

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Diplomová práce

Migrace webové aplikace do Microsoft Azure

Plzeň, 2017

Bc. Josef Hula

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných zdrojů.

V Plzni dne 12. května 2017

Bc. Josef Hula

Poděkování

Děkuji vedoucímu diplomové práce Ing. Tomáši Riegrovi za odborné vedení a cenné rady při vypracování diplomové práce.

Abstract

This diploma thesis is focused on migration of existing web application into Microsoft Azure environment. The first part describes services provided in Azure environment, that are going to be used and their possibilities for backup, monitoring and scalability. The theoretical part also includes information about application security and personal data protection connecting with using of the application.

The main target of this work is to do migration of the chosen part of the existing application and its database into the Microsoft Azure environment. At the beginning of the work there is an analytical phase describing existing solution. There is also designed and realized solution for deployment, backup and monitoring.

Abstrakt

Tato diplomová práce se zaměřuje na převod stávající webové aplikace do prostředí Microsoft Azure. První část popisuje služby v rámci prostředí Azure, které budou pro migraci využity a jejich možnosti pro zálohování, monitorování a škálování. Teoretická část také obsahuje informace o zabezpečení a ochraně osobních údajů související s používáním aplikace.

Hlavním úkolem této práce je provedení migrace vybrané části stávající aplikace a databáze do prostředí Microsoft Azure. Fázi migrace předchází analytická část, ve které je popsáno stávající řešení aplikace. Spolu s tím je navrženo a realizováno řešení pro deployment, zálohování a monitorování.

Obsah

1. Úvod.....	1
1.1. Motivace.....	1
1.2. Cíle práce	1
2. Microsoft Azure.....	2
2.1. Distribuční modely.....	2
2.1.1. Infrastruktura jako služba	3
2.1.2. Platforma jako služba.....	4
2.1.3. Software jako služba	4
2.2. Popis služeb a jejich účel	5
2.2.1. Virtuální počítač	5
2.2.2. Databázový server.....	6
2.2.3. App Service.....	7
2.2.4. Redis Cache	8
2.2.5. Automation	8
2.2.6. Storage	8
2.3. Cenová politika služeb	8
2.3.1. Virtuální počítač	9
2.3.2. Databázový server.....	9
2.3.3. App service	10
2.3.4. Přenos dat.....	10
2.3.5. Redis Cache	11
2.3.6. Automation	11
2.3.7. Storage	11
2.3.8. Možnosti podpory	11
2.4. Zálohování	12
2.4.1. Databáze.....	12

2.4.2.	Aplikace	13
2.5.	Monitorování.....	13
2.5.1.	Databáze.....	13
2.5.2.	Aplikace	15
2.6.	Zabezpečení a ochrana osobních údajů.....	16
2.6.1.	Zabezpečení	16
2.6.2.	Ochrana osobních údajů.....	16
2.6.3.	Ochrana osobních údajů z pohledu legislativy	17
2.6.4.	Citlivé osobní údaje	18
3.	Analýza migrace	18
3.1.	Škola OnLine	18
3.2.	Aplikační architektura.....	19
3.2.1.	Aplikace	19
3.2.2.	Databáze.....	19
3.3.	Stávající řešení	21
3.3.1.	Datové centrum.....	21
3.3.2.	Síťová architektura.....	21
3.3.3.	Zálohování dat	22
3.3.4.	Zabezpečení komunikace.....	22
3.3.5.	Monitorování služeb	23
3.4.	Požadavky a jejich řešení	23
3.4.1.	Spolehlivost	23
3.4.2.	Výkon.....	24
4.	Migrace aplikace.....	25
4.1.	Výběr služeb	25
4.1.1.	Databázová část	25
4.1.2.	Aplikační část	26

4.2. Databázová část.....	27
4.2.1. Testování kompatibility	28
4.2.2. Oprava nalezených problémů	29
4.2.3. Migrace databáze	32
4.2.4. Nahrazení úloh SQL serveru.....	33
4.3. Aplikační část.....	34
4.3.1. Publikování aplikace	34
4.3.2. Ukládání dat	35
4.3.3. Nastavení stavu relace	36
4.3.4. Využití vlastní domény	36
4.3.5. Podpora HTTPS	37
4.4. Možnosti pro deployment	37
4.4.1. Deploy z vývojového prostředí.....	38
4.4.2. Deploy s využitím FTP	38
4.4.3. Deploy pomocí synchronizace s cloudovou složkou.....	39
4.4.4. Nepřetržitý deployment z cloudového verzovacího nástroje.....	39
4.4.5. Nasazovací sloty	39
4.4.6. Výběr řešení pro deployment.....	39
4.5. Zálohování	40
4.5.1. Databáze.....	40
4.5.2. Úložiště	41
4.5.3. Aplikace	41
4.6. Monitorování.....	41
4.6.1. Databáze.....	42
4.6.2. Aplikace	45
4.6.3. Redis Cache	46
4.6.4. Úložiště	47

4.6.5. Řídicí panel	47
4.7. Škálování.....	47
4.7.1. Databáze.....	48
4.7.2. Aplikace	48
4.7.3. Redis Cache	49
4.8. Náklady na provoz	50
5. Závěr	51
Seznam obrázků.....	53
Přehled použitých zkratk a značení.....	54
Literatura.....	55
Příloha A: Návod pro migraci do Microsoft Azure	59
Ověření kompatibility databáze	59
Export databáze z SQL serveru	60
Migrace databáze do Azure	61
Migrace aplikace do Azure	64
Nastavení mezipaměti Redis Cache.....	70
Nastavení runbooků pro službu Azure Automation	72

1. Úvod

1.1. Motivace

Cloud computing je v současné době jedním z hlavních trendů v oboru informačních technologií. Společnost, která provozuje webovou aplikaci, zvažuje možnost migrace této aplikace do cloudu a chce zjistit, zdali je možné aplikaci do cloudu pouze přenést nebo zdali bude nutné vytvořit aplikaci určenou přímo pro cloud.

Hlavní motivací této práce je zjistit, jaké jsou možnosti pro migraci již stávající aplikace do prostředí Microsoft Azure a jestli je to pro společnost výhodné z hlediska nároků a složitosti převodu. Společnost funguje na produktech společnosti Microsoft a aplikace byla vyvíjena pomocí technologií společnosti Microsoft, jako cloudová platforma je proto zvolen Microsoft Azure.

1.2. Cíle práce

V teoretické části práce budou nejprve vymezeny základní pojmy z oblasti cloud computingu, které se vztahují k Microsoft Azure a bude provedeno bližší seznámení s možnostmi aplikačních služeb v rámci cloudové platformy Microsoft Azure. Následně bude provedena analýza stávající ASP.NET Web Forms aplikace, která využívá databázový server Microsoft SQL. Hlavním cílem této části je analyzovat aplikaci a databázi s ohledem na možnosti migrace do Microsoft Azure.

Praktická část práce se věnuje převedení vybraného segmentu již hotové aplikace a databáze do Microsoft Azure na základě analýzy provedené v předchozí části. V rámci této činnosti budou ověřeny nároky a složitost převodu. Cílem této části je provedení migrace aplikace do cloudového prostředí.

Následující část práce se bude zabývat návrhem řešení pro deployment, zálohování a monitorování aplikace provozované v rámci Microsoft Azure. Řešení bude následně realizováno dle návrhu.

V závěrečné části práce bude provedeno zhodnocení nároků pro migraci a budou zde popsány praktické dopady, které vyplynuly z převodu stávající aplikace do Microsoft Azure.

2. Microsoft Azure

Microsoft Azure [1] je cloudová platforma společnosti Microsoft, která se řadí do kategorie veřejného cloudu [2]. Jedná se o sadu integrovaných cloudových služeb, mezi které se řadí služby pro analýzu, výpočty, práci s databázemi a další. Nabízí podporu množství operačních systémů, programovacích jazyků či databází. Uživatel může využít řadu nástrojů pro tvorbu nových aplikací nebo provozovat již vytvořené aplikace. Cloudová platforma Azure byla představena na konci roku 2008 a uvedena na trh v roce 2010 pod původním jménem Windows Azure. K přejmenování na Microsoft Azure došlo v roce 2014 [3].

Microsoft v rámci platformy Azure poskytuje tři distribuční modely, kterými jsou infrastruktura jako služba (IaaS¹), platforma jako služba (PaaS²) a software jako služba (SaaS³).

Společnost Gartner zabývající se výzkumem a poradenstvím v oblasti informačních a komunikačních technologií ve svých žebříčcích Magic Quadrants řadí Microsoft mezi lídry pro služby veřejných cloudových úložišť, IaaS a PaaS [4].

Společně s Amazon Web Cloud patří Azure mezi nejrozšířenější veřejné cloudové platformy. Na základě průzkumu společnosti RightScale, v roce 2016 využívá 57 % respondentů Amazon Web Cloud, 17 % využívá Azure IaaS a 13 % Azure PaaS. Další cloudové platformy, které se objevily v průzkumu, ale měly podíl menší než 10 procent, jsou VMware vCloud air, Google Cloud Platform obsahující Google App Engine (PaaS) a Google Compute Engine (IaaS), IBM SoftLayer, DigitalOcean a Oracle Cloud (IaaS) [5]. V současné době jsou služby Azure dostupné ve 30 regionech po celém světě a Microsoft plánuje rozšíření o dalších 8 regionů. [6].

2.1. Distribuční modely

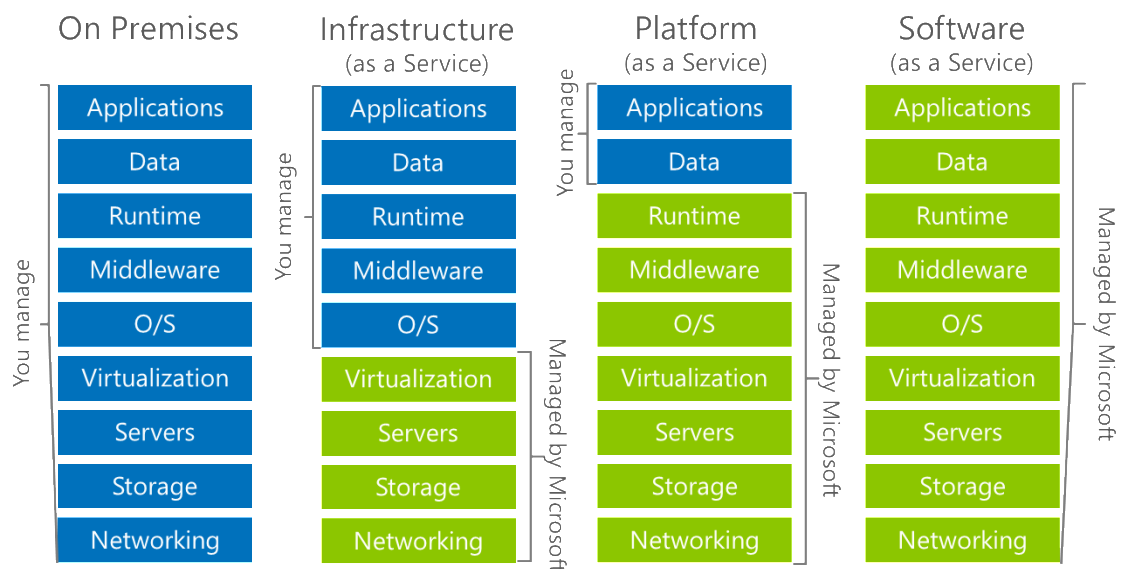
Pro lepší pochopení distribučních modelů nejprve zmíníme tradiční řešení on-premises [7], kdy společnost vlastní hardwarovou infrastrukturu a spravuje si hardware i software sama. Vše od samotného pořízení a instalace hardware až po nastavení datových úložišť, instalace aplikací a následné údržby je kompletně v režii společnosti.

¹ Infrastructure as a Service

² Platform as a Service

³ Software as a Service

Jak již bylo zmíněno dříve, Microsoft Azure poskytuje tři distribuční modely cloudových služeb, IaaS, PaaS a SaaS, které budou podrobněji popsány. V závislosti na zvoleném distribučním modelu je určitá část spravována poskytovatelem služby a část zákazníkem (viz obr. 2.1). Oproti klasickému webhostingu, přesněji řečeno pronájmu hardware a housingu, je hlavní rozdíl v tom, že v případě cloudových služeb dochází k účtování po menších časových úsecích (např. po hodinách či dokonce minutách, ne po měsících či letech) a přidělené prostředky je možné dynamicky měnit. Pokud například dojde k razantnímu nárůstu zátěže, který často bývá označován jako „slashdot effect“ [8], je možné reagovat a nabídnout vyšší výkon po dobu zátěže. U běžného webhostingu toto možné není a dochází k výpadku.



Obrázek 2.1: Distribuční modely [9]

2.1.1. Infrastruktura jako služba

Začneme popisem distribučního modelu IaaS [10]. Infrastruktura jako služba je model, ve kterém je správa infrastruktury v režii poskytovatele. V tomto modelu poskytovatel služby spravuje síťovou infrastrukturu, úložiště, servery a virtualizaci. Naopak zákazník má kompletně ve vlastní správě vše od operačního systému až po samotnou aplikaci (viz obr. 2.1). Výhodou tohoto modelu je, že zákazník je zbaven investičních nákladů, které jsou spojeny s nákupem a provozem vlastních serverů, úložišť a síťové infrastruktury. Tyto zdroje si zákazník pořizuje jako službu, o provoz HW infrastruktury a všechny problémy s tím spojené se v tomto modelu stará poskytovatel.

Tento model je výhodný pro firmy, které vlastní softwarové licence, ale nechtějí investovat do vlastního hardwaru. Využití IaaS dává také smysl v případech, kdy dochází ke značným výkyvům ve využití infrastruktury. Udržení spolehlivého provozu a dostupnosti v období nejvyšší zátěže je možné díky škálovatelnosti. Typickým příkladem použití může být například vývoj a testování, kdy je možné snadno vytvořit vývojová a testovací prostředí a ty je možné rychle škálovat.

Jako konkrétní případ může být uvedeno využití modelu IaaS společností GE Healthcare, jednou z divizí společnosti General Electric. Rozhodnutí k přesunu stávajícího on-premise řešení do Microsoft Azure bylo motivováno jednodušší a centralizovanou distribucí a správnou softwaru společně s lepší škálovatelností [11].

2.1.2. Platforma jako služba

Druhým distribučním modelem v rámci Azure je PaaS [12]. V tomto modelu se zákazník stará pouze o svojí aplikaci a data. Poskytovatel má ve správě veškeré věci týkající se operačních systémů, knihoven, bezpečnosti a rozšiřování. Toto řešení poskytuje podporu pro tvorbu škálovatelných a spolehlivých aplikací. Pro tvorbu aplikací nabízí Azure podporu populárních jazyků .NET, Java, Python, Node.js či PHP. Vytvořená aplikace bude následně spouštěna na virtuálním stroji, kde běží Windows Server. Tento VM se liší od modelu IaaS tím, že tento si Azure spravuje kompletně sám a například instalaci patchů operačního systému provádí automaticky.

Příkladem využití PaaS v praxi je případ společnosti Trader Corporation provozující webové stránky autoTRADER.ca. Hlavním důvodem přechodu bylo, aby se IT tým společnosti mohl plně soustředit na vývoj webových stránek a aplikací bez nutnosti se zabývat provozováním data centra. Díky tomuto kroku společnost očekává velké úspory, protože již nebude muset udržovat a škálovat tradiční infrastrukturu. [13]

2.1.3. Software jako služba

V modelu SaaS [14] je uživatelům umožněn přístup ke cloudovým aplikacím a mohou je používat přes internet. Od poskytovatele si zákazník za pravidelný poplatek pronajímá možnost využití aplikace a uživatelé se k aplikaci obvykle připojují pomocí webového prohlížeče. U tohoto modelu má vše ve správě poskytovatel, od infrastruktury až po software a data aplikace. Jako příklad této služby může být uveden Office 365.

Příkladem využití tohoto modelu je společnost British Airways zaměstnávající kolem 41000 zaměstnanců, která hledala vhodné nástroje pro zlepšení komunikace a spolupráce mezi zaměstnanci. Při výběru volba padla na Office 365 společně s Yammer Enterprise. Díky těmto nástrojům mezi sebou mohou zaměstnanci jednoduše a intuitivně komunikovat a spolupracovat, ať se nachází kdekoliv. [15]

2.2. Popis služeb a jejich účel

Microsoft Azure poskytuje nepřehledné množství služeb a nové služby stále přibývají. Je tedy zbytečné, aby byly v této práci všechny pospány. V této části budou zmíněny pouze služby Microsoft Azure, které se vztahují k této práci a mohlo by být zvažováno jejich využití pro přenášenou webovou aplikaci.

2.2.1. Virtuální počítač

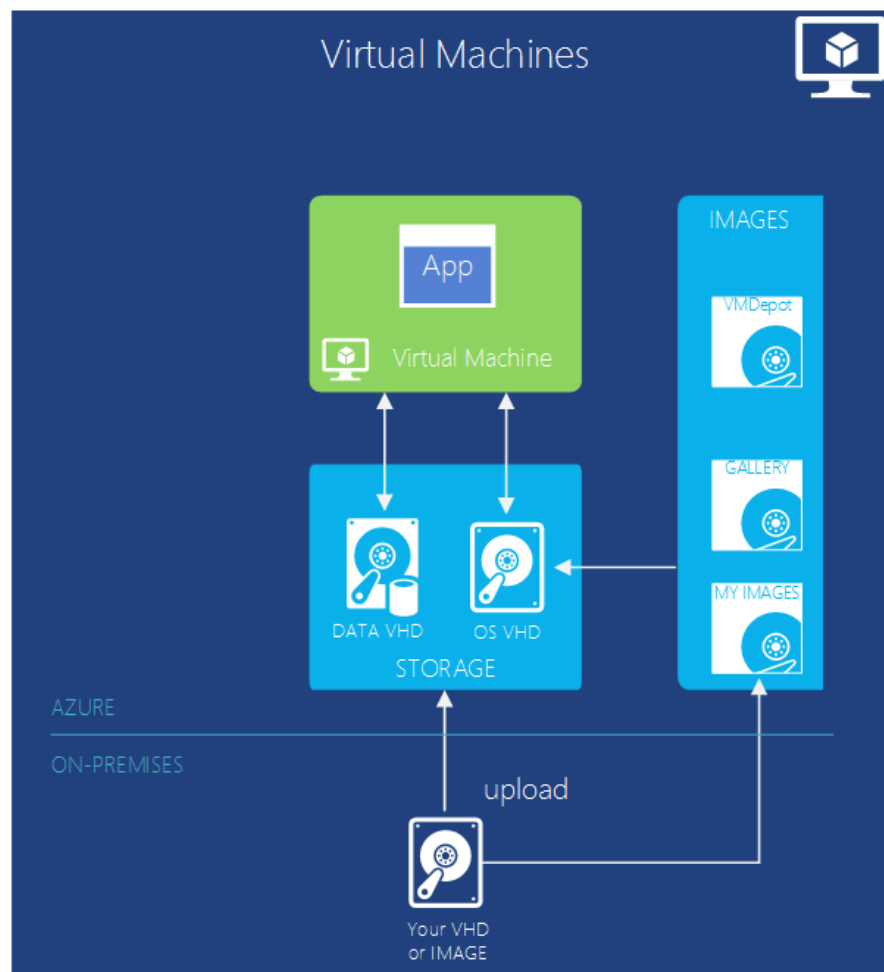
Virtuální počítače [16] neboli Virtual Machines (dále jen VM) je služba v rámci Azure, která reprezentuje cloudový model IaaS a umožňuje vytvořit škálovatelný virtuální počítač. Vytvořené virtuální počítače jsou provozovány v datových centrech společnosti Microsoft. VM nabízí několik výkonových konfigurací virtuálních serverů a jsou taktéž k dispozici výkonnější konfigurace obsahující SSD⁴ disky a rychlejší CPU⁵ hodící se pro náročnější výpočty. Jako operační systémy jsou přímo v Azure k dispozici obrazy disků s Windows (např. Windows Server 2012 R2, Windows Server 2016 Datacenter) nebo linuxovými operačními systémy (např. CentOS, Ubuntu, Red Hat). V nabídce je dále na výběr z obrazů disků vytvořených Microsoftem, jeho partnery či komunitou. Zmíníme například obrazy disků s předinstalovanými aplikačními servery, jakými jsou SQL server, BizTalk server, SharePoint server či Oracle WebLogic server. Virtuální počítače využívají virtuální pevné disky (VHD⁶) a tam ukládají svůj operační systém a data. Virtuální pevné disky se využívají i pro image, ze kterých je možné nainstalovat OS. V Azure je také možné vytvořit VM za použití vlastního image disku. V tomto případě stačí image disk ve formátu VHD nahrát do Azure a poté při vytváření VM vybrat z galerie disků ten, který uživatel nahrál (viz obr. 2.2). Správa VM je možná

⁴ Solid-State Drive

⁵ Central Processing Unit

⁶ Virtual Hard Disk

přes prohlížeč za pomoci Azure portálu, nástrojů příkazové řádky s podporou skriptování nebo prostřednictvím REST⁷ API⁸.



Obrázek 2.2: Azure Virtual Machines [17]

2.2.2. Databázový server

V rámci Azure nabízí Microsoft podporu pro dva cloudové modely, a to IaaS a PaaS. Model PaaS v případě databází bývá také označován databáze jako služba (DBaaS⁹). Azure podporuje MySQL, NoSQL (DocumentDB) a SQL¹⁰ databáze. Pro tuto práci je podstatná SQL databáze. V případě Azure jsou dvě možnosti pro SQL databáze. První možnost je Azure SQL Database, což je nativní cloudová databáze a jedná se tedy o model PaaS. Nabízí kompatibilitu s většinou funkcí systému SQL Server. Druhou možností je SQL Server nainstalovaný a hostovaný v cloudu na virtuálních počítačích

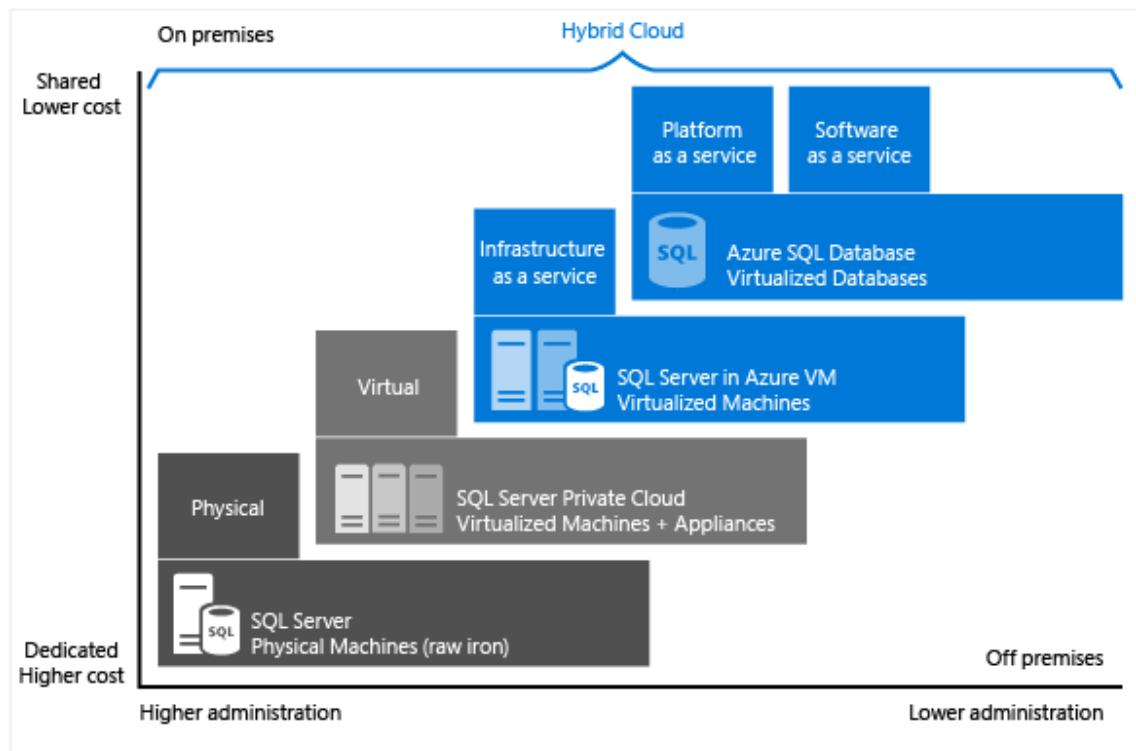
⁷ Representational State Transfer

⁸ Application Programming Interface

⁹ Database as a Service

¹⁰ Structured Query Language

s Windows Servery běžícími v Azure, což je řešení IaaS. SQL Server běžící na virtuálních počítačích je optimalizovaný pro migraci stávajících aplikací systému SQL Server. Databáze se v závislosti na modelu liší nutnými náklady a možnostmi administrace (viz obr 2.3).



Obrázek 2.3: Možnosti pro SQL databáze [18]

2.2.3. App Service

App Service je cloudová platforma, která reprezentuje model PaaS a sdružuje služby pro tvorbu webových aplikací (Web Apps), back-end částí mobilních aplikací (Mobile Apps), aplikací logiky (Logic Apps) a cloudových API rozhraní (API Apps). Blíže si popíšeme Azure Web Apps, která souvisí s touto prací. Jedná se o cloudový model PaaS, optimalizovaný pro hostování webových stránek či webových aplikací, kdy uživatel spravuje pouze svojí aplikaci. Azure poskytuje výpočetní zdroje, které mohou být na sdílených či dedikovaných VM v závislosti na předplacené konfiguraci. Web Apps je možné vytvářet v jakémkoliv jazyce, který je podporován Azure App Service, a to ASP.NET, Node.js, Java, PHP nebo Python.

2.2.4. Redis Cache

Služba Azure Redis Cache umožňuje používat zabezpečenou open source mezipaměť Redis. Pokud aplikace v rámci služby App Service používá stav relace, je doporučené nakonfigurovat externího poskytovatele stavu relace a k tomuto účelu je možné využít mezipaměť Redis Cache.

2.2.5. Automation

Služba Automation slouží ke zjednodušení správy cloudu pomocí automatizace procesů [19]. Uživatelům poskytuje možnosti, jak automatizovat dlouhotrvající a opakující se úlohy. Pro automatizaci procesu se definuje tzv. runbook. Runbook je vlastně sada úloh, které provádějí určité procesy. Jedná se o PowerShell skripty, které je možné naplánovat a pravidelně spouštět. Runbooky mohou být několika typů, a to Powershell, Grafický, PowerShell Workflow a nebo Grafický PowerShell Workflow. Automation umožňuje naplánovat velké množství druhů úloh pro Azure a je tak možné runbooky využít například ke správě SQL databáze v Azure. Runbooky lze vytvářet jako úplně nové PowerShell skripty nebo je na výběr z runbooků vytvořených Microsoftem a komunitou.

2.2.6. Storage

Jedná se o službu poskytující cloudové úložiště v Azure [20]. Toto úložiště je dále rozděleno na úložiště Blob¹¹, tabulek, souborů a front. Do Blob úložiště se ukládají nestrukturované datové objekty. Objektem blob je myšlen jakýkoliv druh textu nebo binárních dat, jako je dokument či soubor médií. Úložiště tabulek slouží k ukládání strukturovaných datových sad na bázi NoSQL, úložiště front poskytuje zasílání zpráv pro zpracování úloh a komunikaci mezi komponenty cloudových služeb a úložiště souborů nabízí sdílený prostor pro starší aplikace, které používají protokol SMB¹².

2.3. Cenová politika služeb

Financování cloudových řešení se oproti tradičním modelům liší a existují jiné způsoby pro optimalizaci nákladů. Hlavní specifika Microsoft Azure jsou ta, že zde nejsou žádné pořizovací náklady, žádné poplatky za ukončení a platíte jen za to, co používáte. Zásadní rozdíl oproti tradičním modelům je ten, že účtování za služby

¹¹ Binary Large Object

¹² Server Message Block

v rámci Azure probíhá po minutách. Nebudeme zmiňovat konkrétní ceny, ale spíše se zaměříme na to, jaká existují kritéria, od kterých se cena odvíjí. Ceny se liší dle konkrétních plánů a nyní si je tedy popíšeme.

2.3.1. Virtuální počítač

Jako první se seznámíme s cenami pro VM. Základní rozdělení je na plány Basic a Standard. V rámci každého plánu je poté možné zvolit instanci, která odpovídá konkrétním potřebám. Instance se liší typem pevného disku, počtem jader CPU, velikostí operační paměti a velikostí datového úložiště. Hlavními rozdíly mezi úrovněmi Basic a Standard jsou ty, že plán Basic poskytuje pouze část instancí, menší IOPS¹³, nepodporuje load balancing¹⁴ a automatické škálování. Cenu také ovlivňuje zvolený region, ve kterém se bude nacházet data centrum s umístěným VM. Součástí VM je vždy operační systém a licence, jejíž cena se promítne do celkové ceny. Je zde na výběr například z operačních systémů Windows nebo Linux. Stejným způsobem se řeší u Azure i aplikační servery jako součást služby VM. Pokud bude VM objednan včetně aplikačního serveru, například SQL serveru, bude součástí i licence k SQL serveru, jejíž cena se promítne do výsledné ceny. U některých typů aplikačních serverů je na výběr druh licence, která bude poskytnuta, lišící se cenou. Například v případě BizTalk serveru je na výběr mezi Standard a Enterprise licencí.

2.3.2. Databázový server

Azure SQL databáze dává na výběr plány Basic, Standard a Premium s několika úrovněmi výkonu. Je možné zvolit buď model izolované databáze, kde má databáze vyhrazené prostředky s vybranou úrovní výkonu, nebo model elastické databáze, kde jsou databáze v elastickém fondu. V elastickém fondu se prostředky sdílejí napříč databázemi.

Úrovně výkonu jsou primárně určeny dostupnými prostředky, které se pro izolované databáze uvádějí v jednotkách databázových transakcí DTU¹⁵ a pro elastické fondy se uvádějí v elastických jednotkách databázových transakcí eDTU¹⁶ [21]. Úrovně se liší dalšími parametry, z nichž zmíníme například maximální velikost databáze pro model

¹³ Input/Output Operations per Second

¹⁴ vyvažování zátěže

¹⁵ Database Transaction Unit

¹⁶ elastic Database Transaction Unit

izolovaných databází, či maximální počet databází pro každý fond v modelu elastických databází. Také u SQL databáze má vliv na cenu zvolený region [22]. U SQL databáze se účtuje za přenos odchozích dat, což bude blíže popsáno v sekci Přenos dat.

2.3.3. App service

V Azure App Service je na výběr z plánů Free a Shared¹⁷, které umožňují hostovat aplikace ve sdíleném prostředí a dále plány Basic, Standard a Premium, které poskytují virtuální počítače vyhrazené pro určený plán. Plány Free a Shared jsou určeny zejména pro experimentování s platformou, vývoj a testování. U těchto plánů je cena udávána za nasazenou aplikaci oproti plánům Basic, Standard a Premium, kde se cena odvíjí od velikosti a počtu instancí virtuálních počítačů, které jsou provozovány. V plánech Basic, Standard a Premium jsou na výběr různé úrovně výkonu, které se liší počtem jader procesoru, pamětí RAM a velikostí úložiště. Plány se dále liší integrovanou podporou dalších možností, jako jsou vyrovnávání zatížení v síti, automatické škálování nebo zálohování. K plánům je dále možné dokoupit vlastní doménu, certifikáty Azure App Service a připojení SSL¹⁸. Je nutné brát v potaz, že i zastavené aplikace jsou zpoplatněné. Cena plánu se také liší podle oblasti, v jaké je služba provozována. Mimo ceny za každý ze zmíněných plánů se platí navíc za každou přidanou instanci a také za data přenesená ven z Azure, což bude popsáno v následující sekci.

2.3.4. Přenos dat

Služba přenos dat představuje data přenesená do datacenter Azure nebo z nich. Pokud se jedná o přenos příchozích dat, tedy když data přicházejí do datových center Azure, je služba zdarma. Pro přenosy dat odcházejících z datacenter Azure je cena určena množstvím přenesených dat a zdrojovou oblastí, odkud data z Azure odcházejí. Pokud nedochází k nějaké extrémnímu využití, není cena za přenos výrazná. Například v hladině 5 GB – 10 TB v oblasti Západní Evropa je cena 0,0734 euro za GB. Prvních 5 GB přenosu dat za měsíc je v každé oblasti zdarma.

Mezi službami Azure, které se nachází ve stejné oblasti, se neplatí žádné další náklady na přenos dat. Naopak za přenos dat mezi službami Azure, které jsou umístěné v různých oblastech, se účtují normální sazby za odchozí data a příchozí data se neplatí.

¹⁷ ve verzi Preview

¹⁸ Secure Sockets Layer

2.3.5. Redis Cache

Azure Redis Cache nabízí výběr z plánů Basic, Standard a Premium. Každý z těchto plánů nabízí různé úrovně, které se liší velikostí mezipaměti, výkonem sítě a počtem připojených klientů. Plán Basic je vhodný zejména pro experimenty a testování. Pro tento plán není poskytována smlouva SLA¹⁹. U plánu Standard se jedná o replikovanou mezipaměť v konfiguraci primárního a sekundárního uzlu. Mezi těmito dvěma uzly je zajištěna automatická replikace. Plán Premium obsahuje veškeré funkce z plánu Standard a navíc přidává například zotavení po havárii a zvýšené zabezpečení.

2.3.6. Automation

Automatizace procesů je dostupná v úrovních Free a Basic. Účtování probíhá na základě počtu minut běhu úloh v měsíci. V úrovni Free je omezení, které udává, že doba běhu všech úloh v měsíci může být maximálně 500 minut. Úroveň Basic je bez omezení, avšak zpoplatněna je každá započatá minuta běhu úlohy.

2.3.7. Storage

Ceny u služby Storage se liší podle druhu použitého úložiště a závisí na množství uložených dat, na objemu a typu transakcí a na zvolené možnosti redundance dat. U úložiště Blobů se ceny liší v závislosti na vrstvě přístupu. Horká úroveň přístupu pro objekty, ke kterým se bude přistupovat častěji, nabízí ukládání dat s nižšími přístupovými náklady oproti studené úrovni, ve které se k datům nebude přistupovat tak často, a tak nabízí ukládání dat s nižšími náklady na úložiště dat.

2.3.8. Možnosti podpory

Azure nabízí flexibilní možnosti podpory pro zákazníky různých velikostí rozdělené do plánů Free, Developer, Standard, Professional Direct a Premier. Plány podpory se liší cenou a obsaženými funkcemi. [23] Plán Free nabízí podporu pro fakturaci a předplatné a samoobslužnou podporu. U plánu Developer, který je vhodný pro neprodukční prostředí, je nabízena technická podpora pouze v pracovní době a doba první odezvy je menší než 8 hodin. Maximální závažnost požadavku pro tento plán je C, což znamená minimální vliv na chod firmy [24]. Plán Standard nabízí neomezenou nepřetržitou technickou podporu s dobou první odezvy menší než 2 hodiny. Maximální

¹⁹ Service Level Agreement

závažnost požadavku může být A, znamenající závažný dopad na chod firmy [24]. V plánu Professional Direct je nabízena služba omezeného poradenství a správa eskalací v rámci fondu. Doba první odezvy je v tomto plánu menší než 1 hodina a maximální závažnost také může být A. Nejvyšší možnost úrovně podpory je plán Premier, který nabízí specifické služby poradenství pro zákazníka a má přiřazenou správu eskalací. Maximální doba první odezvy je menší než 15 min (s Azure Rapid Response) nebo menší než 1 hodina (bez Azure Rapid Response).

2.4. Zálohování

Microsoft Azure poskytuje několik možností pro zálohování. K zálohování lze využít službu Azure Backup, která nabízí zálohování a obnovu dat v cloudu. Tato služba obsahuje různé komponenty, které se liší tím, jaká data umožňují zálohovat. Azure Backup umožňuje zálohovat například soubory, složky, aplikace nebo virtuální počítače. V případě této práce nás ale bude spíše zajímat možnost zálohování v rámci jednotlivých služeb, jako jsou Azure SQL databáze a App service.

2.4.1. Databáze

Databáze v cenových úrovních Basic, Standard a Premium jsou zálohovány automaticky v rámci služby Azure SQL databáze bez nutnosti aktivace a dodatečných nákladů. Tyto zálohy jsou uchovány po dobu 7 dní pro úroveň Basic a 35 dní pro úroveň Standard a Premium. Záloha kompletní databáze je prováděna jednou týdně, záloha přírůstku databáze každou hodinu a záloha transakční logu každých 5 minut. Pro obnovu zálohy databáze je možné provést operaci Point-in-time, která umožňuje obnovit databázi do libovolného času v závislosti na úrovni služby. Služba sama zjistí, zdali má být obnovena záloha celé databáze, přírůstku nebo transakčního logu.

Další možností pro zálohování databáze je geografická replikace, která se dělí na standardní a aktivní. Standardní geografická replikace vytváří offline sekundární databázi v předem spárované oblasti Azure, které se sice nachází ve stejné geografické oblasti, ale je vzdálená alespoň 800 km. Oproti tomu aktivní geografická replikace vytváří až 4 online sekundární databáze, které se mohou nacházet v libovolné oblasti Azure. Tyto databáze jsou synchronizovány s primární databází pomocí asynchronního replikačního mechanismu.

Pokud chce uživatel uchovat zálohu starší, než je maximální doba v rámci úrovně služby, může zvážit možnost pravidelného archivování databáze do BACPAC souboru. BACPAC soubor je komprimovaný soubor obsahující databázové schéma a s ním spojená data. Tento soubor je možné uchovávat například v Azure Blob Storage nebo jiném libovolném umístění.

Pro dlouhodobé zálohování databáze se ještě nabízí možnost Long-term retention, umožňující uchovat zálohu SQL databáze v Azure Recovery Services až po dobu deseti let.

2.4.2. Aplikace

Ve službě App Service nabízí možnost pro vytváření zálohy nastavení aplikace, souborů a také připojené databáze [25]. Podpora pro zálohování je u této služby od úrovně Standard výše. Rozdíl mezi úrovní Standard a Premium je v množství záloh prováděných denně. Každá záloha je kompletní offline kopie aplikace, nejedná se o inkrementální zálohování. Ze zálohy je možné obnovit aplikaci do předchozího stavu nebo například vytvořit novou aplikaci.

Služba App service dále nabízí možnost pro vytváření částečných záloh. To se hodí v případě, že aplikace obsahuje data, která se nikdy nemění a není nutné je tedy zálohovat. Pro vyřazení souborů či složek ze zálohy je nutné vytvořit soubor `_backup.filter` ve složce aplikace `wwwroot` a v něm specifikovat, které souboru a složky budou vyřazeny ze zálohy.

2.5. Monitorování

2.5.1. Databáze

Pro monitorování databáze je zobrazen na stránce přehledu služby Azure SQL Database monitorovací graf. V tomto grafu lze nadefinovat metriky, které se mají zobrazovat a časový rozsah monitorování. Na výběr je z patnácti metrik, z nichž zmíníme například metriku zobrazující využití CPU či velikost databáze.

Pro všechny monitorované metriky umožňuje Azure přidání pravidel výstrah. Pokud pak sledovaná metrika splní definovanou podmínku, dojde k rozeslání výstrahy na e-mailové adresy správců.

Výstrahy kromě zasílání e-mailu jako další možnost upozornění nabízí funkci Webhooks. Webhook je akce, při které Azure odešle POST request se zprávou ve formátu JSON²⁰ na zvolenou URL adresu. Upozornění lze například předat službě, která odešle SMS zprávu správci.

Diagnostické informace můžeme archivovat v účtu úložiště po libovolně dlouhé časové období, sdílet do centra událostí nebo zasílat do služby Log Analytics, která pomáhá shromažďovat a analyzovat data vygenerovaná prostředky [26].

Pro sledování veškerých změn, které se dějí v databázi, nabízí Azure funkci Auditing. Tato funkce databázové události ukládá v účtu úložiště, k jehož prohlížení lze využít desktopovou aplikaci Azure Storage Explorer. V této aplikaci je možné auditové logy analyzovat, dále zpracovávat či exportovat ve formátu CSV²¹.

V Azure SQL databázi neustále běží poradce pro databázi [27], který ji sleduje a s využitím strojového učení sleduje databázové dotazy, jejich výkonost a nabízí doporučení pro zvýšení výkonu. Poradce navrhuje vytváření či rušení indexů, parametrizování dotazů a opravy pro problémy schémat. Vytváření a rušení indexů je možné automatizovat. Doporučení na vytvoření indexů se objevuje, když poradce detekuje chybějící index, který by mohl snížit pracovní zátěž. Pokud se v databázi vyskytnou duplicitní indexy, je poradcem doporučeno jejich smazání.

Pokud existují dotazy, které jsou neustále kompilovány, ale vždy končí se stejným exekučním plánem dotazu, poradce nabízí doporučení pro parametrizaci dotazů. Každý dotaz musí být nejdříve zkompilován do exekučního plánu, který je ukládán do cache a stejné dotazy ho tak mohou znovu použít. Pokud však existují dotazy, které nevyužívají parametrizaci, tak pro každý dotaz s rozdílnými parametry je zkompilován samostatný exekuční plán. V mnoha případech dotazy s rozdílnými parametry mají shodný exekuční plán. Rekompilování exekučních plánů využívá databázové zdroje a zvyšuje dobu trvání dotazů. To je možné vyřešit použitím doporučení pro nucenou parametrizaci dotazů. Pokud je doporučení aplikováno, tak začne monitorovací proces databáze, který trvá přibližně 24 hodin. Po této době je vytvořen report obsahující porovnání využití CPU při aplikování parametrizace a bez ní. Pokud doporučení nevedlo ke zvýšení výkonu, je možné vrátit změny.

²⁰ JavaScript Object Notation

²¹ Comma-Separated Values

Posledním doporučením je oprava problémů ve schématu databáze. Toto doporučení se objevuje v situaci, když se definice SQL dotazu neshoduje s definicí v databázovém schématu. Například když sloupec, který je dotazovaný v cílové tabulce, neexistuje.

Dalším monitorovacím nástrojem je Query Performance Insight (dále jen QPI) [28], který umožňuje sledovat dotazy s největším využitím CPU, počtem spuštění, dobou trvání a další. Data lze agregovat (suma, průměr, maximum) nad zvolenou časovou periodou (6 hodin až měsíc).

V SQL serveru 2016 a Azure SQL Database V12 byla představena funkce Query Store [29], která shromažďuje statistická data o všech prováděných dotazech. Query Store slouží jako základní zdroj dat pro funkce poradce SQL databáze a přehled výkonu. Když je Query Store aktivní, sbírá text SQL dotazů, exekuční plány a exekuční statistiky jako jsou počet spuštění dotazu, doba běhu, logické čtení a další. Tyto informace jsou uchovány v databázi pro možnou pozdější analýzu. Pro nově vytvořené databáze v Azure SQL Database je tato funkce defaultně aktivována. Pro existující databáze je nutné ji aktivovat. Konfiguraci Query Store je možné provádět v Management studiu nebo pomocí T-SQL příkazu ALTER DATABASE SET. Parametry Query Store jsou například maximální velikost úložiště pro data, interval pro sběr statistik či doba, po jakou jsou statistiky uchovávány.

Poslední zmíněnou možností pro monitorování databáze jsou dynamicky spravované pohledy [30] poskytované službou Azure SQL Database. Tyto pohledy umožňují diagnostikovat problémy s výkonem, které mohou být způsobeny například blokovánými nebo dlouho běžícími dotazy. Pohledy jsou rozděleny do třech skupin, na pohledy vztahující se k databázi, spouštění a transakcím.

2.5.2. Aplikace

Stejně jako u monitorování databáze i na přehledu aplikace ve službě App Service je pro monitorování aplikace zobrazen monitorovací graf. Pro sledování aplikace je na výběr z šestnácti metrik, například počet požadavků a průměrný čas odpovědi.

Pro monitorování výkonu aplikace je v Azure nabízena služba Application Insights [31], která usnadňuje detekci problémů s výkonem aplikace. Tato služba obsahuje velké množství analytických nástrojů pomáhajících diagnostikovat problémy.

2.6. Zabezpečení a ochrana osobních údajů

2.6.1. Zabezpečení

Microsoft Azure poskytuje zázemí pro aplikace a data obrovskému množství zákazníků a je tedy potřeba, aby tato data byla v bezpečí. Z toho důvodu se data centra nachází v nezáplavových a bezletových zónách s nepřetržitým dohledem bezpečnostní služby. Jsou postavena na robustní architektuře, nad kterou mají trvalý dohled operační centra, kde jako administrátoři pracují pouze speciálně vyškolení pracovníci, kteří splnili osobní prověrky. Proti výpadkům jsou datacentra vybavena nezávislými napájecími větvemi, záložními zdroji, firewally, anti-DDoS²² zařízeními a dalšími bezpečnostními prvky. Veškerá komunikace mezi data centry a mezi geografickými lokalitami je šifrovaná, aby nebylo možné zachytávat přenášená data. Ochrana před útoky DDoS je součástí řízení a monitoringu Azure infrastruktury. Microsoft provádí také pravidelné penetrační testování a má vlastní týmy, které se snaží zabezpečení prorazit a na základě těchto testů se vylepšují interní procesy a technologie.

Ukazatelem kvalitního zabezpečení cloudu jsou získané certifikace od certifikačních autorit. Microsoft vlastní celou řadu bezpečnostních certifikací, ze kterých stojí za zmínku zejména certifikace ISO/IEC 27001, která definuje požadavky na systém řízení bezpečnosti informací, a ISO/IEC 27018 [32], jenž vymezuje zásady ochrany osobních údajů uživatelů v cloudu.

2.6.2. Ochrana osobních údajů

Bezpečnost v cloudu je zejména o důvěře a poskytovatel cloudu má v tomto ohledu velký zájem na tom, aby mu zákazník důvěřoval a svá data mu svěřil. Pro zákazníky je důležité vědět, že jejich data jsou používána pouze k deklarovaným účelům, a že nejsou používána ke komerčním účelům bez jejich výslovného souhlasu.

Microsoft klade velký důraz na ochranu zákaznických dat, což dokládá získání první mezinárodní certifikace ISO/IEC 27018 [32] na ochranu soukromí v cloudových službách. Tato certifikace například požaduje to, že osobní údaje v cloudu budou poskytovatelem služby zpracovány pouze k účelům, k nimž dal uživatel explicitní

²² Distributed Denial of Service

souhlas. Nebudou tedy použity k marketingovým účelům nebo k dolování dat. Když se uživatel rozhodne opustit Azure, Microsoft následuje striktní standardy a specifické procesy pro odstranění dat z jeho systému. Uživatel vždy ví, kde jsou jeho data uložena a pokud musí spravovat data ve specifické geografické poloze, například pouze v zemích EU, je to Microsoftem zajištěno. Když nastane situace, že je nutné vydat osobní údaje uživatelů třetím stranám, tak pouze na základě právně závazných požadavků (např. soudní příkaz) a musí o tom být vždy nejdříve informován uživatel.

2.6.3. Ochrana osobních údajů z pohledu legislativy

Z pohledu legislativy je uživatel cloudu zodpovědný za zpracování osobních údajů, a to i v případě, pokud jsou osobní údaje uloženy v cloudu a zpracovávány poskytovatelem cloudu jakožto zpracovatelem osobních údajů. Poskytovatel je správcem pověřen ke zpracování osobních údajů a ze zákona má povinnost uzavřít se správcem osobních údajů smlouvu o zpracování osobních údajů. Uživatel cloudu jakožto správce osobních údajů nese zodpovědnost za zpracování osobních údajů a je tedy jeho povinností, aby zajistil adekvátní opatření pro zabezpečení dat (např. pomocí jejich šifrování) [33].

U cloud computingu je běžné, že poskytovatel cloudových služeb sídlí v zahraničí a data jsou v rámci cloudu uchováвана v různých zemích i mimo členské státy EU. Povinností poskytovatele cloudových služeb je informování zákazníka o případném předávání dat do jiných států. Pokud se úložiště dat nachází na území EU a předávání dat zahrnuje pouze oblast EU, je možné data předávat bez svolení Úřadu pro ochranu osobních údajů, na základě zásady volného pohybu osobních údajů.

V rámci EU má ukládání dat k poskytovateli cloudu jasně daná pravidla, která jsou upravena evropskou směrnicí č. 95/46/ES. Tato směrnice určuje, jaká jsou práva a povinnosti poskytovatelů cloudových služeb a jejich uživatelů. Hlavním účelem této směrnice je zajištění ochrany osobních údajů subjektů, jejichž údaje jsou zpracovávány. V České republice je právní úprava obsažena v zákoně č. 101/2000 Sb., o ochraně osobních údajů, který v sobě zakotvuje principy evropské směrnice č. 95/46/ES.

Předávání dat do zemí mimo EU, tzv. třetích zemí, je obecně považováno za rizikové. Bez předchozího souhlasu Úřadu pro ochranu osobních údajů mohou být osobní údaje předávány do třetích zemí, pouze pokud jsou tyto země označeny na

základě rozhodnutí orgánu EU za země, které poskytují odpovídající úroveň ochrany dat. Za země poskytující odpovídající úroveň ochrany jsou zejména považovány ty, které ratifikovaly Úmluvu Rady Evropy 108, o ochraně osob se zřetelem na automatizované zpracování osobních údajů z roku 1981 [34]. Pokud země toto nespĺňuje, je nutné požádat úřad o schválení.

Z hlediska zákonných pravidel a záruk požadovaných pro předávání osobních údajů Úřad pro ochranu osobních údajů doporučuje využívat pouze cloudová úložiště, která se nacházejí na území Evropské unie a jsou plně pod jurisdikcí práva EU [33].

2.6.4. Citlivé osobní údaje

Citlivý údajem se rozumí podle § 4 písm. b) zákona č. 101/2000 Sb. osobní údaj vypovídající o národnostním, rasovém nebo etnickém původu, politických postojích, členství v odborových organizacích, náboženství a filozofickém přesvědčení, odsouzení za trestný čin, zdravotním stavu a sexuálním životě subjektu údajů a genetický údaj subjektu údajů. Citlivým údajem je také biometrický údaj, který umožňuje přímou identifikaci nebo autentizaci subjektu údajů.

Citlivé osobní údaje mohou být přesunuty do cloudu v rámci EU pokud se splní podmínky stanovené v § 13 zákona č. 101/2000 Sb. Citlivé osobní údaje mohou být přesunuty do cloudu ve třetí zemi mimo zemí EU/EHP jen s písemným souhlasem dotčených osob. Přesun biometrických údajů do cloudu regulátor nedoporučuje [35].

3. Analýza migrace

V následující části se zaměříme na samotnou aplikaci Škola OnLine, která bude přenášena do prostředí Microsoft Azure. Seznámíme se s tím, k čemu aplikace slouží a zmíníme některé její funkce, které poskytuje uživatelům. Dále popíšeme aplikační architekturu a stávající technické řešení aplikace. Na závěr analýzy se budeme zabývat požadavky, kterým je nutné vyhovět, aby aplikace mohla být provozována v cloudu. Kromě požadavků bude popsáno i jejich možné řešení v rámci Microsoft Azure.

3.1. Škola OnLine

Škola OnLine je webový školní informační systém, který umožňuje zpracovávat veškerou školní agendu. Systém je určen pro základní školy, střední školy, konzervatoře

a vyšší odborné školy. Jedná se o placený systém a cena Školy OnLine se liší podle typu školy, počtu žáků. Cenu dále ovlivňují funkce, které škola využívá.

Jedná se o komplexní systém, který nabízí velké množství funkcí. V práci budou zmíněny jen některé z nich. Prostřednictvím Školy OnLine je možné spravovat školní matriku, tisknout vysvědčení, provádět evidenci žáků, zadávat žákům domácí úkoly nebo s nimi sdílet výukové materiály. Rodiče mají například možnost sledovat prospěch a docházky svého potomka pomocí elektronické žákovské knížky nebo omlouvat absenci elektronicky.

Systém je možné propojit i s dalšími IT řešeními, která jsou používána na školách. Propojením se stravovacím systémem je možné objednávat jídla nebo vazbou na docházkový systém evidovat příchody a odchody žáků do školy. Podle informací z docházkových terminálů je možné zapisovat nepřítomné žáky do elektronických třídních knih. Škola OnLine dále nabízí integraci s Microsoft Office 365, propojení s Active Directory a systémem na generování rozvrhů aSc TimeTables.

Škola OnLine také nabízí mobilní aplikaci, která ovšem neobsahuje veškerou funkčnost webové aplikace.

3.2. Aplikační architektura

3.2.1. Aplikace

Aplikace Škola OnLine je vytvořena pomocí technologie ASP.NET Web Forms. Pokud se blíže podíváme na strukturu aplikace, tak se jedná o Solution ve Visual Studiu, které se skládá z několika samostatných projektů. Tyto projekty jsou sdruženy do složek podle svého určení. Z větší části se jedná o projekty realizované v jazyce C #, ale nachází se zde i projekty realizované v jazyce Visual Basic. V této práci bude do Azure převáděna pouze část aplikace, takže v Solution nejsou dostupné všechny projekty. Hlavním projektem je *UIKatedra*, což je samotná webová aplikace. Ostatní projekty vytváří knihovny, které aplikace využívá.

3.2.2. Databáze

Databázová část aplikace využívá Microsoft SQL Server 2012. Aplikace využívá jednu databázi a v té jsou uchována veškerá data Školy OnLine. Tato databáze je rozdělena do jednotlivých databázových schémat, kde jednotlivá schémata obsahují

tabulky, uživatele, procedury, triggerů a další objekty. Tato schémata se liší podle svého určení, kdy existují například schémata pro domácí úkoly či tematické plány.

Databáze Školy OnLine obsahuje dva druhy uživatelů, kteří se k ní mohou připojovat. První skupina jsou uživatelé, kteří se přihlašují pomocí přihlašovacích údajů do Windows (například doménoví uživatelé) a druhou skupinou jsou uživatelé SQL Serveru, kteří se přihlašují pomocí přihlašovacích údajů pro SQL Server. Tito uživatelé vlastní databázová schémata a mají přiřazené databázové role podle toho, jaká mají mít práva. V databázi existují předdefinované role, ale také specifické role vytvořené pro databázi aplikace. Příkladem je uživatel *sol*, který má přiřazenou roli *SOL_Execute* a tato role má přiřazené určité procedury a funkce, jenž může spouštět.

V programové části databáze se nachází velké množství uložených procedur, které umožňují opakovaně provádět kód nad daty v databázi. Procedury patří jednotlivým schématům, kde například pro schéma Platby existuje procedura zajišťující vytvoření upozornění na platbu. Dalšími programovými objekty v databázi jsou funkce, které se dělí na skalární uživatelsky definované funkce a přímé tabulkové uživatelsky definované funkce. U skalárních funkcí je vrácena jediná hodnota a jako příklad můžeme uvést funkci *CelkoveHodnoceni*, která vrací celkové hodnocení studenta. Jedná-li se o tabulkové funkce, je jejich výstupem tabulka. Příkladem může být funkce *PrumerStudent*, vracějící tabulku s průměry studentů v jednotlivých předmětech.

Pro specifické účely aplikace obsahuje databáze uživatelsky definované datové typy, které jsou použité v tabulkách a funkcích. Můžeme zmínit například datové typy *DATUM_IU* nebo *UZIV_JMENO*. V databázi se také nachází uživatelsky definovaný tabulkový typ, který reprezentuje definici struktury tabulky. Tímto typem je *StudentType* definovaný ve schématu *ImportyDM*.

U některých tabulek jsou využity triggerů, které definují činnost, jež se má provést v případě určité události nad tabulkou. V naší databázi se jedná většinou o triggerů, které se spouštějí, pokud dochází k úpravě tabulky.

Na databázovém serveru je nainstalována služba SQL Server Agent, která má za úkol spouštět plánované úlohy SQL Serveru. Úlohy pro Školu OnLine je možné rozdělit do tří kategorií. První kategorií jsou úlohy zálohovací, které provádějí zálohování databáze. Druhá kategorie jsou úlohy pro údržbu, mezi které patří úlohy pro defragmentaci indexů

na tabulkách nebo aktualizaci statistik pro tabulky. A poslední kategorií jsou servisní úlohy. Zde můžeme zmínit například úlohu, která každý den v noci spouští proceduru kontrolující, zdali žáci dosáhli plnoletosti.

3.3. Stávající řešení

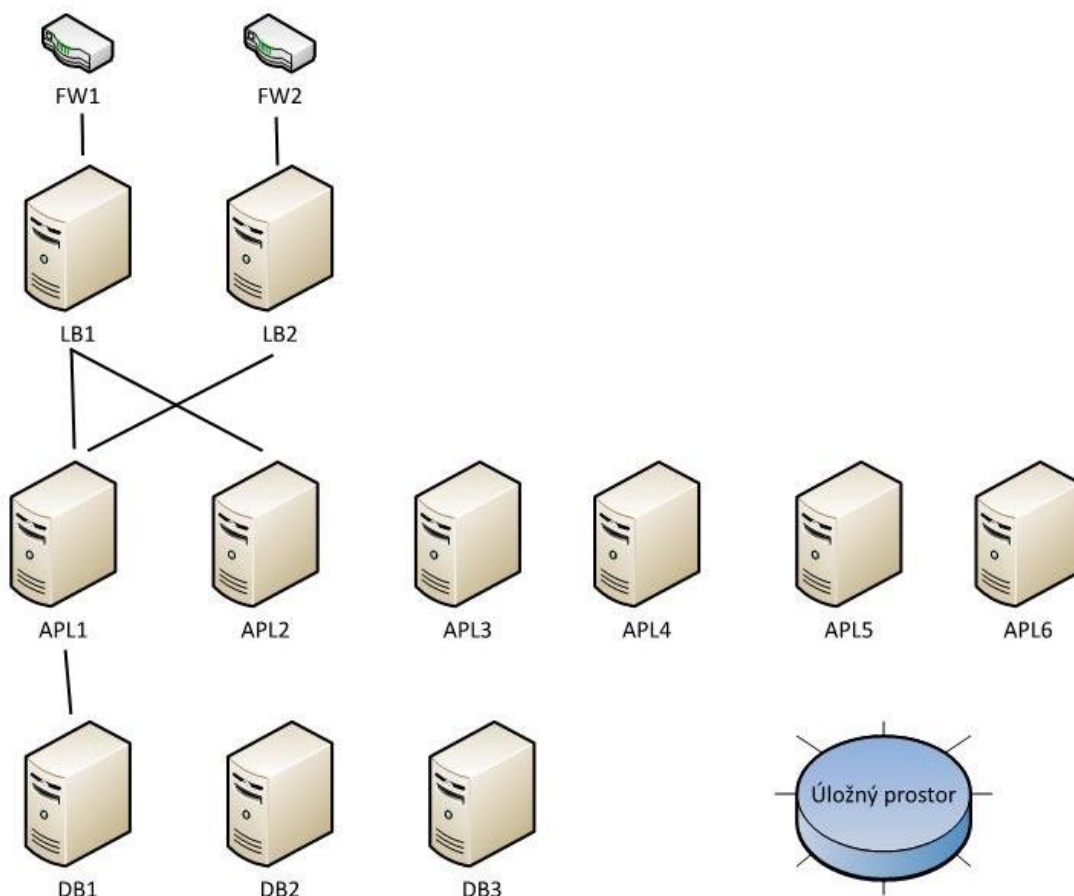
V současné době jsou servery aplikace provozovány v datovém centru, které bude blíže popsáno. Poté se seznámíme se sítíovou architekturou a tím, jak jsou zpracovány příchozí požadavky. U stávajícího řešení nás také zajímá, jakým způsobem je řešeno zálohování dat, zabezpečení komunikace a monitoring služeb.

3.3.1. Datové centrum

Školní informační systém Škola OnLine je v současnosti provozován v datovém centru DC Tower společnosti České Radiokomunikace. V tomto datovém centru má Škola OnLine pronajatý rack, ve kterém má umístěny vlastní servery. Datové centrum splňuje požadavky Národního bezpečnostního úřadu na zvýšený bezpečnostní režim. Nachází se v objektu mimořádného strategického významu pro stát, Tento objekt je situován v bezletové zóně a mimozáplavové oblasti. Napájení budovy je řešeno třemi nezávislými větvemi v kombinaci se záložními zdroji pro zabránění výpadku [36].

3.3.2. Sítíová architektura

Nyní popíšeme sítíovou architekturu Školy OnLine a to, jak probíhá zpracování příchozí komunikace z internetu (viz obr. 3.1). Příchozí požadavky z internetu přicházejí nejprve na firewally, které tyto požadavky dále přeposílají na load balancery, sloužící k rozložení zátěže mezi aplikační servery. Load balancer v závislosti na zátěži nasměruje komunikaci na určitý aplikační server. Aplikační servery Školy OnLine běží v rámci webové farmy a toto řešení zajišťuje provoz aplikace i v případě, že by došlo k výpadku některého z nich. Aplikační servery dále komunikují s databázovými servery, které jsou provozovány v rámci databázového clusteru. Session se udržuje pouze v rámci jednoho aplikačního serveru. Pokud tedy při dalším přístupu dojde k přesměrování uživatele na jiný aplikační server, je nutné, aby se znovu přihlásil.



Obrázek 3.1: Síťová architektura

3.3.3. Zálohování dat

Veškerá data, která školy uchovávají v rámci Školy OnLine, jsou uložena v úložném prostoru na diskovém poli s RAID²³ řadičem. Díky tomuto řešení jsou data zabezpečena proti selhání pevného disku a ztrátě dat. Pro zajištění bezpečného uchování i v případě aplikačního selhání je prováděna pravidelná záloha dat do jiného úložiště. Dále jsou data zálohována i do úložiště, které se nachází v jiné zabezpečené geografické lokalitě. Do tohoto umístění jsou data pravidelně přenášena v zašifrované podobě.

3.3.4. Zabezpečení komunikace

Co se týká bezpečnosti, tak přenos důvěrných dat mezi uživatelem a servery je zajištěn zabezpečeným síťovým protokolem HTTPS²⁴. Škola OnLine využívá SSL certifikát vydaný certifikační autoritou a je automaticky akceptovaný ve všech internetových prohlížečích.

²³ Redundant Array of Independent Disks

²⁴ Hypertext Transfer Protocol Secure

3.3.5. Monitorování služeb

Všechny hardwarové a softwarové komponenty řešení jsou neustále kontrolovány, aby byl zajištěn jejich řádný běh a dostupnost. Pokud jsou identifikovány problémy, dochází k okamžitému kontaktování zodpovědných techniků. Technici mají neustále přehled o stavu řešení a jeho vytížení, což jim umožňuje včas zasáhnout, pokud dochází k problémům.

3.4. Požadavky a jejich řešení

Pro Školu OnLine jsou základními požadavky spolehlivost a dostupnost, aby byl systém stále dostupný pro uživatele bez výpadků. Učitelé systém používají při své každodenní práci a proto je nutný jeho bezproblémový běh. Musí být možné Školu OnLine používat v kteroukoliv denní dobu, nejen v době výuky, protože systém využívají žáci pro získávání učebních materiálů nebo k vypracování domácích úkolů. Pokud by docházelo k výpadkům systému, mohlo by to mít dopad na fungování celé školy.

Důležitým parametrem pro Školu OnLine je také výkon, aby systém dokázal odolávat zvýšené zátěži. Ke zvýšené zátěži může docházet zejména v určitých obdobích školního roku, jako jsou pololetí nebo konec školního roku, kdy dochází k přípravě vysvědčení. V tu dobu souběžně do systému přistupuje velké množství uživatelů, kteří připravují vysvědčení a tisknou je. Pokud by na takovou zátěž systém nebyl připraven, mohlo by docházet k jeho zpomalení, či dokonce výpadkům.

Nyní bude popsáno, jak je možné těmto požadavkům vyhovět v Microsoft Azure.

3.4.1. Spolehlivost

U Microsoft Azure je kladen velký důraz na spolehlivost a dostupnost. Dostupnost je garantována v rámci smlouvy SLA, která popisuje závazky společnosti Microsoft týkající se dostupnosti a konektivity. Smlouvy SLA se liší pro jednotlivé služby Azure.

U služby VM je pro všechny virtuální počítače, které mají ve stejné skupině dostupnosti nasazeny dvě a více instancí zaručeno připojení virtuálního počítače minimálně k jedné instanci v 99,95 % času. Pokud se jedná o virtuální počítač pouze s jednou instancí, který využívá prémiové úložiště pro všechny disky, je garantována dostupnost minimálně v 99,9 % času [37].

V rámci smlouvy SLA pro službu App Service Microsoft garantuje, že všechny aplikace spuštěné v rámci předplatného budou dostupné minimálně v 99,95 % času. Pro plány Free a Shared není poskytována žádná smlouva SLA [38].

U SQL databáze je dostupnost garantována v 99,99 % pro izolovanou nebo elastickou databázi, a to v úrovních Basic, Standard a Premium [39].

Pro Azure Redis Cache je zaručeno, že mezipaměť na úrovni Standard nebo Premium bude dostupná minimálně 99,9 % času. Smlouva SLA v tomto případě pokrývá jenom připojení ke koncovým bodům mezipaměti a součástí smlouvy není ochrana proti ztrátě dat [40].

3.4.2. Výkon

Co se týká výkonu, tak u všech služeb v Azure je možné vybírat z velkého množství výkonnostních konfigurací, aby bylo možné vyhovět požadavkům uživatelů.

U služby virtuálních počítačů jsou výkonnostní plány rozděleny podle zaměření. Jedná se například o virtuální počítače, které jsou výpočtově orientované a nabízí tak vysoký poměr procesorů k paměti. Dále mohou být paměťové orientované. Ty nabízí vysoký poměr paměti k jádrům. U většiny virtuálních počítačů je nabízena podpora pro vyrovnávání zatížení a automatické škálování.

Také u služby App Service je nabízeno několik výkonnostních variant, z kterých je možné vybrat tu nejvhodnější. Plány Standard a Premium u App Service obsahují podporu vyrovnávání zatížení v síti a automaticky je distribuován provoz mezi instance virtuálních počítačů. U těchto plánů je také integrována podpora automatického škálování, která dokáže automaticky upravovat počet běžících instancí virtuálních počítačů podle aktuálních požadavků na provoz.

Důležité je zvolit odpovídající výkonovou úroveň u služby Azure SQL Database, která slouží k hostování databáze. U této služby se výkon jednotlivých úrovní udává v jednotkách DTU [21]. Čím větší počet jednotek DTU, tím větší je výkon databáze.

Rychlost mezipaměti má také vliv na výkon celé aplikace, a proto i u Redis Cache existuje možnost pro výběr z několika výkonových úrovní.

4. Migrace aplikace

V této části se budeme zabývat samotnou migrací aplikační a databázové části do prostředí Azure. Na základě předchozí analýzy budou vybrány cloudové služby, do kterých bude provedena migrace.

4.1. Výběr služeb

4.1.1. Databázová část

První možností, která se nabízí pro hostování databáze Školy OnLine, je SQL Server provozovaný na virtuálních počítačích Azure. SQL Server na virtuálních počítačích je optimalizovaný pro migraci stávajících databází do Azure. Dostupné jsou všechny verze a edice SQL Serveru. Je 100% kompatibilní se stávajícím SQL Serverem a dovoluje hostovat libovolné množství databází a provádět mezidatabázové transakce. Toto řešení nabízí plnou kontrolu nad SQL serverem a systémem Windows. Pokud by bylo zvoleno toto řešení, tak nejsou očekávané žádné problémy s migrací stávající databáze do prostředí Azure.

Když se zaměříme na náklady u toho řešení, budou záviset na velikosti virtuálního počítače a na zvolené edici SQL Serveru. Cena za licenci SQL Serveru a systému Windows je účtována po minutách a je k ní ještě nutné připočítat cenu na úložiště Azure pro disky virtuálních počítačů. Dále je potřeba počítat s náklady za administraci, kam patří správa databázového systému, operačního systému, instalace upgradů a patchů.

Toto řešení by bylo ideální, pokud by stávající aplikace používala rozsáhlé databáze nebo potřebovala mít přístup ke všem funkcím SQL serveru a systému Windows. Bylo by to také vhodné, pokud bychom chtěli migrovat bez toho, abychom databázi měnili. Dále také pokud by bylo požadavkem mít plnou kontrolu nad SQL serverem a Windows.

Druhou možností je využití služby Azure SQL Database, která je optimalizována pro snížení celkových nákladů na minimum pro zřízení a správu velkého množství databází. Jsou u ní také sníženy náklady na průběžnou administraci a není nutné se zabývat správou upgradů, či zajištěním vysoké dostupnosti. Cena za službu se liší podle zvolené cenové úrovně, která je účtována po hodinách dle zvolené cenové a výkonové úrovně. U této služby jsou účtována odchozí data. Předností tohoto řešení je možnost

přepínat podle potřeby mezi cenovými a výkonovými úrovněmi služeb, aby bylo možné vyhovět měnícím se potřebám aplikace.

U tohoto řešení je nutno brát v potaz, že se jedná o logický server Azure SQL Database, který se liší od instance SQL Serveru. V Azure SQL Database existují omezení a ne všechny funkce, nabízené u SQL serveru, jsou zde podporované. Například SQL Server agent v Azure není podporován. Stejně tak některé příkazy jazyka T-SQL²⁵ nejsou podporované. Z toho důvodu, že Azure SQL Database je koncipovaná tak, aby funkce byly izolovány od závislostí na hlavní databázi a operačním systému. Některé serverové aktivity jsou tedy pro službu SQL Database nevhodné. Jedná se většinou o příkazy konfigurující možnosti serveru nebo operačního systému. Pro chybějící příkazy ale Azure často nabízí odpovídající alternativu nebo je možné využít jinou funkci či službu Azure [41]. Můžeme zmínit například příkaz *CREATE PROCEDURE*, který je dostupný, ale již nejsou dostupné všechny možnosti tohoto příkazu.

Toto řešení se jeví jako ideální v případě požadavku na úsporu nákladů a lepší optimalizaci výkonu. Další výhodou je integrovaná funkce pro zálohování, která pomáhá dosáhnout snížení nákladů a to hlavně u velkého množství databází. Hodí se pro řešení, u kterého je nutná integrovaná vysoká dostupnost a možnost zotavení po havárii.

Pro naši práci je preferovaným řešením využití Azure SQL Database, a to hlavně z důvodů snížení nákladů na administraci. Nechceme se zabývat správou operačního systému a nastavení konfigurace.

4.1.2. Aplikační část

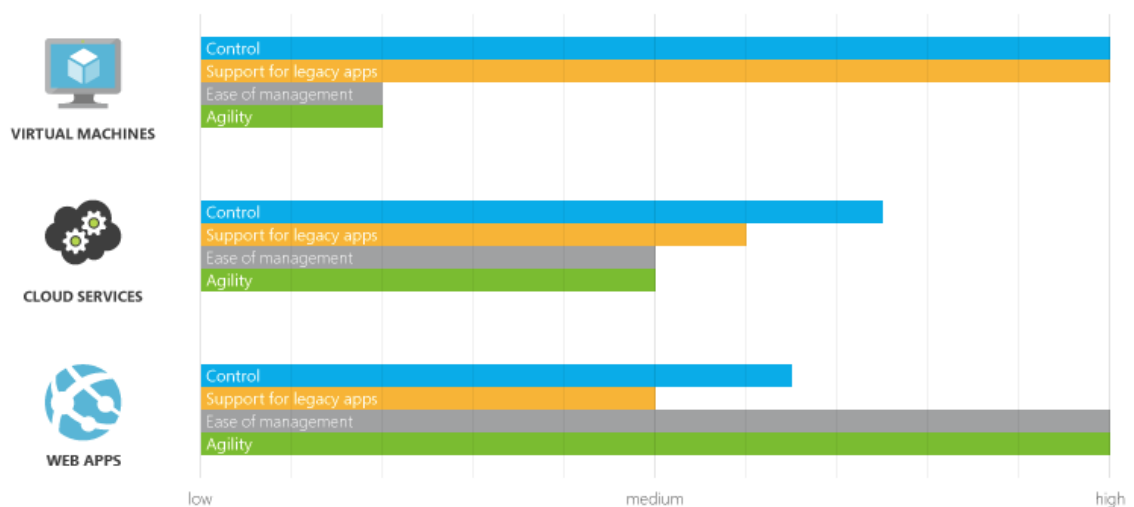
Azure nabízí několik možností pro hostování webových aplikací. Pro aplikační část lze vybírat z Virtual Machines, Cloud Services a App Service. První řešení je založeno na modelu IaaS. Jedná se tedy o zakoupení virtuálního počítače, na kterém by byl provozován webový server a v něm hostována samotná aplikace. Toto řešení by se pro aplikaci hodilo v případě, že by pro běh v Azure App Service byly vyžadovaly velké změny v kódu aplikace. Na druhou stranu konfigurace, zabezpečení a údržba virtuálního počítače si vyžaduje značné časové nároky a odbornost v porovnání s App Service.

²⁵ Transact-SQL

U virtuálního počítače je potřeba brát v potaz nároky na neustálou údržbu, například instalace patchů či update prostředí.

Druhou možností je Azure App Service. Toto řešení bývá nejčastější volbou pro většinu webových aplikací. Deployment a správa jsou přímo zabudovány v platformě, aplikace mohou být snadno škálovatelné v případě větší zátěže a zabudovaný load-balancing a manažer provozu zaručují vysokou dostupnost. Microsoft poskytuje pro migraci webových aplikací do Azure online nástroj Azure App Service Migration Assistant. Při vývoji a nasazení aplikace je potřeba brát v potaz určitá specifika provozu v cloudu. Aby bylo možné přidávat instance a aplikace fungovala bez problémů, musí být bezstavová. Informace o uživatelských relacích je tedy potřeba ukládat do odděleného stálého úložiště (např. Azure Redis Cache). Škálování také nebude fungovat, když aplikace využívající lokální databázi (např. SQL Server Compact). Aby škálování mohlo fungovat, je vhodné využít oddělenou databázi.

Jednotlivá řešení se liší úrovní kontroly, podporou legacy aplikací, jednoduchostí správy a agilitou (viz obr. 4.1).



Obrázek 4.1: Porovnání Azure služeb [42]

4.2. Databázová část

Pro databázovou část byla vybrána služba Azure SQL Database, která nejvíce vyhovovala požadavkům. Před migrací bylo nejprve nutno ověřit kompatibilitu stávající databáze, která je provozována v rámci SQL Serveru, se službou Azure SQL Database. Během analýzy migrace jsme zjistili, že SQL Server agent v Azure není podporován

a bylo nutné nalézt alternativu, která umožňuje nahradit funkčnost úloh spouštěných v rámci SQL Server agenta.

Migrace databáze vyžadovala otestování kompatibility, opravu nalezených problémů a samotný přenos databáze do Azure SQL Database. Budou popsány metody, které je možné použít pro otestování problémů s kompatibilitou databáze. Poté bude popsáno, jak byly vyřešeny nalezené problémy, jak byla SQL databáze přenesena do Azure a jaké nástroje a metody k tomu byly použity.

4.2.1. Testování kompatibility

Proces migrace přenáší schéma a data z databáze systému SQL Server ve stávajícím prostředí do služby Azure SQL Database. Pro úspěšné provedení migrace vyžadovala stávající databáze otestování kompatibility. Příchod služby SQL Database V12 se podporou funkcí začal přibližovat stávajícímu systému SQL Server, ale stále se mohou vyskytovat problémy s kompatibilitou.

Existuje několik nástrojů, pomocí kterých je možné ověřit kompatibilitu databáze systému SQL Server se službou SQL Database.

- **Průvodce aplikací Export Data Tier v sadě SQL Management Studio:** v tomto průvodci jsou zjišťovány chyby a pokud nejsou nalezeny, je možné v něm dokončit migraci. Pro opravu zjištěných problémů musí být použit jiný nástroj.
- **SQL Azure Migration Wizzard:** jedná se o nástroj, který není oficiálně podporován Microsoftem, ale lze ho také využít k ověření kompatibility. Je možné přímo v tomto nástroji opravit nalezené problémy nebo dokončit migraci do Azure.
- **SQL Server Data Tools pro Visual Studio:** tento nástroj používá nejnovější pravidla kompatibility k zjišťování nekompatibilit služby SQL Database V12. Při objevení problémů je možné je v tomto nástroji opravit.
- **SqlPackage:** jedná se o nástroj příkazového řádku, který testuje kompatibilitu a generuje sestavu obsahující zjištěné problémy s kompatibilitou. V případě zjištění problémů je nutné použít jiný nástroj pro jejich opravu.

- **Data Migration Assistant** (dále jen DMA): nástroj od společnosti Microsoft sloužící ke kontrole kompatibility. Neumožňuje nalezené problémy opravit a k opravě je nutné použít jiný nástroj.

K našemu testu kompatibility byl použit nástroj DMA a opravy provedeny v SQL Management studiu. Popis nalezených problémů a jejich oprava bude obsahem následující části.

4.2.2. Oprava nalezených problémů

Nalezené problémy jsou v nástroji DMA rozděleny na několik skupin, a to na problémy bránící migraci, změny chování a vlastnosti označené jako deprecated.

První nalezený problém se týkal osiřelých uživatelů. V databázi *master* jsou evidovány loginy uživatelů a v uživatelské databázi jsou evidovány uživatelské účty, které umožňují loginům přistupovat k databázi. Uživatelský účet obsahuje informaci, ke kterému ID loginu patří. Tím, že záloha databáze byla přenesena na jinou instanci SQL serveru, uživatelský účet nemá správné ID loginu, ke kterému patří. Řešením tohoto problému je příkaz `ALTER USER user WITH LOGIN = "login"`. Tímto příkazem řekneme SQL serveru, aby opravil ID loginu podle zadaného uživatelského účtu a loginu.

Další problém s uživateli databáze se týkal těch uživatelů, kteří využívají přihlašovací údaje Windows. Nástroj je reportoval tak, že se jedná u uživatele, kteří mají parametr *AuthenticationType* nastaven na hodnotu, která není podporována v Azure SQL Database V12. Řešením v tomto případě bylo smazání těchto uživatelů. Protože někteří z těchto uživatelů vlastnili schéma, bylo nutné změnit vlastníka tohoto schématu a až poté bylo možné uživatele smazat. Slouží k tomu příkazy: `ALTER AUTHORIZATION ON SCHEMA::SchemaName TO user` a `DROP USER myUser`.

Procedura `dbo.CCAK_EXISTUSER` využívá procedury `sp_dropserver`, `sp_addlinkedserver` a `sp_addlinkedsrvlogin`, které nejsou v Azure podporovány. Tato procedura není v databázi nijak využívána, a proto bylo možné ji smazat.

V proceduře `dbo.CCAK_PR_SlouceniKol` se vyskytovala chyba syntaxe příkazu `RAISERROR 50001 @ERROR_MESSAGE`, protože tato syntaxe již není podporována a byla nahrazena syntaxí `RAISERROR()`. Navíc od verze SQL serveru 2012 byl

představen příkaz *THROW*, který Microsoft doporučuje používat [43]. Příkaz byl tedy nahrazen příkazem *THROW 50001, @ERR_MESSAGE, 1*. Dále muselo být zajištěno, aby předcházející příkaz končil středníkem.

V databázi se nachází databázové trigger *TDB_CREATE_FUNCTION* a *TDB_CREATE_PROCEDURE*, které již nejsou podporovány v rámci SQL serveru a byly odstraněny.

Existují procedury *dbo.CalcSpace*, *dbo.dt_checkinobject* a *dbo.dt_checkoutobject*, které obsahují reference na nedokumentované systémové tabulky, jenž byly přejmenovány. Zmíníme například tabulku *sysobjects*, která byla od verze SQL serveru 2008 přejmenována na *sys.objects*. Jelikož procedura *dbo.CalcSpace* nebyla již nijak využívána, mohla být smazána. Ostatní dvě procedury využívají systémovou tabulku *syscomments*, která v SQL serveru 2008 byla přejmenována na *sys.syscomments* a v následujících verzích SQL serveru již není podporována. V databázi se nachází celý balíček procedur, které začínají „*dt.*“. Jedná se o procedury, které jsou vytvořeny po připojení k databázi z Visual Studio Tools. Tyto procedury nejsou pro migraci důležité a byly tedy smazány.

V databázi existoval problém týkající se externích referencí, kdy databáze v procedurách *dbo.CCAK_Dok_Vzor_Export*, *dbo.CCAK_Organizace_Export*, *dbo.CCAK_Organizace_Import* a *dbo.CCAK_Organizace_Insert* obsahovala referenci na uloženou proceduru *xp_cmdshell*, která se nachází v databázi master a umožňovala spouštění příkazů v příkazové řádce Windows. Protože tyto uložené procedury byly přímo v aplikaci programově nahrazeny, bylo možné tyto procedury smazat.

U procedur *dbo.CCAK_Dok_Vzor_Export* a *dbo.CCAK_Vzor_Export_To_String* jsou využívány globální dočasné objekty, které nejsou v Azure SQL podporovány. Řešením tohoto problému by bylo použití lokálních dočasných tabulek nebo tabulkových proměnných namísto globálních dočasných tabulek. To by mohlo vyžadovat přepsání procedur k docílení stejné funkcionality. Avšak jak již bylo zmíněno dříve, tyto procedury byly nahrazeny v aplikaci programově a byly z databáze smazány.

V pohledech *dbo.CCAK_AKTIVNI_UCITELE_VIE*, *dbo.CCAK_UZIVATEL_VIE* a *dbo.CCAK_DOK_DRUH_VYCH_OP* byla identifikována syntaxe *Unqualified Join*. Od

databáze kompatibility 90 a vyšší může v některých případech tato syntaxe způsobit varování, že chybí příkaz *Join*, a to vede k dlouhému běhu dotazu. Příklad *Unqualified Join*:

```
SELECT * FROM table1, table2
WHERE table1.col1 = table2.col1
```

Tento problém byl vyřešen tak, že za pomoci příkazu *ALTER VIEW* byly pohledy upraveny a byl použit explicitní příkaz *Join*. Příklad jednoho upraveného pohledu:

```
ALTER VIEW [dbo].[CCAK_UZIVATEL_VIE] ( UZIVATEL_ID,
OSOBA_ID,
ORGANIZACE_ID,
JMENO,
PRIJMENI,
UZIV_JMENO,
HESLO,
CS_OSOBA_ID)
AS SELECT UZIVATEL.UZIVATEL_ID UZIVATEL_ID
, OSOBA.OSOBA_ID OSOBA_ID
, OSOBA.ORGANIZACE_ID ORGANIZACE_ID
, OSOBA.JMENO JMENO
, OSOBA.PRIJMENI PRIJMENI
, UZIVATEL.UZIV_JMENO UZIV_JMENO
, UZIVATEL.HESLO HESLO
, OSOBA.CS_OSOBA_ID CS_OSOBA_ID
FROM CCAK_OSOBA OSOBA
INNER JOIN CCAK_UZIVATEL UZIVATEL
ON osoba.organizace_id = uzivatel.organizace_id
and osoba.osoba_id = uzivatel.osoba_id
```

Stejný problém se vyskytoval také u triggerů *dbo.TU_DOT_RESP*, *dbo.TU_UZAV_HODN* a *dbo.tiu_praxe_student*. I u těchto triggerů byl u příkazu *SELECT* použit *INNER JOIN*.

V databázi se nacházely datové typy *text*, *ntext* a *image*, které budou v budoucích verzích SQL serveru odstraněny. Nástroj DMA doporučil nepoužívat tyto datové typy a nahradit je datovým typem *varchar(max)*, *nvarchar(max)* a *varbinary(max)*. Například u databáze *dbo.CCAK_AKTUALITY* to bylo provedeno příkazem:

```
ALTER TABLE dbo.CCAK_AKTUALITY ALTER COLUMN DETAIL
nvarchar(max) Null
```

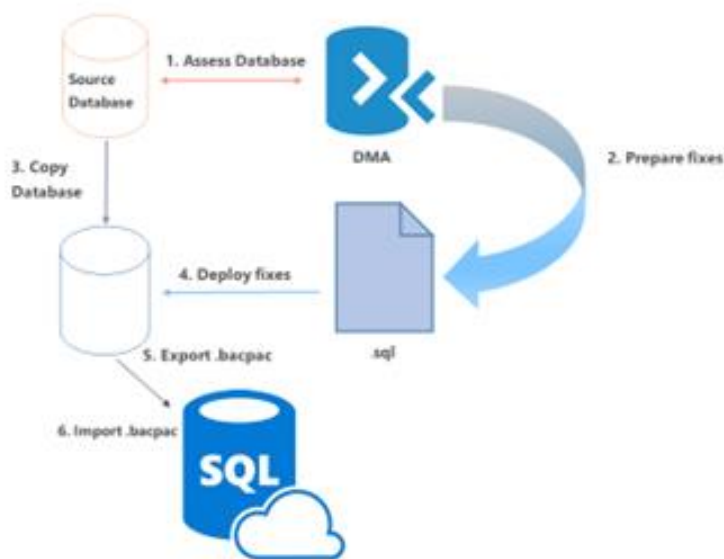
Posledním problémem, který nástroj objevil, byl Unresolved reference. U pohledu *dbo.CCAK_DOK_INV_MISTNOSTI_DLE_SB* se to týkalo sloupce *BUDOVA_ID*

z tabulky *CCAK_MISTNOST*, který byl v pohledu špatně nadefinován jen jako *BUDOVA*, a tak došlo k problému s referencí. U triggeru *dbo.TU_BLOKY_REAL* byl problém s Unresolved reference způsoben tím, že trigger během úpravy tabulky při splnění určité podmínky maže sloupec v tabulce *dbo.CCAK_REALIZACE_ROZSAH*. V současném stavu databáze tento sloupec neexistoval a to způsobovalo problém s referencí. Tato chyba byla také reportována u pohledu *dbo.PREVOD_REFERENCES_VIE*, ale v tomto případě bylo možné ji ignorovat. Po provedení všech zmíněných oprav byla databáze připravena k migraci do Azure SQL Database.

4.2.3. Migrace databáze

Doporučený postup pro provedení migrace do Azure je ověření kompatibility pomocí nástroje DMA, příprava nutných oprav ve formátu T-SQL skriptů, vytvoření kopie stávající databáze a spuštění opravných skriptů nad kopií databáze. Následně je nutné tuto kopii databáze vyexportovat do souboru ve formátu BACPAC na lokální disk a importovat tento BACPAC soubor do nové Azure SQL databáze za použití nějakého nástroje pro import (viz obr. 4.2). Doporučeným nástrojem je SQLPackage.exe kvůli jeho nejlepšímu výkonu ze všech nabízených nástrojů [44].

Azure SQL Database migration



Obrázek 4.2: Postup migrace do Azure SQL databáze [44]

Dalším možným způsobem je publikování databáze přímo z SQL Management studia. Nad zvolenou databází je spuštěna úloha Deploy Database to Microsoft Azure SQL Database, kde se stačí přihlásit k SQL serveru v Azure, definovat název databáze v Azure, druh edice a umístění dočasného BACPAC souboru.

V práci byl následován doporučený postup a nyní bude blíže popsán export databáze z SQL serveru a její import do služby Azure SQL Database za použití nástroje SQLPackage.exe.

Export databáze

K exportu byl použit nástroj SQLPackage.exe, který bývá dodáván společně s SQL Server Management studiem. Nástroj se spouští z příkazové řádky, kde se jako parametry definují druh akce, název databázového serveru, název databáze pro export a název cílového souboru. Příkaz pro export databáze do souboru formátu BACPAC vypadá takto:

```
sqlpackage.exe /Action:Export /ssn:<server_name>  
/sdn:<db_name> /tf:<target_file>
```

Import databáze

Před samotným importem databáze bylo potřeba vytvořit v Azure logický SQL server, který bude hostovat databázi. Po vytvoření SQL serveru bylo možné provést samotný import. Zde je příkaz, který k tomuto účelu byl použit:

```
sqlpackage.exe /Action:Import /tsn:sol.database.windows.net  
/tdn:sol /tu:user /tp:password /sf:c:\sql\SOL-DB.bacpac
```

4.2.4. Nahrazení úloh SQL serveru

Jak již bylo zmíněno dříve, v Azure SQL neexistuje podpora pro SQL Server agenta a bylo potřeba nalézt řešení pro nahrazení úloh, které byly pravidelně spouštěny nad databází. Jako náhradu SQL Server agenta je možné k tomuto účelu využít službu Azure Automation. S pomocí této služby budeme definovat runbooky, které budou spouštět uložené procedury.

Úlohy na SQL serveru vlastně pouze spouští uložené procedury v předem definovaný čas. Můžeme zmínit například úlohu *Utils-SOL*, která je spouštěna každou noc a volá

uloženou proceduru *dbo.CCAK_GenerujNarozeninyStud*. Tato procedura kontroluje narozeniny žáků a vytváří upozornění na narozeniny pro třídní učitele.

Nejprve bylo nutné vytvořit účet pro Azure Automation, což se provádí na portálu Azure. V tomto vytvořeném účtu byly pro potřeby runbooku nadefinovány přihlašovací údaje k databázi. Poté již byly vytvořeny samotné runbooky, které je možné vytvořit buď přímo na portálu nebo importovat již hotový powershellový skript. U našeho řešení byly runbooky vytvořeny přímo v prostředí portálu a to jako typ PowerShell workflow.

Pro takto vytvořené runbooky bylo poté nutné vytvořit plány, které slouží k naplánování spouštění runbooků. Jeden runbook může být obsažen v několika plánech a zároveň jeden plán může obsahovat několik runbooků. V současném řešení jsou v SQL serveru joby rozděleny podle svého zaměření a doby spouštění. Každý job obsahuje několik procedur, které se spouštějí ve stejný čas. Bylo tedy možné stejně rozdělit runbooky podle doby spouštění a druhu do jednotlivých plánů.

4.3. Aplikační část

Pro aplikační část bylo na základě analýzy jako preferované řešení zvoleno využití služby Azure App Service. V následující části bude popsáno publikování aplikace do Azure.

4.3.1. Publikování aplikace

Prvotní testovací publikování aplikace do Azure bylo provedeno přímo z Visual Studia s využitím metody Web deploy. Dále budou v práci blíže popsány všechny metody pro deployment aplikace do Azure. Podmínkou pro publikování aplikace je mít v prostředí Azure vytvořený plán služby App Service. Plán je možné vytvořit předem na Azure portálu, a nebo ho vytvořit během publikování aplikace. Dále se během publikování definují parametry připojení k aplikaci a připojení k databázi. Připojení k databázi je možné nastavit přímo v souboru *web.config*. Pokud je připojení k databázi nastaveno během publikování aplikace, tak přepisuje nastavení definované v souboru *web.config*.

Solution ve Visual Studiu obsahuje několik projektů, z nichž do Azure byl publikován hlavní projekt UIKatedra. Po úspěšném publikování je aplikace dostupná na webové adrese, která byla nadefinována.

4.3.2. Ukládání dat

Problém při migraci aplikace může nastat, pokud aplikace ukládá soubory do lokálního úložiště. Aplikace Škola OnLine využívá lokální úložiště k ukládání pracovních souborů, jako jsou úkoly, výukové materiály či žádosti.

V takovém případě se jako řešení jeví využití služby Azure Storage, která nabízí čtyři typy úložišť. Pro ukládání dat aplikace Škola OnLine nejvíce vyhovuje typ Blob storage, který slouží k ukládání nestrukturovaných datových objektů. Do Blob úložiště je možné ukládat jakýkoli druh textu nebo binárních dat, jako je dokument nebo soubor médií.

K využití služby Azure Storage je nejprve potřeba vytvořit účet úložiště. V tomto účtu se nachází Blob Storage, ve kterém musíme vytvořit kontejnery, které zajišťují seskupení objektů blob. Účet může obsahovat neomezený počet kontejnerů a kontejner může obsahovat neomezený počet objektů blob. Objekt blob je soubor libovolného typu a velikosti. Konkrétně pro Školu OnLine je možné vytvořit jednotlivé kontejnery pro úkoly, výukové materiály a další.

Pro použití v aplikaci je potřeba nejprve stáhnout potřebné balíčky knihoven, a to Microsoft Azure Storage Client Library for .NET zajišťující programový přístup k datovým prostředkům na účtu úložiště a Microsoft Azure Configuration Manager Library for .NET, který poskytuje třídu pro potřeby analýzy připojovacího řetězce v konfiguračním souboru. Poté je potřeba upravit soubor web.config, kde je definováno lokální úložiště a změnit tuto definici na připojení k Azure Storage.

Současná definice:

```
<add key="DataDir" value="c:\ccaapl\SOL"/>
```

Definice pro Azure Storage:

```
<add key="DataDir"
value="DefaultEndpointsProtocol=https;AccountName=account-
name;AccountKey=account-key" />
```

Poté musíme ve třídách využívajících úložiště přidat jmenné prostory Microsoft.Azure, Microsoft.WindowsAzure.Storage a Microsoft.WindowsAzure.Storage.Blob. V programu je nejprve potřeba analyzovat

připojovací řetězec k Azure Storage a poté po vytvoření klienta služby objektu blob je možné s úložištěm pracovat. Je možné vytvářet nové kontejnery, bloby, nahrávat objekty blob do kontejnerů, zobrazovat seznam objektů blob v kontejneru, stáhnout objekty blob a další.

4.3.3. Nastavení stavu relace

Ve stávajícím řešení aplikace udržovala stav relace v režimu *InProc*, což znamená, že se stav relace udržuje pouze v paměti na webovém serveru. Toto řešení ale není ideální pro použití v cloudu, kde webové aplikace často běží na více instancích. Vhodnější je proto použití externího poskytovatele pro udržení stavu relace, protože bez něho by byla aplikace omezena pouze na jednu instanci. Proto bylo navrženo a realizováno řešení využívající mezipaměť Redis Cache, která je nabízena v Azure pro udržení stavu relace. Samotná konfigurace je snadná a provádí se v souboru `web.config`. Bylo potřeba změnit parametr `sessionState` z režimu *InProc* na režim *Custom* a přidat parametry poskytovatele, což je v našem případě Redis Cache. Nejprve tedy bylo nutné v Azure vytvořit plán služby Redis Cache. Po jeho vytvoření jsme získali přístupové údaje, které se použili v souboru `web.config`. Služba Redis Cache nabízí přístup k sadě funkcí a ekosystému Redis a je zajištěno spolehlivé hostování a monitorování od Microsoftu.

4.3.4. Využití vlastní domény

Když je vytvořena webová aplikace, Azure jí zařadí do subdomény `azurewebsites.net`. V našem případě byla aplikace pojmenována `skolaonline` a tak byla vytvořena URL adresa `skolaonline.azurewebsites.net`. Zároveň dojde k vytvoření virtuální IP adresy. Škola OnLine má zakoupenou vlastní doménu a Azure poskytuje možnost pro použití vlastní domény [45]. Podpora pro vlastní doménu je nabízena od plánu Shared výše. Pro nasměrování vlastní domény na webovou aplikaci vytvořenou v Azure je nejdříve nutné správně nastavit DNS²⁶ záznamy, aby tuto doménu směřovali na Azure. Konfigurace DNS záznamu se obvykle provádí v konfiguračním nástroji poskytovaném registrátorem domény. Po nastavení DNS je možné přiřadit vlastní doménu v nastavení aplikace na portále.

²⁶ Domain Name System

4.3.5. Podpora HTTPS

V současné době Škola OnLine podporuje zabezpečenou komunikaci přes protokol HTTPS. Pro aplikaci v Azure také požadujeme zabezpečené připojení. Pro zabezpečené připojení přes protokol HTTPS pro aplikaci s vlastní doménou je nutné k této doméně přidat SSL certifikát. V základu Azure poskytuje zabezpečení wildcard certifikátem [46] pro doménu *.azurewebsites.net. Tento certifikát však nelze použít pro vlastní domény. Škola Online vlastní SSL certifikát vydaný certifikační autoritou a ten bude využit pro zabezpečení aplikace v Azure. Než bude možné certifikát použít, je potřeba provést nastavení vlastní domény, což bylo popsáno v předchozí sekci. Dále je nutné používat plán Basic nebo vyšší, ve kterém je podpora vlastní SSL certifikáty. Když jsou tyto podmínky splněny, na portále v nastavení aplikace nahrajeme certifikát na portál. Po úspěšném nahrání certifikátu, stačí vytvořit vazbu mezi certifikátem a vlastní doménou.

4.4. Možnosti pro deployment

Azure App Service podporuje tři metody pro deployment aplikace. V této části budou uvedeny možnosti pro deployment aplikace a z nich vybrána ta nejvhodnější. Všechny možnosti pro deployment využívají některé z následujících metod:

- **FTP²⁷ nebo FTPS²⁸**: za použití libovolné FTP klienta je možné přenést soubory aplikace do Azure. V tomto případě nejsou App Service poskytovány žádné další služby [47].
- **Kudu**: jedná se o nástroj, který má na starosti deployment v Azure App Service. Tento nástroj podporuje i další služby, jako jsou kontrola verzí, MSBuild a různé automatizační úlohy. Do Kudu je možné nahrávat kód ze tří různých zdrojů [47]:
 - Synchronizace obsahu z OneDrive a Dropboxu
 - Continuous deployment s automatickou synchronizací obsahu z GitHubu, BitBucketu či Visual Studio Team Services
 - Deployment s manuální synchronizací obsahu z lokálního Git repositáře
- **Web Deploy**: deploy kódu do App Service přímo z vývojového prostředí, např. Visual Studia. Oproti Kudu se tento postup liší tím, že build binárních souborů

²⁷ File Transfer Protocol

²⁸ FTP s podporou SSL a TLS

aplikace probíhá před deployem aplikace. Stejně jako u FTP, ani u této metody neposkytuje App Service žádné další služby [47].

4.4.1. Deploy z vývojového prostředí

Aplikaci je do Azure možné publikovat přímo z vývojového prostředí [47], což je v našem případě Visual Studio s Azure SDK²⁹. Je ovšem možné využít i jiná vývojová prostředí, např. Xcode, Eclipse nebo IntelliJ IDEA. Visual Studio podporuje všechny dříve zmíněné metody pro deployment (FTP, GIT, Web deploy).

Zaměříme se na metodu Web deploy, která je v porovnání s FTP rychlejší [48]. Toto řešení je ideální pro jednoho vývojáře, který si veškeré zdrojové kódy spravuje sám. Dále je výhodou, že celý životní cyklus aplikace od vývoje, debugování až po publikování do Azure je možné spravovat přímo z vývojového prostředí a není potřeba žádný jiný nástroj. Nevýhodou je, že v případě týmového projektu je nutné použít dodatečný nástroj pro správu zdrojových kódů.

Další výhody přináší Azure SDK ve Visual Studiu, což umožňuje spravovat aplikace na Azure přímo z vývojového prostředí. Je tedy možné aplikace vytvářet, editovat mazat či spouštět. Poskytuje také možnosti pro editování kódu a debugování přímo v Azure. Deployment je prováděn v režimu diff-only, což znamená, že jsou publikovány pouze nové a modifikované soubory.

4.4.2. Deploy s využitím FTP

Publikování aplikace je možné provést nahráním souborů do Azure přes FTP [47]. K tomuto účelu postačuje jakýkoliv FTP klient. Mezi výhody se řadí jednoduchost, bezpečnost při použití FTPS a to, že uživatel přesně ví, kam jsou soubory kopírovány. Nevýhodou je, že uživatel musí vědět, jak zkopírovat soubory do správného umístění v App Service. Není podporována správa verzí pro návrat k předchozí verzi v případě chyby. Většina FTP klientů nepodporuje diff-only kopírování a jsou kopírovány všechny soubory, což může způsobovat delší dobu publikování.

²⁹ Software Development Kit

4.4.3. Deploy pomocí synchronizace s cloudovou složkou

K publikování je možné využít metodu synchronizace souborů a složek do App Service z cloudového úložiště jako je OneDrive nebo Dropbox [47]. K tomuto účelu je využíván Kudu nástroj zmíněný dříve.

Výhodou tohoto řešení je jednoduchost deploymentu, jelikož OneDrive i Dropox poskytují své klienty, kteří se starají o synchronizaci změn z lokálního umístění do cloudového úložiště. Deployment je tak možné provést jedním kliknutím na Azure portále. Nevýhodou je, že není možné provádět automatický deployment, ale vždy je nutné provést ruční synchronizaci. Ani u této metody není podporována správa verzí pro návrat k předchozí verzi v případě chyby.

4.4.4. Nepřetržitý deployment z cloudového verzovacího nástroje

Pokud je pro vývoj využíván cloudový verzovací nástroj jako Visual Studio Team Services, GitHub nebo BitBucket, je možné App Service nakonfigurovat tak, aby se propojil s repositářem a deployment probíhal nepřetržitě [47].

Výhodou je podpora správy verzí pro návrat k předchozí verzi v případě chyby. Dále je možné provádět deployment různých vývojových verzí aplikace do různých slotů ve službě App Service. Jako nevýhoda se může jevit nutná znalost práce s verzovacími nástroji.

4.4.5. Nasazovací sloty

Od úrovně služby Standard a výše je možné využít deployment do různých nasazovacích slotů [49]. Nasazovací slot je vlastně běžící aplikace s unikátním hostname. Počet dostupných slotů se liší podle úrovně služby. Je možné tedy aplikaci například publikovat nejdříve do staging slotu a poté jen tento slot prohodit s produkčním slotem. Toto řešení je možné využít, pokud není nutné provést změny v databázi. Pokud by bylo potřeba změny v databázi provést, je nutné aplikaci po dobu provádění změn zastavit, protože by mohlo dojít k problémům během práce s aplikací.

4.4.6. Výběr řešení pro deployment

V současné době je řešen deployment aplikace tak, že pomocí nástroje MSBuild jsou vytvořeny balíčky specifické pro jednotlivé zákazníky. Tyto balíčky jsou následně komprimovány do ZIP souborů, které jsou nahrány na aplikační servery. Zde jsou

pomocí automatizačních skriptů soubory extrahovány a rozkopírovány do cílových umístění.

Pro úpravy databáze jsou vytvářeny skripty, které jsou spouštěny nástrojem sqlcmd v příkazové řádce.

Na základě požadavku zadavatele bude zvolena shodná metoda pro deployment aplikace jako u stávajícího řešení. S využitím nástroje MSBuild bude proveden build aplikace a výsledné balíčky budou přes FTP nahrány do Azure App Service. Pro deployment je nezbytné nejprve vytvořit přihlašovací údaje pro nasazení, protože Azure neumožňuje využití účtu, kterým je uživatel přihlášený do Azure. Poté je nutné získat hostitelské jméno pro FTP připojení k naší aplikaci. S využitím těchto přihlašovacích údajů se můžeme s využitím FTP klienta připojit k naší aplikaci a zkopírovat data.

Pro úpravy databáze bude využíván stejný postup, jako u současného řešení. Bude tedy používán nástroj příkazové řádky sqlcmd. Stačí pouze nahradit stávající připojovací řetězec novým připojovacím řetězcem s přístupovými údaji, které jsou dostupné na portále. Následně je možné spouštět skripty nad databází umístěnou v Azure.

4.5. Zálohování

4.5.1. Databáze

Pro zálohování databáze existuje několik možností, jak již bylo zmíněno v sekci 2.4.1. Pro naše řešení se z těchto možností jako nejvýhodnější jeví dlouhodobá záloha databáze. Uchování zálohy po dobu 7 dní pro úroveň Basic nebo 35 dní pro úroveň Standard a Premium je pro potřeby Školy OnLine málo.

Konfigurace dlouhodobé zálohy probíhá v nastavení SQL serveru na portále Azure. V sekci nastavení se nachází položka Dlouhodobé uchovávání zálohy, kde bylo zálohování nakonfigurováno. Aby dlouhodobé zálohování mohlo být zapnuto, bylo nejdříve potřeba vytvořit účet služby Trezor Recovery Services, který bude sloužit pro uchování těchto záloh. Po vytvoření tohoto účtu je možné vytvořit zásady pro zálohování, čímž se rozumí to, jak dlouho bude záloha databáze uchována. Doba zálohování může být definována v týdnech, měsících a letech.

V sekci 3.3.3 bylo zmíněno, že ve stávajícím řešení probíhá zálohování také do úložiště, které se nachází v jiné geografické lokalitě. Tento způsob zálohování je požadován i pro databázi v Azure, aby byl zajištěn nepřetržitý provoz v případě výpadku datacentra. Pro databázi tedy byla nastavena aktivní geografická replikace. Při konfiguraci na portále je možné vybrat doporučenou oblast, kam umístit sekundární databázi, nebo si vybrat jakoukoliv jinou. Pro vytvoření této sekundární databáze je v cílové oblasti nejdříve nutné vytvořit SQL server a nakonfigurovat přístupové údaje. Následně je sekundární databáze vytvořena a je možné v případě selhání vynutit převzetí služeb do geograficky replikované sekundární databáze. Během několika sekund se tato sekundární databáze povýší a stane se z ní nová primární databáze.

4.5.2. Úložiště

Služba Azure Storage v základu nabízí replikaci do geograficky redundantního úložiště s přístupem pro čtení, aby byla zajištěna spolehlivost a vysoká dostupnost. Při tomto nastavení jsou vytvořeny synchronní kopie dat v data centru a zároveň asynchronní kopie dat v dalším data centru, které je vzdáleno stovky kilometrů. Data mohou být čtena z jak z primárního, tak sekundárního umístění.

Azure Storage nenabízí přímou možnost pro zálohování, ale je možné použít Copy Blob API [50] a vytvořit skript pro zálohování blobů do jiného účtu úložiště. Nebo je možné využít například knihovnu Azure Blob Backup [51], nadefinovat pro ni pravidelný WebJob v Azure a provádět tak pravidelné zálohy.

4.5.3. Aplikace

Pro zálohování aplikace je nutné mít vytvořený účet úložiště a v něm kontejner pro zálohy. Nastavíme pravidelné zálohování, které se bude provádět každý den v noci. Dále je možné nadefinovat, jak dlouho se záloha má uchovávat. Pro naši aplikaci nastavíme uchovávání zálohy jeden den. Společně s aplikací nabízí Azure možnost i pro zálohování připojené databáze. V našem řešení tuto možnost nevyužijeme a zálohu databáze budeme provádět samostatně, jak bylo zmíněno v sekci 4.5.1.

4.6. Monitorování

Jedním z požadavků při migraci bylo monitorování služeb, aby byl zajištěn jejich řádný běh a dostupnost. Jak již bylo zmíněno v sekci 3.4.1., dostupnost služeb je

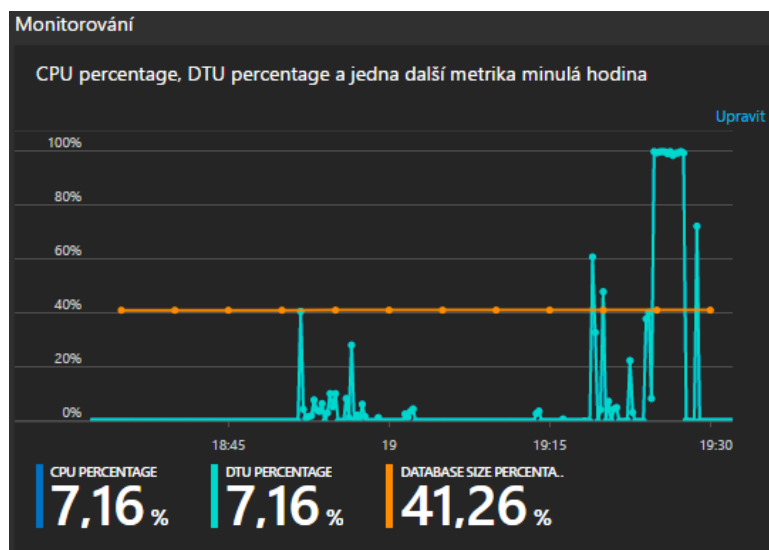
garantována smlouvou SLA a Microsoft služby neustále monitoruje, aby byly dostupné a v bezchybném stavu. Azure nabízí dostatek možností pro monitorování prostředků běžících v rámci jeho služeb. Nyní bude popsána realizace monitorování pro jednotlivé služby.

4.6.1. Databáze

V současné době jsou u databáze sledovány výsledky úloh spouštěných na SQL serveru a jejich výsledek. Dále je sledováno, zdali je možné spouštět dotazy v databázi, protože v minulosti se již stalo, že databáze byla dostupná, ale dotazy nebylo možné spouštět kvůli chybě v databázi. U databáze je také sledována její velikost.

V sekci 4.2.4 bylo popsáno, že úlohy SQL serveru byly nahrazeny runbooky spouštěnými ve službě Azure Automation. V této službě tedy budeme sledovat, zdali všechny spouštěné runbooky jsou úspěšně dokončeny se správným výsledkem. Pro ověření toho, jestli je možné spouštět dotazy v databázi, nadefinujeme runbook, který bude pravidelně spouštět databázový dotaz.

Na stránce přehledu služby Azure SQL Database je zobrazen monitorovací graf, jak bylo zmíněno v sekci 2.5.1. Pro databázi Školy OnLine nakonfigurujeme tento monitorovací graf pro zobrazení procentuálního využití CPU, DTU a velikosti databáze v závislosti na zvolené úrovni služby. Časový rozsah bude nastaven pro sledování metrik za poslední hodinu (viz obr. 4.3). Při otevření detailního zobrazení jsou zobrazeny podrobnější informace o metrikách jako průměr, minimum a maximum.



Obrázek 4.3: Monitorování databáze v Azure

Pro námi sledovanou metriku procentuálního využití DTU nastavíme pravidlo výstrahy. Pokud databáze bude využívat přes 80% dostupných DTU za posledních 5 minut, bude rozeslán e-mail správčům. Správci tak mohou reagovat a rozhodnout se, zdali přejít na vyšší výkonnostní úroveň služby. Správci se také na základě sledování výkonu databáze mohou rozhodnout, zdali nezvolit nižší úroveň služby, pokud například databáze využívá v průměru jen malou část dostupného výkonu. Dále vytvoříme pravidlo výstrahy pro velikost databáze. Když velikost databáze překročí hranici 80% dostupného prostoru za posledních 5 minut, opět dojde k rozeslání e-mailových výstrah správčům. Další sledované metriky a pravidla výstrah je možné kdykoliv přidat.

Veškeré diagnostické údaje budeme archivovat v účtu úložiště, který máme v Azure vytvořen. Je možné zvolit, jak dlouho mají tyto údaje v úložišti uchovány.

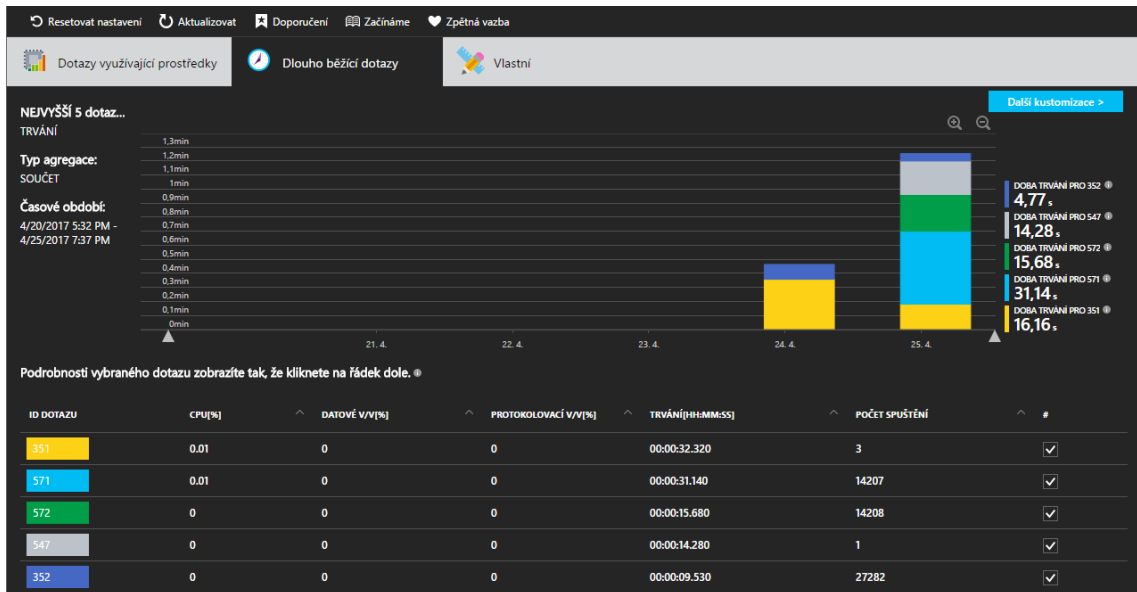
Doporučení navržená poradcem databáze budou aplikována pro optimalizaci výkonu databáze. Po migraci databáze do Azure poradce databázi poprvé zanalyzoval a nabídl doporučení na smazání duplicitních indexů. Tuto činnost je možné automatizovat, a proto pro databázi Školy OnLine povolíme akce automatického ladění pro tvorbu a odstraňování indexů.

Bylo zmíněno, že poradce také nabízí doporučení pro parametrizaci dotazů. Během testování nebylo nabídnuto žádné doporučení pro parametrizaci. Pokud ale v budoucnu bude parametrizace doporučena, tak ji využijeme. Když poté z reportu vyplyne, že parametrizace dotazu nevedla k optimalizaci výkonu, vrátíme se do předchozího stavu.

K monitoringu budeme používat také nástroj QPI. Aby bylo možné tento nástroj využít, je nutné mít pro databázi aktivní Query Store [29]. Aktivaci provedeme použitím T-SQL příkazu. Parametry funkce necháme na defaultních hodnotách. Pro naši databázi použijeme T-SQL příkaz, který vypadá následovně:

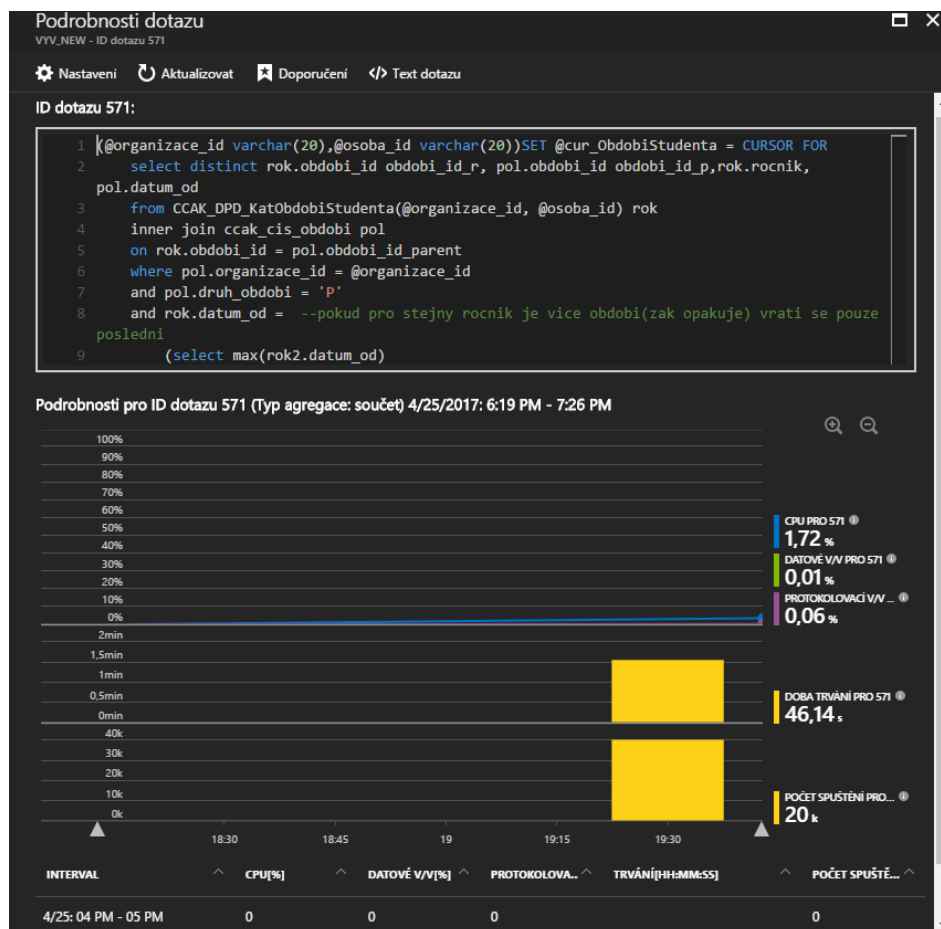
```
ALTER DATABASE skolaonline SET QUERY_STORE = ON;
```

S využitím nástroje QPI budeme sledovat dotazy, které využívají velké množství prostředků a dlouho běžící dotazy (viz obr. 4.4). Budeme zobrazovat vždy pět dotazů za posledních 24 hodin a pro data dotazů použijeme agregační funkci součtu.



Obrázek 4.4: Monitorování dlouho běžících dotazů

Vybrané dotazy s přiděleným ID můžeme dále analyzovat, zobrazit si jejich text a zjistit například jejich chování v čase (viz obr. 4.5).



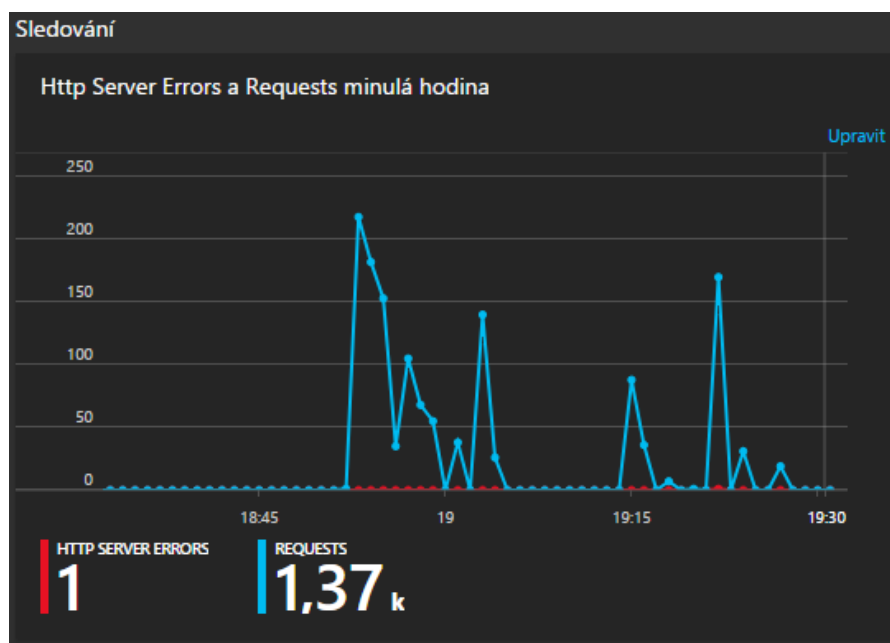
Obrázek 4.5: Podrobnosti dotazu

U monitorování databáze má zadavatel zájem o sledování a ladění exekučních plánů [52]. Pro tuto potřebu použijeme již zmíněnou funkci Query Store [29]. Sledování statistik budeme provádět v Management studiu, kde se u databáze nachází položka Query Store. Query Store nabízí čtyři panely pro sledování statistiky dotazů, celé databáze a je možné provádět případné úpravy exekuční plány dotazů pro zlepšení výkonu. Pro zobrazení všech dotazů, které se nachází v Query Store slouží panel Regressed Queries. V panelu Overall Resource Consumption budeme sledovat využití prostředků celou databází za určité období. Panel Top Resource Consuming Queries zobrazuje v základním nastavení 25 dotazů nejvíce využívajících prostředky. Můžeme sledovat také konkrétní dotazy, které se budou zobrazovat v panelu Tracked Queries.

4.6.2. Aplikace

Ve stávajícím řešení je u aplikačního serveru monitorováno využití CPU, paměti, místo na disku a dostupnost aplikace. Dále je testována dostupnost běžících služeb. U aplikace probíhá také sledování počtu požadavků a ověřování dostupnosti pomocí příkazu ping.

Na stránce přehledu webové aplikace ve službě App Service nakonfigurujeme monitorovací graf pro námi požadované metriky. Nadefinujeme časový rozsah na poslední hodinu a vybereme metriky pro počet požadavků a počet serverových chyb. (viz obr. 4.6).



Obrázek 4.6: Monitorování aplikace v Azure

Pro službu App Service budeme monitorovat využití CPU a paměti. Zároveň pro tyto metriky vytvoříme pravidla výstrah, aby byl kontaktován správce, pokud využití obou metrik překročí hranici 80% využití.

Pro aplikaci a webový server aktivujeme uchovávání diagnostických logů. Logy je možné ukládat přímo do souborového systému webového serveru nebo do účtu úložiště jako blob soubor. My zvolíme ukládání logů do souborového systému, kam k nim můžeme přistupovat přes FTP. Také je možné logy stahovat jako ZIP soubor pomocí Azure PowerShell nebo z Azure příkazové řádky. U webového serveru zapneme protokolování taktéž do souborového systému a budeme záznamy mazat po jednom týdnu. U protokolování webového serveru je ještě nutné nadefinovat limit pro velikost protokolů diagnostiky. Dále pro webový server aktivujeme uchovávání podrobných chybových zpráv a trasování chybových požadavků. U aplikace taktéž zapneme protokolování a definujeme uchování záznamů po dobu jednoho týdnu. Pro logy aplikace je možné zvolit úroveň logů. Na výběr je několik úrovní pro logování. Můžeme aktivovat logování chyb, upozornění, informací nebo všeho. My zvolíme logování pouze chyb.

U aplikace budeme taktéž monitorovat její dostupnost. K tomuto slouží test dostupnosti v rámci služby Application Insight [53]. Vytvoříme nový test dostupnosti, který s využitím příkazu ping bude ověřovat dostupnost URL adresy. V testu nakonfigurujeme URL adresu, jejíž dostupnost se má testovat, frekvenci testu, lokace odkud má test probíhat a očekávaný výsledek. U testu definujeme výstrahu, že pokud aplikace není dostupná, dojde k informování správce.

4.6.3. Redis Cache

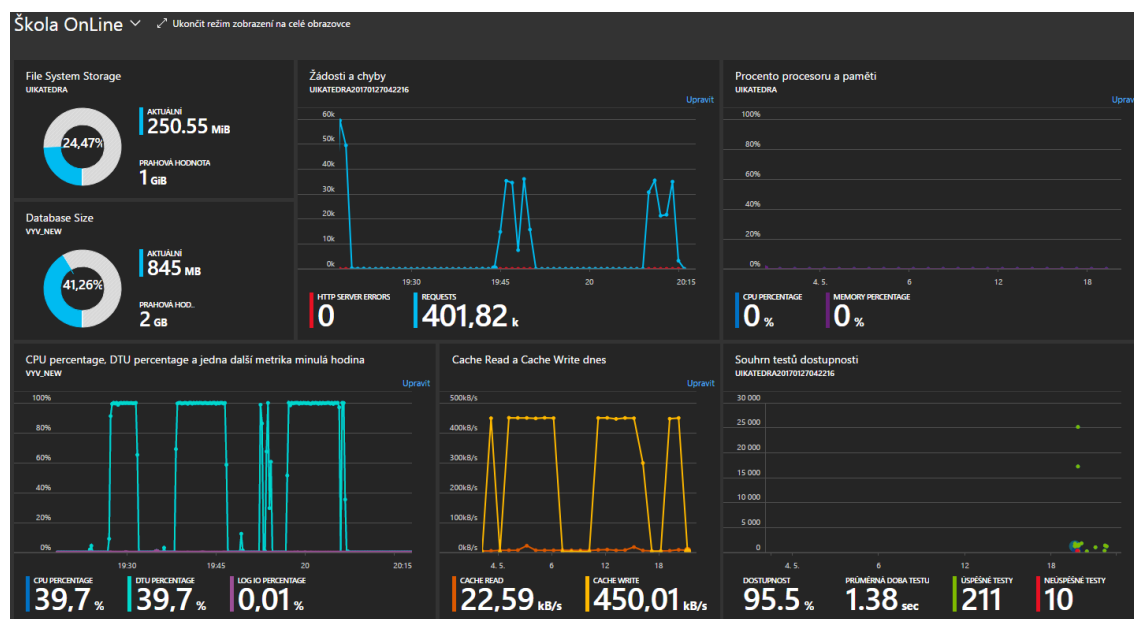
Další ze služeb, pro kterou také nastavíme monitorování je Redis Cache. Pro tuto službu na stránce přehledu nakonfigurujeme graf, který bude zobrazovat čtení z mezipaměti a zápisy do mezipaměti za posledních 24 hodin. Další graf bude monitorovat počet současně připojených klientů. Budeme také sledovat využití prostředků a to zatížení serveru Redis, využití CPU, celkovou velikost mezipaměti a šířku pásma. Pro všechny zmíněné metriky týkající se využití prostředků vytvoříme pravidla výstrah, aby při překročení 80% využití byl kontaktován správce. Aktivujeme také uchovávání diagnostických informací v účtu úložiště.

4.6.4. Úložiště

Poslední službou, kterou budeme monitorovat je úložiště. Pro něj na přehledu služby nastavíme pouze monitorovací graf zobrazující celkový počet požadavků za posledních 24 hodin.

4.6.5. Řídicí panel

Azure nabízí možnost vytvoření řídicího panelu a na tento panel připnout různé informace. Pro monitoring používaných služeb vytvoříme panel Škola OnLine (viz obr. 4.7). Bude zde zobrazena informace o aktuální velikosti databáze a její maximální možné velikosti. Dále graf zobrazující využití CPU, DTU a I/O operace. U služby App Service bude monitorováno využití CPU a paměti. U samotné aplikace pak počet dotazů a HTTP chyby. Pro službu Redis Cache je zobrazen graf, zobrazující rychlost zápisů do vyrovnávací paměti a rychlost čtení z vyrovnávací paměti. Pro aplikaci je také zobrazen graf zobrazující souhrn testů dostupnosti aplikace.



Obrázek 4.7: Řídicí panel Školy OnLine

4.7. Škálování

V této sekci popíšeme možnosti škálování služeb Azure SQL Database, App Service a Redis Cache, tak aby všechny služby měly dostatek prostředků pro bezproblémový běh aplikace.

4.7.1. Databáze

Po prvotním výběru plánu služby a úrovně výkonu je možné kdykoliv plán či výkonovou úroveň změnit. Na základě monitorování databáze lze okamžitě reagovat na nedostatek prostředků databáze. Tuto změnu lze provádět přímo na Azure portále, PowerShellovým skriptem nebo T-SQL příkazem. Pokud je změněn plán či výkonová úroveň, je vytvořena replika originální databáze s nově zvoleným plánem a poté co je replika vytvořena, dojde k přepnutí připojení na ni. Během tohoto procesu nedochází ke ztrátě dat.

Azure nabízí funkci DTU Calculator [54], která poskytuje doporučení, jakou cenovou úroveň služby a jaký výkon zvolit na základě analýzy současného řešení. K tomuto účelu slouží nástroj, který se spustí na současném databázovém serveru, kde bude měřit využití prostředků při běžném běhu aplikace. Report z tohoto měření se nahraje do kalkulačky a na základě výpočtu je doporučen cenový plán a počet potřebných DTU.

4.7.2. Aplikace

U služby App Service existuje možnost pro horizontální a vertikální škálování. Pod pojmem vertikální škálování rozumíme zvyšování výkonu serveru. V prostředí Azure to znamená změnu plánu služby na jinou cenovou úroveň pro získání většího výpočetního výkonu, paměti RAM, úložiště a dalších funkcí jako například podpora pro automatické škálování. Horizontální škálování je synonymem pro přidávání dalších instancí virtuálních serverů, na kterých běží aplikace. Počet dostupných instancí závisí na zvoleném plánu služby.

Pokud je prováděno škálování, tak je to otázkou několika vteřin a ovlivňuje to všechny aplikace, které se nachází ve službě App Service. Není nutné nijak zasahovat do kódu aplikace nebo provádět její opětovné nasazení.

Škálování lze provést na Azure portále buď na stránce služby App Service nebo přímo na stránce aplikace. Pro vertikální škálování zvolíme volbu Vertikálně navýšit kapacitu, kde je výběr z dostupných plánů a jejich cenových úrovní.

Možnosti pro horizontální škálování závisí na tom, jaký plán služby je používán. Plán Basic podporuje pouze ruční přidávání instancí a maximální počet jsou tři instance.

Pokud tedy nastavíme určitý počet instancí, tak jejich počet se nezmění, ani když se změní zátěž. Pro plány Standard a Premium lze kromě manuálního nastavení počtu instancí definovat parametry pro automatické škálování.

Na stránce služby pro horizontální škálování vybereme volbu Horizontálně navýšit kapacitu. Zde se nachází graf, který zobrazuje historii automatického škálování. Počet instancí lze měnit automaticky na základě sledované metriky. U App Service lze škálovat podle využití CPU. Definujeme, jaký minimální a maximální počet instancí chceme využívat a v jakých mezích chceme, aby se pohybovalo průměrné využití CPU pro všechny instance. Když například nadefinujeme využití CPU mezi 40% a 80%, tak v případě, že je překročena maximální hranice, dojde ke škálování nahoru. V opačném případě, když využití CPU klesne pod minimální hranici, bude provedeno škálování dolů. Oznámení o proběhlém automatickém škálování lze odeslat na e-mail správce. Další volbou je možnost vytvoření různých škálovacích profilů. Tyto profily mohou obsahovat sledování jiných metrik, než které jsou v základu nabízené. V profilech lze nastavit několik pravidel, například pro agresivnější škálování, kdy například při využití CPU větším, než 60% bude přidána jedna instance. Pokud však využití překročí hranici 80%, dojde k přidání tří instancí. Tyto profily škálování mohou být aplikovány pouze pro určité dny, takže například pro pracovní dny může být nastaveno agresivnější škálování, než pro víkend. Kromě toho lze nadefinovat i konkrétní dny pro škálovací profil, což se může hodit v obdobích zvýšené zátěže kolem období tvorby vysvědčení nebo konce školního roku.

Služba App Service nabízí funkci Testování výkonu pro tvorbu performance testů. U těchto testů umožňuje nastavit, kolik uživatelů souběžně bude k aplikaci přistupovat, z jaké oblasti budou přistupovat a jak dlouho má test trvat. Na základě výsledků těchto testů je možné reagovat na potřebu vyššího výkonu škálováním aplikace.

4.7.3. Redis Cache

Službu Redis Cache lze jednoduše vertikálně škálovat přímo na portále Azure výběrem libovolného plánu a jeho cenové úrovně. Pro změnu plánu se můžeme rozhodnout na základě sledovaných metrik mezipaměti. K automatizaci škálování mezipaměti je možné využít PowerShell nebo Azure příkazovou řádku.

4.8. Náklady na provoz

V analytické části již byla popsána cenová politika služeb, která má určitá specifika a liší se od tradičních modelů. Nebylo zabíháno do přílišných detailů, ale spíše zde byla snaha o to, přiblížit od čeho se odvíjí ceny služeb v Azure. Nebudeme počítat náklady na provoz aplikace, ale zmíníme nástroje, které k tomu lze použít. Pro odhad měsíčních cenových nákladů nabízí Microsoft Azure na svých stránkách cenovou kalkulačku [55]. K porovnání celkových nákladů na vlastnictví místní infrastruktury s porovnatelným nasazením Azure slouží kalkulačka TCO³⁰, pomocí které lze odhadnout úspory při přechodu do Azure. Tyto nástroje může zadavatel využít pro porovnání cen stávajícího řešení a provozu aplikace v Azure.

Testování provozu aplikace v Azure bylo omezeno předplatným Visual Studio Professional, které nabízí měsíční kredit 45 €, a tak nebylo možné dlouhodobější vyzkoušení vyšších plánů použitých služeb. Účelem práce nebylo navrhnout konkrétní plány služeb, které by měly být využity, avšak z provedené migrace vyplývá, které funkce musí být v plánech dostupné pro splnění požadavků. Až při reálné zátěži by bylo možné zjistit, jaké plány v jakých cenových a výkonových úrovních jsou pro provoz aplikace ideální.

³⁰ Total Cost of Ownership

5. Závěr

Větší část práce byla věnována důkladnému seznámení s cloudovou platformou Microsoft Azure. V první fázi byly nejprve popsány distribuční modely, které jsou v Azure nabízeny, společně s jejich výhodami a nevýhodami. Poté bylo nutné prostudovat informace o poskytovaných službách a jejich funkcích. Zejména se jednalo o služby, jejichž využití bylo zvažováno pro migraci webové aplikace.

Z analýzy stávající ASP.NET Web Forms aplikace využívající Microsoft SQL databázi vyplynula specifika, které bylo nutno při migraci aplikace a databáze do Azure vyřešit. Kromě toho byly zmíněny požadavky na spolehlivost a výkon aplikace společně s popisem toho, jak jim vyhovět. Jelikož aplikace pracuje s osobními údaji a také s citlivými osobními údaji, bylo nutné prostudovat legislativu České republiky, aby jejich zpracování probíhalo podle zákona.

V další fázi byl na základě porovnání distribučních modelů zvolen model PaaS a vybrány jednotlivé cloudové služby, jež měly být k migraci použity. S využitím závěrů plynoucích z provedené analýzy byla úspěšně realizována migrace vybrané části aplikace společně s databází do Microsoft Azure. Při migraci databáze byly objeveny problémy s kompatibilitou, které se podařilo zdárně vyřešit. Avšak databáze běžící na Microsoft SQL Serveru využívá určité funkce, které v Azure SQL Database nejsou poskytovány. Bylo potřeba nalézt ideální řešení a to následně realizovat. Aplikační část byla publikována do služby App Service a samotné publikování se obešlo bez závažnějších problémů. Ovšem pro provoz aplikace ve službě App Service si aplikace vyžádala určité změny, které vycházejí z vlastností cloudového řešení. Jednalo se například o změnu režimu pro udržení stavu relace, což bylo vyřešeno použitím mezipaměti Redis Cache. Do Azure byla migrována vybraná část aplikace, která pokrývá majoritní většinu funkcionalitu, ale je nutné počítat s rizikem, že při migraci zbylé části by se mohly vyskytnout jiné problémy.

Kromě samotné migrace bylo navrženo a realizováno optimální řešení pro zálohování a monitorování u použitých služeb. Dále byly z různých metod, které lze využít pro deployment databáze a aplikace vybrány ty, které se nejvíce přibližují stávající procesu nasazování.

Z práce vyplynulo, že provedení migrace je možné, avšak nelze aplikaci a databázi přenést do Azure jedna ku jedné. Dále práce ukázala, že nejen výběr služeb, ale také podrobné seznámení se službami je důležité pro úspěšnou realizaci. Platforma Azure a nabízené služby totiž mají určitá specifika, která je nutné brát v potaz. V důsledku toho vznikají nároky na seznámení se službami a úpravu aplikace pro potřeby cloudového řešení, ale z dlouhodobého hlediska dojde ke snížení nároků nutných pro správu vlastního HW díky zvolenému distribučnímu modelu.

Výsledek práce poskytl ucelený přehled o tom, jak provést migraci aplikace Škola OnLine do prostředí Microsoft Azure a s jakými úskalími je nutno počítat. Práce pokládá základ pro možný budoucí přechod celé aplikace do cloudového prostředí.

Seznam obrázků

Obrázek 2.1: Distribuční modely [9]	3
Obrázek 2.2: Azure Virtual Machines [17]	6
Obrázek 2.3: Možnosti pro SQL databáze [18]	7
Obrázek 3.1: Síťová architektura	22
Obrázek 4.1: Porovnání Azure služeb [38].....	27
Obrázek 4.2: Postup migrace do Azure SQL databáze [40]	32
Obrázek 4.3: Monitorování databáze v Azure	42
Obrázek 4.4: Monitorování dlouho běžících dotazů.....	44
Obrázek 4.5: Podrobnosti dotazu.....	44
Obrázek 4.6: Monitorování aplikace v Azure.....	45
Obrázek 4.7: Řídící panel Školy OnLine.....	47

Přehled použitých zkratk a značení

API	Application Programming Interface
ASP.NET	Active Server Pages .NET
BLOB	Binary Large Object
CPU	Central Processing Unit
CSV	Comma-Separated Values
DDoS	Distributed Denial of Service
DNS	Domain Name System
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
NoSQL	Not SQL
RAID	Redundant Array of Independent Disks
REST	Representational State Transfer
SDK	Software Development Kit
SLA	Service Level Agreement
SMB	Server Message Block
SMS	Short Message Service
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCO	Total Cost of Ownership
T-SQL	Transact-SQL
URL	Uniform Resource Locator

Literatura

1. Co je Azure. *Microsoft Azure*. [Online] [Citace: 12. 11 2016.] <https://azure.microsoft.com/cs-cz/overview/what-is-azure/>.
2. Public Cloud. *Search for Cloud Computing*. [Online] [Citace: 12. 11 2016.] <http://searchcloudcomputing.techtarget.com/definition/public-cloud>.
3. Martin, Steven. Upcoming Name Change for Windows Azure. *Microsoft Azure*. [Online] [Citace: 12. 11 2016.] <https://azure.microsoft.com/en-us/blog/upcoming-name-change-for-windows-azure/>.
4. Magic Quadrant. *Microsoft Azure*. [Online] [Citace: 12. 11 2016.] <https://azure.microsoft.com/cs-cz/campaigns/magic-quadrant/>.
5. Weins, Kim. Cloud Computing Trends: 2016 State of the Cloud Survey. *RightScale*. [Online] [Citace: 4. 12 2016.] <http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2016-state-cloud-survey#awsleads>.
6. Oblasti Azure. *Microsoft Azure*. [Online] [Citace: 4. 12 2016.] <https://azure.microsoft.com/cs-cz/regions/>.
7. On-Premises Software. *Technopedia*. [Online] [Citace: 24. 11 2016.] <https://www.techopedia.com/definition/26714/on-premises-software>.
8. Slashdot effect. *Wikipedia*. [Online] [Citace: 15. 11 2016.] https://en.wikipedia.org/wiki/Slashdot_effect.
9. Cloud Services. *Wilde Technology*. [Online] [Citace: 4. 12 2016.] <http://www.wildetech.com.au/services/cloud-services/>.
10. Co je IaaS? *Microsoft Azure*. [Online] [Citace: 4. 12 2016.] <https://azure.microsoft.com/cs-cz/overview/what-is-iaas/>.
11. GE Healthcare Delivers Core Customer Solutions on the Microsoft Cloud. *Microsoft Customer Stories*. [Online] [Citace: 4. 12 2016.] <https://customers.microsoft.com/en-us/story/ge-healthcare-delivers-core-customer-solutions-on-the>.
12. Co je PaaS? *Microsoft Azure*. [Online] [Citace: 4. 12 2016.] <https://azure.microsoft.com/cs-cz/overview/what-is-paas/>.
13. Canada's largest automobile search site takes Azure for a spin, drives business success. *Microsoft Customer Stories*. [Online] [Citace: 4. 12 2016.] <https://customers.microsoft.com/en-US/story/canadas-largest-automobile-search-site-takes-azure-for>.
14. Co je SaaS? *Microsoft Azure*. [Online] [Citace: 4. 12 2016.] <https://azure.microsoft.com/cs-cz/overview/what-is-saas/>.
15. Global Airline Uses Enterprise Social and Online Communication Tools to Improve Service. *Microsoft Customer Stories*. [Online] [Citace: 4. 12 2016.]

<https://customers.microsoft.com/en-US/story/global-airline-uses-enterprise-social-and-online-commu>.

16. Virtual Machines. *Microsoft Azure*. [Online] [Citace: 4. 12 2016.] <https://azure.microsoft.com/cs-cz/services/virtual-machines/>.

17. Introducing Microsoft Azure. *Microsoft Azure*. [Online] [Citace: 4. 12 2016.] <https://docs.microsoft.com/en-us/azure/fundamentals-introduction-to-azure>.

18. Choose a cloud SQL Server option: Azure SQL (PaaS) Database or SQL Server on Azure VMs (IaaS). *Microsoft Azure*. [Online] [Citace: 4. 12 2016.] <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas>.

19. Azure Automation Overview. *Microsoft Azure*. [Online] [Citace: 14. 02 2017.] <https://docs.microsoft.com/en-us/azure/automation/automation-intro>.

20. Introduction to Microsoft Azure Storage. *Microsoft Azure*. [Online] [Citace: 14. 02 2017.] <https://docs.microsoft.com/en-us/azure/storage/storage-introduction>.

21. Explaining Database Transaction Units (DTUs) and elastic Database Transaction Units (eDTUs). *Microsoft Azure*. [Online] [Citace: 12. 03 2017.] <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-what-is-a-dtu>.

22. SQL Database - ceny. *Microsoft Azure*. [Online] [Citace: 10. 1 2017.] <https://azure.microsoft.com/cs-cz/pricing/details/sql-database/>.

23. Podpora Azure pro zákazníky. *Microsoft Azure*. [Online] [Citace: 23. 1 2017.] <https://azure.microsoft.com/cs-cz/support/plans/>.

24. Rozsah podpory a rychlost odezvy. *Microsoft Azure*. [Online] [Citace: 23. 1 2017.] <https://azure.microsoft.com/cs-cz/support/plans/response/>.

25. Back up your App in Azure. *Microsoft Azure*. [Online] [Citace: 23. 04 2017.] <https://docs.microsoft.com/en-us/azure/app-service-web/web-sites-backup>.

26. Log Analytics. *Microsoft Azure*. [Online] [Citace: 24. 04 2017.] <https://azure.microsoft.com/cs-cz/services/log-analytics/>.

27. SQL Database Advisor. *Microsoft Azure*. [Online] [Citace: 25. 04 2017.] <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-advisor>.

28. Azure SQL Database Query Performance Insight. *Microsoft Azure*. [Online] [Citace: 25. 04 2017.] <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-query-performance>.

29. Monitoring Performance By Using the Query Store. *Microsoft*. [Online] [Citace: 01. 04 2017.] <https://docs.microsoft.com/en-us/sql/relational-databases/performance/monitoring-performance-by-using-the-query-store>.

30. Monitoring Azure SQL Database using dynamic management views. *Microsoft Azure*. [Online] [Citace: 01. 05 2017.] <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-monitoring-with-dmvs>.
31. What is Application Insights? *Microsoft Azure*. [Online] [Citace: 01. 05 2017.] <https://docs.microsoft.com/en-us/azure/application-insights/app-insights-overview>.
32. ISO/IEC 27018 Code of Practice for Protecting Personal Data in the Cloud. *Microsoft Trust Center*. [Online] [Citace: 14. 12 2016.] <https://www.microsoft.com/en-us/trustcenter/Compliance/ISO-IEC-27018>.
33. Lze využít cloud computing pro zpracování osobních údajů? *Úřad pro ochranu osobních údajů*. [Online] [Citace: 15. 12 2016.] <https://www.uouu.cz/lze-vyuzit-cloud-computing-pro-zpracovani-osobnich-udaju/d-11337>.
34. Přehled případů předávání osobních údajů do zahraničí, u nichž není nutné žádat Úřad o povolení. *Úřad pro ochranu osobních údajů*. [Online] [Citace: 15. 12 2016.] <https://www.uouu.cz/prehled-pripadu-predavani-osobnich-udaju-do-zahranici-u-nichz-neni-nutne-zadat-urad-o-povoleni/ds-1649/p1=1649>.
35. Berger, Lukáš. *Nástroje IT manažera*. Praha : Forum, 2006-. ISSN 1802-3886.
36. Datová centra. *České Radiokomunikace*. [Online] [Citace: 05. 03 2017.] <https://www.cra.cz/datova-centra>.
37. SLA for Virtual Machines. *Microsoft Azure*. [Online] [Citace: 02. 03 2017.] https://azure.microsoft.com/en-us/support/legal/sla/virtual-machines/v1_6/.
38. SLA for App Service. *Microsoft Azure*. [Online] [Citace: 02. 03 2017.] https://azure.microsoft.com/en-us/support/legal/sla/app-service/v1_4/.
39. SLA for SQL Database. *Microsoft Azure*. [Online] [Citace: 02. 03 2017.] https://azure.microsoft.com/en-us/support/legal/sla/sql-database/v1_1/.
40. SLA for Managed Cache Service. *Microsoft Azure*. [Online] [Citace: 02. 03 2017.] https://azure.microsoft.com/en-us/support/legal/sla/cache/v1_0/.
41. Rozdíly u Transact-SQL ve službě Azure SQL Database. *Microsoft Azure*. [Online] [Citace: 13. 03 2017.] <https://docs.microsoft.com/cs-cz/azure/sql-database/sql-database-transact-sql-information>.
42. Badola, Vineet. Microsoft Azure App Service, Cloud Services, or VMs? *Cloud Academy*. [Online] 20. 02 2017. <http://cloudacademy.com/blog/microsoft-azure-app-service-virtual-machines/>.
43. RAISERROR-Transact-SQL. *Microsoft Developer Network*. [Online] [Citace: 20. 03 2017.] <https://msdn.microsoft.com/en-us/library/ms178592.aspx>.
44. SQL Server database migration to SQL Database in the cloud. *Microsoft Azure*. [Online] [Citace: 27. 03 2017.] <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-cloud-migrate>.

45. Map a custom domain name to an Azure app. *Microsoft Azure*. [Online] [Citace: 09. 05 2017.] <https://docs.microsoft.com/en-us/azure/app-service-web/web-sites-custom-domain-name#createdns>.
46. What is a Wildcard SSL certificate? *Comodo*. [Online] [Citace: 09. 05 2017.] <https://ssl.comodo.com/wildcard-ssl-certificates.php>.
47. Deploy your app to Azure App Service. *Microsoft Azure*. [Online] [Citace: 10. 04 2017.] <https://docs.microsoft.com/en-us/azure/app-service-web/web-sites-deploy>.
48. Introduction to Web Deploy. *IIS*. [Online] [Citace: 12. 04 2017.] <https://www.iis.net/learn/publish/using-web-deploy/introduction-to-web-deploy>.
49. Set up staging environments in Azure App Service. *Microsoft Azure*. [Online] [Citace: 23. 04 2017.] <https://docs.microsoft.com/en-us/azure/app-service-web/web-sites-staged-publishing>.
50. Copy Blob. *Microsoft Azure*. [Online] [Citace: 23. 04 2017.] <https://docs.microsoft.com/en-us/rest/api/storageservices/fileservices/Copy-Blob?redirectedfrom=MSDN>.
51. GitHub. *Azure Blob Backup*. [Online] [Citace: 23. 04 2017.] <https://github.com/levibotelho/azure-blob-backup>.
52. Execution Plan Basics. *Simple Talk*. [Online] [Citace: 01. 05 2017.] <https://www.simple-talk.com/sql/performance/execution-plan-basics/>.
53. Monitor availability and responsiveness of any web site. *Microsoft Azure*. [Online] [Citace: 07. 05 2017.] <https://docs.microsoft.com/en-us/azure/application-insights/app-insights-monitor-web-app-availability>.
54. *Azure SQL Database DTU Calculator*. [Online] [Citace: 07. 05 2017.] <http://dtucalculator.azurewebsites.net/>.
55. Cenová kalkulačka. *Microsoft Azure*. [Online] [Citace: 10. 05 2017.] <https://azure.microsoft.com/cs-cz/pricing/calculator/>.

Příloha A: Návod pro migraci do Microsoft Azure

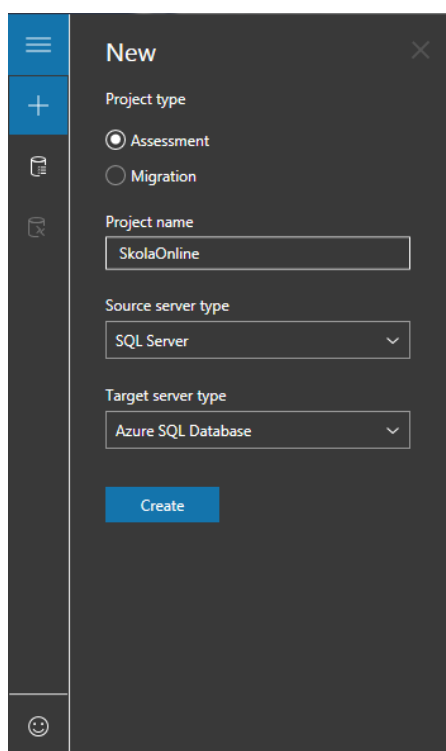
Ověření kompatibility databáze

Předpokladem je běžící SQL server s databází určenou k exportu. V Management studiu se přihlaste k SQL serveru, poté klikněte pravým tlačítkem na Databases a vyberte volbu Restore Database. Jako zdroj zvolte zálohu databáze, která se nachází na přiloženém DVD. Potvrďte a dojde k nahrání databáze do SQL serveru.

Nejprve je nutné provést ověření kompatibility stávající databáze s Azure SQL Database. K tomu slouží nástroj Data Migration Assistant, který je možné stáhnout z webových stránek společnosti Microsoft.

Po spuštění nástroje následujte tyto kroky:

1. Klikněte na tlačítko New
2. Vyplnit následující údaje
 - Project type: Assessment
 - Project Name: SkolaOnline
 - Source server type: SQL Server
 - Target server type: Azure SQL Database
3. Klikněte tlačítko Create

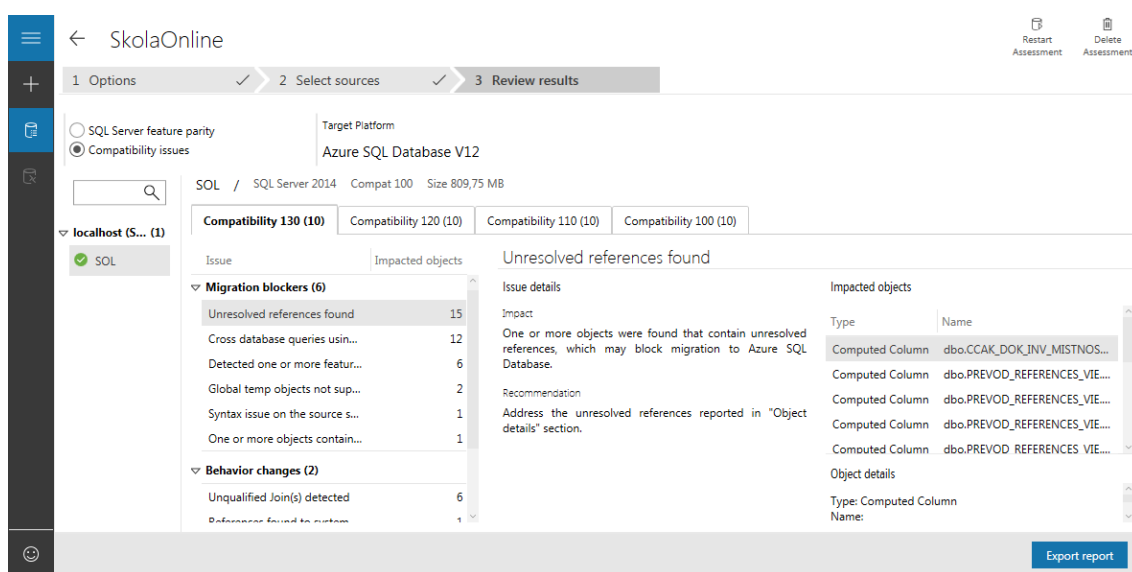


Welcome to Data Migration Assistant



4. Na další obrazovce vybrat Target version Azure SQL Database V12. Ostatní volby nechte zaškrtnuté a klikněte na tlačítko Next.
5. Nyní zvolte Server name a Authentication type a stiskněte tlačítko Connect.
6. Po připojení k SQL serveru vyberte zdrojovou databázi.
7. Po přidání databáze stiskněte tlačítko Start Assessment.

Po dokončení testu kompatibility jsou vypsány nalezené problémy. Tyto problémy jsou blíže popsány v textu diplomové práce v kapitole 4.2.2. Opravu problémů je možné provést v SQL Management studiu spuštěním opravných skriptů. Po opravě všech problémů je možné databázi přesunout do Azure.



Export databáze z SQL serveru

Databázi připravenou na migraci je potřeba exportovat ze současného SQL serveru. K tomuto účelu bude použit nástroj SQLPackage.exe, který je dodáván společně s SQL Server Management Studiem. Nástroj se spouští z příkazové řádky. Obvykle je umístěn v následující složce: C:\Program Files (x86)\Microsoft SQL Server\130\DAC\bin. Příkaz pro export databáze do souboru formátu BACPAC vypadá takto:

```
sqlpackage.exe /Action:Export /ssn:<server_name> /sdn:<db_name> /tf:<target_file>
```

Jednotlivé parametry jsou:

- Action – druh akce, v našem případě Export
- ssn – adresa SQL serveru

- sdn – název databáze
- tf – umístění a název exportovaného souboru, např C:\exportDB.BACPAC

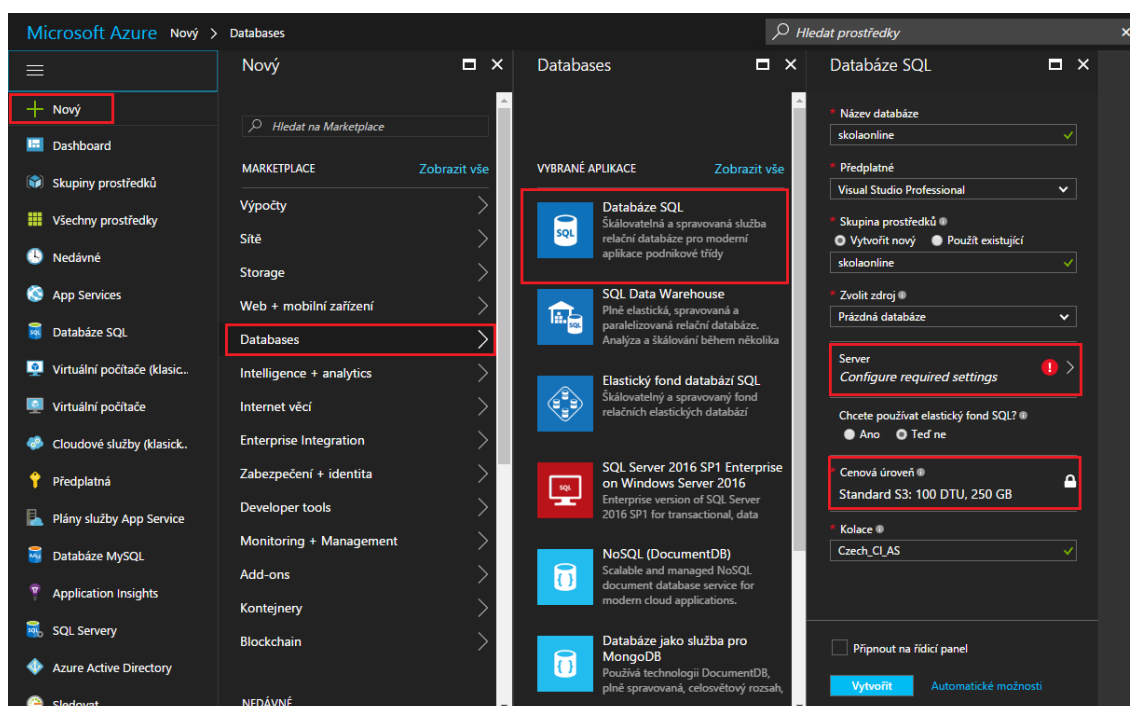
Migrace databáze do Azure

Podmínkou je existující předplatné ve službě Azure.

Nejprve je nutné vytvořit účet Azure SQL Database. Během vytváření databáze bude vytvořena služba Azure SQL Database logický server a dále bude vytvořena skupina prostředků. Skupina prostředků je logický kontejner, ve kterém se nasazují a spravují prostředky Azure, jako například webové aplikace, databáze a účty úložiště. Vytvoření je možné provést na portále Azure, pomocí příkazové řádky Azure nebo pomocí Powershell skriptů.

Po přihlášení do portálu Azure následujte tyto kroky pro vytvoření databáze:

1. Klikněte na tlačítko Nový v levém horním rohu na Azure portále
2. Vyberte Databases na stránce Nový a poté vyberte Databáze SQL na stránce Databases



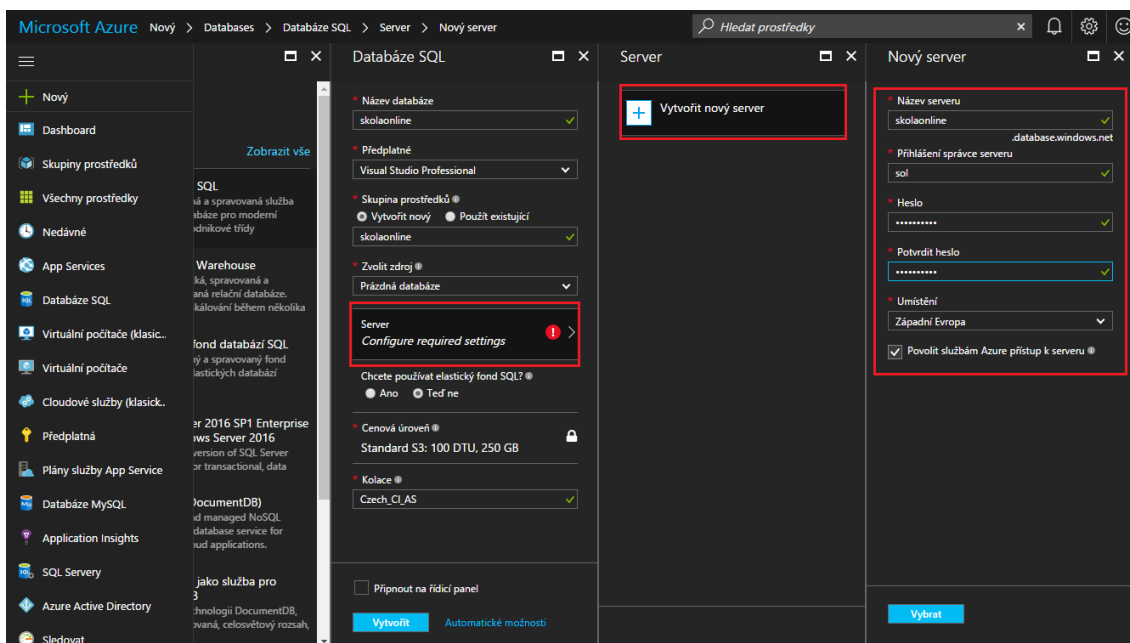
3. Vyplňte parametry následujícími hodnotami, stejně jak je uvedeno na obrázku:

- Název databáze: skolaonline
- Skupina prostředků: skolaonline

- Kolace: Czech_CI_AS

4. Klikněte na Server pro vytvoření a nastavení SQL serveru pro novou databázi. Vyplňte globálně unikátní jméno serveru, přihlašovací údaje správce serveru a umístění serveru.

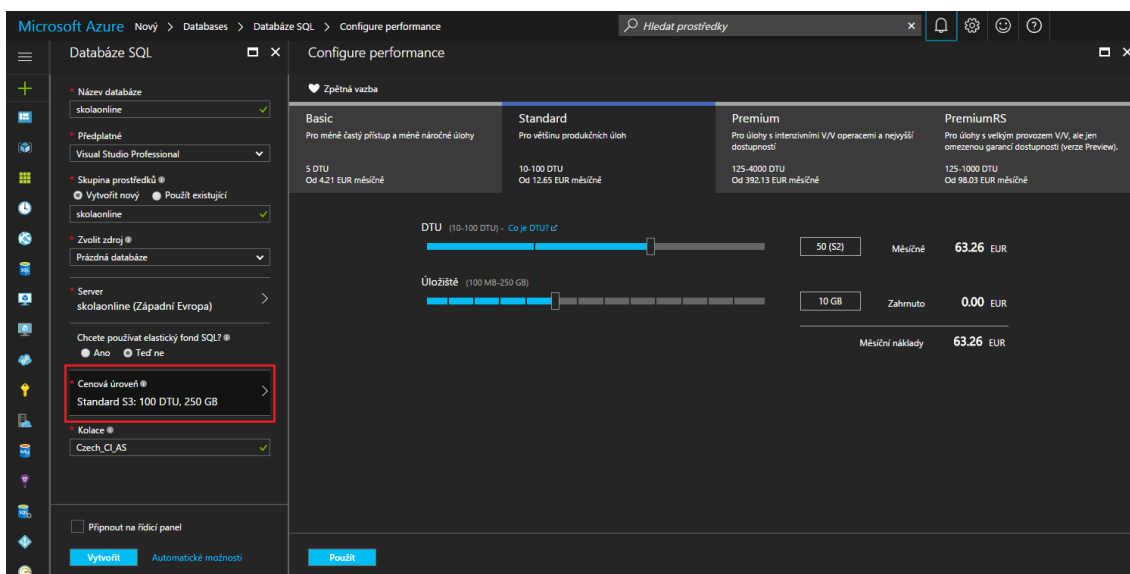
5. Klikněte na tlačítko Vybrat



6. Klikněte na Cenová úroveň pro výběr úrovně služby a výkonu pro nově vytvořenou databázi.

7. Zde nadefinujte požadované DTU a velikost úložiště.

8. Po nastavení klikněte na tlačítko Použít



9. Nyní pro vytvoření databáze klikněte na tlačítko Vytvořit. Vytváření potrvá několik minut a o průběhu budete informováni v panelu Oznámení.

Po úspěšném vytvoření databáze je nutné provést import databáze. K tomuto účelu použijte nástroj SQLPackage.exe s následujícími parametry:

```
sqlpackage.exe /Action:Import /tsn:sol.database.windows.net  
/tdn:sol /tu:user /tp:password /sf:c:\sql\SOL-DB.bacpac
```

Jednotlivé parametry jsou:

- Action – druh akce, v našem případě Import
- tsn – adresa Azure SQL serveru
- tdn – název databáze
- tu – jméno uživatele SQL serveru
- tp – heslo uživatele
- sf – umístění a název importovaného souboru, např C:\exportDB.BACPAC

Po dokončení importu je databáze dostupná na portále.

The screenshot displays the Microsoft Azure portal interface for an Azure SQL database. The breadcrumb navigation shows the path: Skupiny prostředků > SOL2 > skolaonline > skolaonlineTest. The main content area is divided into several sections:

- Navigation and Actions:** A top bar contains a search box and several action buttons: 'Nástroje' (highlighted with a red box), 'Kopírovat', 'Obnovit', 'Exportovat', 'Nastavit bránu firewall serveru', and 'Odstranit'.
- Základní údaje (Basic Information):** This section provides details about the database instance:
 - Skupina prostředků: SOL2
 - Název serveru: skolaonline.database.windows.net
 - Stav: Online
 - Připojovací řetězec: Zobrazit databázové připojovací řetězce
 - Umístění: West Europe
 - Cenová úroveň: Basic (5 DTUs)
 - Název předplatného: Visual Studio Professional
 - Role geografické replikace: Nenakonfigurované
 - ID předplatného: cc861476-1ee0-4bb0-9a9f-94f8c644e2cc
- Monitorování (Monitoring):** This section includes a 'Využití prostředků' (Resource Usage) chart showing DTU percentage over time. The current DTU usage is 8.63%.

Kliknutím na tlačítko Nástroje můžeme spustit Editor dotazů. Po přihlášení k SQL serveru je možné ověřit naplnění databáze příkazem Select nad nějakou tabulkou.

Editor dotazů (preview)

Spustit Přikázání Otevřít dotaz Uložit dotaz Zpětná vazba

Ověřeno jako Jhula

```
1 select * from dbo.CCAK_AKTUALITY
```

Výsledky Zprávy

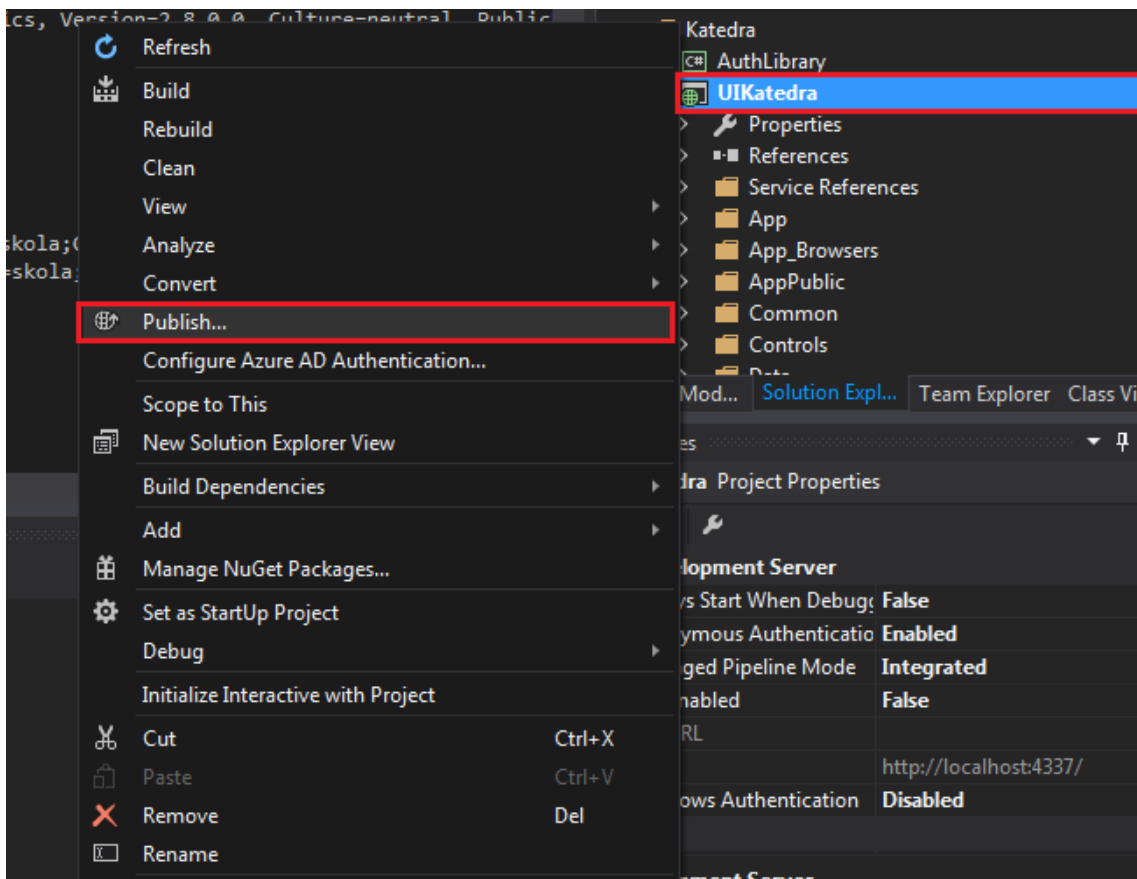
Filtrovat položky hledáním...

ORGANIZACE_ID	AKTUALITY_ID	NADPIS	TEXT	DETAIL	PRIZNAK_ZOBRAZIT	AUTOR
117	10	dsdsds	sssss		False	
117	11	blab la mkgjkj oip o pioio pi op	dfsdfsdfsdf		False	dfdf
117	12	text	textds dsd sds sd		True	
117	17	BIK - jedina	BIK - muj text	<p>gřřgčbbb...	True	BIK
117	3	První aktualita	dasjrtthe rjfoewrf éřěš cré řčf+ěš...		True	Já

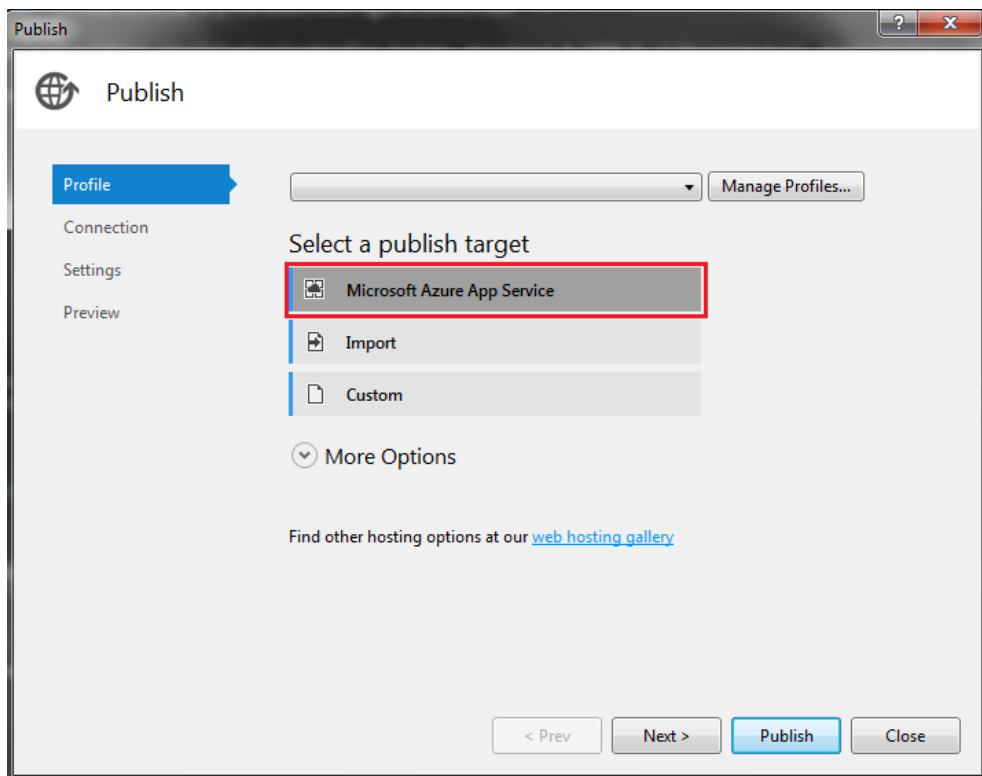
Migrace aplikace do Azure

V této části bude popsán postup pro publikování aplikace do Azure App Service z vývojového prostředí Visual Studio metodou Web Deploy.

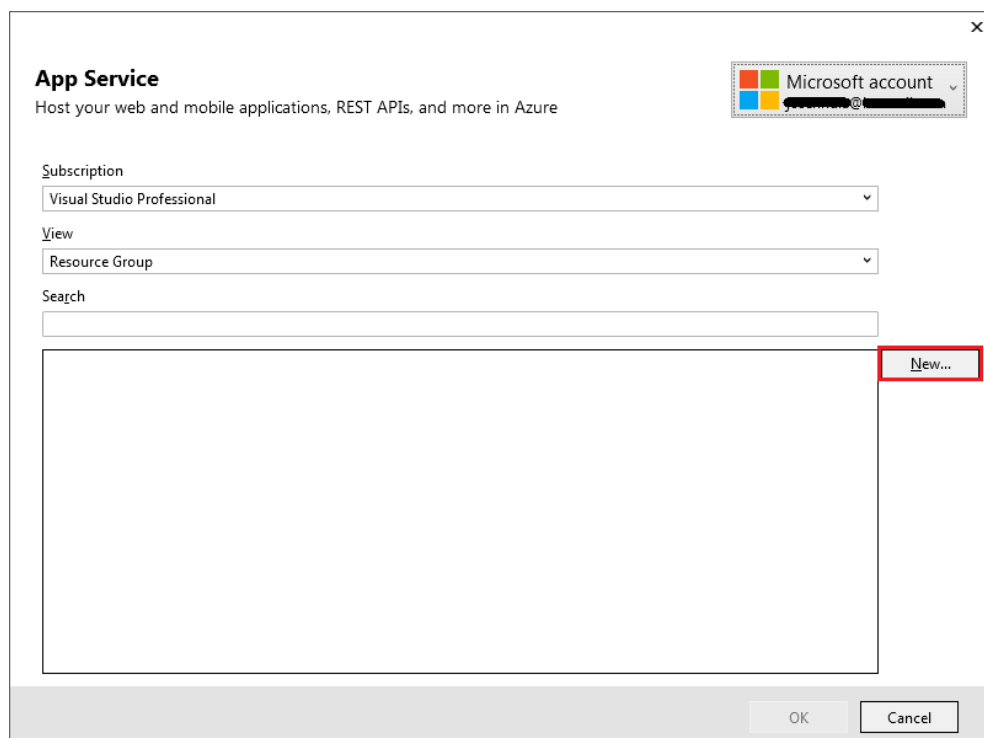
Po otevření stávajícího řešení ve Visual Studiu klikněte v průzkumníkoví řešení na projekt UIKatedra a vyberte možnost Publish.



Vyberte volbu Microsoft Azure App Service. Bude otevřené nové okno k vytvoření účtu služby App Service, kde bude aplikace hostována.



Pokud jste přihlášení k vašemu Azure účtu, v pravém horním rohu bude zobrazen váš účet. Klidněte na New pro vytvoření nového účtu služby App Service.



V nově otevřeném okně zvolte název webové aplikace, které musí být jedinečný. Název aplikace bude použit jako výchozí DNS název pro webovou aplikaci (<název_>.azurewebsites.net) a proto musí jedinečný mezi všemi aplikacemi v Azure. Zkontrolujte, zdali je vybráno správné předplatné a u položky Resource group vyberte skupinu prostředků skolaonline, která byla vytvořena při migraci databáze. Je nutné vytvořit plán služby App Service, a to proved'te kliknutím na tlačítko New.

Create App Service
Host your web and mobile applications, REST APIs, and more in Azure

Microsoft account
josef.hula@hotmail.com

Hosting ⓘ
Services

Web App Name Change Type ▾
skolaonline

Subscription
Visual Studio Professional ▾

Resource Group
skolaonline ▾ New... ⓘ

App Service Plan
UIKatedra (D1, West Europe) ▾ New...

Clicking the Create button will create the following Azure resources
[Explore additional Azure services](#)
App Service - skolaonline

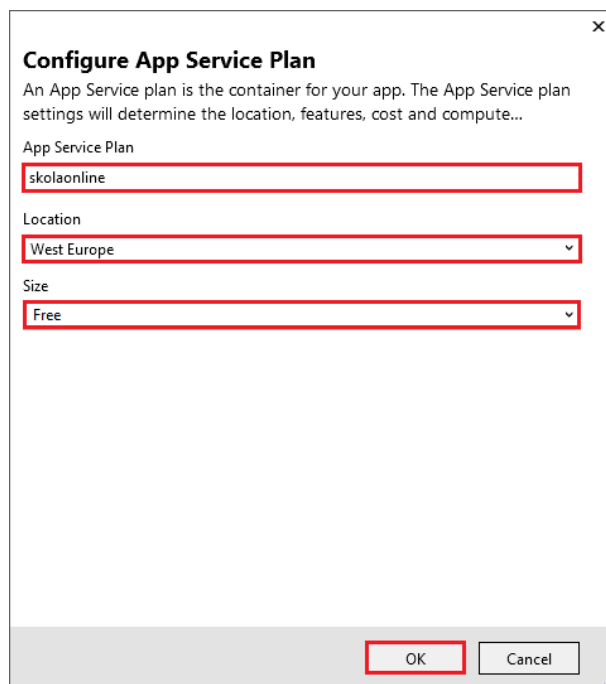
If you have removed your spending limit or you are using Pay as You Go, there may be monetary impact if you provision additional resources.
[Learn More](#)

Export... Create Cancel

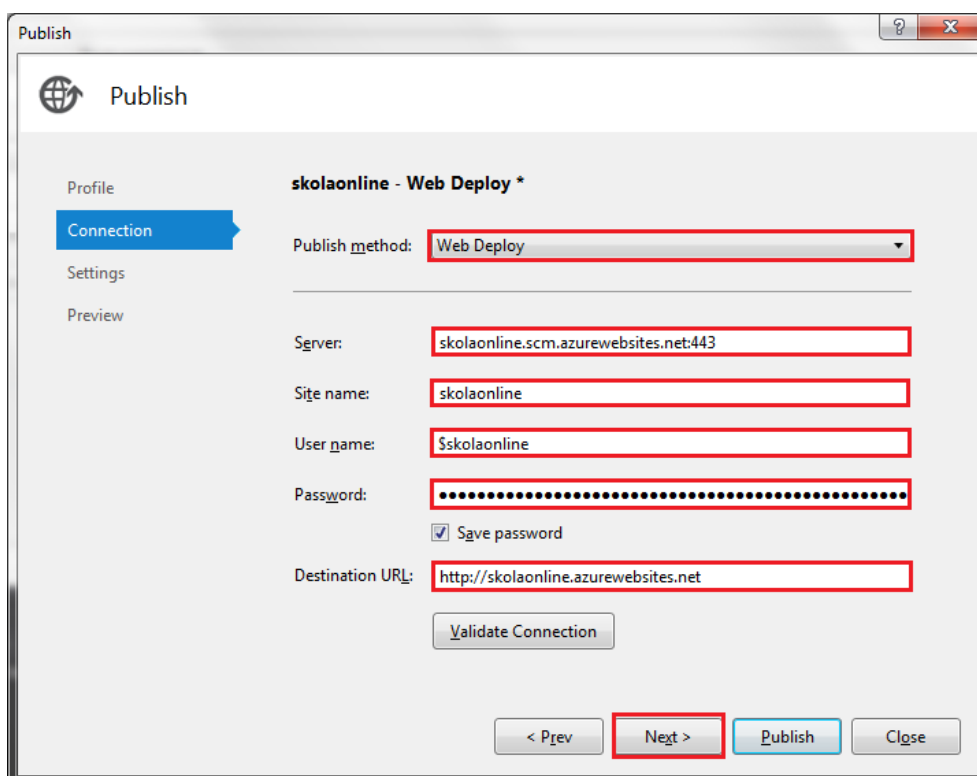
Nyní nakonfigurujte v dialogovém okně plán služby App Service. Nastavte plán takto:

1. App Service Plan: skolaonline
2. Location: West Europe nebo kteroukoliv jinou oblast
3. Size: Free nebo kteroukoliv jinou cenovou úroveň

Klikněte na tlačítko OK.



V okně je zobrazen souhrn, které Azure služby budou vytvořeny. Klikněte na tlačítko Create pro vytvoření služeb. Po vytvoření služeb se opět otevře okno pro publikování na záložce Connection, kde vybere metodu publikování Web Deploy. Je možné změnit název serveru, jméno stránky, jméno uživatele, heslo uživatele a URL. Můžete vyzkoušet připojení k URL kliknutím na tlačítko Validate Connection. Klikněte na tlačítko Next pro další krok publikování.



Je otevřena záložka Settings, kde je možné vybrat konfiguraci cílové platformy pro publikování a připojení k databázi. Konfiguraci platformy ponechte předdefinovanou. Nastavte připojení databáze kliknutím na tlačítko ...

V dialogovém okně nadefinujte připojení k SQL serveru a vyberte databázi. Nastavte parametry tak, jak byly nadefinovány při vytváření Azure SQL serveru:

1. Server name: skolaonline.database.windows.net
2. Authentication: SQL Server Authentication
3. User name: přihlašovací jméno uživatele
4. Password: heslo uživatele

Vyberte ze seznamu dostupných databází na SQL serveru vytvořenou databázi skolaonline. Můžete otestovat připojení k databázi tlačítkem Test Connection. Pokud je připojení úspěšné, klikněte na OK.

Destination Connection String

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient) Change...

Server name: skolaonline.database.windows.net Refresh

Log on to the server

Authentication: SQL Server Authentication

User name: sol

Password: ●●●●●●●● Save my password

Connect to a database

Select or enter a database name: skolaonline

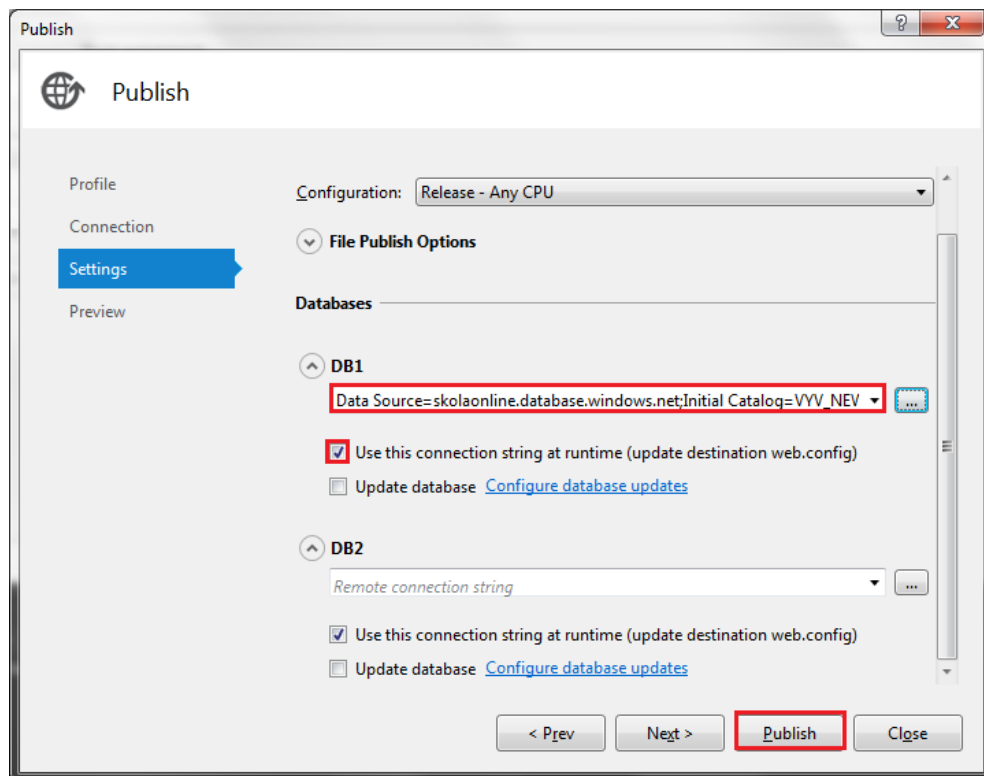
Attach a database file: Browse...

Logical name:

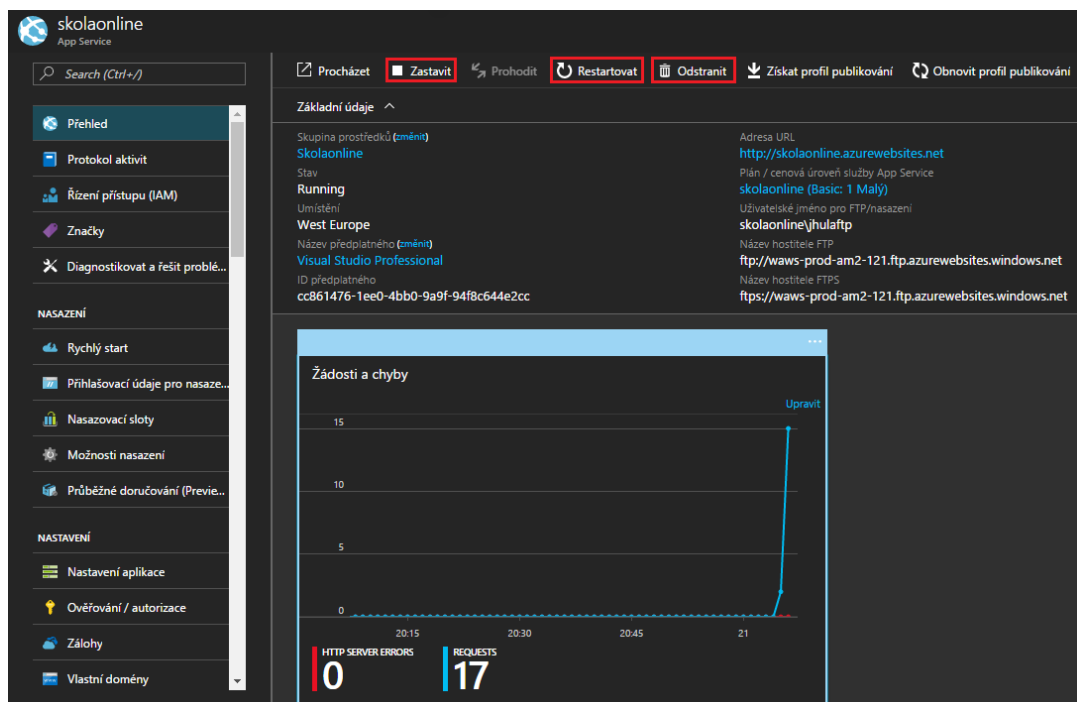
Advanced...

Test Connection OK Cancel

Do pole DB1 je doplněn Connection string pro připojení k databázi. Zaškrtněte volbu Use this connection string at runtime. Klikněte na Publish.



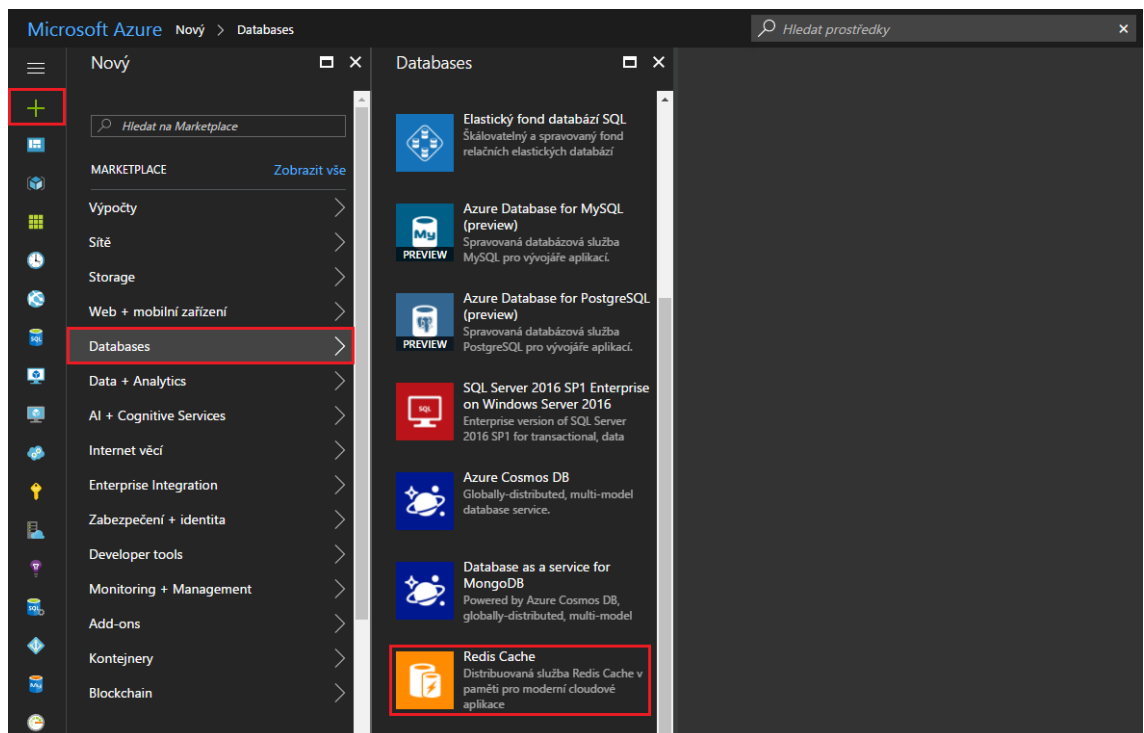
Po úspěšném publikování bude otevřen webový prohlížeč s publikovanou webovou aplikací. Sledování aplikace a veškeré nastavení aplikace je možné provádět přímo na portále Azure. Aplikaci je možné kdykoliv zastavit, restartovat nebo odstranit.



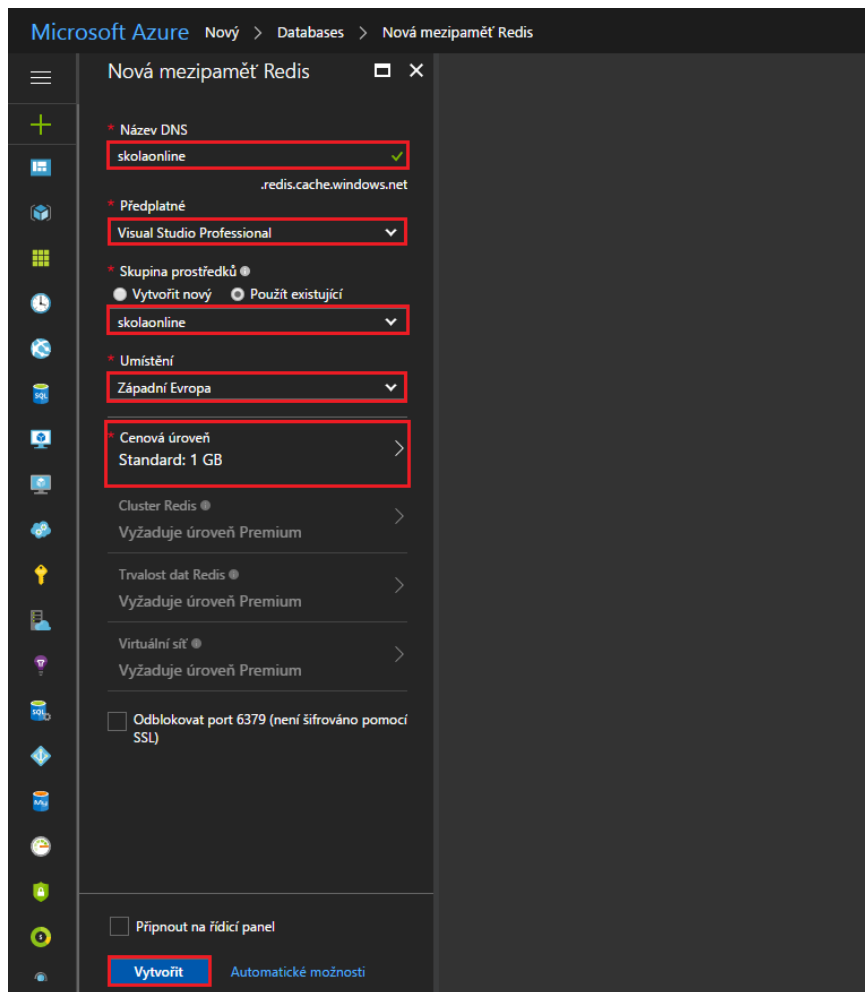
Nastavení mezipaměti Redis Cache

K udržení stavu relace bude použita mezipaměť Redis Cache, jak bylo popsáno v sekci 4.3.3. Pro použití mezipaměti je nutné provést její vytvoření na portále Azure. Následujte tyto kroky pro vytvoření mezipaměti:

1. Klikněte na tlačítko Nový v levém horním rohu na Azure portále
2. Vyberte Databases na stránce Nový a poté vyberte Redis Cache na stránce Databases



V okně Nová mezipaměť Redis zvolte DNS jméno, skupinu prostředků, umístění a cenovou úroveň. Poté klikněte na tlačítko Vytvořit.



Po vytvoření Redis Cache můžete její použití nastavit v aplikaci.

Otevřete Solution ve Visual Studiu. Klikněte pravým tlačítkem na Solution a vyberte Manage NuGet Packages. Najděte balíček RedisSessionStateProvider a nainstalujte ho. Po úspěšném nainstalování je do souboru web.config přidána tato sekce:

```
<sessionState mode="Custom"
customProvider="MySessionStateStore">
  <providers>
    <!--
    <add name="MySessionStateStore"
      host = "127.0.0.1" [String]
      port = "" [number]
      accessKey = "" [String]
      ssl = "false" [true|false]
      throwOnError = "true" [true|false]
      retryTimeoutInMilliseconds = "0" [number]
      databaseId = "0" [number]
      applicationName = "" [String]
      connectionTimeoutInMilliseconds = "5000" [number]
      operationTimeoutInMilliseconds = "5000" [number]
```

```

/>
-->
<add name="MySessionStateStore"
type="Microsoft.Web.Redis.RedisSessionStateProvider"
host="mycache.redis.cache.windows.net"
accessKey="..." ssl="true" />
</providers>
</sessionState>

```

Definujte adresu hosta a accessKey. Tyto informace získáte na stránce vytvořené služby Redis Cache na portále. Dále zakomentujte stávající nastavení session:

```

<sessionState mode="InProc"
stateConnectionString="tcpip=127.0.0.1:42424"
sqlConnectionString="data
source=127.0.0.1;Trusted_Connection=yes" cookieless="false"
timeout="300" />

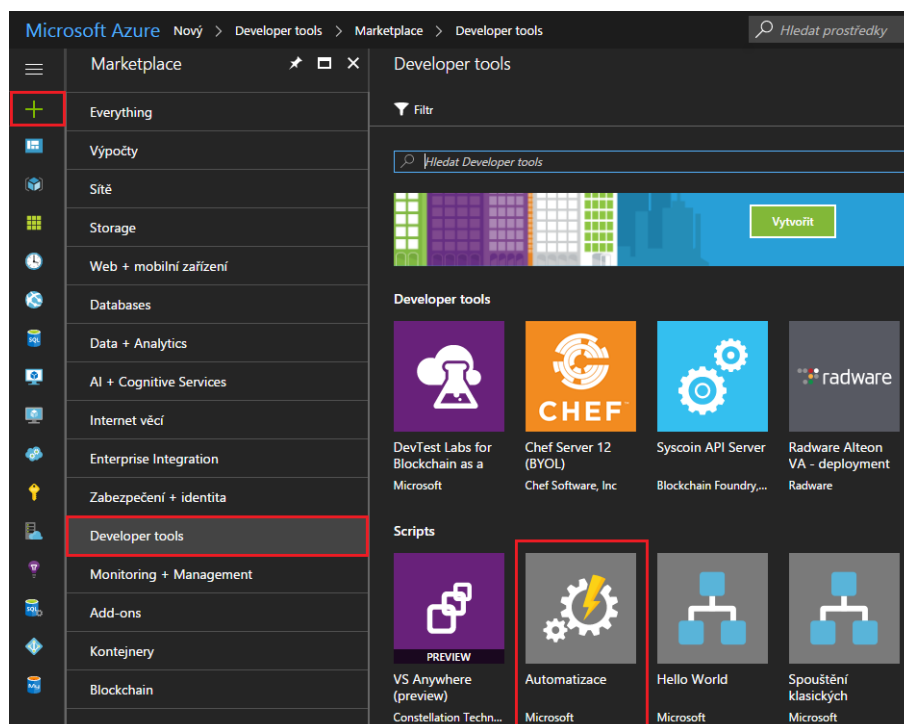
```

Po těchto změnách bude aplikace využívat pro udržení stavu relace Redis Cache.

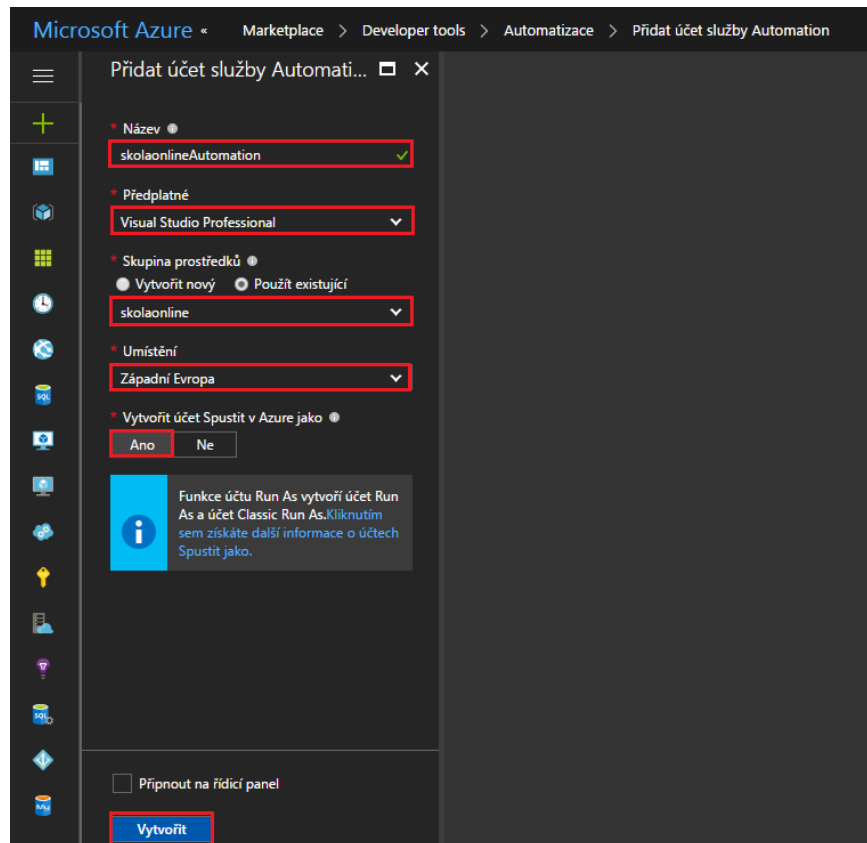
Nastavení runbooků pro službu Azure Automation

Úlohy spouštěné SQL Server agentem nahradíme runbooky ve službě Automation, jak bylo popsáno v sekci 4.2.4. Následujte tyto kroky k vytvoření účtu služby Automation:

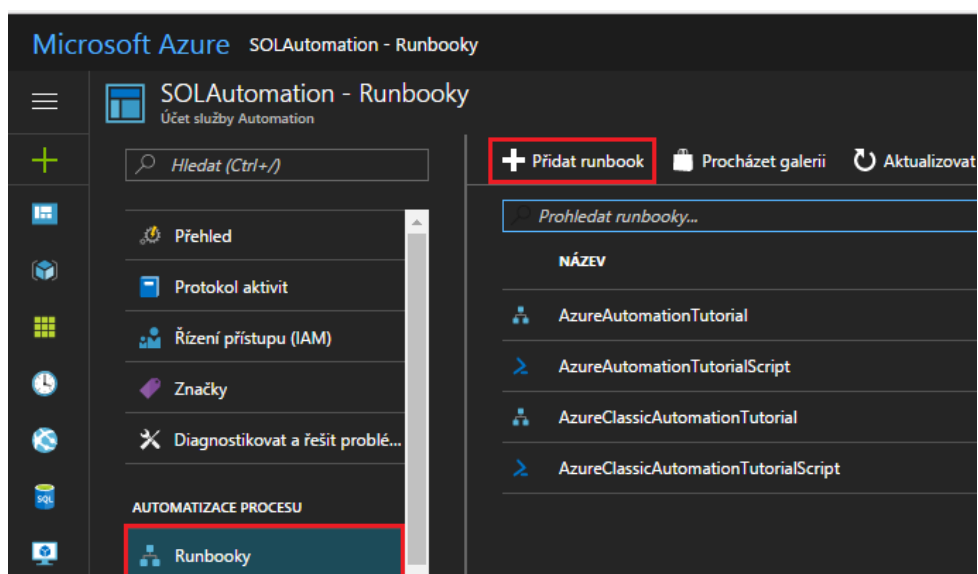
1. Klikněte na tlačítko Nový v levém horním rohu na portále
2. Vyberte Developer Tools na stránce Nový a poté vyberte službu Automatizace



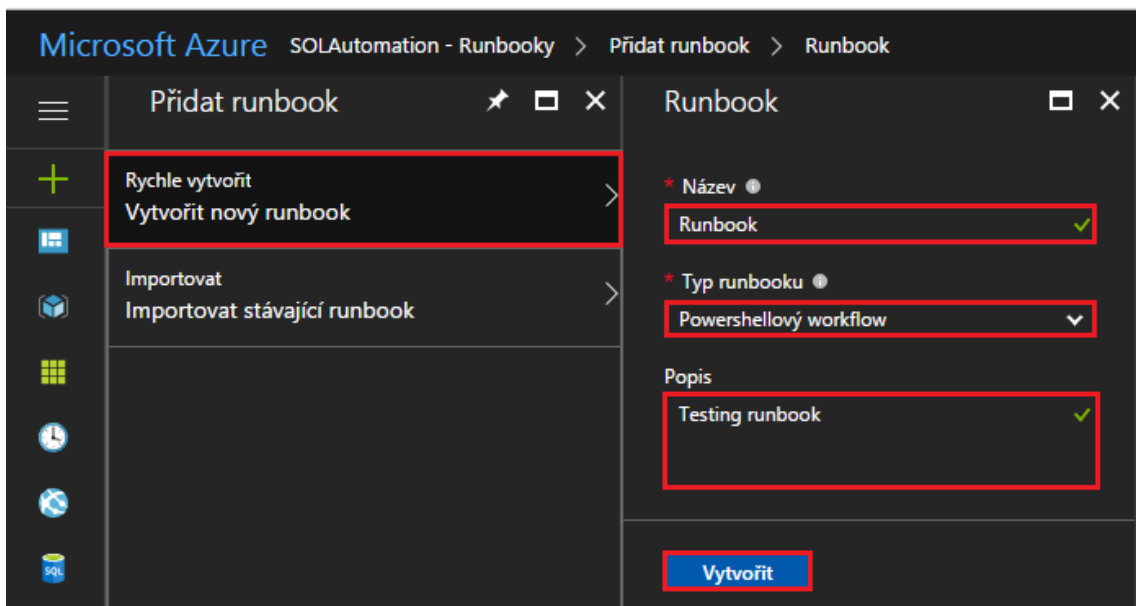
V okně Přidat účet Automatizace vyplňte název služby, vyberte předplatné, skupinu prostředků a oblast. U možnosti Vytvořit účet Spustit v Azure jako se ujistěte, že je vybráno Ano a klikněte na tlačítko Vytvořit.



Po vytvoření můžete vytvořit runbook a naplánovat jeho spuštění. Pro vytvoření runbooku přejděte na přehled služby Automatizace, v levém panelu vyberte Runbooky a klikněte na Přidat runbook.



V okně Přidat runbook vyberte Vytvořit nový runbook a vyplňte jeho název a popis. Typ runbooku zvolte Powershellový workflow.



Po vytvoření runbooku se otevře okno pro editaci kódu runbooku. Zde je ukázka kódu runbooku:

```
workflow Platby
{
    Param(
        [Parameter(Mandatory=$True)]
        [string] $SqlServer,

        [Parameter(Mandatory=$False)]
        [int] $SqlServerPort = 1433,

        [Parameter(Mandatory=$True)]
        [string] $Database,

        [Parameter(Mandatory=$True)]
        [PSCredential] $SqlCredential
    )

    # Get the username and password from the SQL Credential
    $SqlUsername = $SqlCredential.UserName
    $SqlPass = $SqlCredential.GetNetworkCredential().Password

    inlinescript {
        # Define the connection to the SQL Database
        $Conn = New-Object
System.Data.SqlClient.SqlConnection("Server=tcp:$using:SqlServer,
$using:SqlServerPort;Database=$using:Database;User
ID=$using:SqlUsername;Password=$using:SqlPass;Trusted_Connectio
n=False;Encrypt=True;Connection Timeout=30;")
    }
```

```

# Open the SQL connection
$conn.Open()

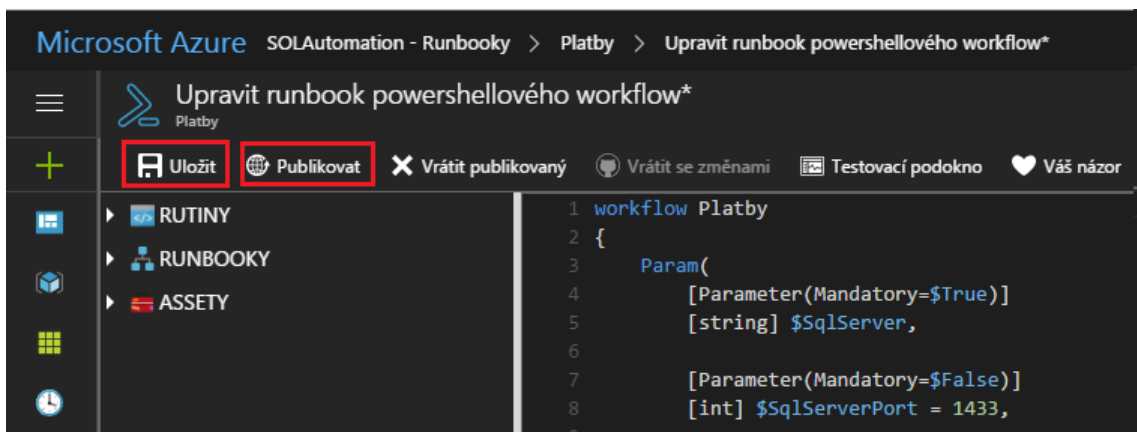
# Define the SQL command to run. Run the stored
procedure GenerujDoklady
    $Cmd=new-object system.Data.SqlClient.SqlCommand("EXEC
Platby.GenerujDoklady", $Conn)
    $Cmd.CommandTimeout=120

# Execute the SQL command
$Da=New-Object
system.Data.SqlClient.SqlDataAdapter($Cmd)

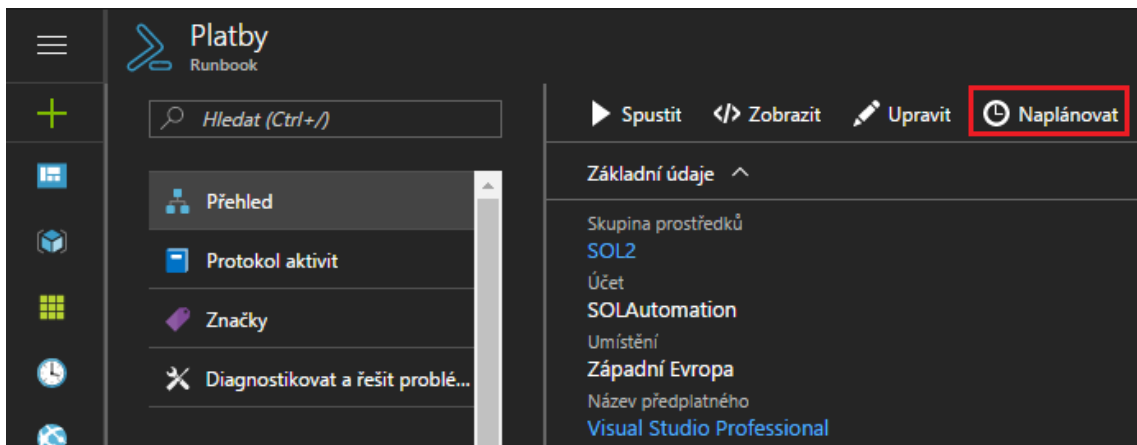
# Close the SQL connection
$conn.Close()
}
}

```

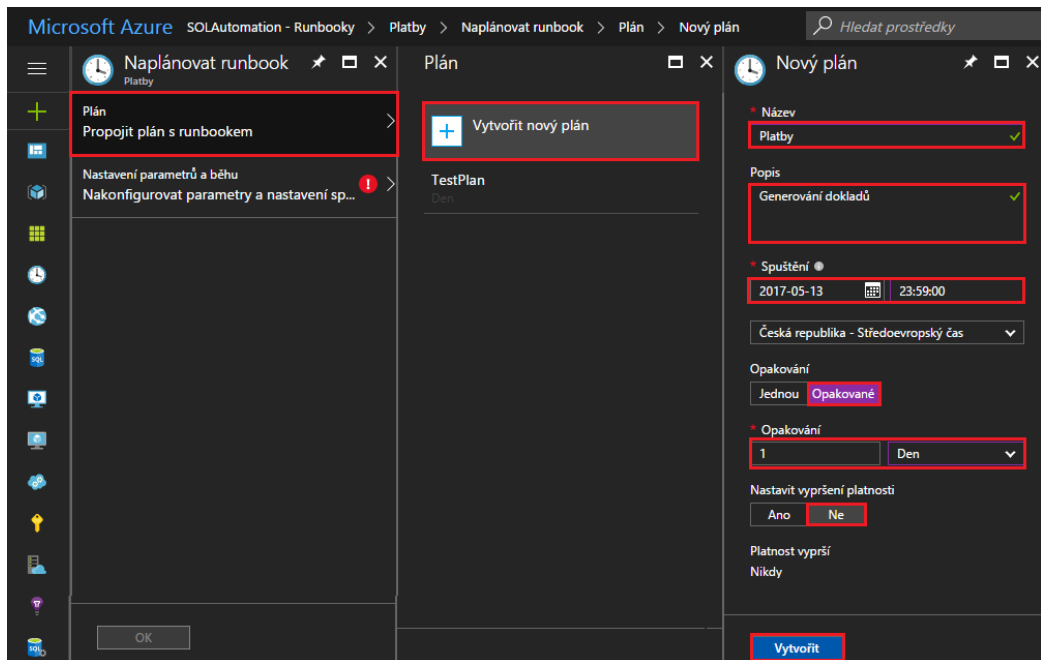
Poté co vytvoříte runbook, klikněte na Uložit a po uložení na Publikovat.



Když je runbook úspěšně publikován, můžete naplánovat jeho spuštění. Klikněte na Naplánovat.



V okně Naplánovat runbook klikněte na Propojit plán s runbookem poté na Vytvořit nový plán a nadefinujte spouštění runbooku. Pro dokončení klikněte na Vytvořit.



Pokud runbook má parametry, tak je potřeba je nastavit kliknutím na Nastavení parametrů a běhu. Po nastavení parametrů klikněte na OK a pro naplánování znovu OK.

