsou zároveň použity při nasazení aplikace na cloudovou platformu.

8.1 Skripty

Dohledový systém si klade za cíl jednoduché zprovoznění. Dle návodu ho musí umět zprovoznit i začínající IT pracovník. Kroků, které jsou potřeba ke kompletnímu zprovoznění systému je celá řada. Vypisování všech příkazů do návodu by bylo zdlouhavé. Navíc není řešena reakce na chybové stavy, se kterými není pracovník seznámen. Z toho důvodu jsou nachystány skripty, které činnost automatizují a reagují na vybrané chyby. Ať už jde o instalaci nebo aktualizaci dohledového systému.

8.1.1 Instalační skript

Pro automatickou instalaci aplikace jedním krokem byl vytvořen instalační skript. Ten předpokládá, že má uživatel zprovozněný podkladový systém Linux. Vytváření obrazu disku by bylo zbytečné, časem by zastaral. Tento přístup je vhodný pro svou univerzálnost. Uživatel se může rozhodnout, jestli využije cloudové platformy, anebo má svůj vlastní aplikační server.

Skript vytvoří potřebnou adresářovou strukturu a nakopíruje provozní soubory aplikace. Dále doinstaluje potřebný software a nachystá virtuální prostředí Python pro první spuštění aplikace. Zkontroluje nastavenou časovou zónu, některé instalace nemají nastavenou správnou. Chybně nastavená časová zóna by ovlivnila vyhodnocení dochvilnosti zpráv. Systémový čas se používá jako zarážka (konec generování časových slotů).

Nicméně stále bude nutný zásah uživatele. Ten musí nachystat konfigurační a autentizační soubory přes grafické rozhraní. Soubory následně ručně nahraje do připravené složky dle návodu.

Příkazy jsou poskládány tak, aby se v případě neúspěchu předchozího kroku skript ukončil. Některé příkazy jsou závislé na kroku předchozím. Například virtuální prostředí Python nelze vytvořit pokud není instalováno. Jestliže selže instalace „python-venv“, je nesmysl spouštět instalaci knihoven. Příkaz by stejně selhal.

8.1.2 Spouštěcí skript

Instalačním skriptem bylo vytvořeno virtuální prostředí Python s nainstalovanými knihovnami. Pokud se pracovník pokusí spustit dohledový systém ve výchozím Python prostředí, dojde k chybě. Neobsahuje totiž potřebné knihovny.

Vytvořené virtuální Python prostředí není po zapnutí OS aktivní. K jeho aktivaci je potřeba znalost cesty k prostředí. Což začínající IT pracovník nemusí znát. Spouštěcí skript zjistí cestu a uloží si ji do proměnné. Následně se pracovníka dotáže na název konfiguračního souboru, který plánuje použít, pokud nebyl zadán dříve. Název souboru lze totiž zadat přímo při spuštění skriptu jako argument „run.sh nazev.txt“. Název si uloží do další proměnné a provede aktivaci virtuálního Python prostředí. Jakmile je aktivováno, spustí dohledový systém. Po ukončení běhu opět prostředí deaktivuje.

Obr. 8.1: Spouštěcí skript - dotaz

8.2 Vlastní server

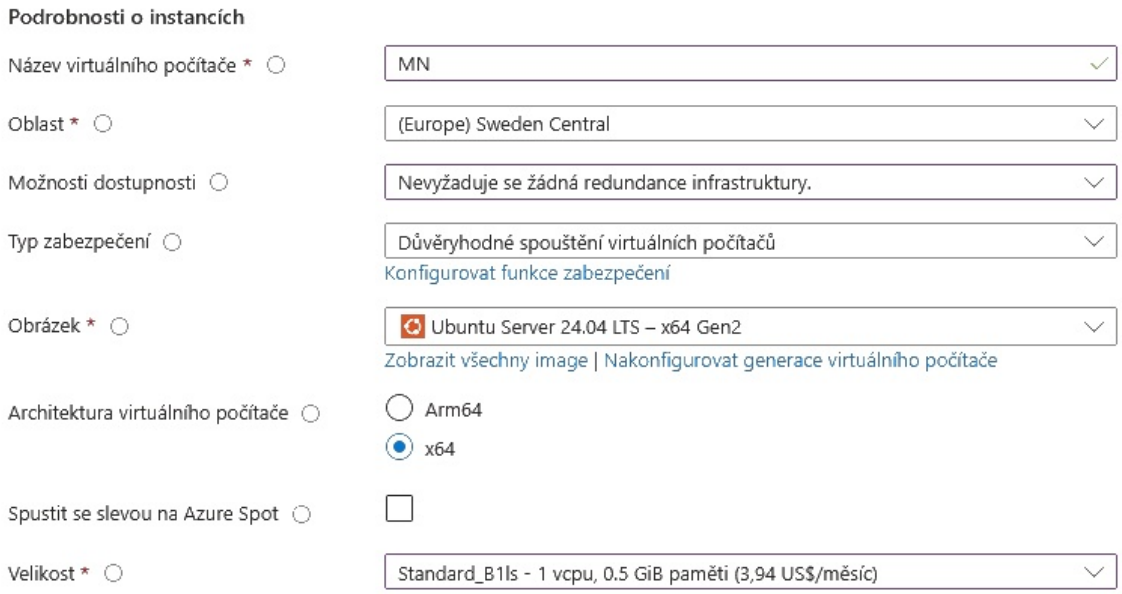
Zprovoznění na vlastním serveru je vhodné pro uživatele, kteří nechtějí využít cloudové platformy. Samotná instalace podkladového systému nebyla součástí řešení. To si uživatel zajistí samostatně. Doporučený a testovaný OS je Ubuntu Server ve verzi minimal. Vyzkoušené minimální požadavky na provoz systému jsou jedno jádro procesoru a 512 MB operační paměti RAM. Nicméně lze alespoň zpočátku doporučit přiřadit více paměti, aby nebyl ovlivněn aktualizační a instalační proces. Po ukončení zprovozňování je možné přiřazenou paměť RAM opět snížit na 512 MB.

Jakmile bude systém připraven, musí se nakopírovat instalační skript do domácí složky vzdáleného uživatele. Poté uživatel skript spustí. Ten nainstaluje vše potřebné. Pak se čeká na uživatele, až nakopíruje konfigurační soubory. Následně se spouštěcím skriptem provede prvotní spuštění. Tím se zavede první úloha do plánovače úloh (cron). Všechny plánované úlohy lze vypsat příkazem „crontab -l“. Editace plánovaných úloh pak probíhá příkazem „crontab -e“.

8.3 Cloudové prostředí

Na základě preference zadavatele byla vybrána platforma Microsoft Azure. Proto bude postup zprovoznění blíže popsán právě na této platformě. Nicméně není vyloučeno zprovoznění u konkurenčního poskytovatele. Jediný rozdíl spočívá ve způsobu vytvoření virtuálního počítače a instalace podkladového operačního systému. Ostatní kroky budou koncipovány obecně a půjde je použít i jinde.

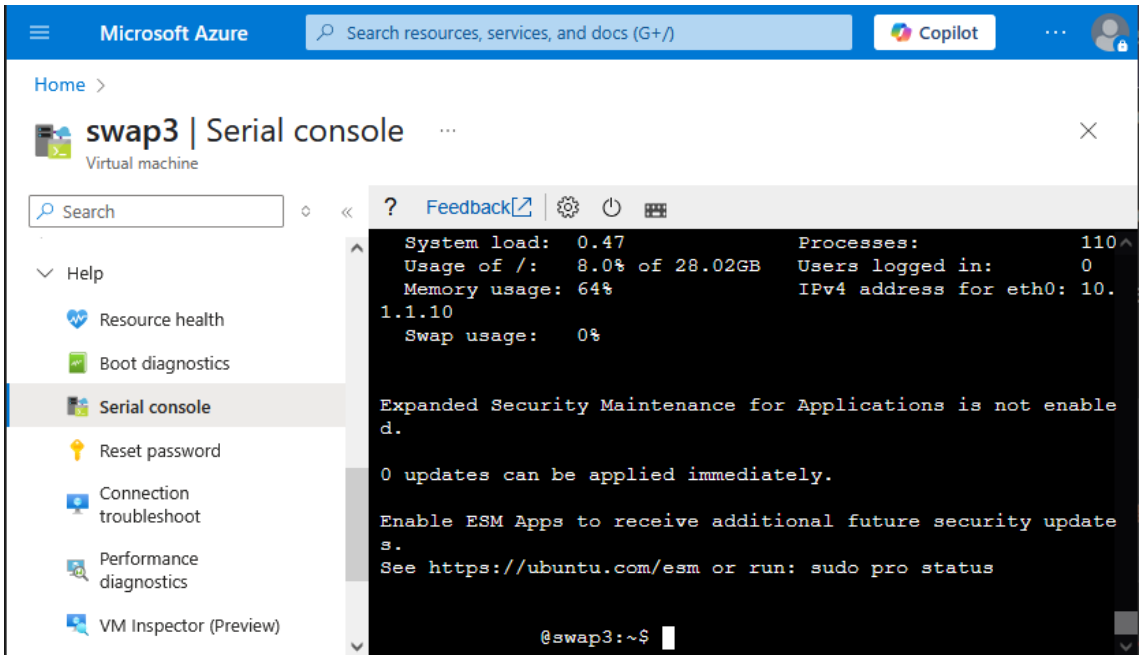
Prvním krokem je založení virtuálního počítače. Nejlevnější tarif je „B1ls“, který parametrově plně postačuje. Následuje výběr operačního systému. Platforma nabízí připravené obrázky (viz obrázek 8.2) operačních systémů, které jsou přizpůsobeny pro provoz v cloudu. Obsahují různé monitorovací funkce, které hlásí stav na ovládací panel. Vybereme obrázek Ubuntu Server nejnovější verze. Nasazení (vytváření VM) probíhá automaticky. Dále se nastavuje požadované úložiště. Z úsporných důvodů byl vybrán typ „HDD standard“ s nejnižší možnou kapacitou 30 GiB. Virtuální počítač má pouze 512 MB operační paměti RAM, což zabraňuje aktualizaci OS. Občas to spadlo a nebylo možné dokončit instalaci aktualizací. Proto se musí povolit swapování na dočasný disk, který je ve VM přítomen. Při vytváření VM je možné definovat parametry cloud-init, které slouží pro nastavení automatické instalace. Do pole „vlastní data“ se vloží připravená sestava příkazů. Ta zajistí konfiguraci swapu zároveň s vytvářením VM.



Obr. 8.2: Vytváření virtuálního počítače v Microsoft Azure

Jakmile se vytvoří virtuální počítač, musí se uživatel do systému přihlásit. To lze udělat přes protokol SSH nebo přes virtuální konzoli, která je dostupná přímo přes webové rozhraní. V dalším kroku je nutné zkopírovat instalační skript aplikace společně s archivem do kořenové složky uživatele. Uživatel spustí skript. Ten nastaví potřebné systémové prostředí, změní časovou zónu na „Europe/Prague“. Dále instaluje knihovny jazyka Python a rozbalí archiv se soubory aplikace. Až skript úspěšně doběhne, je vše připraveno. Uživatel přes grafické rozhraní vytvoří prvotní úlohu a provede provede přihlášení k e-mailové schránce. Poté přes protokol SFTP nakopíruje prvotní konfigurační soubory do složky s aplikací. V posledním kroku aplikaci ručně spustí. Dojde k zavedení plánovaných úloh. Poté již spouštění probíhá automaticky. Nové konfigurační soubory se stáhnou při dalším spuštění.

Virtuálnímu počítači je možné přiřadit veřejnou IPv4 adresu. Ta ale není pro provoz aplikace nutná. Může být vyžadována pouze pro nahrání konfiguračních souborů při prvotním zprovoznění. Pokud tedy provozovatel tuto cloudovou platformu již využívá, nemusí zbytečně platit za využití veřejné IPv4 adresy. Po zkopírování souborů je možné ji odstranit. Přístup k administraci VM zůstane přes webovou konzoli (viz obrázek 8.3). Adresy IPv6 nejsou bez další konfigurace dostupné [46].



Obr. 8.3: Webová virtuální konzole

Závěr

Diplomová práce se zabývala úpravou existujícího dohledového systému. V rámci úprav došlo k inovaci některých částí a zároveň byly přidány zcela nové funkce.

Nejprve proběhla analýza původního řešení, aby bylo zřejmé, že lze požadované změny implementovat. Nebyla zjištěna žádná překážka, která by bránila dalšímu rozvoji aplikace. Modulový systém je vyhovující.

Dále proběhla analýza vybraných cloudových platforem. Tedy jaké jsou dostupné technologie a způsoby provozu vlastních aplikací v cloudovém prostředí. Na základě analýzy a preference zadavatele byla vybrána platforma Microsoft Azure. Proběhlo srovnání s konkurenčními produkty. Na základě srovnávací tabulky 3.1 lze konstatovat, že rozdíly mezi produkty nejsou podstatné. Parametrově vyhovují všechny.

Nejpodstatnější změnou bylo přidání podpory pro protokol IMAP. Užitná hodnota dohledového systému se tím razantně zvýšila. Aplikace již není vázána na konkrétního poskytovatele e-mail serveru.

Modul vyhodnocení zpráv byl značně inovován. Byl přidán systém detekce výjimek, který zachycuje opakované chybové stavy. Funkce vyhodnocení dochvilnosti byla upravena tak, aby informovala o chybějících zprávách. Nově se zprávy vyhodnocují i na základě obsahu. Hledají se klíčová slova.

Notifikace pro správce využívají nový šablonový systém. Uživatel si může nadefinovat podobu notifikace, případně vynechat nepotřebné informace. Hlášky, které způsobovaly nepřehlednost byly revidovány.

Grafické rozhraní se také dočkalo nových funkcí. Zásadní novou funkcí z pohledu uživatele je „Přehled problémů“. Ta zobrazuje stav složky. Pokud obsahuje problémové zprávy, je to přehledně zobrazeno číselnou hodnotou a také barevně odlišeno. Dále proběhly interní změny, které uživatel nevidí. Byl nahrazen původní zdroj dat, který byl pomalý. Nový zdroj je rychlejší a data načítá přímo z e-mailového serveru. Přidání statistiky bylo pojato přidáním dvou grafů, které zobrazují počty zpracovaných a chybových zpráv za daný časový úsek.

Se zprovozněním dohledového systému pomohou instalační skripty. Ty zajistí přípravu prostředí na OS Linux. Uživatel se může rozhodnout, jestli si systém nainstaluje na vlastní server nebo cloudovou platformu.

Literatura

[1] *IT monitoring — co to je a jak vám dokáže ušetřit náklady.* Online. Dostupné z: https://www.totalservice.cz/novinky/it-monitoring-co-to-je-a-jak-vam-dokaze-usetrit-naklady-2024-05-09/. [cit. 2025-05-22].

[2] *Kybernetická bezpečnost — III. Kybernetický bezpečností incident.* Online. Dostupné z: https://lepsi-reseni.cz/ochrana-osobnich-udaju-gdpr/kyberneticka-bezpecnost-iii-kyberneticky-bezpecnosti-incident/. [cit. 2025-05-22].

[3] *SolarWinds hack explained: Everything you need to know.* Online. 2023. Dostupné z: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know. [cit. 2025-05-22].

[4] BARAK, Zack. *SIEM Pricing: 4 Licensing Models & 4 Ways to Cut Your Costs.* Online. Dostupné z: https://coralogix.com/guides/siem/siem-pricing/. [cit. 2025-05-22].

[5] CHOUDHARY, Meghraj. *What is Infrastructure Monitoring?* Online. 2025. Dostupné z: https://middleware.io/blog/what-is-infrastructure-monitoring/. [cit. 2025-05-22].

[6] *What is infrastructure monitoring?* Online. Dostupné z: https://www.ibm.com/think/topics/infrastructure-monitoring. [cit. 2025-05-22].

[7] *Notification.* Online. 2025. Dostupné z: https://kb.synology.com/en-br/DSM/help/DSM/AdminCenter/system_notification_desc?version=7. [cit. 2025-05-22].

[8] CASE, J a FEDOR, M. A Simple Network Management Protocol (SNMP). Online. 1990. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1157. [cit. 2025-05-22].

[9] *SNMP - Simple Network Management Protocol.* Online. Dostupné z: https://www.thousandeyes.com/learning/techtorials/snmp-simple-network-management-protocol. [cit. 2025-05-22].

[10] *What is NetFlow?* Online. Dostupné z: https://www.ibm.com/think/topics/netflow. [cit. 2025-05-22].

[11] KUNČICKÝ, Daniel. *Automatizovaný dohledový systém pro IT infrastruktury*. Online, Bakalářská. Brno: Vysoké učení technické v Brně, 2023. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=255675. [cit. 2025-05-22].

[12] FIEDLER, Mike. *Safety and Security Engineer: First Year in Review*. Online. In: Blog.pypi.org. 2024. Dostupné z: https://blog.pypi.org/posts/2024-08-16-safety-and-security-engineer-year-in-review/. [cit. 2024-11-17].

[13] ABRAMOVSKY, Ori. *PyPI Inundated by Malicious Typosquatting Campaign*. Online. Check Point. 2024. Dostupné z: https://blog.checkpoint.com/securing-the-cloud/pypi-inundated-by-malicious-typosquatting-campaign/. [cit. 2024-11-17].

[14] FIEDLER, Mike. *Incident Report: User Account Takeover*. Online. In: Blog.pypi.org. 2023. Dostupné z: https://blog.pypi.org/posts/2023-12-04-account-takeover/. [cit. 2024-11-17].

[15] TOULAS, Bill. *Malicious PyPI package with 37,000 downloads steals AWS keys*. Online. In: Bleepingcomputer. 2024. Dostupné z: https://www.bleepingcomputer.com/news/security/malicious-pypi-package-with-37-000-downloads-steals-aws-keys/. [cit. 2024-11-17].

[16] MONTALBANO, Elizabeth. *Cyberattackers Hide Infostealers in YouTube Comments, Google Search Results*. Online. 2025. Dostupné z: https://www.darkreading.com/threat-intelligence/cyberattackers-infostealers-youtube-comments-google-search. [cit. 2025-05-22].

[17] *What is a supply chain attack?* Online. Cloudflare. Dostupné z: https://www.cloudflare.com/learning/security/what-is-a-supply-chain-attack/. [cit. 2024-11-17].

[18] *Windows Virtual Machines Pricing*. Online. Dostupné z: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows/#pricing. [cit. 2025-05-21].

[19] LOBOTKA, Andrej. *7 hlavních GDPR pravidel při využívání cloudu*. Online. 25.3.2024. Dostupné z: https://www.gdpr.cz/7-hlavnich-gdpr-pravidel-pri-vyuzivani-cloudu. [cit. 2025-05-21].

[20] *Najít partnera Azure*. Online. Dostupné z: https://azure.microsoft.com/cs-cz/partners/. [cit. 2025-05-21].

[21] *Co je autorizace poskytovatele cloudových řešení?* Online. 2025. Dostupné z: https://learn.microsoft.com/cs-cz/partner-center/enroll/csp-overview. [cit. 2025-05-21].

[22] *Hosting applications on Azure*. Online. Dostupné z: https://learn.microsoft.com/en-us/azure/developer/intro/hosting-apps-on-azure. [cit. 2025-05-13].

[23] KUNERT, Paul. *Microsoft Azure faceplants in Norway, taking government services with it*. Online. In: . 2025. Dostupné z: https://www.theregister.com/2025/02/20/microsoft_azure_outage_norway/. [cit. 2025-05-21].

[24] TUSJAK, Štefan. *Plánujete přechod na cloud? Zvažte výhody a potenciální rizika*. Online. 2024. Dostupné z: https://ittrendy.cz/clanky/planujete-prechod-na-cloud-zvazte-vyhody-a-potencialni-rizika. [cit. 2025-05-21].

[25] *Benefits of cloud migration*. Online. 2025. Dostupné z: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/benefits-of-cloud-migration. [cit. 2025-05-21].

[26] *What happened in the Ubiquiti data breach?* Online. 2024. Dostupné z: https://www.twingate.com/blog/tips/ubiquiti-data-breach. [cit. 2025-05-21].

[27] NARAINE, Ryan. *Crash Dump Error: How a Chinese Espionage Group Exploited Microsoft-s Mistakes*. Online. 2023. Dostupné z: https://www.securityweek.com/crash-dump-error-how-a-chinese-espionage-group-exploited-microsofts-errors/. [cit. 2025-05-21].

[28] *Why your IT department is incompetent*. Online. 2007. Dostupné z: https://pietersz.co.uk/2007/03/it-incompetence. [cit. 2025-05-21].

[29] *Pricing calculator*. Online. Dostupné z: https://azure.microsoft.com/en-us/pricing/calculator/?service=managed-disks. [cit. 2025-05-13].

[30] *What is Azure Functions?* Online. 2025. Dostupné z: https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview. [cit. 2025-05-21].

[31] *What is vendor lock-in? | Vendor lock-in and cloud computing*. Online. Dostupné z: https://www.cloudflare.com/learning/cloud/what-is-vendor-lock-in/. [cit. 2025-05-21].

[32] *Google Cloud*. Online. Dostupné z: https://cloud.google.com/pricing. [cit. 2025-05-13].

[33] *Python Anywhere*. Online. Dostupné z: https://www.pythonanywhere.com/. [cit. 2025-05-13].

[34] *Clever Cloud*. Online. Dostupné z: https://www.clever-cloud.com/product/python-applications/. [cit. 2025-05-13].

[35] *Forpsi Cloud*. Online. Dostupné z: https://www.forpsicloud.cz/. [cit. 2025-05-14].

[36] *OVH*. Online. Dostupné z: https://www.ovhcloud.com/en/. [cit. 2025-05-13].

[37] *Get access on behalf of a user*. Online. 2024. Dostupné z: https://learn.microsoft.com/en-us/graph/auth-v2-user?tabs=http. [cit. 2025-05-21].

[38] *List mailFolders*. Online. Microsoft Graph. 2024. Dostupné z: https://learn.microsoft.com/en-us/graph/api/user-list-mailfolders?view=graph-rest-1.0&tabs=http. [cit. 2024-12-07].

[39] *Desktop app that calls web APIs: Acquire a token using Device Code flow*. Online. 2025. Dostupné z: https://learn.microsoft.com/en-us/entra/identity-platform/scenario-desktop-acquire-token-device-code-flow?tabs=dotnet. [cit. 2025-05-21].

[40] *The Windows Task Scheduler*. Online. 2023. Dostupné z: https://help.2brightsparks.com/support/solutions/articles/43000335743-the-windows-task-scheduler. [cit. 2025-05-21].

[41] MELNIKOV, A a LEIBA, B. Internet Message Access Protocol (IMAP) - Version 4rev2. Online. 2021. Dostupné z: https://doi.org/10.17487/RFC9051. [cit. 2024-11-17].

[42] KUNČICKÝ, Daniel. *Automatizovaný dohledový systém pro IT infrastruktury*. Online, Bakalářská. Brno: Vysoké učení technické v Brně, 2023. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=255675. [cit. 2024-11-17]. Strana 22-23.

[43] POREMSKY, Diane. *Outlook Categories, Flags, and IMAP Accounts*. Online. 2025. Dostupné z: https://www.slipstick.com/outlook/outlook-categories-flags-and-imap-accounts/. [cit. 2025-05-22].

[44] *Flask template*. Online. 2010. Dostupné z: https://flask.palletsprojects.com/en/stable/templating/. [cit. 2025-05-22].

[45] *Slow Website Performance? You Might Be Using Too Much JavaScript*. Online. 2022. Dostupné z: https://gtmetrix.com/blog/slow-website-performance-you-might-be-using-too-much-javascript/. [cit. 2025-05-22].

[46] MURRAY, Luke. *IPv6 in Microsoft Azure*. Online. 2023. Dostupné z: https://luke.geek.nz/azure/IPv6-on-Azure/. [cit. 2025-05-22].

Seznam symbolů a zkratek

API	Application Programming Interface
EU	European Union
GB	Gigabyte
GiB	Gibibyte
HDD	Hard disk
HTTPS	Hypertext Transfer Protocol Secure
HW	Hardware
ID	Identifier
IMAP	Internet Message Access Protocol
IPv4/6	Internet Protocol Version 4/6
IT	Information technology
JSON	JavaScript Object Notation
MB	Megabyte
OS	Operating System
PyPI	The Python Package Index
RAM	Random Access Memory
SFTP	Secure File Transfer Protocol
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SW	Software
VM	Virtual Machine
VPS	Virtual Private Server

Seznam příloh

A Obsah elektronické přílohy

Elektronická příloha obsahuje zdrojové kódy aplikace. Aplikace byla napsána v jazyce Python. Kompatibilní a testované verze jsou 3.7 - 3.12 případně novější. Dále obsahuje manuály k aplikaci a příklady konfiguračních souborů.

- `/` ..... kořenový adresář přiloženého archivu
  - `LICENSE.txt` ..... licence použité knihovny MSAL
  - `Základní manuál.pdf`
  - `Manuál verze 1.5.pdf`
  - `tvorba_AZUREAD_aplikace.pdf`
  - `Převod na cloud.pdf` ..... Manuál
  - `Automat` ..... Soubory pro automatickou instalaci
    - `install.sh` ..... Instalační skript
    - `run.sh` ..... Spouštěcí skript
    - `mailnotifier.tar.gz` ..... Archiv s aplikací
    - `cloud.txt`
  - `Zdrojový kód`
    - `mailnotifier` ..... vyhodnocovací program
      - `mailnotifier.py`
      - `CONFIG.py`
      - `CORE.py`
      - `ExcActions.py`
      - `GETURL.py`
      - `IMAP.py`
      - `SETUP.py`
      - `notificatins.py`
      - `MSAL.py`
    - `app` ..... grafické rozhraní
      - `vzor_denni.xml`
      - `Xtemp.txt`
      - `app.py`
      - `HISTORY.py`
      - `MNconnector.py`
      - `SETGET.py`
      - `TASKS.py`
      - `CNF.py`
      - `EXC.py`
      - `IMAP.py`
      - `PROBLEMS.py`
      - `SetConnect.py`
      - `static` ..... JavaScript soubory
      - `templates` ..... HTML soubory
  - `varianta exe`
    - `mailnotifier.exe`
    - `app.exe`

```
    ├── vzor_denni.xml
    ├── Xtemp.txt
└── internal ..... Konfigurační soubory
    ├── MNinternal.txt ..... globální konfigurační soubor
    ├── MNuser.txt ..... výchozí uživatelský konfigurační soubor
    ├── timeouts.txt
    ├── Fcache.txt
    └── Exceptions.txt
```

B Základní manuál

Příloha obsahuje základní manuál k aplikaci Mailnotifier v1.9 a grafickému rozhraní. Popisuje nejdůležitější nastavení. Grafické rozhraní obsahuje vlastní nápovědu a tak zde není duplikována.

Základní manuál Mailnotifier v1.9

Tento manuál popisuje pouze některé části dohledového systému. Moduly grafického rozhraní obsahují integrovanou nápovědu. Případně jsou natolik intuitivní, že nebudou popsány.

Také můžete vyjít z návodu k verzi 1.5 protože původní funkce jsou zachovány

1. Předpoklady
    a. Unikátní názvy složek a štítků v e-mail schránce
    b. Připravená AZURE AD aplikace (a znáte ID)
    c. Nebo kompatibilní IMAP server
    d. Pokud používáte cloud verzi, vytvořte si servisní složky v e-mailu
        i. Vytvořte kořenovou složku SERVIS
        ii. V ní vytvořte podsložky: ulohy, config, watchdog, missing, stat, content, template, internal

2. Zprovoznění serverové části na OS Linux
    a. Nachystejte PC s OS Ubuntu server minimal (nejnovější verze)
    b. Dále se řiďte body 13-16 v souboru Převod na cloud.pdf

3. Zprovoznění na OS Windows
    a. Vytvořte si složku, kam systém umístíte
    b. Nakopírujte soubory ze složky „varianta exe“ a složku internal
    c. Vytvořte složky config, watchdog, missing, stat, content, template
    d. Nakopírujte vzorový soubor šablony do složky template (nebo si vytvořte vlastní)
    e. Spusťte app.exe – grafické rozhraní
        i. Otevře se konzolové okno a poté webový prohlížeč (http://localhost:5555)
    f. Mailnotifier.exe – vyhodnocovací část
        i. Jako argument bere název konfiguračního souboru
        ii. Lze spustit přes CMD mailnotifier.exe uloha.txt

4. Grafické rozhraní
    a. Přehled problémů – pro každou úlohu a její složku zobrazí počet chybových zpráv
    b. Historie a export – umožňuje vypsat a exportovat zprávy dle vzoru
    c. Statistika – Grafy, které ukazují počty vyhodnocených zpráv
    d. Synchronizace
        i. Vypnuto – provoz na OS windows (konfigurace pouze lokálně)
        ii. Zapnuto – pro cloud a OS Linux (vlastní server)

5. Globální nastavení
    a. Hodnoty
        i. Odkládací složka – kořenová složka, do které vkládáte archivační složky
        ii. Název vzoru – štítek, kterým označujete vzor
        iii. Složka plánovače – v plánovači úloh na OS Windows vytvořte složku, název vložte sem
        iv. Do_scan – globální kořenové složky, které procházet (oddělení čárkou)
        v. Do_not_scan – tyto složky vyřadit z procházení (oddělení čárkou)
        vi. Následují názvy servisních štítků, které aplikace využívá
            1. Není nutné měnit výchozí hodnoty
            2. Nikdy je neměňte po zprovoznění aplikace

b. Notifikace
    i. Povolit notifikace – povolí/zakáže zaslání notifikací
    ii. Šablona předmětu – předmět notifikace
        1. Šablona – zadejte název šablony (složka template)
    iii. Název šablony – globální šablona notifikací (template.txt)
    iv. Tabulka e-mail – příjemci notifikací, po každém uložení se objeví nový řádek
c. Hlášky
    i. Můžete si nastavit hlášky notifikací
d. Definice kategorií
    i. obsahuje nápovědu, klikněte na šipku
    ii. Zapnuto – určuje rozestup mezi zprávami
    iii. Vypnuto – nemá na vyhodnocení vliv

6. Přihlášení
    a. Na hlavní stránce zvolte „Připojení“
    b. Pokud provozujete Exchange Oline, zvolte „Přihlášení k AZURE AD“
        i. Postupujte podle uvedených kroků (zobrazeny přímo na stránce)
    c. Pokud provozujete IMAP, zvolte „Přihlášení k IMAP“¨
        i. Vyplňte všechny hodnoty a uložte
        ii. Běžte zpět do „Připojení“
        iii. Tlačítkem „Ověřit IMAP“ ověříte kompatibilitu serveru

7. Tvorba úlohy a kontrola dochvilnosti
    a. Nejprve nastavte „definice kategorií“
    b. Založte úlohu tlačítkem „Tvorba úloh“
    c. Nezapomeňte vyplnit pole do_scan
        i. Obsahuje povolené kořenové složky, ve kterých se hledají podsložky
        ii. Doručená pošta/Firma1 => nastavte „Doručená pošta“
        iii. Firma1 => nastavte „Firma1“
        iv. Zadejte i odkládací složky _TEMP/_Firma1 => _TEMP,_FIRMA1,FIRMA1,SERVIS
    d. Vytvořte úlohu tlačítkem „vytvořit“
    e. Běžte do „nastavení složek dochvilnosti“
        i. Zvolte název úlohy
        ii. Vybrané složky zapněte a uložte (pouze u zapnutých složek se vyhodnotí dochvilnost)
    f. Běžte do „Nastavení vzorů“ a zvolte úlohu a složku
        i. Načtou se vzory (zprávy označené štítkem _TEMPLATE)
        ii. Pokud se vzory nenačtou, asi máte chybně do_scan
    g. Dle nápovědy na stránce nastavte vzory a uložte

8. Výjimky
    a. Obsahuje integrovanou nápovědu

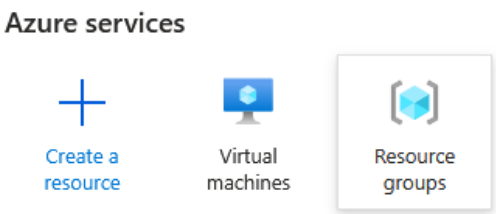
9. Vyhodnocení dle obsahu
    a. Obsahuje integrovanou nápovědu

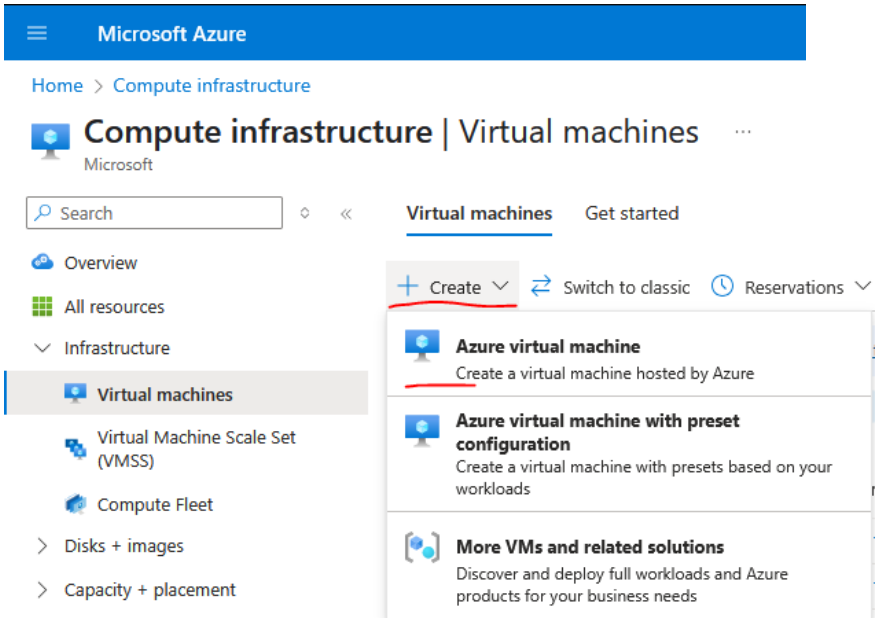
C Převod na cloudovou platformu

Příloha obsahuje manuál, který krok za krokem popisuje zprovoznění aplikace v cloudovém prostředí. Jednotlivé kroky jsou doprovázeny obrázky.

Převod aplikace do Microsoft Azure

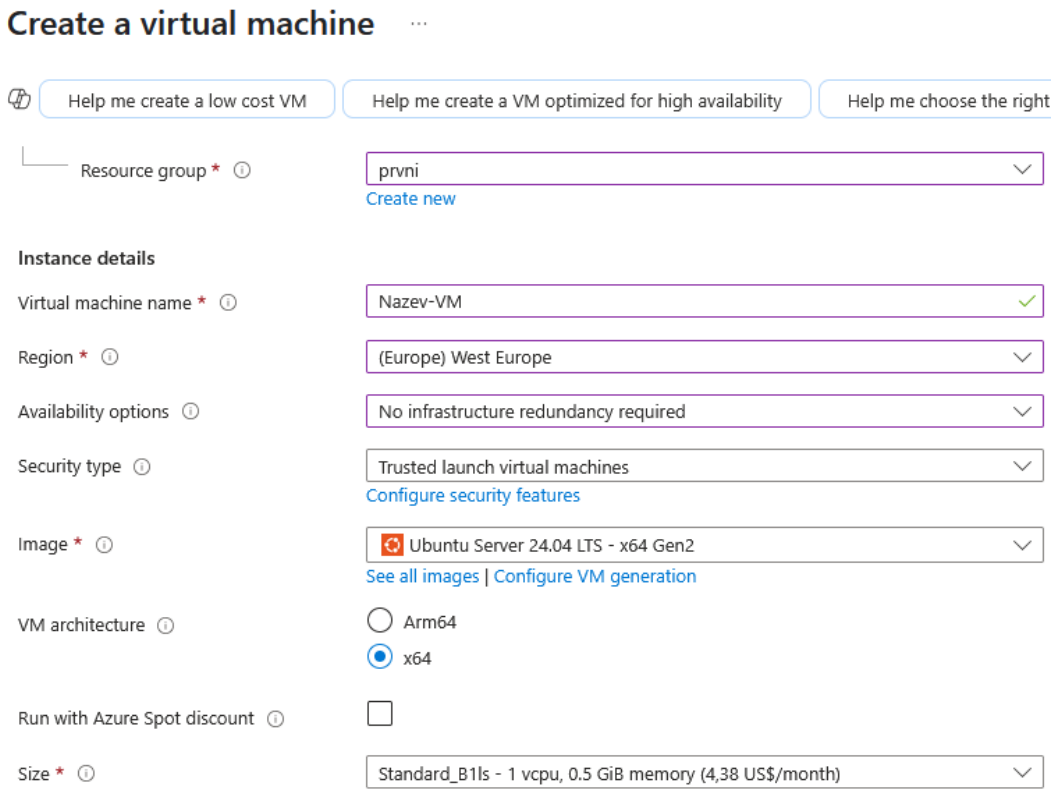
1. **Běžte na portál Azure: https://portal.azure.com**
2. **Klikněte na „Virtual machines“**
3. **Založte nový virtuální počítač tlačítkem create**



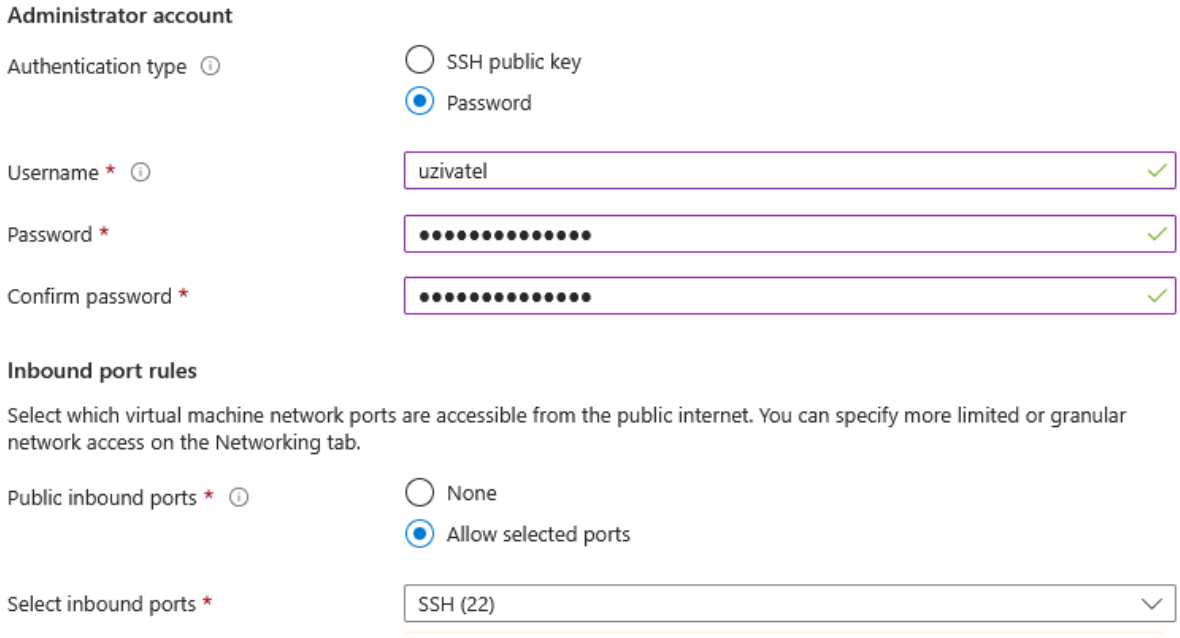


---

4. **Vyplňte požadované parametry**
   a. **Region volte dle vlastního uvážení (testováno v EU Sweden Central)**
   b. **Název VM a „Resource group“ volte vlastní**
   c. **Size zvolte: Standard_B1ls**

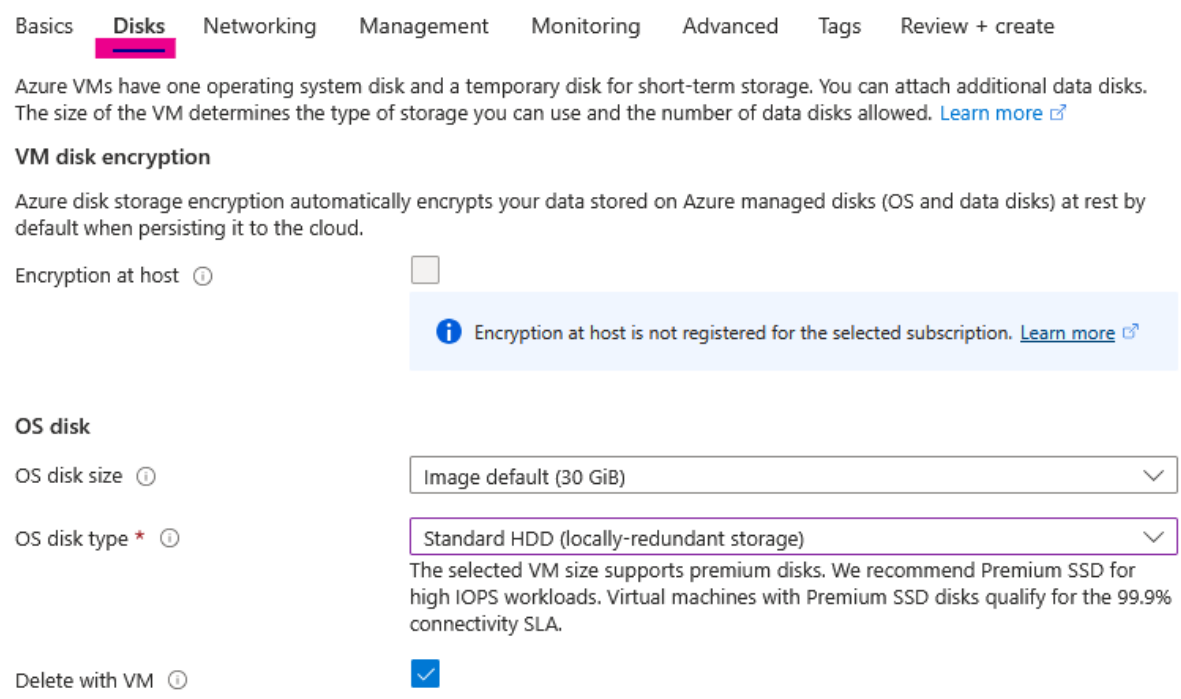


5. **Nastavte si přístup přes internet (později využijete SSH a SFTP)**
    a. **Pro jednoduchost postačuje volba Password (přístup lze pak zrušit)**
    b. **Bezpečnější je varianta SSH public key (nebude v návodu popsáno)**

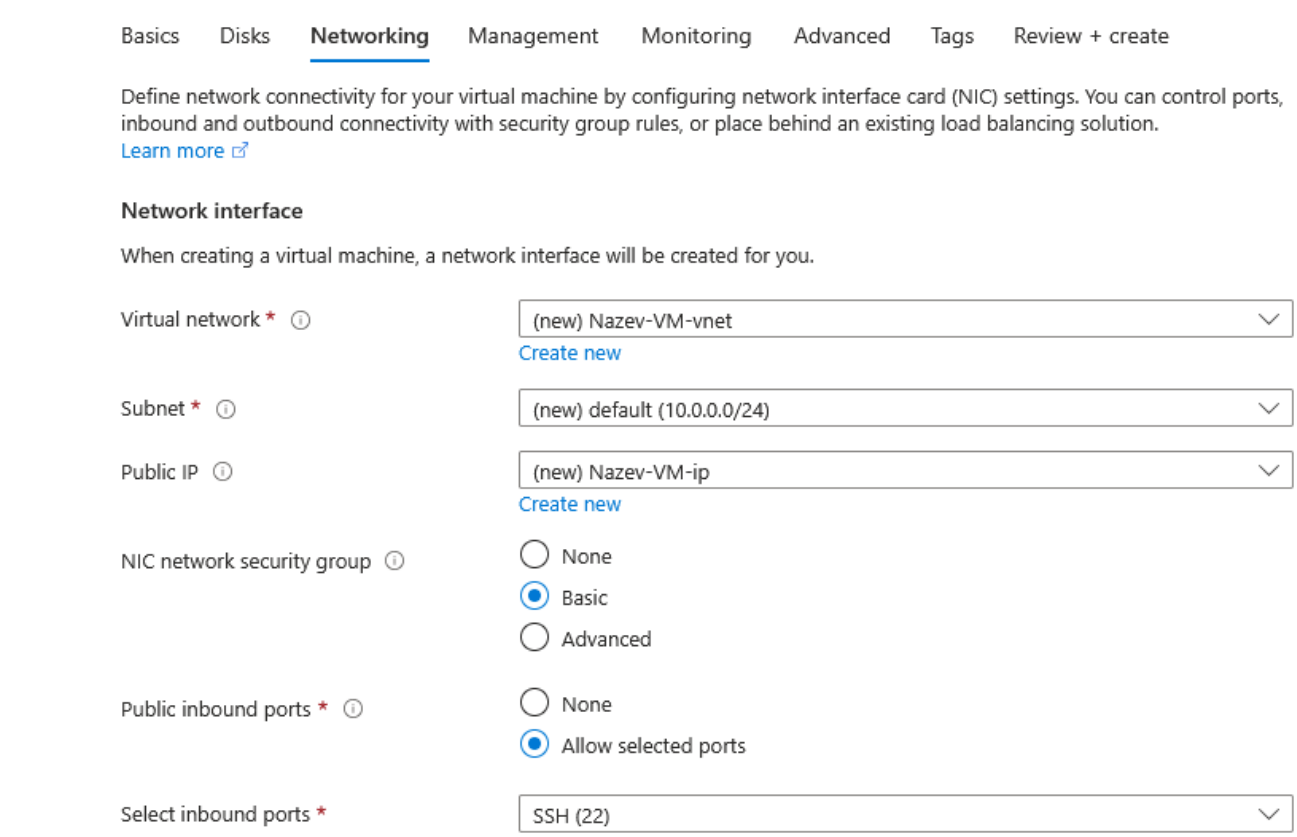


---

6. **V záložce „Disks“ zvolte „Standard HDD“**
    a. **Velikost postačuje nejnižší nabízená (30 GiB)**
    b. **Lze zvolit i dražší varianty SSD**

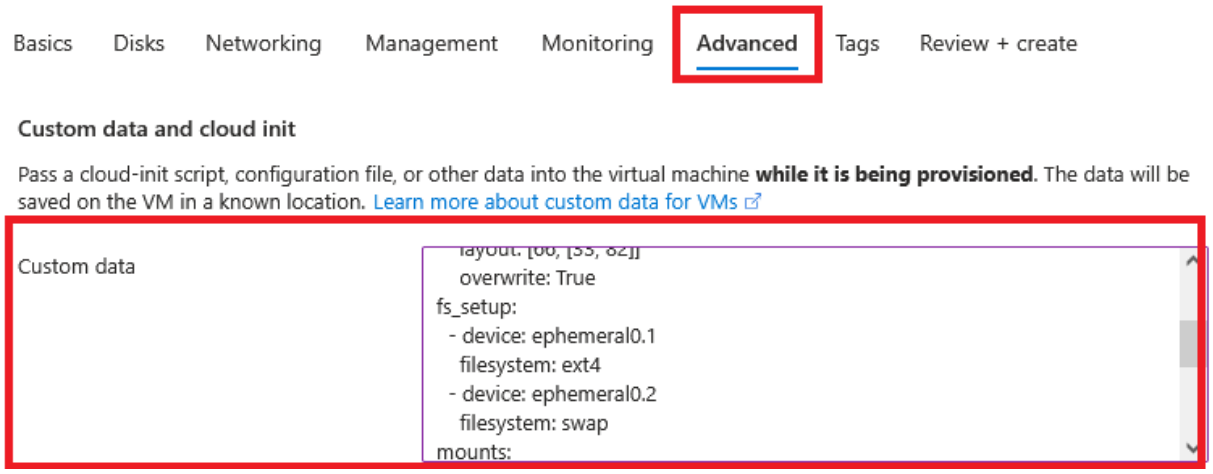


7. **V záložce „Networking“ ověřte nastavení**
    a. **Pro nastavení dle návodu je vyžadována veřejná IP (Public IP)**



---

8. **V záložce „Advanced“ vložte do pole „Custom data“ obsah souboru cloud.txt**



---

9. **V záložce „Review + create“ zkontrolujte data a potvrďte založení tlačítkem „create“**

10. Přejděte na vytvořený PC a zkopírujte si veřejnou IP adresu

Essentials

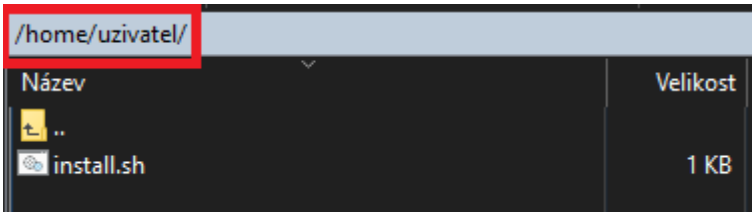
Resource group (move)	: prvni	Operating system	: Linux
Status	: Stopped (deallocated)	Size	: Standard B1ls (1 vcpu, 0.5 GiB memory)
Location	: Sweden Central	Public IP address	: **Zkopírujte IP**
Subscription (move)	:	Virtual network/subnet	: /net/default
Subscription ID	:	DNS name	: Not configured
		Health state	: -

---

11. Připojte se přes SFTP (například programem WinSCP)

a. **Přihlašovací údaje viz bod 5**

b. **Nakopírujte soubor „install.sh“ a „run.sh“ do domovské složky uživatele**



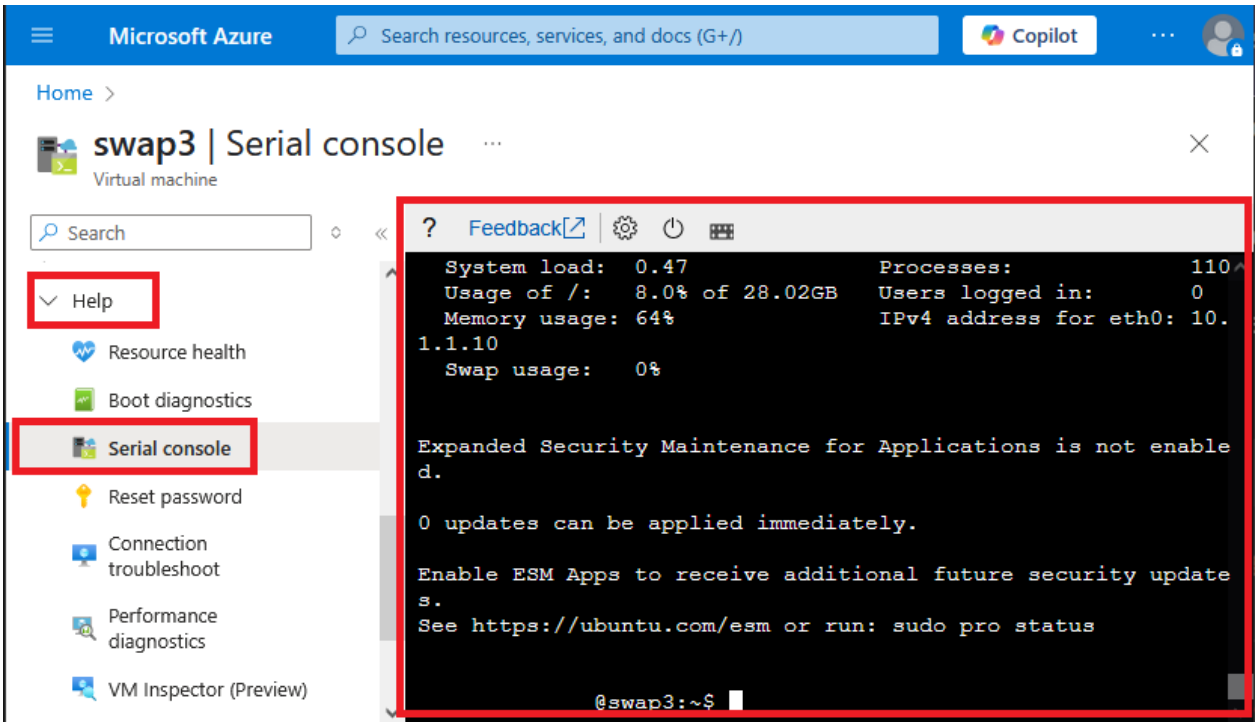
c. **Jméno uživatele dejte dle vašeho systému**

---

12. Přihlaste se k PC

a. **přes SSH (například programem PuTTY)**

b. **nebo přes webovou konzoli**



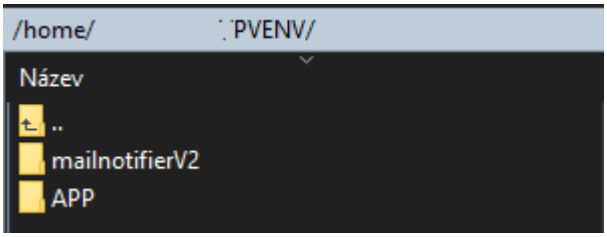
13. **Spusťte instalační skript**
    a. **Do konzole zadejte**
        i. **sudo chmod +x install.sh**
        ii. **./install.sh**

```
user@user:~$ chmod +x install.sh
```

    b. **Proběhne instalace dohledového systému**

---

14. **Ověřte instalaci aplikace**
    a. **Byla vytvořena složka PVENV**



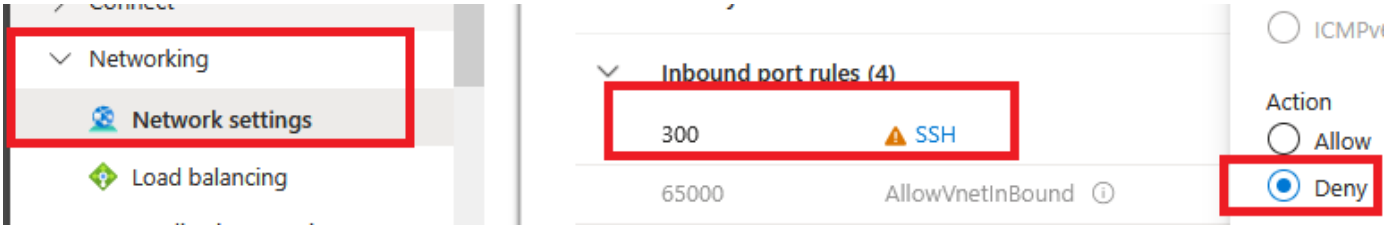
    b. **Složka APP obsahuje složku „bin“**
    c. **Složka „mailnotifierV2“ má obsah (Pokud tam nic není, nastala chyba instalace)**

---

15. **Podle uživatelského návodu vytvořte prvotní soubory**
    a. **Do složky „mailnotifierV2“ kopírujte: my_cache.bin / udaj.txt**
    b. **Do podsložek (mailnotifierV2/podsložka) nakopírujte obsah lokálních podsložek (soubory vezmete tam, kde máte spuštěno grafické rozhraní)**

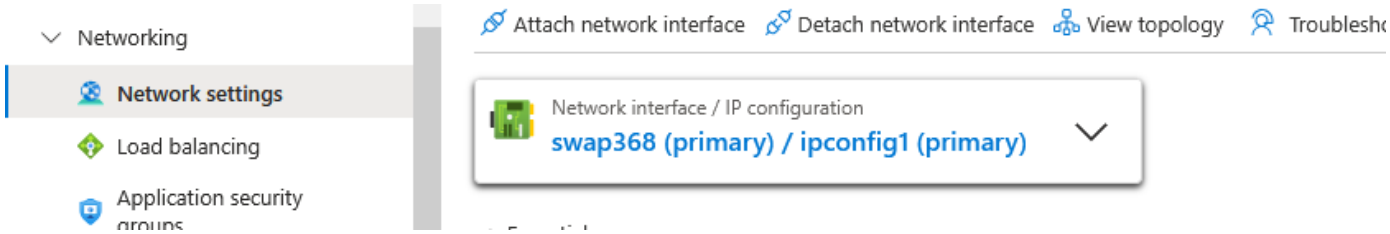
16. **Proveďte prvotní spuštění**
    a. **Pokud nemáte, nakopírujte soubor „run.sh“ do domovské složky uživatele**
    b. **Do konzole zadejte**
        i. **sudo chmod +x run.sh**
        ii. **./run.sh uloha.txt**
        iii. **Název úlohy zadejte svůj (jak byl vytvořen v grafickém rozhraní)**
    c. **Aplikace se spustí a provede prvotní vyhodnocení**
    d. **Automaticky se stáhnout nové úlohy (vytvořené přes grafické rozhraní)**
        i. **Pokud ne, zkontrolujte parametr sync=1 v MNinternal.txt**
    e. **Od této chvíle se budou po každém spuštění aktualizovat úlohy a jejich data**
        i. **Stahují se ze servisní složky e-mailu**
        ii. **Zavádí se do plánovače úloh (cron)**

17. **Dohledový systém je nyní zprovozněn na cloudové platformě.**
18. **(volitelné) Zrušte/zabezpečte přístup přes veřejnou IP adresu**
    a. **Ve vlastnostech virtuálního počítače si nastavte firewall (zakázat SSH)**



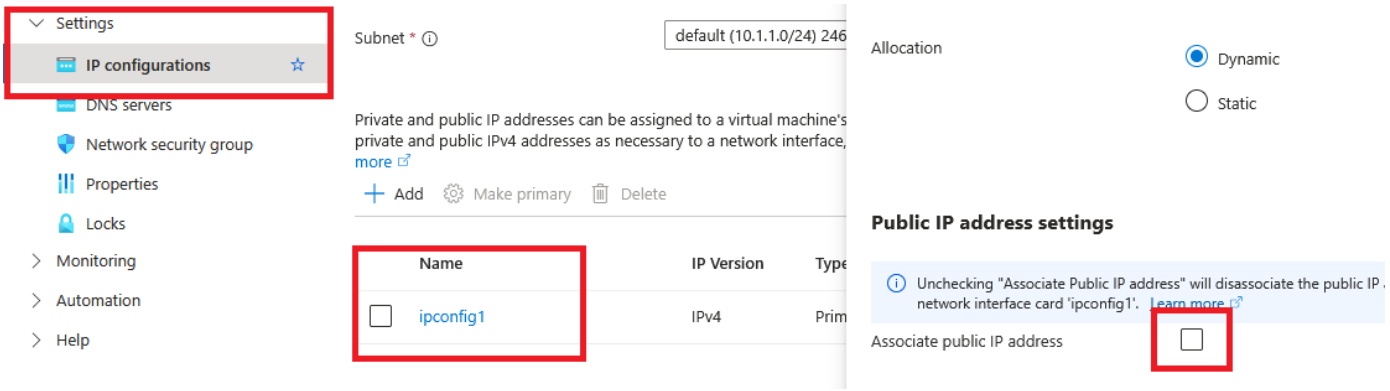
19. Zrušte veřejnou IP (úspora nákladů)

a. Ve vlastnostech VM klikněte na „Network interface“



---

b. Zrušte přiřazení veřejné IPv4 adresy



---

c. Otevřete si „Resource groups“, najděte a smažte veřejnou IP