

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačních technologií**



**Diplomová práce**

**Převod on-premise aplikace do cloudového prostředí**

**Bc. Felix Hampapa**

© 2024 ČZU v Praze

# ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Felix Hampapa

Informatika

Název práce

**Převod on-premise aplikace do cloudového prostředí**

Název anglicky

**Migration of on-premise application to cloud**

---

## Cíle práce

Hlavním cílem této práce bude analýza možností převodu aplikace typu on-premise do cloudového prostředí a návrh postupu pro praktickou implementaci.

Dílní cíle práce jsou:

- Studium a analýza dostupných odborných informačních zdrojů v oblasti cloud computingu
- Analýza výchozí situace a specifikace požadavků pro migraci do cloudu
- Návrh postupu implementace migrace on-premise aplikace do zvoleného cloudového prostředí
- Formulace doporučení a závěrů práce

## Metodika

Teoretická část práce bude založena na studiu odborné a vědecké literatury a online zdrojů se zaměřením na problematiku počítačové infrastruktury a cloudových služeb. V praktické části bude provedena analýza stávající situace a bude vytvořen přehled požadavků pro cloudové řešení. Na základě těchto požadavků budou stanovena vhodná kritéria pro výběr cloudové služby. Následně bude navržen postup implementace migrace on-premise aplikace do zvoleného cloudového prostředí a bude zhodnocena jeho vhodnost. Výsledky teoretické a praktické části budou použity pro formulaci závěrů práce.

## Doporučený rozsah práce

50-60

## Klíčová slova

cloud, cloudové služby, migrace infrastruktury, on-premise, aplikace, softwarové nástroje

---

## Doporučené zdroje informací

BAUN, Christian, Marcel KUNZE, Jens NIMIS a Stefan TAI, 2011. Cloud Computing . B.m.: Springer Berlin Heidelberg. ISBN: 978-3-642-20916-1

ERL, Thomas, Ricardo PUTTINI a Zaigham Mahmood. Cloud Computing: Concepts, Technology & Architecture: The Pearson Service Technology Series from Thomas Er. 2nd ed. Pearson Education, 2023. ISBN 978-0138052256.

GHOLAMI, Mahdi Fahmideh et al. Challenges in migrating legacy software systems to the cloud — an empirical study. Elsevier BV, 2017. DOI: 10.1016/j.is.2017.03.008

JAMSHIDI, Pooyan, Aakash AHMAD a Claus PAHL. Cloud Migration Research: A Systematic Review. Institute of Electrical and Electronics Engineers (IEEE), 2013. DOI: 10.1109/tcc.2013.10

RAMACHANDRAN, Muthu a Zaigham MAHMOOD, eds. Software Engineering in the Era of Cloud Computing. Springer International Publishing, 2020. ISBN 978-3-030-33624-0

---

## Předběžný termín obhajoby

2023/24 LS – PEF

## Vedoucí práce

Ing. Jan Pavlík, Ph.D.

## Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 4. 7. 2023

**doc. Ing. Jiří Vaněk, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 3. 11. 2023

**doc. Ing. Tomáš Šubrt, Ph.D.**

Děkan

V Praze dne 31. 03. 2024

### **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci " Převod on-premise aplikace do cloudového prostředí" jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2024

---

## **Poděkování**

Rád bych touto formou poděkoval svému vedoucímu práce, Ing. Janu Pavlíkovi Ph.D., za jeho trpělivost, věcné rady a odborný dohled při tvorbě této práce. Také bych rád poděkoval všem, kteří mě v průběhu zpracování této práce podporovali, především mé rodině a přátelům.

# Převod on-premise aplikace do cloudového prostředí

## Abstrakt

Tato diplomová práce se soustředí na proces migrace aplikace typu on-premise do cloudového prostředí a s tím spojené problematiky. V teoretické části je čtenář seznámen se základními myšlenkami souvisejícími s cloudovým výpočetním modelem, jednotlivými deployment a service modely používanými v cloudu, výhodami a nevýhodami cloudového prostředí a analýzou největších cloudových poskytovatelů.

V praktické části této práce je provedena analýza migrovaného řešení. Na základě specifikací získaných pro potřebu migrace a jejich zhodnocení byla provedena vícekriteriální analýza variant pro výběr migračního modelu a vhodného cloudového prostředí.

Díky těmto zjištěním byl navržen způsob implementace nové architektury do zvoleného prostředí, vypracován model převodu on-premise databáze do cloudového prostředí, zjištěna finanční náročnost nového řešení a zhodnocena vhodnost.

V poslední části práce bylo na základě zjištěných skutečností formulovány doporučení a závěry práce.

**Klíčová slova:** Cloud, Migrace, On-premise, Legacy-app, Analýza, Implementace, Cloudová architektura, Migrační strategie, PostgreSQL, Microsoft Azure, Google Cloud Platform, AWS

# Migration of on-premise application to cloud

## **Abstract**

This thesis deals with the migration of an on-premise application to a cloud environment and related issues. In the theoretical part, the reader is introduced to the basic ideas related to the cloud computing model, the different deployment and service models used in the cloud, the advantages and disadvantages of the cloud environment and an analysis of the major cloud providers.

In the practical part of this thesis, an analysis of the migrated solution is performed. Based on the specifications obtained for the need of migration and their evaluation, a multi-criteria analysis was performed to select the migration model and the appropriate cloud environment.

With these findings, a method for implementing the new architecture in the selected environment was proposed, a model for converting the on-premise database to the cloud environment was designed, the financial requirements of the new solution were determined, and the suitability was evaluated.

In the last part of the thesis, recommendations and conclusions were formulated based on the findings.

**Keywords:** Cloud, Migration, On-premise, Legacy-app, Analysis, Implementation, Cloud Architecture, Migration Strategy, PostgreSQL, Microsoft Azure, Google Cloud Platform, AWS

# Obsah

<b>1 Úvod.....</b>	<b>11</b>
<b>2 Cíle a metodika.....</b>	<b>12</b>
2.1 Cíle.....	12
2.2 Metodika.....	12
<b>3 Teoretická část práce.....</b>	<b>13</b>
3.1 Definice cloud computingu.....	13
3.2 Počátky Cloudu.....	14
3.3 Dělení cloud computingu.....	15
3.3.1 Deployment model.....	15
3.3.2 Service model.....	16
3.4 Výhody a nevýhody cloudových služeb.....	19
3.4.1 Výhody Cloudu.....	19
3.4.2 Škálování.....	21
3.4.3 Nevýhody Cloudu.....	23
3.4.4 Ztráta kontroly nad infrastrukturou.....	27
3.4.5 Multiregionalita a právní otázky.....	27
3.4.6 Přesáhnutí budgetu.....	28
3.5 Analýza cloudových platforem.....	28
3.5.1 Přehled cloudových poskytovatelů.....	28
3.5.2 Amazon Web Services (AWS):.....	29
3.5.3 Microsoft Azure:.....	31
3.5.4 Google Cloud Platform (GCP).....	33
3.5.5 Ostatní poskytovatelé.....	34
<b>4 Praktická část práce.....</b>	<b>36</b>
4.1 Analytická část.....	36
4.1.1 Popis Aplikace.....	36
4.1.2 Analýza stávajícího řešení.....	36
4.1.3 Výběr migračního modelu.....	40
4.1.4 Volba vhodného cloudového řešení.....	43
4.2 Implementace.....	49
4.2.1 Nová architektura.....	49
4.2.2 Architektura pro MS azure.....	50
4.2.3 Návrh infrastruktury pro Azure.....	51
4.2.4 Migrace PostgreSQL.....	54
<b>5 Výsledky a diskuse.....</b>	<b>59</b>
5.1 Zhodnocení vhodnosti prostředí.....	59



5.2	Analýza nákladů.....	59
5.3	Doporučení.....	60
<b>6</b>	<b>Závěr.....</b>	<b>61</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>62</b>
<b>8</b>	<b>Seznam obrázků, tabulek, grafů a zkratk.....</b>	<b>67</b>
8.1	Seznam obrázků .....	67
8.2	Seznam tabulek .....	67
8.3	Seznam použitých zkratk.....	68

# 1 Úvod

Moderní doba informačních technologií přináší kontinuální vývoj a nové výzvy pro organizace všech velikostí. S narůstajícími požadavky na výpočetní výkon, flexibilitu a efektivitu, mohou tradiční IT infrastruktury postupně dosahovat svých limitů. V tomto kontextu se ukazuje Cloud Computing jako revoluční koncept, který nabízí nové možnosti a perspektivy pro provoz a rozvoj informačních systémů jako takových. Trvání společností na svých stávajících on-premise řešeních může být nahlíženo různorodě, avšak je potřeba v každé situaci zvážit, zdali se ve všech případech vyplatí přechod z on-premise do cloudu, například u legacy řešení (aplikací), sloužící pouze k jednoduchým úkonům.

S rostoucí popularitou Cloud Computingu a stále se rozšiřujícím portfoliem cloudových služeb se jeví jako důležité porozumět výhodám a potenciálním rizikům spojeným s migrací do cloudu.

V rámci této práce budou zkoumány cloudové prostředí a služby třemi nejpoužívanějšími platformami pro cloud v roce 2023 (1), které jsou v následujícím pořadí, Amazon AWS, Microsoft Azure a Google Cloud.

## **2 Cíle a metodika**

### **2.1 Cíle**

Hlavním cílem této práce bude analýza možností převodu aplikace typu on-premise do cloudového prostředí a návrh postupu pro praktickou implementaci.

Dílní cíle práce jsou:

- Studium a analýza dostupných odborných informačních zdrojů v oblasti cloud computingu
- Analýza výchozí situace a specifikace požadavků pro migraci do cloudu
- Návrh postupu implementace migrace on-premise aplikace do zvoleného cloudového prostředí
- Formulace doporučení a závěrů práce

### **2.2 Metodika**

Teoretická část práce bude založena na studiu odborné a vědecké literatury a online zdrojů se zaměřením na problematiku počítačové infrastruktury a cloudových služeb. V praktické části bude provedena analýza stávající situace a bude vytvořen přehled požadavků pro cloudové řešení. Na základě těchto požadavků budou stanovena vhodná kritéria pro výběr cloudové služby. Následně bude navržen postup implementace migrace on-premise aplikace do zvoleného cloudového prostředí a bude zhodnocena jeho vhodnost. Výsledky teoretické a praktické části budou použity pro formulaci závěrů práce.

## 3 Teoretická část práce

### 3.1 Definice cloud computingu

Cloud computing je model, který umožňuje pohodlný, on-demand přístup ke sdílenému fondu konfigurovatelných výpočetních zdrojů (např. sítě, servery, úložiště, aplikace a služby), které lze rychle poskytnout a uvolnit s minimálním úsilím na správu nebo interakci s poskytovatelem služeb a je možné se k němu připojit odkudkoliv. (2) Model cloudového computingu dle NIST (2) se skládá z pěti základních charakteristik, *tří modelů služeb a čtyř modelů nasazení*.

Základní charakteristiky dle NIST (2):

- *On-demand self-service: Zákazníkovi mohou být poskytnuty výpočetní možnosti, jako je čas serveru, či síťové úložiště, podle potřeby, automaticky bez nutnosti lidské interakce s poskytovatelem služeb.*
- *Broad network access: Možnosti cloudu jsou k dispozici přes síť a jsou přístupné prostřednictvím standardních mechanismů, které podporují použití heterogenními tenkými nebo tlustými klientskými platformami (např. mobilní telefony, tablety, notebooky a pracovní stanice).*
- *Resource pooling: Výpočetní zdroje poskytovatele jsou sdruženy tak, aby sloužily více spotřebitelům pomocí modelu multi-tenant, přičemž různé fyzické a virtuální zdroje jsou dynamicky přiřazovány a přidávány podle poptávky spotřebitele.*
- *Rapid elasticity: Zdroje lze pružně poskytovat a uvolňovat, v některých případech automaticky, aby se rychle škálovaly v souladu s poptávkou. Pro spotřebitele tyto schopnosti jsou kdykoliv k dispozici, působí neomezeně a mohou být přisvojeny v jakémkoli množství kdykoli.*
- *Measured service: Cloudové systémy automaticky řídí a optimalizují využití zdrojů tím, že využívají měřicí schopnost na nějaké úrovni abstrakce vhodné pro typ služby (např. úložiště, zpracování, šířka pásma a aktivní uživatelské účty). Využití zdrojů lze sledovat, řídit a hlásit, což poskytuje transparentnost jak pro poskytovatele, tak pro spotřebitele užívané služby.*

## 3.2 Počátky Cloudu

Myšlenka Cloud computingu není žádnou novinkou jednadvacátého století. Již roku 1961 prohlásil (3) profesor John McCarthy na půdě MIT: „*Výpočty by se jednoho dne mohly organizovat jako veřejná služba, stejně jako je telefonní systém veřejnou službou. Každý předplatitel musí platit pouze za kapacitu, kterou skutečně využívá, ale má přístup ke všem programovacím jazykům charakteristickým pro velmi velký systém ... Někteří předplatitelé by mohli nabízet služby jiným předplatitelům ... Počítačová služba by se mohla stát základem nového a důležitého průmyslu.*“.

Za předpoklad vzniku cloudových služeb lze považovat vznik technologie virtualizace kolem roku 1960 firmou IBM a jejím projektu CP-40, který byl prvním CTSS systémem. Tento systém umožňoval sdílení výpočetní paměti stroje (počítače) pro výpočetní úkony a na rozdíl od jejího předchůdce Batch procesingu umožňoval efektivnější rozdělení času na stroji v neprospěch maximálního vytižení procesoru.

V roce 1999 představila firma Salesforce svou CRM službu umožňující přímou komunikaci s klienty fungující na principu SaaS. (4) Jednou z klíčových výhod Salesforce CRM (a SaaS obecně) bylo umožnění firmám odebírat produkt za pomoci předplatného, namísto potřebné masivní počáteční investice a následných dalších plateb za údržbu/provoz. (5)

Další významný posun v oblasti cloudu zaznamenala firma Amazon, respektive její odnož AWS (Amazon Web Services). V rámci svého působení jakožto internetový prodejce se firma potýkala s neefektivně využívaným hardwarem pro své služby a díky aplikování principů dnes známých z cloud computingu se jí výrazně podařilo zvýšit efektivitu svého provozu. V roce 2006 poté nabídla své služby pro komerční využití skrze Elastic Compute Cloud (EC2) čímž zahájila éru cloud computingu, který se dostal do doby, jak jej známe dnes. (6)

## 3.3 Dělení cloud computingu

### 3.3.1 Deployment model

#### 3.3.1.1 Veřejný cloud

Veřejný cloud je forma výpočetního modelu, při kterém jsou cloudové služby či kapacity poskytovány poskytovateli cloudových služeb a jsou veřejně dostupné pro zakoupení pro uživatele, firmy či organizace. V rámci tohoto modelu poskytovatelé nabízejí portfolia služeb a infrastruktury, která organizacím umožňují využívat výpočetní výkon bez nutnosti provozování vlastní infrastruktury (7).

Mezi poskytovatele veřejného cloudu lze zařadit služby Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform , IBM Cloud a další. (1)

#### 3.3.1.2 Privátní cloud

Privátní cloud je model cloudového computingu, kdy jsou infrastruktura a služby provozovány výhradně pro určitou entitu či jednotlivce. Tento typ cloudu může být zajišťován samotnou organizací, případně prostřednictvím externího poskytovatele, který poskytne organizaci izolovaný prostor na svém hardwaru. (7)

Privátní cloud je především vhodný, je-li zapotřebí vysoká úroveň kontroly nad bezpečností a soukromím nad daty organizace a infrastrukturou, například z důvodů legislativních (například bankovníctví a zdravotnictví). (8)

#### 3.3.1.3 Hybridní cloud

Hybridní cloud je model cloudového computingu, který kombinuje prvky veřejného cloudu a privátního cloudu do jednoho celku. V tomto modelu organizace využívají jak vlastní privátní cloud, tak i služby veřejného cloudu. Tímto způsobem mohou organizace optimalizovat své IT prostředí tak, aby splňovalo různé požadavky na bezpečnost, výkon, soukromí a náklady. (7)

Příklady využití hybridního cloudu zahrnují organizace, které mají citlivá data, jako jsou osobní údaje zákazníků, a zároveň potřebují škálovatelnost a pružnost veřejného cloudu pro své aplikace. Tento model je také vhodný pro organizace, které postupně migrují své služby do cloudu a chtějí udržet některé zdroje v interní infrastruktuře. V hybridním, cloudu

se. Však mohou skrývat úskalí v rámci managementu prostředí, které je rozdělené do několika odlišných architektur a rozdělení odpovědnosti za prostředí. (8)

Dále jsou v závislosti na poskytovateli účtovány poplatky za odesílání dat mezi cloudem a on-premise řešení, čímž se tento model může dále prodražit. (9)

#### 3.3.1.4 Multicloud

Multicloud je model, při kterém zákazník využívá služeb od více cloudových poskytovatelů naráz. Tento model může existovat například z důvodu potřeby využití služeb, které v ostatních cloudech nejsou dostupné, či z historických důvodů, kdy by se nevyplatilo přenést aplikaci do nového prostředí. V tomto ohledu je však potřeba brát zřetel na to, že většina poskytovatelů cloudu si účtuje částky za odchozí komunikaci mimo své prostředí. (7) (8) (9)

### 3.3.2 Service model

#### 3.3.2.1 IaaS

Infrastructure as a Service (IaaS) je koncept, který v sobě skrývá přístup k virtuální infrastruktuře a výpočetním zdrojům od poskytovatelů cloudu skrze internetovou síť. IaaS poskytuje flexibilní a on-demand přístup k základním výpočetním zdrojům, jako jsou virtuální servery, úložiště a síťové prvky, aniž by bylo nutné provozovat vlastní fyzický hardware. (7)

V rámci modelu IaaS zajišťuje poskytovatel správu fyzické infrastruktury, jako jsou serverové farmy, úložiště dat a síťové prvky. Klienti mají možnost virtuálně vytvářet, spravovat, či konfigurovat své vlastní virtuální servery, které běží na fyzickém hardwaru poskytovatele. Tím je umožněno efektivní využívání zdrojů, jelikož virtuální servery mohou být snadno škálovány podle aktuálních potřeb a nároků na výpočetní výkon. (7)

IaaS nabízí oproti on-premise několik klíčových výhod. Jedním z nich je elasticita, která umožňuje klientům rychle a pružně škálovat své výpočetní zdroje dle aktuálních požadavků, což může zvýšit efektivitu využití zdrojů a umožňuje lepší reakci na aktuální potřeby uživatelů. Dále je oproti on-premise řešení snižena hodnota vstupních investic, jelikož organizace nepotřebuje investovat například nákupem serverů do hardware a infrastruktury, ale zakoupit si skrze IaaS pouze tolik výpočetní kapacity, kolik potřebuje.

Modelem IaaS se oproti on-premise minimalizuje časová a finanční náročnost na správu a údržbu infrastruktury, jelikož ta je spravována ze strany poskytovatele, což může snížit provozní náklady v rámci společnosti a umožňuje zdroje přesunout na vhodnější oblasti. (7)

S ohledem na IaaS je klíčovým aspektem flexibilní fakturace, která je založena na modelu pay-as-you-go. Uživatel cloudových služeb platí za skutečně využitý zdroj, což umožňuje efektivní správu nákladů a umožňuje lépe předvídat a kontrolovat IT výdaje. V rámci cloudových poskytovatelů však také existuje možnost předplacení konkrétních zdrojů a služeb v určené kapacitě, čímž se dá snížit cenu zdroje/služby na úkor flexibility (10).

Celkově lze konstatovat, že Infrastructure as a Service představuje fundamentální stavební kámen pro poskytování výpočetních zdrojů v cloudovém prostředí. Jeho klíčové charakteristiky, jako je elasticita, snížená investiční náročnost a flexibilní fakturace, přispívají k efektivnějšímu využívání IT zdrojů a podporují agilní a inovační přístup organizací k řízení svých informačních technologií.

### 3.3.2.2 PaaS

Platform as a Service (PaaS) je jedním z konceptových pilířů cloudového computingu, který poskytuje klientům kompletní vývojové a nasazovací prostředí pro tvorbu aplikací bez nutnosti starat se o infrastrukturu a složité technické detaily. PaaS představuje jednu ze tří hlavních kategorií cloudových služeb, společně s Infrastructure as a Service (IaaS) a Software as a Service (SaaS). (7)

V rámci modelu PaaS poskytuje cloudový poskytovatel platformu obsahující vývojové nástroje, programovací jazyky, databáze, systémy pro správu identit a další potřebné služby. Uživatelé mohou díky tomu vytvářet aplikace, které využívají tyto nástroje, aniž by museli provádět složité konfigurace infrastruktury pro fungování těchto aplikací. Tím může být dosaženo značného zjednodušení procesu vývoje a nasazování aplikací. (11)

PaaS nabízí několik klíčových výhod. Jednou z nich je rychlost a efektivita vývoje, neboť vývojáři se mohou soustředit na vývoj samotných funkcí aplikace namísto zdoluhavých konfigurací infrastruktury. Dále je zde značná pružnost, neboť klienti mohou rychle škálovat aplikace podle potřeby, což je zejména výhodné pro aplikace s proměnlivým zatížením. PaaS také usnadňuje spolupráci vývojových týmů, jelikož umožňuje sdílet a spolupracovat na kódu a projektech v rámci jednotného prostředí. Díky správě ze strany dodavatele a pravidelným aktualizacím infrastruktury jsou také zajištěny konzistentní



výkonnostní parametry a zabezpečení aplikací, což nemusí být pravidlem v on-premise prostředí. Klienti, stejně jako tomu je u modelu IaaS, mohou platit za využívání platformy a služeb, které skutečně využívají díky modelu pay-as-you-go, což umožňuje efektivní správu nákladů. (11) V rámci cloudu můžeme na služby typu SaaS považovat manažerované databáze, jako je Azure Database for PostgreSQL, pokud je provozována jako služba, či Azure app service pro hostování webových služeb (12).

Celkově lze konstatovat, že Platform as a Service představuje významný krok vpřed v rámci cloudových služeb, jelikož umožňuje organizacím a vývojářům soustředit se na kreativní proces vývoje aplikací a přesunout komplexní technické detaily na cloudové poskytovatele. To zvyšuje efektivitu vývoje, umožňuje inovace a podporuje agilní přístup k tvorbě a správě softwaru.

### 3.3.2.3 SaaS

Software as a Service (SaaS) je další z hlavních konceptů v rámci cloudového výpočetního modelu, který poskytuje klientům přístup k softwarovým aplikacím a službám skrze internet. Model SaaS je založen na poskytování softwaru jako služby, přičemž uživatelé nejsou povinni provozovat a spravovat aplikace na svých zařízeních a jsou jim k dispozici skrze internetovou síť. V rámci modelu SaaS zajišťuje cloudový poskytovatel nasazení, správu a údržbu softwaru na svých serverech a infrastruktuře. (7)

Hlavní výhodou modelu SaaS je rychlost nasazení a snadná dostupnost aplikací. Uživatelé mohou začít využívat aplikace okamžitě po přihlášení a nemusí se starat o složité procesy instalace a konfigurace. To je zvláště výhodné pro organizace, které mají potřebu efektivně využívat software bez zbytečných prodlev.

SaaS také přináší výhody z hlediska údržby a aktualizací. Poskytovatelé zajišťují automatické aktualizace softwaru, což znamená, že uživatelé vždy pracují se základnou verzí softwaru, která je aktuální a zabezpečená. Tím je minimalizován výskyt zastaralých verzí a potřeba manuálních aktualizací.

Model SaaS často využívá modelu fakturace založeného na platbách za užívání služby. Uživatelé platí za přístup k softwaru a jeho funkcionality, což umožňuje efektivní správu nákladů a minimalizaci investic do licencí a infrastruktury. (7)

Celkově lze SaaS považovat za důležitý model v rámci cloudových služeb, který umožňuje organizacím a jednotlivcům rychle a snadno využívat softwarové aplikace bez

složitostí spojených s jejich provozem a údržbou. Mezi SaaS můžeme řadit služby jako je Salesforce, Microsoft Teams, či Netflix.

## **3.4 Výhody a nevýhody cloudových služeb**

### **3.4.1 Výhody Cloudu**

#### **3.4.1.1 Redukce nákladů**

Jedním z důvodů, proč ve firmě implementovat cloudové řešení je potenciál snížení nákladů.

V rámci on-premise se dají rozdělit náklady na infrastrukturu do dvou kategorií, na kterých které lze pohlížet jako na místa pro případnou úsporu:

#### **Fyzická Infrastruktura**

Využití cloudového řešení s vysokou pravděpodobností sníží potřebu počáteční investici do hardware. Díky využití úspor z rozsahu, které nabízejí poskytovatelé cloudu, může klient získat přístup k podnikovému hardwaru a softwaru za zlomek ceny, kterou by zaplatil za jejich pořízení. Může se jednat o fyzické servery, kabeláž a další vybavení prostor, či zařízení sloužící pro zálohy. Tyto kapitálové investice pak mohou být využity k jiným účelům. Na rozdíl on on-premise řešení mohou být, v případě podhodnocení či nadhodnocení potřebných výpočetních kapacit pro provoz chtěné služby či aplikace, kapacity navýšeny či sníženy dle potřeby bez finanční sankce (při pay-as-you-go modelu). (7) V případě, kdy společnost již vlastní své servery či infrastrukturu, úspora na fyzické infrastruktuře nemusí být žádná, ba naopak, pokud by měla firma fyzický hardware, který by nevyužívala a raději by sáhla po cloudovém řešení, nemuselo by se jí to vyplatit. (13)

#### **Provoz infrastruktury**

Při pořízení vlastní IT infrastruktury je nutné počítat také s provozními náklady na její provoz. Může se jednat o účty za elektřinu, licence na software, zabezpečení přístupů k hardwaru, či pronájem prostorů pro provoz infrastruktury apod. On-premise řešení také vyžaduje specialisty, kteří se budou starat o její provoz a údržbu, a to i v případě, že IT systém není využíván naplno. Z těchto důvodů může být přechodem na cloudové řešení možné ušetřit na provozních a personálních nákladech. S cloudem již není zapotřebí tolika

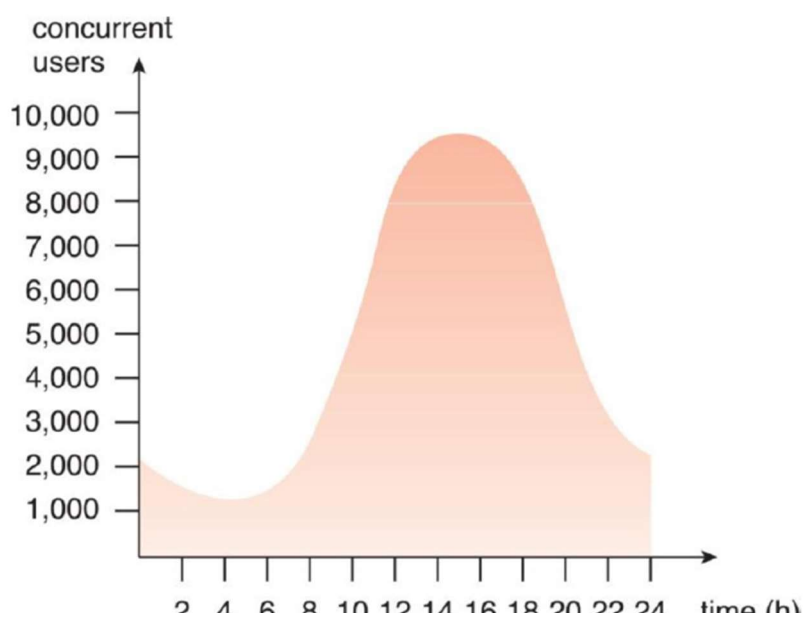
IT specialistů, protože servery jsou v režii poskytovatele cloudu. Další výhodou je Pay-as-you-go model, kdy klient platí pouze za skutečně odebrané výpočetní kapacity v předem domluvených časových intervalech. Pokud jsou navíc využity platformy pro vývoj, či předpřipravené zdroje od poskytovatele, mohou být ušetřeny náklady potřebné pro nastavení řešení oproti on-premise. (7)

### 3.4.1.2 Agilita

Agilita je v oblasti IT potřeba pro rychlé a efektivní reakce v místech, kde je to v daný čas třeba, tak aby firma mohla v krátkém čase reagovat na vnější či vnitřní potřeby a udržela si tím svou konkurenceschopnost. V praxi je běžně možné se setkat například i se změnou požadavků aplikace na výkon či paměť, potřebou změny v její infrastruktuře, či se může jednat o přechod na zdroj s vyšším SLA. Veškeré změny je možné provádět s nulovou či minimální počáteční investicí v poměru s on-premise řešením, a také téměř okamžitě. (7)

Díky agilitě v rámci cloudu může klient naplánovat, že bude využívat pouze zdroje, které v daný okamžik potřebuje a na základě potřeby se buď automaticky, nebo manuálně vyškálují či naopak sniží.

Můžeme si jako modelový příklad představit aplikaci sloužící k zapisování docházky, ve které dochází k vysokému vytížení každý všední den kolem 17 h. Nejvyšší vytížení poté probíhá v pátek kolem 17 h, kdy přijde uživatelům, kteří ještě za uplynulý týden nevyplnili docházku, e-mail ohledně připomenutí potřeby jejího vyplnění. Aby aplikace zvládala výše



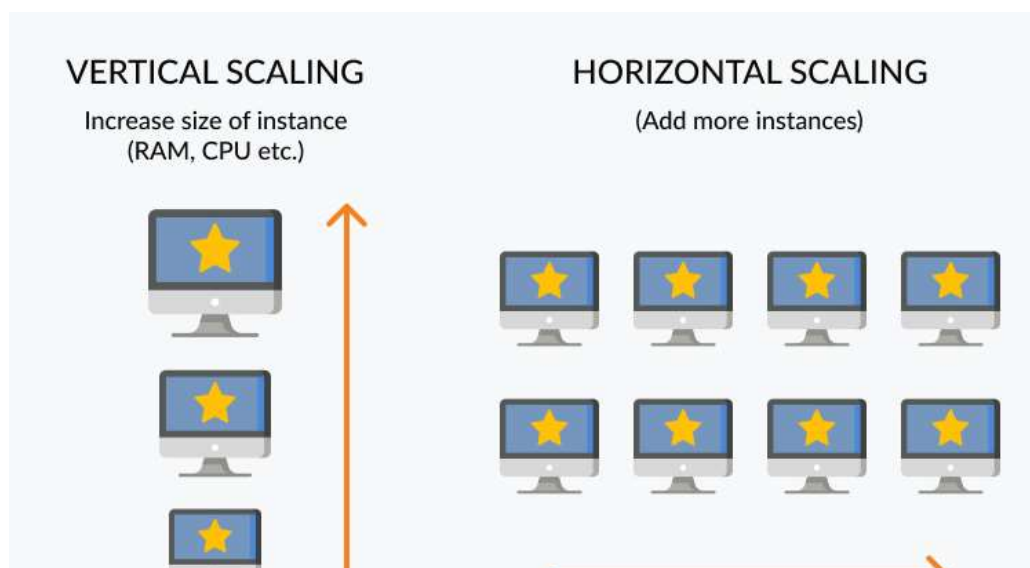
Obrázek 1- Příklad výkyvu při fungování aplikace, Zdroj: (7)

uvedená maxima, musí její minimální potřebná kapacita dosahovat alespoň pátečního maxima, přičemž mimo výkyvy poběží na minimální výkon. Pokud by se jednalo o on-premise řešení, museli bychom mít takovou infrastrukturu, která by tyto výkyvy zvládla i přesto, že by po většinu času byla nevyužitá. V případě cloudového řešení můžeme nastavit horizontální škálování tak, aby se při dosažení předem stanoveného vytížení vytočila další virtuální stanice, či kontejner, který pojme potřebnou kapacitu. Jakmile poté vytížení začne klesat, dle nastavených preferencí se začne snižovat počet stanic či kontejnerů obsluhující klienty, a díky využití pay-as-you go modelu zajistíme to, že uživatel platí pouze za odebrané kapacity. (11)

### 3.4.1.3 Škálování

Škálování neboli navyšování kapacit slouží k úpravě výkonosti jednotlivých virtuálních zdrojů, jako jsou například virtuální stroje. Škálování také přímo souvisí s agilitou.

Škálování můžeme dělit na dvě základní rozdělení, vertikální a horizontální. Vertikální škálování znamená zvyšování nebo snižování výkonu jednoho virtuálního stroje. Horizontální škálování znamená zvyšování nebo snižování počtu virtuálních strojů v závislosti na aktuálních potřebách. Vertikální škálování se týká škálování přidáním většího výkonu (např. přidáním silnějšího procesoru, RAM apod.), zatímco horizontální škálování se týká škálování přidáním dalších strojů do fondu zdrojů. V souvislosti s cloudem nejsou možnosti škálování omezeny kapacitou vlastněných na on-premise infrastruktuře a díky



Obrázek 2 - Principy škálování, zdroj: (58)

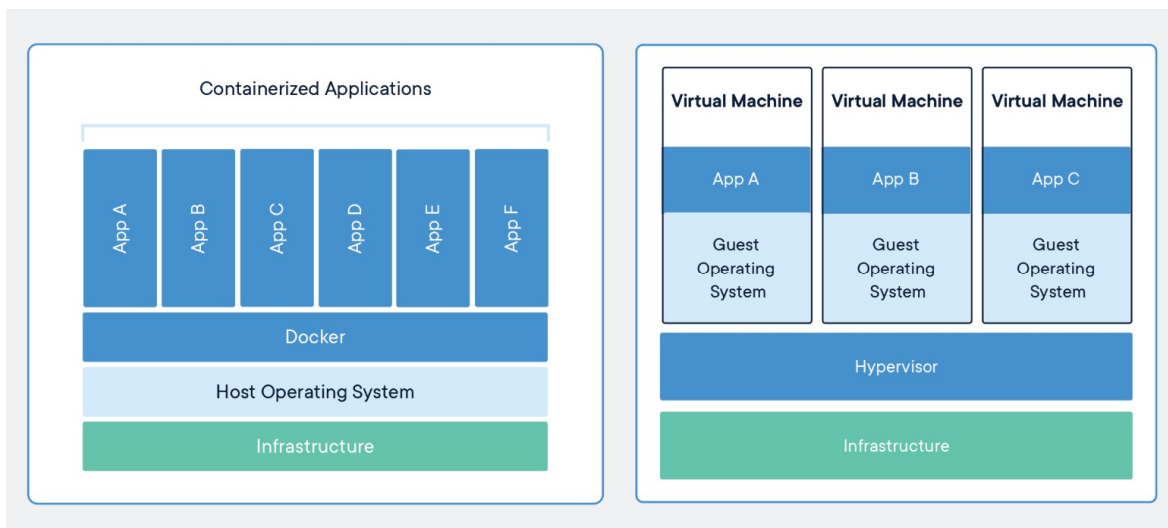
tomu se otevírá možnost dynamičtějšího (agilnějšího) škálování přímo dle aktuální potřeby.  
(11) (14)

#### 3.4.1.4 Virtualizace

Virtualizace je technologie, která umožňuje vytvářet virtuální počítačové zdroje, které byly tradičně vázány na hardware. Tento proces umožňuje využití kapacity fyzického stroje rozdělením jeho schopností mezi více uživatelů nebo prostředí. To v praxi znamená, že na jednom fyzickém serveru můžete provozovat více operačních systémů a aplikací ve formě virtuálních strojů. Tento přístup umožňuje efektivnější využití serverů a snížení provozních nákladů díky snížení počtu fyzických serverů potřebných pro provoz stejného množství služeb (oproti tomu, pokud bychom měli pro každý počítač vlastní hardware). Virtualizace není specifickou technologií, která funguje pouze v souvislosti s cloudem, avšak jsou v rámci cloudu plně využívány a jsou díky němu rozšiřovány jeho možnosti, především, jedná-li se o škálovatelnost, či agilitu. (7)

#### 3.4.1.5 Kontejnerizace

Kontejnerizace je koncept, který umožňuje spouštět aplikace bez potřeby mít pro každou jednotlivou aplikaci vlastní virtuální stroj s vlastním operačním systémem, namísto toho se používají takzvané pody. Ve virtualizaci se hypervisor stará životní cyklus jednotlivých virtuálních stanic. Ony stanice se poté tváří jako standardní stanice s vlastním operačním systémem, knihovnamí apod. Naproti tomu v dockeru fungují jednotlivé pody na bázi virtualizaci jádra operačního systému, což má za příčinu to, že běží v rámci jednoho



Obrázek 3 - Rozdíl mezi kontejnerizací a virtualizací, zdroj: (15)

operačního systému, navzájem sdílejí operační paměť a mají nainstalovány jen software, který používají pro své fungování, čímž je, na rozdíl od klasické virtualizace, výrazně efektivnější, především díky agilitě a výrazně hospodárnějšímu využívání systémových prostředků. Kontejnerizace byla poprvé využita v 70 letech pro lepší separaci aplikačního kódu v Unixových systémech, čímž bylo dosaženo sandboxu vhodného pro testování aplikací, servisů a procesů. (15)

### 3.4.2 Nevýhody Cloudu

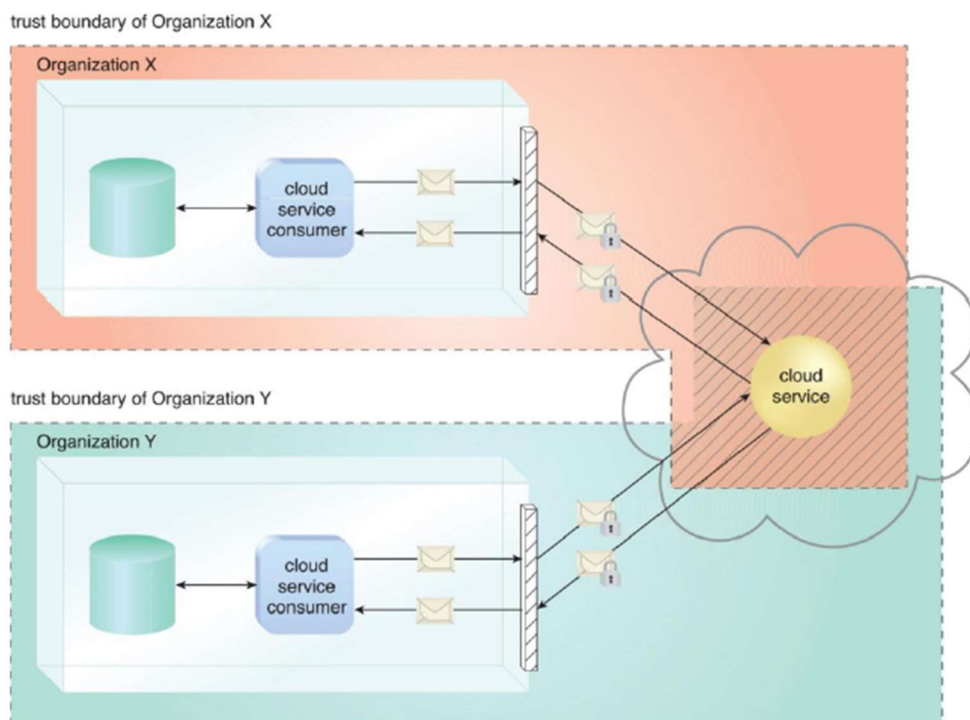
#### 3.4.2.1 Trust boundaries

Při přesunutí firemních dat do cloudu je nutné počítat se sdílením zodpovědnosti za bezpečnost dat s poskytovatelem cloudu. Využívání vzdálených IT zdrojů z cloudu vyžaduje rozšíření důvěryhodných hranic (trusted boundaries) ze strany uživatele cloudu tak, aby zahrnovaly samotný cloud i externí oblasti mimo organizaci. Navrhnout bezpečnostní architekturu, která by pokrývala takovou hranici důvěry, může být obtížné a může snadno vést k implementaci bezpečnostních zranitelností, pokud uživatelé cloudu a poskytovatelé cloudu nepodporují stejné nebo kompatibilní bezpečnostní rámce, což nemusí být u veřejných cloudů pravdou. (7)

Další důsledek spojených důvěryhodných hranic se týká privilegovaného přístupu poskytovatele cloudu k datům uživatele. Míra, do jaké jsou data zabezpečena, je omezena bezpečnostními kontrolami a politikami stanovenými jak uživatelem cloudu, tak

poskytovatelem cloudu. Navíc mohou vznikat překrývající se hranice důvěry mezi různými uživateli cloudu, protože IT zdroje založené na cloudu se často sdílejí.

Tyto překrývající se hranice důvěry a zvýšená expozice dat poskytují dalším uživatelům cloudu příležitosti k útokům na IT zdroje a k odcizení nebo poškození firemních dat. Pokud dvě organizace využívají stejnou cloudovou službu, jsou nuceny rozšířit své důvěryhodné hranice směrem k cloudu, což vytváří překrývající se hranice důvěry. (7)



Obrázek 4 - Trust boundary, zdroj. (7)

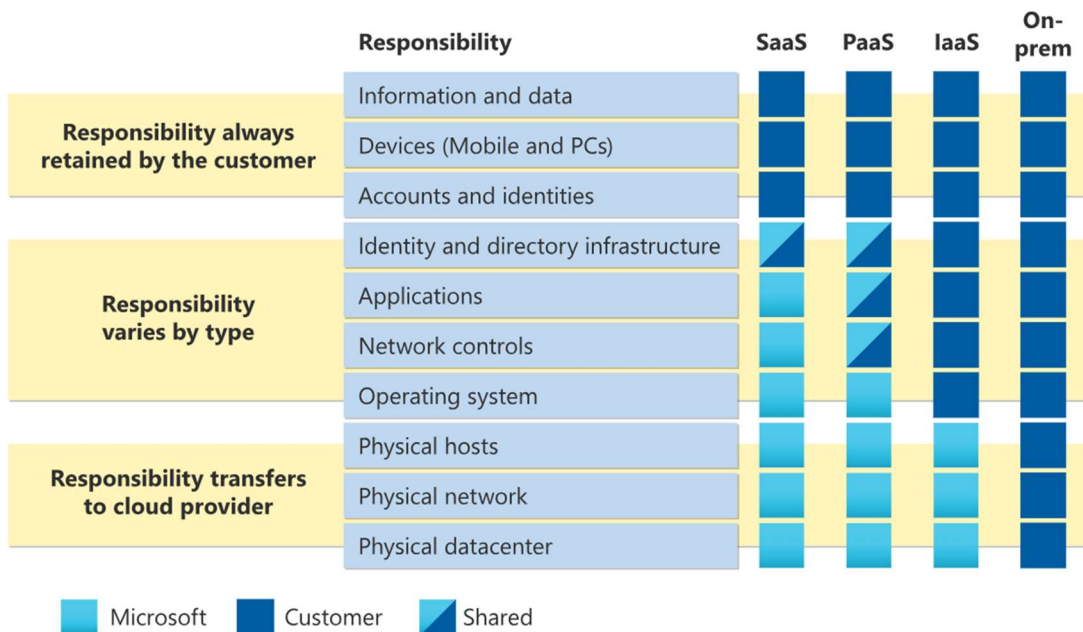
### 3.4.2.2 Sdílená odpovědnost za zabezpečení

Model sdílené odpovědnosti určuje, jaké bezpečnostní úkoly jsou na straně poskytovatele cloudu a jaké na straně zákazníka.

V následujícím textu si popíšeme responsibility model (SRM) pro MS Azure, AWS a Google Cloud Platform.

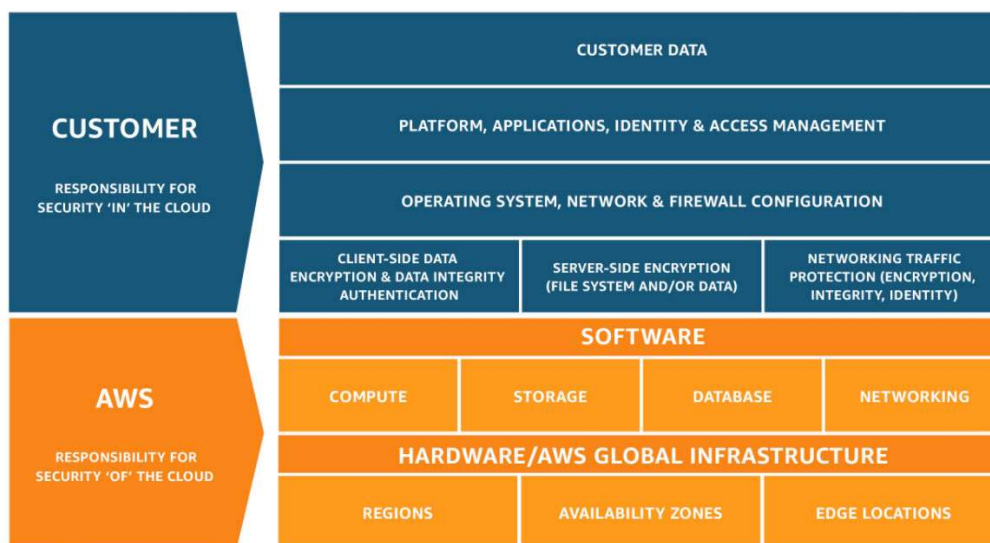
V případě Azure je při modelu IaaS Microsoft zodpovědný za zabezpečení "fyzické cloudové infrastruktury", zatímco klienti jsou zodpovědní za zabezpečení svých dat a aplikací v cloudu. Jinými slovy, zatímco Azure zajišťuje bezpečnost fyzických serverů, sítí a operačních systémů, zákazníci musí zajistit bezpečnost svých aplikací, dat a uživatelských

účtů. V modelech SaaS a PaaS jsou některé body sdílené společně, avšak v případě Azure má klient odpovědnost za data, koncové body a účty. (16)



Obrázek 5- Azure - Sdílená odpovědnost, zdroj: (16)

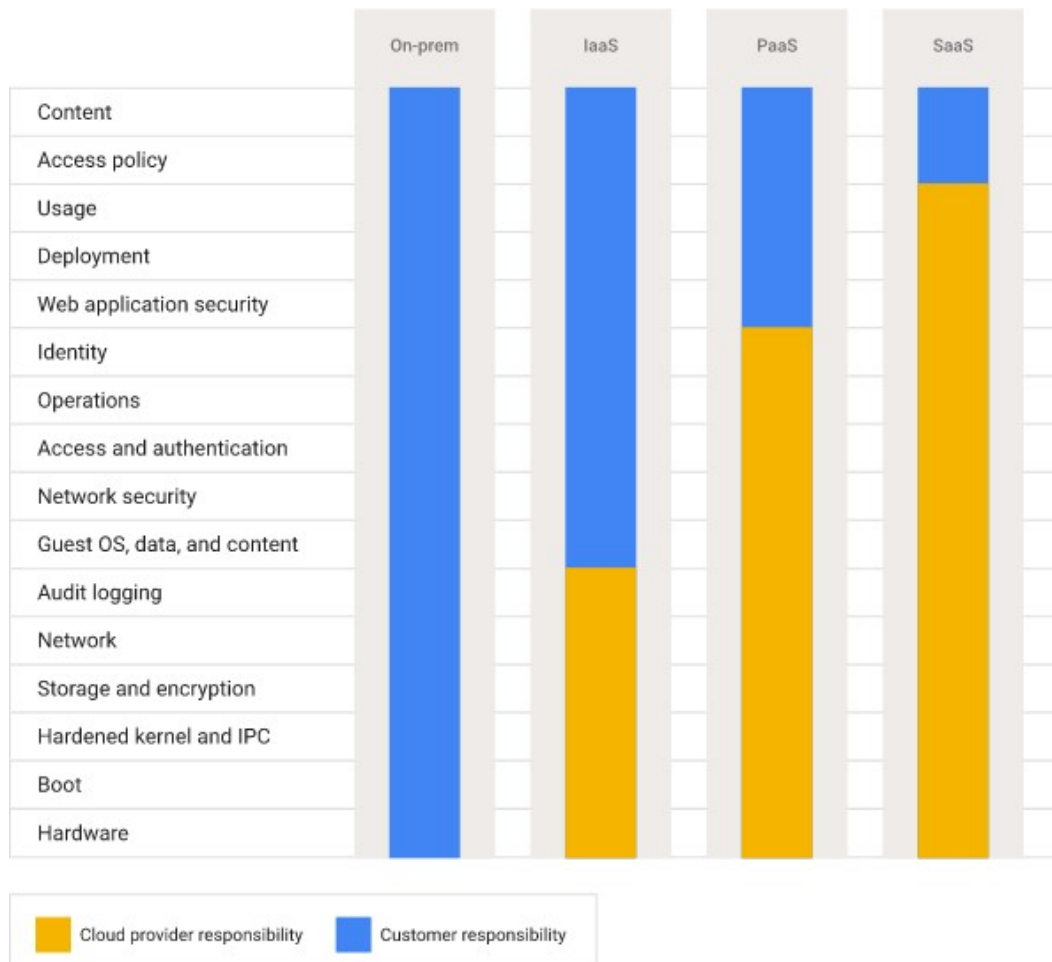
AWS zajišťuje bezpečnost cloudové infrastruktury, včetně fyzických zařízení a sítí. Zákazníci nesou odpovědnost za bezpečnost svých systémů, aplikací, identit nastavení sítí a dalších konfigurací v cloudu. Povinnosti se liší podle využívaných služeb a platných právních předpisů v dané lokalitě. (17)



Obrázek 6 - AWS Sdílená odpovědnost, Zdroj: (17)



Google dle svých slov přistupuje k modelu sdílené odpovědnosti s filozofií sdíleného osudu, čímž je myšleno, že Google a klienti společně spolupracují na zajištění bezpečnosti cloudových služeb. Google zajišťuje bezpečnost infrastruktury a operačních systémů, zatímco zákazníci jsou odpovědní za bezpečnost svých dat, aplikací a identit. Tento model zdůrazňuje důležitost spolupráce a transparentnosti mezi Googlem a jeho uživateli pro účinnou ochranu cloudových prostředků. (18)



Obrázek 7 - GCP Sdílená odpovědnost, Zdroj: (18)

U modelu sdílené odpovědnosti je potřeba pamatovat na několik bodů, které mohou představovat úskalí:

Klienti musí mít jasno v tom, za co jsou zodpovědní, a musí mít dostatečné znalosti a dovednosti k zajištění svých aplikací a dat.

Pokud klienti nesplní své bezpečnostní povinnosti, mohou se vyskytnout bezpečnostní problémy. Například, pokud zákazník nesprávně nakonfiguruje nastavení

zabezpečení, mohou být data nechráněná. Je důležité si uvědomit, že model sdílené odpovědnosti neznámá, že bezpečnost je zajištěna automaticky, a že bez ohledu na to, kterého poskytovatele cloudových služeb si zákazník vybere, je stále nezbytné se aktivně zajímat, spravovat a monitorovat bezpečnost svých dat a aplikací v cloudu. (7) (17) (16)

### **3.4.3 Ztráta kontroly nad infrastrukturou**

Jeden z hlavních benefitů, proč přejít do cloudu, je odstranění potřeby starat se o fyzickou infrastrukturu. Tato výhoda však má i stinnou stránku, klient se tím připraví o možnost ovlivňovat, například kdy provozovatel provádí aktualizaci jednotlivých fyzických komponent, či jak přistupuje k zabezpečení obecně. Další problém může být v nedodržení SLA ze strany cloudového poskytovatele pohledu dostupnosti služeb. Na to je sice možné se pojistit rozdělením služeb například do více regionů apod, či jejich replikací do nich, avšak ne zadarmo a navíc, pokud se jedná o kritickou infrastrukturu, i krátkodobý výpadek může mít dalekosáhlé následky. Pokud se server, který klient využívá pro svou aplikaci/službu nachází ve větší vzdálenosti, může se klient setkávat s problémy ve vyšší latenci, která v konečném součtu s ostatními komponentami zvyšující latenci, může zapříčinit problémy s potřebnou dostupností. (7)

### **3.4.4 Multiregionalita a právní otázky**

Poskytovatelé obvykle stavějí datová centra na pro ně ekonomicky dobře cenově vycházejících lokalitách a spotřebitelé nemusí být vždy informováni o přesné poloze datového centra, kde jsou jejich data uložena. Toto může představovat problém v rámci legislativy, kdy například důvěrné informace, týkající se zdravotních záznamů či dalších citlivých problematik, musí být dle zákonu některých zemí uchovávány jen v rámci určitých oblastí. I bez právních závazků je na pováženou, zdali mohou být data uložena v zemích, které k ochraně soukromí přistupují volněji než Evropská unie (například USA či Čína), pokud se jedná například o vysoce senzitivní data. (7) (19)

### **3.4.5 Přesáhnutí budgetu**

Cloud díky svým možnostem spočívajícím v podstatě neomezených zdrojích, které jsou klientovi okamžitě k dispozici, přináší však i úskalí, a to v podobě překročení finančních plánů pro dané řešení. To se může stát z různých důvodů, nekontrolovaných nákladů, neodebrání předjednaných závazků, plýtvání v rámci dev/tes prostředí (které funguje, i když jej nikdo nevyužívá) a bezúčelně přebujelého produkčního prostředí (20). V případě možného předpokládání potřeb vyřízení pro jednotlivé zdroje nabízí jednotliví poskytovatelé masivní slevy v případě předplacení služby na pevně stanovenou dobu. Tento přístup se však hodí pouze pro řešení, u kterých dokážeme s vysokou pravděpodobností předvídat potřeby do budoucna, například díky jednoúčelnosti, či na základě předchozích zkušeností a nehodí se například pro nové aplikace, které ještě nebyly dostatečně otestovány.

Další oblast, která je v této problematice podstatná, je správné nastavení škálování zdrojů, ať už vertikální či horizontální. Díky tomu lze efektivněji pracovat se zdroji tak, jak je zapotřebí. Je však třeba mít na paměti, že pokud bude špatně nastaveno škálování, společně s ochranou proti dlouhodobému vyřízení aplikace, například při útoku typu DDOS, dojde k maximálnímu vyškálování aplikace z důvodu útoku a pokud nebude zaznamenán včas, či nebude nastavena ochrana proti takovému scénáři, může dojít k výraznému překročení plánovaného budgetu pro dané řešení.

## **3.5 Analýza cloudových platforem**

### **3.5.1 Přehled cloudových poskytovatelů**

V posledních letech se cloud computing vypracoval na klíčový prvek digitální transformace, který firmám umožňuje rychlou škálovatelnost jejich IT zdrojů a inovace s nižšími počátečními náklady, oproti on-premise řešení. Adaptace cloudových služeb může vést k větší efektivitě, flexibilitě IT ve firmách. V této sekci se zaměříme na přehled nejpopulárnějších cloudových poskytovatelů, kteří hrají klíčovou roli v poskytování široké škály cloudových služeb pro různorodé případy použití.

Mezi hlavní poskytovatele cloudových služeb patří Amazon Web Services (AWS), Microsoft Azure a Google Cloud Platform (GCP), kteří dohromady zabírali ke konci roku 66% trhu. Každý z nich nabízí specifické služby, nástroje a technologie, které je odlišují a definují jejich postavení na trhu. AWS jako průkopník v oblasti cloudových služeb nabízí

rozsáhlý ekosystém služeb uspokojující širokou škálu podnikových potřeb. Microsoft Azure, s pevnou integrací do rozsáhlého spektra produktů Microsoftu, je ideální volbou pro firmy, které jsou již začleněny do Microsoft ekosystému. Google Cloud Platform se vyznačuje ve sférách jako analýza velkých dat, strojové učení a kontejnerizace, což ji může činit preferovanou volbou pro firmy hledající inovace v těchto oblastech. (21)

V následující části se podrobněji zaměříme na klíčové charakteristiky, služby a strategické přednosti jednotlivých hlavních poskytovatelů cloudových služeb, pro ucelený přehled současné situace na trhu s cloudovými službami a pochopení, jak tyto platformy mohou podporovat podnikové cíle.

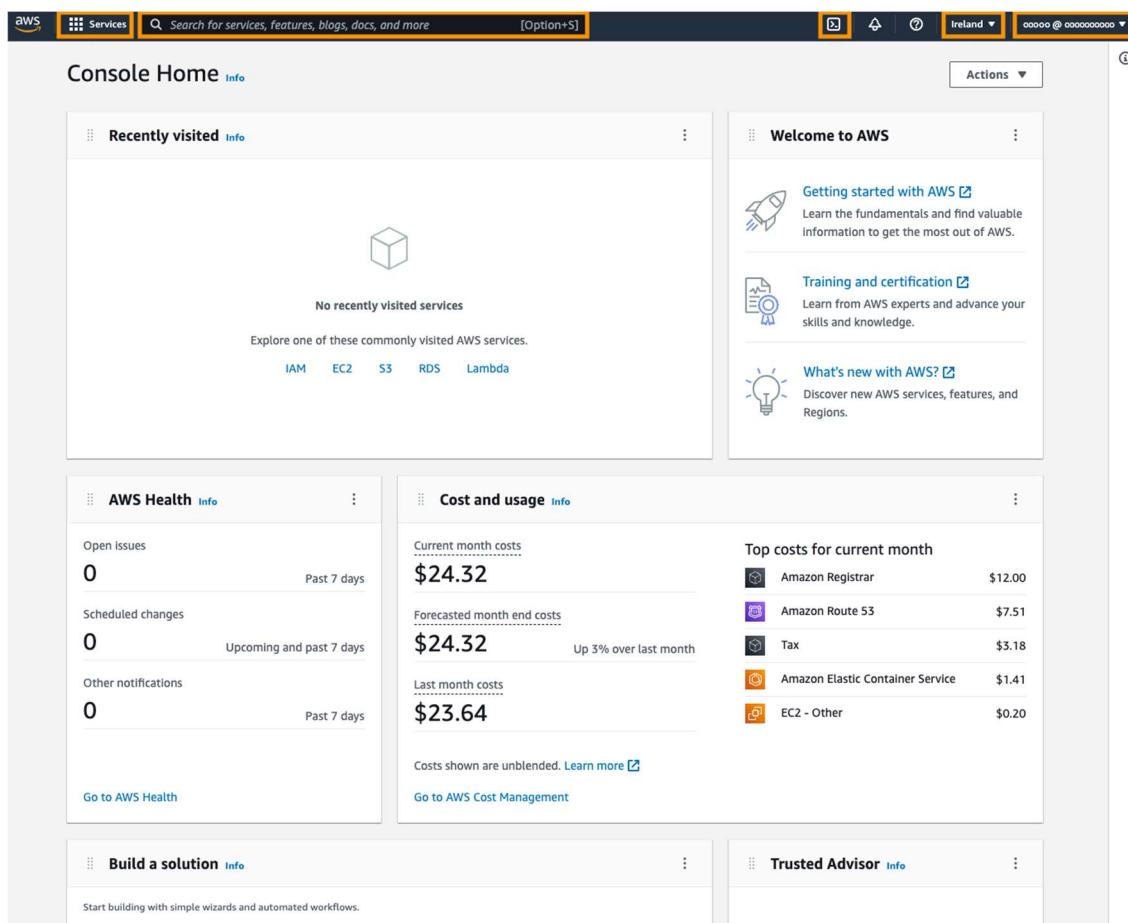
### **3.5.2 Amazon Web Services (AWS):**

Firma Amazon na začátku nultých let jednadvacátého století uvažovala, jak umožnit třetím stranám lépe spolupracovat s Amazon e-commerce enginem. Při plnění úkolu zpřehlednit, zdokumentovat práci se zdroji a navrhnout použitelné API, vytvořil tým, který dostal na starost tuto změnu, standardizované vývojářské nástroje, které urychlily vytváření potřebných zdrojů, bez nutnosti kompletně nového návrhu, a které později posloužily jako základ pro AWS. (22)

V roce 2006 firma Amazon spustila první veřejně dostupný Amazon Elastic Compute Cloud (zkráceně Amazon EC2), který umožňoval vytváření výpočetních kapacit, použitelných v rámci minut od vytvoření (23). Touto událostí byla zahájena revoluce, kterou postupem času následovaly další firmy, jako je Microsoft, Google, IBM apod. Díky své startovní pozici a opožděnému nástupu konkurence si Amazon se svým AWS udržuje prvenství v podílu na trhu (1).

### 3.5.2.1 AWS management console

AWS Management Console je webové rozhraní, které slouží ke správě služeb a zdrojů v rámci Amazon Web Services. Umožňuje uživatelům snadno přistupovat, konfigurovat a monitorovat jejich v rámci AWS z jednoduchého, uživatelsky přívětivého webového rozhraní. Console umožňuje uživatelům spravovat všechny své AWS služby a zdroje z jednoho místa, což zjednodušuje administraci. AWS management console nabízí možnost přizpůsobení konzole podle uživatelských potřeb, například možnost přidávání, odebrání a uspořádávání widgetů pro rychlý přístup k často používaným službám. V případě preference rozhraní typu command line je možné AWS ovládat skrze nástroj AWS CLI přímo v rámci AWS Management Console. (24)



Obrázek 8- AWS ukázka Management console, zdroj: AWS (24)

### 3.5.3 Microsoft Azure:

Microsoft Azure je cloudová služba, která nabízí přes 600 cloudových služeb a nespočet databází, operačních systémů a vývojářských nástrojů. Výhodou společnosti Microsoft může být její rozsáhlý ekosystém využívaný korporátní klientelou a s tím související integrace do ostatních služeb, které Microsoft poskytuje. Stejně jako jeho konkurenti poskytuje své služby také v režimu pay-as-you go, což umožňuje větší přístupnost klientům, kteří nemají dostatečný kapitál pro nákup vlastních serverů, či kteří chtějí mít lepší přehled o své útratě (25). Služba dnes známá jako Microsoft Azure byla oznámena v říjnu roku 2008 pod názvem Project Red Dog, jako přímá odpověď na Amazon EC2, která byla vydána o dva roky dříve. Již v době oznámení služba AWS pracovala s virtuálními stanicemi používající OS ze strany Microsoftu. V roce 2010 byla spuštěna služba nesoucí název Windows Azure. (26)

Spuštění tohoto produktu znamenalo faktický vstup společnosti Microsoft na trh cloud computingu. Počáteční služby nabízené při spuštění služby Windows Azure zahrnovaly výpočetní funkce a funkce pro správu zdrojů a zaměřovaly se na poskytování škálovatelné, vysoce dostupné a trvalé služby pro ukládání dat aplikací a provozování aplikací v cloudu. Úvodní spuštění platformy se soustředilo na poskytnutí nové možnosti pro vývojáře, kteří mohli vytvářet cloudové aplikace s využitím technologií společnosti Microsoft. (26)

Byly dány na odiv především funkce, jako je přístup ke vzdálené ploše, dynamické ukládání obsahu do mezipaměti a bezpečné doručování obsahu pomocí protokolu SSL, kterými se snažila firma Microsoft prezentovat výkonost své platformy. Platforma byla rozdělena na klíčové části zahrnující samotný systém Windows Azure, SQL Azure a Windows Azure AppFabric, z nichž každá hraje jedinečnou roli ve fungování cloudových služeb. (27)

Významnou součástí spuštění byly služby úložiště Azure storage, které poskytovaly škálovatelná a trvanlivá úložná řešení. Tyto služby zahrnovaly tzv. bloby (pro ukládání velkého množství nestrukturovaných dat), fronty (pro ukládání velkého množství zpráv) a tabulky (pro ukládání strukturovaných dat ve formě entit a vlastností). Tyto služby byly navrženy tak, aby podporovaly širokou škálu aplikačních dat a případů použití, od uživatelských profilů a informací o relacích až po velkoobjemové internetové služby. (27)

Počáteční hodnocení výkonu systému Windows Azure poukázala na relativně dobrý výkon, ačkoli některé oblasti, jako je doba spuštění virtuálních strojů a výkon SQL Azure,

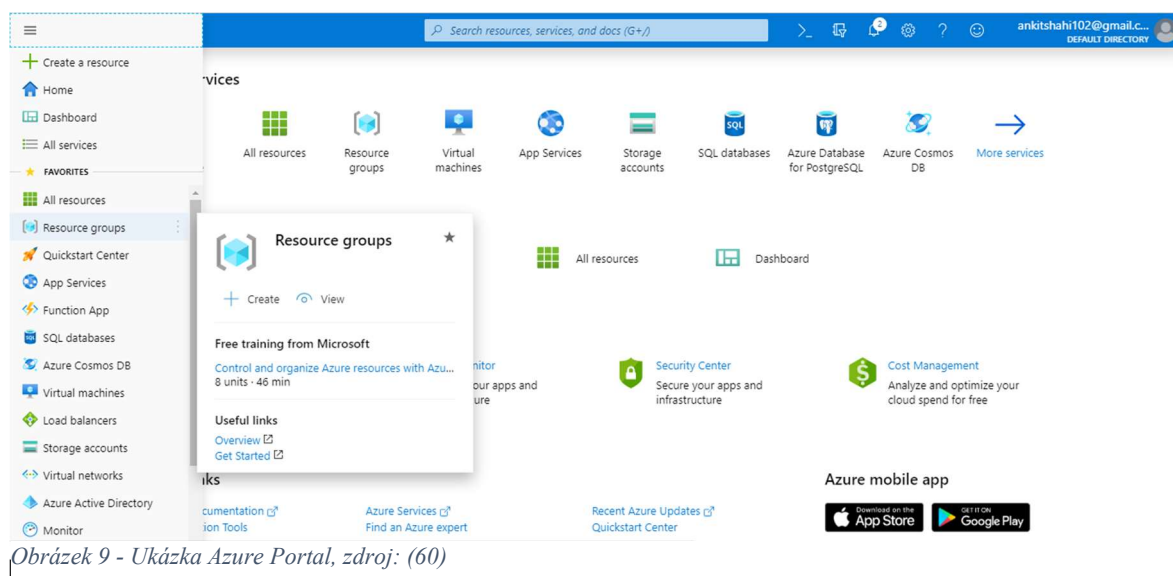
bylo třeba brát v potaz při návrhu aplikací brát. Ranné doporučení pro službu Windows azure se zmiňuje o vhodnosti využití jeho služeb zejména pro aplikace používané pro výzkum, či pro obecné využití. (28)

Obecně vzato uvedení platformy Windows Azure na trh bylo přijato kladně a platforma byla oceněna pro svůj potenciál poskytovat škálovatelné a flexibilní prostředí pro vývoj a hostování aplikací v cloudu. Jako u každé nové technologie však byly identifikovány oblasti, které je třeba zlepšit a rozvíjet, zejména pokud jde o výkon a rozsah nabízených služeb. Od té doby se platforma významně vyvíjela a v reakci na zpětnou vazbu uživatelů a požadavky trhu se rozšířila o širokou škálu služeb a funkcí. (27) (28)

V roce 2014 byla služba přejmenována na dnes používaný název Microsoft Azure. (26)

### 3.5.3.1 Azure Portal

Azure Portal je centrálním bodem pro správu široké škály služeb Azure, včetně virtuálních strojů, databázových služeb, kontejnerů a mnoha dalších. Uživatelé mohou snadno vytvářet, konfigurovat a monitorovat své služby z jednoho místa. Azure podporuje využití vlastních Azure Resource Manager šablon a Azure Automation pro automatizaci opakovaných úkolů a vytváření nových zdrojů. Azure Portal umožňuje přizpůsobení jak hlavní stránky, tak u jednotlivých zdrojů pomocí dashboardů, které lze individuálně upravit. (29)



Obrázek 9 - Ukázka Azure Portal, zdroj: (60)

### 3.5.4 Google Cloud Platform (GCP)

Google Cloud Platform (GCP) je společně s AWS a MS Azure jedním z hlavních lídrů v oblasti cloudového computingu, nabízející širokou škálu cloudových zdrojů od virtuálních stanic, přes kubernetes, nástroje strojového učení, k Big data a analýze a spoustě dalších oblastí. (30) Společnost Google je považována za jednoho z předních technologických lídrů v oblasti internetových služeb s produkty jako Gmail, YouTube a Google Maps a v neposlední řadě vyhledávač Google, které každodenně zpracovávají kvanta dat od uživatelů z celého světa. Google v průběhu let výrazně investoval do infrastruktury pro zpracování a ukládání dat a dnes se může pochlubit některými z nejimpresivnějších designů datových center a technologií na světě, které podporují jeho výpočetní požadavky a výpočetní služby. GCP umožňuje veřejnosti využívat tyto výkonné výpočetní zdroje pro návrh a vývoj modelů strojového učení jako je GCloud ML Engine (30)

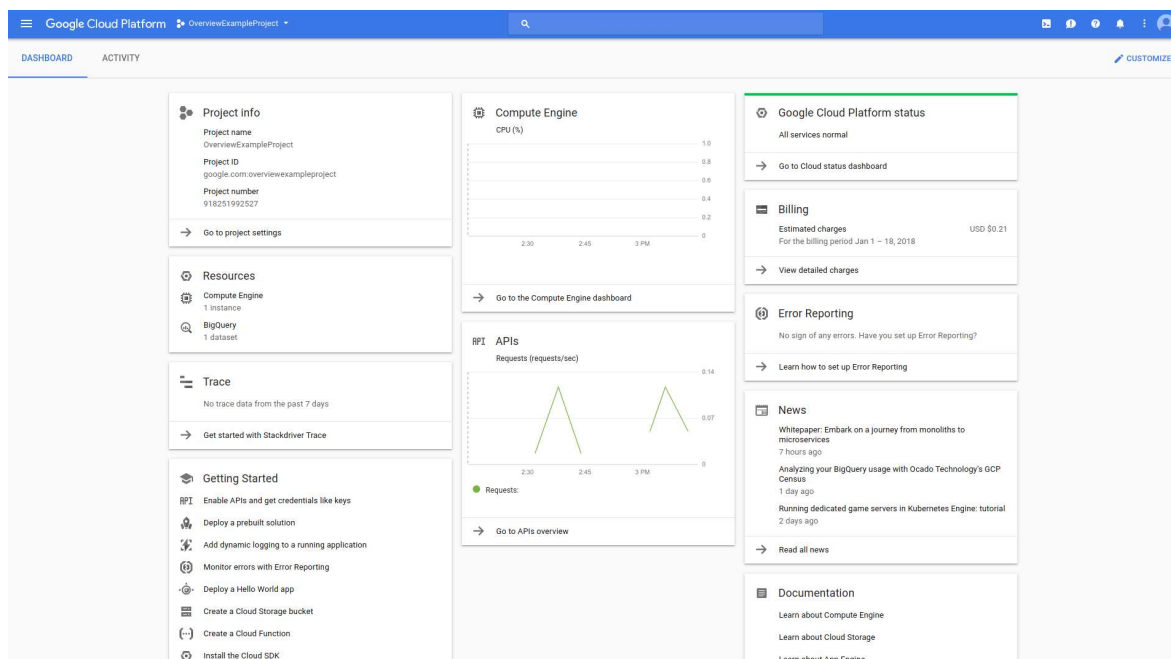
Google Cloud platforma byla spuštěna v roce 2008 jako reakce na nastupující trend veřejného cloudu spuštěného několik roků předtím společností Amazon a jejím AWS. Zpočátku GCP sloužil se svým Google App engine (fungujícím na principu SaaS), jako hostování služba pro webové aplikace na infrastruktuře společnosti Google. V následujících letech přibýly služby jako je Google Compute Engine pro virtuální stanice, Google Kubernetes engine a Google BigQuery pro big data a spoustu dalších. Na konci roku 2023 se Google Cloud Platform držel na pomyslném třetím místě za AWS a Azure, s 11% podílu na trhu s rostoucí tendencí (31) (1).

#### 3.5.4.1 Google Cloud Console

Google Cloud Console je webové rozhraní, které umožňuje uživatelům spravovat služby poskytované Google Cloud Platformou (GCP). Poskytuje přístup k široké škále cloudových služeb, které GCP nabízí svým klientům.

Console se odlišuje od konkurentů jako je AWS či MS Azure důrazem na čistotu a jednoduchost, s cílem umožnit uživatelům rychle nalézt a spravovat své zdroje.





Obrázek 10 - Ukázka Google cloud console, zdroj: (59)

### 3.5.5 Ostatní poskytovatelé

#### 3.5.5.1 IBM Cloud

IBM Cloud, dříve známý pod jménem SoftLayer, byl založen v roce 2005 a v roce 2013 byl akvírován společností IBM. Tato platforma byla jednou z prvních, která nabídla služby bare-metal compute, ještě předtím, než se tímto směrem vypravily další velcí poskytovatelé cloudových služeb jako je Amazon AWS, GCP, či Azure. IBM Cloud je zaměřený na poskytování cloudových služeb pro velké podniky a organizace, či menší vývojářské týmy po celém světě. (32) (33)

IBM Cloud nabízí širokou škálu funkcí a služeb, které jsou rozděleny do několika hlavních kategorií, včetně výpočtu v cloudu, sítí, úložiště a nástrojů pro vývojáře. Mezi ty patří možnost výběru mezi bare-metal s virtuálními servery, podpora pro populární programovací jazyky, tvorbu databází, využití AI a strojového učení, nasazení řešení založených na blockchain technologii a silné zabezpečení. IBM Cloud také používá přístup pay-as-you-use (totožný u ostatních uživatelů používaných pay-as-you-go) pro fakturaci, což umožňuje efektivnější řízení nákladů. (32) (33)

### 3.5.5.2 Oracle Cloud Infrastructure

Služba Oracle Cloud infrastructure je jedním z menších hráčů na poli cloud computingu (1). Poprvé byla služba veřejnosti zpřístupněna v říjnu 2016, tehdy pod názvem "Oracle Bare Metal Cloud Services". V roce 2018 poté služba změnila svoje pojmenování na dnešní název Oracle Cloud Infrastructure. (34)

Oracle Cloud Infrastructure (OCI) nabízí bohatou sadu cloudových služeb, jako jsou Infrastruktura jako služba (IaaS), Platforma jako služba (PaaS), Software jako služba (SaaS) a Data jako služba (DaaS). Služby poskytované firmou Oracle jsou designovány tak, aby podporovaly výkonné, škálovatelné a bezpečné cloudové prostředí pro nejrůznější aplikace a databáze. Oracle Cloud nabízí několik specifik, jako je SLA na úrovni až 99,99%, či specializaci na hostování databází od společnosti Oracle. (35)

## **4 Praktická část práce**

Praktická část této práce se bude zabývat analýzou stávajícího řešení, výběrem nejvhodnějšího poskytovatele pro převod aplikace do cloudové služby a následnou implementaci. V prvních kapitolách praktické části bude provedena analýza současného stavu ve zvolené firmě, která se zabývá vývojem softwaru a správou cloudové infrastruktury. Dále budou popsány důvody potřeby migrace, očekávané přínosy a následně bude navrhnout plán provedení migrace.

### **4.1 Analytická část**

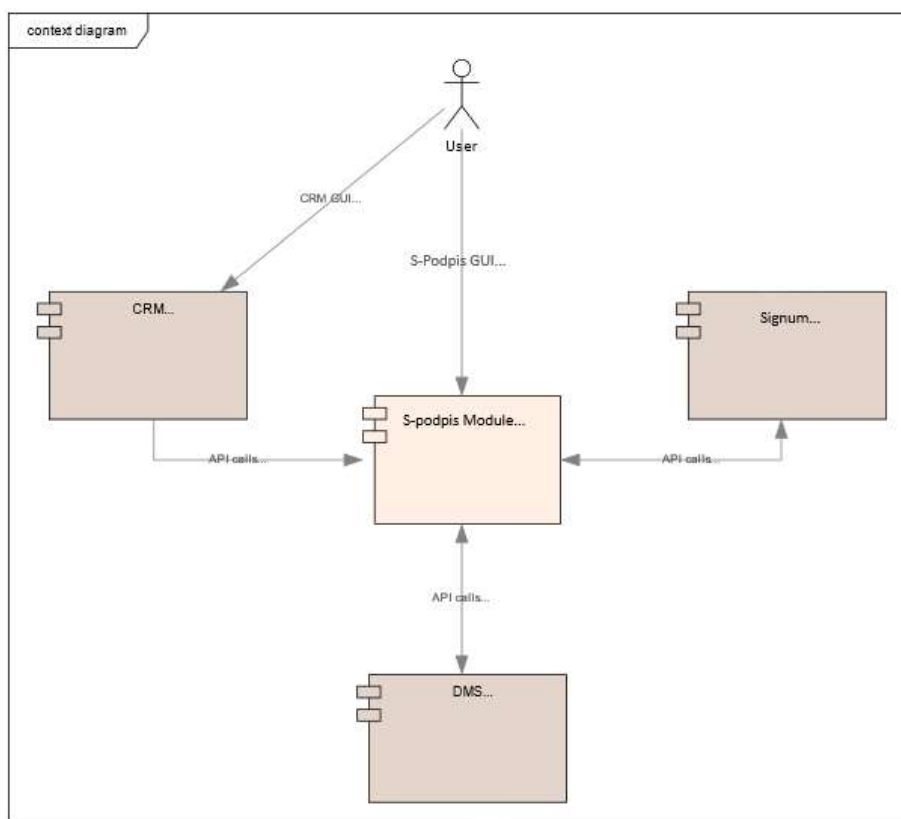
#### **4.1.1 Popis Aplikace**

Aplikace S-Podpis slouží jako podpisová aplikace dokumentů v rámci organizační struktury firmy, která umožňuje elektronické podpisování dokumentů a správu smluvních procesů v digitálním prostředí, které mohou být obdařeny certifikátem. S aplikací S-Podpis mohou uživatelé rychle a bezpečně podepisovat dokumenty z mobilního zařízení, kdykoli a kdekoli.

Aplikace podporuje vícekrokové schvalovací workflow, integraci s dalšími podnikovými systémy, a poskytuje auditní záznamy pro každou transakci, což zajišťuje vysokou úroveň dodržování interních i externích regulačních požadavků. S S-Podpis mohou firmy značně zvýšit efektivitu svých operací, minimalizovat papírování a urychlit procesní cykly, což může přinést potenciální nákladové úspory a zlepšit celkovou spokojenost uživatelů/zaměstnanců.

#### **4.1.2 Analýza stávajícího řešení**

Na základě komunikace se stakeholdery, technickými vlastníky a čerpáním z technické dokumentace aplikace byly zjištěny následující skutečnosti o stavu migrované aplikace a infrastruktury, která slouží k jejímu provozu.



Obrázek 11- S-Podpis monolitická architektura, vlastní zpracování

Zkoumané řešení je monolitická aplikace napsána v Javě, správa a aktualizace aplikace jsou ručně prováděny skrze Apache Tomcat. Aplikace funguje na serveru s Windows Server ve verzi 2012, který není ke konci roku 2023 již plně podporovaný v rámci standardní licence (36). Z hardwarového pohledu se jedná o starší DELL R330 8SFF s procesorem Intel Xeon E3-1220 8 GB. Tento server v tuto chvíli obsluhuje pouze aplikaci S-Podpis a po její migraci je plánováno jeho vyřazení jak s ohledem na jeho současné nevyužití, tak se stářím hardwaru a operačního systému.

Aplikace dále komunikuje s databázovým serverem, sloužící jako dočasné uložení dokumentů při jeho zpracování, který je řešen skrze PostgreSQL ve verzi 10.6, běžící na databázovém serveru, který funguje na virtuální stanici s Windows Server 2019 OS, na hardware HP DL380 GEN10.

S-Podpis se skládá z několika modulů, které jsou napsané v programovacím jazyku JAVA, grafické uživatelské rozhraní, část zajišťující načítání certifikátů z klientské stanice, backend, zpracovávající požadavky na podepsání dokumentů, konektor na databázi fungující na PostgreSQL, která slouží na vytváření záznamu o prováděné činnosti

s jednotlivými soubory a modul, starající se o konverzi XFA šablon a flattening zpracovávaných dokumentů. Aplikace S-Podpis komunikuje skrze definované API s několika externími systémy. Jedním z nich je CRM systém, sloužící pro vytváření dokumentů, které je třeba posléze obdařit elektronickým podpisem. Dále externím DMS systémem, sloužícím ke správně již podepsaných dokumentů a externího podepisovacího systému Signum, sloužící podepsání dokumentu

### **Výhody současné monolitické architektury**

- Jednoduchý deployment – pouze jedna aplikace
- Výkonnost – bez latence mezi jednotlivými moduly
- Snazší debugging – je jen jedna aplikace, kterou je třeba debugovat

### **Nevýhody současné monolitické architektury**

- Špatná škálovatelnost, jak aplikace, tak týmu, který se na vývoji podílí
- Selže-li jedna část může to mít dopad na zbytek aplikace
- Vysoká komplexita systému
- Náročnější testovatelnost
- Silné závislosti v celém systému
- Riziko výpadku celého systému při nasazování nové verze

#### **4.1.2.1 Důvody migrace do cloudu a očekávané přínosy**

V rámci specifikací byly při diskuzi se stakeholdery určeny důvody, proč firma plánuje migraci své webové aplikace do jednoho ze tří největších poskytovatelů veřejných cloudů.

Níže jsou uvedeny jednotlivé body:

- **Zvýšení flexibility a škálovatelnosti:** Cloudové prostředí nabízí výrazně větší flexibilitu a škálovatelnost oproti tradičním on-premise řešením. Aplikace S-Podpis je vytěžována na nepravidelné bázi, tudíž není zapotřebí maximální možný výkon infrastruktury po celou dobu. Z budoucího hlediska je perspektivní možnost upscalování infrastruktury v případě jejího rozšíření o další funkce, či navýšení počtu podepisovaných dokumentů, bez nutnosti počáteční investice.
- **Zlepšení výkonnosti a dostupnosti:** Očekávaným přínosem migrace aplikace S-Podpis do cloudu je mimo jiné navýšení potencionálního výkonu a případné

dostupnosti bez závislosti na lokální on-prem infrastruktuře, což zaručí fungování aplikace, i v případě výpadku lokální sítě v zázemí firmy, kdekoliv na světě

- Zpřehlednění a potenciální snížení provozních nákladů: Správa současné vlastní infrastruktury z pohledu společnosti nenabízí využitelnost, je v tuto chvíli používána pouze pro účely udržení aplikace S-Podpis v provozu. Společnost si od převodu aplikace slibuje snížení nákladů oproti fungování na on-premise řešení také kvůli nekontinuální vytiženosti fyzického hardwaru a nutnosti starání se o server ze strany IT personálu. Cloudové služby fungují na modelu "pay-as-you-go", což umožňuje organizacím platit pouze za zdroje, které skutečně využívají, od čehož si firma slibuje úsporu nákladů. Oproti on-premise je také snazší určit, kolik takováto aplikace bude stát, z důvodů, jako je nenutnost rozpočítávat náročnost údržby fyzického HW, platbu za licence, elektřinu apod. Všechny tyto platby jsou zahrnuty v jedné platbě pro dodavatele řešení a tím je z principu zvýšena přehlednost.
- Zabezpečení a dodržování předpisů: Z popsaného stavu v analýze stavu současné infrastruktury je patrné, že současné řešení infrastruktury je již nedostačující také z bezpečnostního hlediska a pro navýšení bezpečnosti by bylo zapotřebí investic ze strany společnosti, pro nákup nových serverů a licencí pro další používání. Oproti tomu cloudoví poskytovatelé investují značné zdroje do zabezpečení a dodržování regulačních standardů, což může poskytnout lepší ochranu dat, než co mohou organizace často zajistit samy.
- Zjednodušení správy a údržby: Jedním z dalších poznatků při vytváření očekávaných přínosů je snadnější správa a údržba aplikace a infrastruktury za pomoci nástrojů, které cloudové služby nabízejí – nástroje a služby pro automatizaci a zjednodušení správy a údržby aplikací.

### 4.1.3 Výběr migračního modelu

#### 4.1.3.1 Specifikace požadavků pro migraci

Ve spolupráci se stakeholdery, vlastníky a aplikačním týmem byly nalezeny jednotlivé požadavky pro migrovanou aplikaci. Požadavky budou v následujících částech rozděleny na relevantní pro obsah této práce, model migrace a výběr cloudového poskytovatele.

- Použití CI/CD
- Možnost přidávání dalších modulů do aplikace
- Oddělené GUI od aplikace
- Využití původního kódu
- Definované a měřitelné SLA
- Provozní a bezpečnostní monitoring
- Škálovatelnost aplikace
- Využití managed a serverless services

#### 4.1.3.2 Popis migračních modelů

Best practice určuje několik modelů, které lze jsou používány při plánování migrace aplikace (a její infrastruktury) do cloudového prostředí z on-premise řešení v závislosti na možnostech a požadavcích klienta migrované aplikační infrastruktury.

Jedná se o níže zmíněné modely známé také jako 7R (37):

- **Rehost** – Přenos aplikace do cloudového prostředí beze změn
- **Replatform** – Převod aplikace do cloudu s menšími úpravami funkčnosti
- **Re-Architecture** – Přeprocování architektury aplikace pro zvýšení cloudové vhodnosti (Kontejnerizace některých services apod.)
- **Repurchase** – Nahrazení aplikace aplikací, která je již plně přizpůsobena cloudu
- **Retire** – Ukončení používání aplikace (například není aplikace již zapotřebí, případně její funkci zastane aplikace, která pokrývá funkčnost migrované aplikace)
- **Retain** – Ponechání staré aplikace v provozu
- **Rebuild** – Znovuvytvoření aplikace tak, aby plně vyhovovala požadavkům pro cloudové prostředí.

#### 4.1.3.3 Výběr migračního modelu na základě specifikace

Ze specifikace požadavků firmy pro migraci aplikace S-Podpis připadají v úvahu možnosti Rehost, Replatform, Re-Architecture a Rebuild, které v následující části budou popsány z pohledu jejich funkčnosti ve vztahu k aplikaci S-Podpis. Z pohledu vyřazených modelů Model Retire není aplikovatelný, jelikož ve společnosti není aplikace, která by mohla aplikaci nahradit, model Repurchase nepřipadá v úvahu z potřeby nabízet aplikaci potenciálním klientům a Model Retain neumožňuje již z principu nutnosti aplikaci zmigrovat ze současného hardware a dalších výhod, které se od provedení migrace očekávají.

Následně bude za pomoci vícekriteriální analýzy proveden výběr vhodnosti jednoho z přístupů pro migraci aplikaci do cloudu.

#### **Rehost (Lift and shift)**

O Modelu rehost také známý jako lift and shift se dá říci, že se jedná o nejjednodušší model z výše zmíněných modelů 7R. Ve stručnosti proces migrace aplikace nebo workloadu z on-premise infrastruktury do cloudového prostředí bez významnějších změn konfigurace, kódu aplikace, či infrastruktury. Tento postup je nevíce vhodný v případě, že migrace vyžaduje co nejrychlejší a nejméně pracný způsob přenosu aplikace z on-premise infrastruktury do cloudu. Taková jednoduchost přináší s sebou ale určitá úskalí. Z důvodu přenesení aplikace ve stejném stavu, v jakém byla v on-premise, není zpravidla možné, pokud se jedná o aplikaci typu legacy (aplikace není například vytvořena pro fungování s kubernetes a není kontejnerizovaná), využití výhod škálovatelnosti a využít managed/serverless technologií. V praxi je také běžné, že takovýto přesun do cloudu neznamena ani úsporu financí na provozu takovéto aplikace. (37).

Metoda splňuje pouze podmínky pro možnost využití CI/CD technologií pro nasazování aplikace/infrastruktury a potřeby nevytvářet aplikaci od znovu.

#### **Replatform**

Metoda Replatform v kontextu migrace do cloudu značí modifikování stávající aplikace/infrastruktury do cloudového prostředí. V porovnání s výše zmíněnou metodou Rehost se jedná o pokročilejší způsob migrace, který oproti přesunu skrze Rehost umožňuje více možností využití výhod cloudu (37). Jedním z takových, které by šly využít v aplikaci



S-Podpis je využití managed a serverless services, v tomto případně pro databázové řešení, které zbavují potřebu starat se o databázový server, jako je Amazon RDS, Azure Cosmos, či Cloud SQL. V případě Replatform, by v tomto případně mohlo být uvažováno o umožnění škálování aplikace za pomoci nastavení virtual machine scale a napojení na loadbalancer, avšak bylo by zapotřebí v případě zvolení tohoto postupu možnost ověřit v souvislosti s touto aplikací. Ani tento model však neumožňuje využití modulární aktualizace, s výjimkou případné aktualizace databáze, která by mohla být řešena separátně skrze PaaS. Jelikož v rámci aplikování tohoto modelu není možné větších zásahů do fungování aplikace, není také možné oddělit GUI aplikace do separátního modulu. (37).

### **Rearchitecture**

Metoda Rearchitecture v sobě skrývá přetvoření původní aplikace způsobem, aby více vyhovovala potřebám pro maximalizaci využití cloudových služeb a s tím související infrastruktury s využitím již napsaného kódu. V rámci tohoto modelu je aplikace rozdělena na jednotlivé komponenty a ty jsou posléze využity z původní aplikace, či nahrazeny vhodnějším řešením (37). V souvislosti s aplikací S-Podpis se nabízí oddělit grafický interface, aby lépe využila cloudové služby a infrastrukturu. Cílem je optimalizovat aplikaci pro cloud, což může zlepšit výkon, škálovatelnost a odolnost. Tento přístup často zahrnuje dekompozici monolitických aplikací na services, či její kontainerizaci, přepracování databázových vrstev a integraci s cloud-native službami. Přestože může být nákladnější, nabízí dlouhodobější výhody ve flexibilitě a efektivitě.

Při modelu Rearchitecture je možné oproti předchozímu modelu Replatform navíc uvažovat o oddělení GUI a rozdělení aplikace na jednotlivé moduly, které je dále možné rozdělit do kontainerů, případně na služby, při částečné možnosti využití kódu původní aplikace.

### **Rebuild**

Metoda Rebuild v souvislosti s migrací aplikace do cloudu obsahuje kompletní přepracování, či znovuvytvoření aplikace v cloudovém prostředí s využitím nativních cloudových technologií jako jsou mikroservisy, dynamické škálování, serverless architektury a další nativní cloudové služby. Strategie nabírá smyslu především u aplikací, u kterých pozbývá smyslu je přesouvat do cloudu jednou z výše uvedených metod, ať už z důvodu časové, finanční náročnosti, potřeby nových funkcí, či potřeby přidání cloud base

funkcí. Díky této metodě je možné docílit výrazně nižší ceny, avšak za cenu vyšší počáteční časové a finanční investice. Pro aplikaci S-Podpis by bylo tedy zapotřebí kompletně přepracovat její kód s pomocí týmu developerů tak, aby byla kompatibilní s cloudem. (37)

#### 4.1.3.4 Zhodnocení výběru modelu

Na základě výše provedené analýzy byla sestavena vícekriteriální analýza variant. Z této analýzy lze vyčíst, že jako nejvhodnější model pro migraci se v tomto případě stal model Rearchitecture, který splňuje nejvíce položek.

Model Rehost se jeví jakožto nejsnazší pro provedení migrace, avšak přináší minimum benefitů očekávaných od provedení migrace do cloudu.

Model Replatform neumožňuje díky svému principu v přepracování kódu aplikace pro obsahování Odděleného GUI, a také nepřeprogramování aplikace do modulů.

Model Rebuild sice v sobě skrývá všechny požadované benefity, které jsou od aplikace očekávány, zvolení tohoto modelu by však znamenalo vytvořit aplikaci od znovu a s tím zvýšená počáteční investice do vývoje a nasazení aplikace, což dle zadání není žádoucí.

Požadavky\Modely migrace	Rehost	Replatform	Rearchitecture	Rebuild
Využití – CD/CI technologií	X	X	X	X
Škálovatelnost aplikace	-	X	X	X
Odlišné GUI pro klienty	-	-	X	X
Nevytvářet aplikaci od znovu	X	X	X	-
Aktualizace aplikace po modulech	-	-	X	X
Využití managed a serverless services	-	X	X	X

Tabulka 1- Vícekriteriální analýza – Výběr migračního modelu, Vlastní zpracování

#### 4.1.4 Volba vhodného cloudového řešení

Do výběru vhodné cloudové platformy byli zvoleni tři největší poskytovatelé public cloudových služeb (31) AWS, Microsoft Azure, Google Cloud Platform, pro bližší porovnání vhodnosti převodu aplikace do jednoho z těchto poskytovatelů. V následující části budou specifikovány požadavky na platformu, které byly zjištěny ve spolupráci se stakeholdery, vlastníky a aplikačním týmem.

#### 4.1.4.1 Požadavky na Cloudovou platformu

V této části byly specifikovány jednotlivé požadavky na platformu, určena jejich relevance pro výběr platformy a následně bude analyzován požadavek vůči jednotlivým poskytovatelům.

##### **Níže jsou popsány jednotlivé požadavky:**

- Efektivní využití výpočetních zdrojů
- Integrace s Visual studio
- Connector pro Microsoft Sentinel
- Vyšší rychlost dodávky a nasazení nových verzí
- Využívání platformy ve firmě
- Bezplatná verze clusteru pro developerské prostředí
- Zlepšená bezpečnost izolací aplikací od hostitelského systému a od sebe navzájem
- Rychlejší spuštění aplikace a snadnější škálování
- Infrastructure as Code (deklarativní deployment platformy)
- Lokace umístění serverů v Evropské unii
- Zlepšení obchodní kontinuity

##### **Integrace s Visual studio**

V rámci společnosti je využíván nástroj Visual studio code pro programování a úpravu kódu. V souvislosti s migrací do cloudu by společnost preferovala využití již používaných technologií, bez nutnosti přechodu na jinou platformu.

Pro Visual studio code existuje značné množství plug-inů pro AWS, jedním z nich je AWS Toolkit for Visual Studio Code, vydávaný společností Amazon, který funguje na principu open-source a usnadňuje práci při vytváření/správě kódu pro prostředí AWS. (38) Dále AWS SAM (Serverless Application Model) CLI je nástroj, který umožňuje vývojářům jednoduše vytvářet, testovat a nasazovat serverless aplikace do prostředí AWS. Poskytuje sadu příkazů pro správu zdrojů, lokální testování, ladění a nasazení aplikací. (39).

Identicky jako Azure je Visual studio code vyvíjeno společností Microsoft. Při instalování doplňků jako je Azure tools či Azure Developer CLI, které jsou obě vyvíjeny společností Microsoft, je možné docílit větší integrace a snadnější práce při nasazování kódu do prostředí MS Azure (40) (41).

Google cloud platform, stejně jako jeho konkurenti, vyvíjí vlastní nástroje pro spolupráci s Visual studio code tak, aby zajistil co nejvyšší možnost jeho využití. Jedním z nich jsou Cloud code a GCloud CLI, které jsou alternativami k výše zmíněným nástrojům od Amazonu a Microsoftu. (42) (43).

V rámci analýzy lze konstatovat, že všechny doplňky nabízejí obdobné funkce se specializací na jednotlivá prostředí Rozšíření Cloud Code, Azure Tools a AWS Toolkit pro Visual Studio Code. Cloud Code poskytuje širokou podporu pro vývoj aplikací Kubernetes a Google Cloud. Azure Tools se zaměřuje na služby Azure jako jsou Azure Functions a Web Apps, zatímco AWS Toolkit usnadňuje práci s AWS službami jako Lambda a CloudFormation.

### **Connector pro Azure Sentinel**

Microsoft Sentinel je cloud-native platforma poskytována v rámci MS Azure a slouží k zabezpečení cloudového prostředí za pomoci přístupů SIEM a SOAR. Tento nástroj je určen pro správu bezpečnosti prostředí za pomoci inteligentních bezpečnostních analýz, dokáže reagovat a provádět automatizované prošetřování potenciálních hrozeb s využitím umělé inteligence a díky automatizovaným odpovědím dokáže na tyto hrozby reagovat v reálném čase. Sentinel dokáže monitorovat celé prostředí MS Azure napříč všemi uživateli, zařízeními, aplikacemi a infrastrukturou, a i v on-premise infrastruktuře (44).

Za pomoci nativních connectorů je možné napojit Microsoft Sentinel na AWS Google cloud platform pro jejich využití. (44) Je tedy možné Microsoft Sentinel využít pro všechny v této práci zvažované platformy.

### **Lokace umístění serverů v centrální Evropě**

Volba umístění v krajině blízké České republice spočívá jak v předpokládané nižší latenci, tak v předcházení případných komplikací při zpracování osobních dat mimo EU (19). AWS pokrývá k březnu 2024 celkem 26 Regionů, z toho 8 v Evropě a konkrétněji 2 v rámci centrální Evropy (Frankfurt a Curych). Oba disponují třemi Availability Zones, které umožňují případné nastavení redundance v rámci regionu.

Microsoft Azure aktuálně disponuje 62 regiony, z toho 11 v Evropě a 4 (bez započtení Government only regionů) v oblasti centrální Evropy (Polsko, Německo, Švýcarsko a region s názvem Západní Evropa). Stejně jako tomu je u AWS, všechny obsahují 3 Availability zóny (45).

Google cloud platforma disponuje k výše zmíněnému období celosvětově 40 zón, z toho 12 v Evropě a 4 v centrální Evropě. Všechny taktéž disponují třemi zónami.

Z výše provedené analýzy lze konstatovat, že všechny platformy splňují podmínku možnosti lokace serverů v rámci oblasti centrální Evropy.

### **Využívanost platformy**

V rámci zkoumané společnosti jsou využívány pro vývoj a správu aplikací prostředí AWS a MS Azure. U AWS a MS Azure není tedy zapotřebí dalšího školení pro platformu pro jednotlivé účastníky převodu. Oproti tomu GCP je ve zkoumané společnosti neznámou platformou, u které by bylo zapotřebí vyškolení architekta a DevOps specialisty pro lepší porozumění. V modelovém příkladu může být zvolen kurz od společnosti SkillUp (46) v rozložení 7-8 h týdně po dobu 11 týdnů. Tento počet tedy odpovídá zhruba na 11 Mandays, které by byly zapotřebí k zaškolení zaměstnance pro porozumění architektuře GCP. Pokud bychom počítali s teoretickou cenou 10 000 korun českých za jeden Manday, dostali bychom se na částku 110 000 Kč za zaškolení jednoho člověka, bez započtení ceny kurzu.

V tomto požadavku tedy nejlépe vychází využít jedno již známých prostředí. Pokud by však bylo uvažováno o migraci prostředí, u kterého se očekává velká výpočetní náročnost, či více projektů, bylo by na zváženu, zda i přes nutnou počáteční investici nezhodnotit také prostředí GCP.

### **Bezplatná verze clusteru pro developerské prostředí**

V rámci dalšího vývoje aplikace je plánováno vytvoření a následné využívání testovacího prostředí, u kterého jsou preferovány ústupky ze strany výkonnosti i SLA za cenu nulového či minimálního poplatku za AKS, EKS GKE cluster. Z analýzy stránek poskytovatelů lze konstatovat, že služba AWS nenabízí volnou verzi svého EKS, služba AKS pro Azure poskytuje volnou licenci pro developerské prostředí s clusterem s počtem nodů pod 10 a GCP nenabízí svou službu plně zdarma, zato ale nabízí pro svůj základní tier kredit ve formě 74\$ či 50\$ v závislosti na zvoleném modelu GKE. Tyto kredity mohou být využity pouze na GKE a nepřenesují se do plateb za jiné služby v rámci platformy. (47) (48) (49)

## **Infrastructure as code**

Pro řešení Infrastructure as Code (IaC) nabízí každý ze zvažovaných poskytovatelů vlastní řešení této problematiky. AWS vytváří své CloudFormation, Azure má Azure Resource Manager (ARM) šablony, a GCP používá Cloud Deployment Manager (50). Tyto nástroje umožňují definovat a spravovat infrastrukturu prostřednictvím kódu, což zvyšuje efektivitu a reprodukovatelnost. Všechny platformy navíc podporují využití deklarativního konfiguračního jazyka Terraform (50).

### 4.1.4.2 Zhodnocení požadavků na platformy

V předchozí části byly zanalyzovány požadavky dané pro výběr cloudové platformy. Některé požadavky, jako je efektivní využití výpočetních zdrojů, či rychlejší spouštění aplikace a další, nebyly detailně analyzovány z důvodu jejich přílišné obecnosti a nesouvislosti s výběrem platformy, na kterou bude aplikace migrována.

Ze získaných poznatků lze konstatovat, že jednotlivé platformy mají mnoho společného a až na jednotlivá specifika se dají vzájemně nahradit. Nevýhodou Google cloud platformy z pohledu níže popsané vícekritériální analýzy je především neznámost prostředí v rámci společnosti, což by mělo za důsledek potřebu se nové technologii doučit a s tím spojené náklady na čas a práci.

Řešení AWS je oproti GCP ve společnosti známé, avšak nenabízí možnost clusteru zdarma, který by byl využíván pro testovací prostředí, jenž by zadavatel preferoval pro ověřování funkčností nových verzí aplikace.

Služba Microsoft Azure splnila všechny výše specifikované podmínky a bude vybrána pro následující návrh převedení aplikace do cloudového prostředí.

	<b>AWS</b>	<b>GCP</b>	<b>AZURE</b>
<b>Servery v centrální Evropě</b>	<b>X</b>	<b>X</b>	<b>X</b>
<b>Známost prostředí ve firmě</b>	<b>X</b>	<b>-</b>	<b>X</b>
<b>Bezplatná verze clusteru</b>	<b>-</b>	<b>X</b>	<b>X</b>
<b>Infrastructure as code</b>	<b>X</b>	<b>X</b>	<b>X</b>
<b>Integrace s Visual studio</b>	<b>X</b>	<b>X</b>	<b>X</b>
<b>Connector pro Azure Sentinel</b>	<b>X</b>	<b>X</b>	<b>X</b>

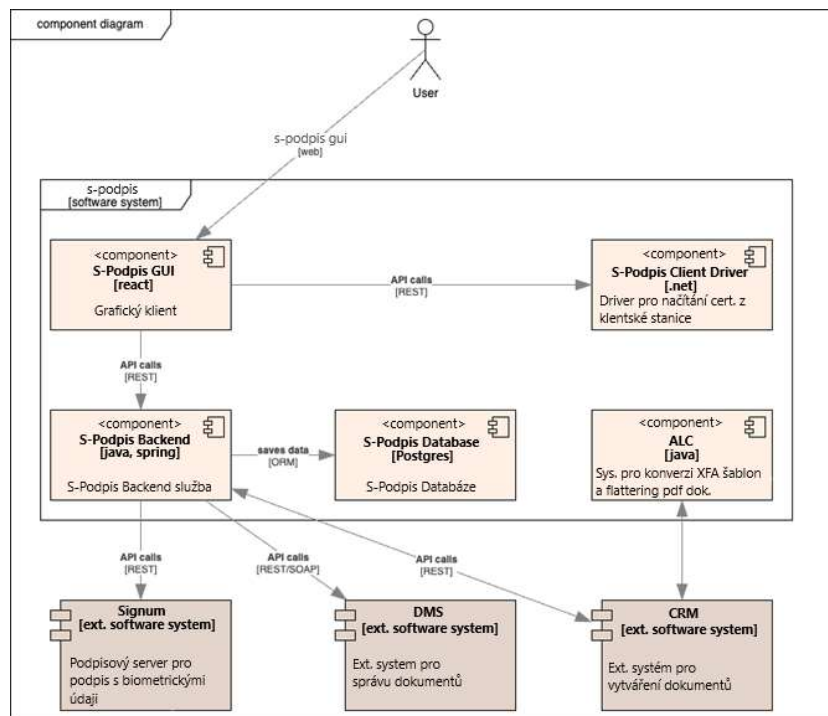
*Tabulka 2 - Vícekriteriální analýza – Výběr cloudové služby, Vlastní zpracování*

## 4.2 Implementace

Na základě informací získaných o aplikaci, chtěném migračním modelu a vybrané migrační platformě v analytickém segmentu praktické části bude v následující sekci zpracován plán přepracování aplikace do cloudového prostředí s návrhem implementace řešení do cloudového prostředí.

### 4.2.1 Nová architektura

Na základě poznatků získaných při tvorbě předchozích částí praktické části byla ve spolupráci se stakeholdery, vlastníky aplikace a technické dokumentace k aplikaci aplikace rozdělena na jednotlivé komponenty a byla navržena architektura vycházející z modelu Rearchitecture.



Obrázek 12 - S-podpis – Diagram komponent, vlastní zpracování

Aplikace byla rozdělena na komponenty GUI, Client driver, Backend a ALC.

S-Podpis GUI je nově vytvořená komponenta běžící v samostatném kontejneru obsluhující grafický interface pro práci s aplikací ze strany uživatele, je napsána ve frameworku React. Tento framework byl vybrán díky svým vlastnostem pro jednoduchý vývoj rozhraní a zkušenostem v rámci development týmu.



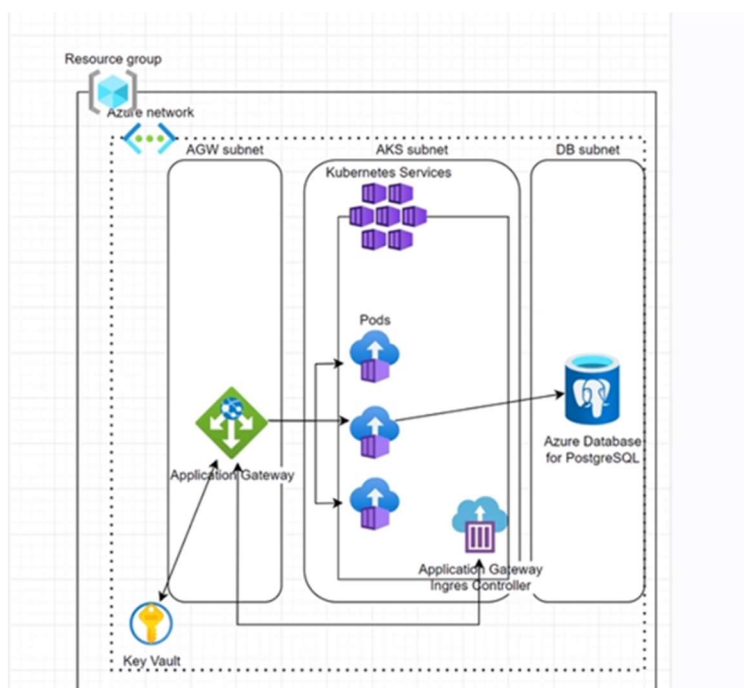
S-Podpis GUI dále komunikuje přes API Rest s komponentou Client driver, která slouží pro načtení klientských certifikátů a Backendem (API REST), který zpracovává data a jednotlivé dokumenty a je založený na původním kódu aplikace.

S-Podpis databáze zaznamenává jednotlivé úkony a slouží pro jednotlivé úkony provedené v rámci aplikace. Pro migrační model Rearchitecture byl zvolen model PostgreSQL, který funguje na principu SaaS a oproti on-prem verzi je plně spravován ze strany MS Azure.

Komponenta ALC slouží ke konverzi XFA šablon a flatteringu PFD dokumentů. Aplikace v rámci své funkce komunikuje s několika externími systémy, jeden z nich je aplikace Signum, které je dodáváno externí společností a funguje jako podpisový server pro podpisy s biometrickými údaji. Jednotlivé zpracované dokumenty jsou spravovány za pomoci externího DMS systému. Modul CRM je externí systém, který slouží pro vytváření dokumentů, ve své funkčnosti komunikuje s ALC a s S-Podpis Backendem.

#### 4.2.2 Architektura pro MS azure

V této části diplomové práce bude zpracován návrh infrastrukturové architektury pro prostředí Microsoft Azure, popsány konkrétní komponenty. V následující části bude za pomoci služby Azure Pricing calculator provedena analýza cenové náročnosti navržené architektury.



Obrázek 13 – Azure Architektura S-Podpis, vlastní zpracování

Architektura byla navržena na jednu síť rozvrženou na tři subnety, které jsou rozděleny podle jejich účelů (jeden pro aplikační Gateway, jeden pro AKS a jeden pro databázi) a tři veřejné IP adresy.

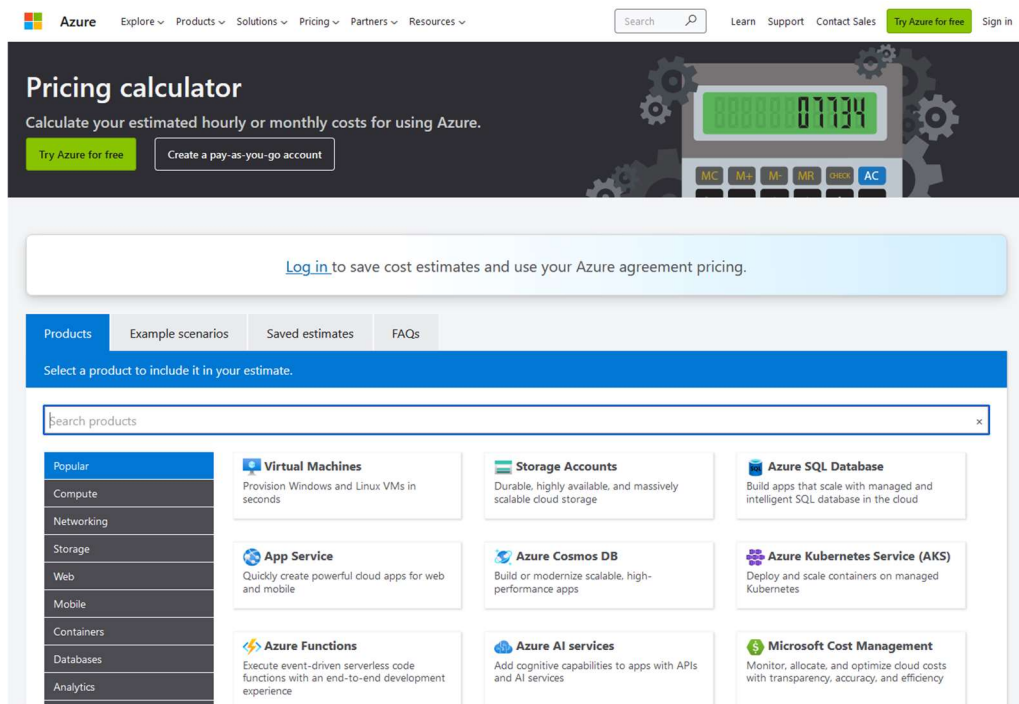
V rámci AKS subnetu se skrývá Kubernetes server fungující na virtuálním stroji. Jednotlivé pody (kontejnery) fungují v rámci výše zmíněného AKS, pro které je zapotřebí Azure Key Vault ve spojení s Application Gateway slouží k centralizovanému ukládání, správě a ochraně kryptografických klíčů a secrets, jako jsou certifikáty SSL/TLS, které Application Gateway aplikuje na svou šifrovanou komunikaci s jednotlivými externími komponentami/prostředí. Tato integrace umožňuje bezpečné a efektivní spravování tajných klíčů a certifikátů, nezbytných pro komunikaci skrze zabezpečené webové aplikace a API. (51)

Azure Application Gateway Ingress Controller (AGIC) je použitý k automatické konfiguraci Azure Application Gateway tak, aby fungoval jako vstupní bod (ingress) pro aplikace v Azure Kubernetes Service (AKS). AGIC monitoruje změny v definicích ingress prostředků v AKS a aplikuje je na Application Gateway, která posléze zajišťuje správné směrování provozu v rámci AKS. (52)

### **4.2.3 Návrh infrastruktury pro Azure**

#### **4.2.3.1 Azure Pricing Calculator**

Azure Pricing Calculator je nástroj od Microsoftu, který umožňuje odhadnout náklady na prostředky a služby z prostředí Azure před jejich skutečným nasazením. Umožňuje přizpůsobit specifikace zdrojů a konfigurace služeb podle potřeb uživatele, či aplikace a poskytuje přehled očekávaných měsíčních výdajů. Pomocí tohoto nástroje mohou jednotlivci a společnosti plánovat svůj rozpočet a experimentovat s různými konfiguracemi, pro nalezení nejefektivnější konfigurace pro zamýšlené řešení. (53)



Obrázek 14 - Ukázka Prostředí Azure Pricing calculator, zdroj: (53)

#### 4.2.3.2 Výběr konfigurace a kalkulace cenové hladiny

V této části budou zvoleny konfigurace pro jednotlivé položky z návrhu infrastruktury a bude vypočítána jejich cenová náročnost. Všechny zdroje budou situovány do regionu West Europe.

#### **AKS**

Pro AKS (Azur Kubernetes Service) je třeba zvolit vhodnou virtuální stanici, na které bude AKS provozováno. Pro fungování aplikace S-Podpis byla zvolena virtuální stanice v konfiguraci Standard\_D2s\_v3, který je vhodný pro AKS, pokud je potřebný vyvážený výkon CPU a paměti pro běžnou pracovní zátěž. Standard\_D2s\_v3 má 2 vCPU a 8 GiB RAM, což je dobré pro menší nebo střední workloads a může sloužit jako výchozí bod pro cluster s možností škálování v budoucnu podle potřeb aplikace (54) (55).

#### **Application Gateway**

Pro aplikační gateway byla zvolena konfigurace Azure Application Gateway Standard\_v2, s kapacitou 2. Konfigurace Standard\_v2 přináší oproti basic vylepšený výkon,

autoscaling, a funkce jako je Web Application Firewall (WAF), URL-based routing, a SSL offload. Konfigurace je obecně vhodná pro scénáře s vyššími požadavky na výkon a zabezpečení. (56)

## **Sítě**

V rámci Microsoft Azure nejsou účtovány poplatky za virtuální síť v něm vytvořené. V rámci vybrané architektury bude však zapotřebí vytvoření tří veřejných IP adres, pro komunikaci aplikace s vnějším prostředím.

## **Databáze**

Pro databázi byl zvolen Azure Database for PostgreSQL, konfiguraci Flexible Server v tieru Burst s výpočetním modelem B1MS a 32 GB úložištěm v lokální redundanci (LRS). Zvolený model umožňuje efektivní řízení nákladů a adaptabilitu na proměnlivé zatížení, zatímco redundance LRS (lokálně redundantní úložiště) poskytuje vysokou dostupnost dat v rámci jednoho regionu. Toto nastavení je ideální pro vývojové prostředí nebo aplikace s nižším zatížením, které mohou těžit z možnosti výkonových špiček (bursting).

## **Key Vault**

Key Vault bude aplikací využíván pro zabezpečení komunikace. Neočekává se z jeho strany vysoká finanční nákladnost. Cena za 10 000 operací je v rámci regionu West Europe 0.03 dolaru.

## **Kalkulace**

Na základě vložených dat z Azure Pricing Calculator byla zhodnocena finanční náročnost nového prostředí. Celkové finanční náklady za jeden měsíc vychází na 498,223\$. Náklady jsou rozděleny na 5 kategorií. Nejvíce finančně náročnou položkou je kategorie Networking – Aplikační gateway s 201,48\$, dále Compute – AKS s 162,14\$, Databases se 120 \$, Networking – Public IP address 10,95\$ a Security – Key Vault za 3,06\$.

Kategorie	Typ služby	Očekávaná měsíční cena
Compute	Azure Kubernetes Service (AKS)	\$162,14
Networking	Application Gateway	\$201,48
Databases	Azure Database for PostgreSQL	\$120,60
Networking	IP Addresses	\$10,95
Security	Key Vault	\$3,06
		<b>\$498,223</b>

Tabulka 3 - Výsledek cenové kalkulace

#### 4.2.4 Migrace PostgreSQL

Migrovaná on-premise databáze funguje na technologii PostgreSQL, stejně tak, jako nová databáze v rámci MS Azure. Dle dokumentace, která je pro tento migrační scénář k dispozici na webu Microsoft Learn (57) a ze které bude vycházet následující sekce praktické části, je pro provedení migrace zapotřebí splnit několik prekvizit.

##### 4.2.4.1 Prekvizity

Nejprve je zapotřebí se ujistit, zdali je migrovaná databáze v minimální podporované verzi pro migraci, tou je PostgreSQL ve verzi 9.4, což verze 10.6., která je nasazena na on-premise databázi, splňuje a je možné pokračovat vytvořením zdroje Azure Database for PostgreSQL Flexible Server, ve kterém po jeho dokončení vytvoříme prázdnou databázi s názvem `azure_spodpis`.

V rámci dalšího postupu je zapotřebí nakonfigurovat síťové připojení, aby migrace mohla projít v pořádku a zajistit vzájemné routování sítě. Pro to lze použít například službu Azure VPN Gateway. Při konfiguraci virtuální sítě pro databázi je třeba mít na paměti, aby v rámci NSG (Network Security Group) pro virtuální síť pro cloudové PostgreSQL, nebyla blokována odchozí komunikace na portu 443 se ServiceTagem pro ServiceBus, Storage a AzureMonitor. (57)

Jako další krok je zapotřebí vytvoření přístupů na straně on-premise serveru, na kterém běží PostgreSQL databáze tak, aby byly povoleny přístupy na databázi skrze port 5432, nakonfigurovat VPN připojení na on-prem serveru v souladu s Azure VPN Gateway.

Je také třeba zajistit, aby účet na on-premise serveru měl roli, která umožňuje replikování databáze, pro to, aby správně mohla být provedena migrace a upravit postgresql.config file tak, aby obsahoval wal\_level = logical, max\_replication\_slots = 5 , max\_wal\_senders = 10, čímž zajistíme funkčnost replikace ze strany on-premise databáze. (57)

#### 4.2.4.2 Migrace sample schéma databáze

V následujícím kroku je nutné zmigrovat takzvané sample schéma, které nám umožní zmigrovat data databáze ve stejném formátu, jak tomu bylo na on-premise. Následující příkazy je potřeba spustit na on-premise databázi.

Jedná se o tyto formáty (57):

Vytoření dumpfile:

```
pg_dump -O -h hostname_migrovane_databaze -U db_username -d db_name_migrovana -s > nazev_schema.sql
```

Migrace dumpfile do cloudu:

```
psql -h hostname_kam_migruji -U db_username -d db_name_kam_migruji <nazev_schema.sql
```

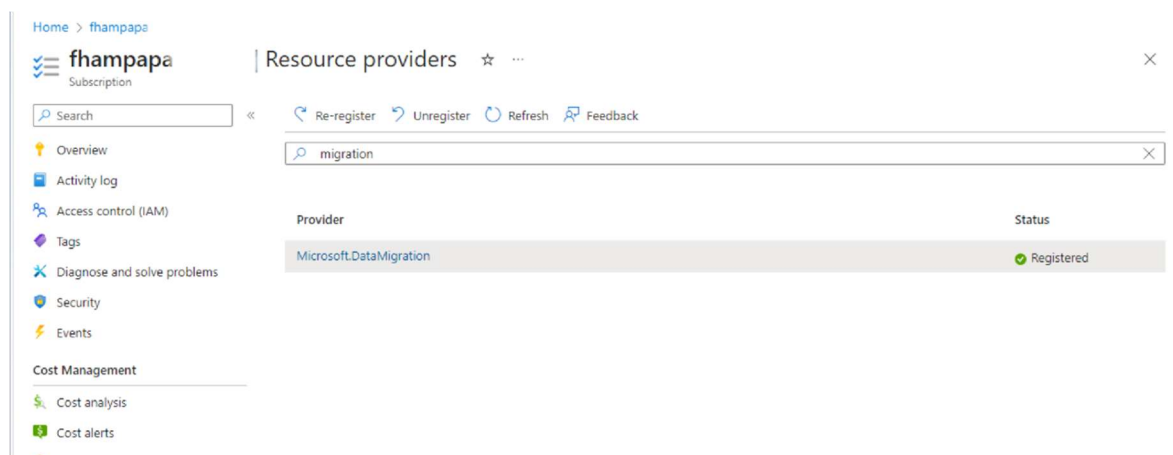
Pro migrovanou databázi se jedná o následující příkazy následovně:

```
pg_dump -O -h localhost -U postgre_admin -d spodpis_db -s > listdb_migrace.sql
```

a

```
psql -h spodpisdb.postgres.database.azure.com -p 5432 -U adminus azure_spodpis<listdb_migrace.sql
```

### 4.2.4.3 Resource Provider



Obrázek 15 - Převod databáze – Azure Resource provider, vlastní zpracování

V subskripci, kde se nalézá databáze, do které bude migrována on-premise databáze S-Podpis ověříme, zdali je zaregistrovaný resource provider Microsoft.DataMigration. V obrázku níže lze zpozorovat, jak vypadá, pokud je resource provider Microsoft.DataMigration povolen.

### 4.2.4.4 Data migration service

V dalším kroku je třeba vytvořit Azure Database Migration Service. To je možné učinit skrze azure CLI, nebo skrze Azure portal. Při jeho vytváření je třeba specifikovat, o jaký typ migrace se jedná (z databáze typu PostgreSQL do Azure Database for PostgreSQL). Pro migraci PostgreSQL je zapotřebí zvolit Pricing tier Premium.

#### Create Migration Service

Basics Networking Tags Review + create

Azure Database Migration Service is designed to streamline the process of migrating on-premises databases to Azure. [Learn more.](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups as you would use folders, to organize and manage all of your resources.

Subscription \*

Resource group \*  [Create new](#)

**Instance details**

Migration service name \*

Location \*

Obrázek 16- Převod databáze – Vytváření migration service v Azure, vlastní zpracování

V následujícím kroku je zapotřebí zvolit síť, kterou jsme nakonfigurovali dle zadání v přechozí části tak, aby byla zajištěna prostupnost mezi on-premise a cloudem pro přenos dat a následovně je zapotřebí stvrdit vytvoření service.

## Create Migration Service ...

Basics **Networking** Tags Review + create

Select an existing virtual network or create a new one.

**i** Select from a list of existing virtual networks. Click on the links to see more details about the selected virtual network. [Learn more.](#)

spodp

↑↓	Name	↑↓	Resource group	↑↓	Gateways	↑↓	Connections	↑↓
t...	spodpis_vn/default		spodpis_db		Network without gateway			

**i** Create a new virtual network by entering a name below. This will create a basic VNET that can connect to source servers with public facing IPs. You can then take additional steps to upgrade this network and increase your connectivity options. [Learn more.](#)

Virtual network name

Obrázek 17 - Převod databáze – Vytváření migration service v Azure– Volba sítě, vlastní zpracování

### 4.2.4.5 Vytvoření migračního projektu

S použitím migration service, která byla vytvořena v předchozí části, vytvoříme nový migrační projekt, ve kterém proběhne po jeho konfiguraci samotná migrace databáze.

## New migration project ...

A database migration project is a group of database activities that you can migrate together.

**Migration project name**

Project name \* ⓘ

**Choose your source and target server type.**

Source server type \* ⓘ

Target server type \* ⓘ

**Choose your migration activity type.**

Migration activity type \* ⓘ

Obrázek 18 - Převod databáze – Vytváření Migration service v Azure, vlastní zpracování



Je zvoleno vytvoření nového projektu a zadány parametry ohledně migrace z PostgreSQL do Azure Database for PostgreSQL.

Po zadání přihlašovacích údajů pro on-premise databázi je uživatel vyzván k zadání detailů ohledně přihlašovacích údajů pro databázi, která byla vytvořena v předchozích krocích jako destinace pro migrovaná data.

V další části vytváření migračního procesu je zvoleno, jaké položky budou z databáze migrovány a které tabulky v nich obsazené a následně může, po potvrzení dalších skutečností začít samotná migrace.

#### 4.2.4.6 Monitorování migrace

V rámci procesu migrace databáze je možné monitorovat uběhlý čas a aktuální status migrace. Jakmile je migrace dokončena, jsou zobrazeny uživateli informace ohledně celkového průběhu migrace. Následně uživatel ukončí migraci za pomoci tlačítka Start Cutover.

## **5 Výsledky a diskuse**

### **5.1 Zhodnocení vhodnosti prostředí**

Při výběru migračního modelu vyšlo z výsledků vícekritériální analýzy variant jako nejvhodnější použití modelu Rearchitecture, které umožňuje převod aplikace do cloudového prostředí s maximálním využitím výhod cloudu, bez nutnosti aplikaci kompletně přepracovat a vystavět od znovu.

Ze zjištěných skutečností lze konstatovat, že tři zkoumané platformy jsou si navzájem podobné a rozhodující pro výběr jedné z nich byla známost prostředí cloudové platformy ve společnosti a specifikum potřeby pro zdarma dostupný cluster pro vývojové prostředí aplikace.

Na základě návrhu původního řešení aplikace bylo navrženo řešení nové a byl proveden návrh infrastrukturního řešení v prostředí MS Azure.

Vycházením z navrženého řešení byl poté spuštěn testovací provoz aplikace v Microsoft, kterým se potvrdila funkčnost návrhu. Navržené řešení lze tedy považovat za funkční.

Na základě návrhu migrace databázového řešení PostgreSQL byla úspěšně provedena migrace.

### **5.2 Analýza nákladů**

Na základě návrhu popsaného v této práci byla vytvořena infrastruktura v prostředí Microsoft Azure s očekávanou cenou \$498,22 za jeden měsíc provozu aplikace v navrženém prostředí. Testovacím provozu po dobu používání aplikace se ukázalo, že cena provozu navrhovaného řešení zhruba odpovídá reálným nákladům s výkyvem kolem 5%.

### 5.3 Doporučení

Na základě zjištěných skutečností lze konstatovat, že proces je aplikovatelný pro převod aplikace typu on-premise do cloudového prostředí. Navrhované prostředí v sobě skrývá spoustu možností pro další upgrade zdrojů, pokud by při reálném provozu nebyly navržené zdroje jako dostatečné, či byly nadhodnocené. S případným zvýšením kapacit by však byla spojena i vyšší finanční náročnost provozované infrastruktury.

Navrhované řešení není nízkonákladové na provoz. Pokud by nebyl brán ohled na specifikaci zadání, která požadovala využití původního kódu a s tím spojené nepoužití modelu Rebuild, bylo by na místě zvážit, zdali se nepokusit vytvořit aplikaci od znovu pomocí mikroservis, které nabízejí možnost fungování aplikace v cloudu za zlomek ceny a tím by bylo možné provoz dále optimalizovat. Pro tento způsob fungování aplikace by bylo zapotřebí kompletní přepracování aplikace, což by znamenalo nechat vývojový tým aplikaci přepracovat od začátku, což může mít finanční náklady z této strany a byla by v tomto ohledu zapotřebí podrobná analýza proveditelnosti. Z finančního hlediska by mohlo dávat určitý smysl uvažovat o převodu PostgreSQL na databáze na NoSQL databázi, u které lze očekávat nižší náklady na provoz, avšak by bylo zapotřebí dalších analýz pro potvrzení reálnosti těchto úvah pro řešení S-Podpis.

## 6 Závěr

V rámci této práce byla provedena analýza možnosti převodu aplikace z on-premise prostředí do prostředí cloudu.

V teoretické části byla vysvětlena problematika cloudu, zmapováno dělení cloud computingu podle jednotlivých modelů (Deployment a Service model), výhody a nevýhody cloudové problematiky a analyzovány největší poskytovatelé služeb v oblasti cloudu (AWS, Azure, GCP) pro uvedení čtenáře do zkoumané problematiky.

V rámci praktické části byl zanalyzován stav monolitické aplikace před provedením migrace, jmenovány a zhodnoceny požadavky pro výběr migračního modelu, i pro výběr platformy. Na základě získaných skutečností byl pro způsob migrace zvolen model Reachitecture, a to díky možnosti využití stávajícího kódu aplikace a prostředí Microsoft Azure, mimo jiné z důvodu možnosti využití clusteru zdarma pro testovací prostředí a kvůli známosti prostředí Microsoft Azure v rámci společnosti zadavatele.

V souladu s předchozí analýzou byla navržena implementace aplikace vycházející z modelu Reachitecture do prostředí Microsoft Azure, a navržen způsob migrace dat z on-premise databáze do cloudového řešení. Vhodnost řešení byla ověřena výpočtem finanční náročností nového prostředí a aplikováním uvedeného postupu v rámci testovacího provozu aplikace.

## 7 Seznam použitých zdrojů

1. **Richter, Felix.** Amazon Maintains Cloud Lead as Microsoft Edges Closer . *statista.com*. [Online] Statista Inc., 2024. 02 2024. <https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/>.
2. **Peter M. Mell, Timothy Grance.** The NIST Definition of Cloud Computing. *NIST*. [Online] 28. 09 2011. <https://www.nist.gov/publications/nist-definition-cloud-computing>. Special Publication (NIST SP) - 800-145.
3. **Garfinkel, Simson.** The Cloud Imperative. *MIT Technology Review*. [Online] MIT Technology Review, 3. 10 2011. <https://www.technologyreview.com/2011/10/03/190237/the-cloud-imperative/>.
4. **Salesforce.** The History of Salesforce. *Salesforce*. [Online] Salesforce inc, 1 2024. <https://www.salesforce.com/news/stories/the-history-of-salesforce/>.
5. **McCarthy, Ben.** Salesforce History: Your Guide from 1999 – 2022. *SFBEN*. [Online] SalesforceBen.com, 03. 08 2022. <https://www.salesforceben.com/salesforce-history/>.
6. **Foote, Keith D.** A Brief History of Cloud Computing. *Dataveristy*. [Online] Dataversity Digital LLC, 17. 12 2021. <https://www.dataversity.net/brief-history-cloud-computing/>.
7. **ERL, Thomas, Ricardo PUTTINI a Zaigham Mahmood.** *Cloud Computing: Concepts, Technology & Architecture: The Pearson Service Technology Series from Thomas Er. 2nd ed.* místo neznámé : Pearson Education, 2023. 978-0138052256.
8. **Baun, C., Kunze, M., Nimis, J., Tai, S.** *Introduction. In: Cloud Computing.* Berlin : Springer, 2001. 978-3-642-20916-1.
9. **Bogusch, Kevin.** Cloud Data Egress Costs: What They Are & How to Reduce Them. *Oracle*. [Online] Oracle, 24. 01 2024. <https://www.oracle.com/il-en/cloud/data-egress-costs/>.
10. **Neenan, Sarah.** 8 ways to reduce cloud costs. *TechTarget*. [Online] TechTarget, 05. 05 2022. <https://www.techtarget.com/searchcloudcomputing/feature/5-ways-to-reduce-cloud-costs>.
11. **Ramachandran, Muthu a Mahmood, Zaigham.** *Software Engineering in the Era of Cloud Computing* . místo neznámé : Springer Cham, 2020. 978-3-030-33624-0.
12. **Kiriathy, Yochay.** Inside the Azure App Service Architecture. *Microsoft Learn*. [Online] Microsoft, 02 2017. <https://learn.microsoft.com/en-us/archive/msdn-magazine/2017/february/azure-inside-the-azure-app-service-architecture>.

13. **Jamshidi, Pooyan and Ahmad, Aakash and Pahl, Claus.** *Cloud Migration Research: A Systematic Review*. místo neznámé : IEEE, 2013. 10.1109/TCC.2013.10.
14. **Mahdi Fahmideh Gholami, Farhad Daneshgar, Ghassan Beydounb, Fethi Rabhi.** *Challenges in migrating legacy software systems to the cloud — an empirical study*. Sydney : Elsevier Ltd., 2017. 0306-4379.
15. **GRYGAŘÍKOVÁ, MICHAELA.** Docker, Kubernetes a kontejnery. Jak fungují a proč je chtít. *masterdc*. [Online] MasterDC, 14. 08 2019.  
<https://www.master.cz/blog/docker-kubernetes-kontejnery-jak-funguji-proc-je-chtit/>.
16. **Microsoft.** Shared responsibility in the cloud. *Microsoft Learn*. [Online] Microsoft, 29. 09 2023. <https://learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility>.
17. **AWS.** Shared Responsibility Model. *AWS*. [Online] Amazon Web Services.  
<https://aws.amazon.com/compliance/shared-responsibility-model/>.
18. **Google .** Shared responsibilities and shared fate on Google Cloud. *Google Cloud*. [Online] Google inc., 21. 08 2023.  
<https://cloud.google.com/architecture/framework/security/shared-responsibility-shared-fate>.
19. **Evropská komise.** Ochrana osobních údajů podle nařízení GDPR. *europa.eu*. [Online] Evropská komise - DG Vnitřní trh, průmysl, podnikání a malé a střední podniky.
20. **Leong, Lydia.** Why Cloud Budgets Don't Stay in Check — And How to Make Sure Yours Do. *gartner*. [Online] Gartner, Inc, 28. 01 2022.  
<https://www.gartner.com/en/articles/why-cloud-budgets-don-t-stay-in-check-and-how-to-make-sure-yours-do>.
21. **DigitalOcean.** Comparing AWS, Azure, GCP. *DigitalOcean*. [Online] DigitalOcean, LLC. <https://www.digitalocean.com/resources/article/comparing-aws-azure-gcp>.
22. **Miller, Ron.** How AWS came to be. *Tech crunch*. [Online] Yahoo, 2. 7 2016.  
<https://techcrunch.com/2016/07/02/andy-jassys-brief-history-of-the-genesis-of-aws/>.
23. **Amazon.** Announcing Amazon Elastic Compute Cloud (Amazon EC2) - beta. *AWS Amazon*. [Online] Amazon, 24. 08 2006. <https://aws.amazon.com/about-aws/whats-new/2006/08/24/announcing-amazon-elastic-compute-cloud-amazon-ec2---beta/>.
24. **AWS.** Getting Started with the AWS Management Console. *AWS*. [Online] Amazon, 12. 05 2022. <https://aws.amazon.com/getting-started/hands-on/getting-started-with-aws-management-console/>.

25. **K, Sreyes, a další.** *Internet of Things and Cloud Computing Involvement Microsoft Azure Platform*. Tamilnadu : IEEE, 2022. 978-1-6654-8232-5.
26. **Abandy, Roosevelt.** The History of Microsoft Azure. *Microsoft techcommunity*. [Online] 24. 08 2022. <https://techcommunity.microsoft.com/t5/educator-developer-blog/the-history-of-microsoft-azure/ba-p/3574204>.
27. **Redkar, Tejaswi.** *Windows Azure Platform*. New York : Apress Berkeley, 2009. 978-1-4302-2480-8.
28. **Zach Hill, Jie Li, Ming Mao, Arkaitz Ruiz-Alvarez, Marty Humphrey.** *Early Observations on the Performance of Windows Azure*. Charlottesville : University of Virginia, 2010. 978-1-60558-942-8.
29. **Microsoft.** What is the Azure portal? *Learn Microsoft*. [Online] Microsoft, 06. 12 2023. <https://learn.microsoft.com/en-us/azure/azure-portal/azure-portal-overview>.
30. **Bisong, Ekaba.** *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. [Dokument] OTTAWA : Apress Media LLC, 2019. 978-1-4842-4469-2.
31. **Robinson, Dan.** Big three cloud giants tighten grip as overall spending slows. *TheRegister*. [Online] Situation Publishing, 7. 02 2023. [https://www.theregister.com/2023/02/07/big\\_three\\_cloud\\_market/](https://www.theregister.com/2023/02/07/big_three_cloud_market/).
32. **Talab, Zaher.** Introduction to IBM Cloud Computing. *developer*. [Online] TechnologyAdvice, 05. 08 2021. <https://www.developer.com/cloud/intro-ibm-cloud-computing/>.
33. **Harvey, Cynthia.** What is IBM Cloud? Services Offered, Features & Pricing. *Datamation*. [Online] TechnologyAdvice, 2017. 05 2017. <https://www.datamation.com/cloud/ibm-cloud/>.
34. **Cloud With Ease.** Oracle Cloud Infrastructure (OCI): An Overview. *Cloud With Ease*. [Online] AAR Technosolutions. <https://cloudwithease.com/oracle-cloud-infrastructure-oci/>.
35. **Barney, Nick.** Oracle Cloud. *TechTarget*. [Online] TechTarget. <https://www.techtarget.com/whatis/definition/Oracle-Cloud>.
36. **Microsoft.** Windows Server 2012 R2 . *learn.microsoft*. [Online] Microsoft. <https://learn.microsoft.com/cs-cz/lifecycle/products/windows-server-2012-r2>.
37. **Abhigna L Shastry, Devika S, Nairm Ramya C P, Prathima B, Prof. Phalachandra Hallymysore.** *Approaches for migrating non cloud-native*. [Dokument] Bengaluru, Indie : IEEE, 2022. 978-1-6654-8303-2.

38. **Amazon.** <https://aws.amazon.com/visualstudiocode/>. *AWS*. [Online] <https://aws.amazon.com/visualstudiocode/>.
39. —. What is the AWS Serverless Application Model (AWS SAM)? *AWS*. [Online] Amazon. <https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/what-is-sam.html>.
40. **What is the Azure Developer CLI?** *Microsoft Learn*. [Online] Microsoft. <https://learn.microsoft.com/en-us/azure/developer/azure-developer-cli/overview>.
41. **Azure Tools for Visual Studio Code.** *Visualstudio marketplace*. [Online] <https://marketplace.visualstudio.com/items?itemName=ms-vscode.vscode-node-azure-pack>.
42. **Google Cloud Command Line Interface (gcloud CLI).** *Google CLOUD*. [Online] Google. <https://cloud.google.com/cli?hl=cs>.
43. **Cloud Code overview.** *Google Cloud*. [Online] Google. <https://cloud.google.com/code/docs/vscode/overview>.
44. **Microsoft.** What is Microsoft Sentinel? *learn.microsoft*. [Online] Microsoft, 14. 03 2023. <https://learn.microsoft.com/en-us/azure/sentinel/overview>.
45. —. **Azure geographies.** *Azure*. [Online] Microsoft. <https://azure.microsoft.com/en-us/explore/global-infrastructure/geographies/#overview>.
46. **SkillUp Online.** Google Cloud Architect Program. *SkillUp Online*. [Online] SkillUp Online. <https://skillup.online/google-cloud-architect-certification-program/>.
47. **Google.** Google Kubernetes Engine pricing . *Google CLOUD*. [Online] Google. [https://cloud.google.com/kubernetes-engine/pricing#cluster\\_management\\_fee\\_and\\_free\\_tier](https://cloud.google.com/kubernetes-engine/pricing#cluster_management_fee_and_free_tier).
48. **Microsoft.** Free, Standard and Premium pricing tiers for Azure Kubernetes Service (AKS) cluster management. *Learn Microsoft*. [Online] Microsoft, 12. 12 2023. <https://learn.microsoft.com/en-us/azure/aks/free-standard-pricing-tiers>.
49. **Amazon.** Amazon EKS pricing. *AWS Amazon*. [Online] Amazon Web Services, Inc., 2024. <https://aws.amazon.com/eks/pricing/>.
50. **Dubey, Prateek.** Terraform Cloud Infrastructure — AWS, AZURE & GCP. *Medium*. [Online] A Medium Corporation, 9. 6 2020. <https://medium.com/@prateek.dubey/terraform-cloud-infrastructure-aws-azure-and-gcp-128280d0d4b>.



51. Microsoft. TLS termination with Key Vault certificates. *Microsoft Learn*. [Online] Microsoft, 02. 02 2024. <https://learn.microsoft.com/en-us/azure/application-gateway/key-vault-certs>.
52. —. What is Application Gateway Ingress Controller? *Microsoft Learn*. [Online] Microsoft. <https://learn.microsoft.com/en-us/azure/application-gateway/ingress-controller-overview>.
53. —. Pricing calculator. *Azure*. [Online] Microsoft . <https://azure.microsoft.com/en-us/pricing/calculator/>.
54. —. Dv3 and Dsv3-series. *Microsoft Learn*. [Online] Microsoft. <https://learn.microsoft.com/en-us/azure/virtual-machines/dv3-dsv3-series>.
55. K, Satya. Standard\_D2S\_V3. *Medium*. [Online] A Medium Corporation, 16. 11 2023. <https://medium.com/@oicnotif/standard-d2s-v3-2d1ce133f6a9>.
56. Microsoft. Azure Application Gateway documentation. *Microsoft Learn*. [Online] Microsoft. <https://learn.microsoft.com/en-us/azure/application-gateway/>.
57. —. Tutorial: Migrate PostgreSQL to Azure Database for PostgreSQL online using DMS (classic) via the Azure portal. *Microsoft Learn*. [Online] Microsoft, 11. 04 2023. <https://learn.microsoft.com/en-us/azure/dms/tutorial-postgresql-azure-postgresql-online-portal>.
58. N-iX. Scaling your infrastructure in the cloud: How to handle huge traffic spikes. *N-iX*. [Online] N-iX LTD , 29. 05 2022. <https://www.n-ix.com/scaling-cloud-infrastructure/>.
59. Google. Google Cloud overview. *Google cloud*. [Online] Google Inc. <https://cloud.google.com/docs/overview>.
60. Azure Portal Overview. *JavatPoint*. [Online] JavaTpoint. <https://www.javatpoint.com/azure-portal-overview>.

## 8 Seznam obrázků, tabulek, grafů a zkratk

### 8.1 Seznam obrázků

Obrázek 1- Příklad výkyvu při fungování aplikace, Zdroj: (7) .....	20
Obrázek 2 - Principy škálování, zdroj: (58).....	21
Obrázek 3 - Rozdíl mezi kontejnerizací a virtualizací, zdroj: (15).....	23
Obrázek 4 - Trust boundary, zdroj. (7) .....	24
Obrázek 5- Azure - Sdílená odpovědnost, zdroj: (16) .....	25
Obrázek 6 - AWS Sdílená odpovědnost, Zdroj: (17).....	25
Obrázek 7 - GCP Sdílená odpovědnost, Zdroj: (18).....	26
Obrázek 8- AWS ukázka Management console, zdroj: AWS (24) .....	30
Obrázek 9 - Ukázka Azure Portal, zdroj: (60) .....	32
Obrázek 10 - Ukázka Google cloud console, zdroj: (59) .....	34
Obrázek 11- S-Podpis monolitická architektura, vlastní zpracování.....	37
Obrázek 12 - S-Podpis – Diagram komponent, vlastní zpracování .....	49
Obrázek 13 – Azure Architektura S-Podpis, vlastní zpracování .....	50
Obrázek 14 - Ukázka Prostředí Azure Pricing calculator, zdroj: (53).....	52
Obrázek 15 - Převod databáze – Azure Resource provider, vlastní zpracování.....	56
Obrázek 16- Převod databáze – Vytváření migration service v Azure, vlastní zpracování	56
Obrázek 17 - Převod databáze – Vytváření migration service v Azure– Volba sítě, vlastní zpracování.....	57
Obrázek 18 - Převod databáze - Vytváření Migration service v Azure, vlastní zpracování	57

### 8.2 Seznam tabulek

Tabulka 1- Vícekriteriální analýza – Výběr migračního modelu, Vlastní zpracování .....	43
Tabulka 2 - Vícekriteriální analýza – Výběr cloudové služby, Vlastní zpracování .....	48
Tabulka 3 - Výsledek cenové kalkulace .....	54

### **8.3 Seznam použitých zkratek**

AWS – Amazon Web services

GCP – Google Cloud Platform

NIST – National Institute of Standards and Technology

MIT – Massachusettský technologický institut

CTSS – Conversational Time-Sharing Systém

SaaS – Software as a Service

IaaS – Infrastructure as a Service

PaaS – Platform as a Service

DaaS – Data as a Service

CRM – Customer Relationship Management

Ms – Microsoft

RAM – Random Access Memory

SLA – Service Level Agreement

Dev – Developer

Test – Testovací

DDOS – Distributed Denial of Service

API – Application Programming Interface

XFA – XML Forms Architecture

DMS – Document Management Systém

CI/CD – Continuous Integration/Continuous Delivery

GUI – Graphical User Interface

CLI – Command Line Interface

AKS, EKS, GKE – Kubernetes prostředí Azure, AWS a GCP

AGIC – Application Gateway Ingress Controller

SSL – Secure Sockets Layer

NSG – Network Security Group

VPN – Virtual Private Network