

# Využití bezplatných cloudových služeb pro výuku

Tomáš Sukop

---

Bakalářská práce  
2022



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav automatizace a řídicí techniky

Akademický rok: 2021/2022

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Tomáš Sukop  
Osobní číslo: A19614  
Studijní program: B3902 Inženýrská informatika  
Studijní obor: Informační a řídicí technologie  
Forma studia: Kombinovaná  
Téma práce: Využití bezplatných cloudových služeb pro výuku  
Téma práce anglicky: Using Free Cloud Service Plans in Programming Courses

### Zásady pro vypracování

1. Prostudujte nabídku bezplatných cloudových služeb.
2. Popište vlastnosti a způsob použití bezplatných služeb, které by bylo možné využít ve výuce na FAI UTB ve Zlíně.
3. Vyberte vhodná zadání do cvičení v předmětech jako „Programování mikropočítačů“, „Programování síťových aplikací“, „Paralelní procesy a programování“ apod.
4. Pro vybraná zadání vytvořte vzorové implementace a nasazení na cloudové službě.
5. Popište vývojové nástroje, které studenti mohou využít pro usnadnění vývoje a ladění kódu běžícího v cloudu.

Forma zpracování bakalářské práce: **tištěná/elektronická**

**Seznam doporučené literatury:**

1. BLOKDYK, Gerardus. IBM Cloud Computing A Complete Guide – 2020 Edition. B.m.: SSTARCOOKS, 2021. ISBN 978-0-655-92356-5.
2. SAVILL, John. Microsoft Azure infrastructure services for architects: designing cloud solutions. Indianapolis, IN: John Wiley & Sons, 2020. ISBN 978-1-119-59657-8.
3. PNG, Adrian a Luc DEMANCHE. Getting started with Oracle Cloud Free Tier: create modern web applications using always free resources. New York: Apress, 2020. For professionals by professionals. ISBN 978-1-4842-6010-4.
4. ALONGI, Raoul. AWS: The Most Complete Guide to Amazon Web Services from Beginner to Advanced Level. B.m.: Independently published, 2019. ISBN 978-1-67404-042-4.
5. GEEWAX, J. J. a Urs HÖLZLE. Google Cloud platform in action. Shelter Island, New York: Manning Publications Co, 2018. ISBN 978-1-61729-352-8.

Vedoucí bakalářské práce: **Ing. Tomáš Dulík, Ph.D.**  
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **15. ledna 2022**  
Termín odevzdání bakalářské práce: **20. května 2022**

**doc. Mgr. Milan Adámek, Ph.D. v.r.**  
děkan



**prof. Ing. Vladimír Vašek, CSc. v.r.**  
ředitel ústavu

Ve Zlíně dne 15. ledna 2022

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 18.5.2022

Tomáš Sukop, v. r.

## **ABSTRAKT**

Práce se zabývá možným využitím zdarma poskytovaných cloudových služeb ve výuce. Cílem bylo provést rešerši relevantních cloudových služeb, na základě zjištění vybrat služby vhodné pro dané účely, připravit zadání výukových úloh a vytvořit vzorové implementace s využitím vybraných cloudových služeb. Byla vytvořena dvě zadání a jejich vzorové implementace byly provedeny u třech největších poskytovatelů cloudových služeb.

Klíčová slova: cloud computing, cloudové služby, mikropočítač, programování mikropočítačů, rychlé zasílání zpráv, poskytovatelé služeb

## **ABSTRACT**

The thesis deals with the possible use of free cloud services in teaching. The aim was to search for relevant cloud services, based on the findings to select services suitable for the given purposes, to prepare the assignment of training tasks and to create sample implementations using selected cloud services. Two assignments were created and their sample implementations were performed at the three largest cloud service providers.

Keywords: cloud computing, cloud services, microcomputer, programming microcomputers, instant messaging, cloud providers

Rád bych poděkoval vedoucímu práce panu Ing. Tomáši Dulíkovi, Ph.D. a paní Ing. Daně Třeštíkové za cenné připomínky a podporu při zpracování tématu bakalářské diplomové práce.

Prohlašuji, že odevzdaná verze bakalářské diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>10</b>
<b>I. TEORETICKÁ ČÁST</b> .....	<b>12</b>
<b>1 DEFINICE CLOUD COMPUTINGU</b> .....	<b>13</b>
1.1 TYPY CLOUDŮ .....	13
1.1.1 Veřejné cloudy .....	13
1.1.2 Privátní cloudy .....	13
1.1.3 Hybridní cloudy .....	14
1.1.4 Multicloud .....	15
1.2 TYPY POSKYTOVANÝCH CLOUDOVÝCH MODELŮ .....	15
1.2.1 Infrastruktura jako služba – Infrastructure-as-a-Service (IaaS) .....	16
1.2.2 Software jako služba – Software-as-a-Service (SaaS) .....	16
1.2.3 Platforma jako služba – Platform-as-a-Service (PaaS) .....	16
1.2.4 Funkce jako služba – Function-as-a-Service (FaaS) .....	16
<b>2 PŘÍSTUP K VÝBĚRU POSKYTOVATELE CLOUDOVÝCH SLUŽEB</b> .....	<b>18</b>
<b>3 AMAZON WEB SERVICES (AWS)</b> .....	<b>19</b>
3.1 CLOUD COMPUTING .....	19
3.1.1 Amazon EC2 .....	19
3.2 DATABÁZE .....	20
3.2.1 DynamoDB .....	20
3.2.2 RDS .....	21
3.3 FUNKCE.....	21
<b>4 GOOGLE CLOUD PLATFORM (GCP)</b> .....	<b>22</b>
4.1 CLOUD COMPUTING .....	22
4.2 DATABÁZE .....	23
4.2.1 Cloud SQL .....	23
4.2.2 Cloud Bigtable .....	24
4.3 FUNKCE.....	25
4.3.1 Cloud Functions .....	25
4.3.2 Cloud Run .....	25
<b>5 MICROSOFT AZURE</b> .....	<b>26</b>
5.1 CLOUD COMPUTING.....	26
5.2 DATABÁZE .....	27
5.2.1 Azure SQL .....	27
5.2.2 Azure Cosmos DB.....	28
5.3 FUNKCE.....	28
<b>6 REDHAT</b> .....	<b>29</b>
<b>7 ORACLE</b> .....	<b>30</b>
7.1 CLOUD COMPUTING.....	30
7.1.1 Flexibilní stroje .....	30
7.1.2 Standardní stroje.....	31
7.2 DATABÁZE .....	31



7.2.1	Oracle Exadata .....	31
7.2.2	Autonomní databáze.....	31
7.3	FUNKCE.....	32
<b>8</b>	<b>VÝVOJOVÉ NÁSTROJE PRO PRÁCI S CLOUDEM.....</b>	<b>33</b>
8.1	OBEČNÁ VÝVOJOVÁ PROSTŘEDÍ .....	33
8.1.1	Intellij IDEA.....	33
8.1.2	Visual Studio Code .....	34
8.2	SOFTWARE DEVELOPMENT KIT (SDK).....	35
8.2.1	AWS Java SDK.....	36
8.2.2	MS Azure Java SDK.....	36
8.2.3	Google API Client knihovna pro Javu .....	37
8.3	KONZOLE .....	38
8.3.1	AWS Management Console .....	38
8.3.2	Microsoft Azure Portal.....	38
8.3.3	Cloud Console .....	39
<b>9</b>	<b>VLASTNOSTI A ZPŮSOB POUŽITÍ BEZPLATNÝCH SLUŽEB.....</b>	<b>41</b>
9.1	AWS .....	41
9.1.1	IAM služba.....	42
9.1.2	Organizace.....	44
9.2	GCP .....	44
9.2.1	IAM služba.....	45
9.3	MS AZURE .....	46
9.3.1	Azure Active Directory .....	47
<b>II.</b>	<b>PRAKTICKÁ ČÁST.....</b>	<b>48</b>
<b>10</b>	<b>VYTVOŘENÍ ZADÁNÍ DO CVIČENÍ .....</b>	<b>49</b>
10.1	IM SERVER V JAVĚ.....	49
10.2	MPC v IoT .....	49
<b>11</b>	<b>VZOROVÉ IMPLEMENTACE .....</b>	<b>50</b>
11.1	AWS .....	50
11.1.1	IM Server .....	50
11.1.2	MPC v IoT.....	55
11.2	GCP .....	58
11.2.1	IM Server .....	58
11.2.2	MPC v IoT.....	63
11.3	AZURE.....	68
11.3.1	IM Server .....	68
11.3.2	MPC v IoT.....	72
	<b>ZÁVĚR .....</b>	<b>76</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>78</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>84</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>86</b>

## ÚVOD

Cílem této bakalářské práce je zjistit, jaké bezplatné cloudové služby dostupné na trhu by bylo možné využít ve výuce na Fakultě aplikované informatiky Univerzity Tomáše Bati (dále jen „FAI UTB“) ve Zlíně, dále vytvořit zadání do cvičení ve vybraných předmětech a vypracovat vzorové implementace a nasazení na cloudové službě.

Za tímto účelem bude provedena rešerše vhodných bezplatných cloudových služeb na trhu, popsány jejich vlastnosti a možné způsoby použití relevantní pro dané účely. V další části práce budou popsány vývojové nástroje, které studenti mohou využít pro usnadnění vývoje a ladění kódu běžícího v cloudu při řešení zadaných úloh. Na základě uvedených zjištění bude vytvořeno zadání do cvičení v předmětech jako například „Programování mikropočítačů“, „Programování síťových aplikací“ nebo „Paralelní procesy a programování“. Pro vybraná zadání budou následně vypracovány vzorové implementace a nasazení na cloudové službě.

Teoretická část práce v úvodních kapitolách přináší základní informace a úvod do dané problematiky. Zabývá se definicí cloudu jako takového a vybranými atributy cloudů, a to v rozsahu přiměřeném naplnění daného cíle. Další kapitola zahrnuje přehled vybraných poskytovatelů cloudových služeb. Výběr byl uskutečněn především s přihlédnutím k širší nabídce využitelných bezplatných cloudových služeb. Z uvedených důvodů byli zvoleni a popsáni: tři největší poskytovatelé cloudových služeb na trhu, a to Amazon Web Services (dále také „AWS“), Google Cloud Platform (dále také „GCP“) a Microsoft Azure (dále také „MS Azure“), jeden později vzniklý poskytovatel (Oracle) a také poskytovatel, který má velký vliv na open source řešení například operačních systémů (RedHat), ač nenabízí tak široký rozsah možností využití pro uvedené účely. Principy popsané v této části práce jsou použitelné obecně u většiny vývojových prostředí. Jako další byly popsány vývojářské nástroje software development kit, konzole integrované v prostředích cloudových poskytovatelů a obecná vývojová prostředí v praxi nejpoužívanější (dle vlastní pracovní zkušenosti, pozn. aut.), a to JetBrains a Visual Studio Code. Ve čtvrté kapitole jsou rozepsány služby tří největších poskytovatelů cloudu ve smyslu využití služeb poskytovaných bezplatně, podmínek jejich použití a vytváření dalších uživatelů způsobem umožňujícím cloudové služby využívat bez nutnosti registrace.

Praktická část práce představuje dvě zadání vytvořená pro cvičení do předmětů FAI UTB ve Zlíně. Jedná se o dvě zadání, z nichž jedním je implementace serveru napsaného v jazyce

Java a nahrání serveru na cloud; druhým zadáním je naprogramování programovatelné základní desky s následným připojením do internetu věcí (dále jen „IoT“). Vzorové implementace jsou následně demonstrovány opět na třech největších cloudech AWS, GCP a MS Azure.

## **I. TEORETICKÁ ČÁST**

## 1 DEFINICE CLOUD COMPUTINGU

Pod pojmem cloud computing (dále také „cloud“) se rozumí dodání IT zdrojů na požádání (tzn. v okamžiku, kdy je potřeba) prostřednictvím internetového připojení, přičemž uživatel hradí služby formou průběžných plateb podle využívání daných služeb. Místo vlastnění (tj. nákupu, provozování a udržování) fyzických data center a serverů se přistupuje k využívání technologických služeb, jako je výpočetní síla, úložiště a databáze v rozsahu podle potřeby, od cloudového poskytovatele. [1]

### 1.1 Typy cloudů

V minulosti jasně určovalo rozdíly mezi veřejným, soukromým a hybridním cloudem a multicloudem především to, kdo byl vlastníkem serverů, a jejich lokace. Postupným vývojem v této oblasti došlo k rozšíření počtu určujících atributů i jejich možných hodnot. Aktuálně lze definovat hlavní typy cloudů tak, jak je popsáno dále, nicméně je pravděpodobné, že bude v budoucnu potřeba tyto definice s ohledem na neustále se vyvíjející prostředí přehodnotit. [2]

#### 1.1.1 Veřejné cloudy

Veřejný cloud je fond virtuálních zdrojů vyvíjený a spravovaný třetí stranou. Tyto zdroje jsou automaticky poskytovány a alokovány mezi více uživatelů prostřednictvím samoobslužného rozhraní. Je to jednoduchý způsob jak škálovat pracovní zátěž, která vykazuje nepredikovatelné výkyvy v souladu s proměnnou poptávkou. [3]

Dnešní veřejné cloudy nejsou obvykle vystavovány jako samostatná řešení infrastruktury, ale spíše jako součást heterogenního mixu prostředí, jež vede k vyšší bezpečnosti a výkonu, nižším cenám a širší dostupnosti infrastruktury, služeb a aplikací. [3]

Poskytovat veřejný cloud může kdokoliv a na světě jsou tisíce takových poskytovatelů. Mezi největší a v dnešní době klientsky nejoblíbenější patří Alibaba Cloud, Amazon Web Services, Google Cloud, IBM Cloud a Microsoft Azure. [3]

#### 1.1.2 Privátní cloudy

Soukromé cloudy jsou cloudová prostředí výhradně dedikovaná konkrétnímu koncovému uživateli, obvykle za jeho firewallem. Uživatelem je pro tyto účely myšlen koncový zákazník, klient, organizace využívající cloud. Ačkoliv soukromé cloudy tradičně běžely na

místě (tzv. on-premise, v místě sídla či prostorách používaných uživatelem, pozn. aut.), organizace dnes vytvářejí své soukromé cloudy v pronajatých centrech ve vlastnictví poskytovatele umístěných mimo lokaci (tzv. off-premise, tedy mimo sídlo či prostory používané uživatelem, pozn. aut.). [5]

Všechny cloudy se tedy stávají soukromými v momentě, kdy je veškerá IT infrastruktura daného cloudu dedikována jednomu uživateli se zcela izolovaným přístupem. [5]

### 1.1.3 Hybridní cloudy

Hybridní cloud je IT architektura, která zahrnuje určitou úroveň přenositelnosti pracovní zátěže, orchestrace a správy přes dvě nebo více prostředí. Tato prostředí mohou obsahovat: [6]

- alespoň jeden soukromý a alespoň jeden veřejný cloud;
- dva nebo více soukromých cloudů;
- dva nebo více veřejných cloudů;
- fyzický stroj nebo virtuální prostředí propojené s alespoň jedním veřejným nebo soukromým cloudem. [6]

Tyto na první pohled různorodé požadavky vznikly evolucí z dřívějších dob cloud computingu, kdy rozdíly mezi soukromým a veřejným cloudem byly jednoduše rozpoznatelné (viz 1.1). Dnešní typy cloudů jsou mnohem komplexnější, protože lokace a vlastnictví jsou abstraktní povahy. Například veřejné cloudy kdysi běžely mimo fyzickou infrastrukturu, avšak poskytovatelé dnes provozují cloudové služby také uvnitř fyzické infrastruktury datacenter uživatelů. Soukromé cloudy oproti tomu obvykle fungovaly uvnitř fyzické infrastruktury, dnes organizace staví privátní cloudy v pronajatých datacentrech umístěných mimo jejich fyzickou infrastrukturu. [6]

Z uvedených důvodů je vhodné definovat hybridní cloud spíše na základě toho, jaké služby poskytuje, než podle toho, kde se nachází. Všechny hybridní cloudy by v zásadě měly:

- spojovat více počítačů přes síť;
- slučovat IT zdroje;
- zvětšovat a rychle poskytovat nové zdroje;
- být schopné rozdělovat zátěž mezi různá prostředí;

- obsahovat jeden unifikovaný nástroj pro správu;
- organizovat procesy za pomoci automatizace. [6]

#### 1.1.4 Multicloud

Multicloud je označení cloudového řešení, které se skládá z více služeb od více než jednoho poskytovatele – veřejných nebo soukromých. Příklad multicloudového řešení by mohl být následující: [7]

Organizace investuje do rozšíření cloudové infrastruktury, proběhl přesun stěžejní aplikace z fyzických serverů na virtualizované servery.

Proběhlo vyhodnocení na trhu dostupných možností veřejných cloudů – ne pro všechny služby, které organizace poskytuje, ale pro podporu specifických aplikací pro zákazníky s vysoce variabilní mírou využití.

Po průzkumu trhu organizace našla vhodného poskytovatele a přesunula část své infrastruktury k němu na veřejný cloud.

Zákazníci začali požadovat funkce, které jsou dostupné pouze z proprietární aplikace jiného poskytovatele (poskytovatele č. 2). Pro integraci takových funkcionalit do aplikace je nejen nutné, aby organizace zakoupila zmíněnou aplikaci, ale také ji musí hostovat v cloudu poskytovatele č. 2 v prostředí, které umožňuje zvyšování výkonu obou aplikací (její vlastní i aplikace prodejce č. 2) podle poptávky. [7]

Takto vznikne multicloud.

## 1.2 Typy poskytovaných cloudových modelů

V průběhu času a vývoje cloud computingu vzniklo několik různých modelů a způsobů nasazování aplikací, aby se vyhovělo specifickým potřebám různých uživatelů. Každý typ cloudové služby a metody nasazení poskytuje různé úrovně kontroly, flexibility a správy. Vysvětlení rozdílů mezi různými modely, tedy infrastrukturou jako službou, platformou jako službou a softwarem jako službou, obdobně jako pochopení rozdílů mezi strategiemi nasazení, může být nápomocno v rozhodování, jaké služby použít v konkrétních případech. [8]

### **1.2.1 Infrastruktura jako služba – Infrastructure-as-a-Service (IaaS)**

Infrastruktura jako služba (Infrastructure-as-a-Service, zkráceně „IaaS“) obsahuje základní blok pro cloud IT a obvykle poskytuje přístup k síťovým funkcím, počítačům (ať už virtuálním nebo dedikovaným hardwarům) a datovým úložištím. [8]

IaaS poskytuje nejvyšší úroveň flexibility a kontroly řízení IT zdrojů a je nejpodobnější již existujícím IT zdrojům, se kterými jsou dnešní IT oddělení a vývojáři dlouho seznámeni. [8]

### **1.2.2 Software jako služba – Software-as-a-Service (SaaS)**

Software jako služba (Software-as-a-Service, zkráceně „SaaS“) poskytuje kompletní a hotový produkt, který je provozován a spravován poskytovatelem služby. Ve většině případů, kdy se hovoří o softwaru jako službě, se myslí aplikace poskytované koncovým uživatelům. Díky nabídce SaaS není potřeba přemýšlet nad tím, jak je služba udržovaná nebo jak vypadá její příslušná infrastruktura, či jak je spravovaná. Jediné, nad čím je potřeba přemýšlet, je způsob použití daného softwaru. [8]

Běžný příklad SaaS aplikace je e-mailová služba ve webovém rozhraní, kde lze odesílat a přijímat e-maily, aniž by bylo potřeba spravovat různé doplňky funkcionalit e-mailu nebo udržovat servery a operační systémy, na kterých běží serverová část e-mailové služby. [8]

### **1.2.3 Platforma jako služba – Platform-as-a-Service (PaaS)**

Platforma jako služba (Platform-as-a-Service, zkráceně „PaaS“) odstraňuje potřebu organizací spravovat základní infrastrukturu (obvykle hardware a operační systémy) a dovoluje jim soustředit se na vývoj a správu aplikací jako takových. Výsledkem je možnost pracovat účinněji, jelikož není potřeba se strachovat o zprostředkování zdrojů, plánování kapacity, údržbu softwaru, opravování nebo o další nediferencovanou složitou práci spojenou s provozem aplikace. [8]

### **1.2.4 Funkce jako služba – Function-as-a-Service (FaaS)**

Funkce jako služba (Function-as-a-Service, zkráceně „FaaS“) je druh cloudové výpočetní služby, která dovoluje vývojáři vytvářet, kalkulovat, spouštět a spravovat aplikační balíčky jako funkce bez nutnosti udržovat vlastní infrastrukturu. [9]

FaaS je exekuční model řízený událostmi běžící v kontejnerech, které se vyznačují tím, že dokud nejsou použity, tak neběží a jsou beze stavu. Říká se jim tedy bezstavové kontejnery.



Tyto funkce spravují logiku a stavy na straně serveru, a to využitím služeb poskytovatele. [9]

Řešení typu FaaS jsou dostupné na většině veřejných cloudů a mohou být poskytnuty on-premise, čímž signifikantně zvětšují rozsah schopnosti oddělení IT k vývoji aplikací. [9]

Vybrané příklady nejčastěji používaných FaaS:

- Amazon AWS Lambda,
- Google Cloud Functions,
- Microsoft Azure Functions.

## 2 PŘÍSTUP K VÝBĚRU POSKYTOVATELE CLOUDOVÝCH SLUŽEB

Poskytovatelé cloudových služeb jsou společnosti, které založily veřejné cloudy, spravují soukromé cloudy nebo nabízejí komponenty cloudových výpočtů na požádání (taktéž známy jako výpočetní služby cloudu) jako jsou IaaS, PaaS, SaaS a FaaS, které byly podrobněji popsány v první kapitole jako typy poskytovaných cloudových modelů. Cloudové služby mohou snížit náklady na firemní procesy v porovnání s infrastrukturou on-premise. [10]

Nejlepší způsob, jak vybrat správného poskytovatele je podle toho, jaké jsou potřeby a velikost podniku, aktuální výpočetní platforma, IT infrastruktura a také jaké jsou vytyčené cíle. První věcí, kterou lze udělat, je vyhodnotit, zda využití konkrétního poskytovatele odpovídá strategii podniku. [10]

Pokud ano, dalším krokem je verifikovat, které služby budou v cloudu potřeba k podpoře dané strategie – jaké technologie jsou ve firmě udržitelné a které by měly být delegovány na poskytovatele cloudu. [10]

Infrastruktura, platforma nebo software, který je spravován poskytovatelem, může uvolnit prostředky organizace k obsluze klientů, zvýšit účinnost v celkovém fungování a přidat více času ke zlepšování nebo expandování vývojových operací (DevOps). [10]

V následujících podkapitolách jsou popsány společnosti AWS, Google Cloud Platform, Microsoft Azure, RedHat a Oracle. Jde o obecně známé a velké poskytovatele cloudových služeb. Kromě těchto výše zmíněných jsou další stovky poskytovatelů různé velikosti a zaměření po celém světě. [10]

U každého výše zmíněného poskytovatele jsou dále zmíněné služby, které nabízí pro potřeby cloud computingu, databází a funkcí, tzn. všechny služby potřebné pro všechny zmíněné cloudové modely v první kapitole.

### 3 AMAZON WEB SERVICES (AWS)

V roce 2006 začala firma Amazon Web Services nabízet služby IT infrastruktury podnikům ve formě webových služeb – dnes již běžně známých jako cloud computing. Dnes AWS poskytuje vysoce spolehlivou, škálovatelnou, nízko nákladovou platformní infrastrukturu v cloudu, která pohání stovky tisíc společností ve 190 zemích světa. S data centry v U.S., Evropě, Brazílii, Singapuru, Japonsku a Austrálii zákazníci využívají všech výhod, které AWS nabízí. [11]

AWS nabízí přes 200 plně funkčních a aktualizovaných služeb ve svých datacentrech na celém světě a obsluhuje miliony zákazníků. [12]

#### 3.1 Cloud computing

##### 3.1.1 Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) nabízí nejširší a nejhlubší výpočetní platformu s více než pěti sty instancemi a výběrem nejmodernějších procesorů, úložišť, síťových zařízení, operačních systémů a nákupních modelů, které nejlépe splní veškeré podmínky pracovní zátěže. Je to první velký poskytovatel cloudu, který podporuje Intel, AMD a Arm procesory, jediný cloud, který nabízí instance EC2 Mac na požádání a jediný cloud se 400Gbps ethernetovou sítí. Nabízí výkon za nejlepší cenu pro strojové učení, stejně jako nejnižší cenu per interface instance v cloudu. Na AWS funguje více SAP (systémy, aplikace a produkty ve zpracování dat), HPC (výkonově náročné výpočty), ML (strojové učení) a Windows instancí než na jiném cloudu. [13]

Způsoby využití EC2 jsou různé. Lze zde spouštět cloud nativní a podnikové aplikace. Amazon EC2 poskytuje bezpečné, spolehlivé, velmi výkonné a nákladově efektivní výpočetní infrastruktury, které jsou schopny splnit požadavky podniků. [13]

EC2 je také vhodné pro výkonově náročné výpočetní aplikace (HPC). Umožňuje přístup k infrastruktuře na požádání a ke kapacitě, kterou vyžadují HPC aplikace, opět efektivně co se nákladů týče. [13]

Nabízí také možnost vyvíjet pro platformy Apple. Sestavovat, testovat a podepisovat na požádání macOS pracovní zátěže. Možnosti přístupu do prostředí v rámci minut a dynamické škálování kapacity, jak je potřeba. [13]

Využívá se také pro trénování a vystavování ML aplikací. Díky nejširší nabídce výpočetních prostředí, síťové rychlosti až 400 Gbps a službám úložišť, které byly vyvinuty a optimalizovány právě na ML projekty. [13]

## 3.2 Databáze

AWS nabízí široké spektrum databázových systémů, které jsou stavěné pro specifické problémy. Lze vybrat z více jak patnácti databázových systémů, mezi které patří například relační, key-value, dokumentové, paměťové a grafové databáze. [13]

Relační databáze na AWS jsou třikrát až pětikrát rychlejší než jiné alternativy. Nerelační databáze zase poskytují latenci menší než milisekundu. Všechny klastry databází jsou průběžně sledovány poskytovatelem, aby byla zaručena jejich bezchybná funkčnost. Navíc všechny databáze fungují na samoléčivém úložném prostoru s automatickým škálováním, aby se klient mohl soustředit na vývoj aplikace, a ne na údržbu databáze. Všechny druhy databází jsou vždy dostupné a zabezpečené. [14]

Pro tuto práci jsou níže zmíněny dva druhy databází, které AWS poskytuje, jakožto zástupci dvou hlavních databázových paradigmat, a to SQL a NoSQL databáze.

### 3.2.1 DynamoDB

Amazon DynamoDB je plně obsluhovaná, serverless, key-value NoSQL databáze vytvořená pro vysoce výkonné aplikace v libovolném měřítku. DynamoDB nabízí vestavěné zabezpečení, průběžné zálohování, automatickou multi-regionální replikaci, ukládání do mezipaměti a nástroje pro export dat. [15]

S takovou databází lze vytvářet aplikace napříč internetem, podporující metadata s uživatelským obsahem a ukládat tak do mezipaměti data, která vyžadují vysoký souběh a připojení pro miliony uživatelů a miliony dotazu za vteřinu. [15]

Taková databáze umí navyšovat propustnost a souběh pro média a zábavní pracovní zátěže, jako je streamování videí v reálném čase a interaktivní obsah, a poskytnout nízkou latenci s multi-regionální aplikací napříč AWS regiony. [15]

Další možností této databáze je možnost využívání návrhových vzorů pro vystavování nákupních košíků, pracovních postupů motorů, sledování inventářů a zákaznických profilů. DynamoDB podporuje aplikace s vysokou návštěvností, extrémně náročnými událostmi a zvládne miliony dotazů za vteřinu. [15]

V neposlední řadě se s těmito databázemi lze soustředit na inovace bez provozní režie, například postavit herní platformu s daty hráčů, historií událostí a žebříčky pro miliony uživatelů souběžně požívající danou aplikaci. [15]

### 3.2.2 RDS

Amazon Relational Database Service (RDS) je kolekce spravovaných služeb, které umožňují vytvořit, operovat a škálovat databázi v cloudu. Lze vybrat ze sedmi populární databázových nástrojů – Amazon Aurora s MySQL kompatibilitou, Amazon Aurora s PostgreSQL kompatibilitou, MySQL, MariaDB, PostgreSQL, Oracle a SQL Server – a nasazovat je na svých strojích s Amazon RDS na AWS Outpost. [16]

### 3.3 Funkce

Pro cloudový model „Function as a Service“ nabízí AWS službu AWS Lambda. Je to serverless událostmi řízená výpočetní služba, na které může běžet kód prakticky jakékoliv aplikace nebo backendové služby bez jakéhokoliv řízení nebo spravování serverů. Lambdu lze spustit z více jak 200 AWS služeb a SaaS aplikací. [17]

Lambdy lze použít například na zpracování souborů automatickým spouštěním kódu s potřebnou výpočetní kapacitou při nahrání souboru do Amazon Simple Storage Service (Amazon S3) nebo při jiné vlastní události. [17]

Pre-procesování dat předtím, než se vloží do ML modelu. Se službou Amazon Elastic File System (EFS) AWS Lambda dokáže spravovat a poskytovat infrastrukturu ke zjednodušení zvětšování objemu dat. [17]

Vytvoření aplikací řízených událostmi pro zjednodušenou komunikaci mezi rozdílnými službami. Lze takto zajistit vyšší stabilitu při vysokém nárůstu poptávky a zabránit pádu nebo přečerpání zdrojů. [17]

## 4 GOOGLE CLOUD PLATFORM (GCP)

GCP bylo poprvé spuštěno v roce 2008 spolu se svým produktem App Engine. V dubnu 2008 Google oznámil ukázkou vývojářského nástroje App Engine, který měl umožnit zákazníkům spouštět jejich webové aplikace na infrastruktuře Googlu. Podle Googlu bylo cílem App Engine ulehčit spouštění webové aplikace, a potom ji jednoduše škálovat v momentě, kdy provoz na ní znatelně stoupne kvůli přítomnosti milionů uživatelů. [18]

Ke konci roku 2011 Google spustil App Engine v plném režimu. V nadcházejícím desetiletí Google vytvořil a spustil více služeb a produktů, aby vylepšil uživatelskou zkušenost své cloudové platformy. [18]

Dnes je GCP jeden z nejlepších cloudových poskytovatelů na světě. Mezi zákazníky Google Cloud patří Nintendo, eBay, UPS, The Home Depot, Etsy, Paypal, 20th Century Fox a Twitter. [18]

### 4.1 Cloud computing

GCP nabízí více druhů virtuálních strojů pro výpočet. Patří mezi ně:

- škálovací pracovní zátěže (T2D),
- zátěže pro běžné použití (E2, N2, N2D, N1),
- vysokopaměťové stroje (M2, M1),
- zátěžové stroje pro složité výpočty (C2, C2D),
- nejnáročnější aplikace a zátěže (A2).

T2D nabízí výkon za nejlepší cenu v porovnání se stroji pro běžné použití od jiných cloudových poskytovatelů. Je to první instance typu Tau VM rodiny a je nabízena v předdefinovaných tvarech s až 60 vCPU jednotek na VM a 4 GB paměti na vCPU. T2D je ideální pro zátěže s potřebou škálování jako jsou webové servery, kontejnerizované mikroslužby, kódování videí a velké java aplikace. [19]

E2, N2, N2D a N1 jsou stroje pro běžné použití nabízející rovnováhu mezi cenou a výkonem a jsou vhodné pro široké spektrum běžných zátěží jako jsou databáze, vývojová a testovací prostředí, webové aplikace a mobilní gaming. Podporují až 224 vCPU a 896 GB paměti. [19]

M2 a M1, tedy stroje optimalizované pro paměť, nabízejí nejvyšší paměťové konfigurace s až 12 TB pro jednu instanci. Jsou vhodné pro pracovní zátěž zaměřenou na paměť jako například in-memory databáze jako SAP HANA a in-memory data analytické práce. [19]

Virtuální stroje optimalizované pro výpočetní výkon (C2, C2D) poskytují nejvyšší výkon per jádro na výpočetním prostředí. Jsou optimalizované pro zátěž typu HPC (high performance computing), herní servery a poskytování API, na kterou je požadavek nízké odezvy. [19]

Pro nejnáročnější aplikace a zátěže slouží stroje typu A2. Ty jsou založeny na NVIDIA Ampere A100 Tensor Core GPU. Každé jádro typu A100 nabízí až dvacetkrát vyšší výpočetní sílu v porovnání s předchozími generacemi GPU. Tyto virtuální stroje jsou navrženy pro ML (strojové učení) a HPC. [19]

## 4.2 Databáze

Google Cloud Platform poskytuje neomezenou platformu založenou na desetiletích vývoje jedinečných databázových systémů. Veškeré nabízené systémy nabízí masivní škálovatelnost a odolnost dat. Veškeré nejpoblárnější systémy Googlu běží na těchto platformách. Patří mezi ně YouTube, Search a Maps. Google Cloud splňuje sto procent potřebných kritérií nastavených firmou Gartner. [20]

Jelikož GCP taktěž nabízí spoustu služeb pro různé databázové systémy, byly vybrány dvě nejčastěji využívané: SQL a NoSQL databáze.

### 4.2.1 Cloud SQL

GCP nabízí více služeb obsluhujících relační databáze, pro potřeby této práce byla zvolena ta nejčastěji využívaná (podobně jako u AWS), a to Cloud SQL.

Cloud SQL je služba nabízející plně spravovaný MySQL, PostgreSQL a SQL Server. Lze na ní spouštět stejné relační databáze, které jsou běžně známy, a to i se všemi rozšířeními, možnostmi nastavení a volbou znaků, ovšem bez zbytečného sebeřízení. [21]

Tato služba automaticky zajišťuje, že spravované databáze jsou spolehlivé, zabezpečené a škálovatelné, takže organizace může bez problémů z databázového pohledu dále fungovat. Cloud SQL automatizuje všechny zálohy, replikace, šifrovací záplaty a navýšení kapacit, zatímco je zaručena dostupnost vyšší než 99,95 % kdekoli na světě. [21]

Přístup ke Cloud SQL instancím je umožněn prakticky z jakékoliv aplikace. Jednoduše je lze propojit s App Engine, Compute Engine, Google Kubernetes Engine a také z fyzické stanice. Pomocí Big Query lze zajistit analýzu dat přímými dotazy do Cloud SQL databází. [21]

Pro ochranu dat lze jednoduše nastavit replikaci a zálohování. K zaručení vysoké dostupnosti je možné nastavit automatické převzetí služeb při selhání. Data jsou automaticky šifrována a Cloud SQL vyhovuje normám SSAE 16, ISO 27001 a PCI DSS a také podporuje normu HIPAA. [21]

Databázová migrační služba (DMS) umožňuje jednoduše migrovat produkční databáze do Cloud SQL s minimálními prostoji. Toto serverless řešení eliminuje problém s ručním poskytováním, správou a monitorováním prostředků specifických pro migraci. DMS využívá nativní replikační schopnost MySQL a PostgreSQL k maximalizaci přesnosti a spolehlivosti migrace. [21]

#### 4.2.2 Cloud Bigtable

Jde o plně spravovanou, škálovatelnou NoSQL databázovou službu pro velké analytické a operativní zatížení s až 99,999% dostupností. [22]

Bigtable je ideální pro ukládání velmi velkého objemu dat v key-value úložišti a podporuje vysokou propustnost čtení a zápisu s nízkou latencí pro rychlý přístup k velkému množství dat. Propustnost je škálovaná lineárně – lze navýšit QPS (dotazy za vteřinu) přidáním Bigtable uzlů. Bigtable je postaven s odzkoušenou infrastrukturou, která pohání produkty Googlu používané miliardami lidí jako jsou Search a Maps. [22]

Škálovat lze od tisíců k milionům operací čtení nebo zápisu bezproblémově během několika vteřin. Propustnost Bigtable může být dynamicky upravovaná přidáním nebo odebráním uzlů klastru bez nutnosti restartování. To znamená, že lze zvětšit velikost klastru na pár hodin, aby aplikace ustála nápor dotazů, a poté zase jeho velikost zredukovat – to vše bez odstavky. [22]

Tato služba také nabízí flexibilní, automatizované replikování k optimalizaci pracovního vytížení. Stačí data zapsat pouze jednou a automatická replikace se provede, kdykoliv je potřeba. Není třeba žádných manuálních kroků k zaručení konzistence, opravy dat nebo synchronizace zápisů a mazání. [22]



## 4.3 Funkce

Pro FaaS nabízí GCP, kromě dalších, dvě služby, Cloud Functions a Cloud Run. V dalších dvou podkapitolách jsou popsány právě tyto služby.

### 4.3.1 Cloud Functions

Tato služba slouží ke spuštění kódu v cloudu bez serverů nebo kontejnerů, které by bylo nutné spravovat. Cloud Functions je škálovatelná služba FaaS s platbami podle využití, která pomáhá stavět a propojovat služby řízené událostmi s jednoduchým jednoúčelovým kódem. [23]

Cloud Functions má jednoduché a intuitivní vývojové prostředí. Stačí napsat kód a Google Cloud vyřeší operativní infrastrukturu. Tímto způsobem lze vyvíjet rychleji psáním a spuštěním menších skriptů, které odpovídají daným událostem. Problémy s orchestrací Google Cloud služeb nebo služeb třetích stran lze vyřešit jejich propojením, které vznikne používáním událostí. [23]

### 4.3.2 Cloud Run

Slouží k vývoji a vystavování vysoce škálovatelných, kontejnerizovaných aplikací použitím oblíbených programovacích jazyků nebo frameworků (Go, Python, Java, Node.js, .NET a další) na plně spravované serverless platformě. Všichni zákazníci Google Cloudu mají 2 miliony dotazů měsíčně zdarma. [24]

Podporovaný je jakýkoliv jazyk, knihovna nebo binární soubor. Lze použít jakýkoliv jazyk, knihovny operačních systémů nebo vlastní binární soubory. [24]

Lze využít kontejnerových pracovních postupů a standardů. Kontejnery se staly standardem balení a nasazení kódu a jeho závislostí. Cloud Run se hodí právě na spuštění s kontejnerovým ekosystémem: Cloud Build, Cloude Code, Artifact Registry a Docker. [24]

Služba umožňuje platit pouze za to, co se reálně využívá. Platí se podobně jako u Cloud Function za využitý čas běhu se zaokrouhlením na 100 ms. [24]

## 5 MICROSOFT AZURE

Microsoft Azure je veřejná cloudová platforma s více než 200 produkty a službami dostupnými přes internet. Jako ostatní poskytovatelé cloudových služeb spravuje a udržuje veškerý hardware, infrastrukturu a zdroje, které mohou být použity zdarma nebo s platbou podle využívání dle potřeby. [24]

Od svého vzniku v roce 2008 Microsoft Azure vyrostl do nynější podoby, a to do druhého největšího poskytovatele cloudových služeb – hned za Amazon Web Services (AWS) a před Google Cloud Platform (GCP). [24]

K druhému čtvrtletí roku 2021 AWS kontrolovalo 31 % trhu, Microsoft Azure 22 % a Google Cloud zůstal na 8 % tržního podílu. [24]

Dnes Microsoft Azure slouží milionům aplikací, integrací a zákazníků. [24]

### 5.1 Cloud computing

S Microsoft Azure lze využít Azure infrastrukturu a zvýšit tak operativní účinnost. Lze spouštět SQL Server, SAP, Oracle software a vysoce výkonné aplikace na Azure Virtual Machines. Azure Virtual Machines nabízí jak Linuxové distribuce, tak i Windows Servery.

K využití tato služba nabízí až 416 vCPU a 12 TB paměti. Lze se tak dostat až na 3,7 milionů lokálních IOPS na VM. Další výhodou je až 30Gbps ethernet a 200 Gbps pro první nasazení InfiniBand v cloudu. Lze vybrat z procesorových rodin AMP, Ampere (Arm procesor), nebo Intel. [25]

Microsoft Azure nabízí velkou škálu různých typů virtuálních strojů, níže je pro představu uvedeno několik příkladů. [30]

A-Series VM jsou základním a vstupním virtuálním strojem vhodným pro vývoj a testování. Jsou ekonomicky výhodné a poskytují nízkonákladové možnosti jak začít s Azure. Av2 standard je nejnovější generací non-hyper-threaded A-Series VM s podobným výkonem CPU, ale s více RAM na vCPU a rychlejšími disky. Azure má v plánu od A-Series VMs upustit 30. září 2024. [30]

D-Series jsou určeny pro běžné výpočty. Tyto virtuální stroje nabízí takovou kombinaci virtuálních CPU, paměti a dočasného úložiště, aby splňovala požadavky spojené s většinou produkčních pracovních zátěží. Tyto D-Series se dále dělí na další podskupiny a rozdíl mezi nimi je v konkrétních specifikacích. [30]

D v3 virtuální stroje jsou hyper-threaded a s 2,3 GHz Intel XEON E5-2673 v4 (Broadwell) procesorem jsou navrženy k běžným účelům. Mohou dosáhnout až 3,5 GHz s Intel Turbo Boost Technology 2.0. [31]

D v4 a Dd v4 VM jsou postaveny na procesoru Intel Xeon Platinum 8272CL, který běží základní rychlostí 2,5 GHz a může dosáhnout až turbo frekvence všech jader 3,4 GHz. Dd v4 virtuální stroje mají rychlejší a větší SSD úložiště (až 2 400 GiB) a jsou vhodné pro všechny aplikace s potřebou nízké latence a vysokorychlostního úložiště. D v4 nemají žádné dočasné úložiště. [31]

D v5 a Dd v5 disponují procesorem třetí generace Intel Xeon Platinum 8370C (Ice Lake) v multi-threaded konfiguraci. Mohou škálovat až na 96 vCPU s konfiguracemi jako D v4 a Dd v4. [31]

Da v4 a Das v4 Azure VM-series poskytují až 96 vCPU, 384 GiB RAM, 2 400 GiB SSD dočasného úložiště a procesor AMD EPYC 7452. [31]

Das a Dads v5 série virtuální strojů jsou založeny na třetí generaci AMD EPYC 7763v (Milan) procesoru. Tento procesor lze taktovat až na 3,5 GHz. Tato série je poskytována s (Dads v5) lokálním úložištěm a bez něj (Das v5). [31]

## 5.2 Databáze

Jako u ostatních poskytovatelů i u Microsoft Azure jsou uvedeni zástupci SQL a NoSQL cloudových databází.

### 5.2.1 Azure SQL

Azure SQL je rodina plně spravovaných, bezpečných a inteligentních SQL databázových služeb postavených na stejném SQL server enginu. Nabízí široké spektrum možností nasazení a řídí požadavky aplikačních vzorů, aby vyhovovaly nejnáročnějším iniciativám v oblasti migrace a modernizace. Pomocí Azure SQL lze znovu hostovat SQL úlohy na SQL Server na Azure VM, modernizovat stávající aplikace pomocí spravované instance Azure SQL a podporovat moderní cloudové aplikace pomocí Azure SQL Database. Také je možné spravovat celé portfolio prostřednictvím konzistentního a jednotného prostředí, které poskytuje pokyny založené na scénářích pro vytváření zdrojů, které splňují požadavky byznysu. [27]

### 5.2.2 Azure Cosmos DB

Azure Cosmos DB je globálně distribuovaná, multimodelová databázová služba v libovolném měřítku. Služba byla postavena od základu s globální distribucí a horizontálním měřítkem ve svém jádru. [27]

Díky globální distribuci na klíč napříč libovolným počtem oblastí Azure, Azure Cosmos DB transparentně škáluje a replikuje data, ať jsou uživatelé kdekoliv. Služba flexibilně škáluje zápisy i čtení dat po celém světě a platí se pouze za to, co je využíváno. Azure Cosmos DB poskytuje nativní podporu pro NoSQL a OSS API, včetně MongoDB, Cassandra, Gremlin, etcd, Spark a SQL. Nabízí také několik dobře definovaných modelů konzistence, garantované jednociferné milisekundové latence čtení a zápisu na 99. percentilu a garantovanou 99,999 % dostupnost s multi-homingem kdekoliv na světě – to vše je podpořeno špičkovými komplexními SLA (Service-level agreement). [27]

Azure Cosmos DB je plně spravovaná služba, která umožňuje přesunout administrativní zátěž operování a škálování distribuovaných databází do Azure, takže není třeba se zabývat spravováním virtuálních strojů, poskytováním hardwaru, nastavováním a konfigurací, kapacitou, replikací, záplatou softwaru nebo škálováním klastru. [27]

## 5.3 Funkce

Se službou Azure Functions lze vyvíjet účinněji s událostmi řízenou serverless výpočetní platformou, která pomáhá řešit komplexní orchestrální problémy. Sestavovat a krokovat lokálně bez dalšího nastavení, vystavovat a operovat v měřítku v cloudu a integrovat služby použitím spouštěčů a vazeb. [28]

Azure Functions lze použít s programovacími jazyky C#, JavaScript, Java, F# nebo Python. Mohou být využity k vytvoření API, webových aplikací pro statické webové stránky a single-page aplikací, mikroslužeb pro událostmi řízená scénária, postupů strojového učení, dat procesních pipeline pro soubory nebo dat v reálném čase, automatizace cloudu a hybridní správy zdrojů a dalšího, a to všechno serverless. [28]

## 6 REDHAT

Red Hat je softwarová společnost, která kombinuje open source Linux komponenty se souvisejícími programy do distribučních balíčků, které si může zákazník objednat. Byznys mód firmy Red Hat je založen na open source ekosystému, kde kolaborativní vývoj probíhá v komunitě profesionálů zaměřených na poskytování zajišťování kvality, testování a zákaznické podpory. Red Hat nabízí množství open source softwaru, jež využívají DevOps inženýři a podniky. Mezi tyto produkty patří platformy operačních systémů, úložiště, middlewary, produkty managementu, ale taky výcvik, podpora a konzultační služby. [31]

Tato společnost sice poskytuje cloudová řešení, ale pro naplnění cíle této práce nejsou její cloudové služby uvažovány, jelikož nejsou pro dané účely vhodné. Nicméně na AWS EC2 lze použít Red Hat Enterprise Linux. [32]

## 7 ORACLE

Oracle Cloud (OCI) byl spuštěn na konci roku 2016, jedná se tedy o jednoho z novějších cloudových poskytovatelů. Přestože tento poskytovatel je na trhu poněkud krátce, má i tak přes 30 cloud regionů. Jeho spolehlivost je dána i tím, že má stvrzené partnerství s Microsoftem. Aktuálně je hned několik regionů Oracle Cloudu společných s Microsoft Azure. Spolehlivost je zaručena velmi důkladnou end-to-end SLA smlouvou, která pokrývá výkon, dostupnost a spravovatelnost služeb. [29]

### 7.1 Cloud computing

OCL nabízí více druhů služeb pro cloud computing. Jsou to virtuální stroje, bare-metal stroje a ostatní služby. Tyto služby se dále dělí podle toho, na co je uživatel chce využít. [33]

Virtuální stroje výčtem:

- flexibilní
- standardní,
- Dense-IO,
- stroje s grafickou kartou (GPU-Accelerated),
- optimalizované,
- dedikované.

Dále jsou v podkapitolách podrobněji rozebrány flexibilní a standardní virtuální stroje, jelikož se pro tuto práci hodí nejvíce.

#### 7.1.1 Flexibilní stroje

Flexibilní stroje nabízí možnost téměř libovolné konfigurace, co se počtu jader a operační paměti týče, a to nejen při vytváření, ale i při změně velikosti. Lze takto volbou konkrétního počtu jader a velikosti paměti optimalizovat výdaje a přizpůsobit sílu stroje na potřebu pracovní zátěže. Síťová zátěž je rovnoměrně rozdělena úměrně k počtu jader virtuálního stroje. Konfiguraci lze vybrat z následujících možností: [33]

- 1-64 jader AMD EPYC procesorů s 1-1024 GB RAM. Tyto jsou v OCL také dostupné jako redukovatelné nebo preemptivní instance.
- 1-32 jader procesoru Intel s 1-512 GB RAM.
- 1-18 jader procesoru Intel s 1-256 GB RAM.

### 7.1.2 Standardní stroje

Standardní stroje poskytují na rozdíl od flexibilních předem dané konfigurace, které jsou vyvážené tak, aby splňovaly požadavky uživatele. Všechny tyto sestavy jsou dostupné jako preemptivní. V nabídce jsou tyto sestavy: [33]

- Intel nebo AMD procesory.
- 1, 2, 4, 8, 16 nebo 32 jader.
- 15-320 GB paměti.
- 1-24.6 Gbps síťového pásma.

## 7.2 Databáze

OCI databáze jsou doménou Oracle Cloudu. [29]

Oracle Exadata je výpočetní platforma optimalizovaná pro běh služby Oracle Database. Vyniká v aplikacích na procesování online transakcí (OLTP), které běží souběžně. [29]

Autonomní databáze je směs tradiční databáze s prvky strojového učení – nabízí automatizované záplatování, optimalizaci záloh a sebeopravování. [29]

### 7.2.1 Oracle Exadata

Databázová služba Exadata umožňuje klientům spouštět rychlé OLTP, in-memory analýzy a konvergovanou zátěž v jedné službě s pouhou 19mikrosekundovou latencí a s propustností analytických skenů až 2.88 TB za vteřinu. Díky automatické optimalizaci výkonu nechává běžet databáze v optimálním prostředí, i když se změní vzorce pracovní zátěže. [35]

Tato služba také odkládá SQL a analytické operace na servery s inteligentním úložištěm, které nekonzumují prostředky vCPU, což dovoluje uživatelům spouštět analýzy v kratším čase, spotřebovat méně vCPU prostředků a redukovat náklady. [35]

### 7.2.2 Autonomní databáze

Jedná se o plně automatizovanou databázovou službu optimalizovanou pro běh transakčních, analytických a hromadných pracovních zátěží souběžně. Je přednastavená na řádkový formát, indexy a cacheování dat ke zvýšení výkonu během poskytování škálovatelnosti, dostupnosti, transparentního zabezpečení a operačních analýz v reálném čase. S Autonomní databází lze vyvíjet a vystavovat aplikace s nízkými náklady a jednoduše, aniž by bylo potřeba porušit funkcionalitu ACID (atomicita, konzistence, izolace, odolnost). [36]

### 7.3 Funkce

Oracle Cloud Functions je serverless platforma, která dovoluje vývojářům vytvářet, spouštět a škálovat aplikace, aniž by museli spravovat nějakou infrastrukturu. Functions jsou integrované s OCL infrastrukturou, platformními službami a SaaS aplikacemi. Protože Functions je založena na open source Fn projektu, developéři mohou vytvářet aplikace, které jsou jednoduše přenositelné na jiné cloudové nebo on-premises prostředí. Kódy založeny na Functions obvykle pracují jen po krátký čas a zákazník platí jen za prostředky, které byly použity. [37]



## 8 VÝVOJOVÉ NÁSTROJE PRO PRÁCI S CLOUDEM

Vývojové nástroje pro práci s cloudem lze rozdělit dle zaměření do tří kategorií:

- obecná vývojová prostředí,
- software development kit (SDK) a
- konzole. [48]

Všechny typy knihoven a vývojových nástrojů se odvíjí od programovacího jazyku, který chceme použít. [48]

### 8.1 Obecná vývojová prostředí

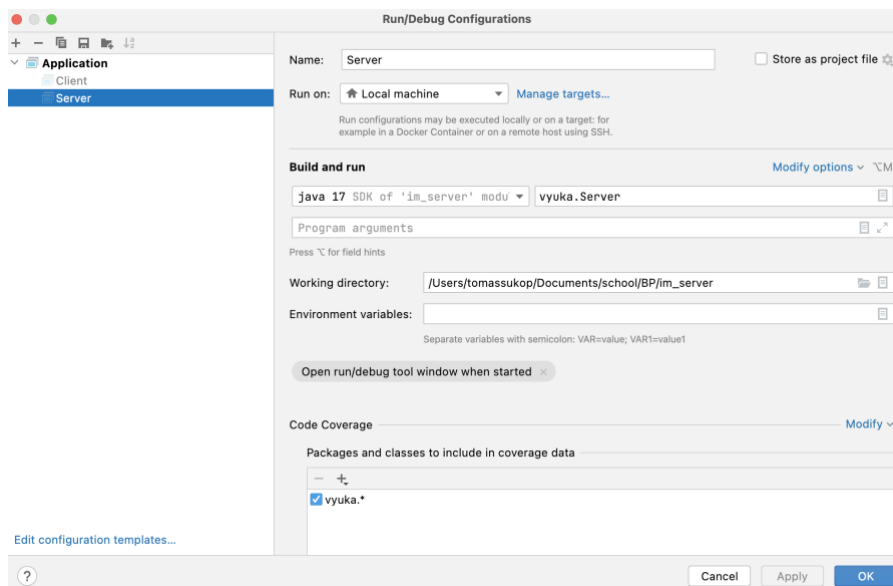
Pro vývoj aplikací se používají klasická vývojová prostředí (IDE). Jako vhodné pro naplnění cíle této práce byly vybrány následující dvě, a to IntelliJ IDEA od JetBrains a Visual Studio Code od Microsoftu.

#### 8.1.1 IntelliJ IDEA

Toto IDE poskytuje ladící program pro kód psaný v Javě. V závislosti na nainstalovaných nebo zapnutých plug-inech, lze ladit také kód psaný v jiných jazycích. [38]

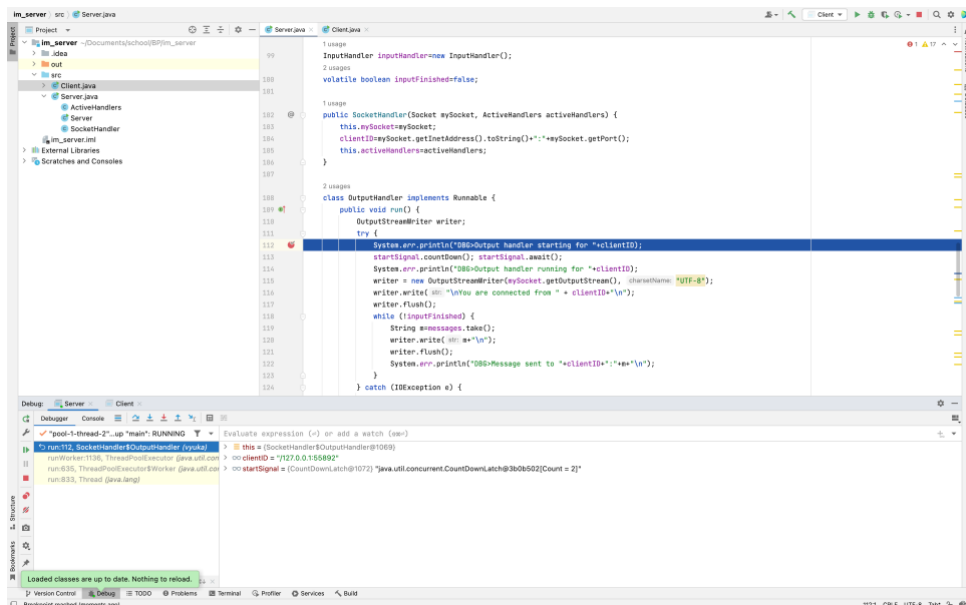
Během procesu ladění se spustí program s připnutým ladícím programem. Účel tohoto programu je zasahovat do vykonávání laděného programu a poskytovat informace o tom, co se děje takzvaně pod pokličkou. Tento proces usnadňuje proces detekování a opravování chyb v napsaném programu. [38]

Předtím, než se spustí ladění, je potřeba zkontrolovat nastavení. Je potřeba se ujistit, zda v nastavení Java kompilátoru je zaznačeno generování ladících informací. Není to úplně nutné pro ladění programu, ale JetBrains to doporučuje pro zrychlení a vyšší kvalitu fungování ladícího programu. [38]



Obrázek 1 Ladící konfigurace v IntelliJ IDEA [38]

Pokud je třeba použít vlastní ladící konfiguraci, je nezbytné ji definovat. Lze nastavit například argumenty při spuštění programu nebo další speciální aktivity, které by se měly provést před spuštěním. Nicméně u jednoduchých programů (jako například v této práci použitý Server.java) je naprosto dostačující ponechat základní nastavení. [38]



Obrázek 2 Ladění serveru v IntelliJ IDEA [38]

## 8.1.2 Visual Studio Code

Stejně jako IntelliJ IDEA i VS Code přichází s vlastním ladícím programem. V defaultním ladícím programu je jediné omezení, a to sice, že je pouze pro Node.js, takže s ním lze ladit

pouze jazyky, které jsou transpilovány do JavaScriptu. Pokud chceme VS Code používat na jiný jazyk, je potřeba stáhnout ladící program z VS Code Marketplace. [39]

Ke spuštění ladícího programu ve VS Code slouží tlačítko „Run and Debug“ v levém panelu, nebo klávesa F5 a IDE se pokusí spustit právě otevřený soubor. Nicméně pro většinu ladících scénářů je výhodné vytvořit konfigurační soubor pro nastavení ladícího programu. VS Code tento soubor ukládá jako „launch.json“ ve „.vscode“ adresáři v kořenové složce projektu. K vytvoření tohoto souboru stačí kliknout na tlačítko pro vytvoření v nabídce pro spuštění projektu. Visual Studio se automaticky pokusí detekovat nainstalovaný ladící program, pokud se mu to nepodaří, musí se vybrat z nabídky. [39]

```
.vscode > {} launch.json > ...
1  {
2      // Use IntelliSense to learn about possible attributes.
3      // Hover to view descriptions of existing attributes.
4      // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
5      "version": "0.2.0",
6      "configurations": [
7          {
8              "type": "java",
9              "name": "Launch Current File",
10             "request": "launch",
11             "mainClass": "${file}"
12         }
13     ]
14 }
```

Obrázek 3 Soubor s konfigurací ladícího programu ve VS Code [39]

Atributy dostupné v konfiguračním souboru se mohou lišit podle použitého programovacího jazyku. Ke zjištění, jaké všechny atributy jsou k dispozici, lze využít Intelli sense. To je většinou dostupné pod klávesovou zkratkou ctrl+mezerník (cmd+mezerník). [39]

## 8.2 Software development kit (SDK)

Pokud již máme aplikaci naprogramovanou a odladěnou, je potřeba ji vystavit na cloud, na server. K tomu lze použít více způsobů. Jedním z nich je využití SDK.

SDK je souprava nástrojů, kterou používají vývojáři aplikací k sestavování aplikací pomocí předpřipravených komponent namísto vytváření každé komponenty samostatně. SDK jsou navrženy tak, aby pracovaly se specifickými operačními systémy, hardwarem nebo programovacími jazyky. [40]

Dále jsou uvedeny příklady SDK pro programovací jazyk Java tří největších cloudových poskytovatelů, tedy AWS, Microsoft Azure a GCP.

### 8.2.1 AWS Java SDK

SDK pro Javu zjednodušuje používání AWS služeb poskytováním množiny knihoven, které jsou konzistentní a známé pro Java vývojáře. Poskytují podporu pro API funkce, jako jsou správa pověření, opakování pokusů, řazení dat a serializace. AWS SDK pro Javu také nabízí vyšší úroveň abstrakce pro zjednodušení vývoje. Veškeré knihovny lze najít na GitHubu AWS. [41]

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.dynamodb.DynamoDbAsyncClient;
import software.amazon.awssdk.services.dynamodb.model.ListTablesRequest;
import software.amazon.awssdk.services.dynamodb.model.ListTablesResponse;
import java.util.List;
import java.util.concurrent.CompletableFuture;
```

Obrázek 4 Příklad použitých knihoven AWS Java SDK [41]

### 8.2.2 MS Azure Java SDK

Knihovny Azure slouží ke komunikaci s Azure službami z kódu napsaného v Javě, který buďto běží na lokálním stroji, nebo na cloudu. Tyto knihovny podporují Javu verze 8 a novější a jsou otestovány na základě Java 8 a nejnovějších dlouhodobě podporovaných verzích. Knihovny SDK jsou rozděleny na administrátorské a klientské. Každá z nich slouží k jinému účelu, jelikož používají jiný kód. [42]

Azure SDK pro Javu je postavená na Azure REST API, čímž dovoluje používat koncové body skrz známá Java paradigmatu. Nicméně pokud by programátor chtěl, je možné používat tyto koncové body přímo z Java kódu. Zdrojové kódy jsou k nalezení na GitHubu Azure. [42]

```

VirtualMachine virtualMachine = computeManager.virtualMachines()
    .define(<your virtual machine>)
    .withRegion(Region.US_WEST)
    .withExistingResourceGroup(<your resource group>)
    .withNewPrimaryNetwork("10.0.0.0/28")
    .withPrimaryPrivateIpAddressDynamic()
    .withoutPrimaryPublicIpAddress()

    .withPopularLinuxImage(KnownLinuxVirtualMachineImage.UBUNTU_SERVER_18_04_LTS)
    .withRootUsername(<virtual-machine username>)
    .withSsh(<virtual-machine SSH key>)
    .create();

```

Obrázek 5 Příklad nastavení virtuálního stroje pomocí Azure SDK [42]

### 8.2.3 Google API Client knihovna pro Javu

Knihovna Google API Client pro Javu poskytuje funkcionality běžné pro všechny Google API, například http požadavky, správu chyb, autorizaci, čtení souborů formátu JSON, stahování nebo nahrávání médií a hromadné operace. Samotná knihovna obsahuje například OAuth 2.0 knihovnu s konzistentním rozhráním. [43]

K zavolání konkrétní Google API je potřeba mít vygenerovanou Java knihovnu specifickou pro API, ke které chceme přistoupit. Tyto knihovny obsahují hlavní knihovnu „google-api-java-client“ společně s informacemi ke specifické API. Také obsahují třídy a entity v kontextu dané API. Mezi vybrané Google API patří například BigQuery API nebo Compute Engine API. [43] [44]

```

/** Authorizes the installed application to access user's protected data. */
private static Credential authorize() throws Exception {
    // Load client secrets
    GoogleClientSecrets clientSecrets = GoogleClientSecrets.load(JSON_FACTORY,
        new InputStreamReader(CalendarSample.class.getResourceAsStream("/client_secrets.json")));
    // Set up authorization code flow
    GoogleAuthorizationCodeFlow flow = new GoogleAuthorizationCodeFlow.Builder(
        HttpTransport, JSON_FACTORY, clientSecrets,
        Collections.singleton(CalendarScopes.CALENDAR)).setDataStoreFactory(dataStoreFactory)
        .build();
    // Authorize
    return new AuthorizationCodeInstalledApp(flow, new LocalServerReceiver()).authorize("user");
}

```

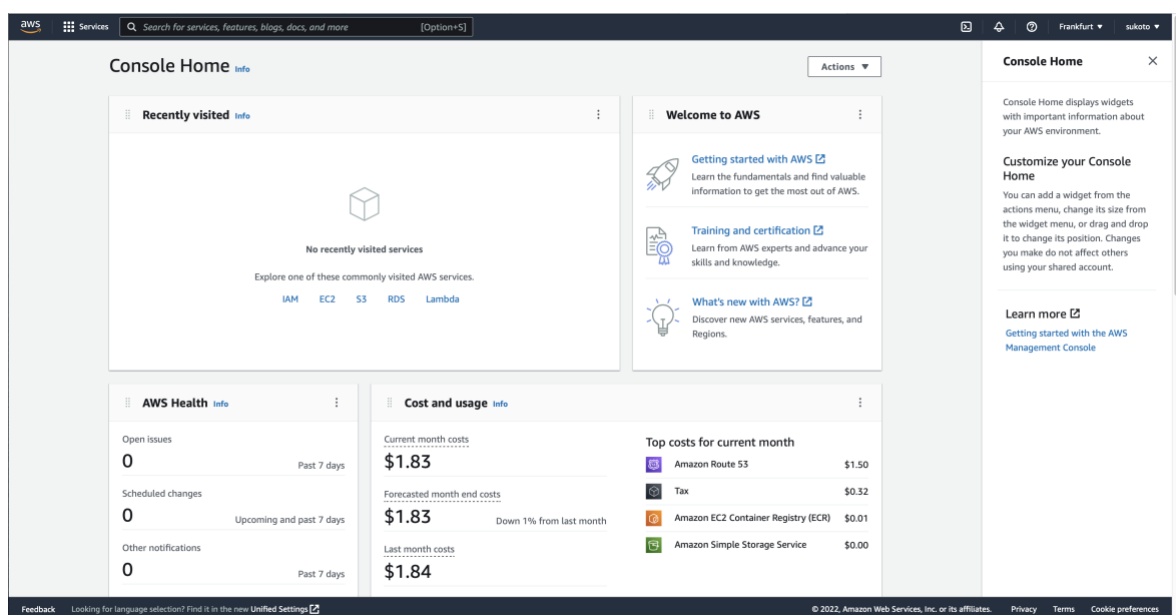
Obrázek 6 Příklad autorizace pomocí Google API Client knihovny [43]

## 8.3 Konzole

Konzole jsou webové aplikace, přes které lze nastavovat a spravovat jednotlivé cloudové služby. Níže je uveden přehled konzolí pro AWS, MS Azure a GCP a popis jejich základních použití. Přes tyto konzole je provedena celá praktická část této práce.

### 8.3.1 AWS Management Console

AWS Management Console je webová aplikace, která nabízí širokou kolekci služeb pro správu AWS prostředků. Při přihlášení lze vidět domovskou stránku celé aplikace. Domovská stránka nabízí přístup ke každé službě a nabízí jednotné místo s veškerými dostupnými informacemi, které jsou potřeba k vykonávání úkonů v rámci AWS. Domovskou stránku je možné uzpůsobit vlastním potřebám přidáním, odebráním, nebo přeskládáním doplňků, mezi které například patří naposledy navštívené služby, AWS Health, Trusted Advisor a další. [45]

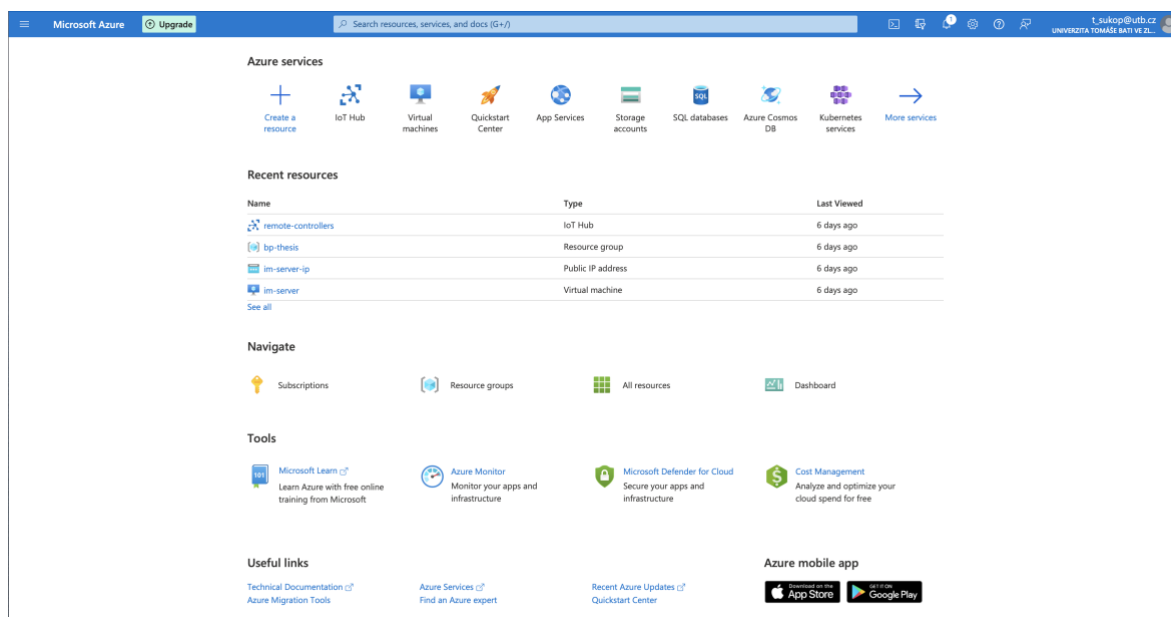


Obrázek 7 AWS Management Console [45]

### 8.3.2 Microsoft Azure Portal

S konzolí Microsoft Azure Portal lze nastavovat různé doplňky, které usnadní přístup ke spravovaným službám. S využitím nastavení rolí lze určovat přístupová oprávnění, respektive jejich rozsah pro jednotlivce nebo skupiny. [47]

Kromě této konzole nabízí ještě Microsoft Azure mobilní aplikaci, takže umožňuje přístup ke službám prakticky odkudkoliv. [47]

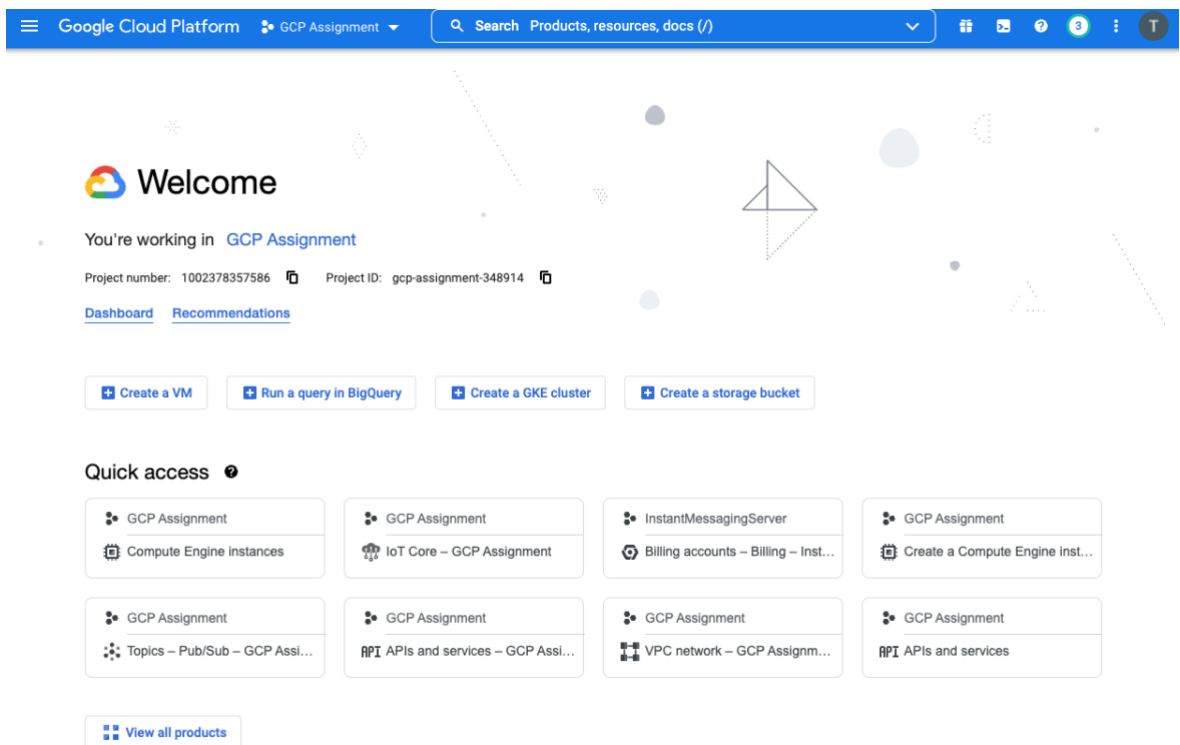


Obrázek 8 Portál Microsoft Azure [47]

### 8.3.3 Cloud Console

Cloud Console od Googlu nabízí možnost spravovat všechno na čem běží cloudová aplikace, ať už to jsou webové aplikace, analýzy dat, virtuální stroje, úložiště dat, databáze, síťová propojení nebo vývojářské služby. Pomáhá vývojářům vystavovat, škálovat a diagnostikovat problémy v aplikacích přes jednoduché webové rozhraní. Umožňuje dále připojení přes SSH k virtuálním strojům přímo v prohlížeči a lze přes ni provádět veškeré pracovní úkony v rámci DevOps. [46]

Mezi klíčové prvky patří například správa prostředků, účtování služeb, SSH v prohlížeči nebo administrace, ve které lze spravovat přístupy ostatních uživatelů. [46]



Obrázek 9 Google Cloud Console [46]



## 9 VLASTNOSTI A ZPŮSOB POUŽITÍ BEZPLATNÝCH SLUŽEB

U služeb AWS a GCP je potřeba při zakládání účtu vložit platební kartu, přes kterou se budou užívané služby platit, a to i v případě, kdy jde o využití pouze služeb poskytovaných bezplatně. Slouží to k ověření toho, že zakládající uživatel není robot, a jako pojistka, že pokud by byl při užívání překročen limit služeb poskytovaných zdarma, bude to uhrazeno (v tomto případě prostřednictvím platební karty zadané při registraci). [48]

U služby MS Azure je situace trochu jiná. Existují dva způsoby, jak využívat tyto služby jako fyzická osoba nebo jako student. Pokud by uživatel mohl využít a využil variantu MS Azure pro studenty, nebyla by zapotřebí žádná platební karta, pouze ověření školního emailu. [54]

Jedním z předpokladů naplnění cíle této práce je právě využívání cloudových služeb poskytovaných bezplatně, a to optimálně bez nutnosti vkládat platební kartu. V následujících podkapitolách je popsáno, jak je možné takového cíle dosáhnout. U každého poskytovatele je vždy možnost vytvořit si vlastní účet vložením platební karty, pro zjednodušení a větší přehlednost je tato možnost brána jako fundamentální; zmíněny jsou jen ty metody a způsoby, které nepotřebují využití vlastní karty. Nutno ovšem podotknout, že všechny způsoby vyžadují mít alespoň jeden účet se zadanou platební kartou (například učitelský), pod který budou různými způsoby vytvářeny účty další (například studentské). [48]

### 9.1 AWS

Služby zdarma od AWS se dělí na 3 typy:

- a) krátkodobé nabídky služeb zdarma,
- b) služby poskytované zdarma po 12měsíční lhůtu od vytvoření účtu a
- c) služby, které jsou zdarma vždy. [48]

Na stránkách poskytovatele lze přesně vyhledat které služby spadají do kterého typu. Například služba EC2 spadá pod b), tedy do služeb poskytovaných zdarma v rámci 12měsíční lhůty. [48]

U AWS existují kromě platební karty další dva způsoby, jak vytvořit uživatelské účty, které mohou využívat nabídku zdarma.

- pomocí služby Identity and Access Management (IAM),

- pomocí organizací.

### 9.1.1 IAM služba

Přidání uživatele pomocí IAM služby je poměrně jednoduché, nicméně to s sebou nese jistá omezení. Vytvořit uživatele lze v konzoli IAM pomocí formuláře. Je potřeba zvolit uživatele s heslem a zadat jeho email. Tomuto uživateli přijde email s informací o tom, že byl přidán k účtu AWS a musí dokončit registraci. Dalším krokem je nastavení oprávnění uživatelů, které přidáváme. Zde je vhodné omezit přístup uživatelů pouze na nezbytný rozsah oprávnění. IAM totiž kromě správy uživatelů umožňuje vytvářet a spravovat také zásady, což jsou skupiny oprávnění a role, kterým lze přiřadit zásadu. Vhodné je tedy vytvořit například zásadu vhodnou pro studenty a s touto zásadou nakonec může být nový uživatel, popřípadě noví uživatelé vytvoření. [48]

Create policy

1 2 3

Review policy

Name\* student\_policy  
Use alphanumeric and '+', '@', '-' characters. Maximum 128 characters.

Description  
Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.

Summary  
This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an applicable resource or condition. For details, choose [Show remaining](#). [Learn more](#)

Q Filter

Service	Access level	Resource	Request condition
Allow (1 of 324 services) <a href="#">Show remaining 323</a>			
EC2	Full: Tagging Limited: List, Read, Write	Multiple	None

Tags  
Key Value  
No tags associated with the resource.

\* Required

Cancel Previous Create policy

Obrázek 10 Příklad vytvoření zásad v IAM AWS [48]

Add user 1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[Add another user](#)

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type\*  Access key - Programmatic access  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

Password - AWS Management Console access  
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password\*  Autogenerated password  
 Custom password

Require password reset  User must create a new password at next sign-in  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

Obrázek 11 Vytvoření uživatele v IAM AWS [48]

Add user 1 2 3 4 5

Set permissions

[Add user to group](#) [Copy permissions from existing user](#) [Attach existing policies directly](#)

[Create policy](#) [Refresh](#)

Filter policies  Showing 1 result

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	▶ student_policy	Customer managed	None

Set permissions boundary

Obrázek 12 Přiřazení zásady účtům v AWS IAM [48]

Takto vytvoření uživatele sdílí jednu společnou službu, takže mohou například studenti fungovat společně s pouze jedním virtuálním strojem EC2. [48]

### 9.1.2 Organizace

Druhou možností je každému novému uživateli vytvořit pod e-mailem ve speciálním tvaru celou organizaci, ve které bude mít každý svoji nabídku zdarma. Tento proces nicméně vyžaduje náročnější zapojení vyučujícího a vytvoření skriptu pro zakládání takovýchto uživatelů. [48]

Každý standardně vytvořený účet může vytvořit organizaci nebo až 50 pod-organizací, kdy každá z nich má svůj platební plán; jelikož veškeré poplatky jsou vztaženy k organizaci hlavní, každá pod-organizace bude mít i svůj plán zdarma. [48]

Algoritmus pro tento skript by byl pak následující.

1. Přihlásit se jako IAM uživatel s administrátorským oprávněním do hlavní organizace.
2. Založit novou organizaci.
3. Přijmout roli, která byla vytvořena v nové organizaci. K tomuto je potřeba id organizace a jméno role v IAM (OrganizationAccountAccessRole).
4. Založit EC2 a povolit na ní přístup s veřejným SSH klíčem studenta. Z tohoto kroku je viditelná podmínka vytvoření SSH klíčového páru studentem a poskytnutí jeho veřejné části vyučujícímu. Nicméně SSH klíč lze dodat později, nemusí jít s vytvořením EC2 instance.
5. Nakonec se již stačí připojit přes SSH ke správné instanci a využívat ji podle plánu zdarma.

Skript může být napsán jako bash skript s pomocí AWS cli, nebo pomocí AWS SDK v libovolném programovacím jazyku. Tento postup vytvoření uživatele pro organizaci umožní studentům používat výše zmíněnou EC2 po celý rok, aniž by se o ni museli dělit, nebo aby museli zadávat citlivé údaje. [48]

## 9.2 GCP

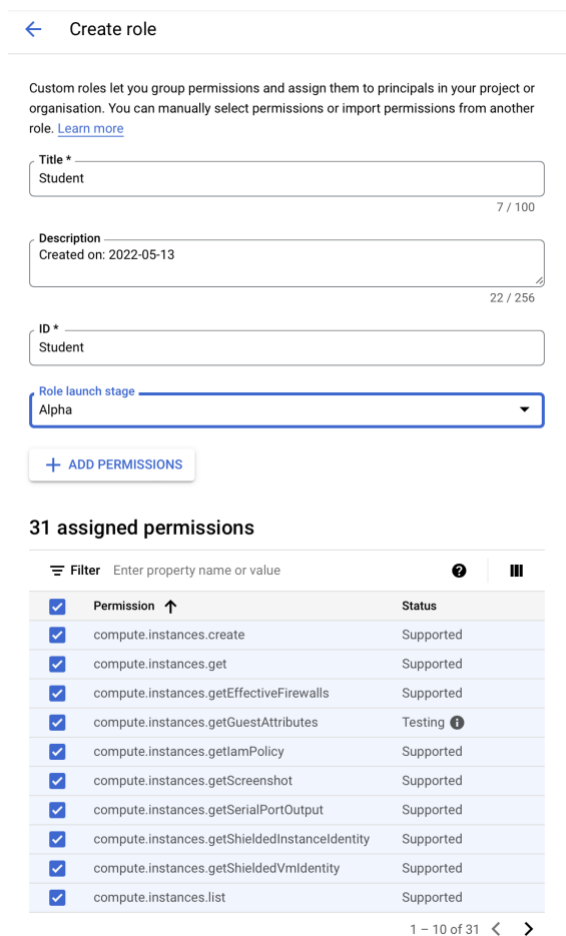
Google Cloud Platform stejně jako AWS má k dispozici IAM službu, přes kterou lze vytvářet další uživatele a tím jim umožnit přístup k prostředkům. Stejně tak umí pracovat s konceptem organizace, bohužel nedovoluje pracovat s organizacemi způsobem jako AWS. Z toho důvodu lze u tohoto poskytovatele přidělovat účty pouze pomocí IAM. [48]

Model, který nabízí GCP zdarma, se od AWS liší tím, že nemá žádné omezené trvání jako tomu je například u EC2 od AWS, kdy po uplynutí 12měsíční lhůty od založení účtu začíná běžet zpoplatňování dané instance. Kromě toho, že nabízí určité produkty zdarma, dává ještě uživateli na 90 dní 300\$ kredit, který uživatel může a nemusí využít. [49][50]

Například nabídka výpočetní VM instance e2-micro je na každý měsíc zdarma s tím, že není omezen počet instancí, ale součet délek běhů jednotlivých instancí nesmí přesáhnout počet hodin v měsíci. Pokud by tedy mělo 30 studentů používat tuto službu v dubnu, kdy v dubnu je dohromady 720 hodin, každý ze studentů by měl k dispozici 24 hodin běhu jedné instance. [50]

### 9.2.1 IAM služba

V IAM službě lze vytvořit roli, která potom může být přiřazena vytvářenému uživateli. Tyto role obsahují povolení ke službám, které uživatel může spravovat. [48]



← Create role

Custom roles let you group permissions and assign them to principals in your project or organisation. You can manually select permissions or import permissions from another role. [Learn more](#)

Title \*  
Student 7 / 100

Description  
Created on: 2022-05-13 22 / 256

ID \*  
Student

Role launch stage  
Alpha

+ ADD PERMISSIONS

#### 31 assigned permissions

Filter Enter property name or value

Permission ↑	Status
<input checked="" type="checkbox"/> compute.instances.create	Supported
<input checked="" type="checkbox"/> compute.instances.get	Supported
<input checked="" type="checkbox"/> compute.instances.getEffectiveFirewalls	Supported
<input checked="" type="checkbox"/> compute.instances.getGuestAttributes	Testing ⓘ
<input checked="" type="checkbox"/> compute.instances.getIamPolicy	Supported
<input checked="" type="checkbox"/> compute.instances.getScreenshot	Supported
<input checked="" type="checkbox"/> compute.instances.getSerialPortOutput	Supported
<input checked="" type="checkbox"/> compute.instances.getShieldedInstanceIdentity	Supported
<input checked="" type="checkbox"/> compute.instances.getShieldedVmIdentity	Supported
<input checked="" type="checkbox"/> compute.instances.list	Supported

1 - 10 of 31 < >

Obrázek 13 Vytváření role v GCP IAM [48]

Add principals to GCP Assignment

**Add principals and roles for GCP Assignment resource**

Enter one or more principals below. Then select a role for these principals to grant them access to your resources. Multiple roles allowed. [Learn more](#)

New principals

T.Sukop@gmail.com

Role \*

Student

Condition

[Add condition](#)

Created on: 2022-05-13

[+ ADD ANOTHER ROLE](#)

SAVE CANCEL

Obrázek 14 Vytvoření uživatele v GCP IAM [48]

Nově vytvořený uživatel pak bude mít povolen přístup přesně tam, kam vyučující zamýšlel. Všichni nově vytvoření uživatelé ale musí mít buď email spojený s Google účtem nebo Google skupinou, popřípadě účet služeb nebo musí mít Google Workspace doménu. Určitě je ovšem možné založit organizaci UTB a zařídit přístup studentům s e-mailovou adresou @utb.cz. [48]

### 9.3 MS Azure

Azure nabízí dvě skupiny služeb, které jsou poskytovány zdarma, a další služby, které jsou dostupné pouze v omezených měsíčních limitech. První skupinou jsou služby, které jsou zdarma prvních 12 měsíců. K tomu uživatel dostane 200\$ kredit na 30 dní. Dalších více než 25 služeb je vždy zdarma. [51]

Mezi služby, které jsou zdarma první rok, patří například Azure Virtual Machines Linux a Windows nebo Azure SQL databáze. Mezi služby, které jsou zdarma vždy, pak třeba Azure Functions nebo Azure DevOps. [51]

Rozdělování přístupů a vytváření uživatelů probíhá u Microsoftu přes službu Azure Active Directory (Azure AD) [52]

Azure nabízí také možnost používání cloudu pro studenty. Každý student dostane 100 \$ na rok a po roce je potřeba se znovu autorizovat jako student dané univerzity – pak k prostředkům k využití pro platbu za služby přibude dalších 100 \$. Aby se uživatel mohl přihlásit o studentský účet u MS Azure, musí splňovat následující podmínky:

- student musí potvrdit, že je starší 18 let a studuje akreditované dvou nebo čtyř leté studium, za jehož úspěšné zakončení obdrží titul, přičemž tento akademický status je nutné ověřit přes školní e-mail,
- musí splňovat další podmínky pro užívání, které jsou dostupné z webových stránek MS Azure.

Velkou výhodou je, že student oproti ostatním službám, pokud využije nabídku MS Azure, nepotřebuje platební kartu k založení účtu. [54]

### 9.3.1 Azure Active Directory

Azure Active Directory je služba spravující přístupy a identity založená na cloudu. Tato služba pomáhá uživatelům přistupovat k externím zdrojům jako Microsoft 365, Azure Portal a dalším SaaS aplikacím. Azure AD také pomáhá s přístupem k interním zdrojům, jako jsou aplikace na intranetu nebo cloudová řešení mimo Azure. [53]

Azure AD má více druhů licencování. Licence zdarma poskytuje správu uživatelů a skupin, synchronizaci složek na místních počítačích, základní reporting, možnost změny hesla pro uživatele na cloudu a jednotné přihlášení na Azure, Microsoft 365 a další populární SaaS aplikace. [53]

## **II. PRAKTICKÁ ČÁST**



## 10 VYTVOŘENÍ ZADÁNÍ DO CVIČENÍ

V souladu se zadáním práce byla navržena a vytvořena dvě zadání do cvičení:

- a) Vytvoření a nahrání Instant Messaging (IM) serveru na cloud, který umožní uživateli komunikaci s ostatními .
- b) Naprogramování mikro počítače na ovládání jiného systému a jeho následné napojení na IoT službu.

Tato zadání jsou popsána podrobně v následujících podkapitolách.

### 10.1 IM server v Javě

Prvně vytvoříme jednoduchý IM server, který umožní přeposílat zprávy všem připojeným klientům. Tento server následně nahrajeme na virtuální server tak, aby se k němu mohla na daném portu připojit jiná aplikace (klient) a umožnila komunikovat s ostatními klienty.

Toto zadání lze uvažovat například v předmětu „Programování síťových aplikací“.

### 10.2 MPC v IoT

Naprogramujeme libovolný mikro počítač s infračerveným portem, který bude schopen ovládat libovolný přijímač. Takto bude náš MPC vysílat signál nahoru nebo dolů a následně jej napojíme do služby IoT, přes kterou ho pak budeme pomocí aplikace ovládat.

Bude tedy potřeba naprogramovat MPC, připojit jej do IoT služby a vytvořit jednoduchou aplikaci, která přes IoT bude ovládat vybrané zařízení.

## 11 VZOROVÉ IMPLEMENTACE

Vzorová řešení implementace byla aplikována u tří největších cloudových poskytovatelů, konkrétně AWS, GCP a Azure. U každé implementace byl použit stejný IM Server a programovací deska.

Za IM server byl zvolen server napsaný v programu Java z předmětu Technologie JAVA. Jedná se o jednoduchý server, který po obdržení zprávy odeslané klientem rozešle všem připojeným klientům předvolenou zprávu.

Jako zařízení pro zadání MPC v IoT byla zvolena vývojová deska s ESP32. Řešení bylo naprogramováno v PlatformIO. K zařízení je připojen RF 443 MHz transmitter.

### 11.1 AWS

U AWS byly použity pro každé zadání jiné služby. Pro IM Server byla použita instance EC2. Pro MPC to bylo AWS IoT Core a AWS IoT MQTT broker.

Následuje popis jednotlivých kroků realizace.

#### 11.1.1 IM Server

Nejprve jsme vytvořili instanci virtuálního serveru EC2, následně k němu byl připojen, nahrán a spuštěn připravený server.

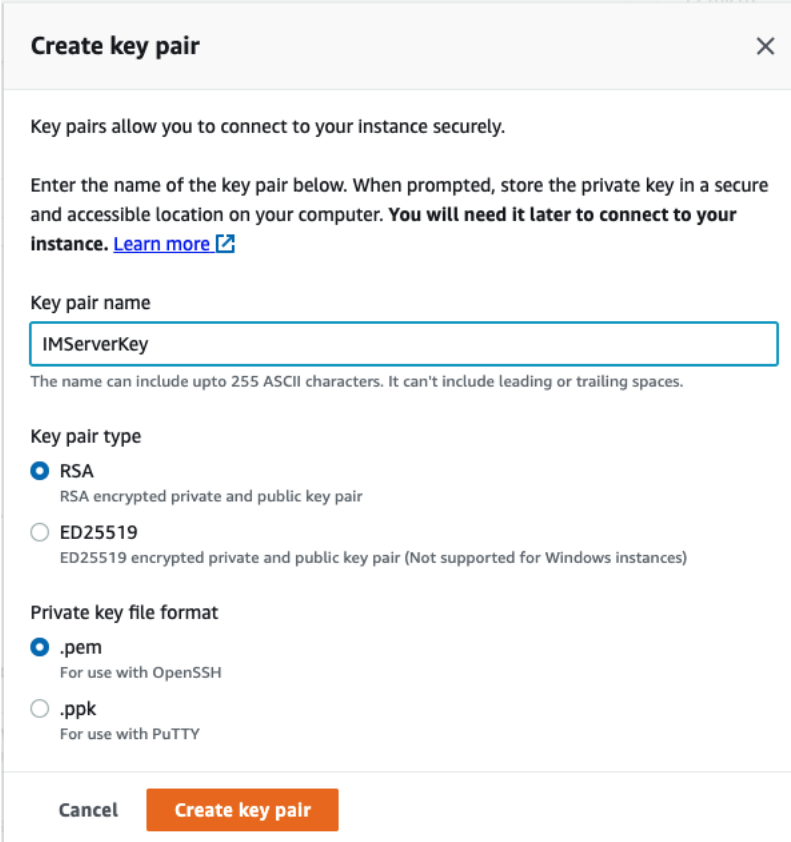
##### *11.1.1.1 Vytvoření instance EC2*

V prvním kroku vytvoření instance EC2 je třeba provést kontrolu volby regionu, a to tak, že se do vyhledávání služeb zadá „EC2“ a následně je provedena kontrola, zda se nacházíme ve správně zvoleném regionu. Z hlediska odezvy se jako ideální pro Českou republiku jeví region v Evropě, v tomto případě byl zvolen eu-central-1 neboli Frankfurt. Nyní vytvoříme EC2 instanci ins názvem Instant Messaging Server a nasazení obrazu Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volume Type s architekturou 64-bit (x86), který je vhodný pro užívání zdarma, na tuto instanci.

Jako typ instance vybere t2.mi, cro z rodiny t2 s 1 vCPU a 1 GiB paměti, opět vhodný pro užívání zdarma.

Je potřeba vytvořit ssh klíčový pár (private, public) k bezpečnému připojení k instanci. Je možné použít vlastní ssh klíč nebo vytvořit klíč nový. K vytvoření klíče lze použít více

možných aplikací, namátkou OpenSSH, PuTTY a v tomto případě i přímo AWS. Vytvoříme klíč pomocí AWS služby.



**Create key pair** ×

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#) ↗

Key pair name

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA  
RSA encrypted private and public key pair

ED25519  
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

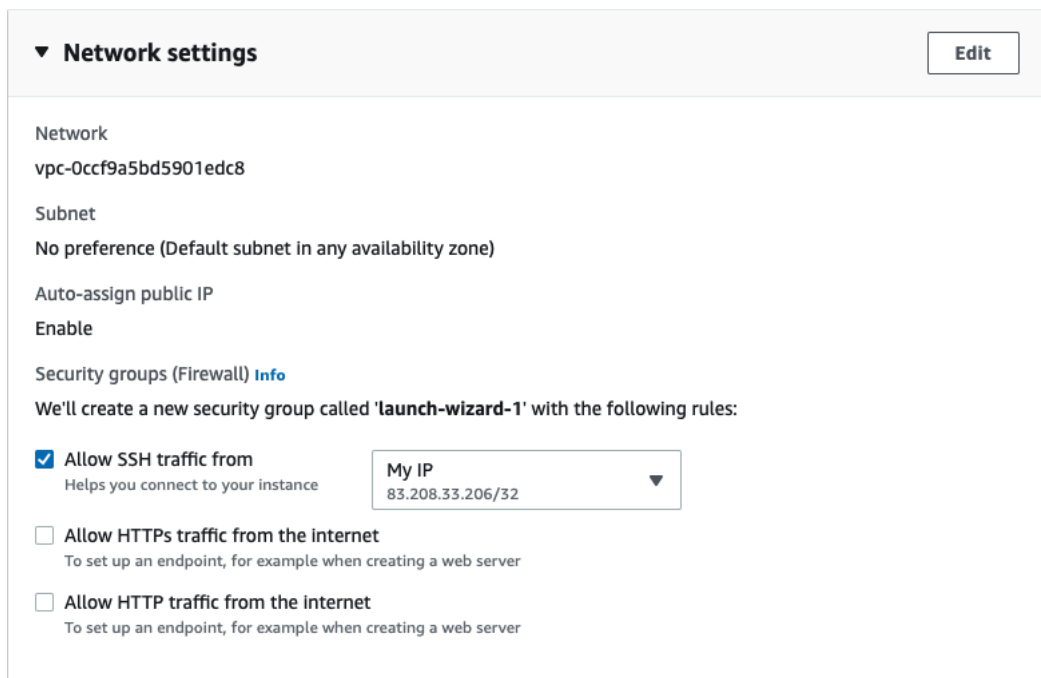
.pem  
For use with OpenSSH

.ppk  
For use with PuTTY

Cancel **Create key pair**

Obrázek 15 Vytvoření klíčového páru pomocí AWS [vlastní]

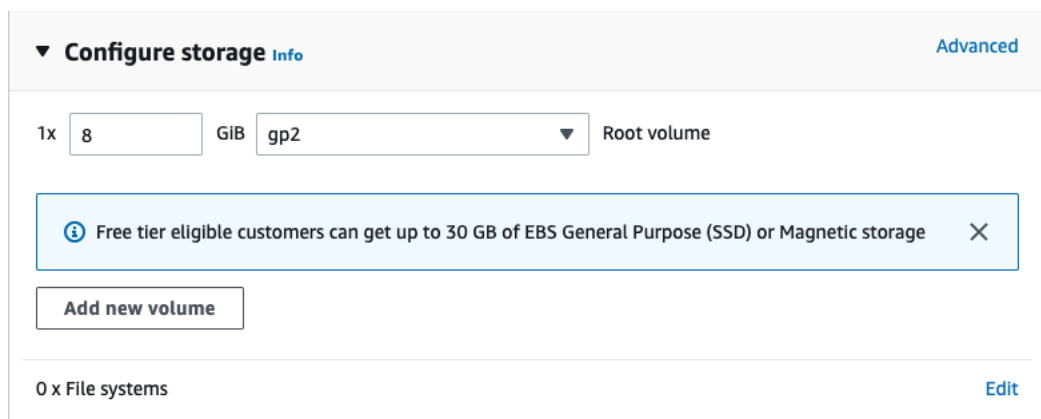
Dále nastavíme chování instance na síti. Defaultní nastavení umožňuje přístup k instanci odkudkoliv, z tohoto důvodu nastavíme povolení přístupu pouze z naší aktuální IP adresy. Toto nastavení je možno upravit později ve skupinách zabezpečení.



The screenshot shows the 'Network settings' section of an AWS console. At the top right is an 'Edit' button. The settings include: Network ID: vpc-0ccf9a5bd5901edc8; Subnet: No preference (Default subnet in any availability zone); Auto-assign public IP: Enable; Security groups: A new group named 'launch-wizard-1' will be created. Underneath, there are three options for traffic: 'Allow SSH traffic from' (checked) with a dropdown set to 'My IP' (83.208.33.206/32); 'Allow HTTPs traffic from the internet' (unchecked); and 'Allow HTTP traffic from the internet' (unchecked).

Obrázek 16 Nastavení sítě pro instanci EC2 [vlastní]

V dalším kroku musíme nastavit úložiště. Uživatelé, kteří dosáhnou na Free tier, mohou užívat až 30 GB EBS General Purpose (SSD) nebo magnetické úložiště. Pro potřeby tohoto zadání si vystačíme s předdefinovanými 8 GiB.



The screenshot shows the 'Configure storage' section. It features a dropdown for '1x' set to '8' GiB and another dropdown set to 'gp2' for the 'Root volume'. A blue information banner states: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage'. Below the banner is an 'Add new volume' button. At the bottom, it shows '0 x File systems' and an 'Edit' button.

Obrázek 17 Nastavení úložiště [vlastní]

Před spuštěním instance AWS vypíše shrnutí toho, co se pokoušíme spustit. Pro naše potřeby postačí 1 instance Amazon Linuxu s typem t2.micro s novou security group a jedním svazkem s velikostí 8GiB.

### ▼ Summary

Number of instances [Info](#)

[Software Image \(AMI\)](#)

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220419.0  
x86\_64 HVM gp2  
ami-0006ba1ba3732dd33

[Virtual server type \(instance type\)](#)

t2.micro

[Firewall \(security group\)](#)

New security group

[Storage \(volumes\)](#)

1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet

[Cancel](#) [Launch Instance](#)

Obrázek 18 Shrnutí instance EC2 před spuštěním [vlastní]

Po spuštění nalezneme instanci v přehledu instancí, odkud můžeme skrze detaily spravovat jednotlivá nastavení.

Instances (1) <a href="#">Info</a>							
<input type="text" value="Search"/>							
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Instant Messa...	i-0c227ec90c4a5c9d6	Running	t2.micro	2/2 checks passed	No alarms	eu-central-1b

Obrázek 19 Přehled instancí EC2 [vlastní]

### 11.1.1.2 Nasazení IM Serveru na EC2 a připojení se k němu

Prvním krokem je připojení se k instanci EC2. Z přehledu instancí vybereme tu, ke které se chceme připojit. V tomto případě se jedná o instanci Instant Messaging Server s Instance ID i-0c227ec90c4a5c9d6.

AWS nabízí 4 možnosti, jak se k dané instanci připojit – EC2 Instance Connect, Session Manager, SSH client a EC2 Serial Console. Pro účely našeho zadání jsou ideální volbou EC2 Instance Connect a SSH Client, díky kterým se vyhneme nadbytečnému nastavování dalších parametrů nepodstatných pro naše řešení.

Vybereme-li EC2 Instance Connect, musíme zvolit uživatele, pod kterého se chceme připojit. Předvyplněný uživatel je zpravidla správný. Potíže jsou nejčastěji způsobeny změnou uživatelského jména, kterou zkontrolujeme skrze AMI instrukce pro používání. Tento problém může nastat v případě, kdy se připojujeme k instanci, kterou nevlastníme. Pokud vše souhlasí, přejdeme k připojení skrze tlačítko „Connect“. Může se stát, že EC2 instance není nakonfigurovaná na EC2 Instance Connect, respektive, že na serveru není tato služba nainstalována. V takovém případě využijeme druhou a nejspolehlivější metodu připojení přes SSH a nastavíme Instance Connect na instanci.

Připojení přes SSH klienta nám ukáže návod, jak se přes SSH připojit. Jako první je potřeba otevřít SSH klienta, v našem případě to bude standardní terminál s nainstalovaným OpenSSH klientem. Nalezneme námi pojmenovaný Amazonem vygenerovaný klíč. V tomto případě hledáme „IMServerKey.pem“. Pokud je potřeba můžeme použít příkaz:

```
chmod 400 IMServerKey.pem
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                               WARNING: UNPROTECTED PRIVATE KEY FILE!                               @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'IMServerKey.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "IMServerKey.pem": bad permissions
ec2-user@ec2-3-73-144-44.eu-central-1.compute.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

Obrázek 20 Problém připojení klíčem kvůli špatným právům [vlastní]

Tento příkaz změní viditelnost klíče na neveřejný.

Nakonec se připojíme k instanci pomocí veřejného DNS, a to konkrétně k této:

```
ec2-3-73-144-44.eu-central-1.compute.amazonaws.com
```

Takže celý ssh příkaz bude následující

```
ssh -i "IMServerKey.pem" ec2-user@ec2-3-73-144-44.eu-central-1.compute.amazonaws.com
```

Před připojením na server na něj musíme nahrát náš naprogramovaný server. Existuje více způsobů, jak nahrát a kdy zkompilovat server, my jsme použili příkaz „scp“, který zkopíruje pomocí ssh klíče data na zvolený server.

Před spuštěním serveru na EC2 instanci je třeba doinstalovat potřebné závislosti. V našem případě se jedná o Javu. Použili jsme na to příkaz „sudo amazon-linux-extras install java-openjdk11“. Nyní už musíme jen zkompilovat a spustit server pomocí javac a java příkazů a máme funkční server.

```
[ec2-user@ip-172-31-34-89 ~]$ sudo java Server.java  
IM server listening on port 81, maximum nr. of connections=2...
```

Obrázek 21 Běžící Java server v EC2 [vlastní]

Veřejnou adresu serveru najdeme v detailu EC2 instance.

### 11.1.2 MPC v IoT

Nastavíme na AWS IoT Core naprogramované zařízení tak, aby naslouchalo na MQTT Brokeru. Následně nastavíme zařízení, aby bylo s AWS schopné komunikovat, a nakonec napíšeme obsluhy zpráv v našem zařízení.

#### 11.1.2.1 Připojení zařízení k AWS IoT

K připojení zařízení do IoT jsou zapotřebí tři kroky. Registrace zařízení, stáhnutí a nastavení aplikace connection kit a nastavení a otestování zařízení.

Do vyhledávání služeb zadáme IoT Core a zvolíme nalezenou službu. Přímo na hlavní stránce služby klikneme na tlačítko „Connect device“.

Jako první potřebujeme vybrat, z jaké platformy budeme zařízení připojovat a jaký programovací jazyk budeme používat. Podle toho nám bude nabídnuto odpovídající SDK (Software development kit). V našem případě se jedná o platformu Linux/OSX a programovací jazyk Java.

Dalším krokem je registrace zařízení. Je potřeba vyplnit jméno zařízení. V našem případě se bude jmenovat RemoteController.

Nyní si musíme stáhnout connection kit. Tento kit v tomto případě obsahuje certifikát (RemoteController.cert.pem) a privátní klíč zařízení (RemoteController.private.key), Java SDK a skript pro odesílání a přijímání zpráv „start.sh“.

Potom co jsme stáhli connection kit jej musíme rozbalit, přidat skriptu start.sh práva ke spuštění (pokud je nemá) a následně jej spustit. Zprávy z našeho zařízení se zobrazí níže.

### 11.1.2.2 Naprogramování zařízení

K propojení zařízení s AWS IoT Core je potřeba v zařízení napsat funkci na připojení zařízení k IoT. V ní použijeme certifikáty a klíče vygenerovány přímo z AWS z registrace zařízení.

```
// Nastavení přístupových certifikátů
// net – WifiClienSecure knihovna
// client – MQTTClient pro posílání a přijímání zpráv
net.setCACert(AWS_ROOT_CERT);
net.setCertificate(AWS_DEVICE_CERT);
net.setPrivateKey(AWS_DEVICE_PRIVAT_KEY);
client.onMessage(messageHandler); // Vytvoření handleru pro posílání zpráv
client.begin(AWS_IOT_ENDPOINT, 8883, net); // Připojení k MQTT brokeru
```

Obrázek 22 Připojení zařízení do IoT z kódu [vlastní]

Po nastavení přístupových dat je potřeba připojit zařízení k topiku na pub-sub. Na to se použije opět knihovna MQTTClient.

Message handler, který je znázorněn v Obrázku 8, je funkce, která zaštiťuje chování našeho zařízení při přijetí zprávy z IoT. V našem případě může ovládat vysílání zpráv z transmitteru.



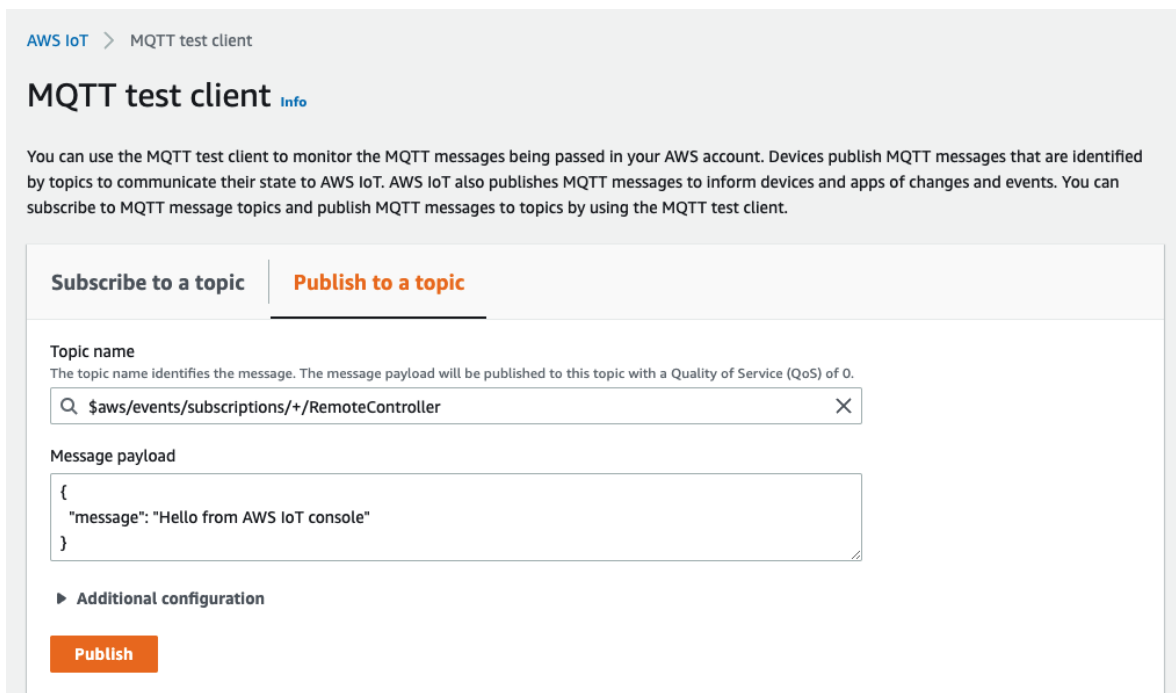
```
int cil = payload.cil;
const char* smer = payload.smer;

if (
    cil != CIL_ENUM.mujCil ||
    (smer != SMER_ENUM.ven || smer != SMER_ENUM.dovnitř)
)
{
    Serial.println("Nevalidni data");
    return;
}

switch (smer) {
    case SMER_ENUM.ven:
        mySwitch.send("011100011010110001010100");
        break;
    case SMER_ENUM.dovnitř:
        mySwitch.send("101000011100010110010100");
        break;
}
```

Obrázek 23 Ovládání zpráv příchozích z AWS IoT [vlastní]

Následně můžeme v AWS MQTT test client poslat do topiku, ke kterému jsme se s MPC zapsali, poslat zprávu a otestovat tak, jestli náš program funguje a jestli náš mikro počítač reaguje správně.



AWS IoT > MQTT test client

## MQTT test client [Info](#)

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

**Subscribe to a topic** | **Publish to a topic**

**Topic name**  
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

**Message payload**

```
{
  "message": "Hello from AWS IoT console"
}
```

▶ **Additional configuration**

**Publish**

Obrázek 24 Poslání zprávy do topiku MPC [vlastní]

Pro rozšíření lze napsat aplikaci, která bude do daného topiku v IoT posílat zprávy sama, například přes AWS Lambda.

## 11.2 GCP

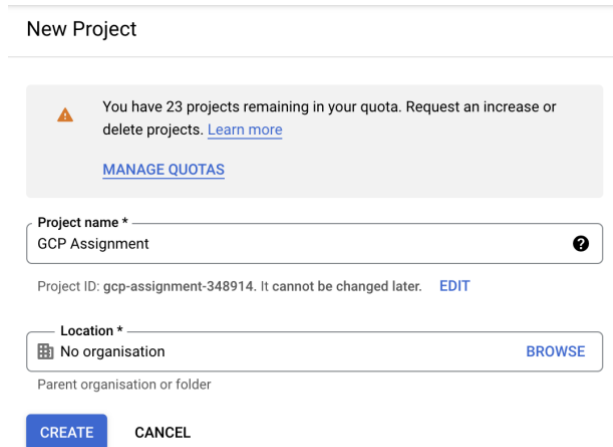
### 11.2.1 IM Server

Vytvoříme instanci virtuálního serveru e2-micro a následně se k němu připojíme, nahrajeme a spustíme námi napsaný server.

#### 11.2.1.1 Vytvoření instance e2-micro

U Google Cloud je potřeba předtím, než začneme vytvářet virtuální stroje, vytvořit si vlastní projekt, pod kterým povedeme všechny stroje v těchto zadáních. Potřebujeme pouze jméno projektu. V tomto případě jsme jej pojmenovali „GCP Assignment“.

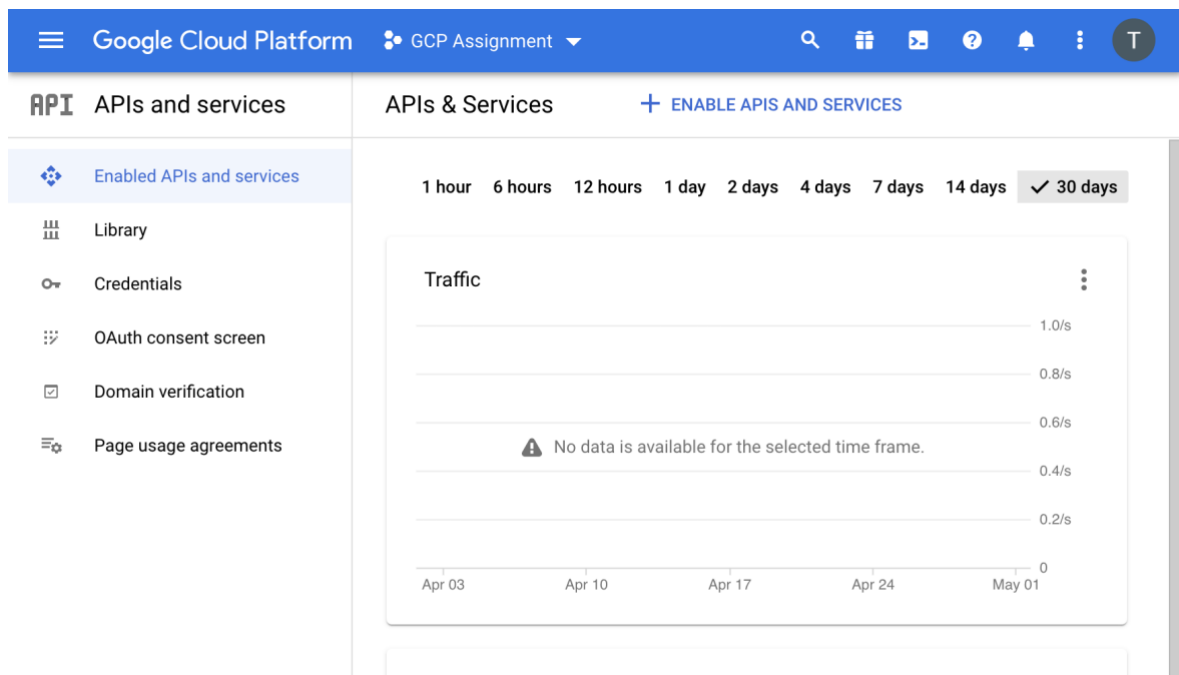
Stránka s vytvářením projektu nám ještě nabízí možnost zvolit organizaci, pod kterou se daný projekt vytváří. Aktuálně není nijak nezbytné toto vyplňovat, takže projekt klidně ponecháme bez organizace.



The screenshot shows the 'New Project' form in Google Cloud. At the top, it says 'New Project'. Below that is a warning message: 'You have 23 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)'. There is a 'MANAGE QUOTAS' link. The 'Project name \*' field contains 'GCP Assignment' and has a help icon. Below it, the 'Project ID' is 'gcp-assignment-348914. It cannot be changed later. [EDIT](#)'. The 'Location \*' field contains 'No organisation' and has a 'BROWSE' button. Below that, it says 'Parent organisation or folder'. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

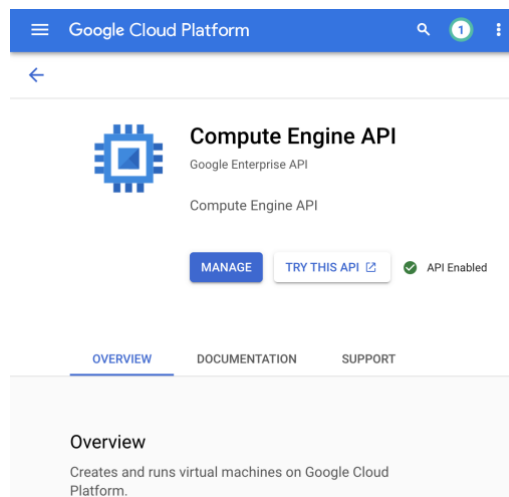
Obrázek 25 Vytváření nového projektu v GCP [vlastní]

Jako první věc se musí v nově založeném projektu zprovoznit Application Programming Interface (API) a další služby. To uděláme ve službě „APIs and services“ tlačítkem „ENABLE APIS AND SERVICES“.



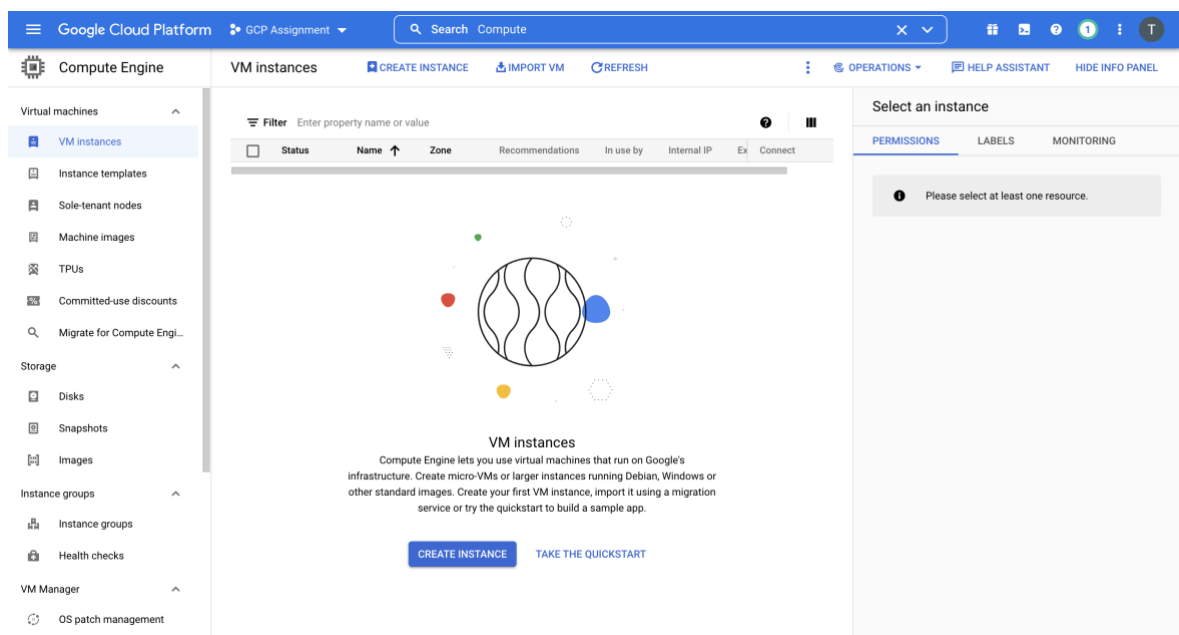
Obrázek 26 Aktivace API v GCP [vlastní]

To nás přeměruje na stránku se všemi možnými službami a API. Zde vyhledáme „Compute Engine API“ a tuto službu zprovozníme.



Obrázek 27 Compute Engine API po zprovoznění [vlastní]

Po jejím zprovoznění přejdeme na stránku Compute Engine do sekce VM Instances. Kde klikneme na „Create instance“ a zahájíme tak tvorbu virtuálního stroje.



Obrázek 28 Stránka s instancemi virtuálních strojů [vlastní]

Na rozdíl od AWS nám GCP rovnou neukazuje, které stroje spadají do Free tieru a které ne. Musíme tedy věnovat pozornost tomu, jaký stroj vytváříme. Vodítkem nám může být kalkulátor výdajů napravo od voleb VM.

GCP ve free tieru obsahuje:

- Instanci e2-micro a to pouze v regionech us-west1, us-central1 a us-east1. Je tedy potřeba počítat s větší odezvou, než kdybychom měli daný stroj v Evropě.
- 30 GB měsíčně standardního pevného disku.
- 5 GB stránkovací paměti v regionech Oregon: us-west1, Iowa: us-central1, South Carolina: us-east1, Taiwan: asia-east1, Belgium: europe-west1.
- 1 GB síťového toku ze Severní Ameriky do všech regionálních destinací (kromě Číny a Austrálie) měsíčně.

Pojmenujeme náš server například „im-server“ a vybereme postupně parametry serveru tak, abychom splňovali požadavky free tieru. Instanci e2-micro jsme vytvořili v regionu us-central1. Disk musíme zvolit „Standard persistent disk“ s kapacitou do 30 GB. Povolíme http traffic u firewallu serveru, abychom se následně mohli připojit k serveru přes http request.

Boot disk
✕

---

Select an image or snapshot to create a boot disk, or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

<
PUBLIC IMAGES
CUSTOM IMAGES
SNAPSHOTS
>

Operating system

Ubuntu
▼

Version \*

Ubuntu 18.04 LTS
▼

amd64 bionic image built on 2022-04-19, supports Shielded VM features

Boot disk type \*

Standard persistent disk
▼

Size (GB) \*

10

Obrázek 29 Výběr bootovacího disku GCP [vlastní]

Po vytvoření se k im-serveru můžeme připojit, a to přes SSH přímo z přehledu vytvořených instancí.

VM instances
CREATE INSTANCE
IMPORT VM
REFRESH
START/RESUME

INSTANCES
INSTANCE SCHEDULE

VM instances are highly configurable virtual machines for running workloads on Google infrastructure. [Learn more](#)

Filter Enter property name or value

<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	Connect
<input type="checkbox"/>	✓	im-server	us-central1-c			10.128.0.2 (nic0)	SSH

Obrázek 30 Vytvořená instance na GCP [vlastní]

### 11.2.1.2 Nasazení IM serveru a připojení se k němu

K připojení k serveru nám GCP nabízí tyto možnosti: připojení v okně, připojení v okně na specifickém portu, připojení v okně s poskytnutým soukromým klíčem, využití gcloud příkazů nebo použitím jiného SSH klienta. Pro náš případ bude naprosto dostačující připojení v okně, což nám otevře terminál v okně prohlížeče a připojí jej k běžícímu serveru.

Po připojení na server postupujeme podobně jak u příkladu z AWS. Jako první bude potřeba doinstalovat závislosti. V našem případě se jedná o Javu 11. Tu nainstalujeme sérií příkazů

- „sudo apt-get update“
- „sudo apt install openjdk-11-jre openjdk-11-jdk“.

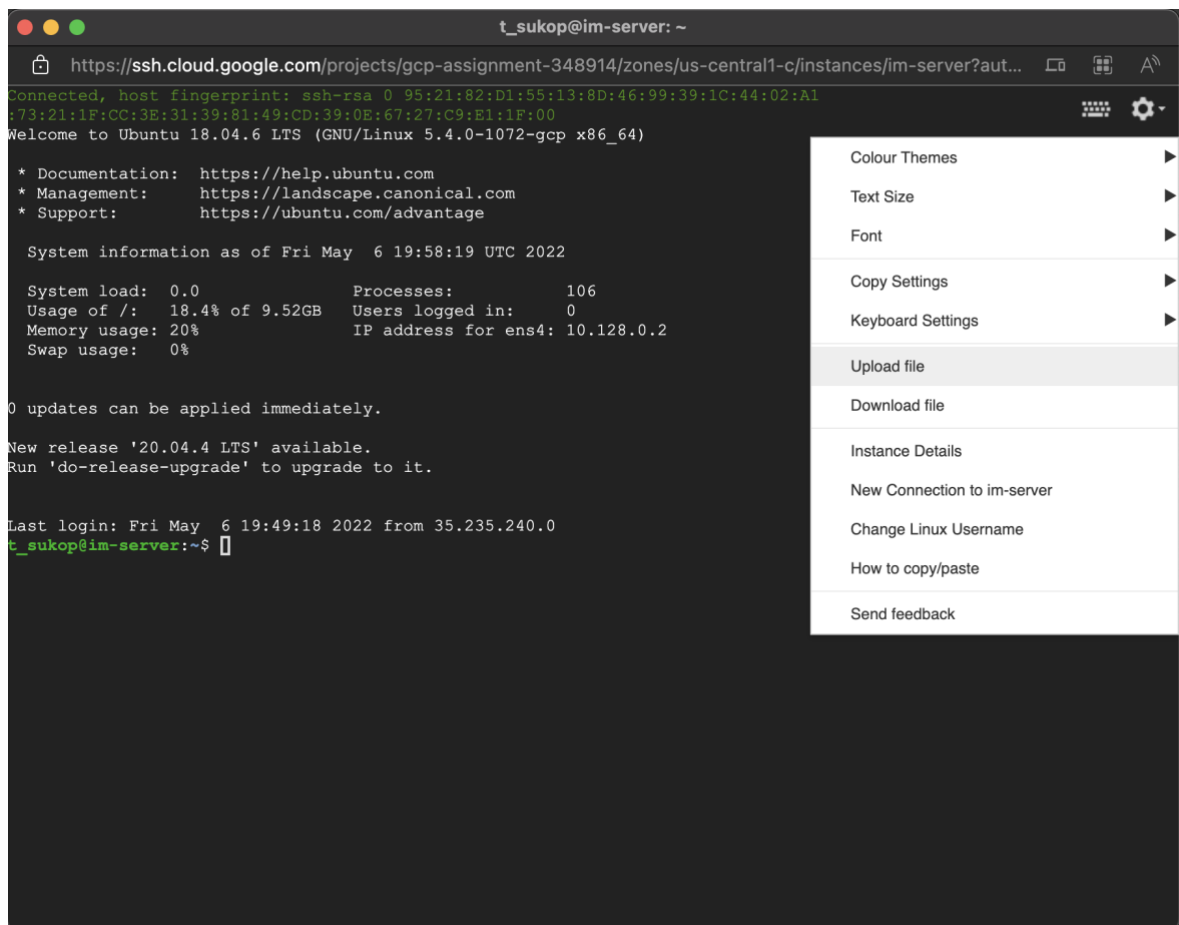
Můžeme poté zkontrolovat zda se správná verze nainstalovala pomocí „java –version“.

Dalším krokem je nahrání souboru serveru do instance. Z kontextové nabídky terminálu vybereme možnost „Upload file“ a vybereme náš serverový soubor s kódem. Po nahrání jej po vzoru příkladu z AWS pomocí příkazů javac a java zkompilujeme a spustíme.

```
t_sukop@im-server:~$ sudo java Server.java
IM server listening on port 81, maximum nr. of connections=2...
```

Obrázek 31 Java server běžící na GCP instanci [vlastní]

Přístupovou adresu, kterou lze poté volat, najdeme v síťových detailech instance. V tomto konkrétním případě se jedná o 34.123.114.215.



Obrázek 32 Nahrání souboru na GCP server [vlastní]

## 11.2.2 MPC v IoT

Ke splnění tohoto zadání použijeme Google Cloud IoT API.

### 11.2.2.1 Připojení zařízení

Jako první musíme aktivovat službu Google Cloud IoT API. To provedeme na stránce služby. Veškeré další nastavení pro připojení zařízení provedeme v konzoli GCP. Tu spustíme tlačítkem pro konzole vpravo nahoře v GUI GCP. Pokud vše provedeme správně, měla by se nám konzole zobrazit dole s informacemi o relaci a projektu.

Existují dva pro nás vhodné způsoby jak vytvořit a připojit zařízení do IoT Core. Prvním způsobem je vytvoření a připojení přes příkazovou řádku Cloudshellu. Druhou možností je přes průvodce v konzoli GCP. Pro náš případ bude vhodnější cestou druhá možnost.

Dalším krokem je tedy vytvořit nový záznam (registry). Zahájíme jeho tvorbu z přehledu IoT tlačítkem „Create registry“.

Musíme zadat identifikátor zařízení a s ním vytvoříme i pub/sub topic a Device state topic. Zbytek můžeme nechat defaultně tak, jak nám to nastavilo GCP.

The screenshot shows the 'Create a registry' page in the Google Cloud Platform IoT Core console. The page is titled 'IoT Core' and 'Create a registry'. It contains the following sections:

- Registry properties:**
  - Registry ID \***: A text input field containing 'remote-controller'. Below it, a note states: 'Permanent identifier for your registry. 3–255 characters. Start with a letter. You can also include numbers and the following characters: + . % - \_ ~'.
  - Region**: A dropdown menu set to 'us-central1'. Below it, a note states: 'Determines where data is stored for devices in this registry. Choice is permanent.'
- Cloud Pub/Sub topics:**
  - A note: 'Cloud IoT Core routes device messages to Cloud Pub/Sub for aggregation. You can route messages to different topics and subfolders in Cloud Pub/Sub based on the type of data in the messages. [Learn more](#)'.
  - Select a Cloud Pub/Sub topic**: A dropdown menu set to 'projects/gcp-assignment-348914/topics/controller'. Below it, a note states: 'Device telemetry events will be published to this topic by default.'
- Additional topics:**
  - A button labeled '+ ADD'.
- Device state topic (optional):**
  - A note: 'Device state data will be published to your selected topic on a best-effort basis, as well as to the default MQTT state topic (if your devices use MQTT protocol). [Learn more](#)'.
  - Select a Cloud Pub/Sub topic**: A dropdown menu set to 'projects/gcp-assignment-348914/topics/controller-mqtt'.

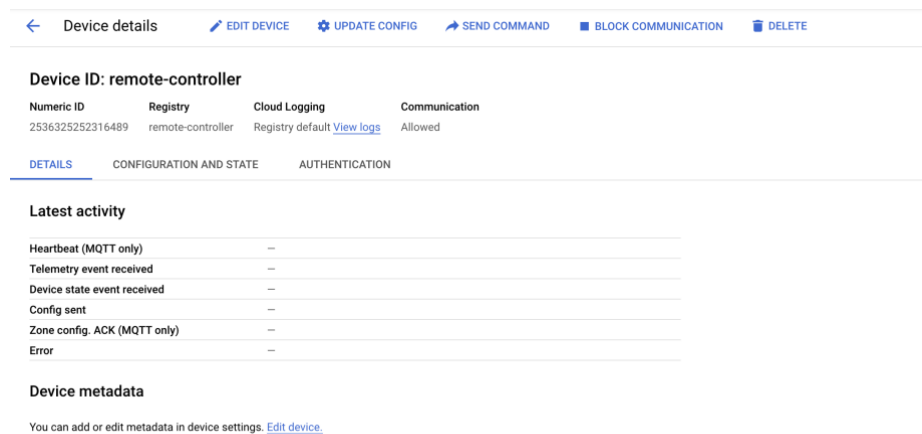
Obrázek 33 Vytvoření registru IoT GCP [vlastní]

Po úspěšném vytvoření registru můžeme v přehledu přidat naše zařízení. Do formuláře zadáme jméno, a aby naše zařízení mohlo komunikovat s GCP, je potřeba přidat veřejný klíč zařízení. Veřejný klíč musí být ve formátu PEM. Takový klíč lze vytvořit příkazy



- `openssl genpkey -algorithm RSA -out rsa_private.pem -pkeyopt rsa_keygen_bits:2048`
- `openssl rsa -in rsa_private.pem -pubout -out rsa_public.pem`

Pokud se nám vše podaří správně vygenerovat, vytvoříme zařízení v IoT. Nyní zbývá se s naprogramovaným zařízením připojit.

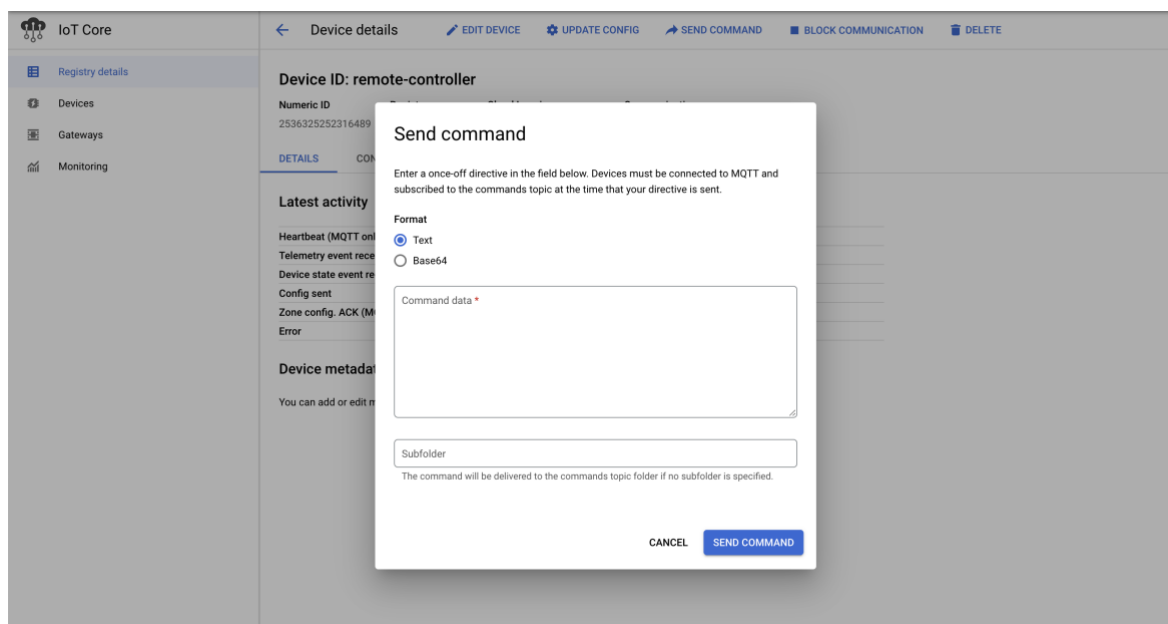


Obrázek 34 Vytvořené zařízení v IoT Core [vlastní]

### 11.2.2.2 Naprogramování zařízení

Podobně jako u AWS je potřeba nastavit přístupové body a klíče v zařízení, aby bylo schopné komunikovat a připojit se k MQTT brokeru vytvořeným GCP. Důležité informace, které je potřeba poskytnout, jsou ukázány v kódu níže. Jakmile se nám podaří připojit k zařízení na IoT Core, naprogramujeme reagování na zprávy příchozí ze serveru.

Následně můžeme k připojenému zařízení v GCP IoT Core poslat příkaz a otestovat tak, jestli náš program funguje a jestli náš mikro počítač reaguje správně.



Obrázek 35 Poslání příkazu do připojeného zařízení GCP [vlastní]

```
// Je potřeba se podepsat k tématu /devices/{device-id}/commands/# abychom
mohli
// dostávat všechny příkazy
client.subscribe(`/devices/${deviceId}/commands/#`, {qos: 0});

client.on('message', (topic, message) => {
  if (
    topic.startsWith(`/devices/${deviceId}/commands`) &&
    (message.target !== CIL_ENUM.mujCil ||
    (message.direction !== SMER_ENUM.ven || message.direction !==
SMER_ENUM.dovnitř))
  )
  {
    Serial.println("Nevalidni data");
    return;
  }

  switch (message.direction) {
    case SMER_ENUM.ven:
      mySwitch.send("011100011010110001010100");
      break;
    case SMER_ENUM.dovnitř:{
      mySwitch.send("101000011100010110010100");
      break;
    }
  }
});
```

Obrázek 36 Přihlášení k odběru zpráv z IoT GCP [vlastní]

```
// const deviceId = `remote-controller`;
// const registryId = `remote-controller`;
// const region = `us-central1`;
// const algorithm = `RS256`;
// const privateKeyFile = cesta k soukromému klíči (`./rsa_private.pem`);
// const serverCertFile = cesta k certifikátu serveru (`./roots.pem`);
// const mqttBridgeHostname = `mqtt.googleapis.com`;
// const mqttBridgePort = 8883;
// const messageType = `events`;

// The mqttClientId je unikátní řetězec, který identifikuje zařízení. Pro GCP IoT
// Core musí vypadat následovně
const mqttClientId =
`projects/${projectId}/locations/${region}/registries/${registryId}/devices/${deviceId}`;

// S GCP IoT Core se uživatelské jméno ignoruje, nicméně nesmí být prázdné. Pole
// s heslem je používáno k odesílání JWT autorizačního tokenu zařízení. Protokol
// „mqttts“ zaručí, že se knihovna připojí pomocí SSL – to je vyžadováno IoT Core
// With Google Cloud IoT Core, the username field is ignored, however it must be
const connectionArgs = {
  host: mqttBridgeHostname,
  port: mqttBridgePort,
  clientId: mqttClientId,
  username: 'unused',
  password: createJwt(projectId, privateKeyFile, algorithm),
  protocol: 'mqttts',
  secureProtocol: 'TLSv1_2_method',
  ca: [readFileSync(serverCertFile)],
};

// Vytvoříme a připojíme se ke klientovi
const client = mqtt.connect(connectionArgs);
```

Obrázek 37 Nastavení a připojení se ke klientovi GCP IoT [vlastní]

## 11.3 Azure

### 11.3.1 IM Server

Vytvoříme instanci VM, na které následně spustíme server, ke kterému se poté budeme moct připojit.

### 11.3.1.1 Vytvoření instance

K vytvoření instance použijeme službu Virtual Machines u MS Azure. Microsoft nabízí 750 hodin měsíčně instance typu B1 zdarma. Za výhodu můžeme považovat funkci, která při vytváření virtuálního stroje MS Azure ukazuje, které možnosti spadají do plánu zdarma.

Vzhledem k vytižení serverů MS Azure bylo nutno při vytváření zvolit server až ve východní Austrálii, z toho můžeme usoudit, že je o tyto servery opravdu velký zájem. Zdarma je pouze možnost používání instance typu B1.

[Home](#) >

## Create a virtual machine ...

**Basics**   Disks   Networking   Management   Advanced   Tags   Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ	<input type="text" value="Azure subscription 1"/>
Resource group * ⓘ	<input type="text" value="(New) bp-thesis"/> <a href="#">Create new</a>

### Instance details

Virtual machine name * ⓘ	<input type="text" value="im-server"/>
Region * ⓘ	<input type="text" value="(Asia Pacific) Australia East"/>
Availability options ⓘ	<input type="text" value="No infrastructure redundancy required"/>
Security type ⓘ	<input type="text" value="Standard"/>
Image * ⓘ	<input type="text" value="Ubuntu Server 20.04 LTS - Gen2 (free services eligible)"/> <a href="#">See all images</a>   <a href="#">Configure VM generation</a>
Azure Spot instance ⓘ	<input type="checkbox"/>
Size * ⓘ	<input type="text" value="Standard_B1s - 1 vcpu, 1 GiB memory (US\$9.64/month) (free services eligible)"/> <a href="#">See all sizes</a>

Obrázek 38 Vytváření instance MS Azure [vlastní]

Při vytváření Azure vygeneruje klíčový pár pro následující připojení k serveru, vybereme pouze uživatelské jméno.

**Administrator account**

Authentication type ⓘ  SSH public key  Password

**i** Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username \* ⓘ  ✓

SSH public key source  ✓

Key pair name \*  ✓

**Inbound port rules**

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \* ⓘ  None  Allow selected ports

Select inbound ports \*  ✓

**⚠ This will allow all IP addresses to access your virtual machine.** This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Obrázek 39 Nastavení SSH připojení do Azure [vlastní]

Následující nastavení ponecháme v základnímu módu. Při vytváření je podstatné stáhnout klíčový pár pro připojení.

Connect ▾ ▶ Start ↺ Restart □ Stop 📄 Capture 🗑 Delete ↻ Refresh 📱 Open in mobile 📄 CLI / PS 🗨 Feedback

^ Essentials

Resource group (move) :	<a href="#">bp-thesis</a>	Operating system :	Linux (ubuntu 20.04)
Status :	Running	Size :	Standard B1s (1 vcpu, 1 GiB memory)
Location :	Australia East	Public IP address :	<a href="#">20.211.127.173</a>
Subscription (move) :	<a href="#">Azure subscription 1</a>	Virtual network/subnet :	<a href="#">bp-thesis-vnet/default</a>
Subscription ID :	205c4697-46c7-4e8f-96ca-45d549655960	DNS name :	<a href="#">Not configured</a>
Tags (edit) :	<a href="#">Click here to add tags</a>		

Obrázek 40 Vytvořená B1 instance na Azure [vlastní]

### 11.3.1.2 Nasazení serveru a připojení se k němu

Nasazení serveru provedeme obdobně jako u AWS stroje. Veřejnou IP adresu najdeme v přehledu virtuálních strojů a poté pomocí ssh příkazů scp a ssh nakopírujeme na server náš naprogramovaný im-server.

Pokud jsme si nechali klíč vygenerovat službou Virtual Machine, bude potřeba stejně jako u AWS nastavit práva privátního klíče.

```
> chmod 400 im-server_key_azure.pem
> clear
> ssh -i im-server_key_azure.pem bp_sukop@20.211.127.173
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1022-azure x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Sat May  7 13:25:49 UTC 2022

System load:  0.08          Processes:           106
Usage of /:   4.9% of 28.90GB Users logged in:    0
Memory usage: 28%         IPv4 address for eth0: 10.0.0.4
Swap usage:   0%

* Super-optimized for small spaces – read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bp_sukop@im-server:~$
```

Obrázek 41 Nastavení práv klíče a připojení na server [vlastní]

Jakmile se nám podaří připojit na server pomocí ssh, musíme doinstalovat prerekvizity pro spuštění serveru. V našem případě jde o instalaci javy. Tyto kroky provedeme stejně jako u GCP.

```
bp_sukop@im-server:~$ java --version
openjdk 11.0.15 2022-04-19
OpenJDK Runtime Environment (build 11.0.15+10-Ubuntu-0ubuntu0.20.04.1)
OpenJDK 64-Bit Server VM (build 11.0.15+10-Ubuntu-0ubuntu0.20.04.1, mixed mode, sharing)
```

Obrázek 42 Nainstalovaná Java na Azure VM [vlastní]

Nyní již jen sestavíme a spustíme server příkazy „javac“ a „java“.

```
bp_sukop@im-server:~$ javac Server.java
bp_sukop@im-server:~$ sudo java Server.java
IM server listening on port 81, maximum nr. of connections=2...
```

Obrázek 43 Sestavení a spuštění java serveru na Azure [vlastní]

### 11.3.2 MPC v IoT

Pro správu zařízení disponuje MS Azure službou IoT Hub. Jako první tedy vytvoříme IoT Hub a poté do něj vytvoříme zařízení. IoT Hub nabízí jednu verzi zdarma s limity na velikost a počet zpráv.

#### 11.3.2.1 Připojení zařízení

Vytvoření IoT hubu probíhá v postupném vyplnění formulářů. Důležité je v záložce „Management“ zvolit „Scale tier“ jako F1 – Free tier. Tato verze zdarma má omezení na 8000 zpráv denně, které lze do všech připojených zařízení posílat. Abychom měli dostatečná práva, je potřeba zaškrtnout přidělení role „IoT Hub Data Contributor“.



Home > IoT Hub >

## IoT hub

Microsoft

✔ Validation passed.

Basics   Networking   Management   Tags   Review + create

### Basics

Subscription	Azure subscription 1
Resource group	bp-thesis
Region	West Europe
IoT hub name	remote-controllers
Disaster recovery enabled	Yes

### Networking

Connectivity configuration	Public access
Private endpoint connections	None
Allow public network access	Enabled

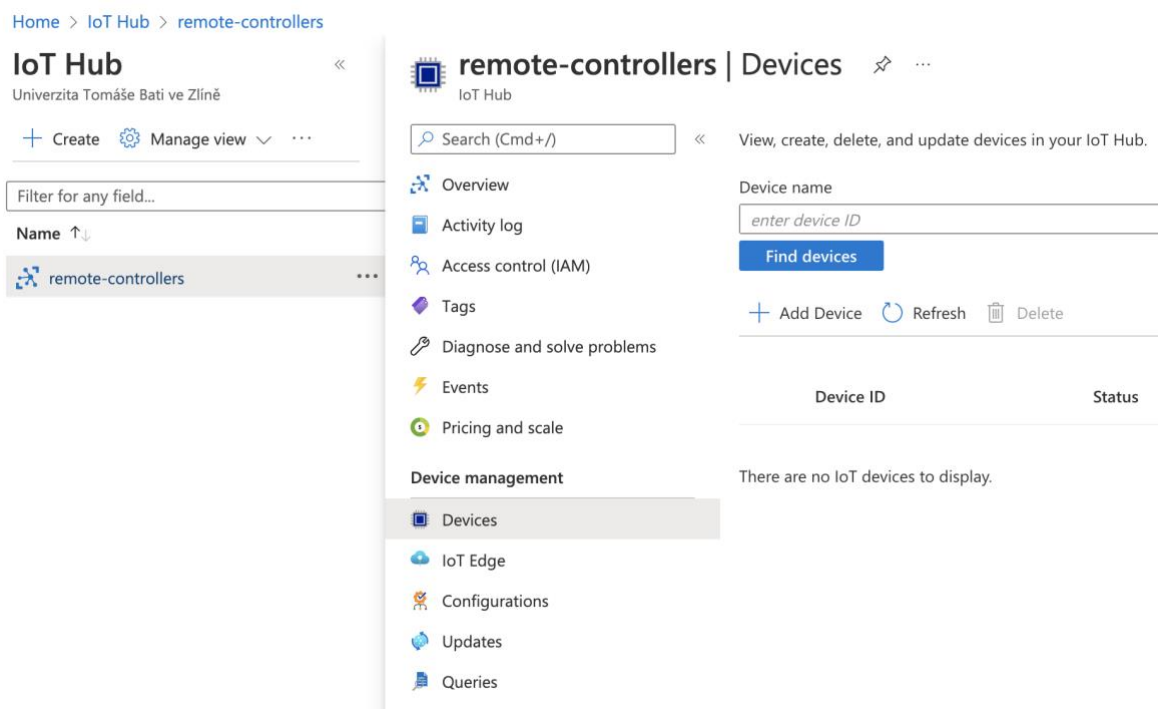
### Management

Pricing and scale tier	F1
Number of F1 IoT hub units	1
Messages per day	8,000
Device-to-cloud partitions	2
Cost per month	0.00 USD
Defender for IoT	Disabled
Preview mode	Off

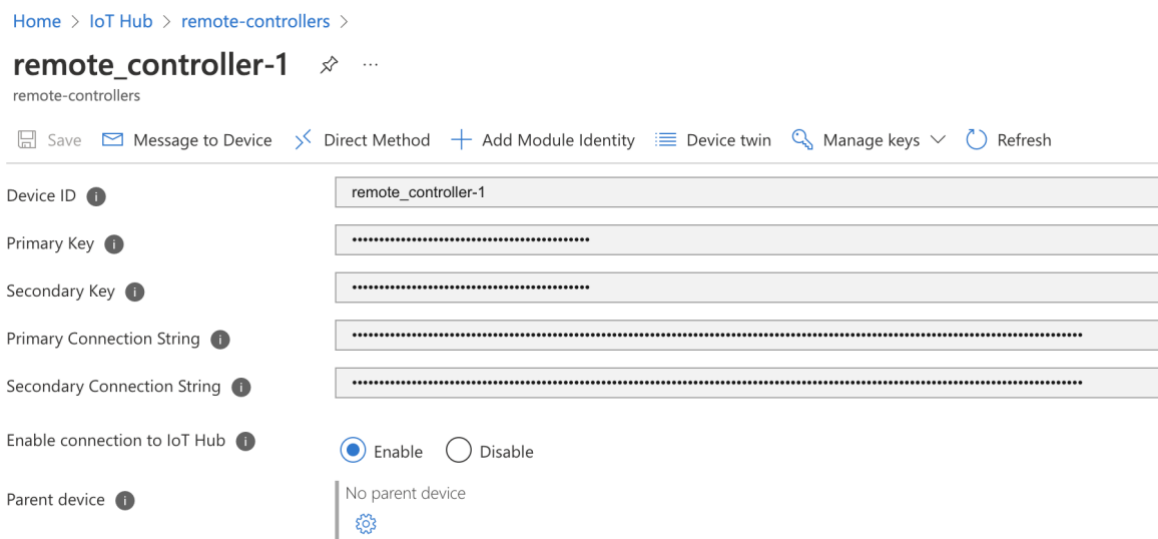
Tags

Obrázek 44 Vytvoření IoT hubu v Azure [vlastní]

Nyní ve správci zařízení přidáme nové zařízení. Pojmenujeme zařízení `remote_controller-1` a necháme automaticky vygenerovat symetrické klíče.



Obrázek 45 Správce IoT zařízení MS Azure [vlastní]



Obrázek 46 Vytvořené zařízení v IoT Hub Azure [vlastní]

### 11.3.2.2 Naprogramování zařízení

K připojení se k IoT Hub bude opět potřeba nastavit potřebné parametry v kódu, kterým se ovládá jak zařízení, tak všechny podřízené systémy. V případě MS Azure je potřeba takzvaný „connection string“ nebo řetězec znaků s definicí připojení. Ten se skládá

z HostName, což je adresa, na které je naše zařízení dostupné, DeviceId, název zařízení, ke kterému se chceme připojit a SharedAccessKey, což je klíč, který nám byl vygenerován při vytváření zařízení v IoT. Celý řetězec se zapisuje takto:

```
"HostName=<host_name>;DeviceId=<device_id>;SharedAccessKey=<device_key>"
```

Data pro připojovací řetězec najdeme v detailech zařízení v IoT Hub v MS Azure. Na obrázku 32 se jedná o „Primary Connection String“. Kód, který zde uvádíme, je minimalizován na nejnižší možnou úroveň tak, aby byl srozumitelný a byl zřejmý jeho účel. Pro připojení MQTT klienta a následnou komunikaci byla použita knihovna Esp32MQTTClient.h a WiFi.h.

```
static const char* connectionString = "HostName=remote-  
controllers.azure-devices.net;DeviceId=remote_controller-  
1;SharedAccessKey=Pk1x8tbK9AiSjd1Rl8xBrcbFQVIXHh46gvgKInom8+M=" ;  
  
static void MessageCallback(const char* payload, int size) {  
    int cil = payload.cil;  
    const char* smer = payload.smer;  
  
    if ( cil != CIL_ENUM.mujCil ||  
        (smer != SMER_ENUM.ven || smer != SMER_ENUM.dovnitř) ) {  
        Serial.println("Nevalidni data");  
        return;  
    }  
  
    switch (smer) {  
        case SMER_ENUM.ven:  
            mySwitch.send("011100011010110001010100");  
            break;  
        case SMER_ENUM.dovnitř:{  
            mySwitch.send("101000011100010110010100");  
            break;  
        }  
    }  
}  
  
void setup() {  
    Esp32MQTTClient_Init((const uint8_t*)connectionString, true)  
    Esp32MQTTClient_SetMessageCallback(MessageCallback);  
}
```

Obrázek 47 Kód pro připojení zařízení do IoT Hub [vlastní]

Nyní lze přímo z IoT Hub poslat zprávu pro připojené zařízení a dál vyvíjet a napojovat IoT Hub na další funkce cloudu, které povedou k automatizaci a širšímu využití zařízení.

## ZÁVĚR

Cílem práce bylo zjistit, jaké bezplatné cloudové služby dostupné na trhu by bylo možné využít ve výuce na Fakultě aplikované informatiky Univerzity Tomáše Bati, dále vytvořit zadání do cvičení ve vybraných předmětech a vypracovat vzorové implementace a nasazení na cloudové službě.

Za tímto účelem byly v teoretické části práce na základě rešerše vhodných na trhu dostupných bezplatných cloudových služeb popsány jejich vlastnosti a možné relevantní způsoby použití, přičemž jsou uvedeny i informace k tomu, jak lze bezplatné služby využívat i bez nutnosti uvést při registraci údaje platební karty. V další části práce pak jsou rozebrány vývojové nástroje, které studenti mohou využít pro usnadnění vývoje a ladění kódu běžícího v cloudu při řešení zadaných úloh. V praktické části práce navazuje vytvořením vzorových zadání a uvedením postupu vypracované vzorové implementace včetně nasazení v cloudu.

V práci je dále ukázáno, že vyvíjet kód, který bude následně nasazen na cloud, není po technické stránce vůbec problém, a programování cloud nativních aplikací je pro stále více programátorů velice populární a atraktivní. Existuje také řada zdrojů, ze kterých lze čerpat informace o tom, jak přesně cloud využívat. Pro účely této práce jsem zvolil dva příklady obecných vývojových prostředí, jedno placené (JetBrains), ovšem s licencí pro studenty, a druhé poskytované bezplatně (VS Code). V obou těchto prostředích lze bez problémů ladit a vystavovat programy na cloud. Na to navazuje použití SDK knihoven, přes které lze vytvářet a používat cloudové služby přímo z kódu. Neopomněl jsem ani zmínit a osvětlit ovládání cloudu přímo z webových aplikací samotných poskytovatelů.

Pro zadání do cvičení byly vybrány dva typové příklady. Jeden typ na programování programovatelné základní desky a její následné napojení do IoT, přičemž ve vzorových implementacích jsou popsány postupy, jak nastavit kód základní desky, aby se zvládla připojit do cloudu a zároveň jak vytvořit zařízení na cloudu, aby se přes zprávy mohl cloud s počítačem dorozumívat. Druhý typ je jednoduchý server pro přijímání a odesílání zpráv. Zde je opět ukázáno, jak daný server nasadit na virtuální instanci v cloudu a jak se na něj připojit z lokálního prostředí. Všechny vzorové implementace jsou vytvořeny a upraveny pro tři největší cloudové poskytovatele AWS, GCP a MS Azure.

Konstatuji, že daný cíl práce byl beze zbytku naplněn tím, že byly připraveny a otestovány úlohy IM server v Javě a MPC v IoT v cloudech poskytovatelů AWS, GCP a MS Azure do předmětů jako jsou „Programování síťových aplikací“ nebo „Programování mikropočítačů“.

Kromě naplnění stanoveného cíle může být tato práce přínosem i pro studenty oborů z oblasti aplikované informatiky z jiných vysokých škol – navržená zadání nastiňují způsob práce s cloudovými službami a mohou tak být motivací pro další samostudium a praktické vyzkoušení cloudových služeb. Povzbuzení studentů v těchto aktivitách je účelné pro jejich studium i pro budoucí uplatnění v praxi, neboť lze předpokládat, že v blízké budoucnosti se bude zájem po tomto způsobu řešení v oblasti infrastruktury a výpočetního výkonu zvyšovat. V neposlední řadě je stále aktuálnější realizace eGovernment cloudu, tedy řešení pro digitalizaci státní správy s využitím cloudových služeb.

**SEZNAM POUŽITÉ LITERATURY**

- [1] What is cloud computing? *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates., 2022 [cit. 2022-05-14]. Dostupné z: <https://aws.amazon.com/what-is-cloud-computing/>
- [2] Understanding cloud computing. *Red Hat* [online]. © 2022 Red Hat, 2022 [cit. 2022-05-15]. Dostupné z: <https://www.redhat.com/en/topics/cloud>
- [3] What is public cloud? *Red Hat* [online]. © 2022 Red Hat, 2022 [cit. 2022-05-15]. Dostupné z: <https://www.redhat.com/en/topics/cloud-computing/what-is-public-cloud>
- [4] What is private cloud? *Red Hat* [online]. © 2022 Red Hat, 2022 [cit. 2022-05-15]. Dostupné z: <https://www.redhat.com/en/topics/cloud-computing/what-is-private-cloud>
- [5] What is hybrid cloud? *Red Hat* [online]. © 2022 Red Hat, 2022 [cit. 2022-05-15]. Dostupné z: <https://www.redhat.com/en/topics/cloud-computing/what-is-hybrid-cloud>
- [6] What is multicloud? *Red Hat* [online]. © 2022 Red Hat, 2022 [cit. 2022-05-15]. Dostupné z: <https://www.redhat.com/en/topics/cloud-computing/what-is-multicloud>
- [7] Types of Cloud Computing. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/types-of-cloud-computing/>
- [8] What is Function-as-a-Service (FaaS)? *Red Hat* [online]. © 2022 Red Hat, 2022, 2022 [cit. 2022-05-15]. Dostupné z: <https://www.redhat.com/en/topics/cloud-native-apps/what-is-faas>
- [9] What are cloud service providers? *Red Hat* [online]. © 2022 Red Hat, 2022, 2022 [cit. 2022-05-15]. Dostupné z: <https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-providers>
- [10] About AWS. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/about-aws/>

- [11] Cloud computing with AWS. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/what-is-aws/>
- [12] Amazon EC2: Secure and resizable compute capacity for virtually any workload. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/ec2/>
- [13] Build Modern Applications with Free Databases on AWS. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/free/database/>
- [14] Amazon DynamoDB. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/dynamodb/>
- [15] Amazon RDS. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/rds/>
- [16] AWS Lambda. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/lambda/>
- [17] KNOX, Karlos. What is Google Cloud Platform (GCP)? *A CLOUD GURU - A PLURALSIGHT COMPANY* [online]. © 2022 Serverless Heroes, 2021 [cit. 2022-05-15]. Dostupné z: <https://acloudguru.com/blog/engineering/what-is-google-cloud-platform-gcp>
- [18] Compute Engine. *Google Cloud* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://cloud.google.com/compute/>
- [19] Google Cloud databases. *Google Cloud* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://cloud.google.com/products/databases>
- [20] Cloud SQL. *Google Cloud* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://cloud.google.com/sql>
- [21] Cloud Bigtable. *Google Cloud* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://cloud.google.com/bigtable>

- [22] Cloud Functions. *Google Cloud* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://cloud.google.com/functions>
- [23] Cloud Run. *Google Cloud* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://cloud.google.com/run>
- [24] KLINT, Lars. What is Azure? Microsoft's cloud platform explained. *A CLOUD GURU - A PLURALSIGHT COMPANY* [online]. © 2022 Serverless Heroes, 2021 [cit. 2022-05-15]. Dostupné z: <https://acloudguru.com/blog/engineering/what-is-microsoft-azure>
- [25] Virtual Machines. *Azure* [online]. © Microsoft 2022, 2022 [cit. 2022-05-16]. Dostupné z: <https://azure.microsoft.com/en-us/services/virtual-machines/>
- [26] Azure SQL. *Azure* [online]. © Microsoft 2022, 2022 [cit. 2022-05-16]. Dostupné z: <https://azure.microsoft.com/en-us/products/azure-sql/>
- [27] Azure Cosmos DB. *Azure* [online]. © Microsoft 2022, 2022 [cit. 2022-05-16]. Dostupné z: <https://azure.microsoft.com/en-us/services/cosmos-db/>
- [28] Azure Functions. *Azure* [online]. © Microsoft 2022, 2022 [cit. 2022-05-16]. Dostupné z: <https://azure.microsoft.com/en-us/services/functions/>
- [29] ROEHM, Brian. What is Oracle Cloud, and why should you use it? *A CLOUD GURU - A PLURALSIGHT COMPANY* [online]. © 2022 Serverless Heroes, 2021 [cit. 2022-05-15]. Dostupné z: <https://acloudguru.com/blog/engineering/what-is-oracle-cloud-and-why-should-you-use-it>
- [30] Virtual Machine series. *Azure* [online]. © Microsoft 2022, 2022 [cit. 2022-05-16]. Dostupné z: <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/series/>
- [31] TERRELL HANNA, Katie. Red Hat: Definition. *TechTarget* [online]. Copyright 2000 - 2022, TechTarget, 2021 [cit. 2022-05-16]. Dostupné z: <https://www.techtarget.com/searchdatacenter/definition/Red-Hat>
- [32] Red Hat Enterprise Linux on Amazon EC2 - FAQs. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/partners/redhat/faqs/>



- [33] Oracle Compute. *Oracle Cloud Infrastructure* [online]. © 2022 Oracle, 2022 [cit. 2022-05-16]. Dostupné z: <https://www.oracle.com/cloud/compute/>
- [34] Oracle Cloud Free Tier. *Oracle Cloud Infrastructure* [online]. © 2022 Oracle, 2022 [cit. 2022-05-16]. Dostupné z: <https://www.oracle.com/cloud/free/>
- [35] Oracle Exadata Database Service. *Oracle Cloud Infrastructure* [online]. © 2022 Oracle, 2022 [cit. 2022-05-16]. Dostupné z: <https://www.oracle.com/engineered-systems/exadata/database-service/>
- [36] Oracle Autonomous Database. *Oracle Cloud Infrastructure* [online]. © 2022 Oracle, 2022 [cit. 2022-05-16]. Dostupné z: <https://www.oracle.com/autonomous-database/>
- [37] Oracle Cloud Functions. *Oracle Cloud Infrastructure* [online]. © 2022 Oracle, 2022 [cit. 2022-05-16]. Dostupné z: <https://www.oracle.com/cloud/cloud-native/functions/>
- [38] Debug code. *JetBrains* [online]. © 2000–2022 JetBrains, 2022 [cit. 2022-05-16]. Dostupné z: <https://www.jetbrains.com/help/idea/debugging-code.html>
- [39] Debugging. *Visual Studio Code* [online]. © 2022 Microsoft [cit. 2022-05-16]. Dostupné z: <https://code.visualstudio.com/Docs/editor/debugging>
- [40] GONTOVNIKAS, Martin. What Is an SDK?. *Auth0* [online]. © 2013-2022 Auth0, 2020 [cit. 2022-05-16]. Dostupné z: <https://auth0.com/blog/what-is-an-sdk/>
- [41] AWS SDK for Java. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/sdk-for-java/>
- [42] ERICKSON, Karl a Jonathan GILES. Use the Azure SDK for Java. *Microsoft* [online]. © Microsoft 2022, 2021 [cit. 2022-05-16]. Dostupné z: <https://docs.microsoft.com/en-us/azure/developer/java/sdk/overview>
- [43] Easily access Google APIs from Java. *Google Developers* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://developers.google.com/api-client-library/java/>

- [44] Supported Google APIs. *GitHub* [online]. © 2022 GitHub [cit. 2022-05-16]. Dostupné z: <https://github.com/googleapis/google-api-java-client-services#supported-google-apis>
- [45] Getting Started with the AWS Management Console: GETTING STARTED GUIDE. *Amazon Web Services* [online]. © 2022, Amazon Web Services, Inc. or its affiliates, 2022 [cit. 2022-05-15]. Dostupné z: <https://aws.amazon.com/getting-started/hands-on/getting-started-with-aws-management-console/>
- [46] Cloud Console. *Google Cloud* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://cloud.google.com/cloud-console/>
- [47] Microsoft Azure portal. *Microsoft* [online]. © Microsoft 2022, 2021 [cit. 2022-05-16]. Dostupné z: <https://azure.microsoft.com/en-gb/features/azure-portal/>
- [48] Téma: Implementace serverových aplikací na cloudu  
Rozhovor s Ing. Michalem POSPÍŠILEM, CTO ve společnosti SnowlyCode, s.r.o. dne 10.5.2022 v Olomouci.
- [49] Solve real business challenges on Google Cloud. *Google Cloud* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://cloud.google.com/free>
- [50] Google Cloud Free Program. *Google Cloud* [online]. Google Ireland Limited, 2022 [cit. 2022-05-16]. Dostupné z: <https://cloud.google.com/free/docs/gcp-free-tier/#compute>
- [51] Explore free Azure services. *Microsoft* [online]. © Microsoft 2022, 2021 [cit. 2022-05-16]. Dostupné z: <https://azure.microsoft.com/en-in/pricing/free-services/>
- [52] Téma: Využití Active Directory v MS Azure  
Rozhovor s Michalem Růžičkou, senior developer ve společnosti SnowlyCode, s.r.o. dne 10.5.2022 v Olomouci.
- [53] What is Azure Active Directory? *Microsoft* [online]. © Microsoft 2022, 2021 [cit. 2022-05-16]. Dostupné z: <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis>

- [54] Build in the cloud free with Azure for Students. *Microsoft* [online]. © Microsoft 2022, 2021 [cit. 2022-05-16]. Dostupné z: <https://azure.microsoft.com/en-us/free/students/>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

ACID	Atomicity, Consistency, Isolation, Durability
AD	Active Directory
API	Application Programming Interface
AWS	Amazon Web Services
DB	databáze
DevOps	Development and Operations
DNS	Domain name system
FaaS	Function as a service
GCP	Google Cloud Platform
HPC	High performance computing
IaaS	Infrastructure as a service
IAM	Identity and Access Management
IDE	integrated development environment
IM	Instant messaging
IoT	Internet of things
IP	Internet protocol
ML	Machine learning
MPC	Micro programable computer
MQTT	Message queue telemetry transport
MS	Microsoft
OCI	Oracle loud
OLTP	online transactional processing
PaaS	Platform as a service
REST	Representational State Transfer
SaaS	Software as a service

SAP	System, applications and products
SDK	Software development kit
SLA	Service Level Agreement
UTB	Univerzita Tomáše Bati
vCPU	Virtual CPU
VM	Virtual Machine
VS	Visual Studio

**SEZNAM OBRÁZKŮ**

Obrázek 1	Ladící konfigurace v IntelliJ IDEA [38].....	34
Obrázek 2	Ladění serveru v IntelliJ IDEA [38].....	34
Obrázek 3	Soubor s konfigurací ladícího programu ve VS Code [39].....	35
Obrázek 4	Příklad použitých knihoven AWS Java SDK [41].....	36
Obrázek 5	Příklad nastavení virtuálního stroje pomocí Azure SDK [42] .....	37
Obrázek 6	Příklad autorizace pomocí Google API Client knihovny [43] .....	37
Obrázek 7	AWS Management Console [45] .....	38
Obrázek 8	Portál Microsoft Azure [47] .....	39
Obrázek 9	Google Cloud Console [46].....	40
Obrázek 10	Příklad vytvoření zásad v IAM AWS [48].....	42
Obrázek 11	Vytvoření uživatele v IAM AWS [48].....	43
Obrázek 12	Přiřazení zásady účtům v AWS IAM [48] .....	43
Obrázek 13	Vytváření role v GCP IAM [48] .....	45
Obrázek 14	Vytvoření uživatele v GCP IAM [48].....	46
Obrázek 15	Vytvoření klíčového páru pomocí AWS [vlastní].....	51
Obrázek 16	Nastavení sítě pro instanci EC2 [vlastní] .....	52
Obrázek 17	Nastavení úložiště [vlastní] .....	52
Obrázek 18	Shrnutí instance EC2 před spuštěním [vlastní] .....	53
Obrázek 19	Přehled instancí EC2 [vlastní].....	53
Obrázek 20	Problém připojení klíčem kvůli špatným právům [vlastní].....	54
Obrázek 21	Běžící Java server v EC2 [vlastní] .....	55
Obrázek 22	Připojení zařízení do IoT z kódu [vlastní].....	56
Obrázek 23	Ovládání zpráv příchozích z AWS IoT [vlastní].....	57
Obrázek 24	Poslání zprávy do topiku MPC [vlastní] .....	57
Obrázek 25	Vytváření nového projektu v GCP [vlastní].....	58
Obrázek 26	Aktivace API v GCP [vlastní].....	59
Obrázek 27	Compute Engine API po zprovoznění [vlastní] .....	59
Obrázek 28	Stránka s instancemi virtuálních strojů [vlastní] .....	60
Obrázek 29	Výběr bootovacího disku GCP [vlastní] .....	61
Obrázek 30	Vytvořená instance na GCP [vlastní] .....	61
Obrázek 31	Java server běžící na GCP instanci [vlastní] .....	62
Obrázek 32	Nahrání souboru na GCP server [vlastní] .....	63

Obrázek 33 Vytvoření registru IoT GCP [vlastní].....	64
Obrázek 34 Vytvořené zařízení v IoT Core [vlastní] .....	65
Obrázek 35 Poslání příkazu do připojeného zařízení GCP [vlastní] .....	66
Obrázek 36 Přihlášení k odběru zpráv z IoT GCP [vlastní] .....	67
Obrázek 37 Nastavení a připojení se ke klientovi GCP IoT [vlastní] .....	68
Obrázek 38 Vytváření instance MS Azure [vlastní] .....	69
Obrázek 39 Nastavení SSH připojení do Azure [vlastní] .....	70
Obrázek 40 Vytvořená B1 instance na Azure [vlastní] .....	70
Obrázek 41 Nastavení práv klíče a připojení na server [vlastní] .....	71
Obrázek 42 Nainstalovaná Java na Azure VM [vlastní].....	71
Obrázek 43 Sestavení a spuštění java serveru na Azure [vlastní] .....	72
Obrázek 44 Vytvoření IoT hubu v Azure [vlastní].....	73
Obrázek 45 Správce IoT zařízení MS Azure [vlastní].....	74
Obrázek 46 Vytvořené zařízení v IoT Hub Azure [vlastní].....	74
Obrázek 47 Kód pro připojení zařízení do IoT Hub [vlastní].....	75