



# Windows Azure pre vývojárov a IT špecialistov



Windows Azure™

## Obsah

Windows Azure – predstavenie platformy .....	4
Kde sú vaše aplikácie, údaje a virtuálne počítače? .....	4
Základné princípy Windows Azure.....	6
Modely využívania služby Windows Azure .....	7
Virtuálne servery (IaaS).....	8
Webové stránky .....	9
Cloudové služby .....	9
Správa údajov.....	10
SQL Database .....	11
Tables .....	12
Blobs.....	12
Networking.....	13
Virtual Network.....	13
Connect .....	13
Messaging .....	13
Queues .....	14
Service Bus .....	14
Caching.....	14
Prakticky neobmedzená škálovateľnosť .....	14
Prvé kroky na platforme Windows Azure .....	16
Skúšobné konto na tri mesiace .....	16
Skúšobné konto v rámci predplatného MSDN.....	19
Konfigurácia vývojárskeho PC.....	21
Praktické príklady.....	22
Vytvorenie nového virtuálneho serveru .....	22
Rozdelenie záťaže medzi viaceré virtuálne stroje.....	26
Konfigurácia IIS na virtuálnych serveroch pre vyrovnanie záťaže.....	28
Vytvorenie virtuálneho servera s nainštalovaným SQL Serverom.....	30
Pripojenie diskov k virtuálnemu stroju .....	32
Webová aplikácia vo Windows Azure – praktický príklad .....	34
Vytvorenie ASP.NET MVC 4 webovej aplikácie .....	34
Vytvorenie Azure Web Site .....	36
Konfigurácia cludového úložiska SQL Database.....	37

Publikovanie ASP.NET MVC 4 aplikácie prostredníctvom Web Deploy .....	39
Migrácia webovej ASP.NET aplikácie do clodu .....	42
Spustenie aplikácie v emulátore platformy Azure .....	44
Ladenie aplikácie v lokálnom emulátore .....	46
Nasadenie projektu do reálneho cloudového prostredia.....	46
Odstránenie aplikácie z clodu .....	50
Vytvorenie cloudového projektu .....	51
Úpravy a ladenie aplikácie .....	55
Staging verzus Production.....	58
Ukladanie údajov .....	60
Vytvorenie a konfigurácia úložiska .....	60
Cvičný projekt pre ukladanie údajov do Table úložiska .....	63
Vytvorenie aplikácie využívajúcej SQL Database .....	71
Vytvorenie dátového modelu .....	73
Nastavenie aplikácie pre spoluprácu s databázou SQL Database.....	78
Príklad komunikácie cez Service Bus.....	79
Nastavenie Service Bus Namespace .....	79
Vytvorenie projektu .....	81
Posielanie správ do fronty Service Bus .....	85
Vytvorenie Worker Role.....	87
Vytvorenie aplikácie pre platformu Node.js a jej publikovanie na Azure.....	88
Test aplikácie v emulátore .....	91
Nasadenie aplikácie do clodu .....	92

## Windows Azure – predstavenie platformy

O novom type IT architektúry s názvom „cloud computing“ určite každý IT špecialista a vývojár už počul, väčšina pochopila princíp, no koľkí z vás sa môžu pochváliť fungujúcou aplikáciou, ktorá si pokojne sídi v „obláčikoch“, teda v dátových centrách. Zjednodušene by sa dalo prostredie Windows Azure definovať ako **platforma umožňujúca vytváranie virtuálnych serverov a spúštanie aplikácií v prostredí clodu**. Zákazníci pracujúci s klientskymi počítačmi pripojenými na Internet môžu vytvárať virtuálne stroje, spúštať svoje masívne škálovateľné aplikácie, vytvárať webové aplikácie, ktoré bežia a ukladajú svoje dátá v dátových centrách spoločnosti Microsoft, prípadne v dátových centrách len bezpečne ukladať svoje údaje. Aby ste mohli z tejto širokej palety služieb profitovať je potrebné pochopiť základy.

### Kde sú vaše aplikácie, údaje a virtuálne počítače?

Záujemcovia o túto cloudovú technológiu si občas kladú otázku, kde a v akej podobe vlastne ten „azúrový“ cloud je situovaný. Na obrázku je odpoveď na otázku: „Kde?“. Aby sme boli presnejší, Azure je prevádzkovany v globálnych dátových centrách, pričom na každom kontinente je ich niekoľko. V Európe sú umiestnené v Dubline a Amsterdame a sú medzi sebou replikovateľné.



Globálne dátové centrá spoločnosti Microsoft. Zeleným krúžkom sú označené centrá poháňané „green“ energiou, napríklad v San Antoniu, Chicagu a Dubline.

Aj odpoveď na druhú časť otázky: „V akej podobe“ bude najnázornejšia vo forme obrázkov, ktoré zachytávajú dátové centrum zvonku aj zvnútra.



Dátové centrum v Dubline



*Serverová infraštruktúra je umiestnená v modulárnych kontejneroch*



*Každý kontejner je napojený na dátovú, energetickú a chladiacu infraštruktúru*



*Pohľad do vnútra kontejneru*

## Základné princípy Windows Azure

Platforma Windows Azure poskytuje celú škálu funkcií pre vytváranie aplikácií, ktorých použitie siahá od spotrebiteľských webov až po nasadenie v podnikoch, a zahŕňa operačný systém pre aplikácie bežiace v prostredí clodu a balík služieb pre vývojárov.

Windows Azure je analógiou operačného systému pre služby typu „cloud computing“, ktorý slúži ako prostredie pre vývoj, hostovanie a správu služieb. Je to flexibilná platforma – podporuje veľké množstvo jazykov a môžete ju integrovať s existujúcimi systémami hostovanými interne vo firme. Pri vývoji aplikácií a služieb na platorme Windows Azure môžu vývojári vychádzať z existujúcich znalostí produktu Microsoft Visual Studio a znalostí štandardov a protokolov ako napríklad SOAP, REST, XML alebo PHP.



# Windows Azure™

Windows Azure umožňuje:

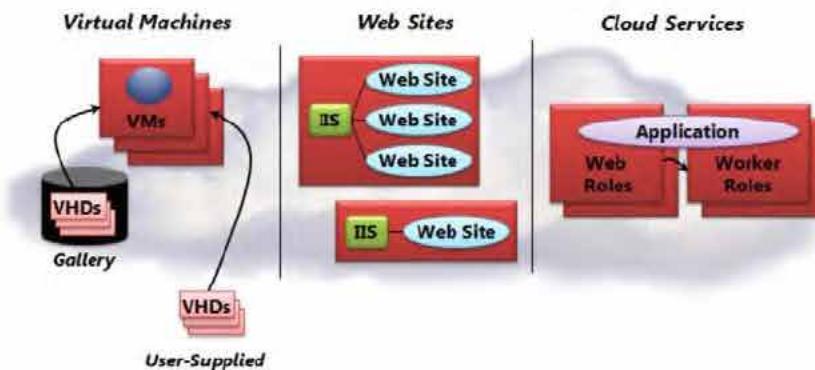
- Vytvárať virtuálne servery v cloude.
- Prevádzkovať štandardizované procesy v cloude.
- Vyvíjať, upravovať a distribuovať škálovateľné aplikácie s minimálnymi nárokmi na interné prostriedky.
- Využívať úložisko s obrovskou kapacitou a vykonávať dávkové spracovanie alebo náročné a rozsiahle výpočty.
- Rýchlo a nízkonákladovo vytvárať, testovať, ladiť a distribuovať webové služby.

## Výhody platformy Windows Azure

- Svoj nápad uvediete na trh rýchlejšie a pritom platíte iba za to, čo používate.
- Znižte nároky na miesto v budovách a rozšírite kapacitu interných výpočtových prostriedkov.
- Znižte nároky na správu IT.
- Budete môcť rýchlejšie reagovať na obchodné potreby a požiadavky zákazníkov.
- Môžete si zvoliť medzi interným alebo externým modelom nasadenia podľa toho, ktorý vám lepšie bude vychovovať.
- IT prostriedky budú škálovateľné presne podľa vašich potrieb.
- Výpočtové prostriedky môžete mať aktívne iba na dobu, počas ktorej budú potrebné.
- Výraznú časť správy preberie za vás dátové centrum.
- Nemusíte sa zaoberať správou hardvéru.
- Pri vývoji cloud aplikácií využijete existujúce znalosti a zručnosti.
- Prostredie pre vývojárov a IT odborníkov je konzistentné s interným aj cloudovým prostredím.

## Modely využívania služby Windows Azure

Na platforme Windows Azure môžete vytvárať virtuálne počítače, prevádzkovať webové aplikácie, clouдовé služby, či ukladať údaje. Každý z modelov je možné využívať separatne.



## Hlavné súčasti platformy Windows Azure

Ak sa pokúsime tieto modely zaškatuľovať podľa klasickej cloubovej klasifikácie, tak virtuálne stroje sú typickým príkladom IaaS, teda Infrastructure as a Service (IaaS), a clouдовé služby je možné zaradiť do kategórie PaaS (Platform as a Service).

Predchádzajúci obrázok vyjadruje len základné modely. Možnosti platformy Windows Azure sú oveľa širšie, platforma zapuzdruje napríklad ukladanie a správu údajov, správu virtuálnych sietí, správu identít, messaging, BI analýzy, prácu s multimédiami, komerčné aplikácie, cachovanie a mnohé ďalšie užitočné funkcie.

Execution Models	Virtual Machines	Web Sites	Cloud Services
Data Management	SQL Database	Tables	Blobs
Networking	Virtual Network	Connect	Traffic Manager
Business Analytics	SQL Reporting	Hadoop	
Messaging	Queues	Service Bus	
Caching	Caching	CDN	
Identity	Windows Azure Active Directory		
High-Performance Computing	HPC Scheduler		
Media	Media Services		
Commerce	Marketplace		
SDKs	.NET Java PHP Python Node.js		

## Komponenty Windows Azure

## Virtuálne servery (IaaS)

Možnosť vytvoriť virtuálny stroj na vyžiadanie, či už zo štandardného obrazu alebo z vašich inštalačných médií môže byť veľmi užitočné, navyše pri hodinovej tarifikácii aj rentabilné. Môžete vytvoriť taký počet virtuálnych serverov, koľko potrebujete a len na nevyhnutne potrebnú dobu. V porovnaní s klasickými servermi odpadá doručovanie, vybaľovanie, či konfigurácia. V galérii štandardných VHD nájdete Windows Server 2008 R2, Windows Server 2012, Linuxové servery, prípadne kombináciu Windows Server 2008 R2 so serverom SQL Server.

Virtuálny server môžete využiť na vývoj, či ako testovaciu platformu. Môžete si zvoliť SQL Server alebo inú databázovú platformu, ktorá beží na jednom alebo viacerých virtuálnych serveroch. Môžete použiť virtuálne servery na rozšírenie kapacity vášho dátového centra napríklad v prípade predvídateľných, ale aj nepredvídateľných špičiek, môžete ich použiť na prevádzkovanie služby SharePoint. Služby Active Directory v systéme Windows Azure VM (Virtual Machine) je možné vytvoriť Windows domény v cloude.

Zosumarizované a podčiarknuté – ako využijete virtuálne servery vo Windows Azure závisí len a len od vás, od vašich potrieb a požiadaviek.

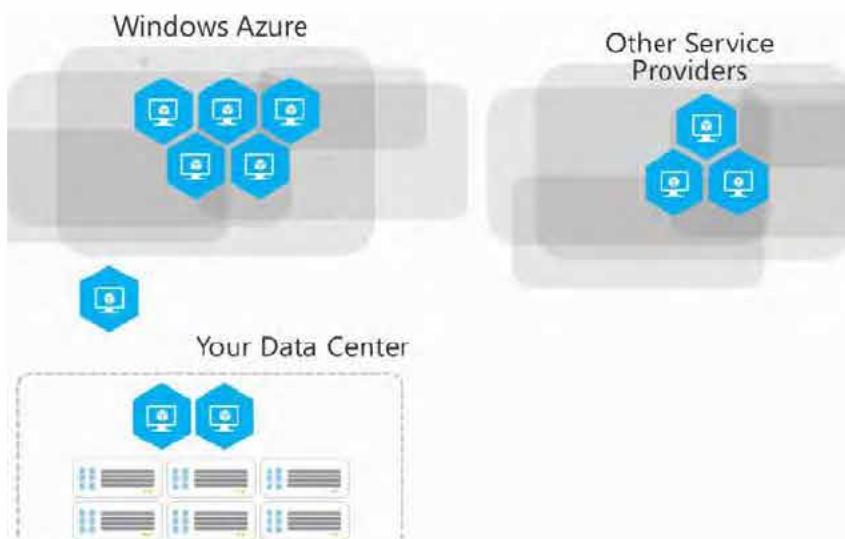
V cloude Windows Azure môžete vytvárať virtuálne servery na platformách Windows Server a Linux. Ak si vytvoríte viac virtuálnych serverov, môžete z nich vytvoriť serverovú farmu a flexibilne medzi servermi prerozdeľovať záťaž. Samozrejmosťou je VPN (Virtual Private Networking).

Pri vytváraní virtuálneho serveru môžete využiť voľbu QUICK CREATE, s tým, že zadáte len niekoľko základných parametrov a virtuálny server nakonfigurujete neskôr. Pri výbere inštalačného image FROM GALLERY zadávate v procese vytvárania viac parametrov, napríklad prístupové privilégia, či sieťové prepojenie s iným už vytvoreným virtuálnym serverom.



### *Možnosti vytvárania virtuálnych serverov*

Virtuálne servery môžete v prípade potreby presúvať nielen medzi svojím dátovým centrom a službou Windows Azure, ale aj medzi inými poskytovateľmi služieb.



*Možnosť presúvania virtuálnych serverov medzi lokálnym dátovým centrom a službou Windows Azure*

### Webové stránky

Jednou z najčastejších činností v cloude je prevádzkovanie webových stránok a webových aplikácií. Ako jedna z alternatív sú v predchádzajúcej stati popísané virtuálne stroje. Ako nevýhoda sa javí nutnosť konfigurácie a správy jedného alebo viacerých virtuálnych počítačov. Z toho logicky vyplýva, že táto možnosť je určená pre IT fundovaných používateľov. Webové stránky na platforme Windows Azure ponúkajú možnosť prevádzkovania stránok a webových aplikácií pre používateľov v oblasti podnikania, či záujmu mimo IT. Veľkou výhodou pre nich je, že sa nemusia staráť o záležitosti spojené s administráciou.

Model „Webové stránky“ ponúka spravované webové prostredie technicky prevádzkované cez IIS. Používateľ môže presunúť existujúcu stránku do IIS lokalít služby Windows Azure, alebo vytvoriť novú webovú stránku priamo v cloude.

Windows Azure Webové stránky podporujú technologické platformy .NET, PHP, a Node.js, SQL Database (od partnerskej firmy ClearDB). Poskytujú tiež integrovanú podporu pre populárne aplikácie, vrátane WordPress, Joomla, či Drupal. Windows Azure Webové stránky sú lacná, škálovateľná a užitočná platforma pre vytváranie webových stránok a webových aplikácií vo verejnem cloude.

### Cloudové služby

Predpokladajme situáciu, kedy potrebujete vytvoriť cloudovú aplikáciu, ktorá môže podporovať mnoho užívateľov súčasne, je stále dostupná a nevyžaduje veľa administratívy. Vtedy je ideálny cloudový model PaaS, teda platforma ako služba. Aké sú jej výhody v porovnaní s modelom „Windows Azure Webové stránky“, popísaným v predchádzajúcej stati? Webové stránky umožňujú nasadenie webových aplikácií, no sú tam zákonite niektoré obmedzenia. Nemáte napríklad prístup správcu, čo znamená, že nie je možné nainštalovať ľubovoľný softvér.

Naproti tomu virtuálne servery Windows Azure VM ponúkajú značnú flexibilitu, vrátane administratívneho prístupu, dajú sa použiť pre vysoko škálovateľné aplikácie, ale budete musieť zvládnuť všetky aspekty správy virtuálneho serveru, nakoľko tento je plne vo vašej rôžii.

Riešením je služba Windows Azure Cloud Services. Táto technológia je navrhnutá pre podporu spoľahlivého behu škálovateľných aplikácií. Tento model sa nazýva Platforma ako služba (PaaS). Ak ho chcete použiť, môžete vytvoriť aplikáciu pomocou vami vybranej technológie, napríklad C#, Java, PHP, Python, Node.js... Váš kód potom beží na virtuálnych strojoch, ktoré sú však odlišné od virtuálnych serverov, ktoré vytvoríte v rámci Windows Azure VM. Virtuálne stroje v rámci Windows Azure Cloud Services sa automaticky spravujú a inštalujú sa opravné balíčky. Windows Azure tieto virtuálne počítače monitoruje a zariadi reštart, či obnovu v prípade zlyhania.

Pri vytváraní inštancie môžete využiť dve role, **Web a Worker**, obe založené na systéme Windows Server. Hlavný rozdiel medzi nimi je, že inštancia Web role beží pod IIS. Obidve role sú riadené rovnakým spôsobom a aplikácia môže využívať obidve. Napríklad webová role môže prijímať požiadavky od používateľov, a potom ich odovzdá worker roli na spracovanie.

Webové role môžu cez služby IIS (Information Server) priamo komunikovať s klientom a reagovať na jeho požiadavky. Rola je vhodná pre aplikácie, ktoré sprístupňujú svoje funkcie cez HTTP protokol, prevažne vhodná teda pre webové servery a webové aplikácie. Inštalačný balíček webovej role sa po publikovaní do clodu umiestní do pracovných priečinkov IIS na jednotlivých inštanciach.

Worker role nemajú priamy kontakt s okolitým svetom, takže nedokážu reagovať na žiadne vonkajšie podnety, no môžu generovať vlastné požiadavky na vstup. Po spustení svojho kódu dokážu prijímať správy z fronty, takže fungujú ako batch job alebo Windows služba. Rola je vhodná pre aplikácie, ktoré vykonávajú spracovanie na pozadí. Ak napríklad webová role prijme „zvonku“ požiadavku, ktorej splnenie je výpočtovo náročné, pridelí tieto úkony pre worker role. Worker role sa dá použiť aj pre webové aplikácie, ktoré budú prevádzkované na inom webovom serveri než IIS.

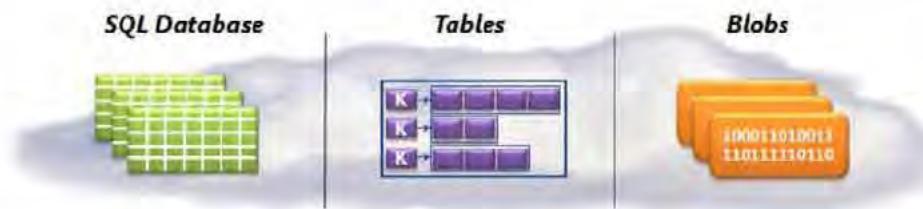
Pre každú rolu je dedikovaný požadovaný počet inštancií virtuálnych strojov požadovanej konfigurácie (XS, S, M, L, XL ) s operačným systémom Windows Server 2008 R2 – 64-bitovým, súčasťou inštancie je aj technologická platforma .NET Framework vo verziách 3.5 a 4.0.

## Správa údajov

Rôzne druhy aplikácií potrebujú rôzne druhy údajov. Preto Windows Azure ponúka niekoľko rôznych spôsobov ukladania a správy údajov. Jeden z intuitívnych spôsobov už bol spomenutý – SQL Server alebo iný DBMS na vami vytvorenom (a spravovanom) virtuálnom serveri. Na virtuálnych serveroch môžete spúštať akýkoľvek, aj nerelačný systém (NoSQL) správy údajov, ako je napríklad MongoDB, či Kasandra. Jednoducho povedané, na virtuálny server si nainštalujete a budete prevádzkovať taký systém správy údajov aký používate vo vlastnom dátovom centre.

Okrem toho Windows Azure ponúka tri hlavné možnosti ukladania a správy údajov:

- SQL Database,
- Tables,
- Blobs.



Typy dátových úložísk Windows Azure

### SQL Database

Microsoft SQL Database je relačná databázová služba v cloude, ktorá vychádza z technológií servera SQL Server. *Predtým mala označenie SQL Azure.* Služba poskytuje všetky kľúčové funkcie relačného databázového systému, vrátane atomických transakcií, či multipoužívateľského prístupu pri zachovaní integrity údajov. SQL Database zjednodušuje sprevádzkovanie a nasadenie veľkého počtu databáz. Vývojár nemusí inštalovať, nastavovať, aktualizovať ani spravovať žiadny softvér. K základným funkciám patrí vysoká dostupnosť a odolnosť proti chybám. SQL Database podporuje jazyk Transact-SQL (T-SQL), služby pre správu relačných databáz (RDBMS), vytváranie, spracovávanie a úpravy tabuľiek, pohľadov, indexov, rolí, uložených procedúr a funkcií, transakcie, dočasné tabuľky.

Zákazník môže vychádzať z existujúcich znalostí vývoja v T-SQL a známeho relačného dátového modelu, ktorý zodpovedá existujúcim interne prevádzkovaným databázam. Taktiež môže využívať podmnožinu uložených procedúr a systémových zobrazení z existujúcej verzie systému SQL Server. K údajom je možné pristupovať z aplikácií prostredníctvom známych programovacích modelov ako sú Entity Framework, ADO.NET, JDBC, a ďalšie. Pre správu údajov je možné využívať SQL Server Management Studio.

SQL Database sa nedá zjednodušene charakterизovať ako DBMS v cloude. Je to komplexná PaaS služba. Ponúka možnosť federácie, ktorá distribuuje dátá medzi viac serverov. To je užitočné pre aplikácie, ktoré pracujú s veľkými objemami údajov, alebo požadujú extrémny výkon.

K údajom je možné pristupovať z rôznych klientskych systémov vrátane stolových počítačov, notebookov, tabletov a smartfónov.

### SQL Database umožňuje:

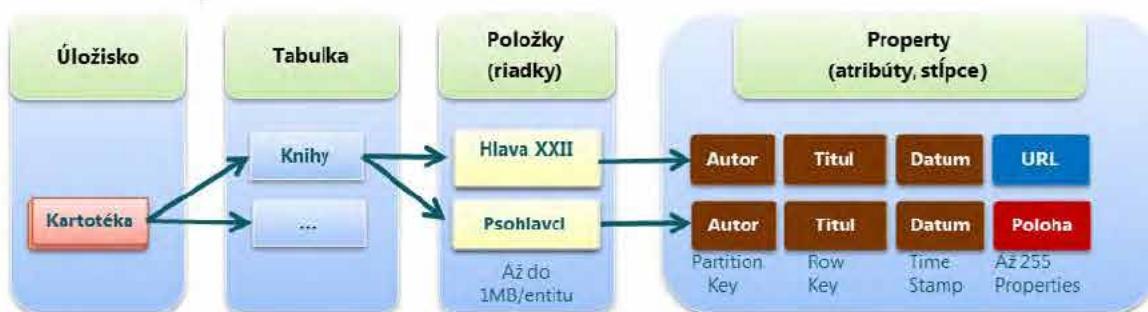
- Vyvíjať vlastné škálovateľné webové aplikácie pre malé a stredné podniky, nadšencov a začínajúce firmy.
- Vyvíjať balíčky obchodných aplikácií, čo využijú predovšetkým tradiční výrobcovia softvéru, poskytovatelia SaaS a vývojári aplikácií na zákazku.
- Vyvíjať aplikácie pre jednotlivé oddelenia veľkých spoločností.
- Konsolidovať v prostredí cloud rôzne zdroje dát a poskytovať k nim zabezpečený prístup z rôznych miest, stolových počítačov aj mobilných zariadení.

## Výhody technológie SQL Database

- Nie je potrebné inštalovať a aktualizovať softvér.
- Automaticky zaisťuje vysokú dostupnosť a odolnosť proti chybám.
- Umožňuje jednoducho sprevádzkovať a nasadiť väčší počet databáz.
- Podľa potrieb firmy môže škálovať databázy vertikálne aj horizontálne.
- Podporuje koncept multitenancy.
- Dá sa integrovať so systémom SQL Server a balíkom nástrojov vrátane produktu Visual Studio.
- Podporuje známy relačný databázový model jazyka T-SQL.

## Tables

Napriek tomu, že ide o štruktúrované úložisko, nie je to relačná databáza, nakoľko neumožňuje relácie cez cudzie kľúče, ani priame spájanie údajov z viacerých tabuľiek. Môžete použiť len jeden index a transakčné spracovanie je možné len v rámci partície.



### Princíp fungovania úložiska Tables

Predpokladajme, že chcete vytvoriť Windows Azure aplikáciu, ktorá potrebuje rýchly prístup k údajom, ktoré majú jednoduchú štruktúru a nie je nutné robiť zložité SQL dopyty nad týmito údajmi. Typickým príkladom je aplikácia na uloženie profilu zákazníka. Ak bude aplikácia úspešná, je potrebné počítať s veľkým množstvom údajov. Vtedy má význam technológia „Tables“, ktorá je typickým príkladom NoSQL databázy s prístupom typu kľúč/hodnota.

## Blobs

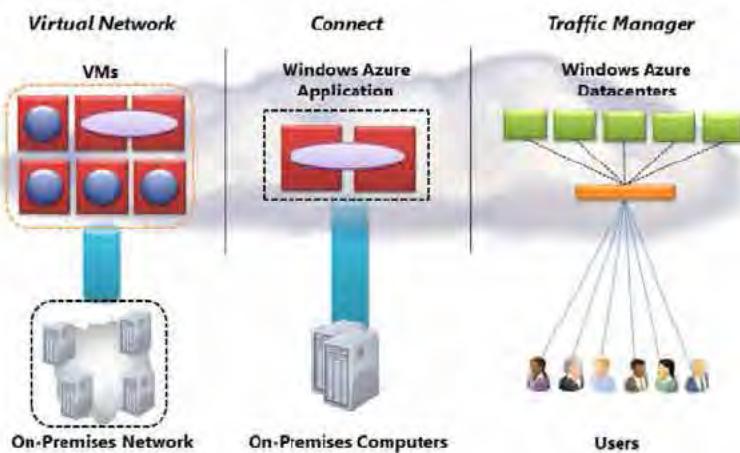
**Blob (Binary Large Objects)** je úložisko určené pre binárne údaje, prípadne multimediálne údaje (fotografie, MP3 súbory, videá...). Sú zapuzdené v kontajneroch, pričom každý kontajner jeden, alebo viac blob úložísk, ktoré môžu mať veľkosť až do jedného terabajtu a môžu mať asociované metadáta, ako sú napríklad údaje o súbore (napríklad čo je na fotografii, či v hudobnej nahrávke). Aplikácie (role) môžu spolupracovať s týmto úložiskom rovnako, ako keby bolo lokálnym súborovým systémom NTFS.

Rozlišujeme dva typy:

- **Block Blob** – optimalizovaný pre streamovacie scenáre. Blob pozostáva zo sekvencie blokov o veľkosti maximálne 4 MB, pričom limit na blob je 200 GB.
- **Page Blob** – pre klasické „náhodné“ read/write scenáre. Blob pozostáva z poľa stránok, pričom limit na blob je 1 TB.

## Networking

Windows Azure fyzicky beží v niekoľkých veľkých dátových centrach v Spojených štátoch, Európe a Ázii. Pri vytváraní virtuálnych serverov, spúštaní aplikácií alebo ukladaní údajov si môžete vybrať jedno alebo viac dátových centier. Môžete sa k nim pripojiť cez Windows Azure Virtual Network, alebo Windows Azure Connect. Ak vaše Windows Azure aplikácie bežia vo viacerých dátových centrach, môžete použiť Windows Azure Traffic Manager pre sofistikované smerovanie požiadaviek od užívateľov.



### Sieťová architektúra

#### Virtual Network

Jedným zo spôsobov ako využiť verejný cloud je považovať ho za rozšírenie vlastného dátového centra. Týka sa to hlavne možností vytvárania virtuálnych serverov, za ktoré nemusíte platiť, keď ich už nebudeš potrebovať. Na prepojenie sa používa Windows Azure Virtual Network. Správca môže nastaviť virtuálne privátne siete (VPN) medzi miestnou sieťou a definovanou skupinou virtuálnych počítačov bežiacich vo Windows Azure. Vzhľadom k možnosti priradiť vlastné IP adresy do clouдовých virtuálnych strojov sa tieto javia, ako keby boli vo vašej sieti a používateľia vo vašej firme, či organizácii k nim aj takto budú pristupovať.

#### Connect

Vytvorenie VPN medzi miestnou sieťou a skupinou virtuálnych serverov je užitočné, ale na druhej strane vyžaduje VPN brány a služby správca siete. Predpokladajme, že ste vývojár, ktorý chce len jediné pripojenie Windows Azure aplikácie na server, alebo skupinu serverov vo vašej firme. To je presne úloha pre Windows Azure Connect. Vývojár len nainštaluje príslušný softvér a aplikácia môže komunikovať ako keby bola v lokálnej sieti.

#### Messaging

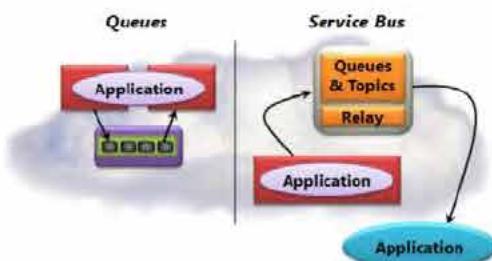
Aplikácie musia často komunikovať pomocou správ. V niektorých prípadoch vystačia s jednoduchou frontou správ, inokedy sú potrebné zložitejšie interakcie. Windows Azure ponúka dva hlavné spôsoby komunikácie pomocou správ:

## Queues

Funguje to jednoducho. Aplikácia posiela správy do fronty, a tietože správy číta iná aplikácia. Tako môže napríklad fungovať komunikácia medzi web a worker rolou vo webovej aplikácii, zdieľanie videa a podobne.

## Service Bus

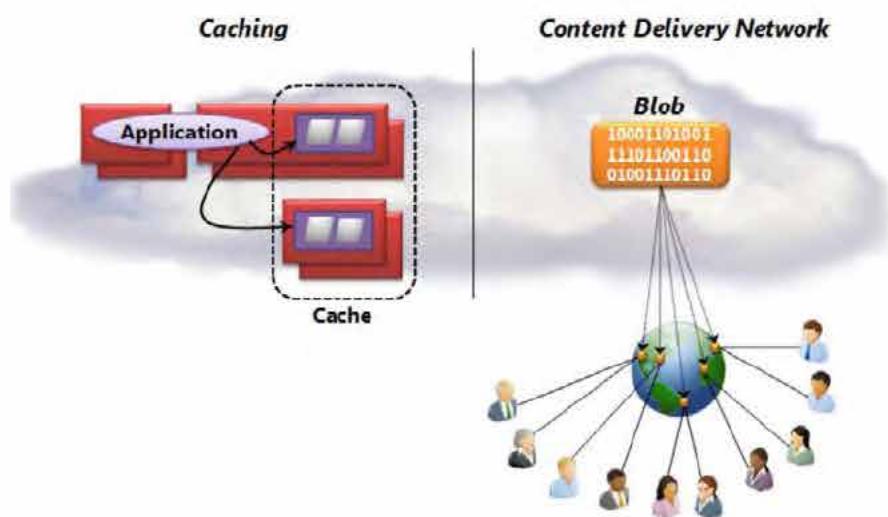
Cieľom Windows Azure Service Bus je, aby si aplikácie bežiace prakticky kdekoľvek mohli vymieňať dátu. Na rozdiel od Windows Azure Queues, Service Bus poskytuje mechanizmus typu publish-and-subscribe. Jedna aplikácia môže posielat správy na určenú tému, zatiaľ čo iné aplikácie môžu vytvoriť kontrakt na túto tému. To umožňuje one-to-many komunikáciu medzi súborom aplikácií.



## Messaging

### Caching

Aplikácie majú prirodzenú tendenciu pristupovať k rovnakým údajom znova a znova. Preto je vhodné zachovať kópiu týchto údajov pre prípadné opäťovné použitie v krátkom čase. Windows Azure ponúka pre tento účel dve rôzne služby – vyrovnávaciu pamäť údajov používaných aplikáciami a doručovanie obsahu (CDN network), ktorý ukladá údaje do vyrovnávacej pamäte BLOB dát na disku.



## Caching

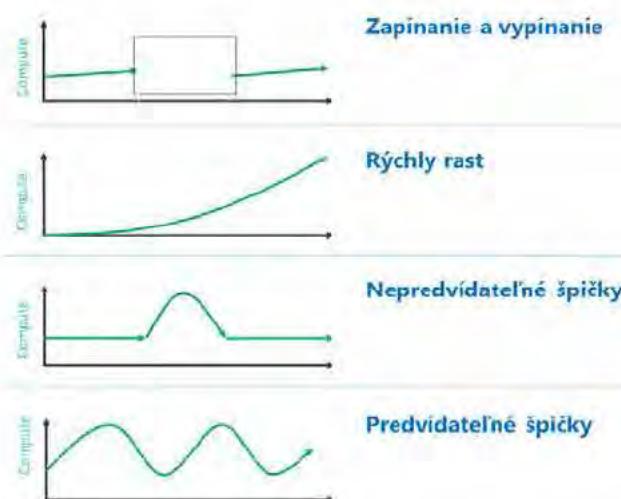
### Prakticky neobmedzená škálovateľnosť

Aplikácia sa dá vypublikovať na jeden virtuálny stroj, niekoľko strojov a v prípade potreby aj na desiatky, stovky, či tisícky virtuálnych strojov. Ak pominie obdobie vysokej záťaže, môžete počet

prenajatých strojov, či množstvo úložnej kapacity operatívne znížiť, čím sa samozrejme proporcionalne zníži aj mesačná platba za službu, zjednodušene povedané, zákazník platí len za to, čo skutočne využíva.

Výhody verejného clodu sa prejavujú hlavne pri scenároch kedy klasická IT infraštruktúra jednoznačne zlyháva, prípadne je výrazne drahšia. Napríklad:

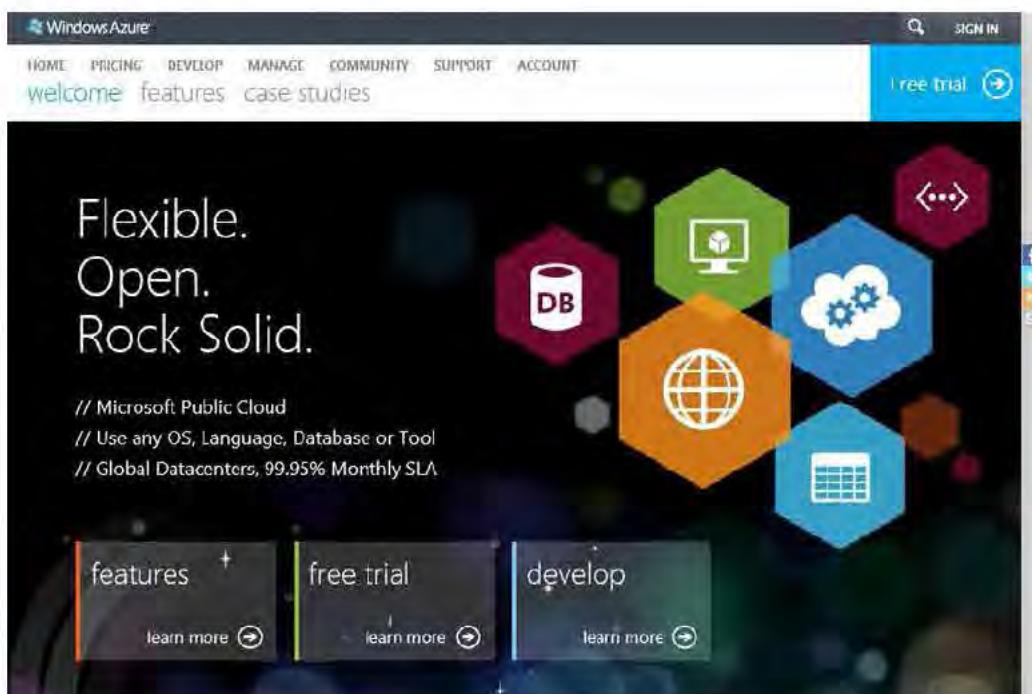
- **Nárazové dávkové spracovanie údajov** napríklad len na konci mesiaca, či fiškálneho roku. Kupovať pre tento účel výkonnú infraštruktúru, ktorá väčšinu mesiaca bude ležať nevyužitá je veľmi neekonomicke. Podobný scenár „predpokladaných špičiek“ sa týka aj e-shopov, hlavne pred Vianocami, kedy vzrástie objem návštěvníkov, či zákazníkov aj na niekoľkonásobok ročného priemeru, aby v januári zase klesol na pôvodnú hodnotu.
- **Rýchly rast firmy**, či už predpokladaný, alebo neočakávaný. V takomto prípade veľmi rýchlo rastú aj požiadavky na IT. Takýto scenár je typický pre úspešné startupy, aplikácie online, napríklad e-shopy, ktoré sa rozbiehajú a záťaž aj počet zákazníkov môže rásť exponenciálne a to zo dňa na deň.
- **Predvídateľné a nepredvídateľné špičky** – kedy zrazu na určité obdobie vzrástie z nejakého niekedy dokonca aj neidentifikovateľného dôvodu záťaž na niekoľkonásobok pôvodnej hodnoty. Môže to trvať niekoľko dní a môže sa to opakovať. Ak využívate služby verejného clodu nemusíte rýchlo objednávať hardvérovú infraštruktúru, ktorá vám po odznení špičky bude nanič. Stačí niekoľko kliknutí a na hodinu, deň, týždeň alebo mesiac prikúpite potrebnú virtuálnu kapacitu a problém je vyriešený. Keď už táto kapacita nebude potrebná, tak ju zase deaktivujete.



Výhody verejného clodu sa prejavia hlavne v niektorých špecifických typoch scenárov

## Prvé kroky na platforme Windows Azure

Každý začiatok je trochu ťažký, lebo sa treba naučiť niečo nové, no následne môže osvojenie novej technológie znamenať vyššiu efektívnosť, úsporu nákladov a v konečnom dôsledku významnú konkurenčnú výhodu. Námetom publikácie je vývoj aplikácií pre cloudovú platformu Microsoft Windows Azure. Dobrou správou je, že pre „azúrové“ aplikácie sa využíva rovnaké vývojové prostredie Visual Studio 2010 alebo Visual Studio 2012, ktoré používate aj pre vývoj klasických, či webových aplikácií.

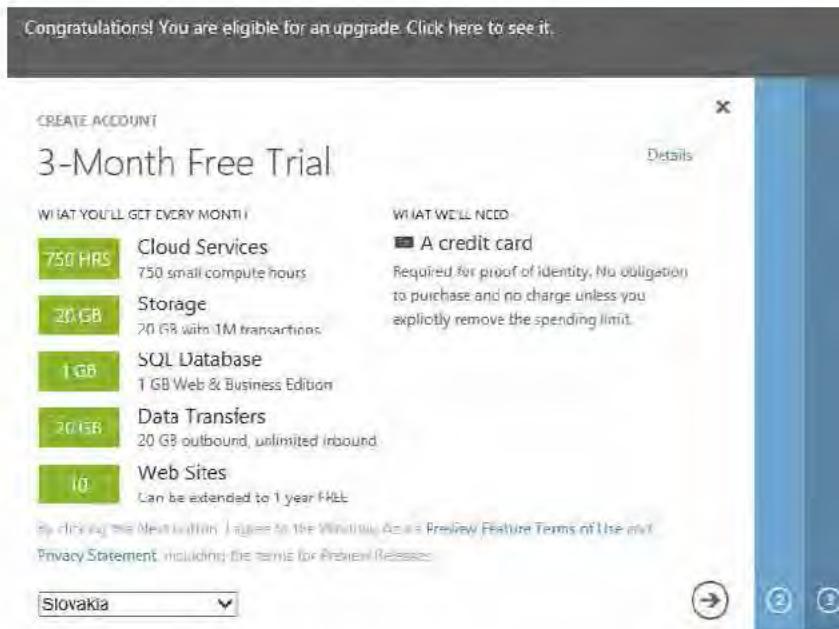


Vstupný bod do služby Windows Azure [www.windowsazure.com](http://www.windowsazure.com)

### Skúšobné konto na tri mesiace

Pre vytvorenie skúšobného konta 3 Month free trial použite adresu [www.windowsazure.com](http://www.windowsazure.com). Počas 90 dní máte zdarma k dispozícii:

Výpočtová kapacita	750 small compute hours/mesiac
Webové stránky	10 Shared Web Sites
Relačná databáza	1 GB web edition Windows Azure SQL Database
Úložná kapacita	20 GB zahrňujúca 1 000 000 transakcií
Prenosová kapacita	Unlimited inbound/20 GB Outbound



Dialóg pre vytvorenie skúšobného konta na tri mesiace

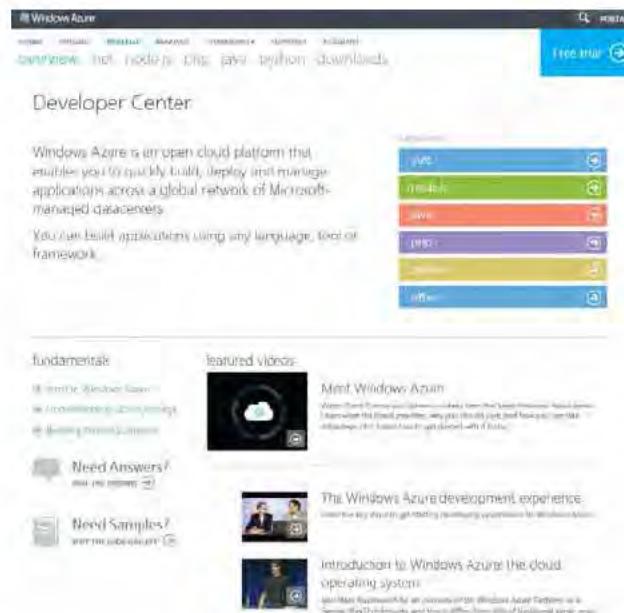
Napriek tomu, že máte garantované, že nebudeťe nič platiť, ani keď prekročíte trojmesačnú lehotu a neupracete po sebe, teda neodstráňte publikované služby, je pri vytváraní skúšobného konta potrebné zadať ako povinný parameter číslo kreditnej karty. Kreditná karta v tomto prípade slúži výhradne na overenie vašej totožnosti a bude z nej stiahnutá maximálne suma 1 USD aby sa overilo, že karta funguje. Tieto peniaze sa vám následne na kartu vrátia. Dôvodom takého postupu je, aby sa výpočtový potenciál týchto účtov nezneužíval na rôzne nekalé účely.



Úvodná obrazovka po vytvorení free trial konta

V spodnej časti úvodnej obrazovky, ktorá sa zobrazí po vytvorení konta sú tri hlavné odkazy – DEVELOP, ACCOUNT a MANAGE.

Kliknutím na odkaz **DEVELOP** získejte informácie a nástroje na konfiguráciu vývojárskeho počítača.

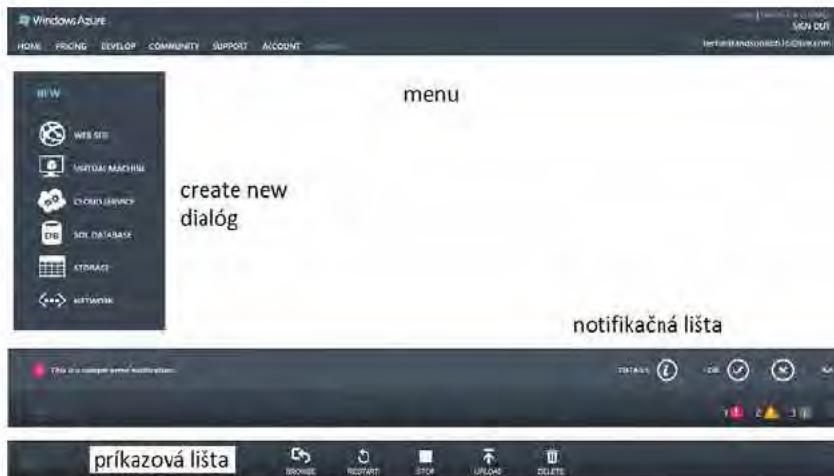


**Develop Center – Windows Azure umožňuje nasadenie aplikácií vyvinutých v rôznych programovacích jazykoch a aplikačných frameworkoch**

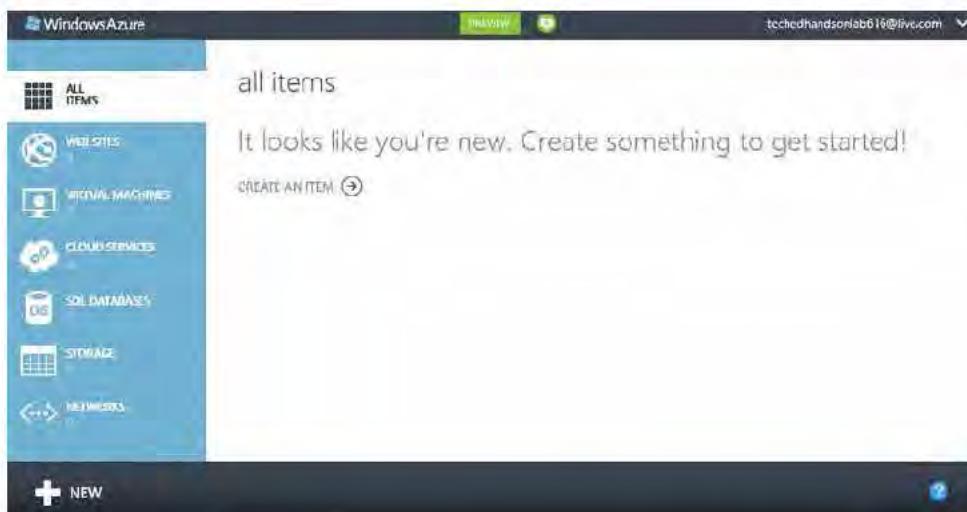


Kliknutím na odkaz **ACCOUNT** získejte prehľad o aktuálne predplatených a využitých službách.

Odkaz **MANAGE**, alebo v pravom dolnom rohu odkaz **MANAGEMENT PORTAL** vás presmeruje na portál pre správu vašich služieb, virtuálnych počítačov, webových stránok...



*Portál pre správu Windows Azure (ako inak) využíva filozofiu Metro dizajnu a ovládania. Kľúčové ovládacie prvky sú hlavné menu v hornej časti stránky a príkazová lišta v spodnej časti stránky. Nad ňou sa môže zobraziť notifikačná lišta s príslušným obrazom. Pri vytváraní nových objektov budete často využívať dialóg Create New.*



*Východiskový stav portálu po vytvorení konta. Kliknutím na hornú čiernu lištu zobrazíte hlavné menu. Kľúčový ovládací prvk je tlačidlo „+ New“ vľavo dole.*

### Skúšobné konto v rámci predplatného MSDN

Vývojári s predplatenou službou MSDN môžu využívať bezplatné konto počas celého trvania predplatného. Objem dostupných prostriedkov závisí od úrovne MSDN, ktorú máte predplatenú. Aj v tomto prípade je potrebné zadať číslo kreditnej karty pre overenie totožnosti no podobne ako pri trojmesačnom skúšobnom konte máte garantovanú nulovú platbu. Ak vyčerpáte pridelené limity, služba bude zvyšok mesiaca obmedzená.

## msdn subscriptions

Search MSDN with Bing

United States (English) Sign out  
lubolacko

msdn

Home My Account Buy Renew or Upgrade Subscriber Downloads My Product Keys Help

**ACCOUNT**

**My Subscriptions**

**Visual Studio Ultimate with MSDN**  
Ľuboslav Laco, llaco@pcrevue.sk  
Subscriber ID:00064078

**Status:** Active  
Expires on 8/25/2012 [Renew](#)

[Remove this Subscription](#) from my account  
[Add an existing subscription](#) to my account  
[Change Windows Live ID](#) for this subscription  
[View Support information](#)

**SUBSCRIPTION BENEFITS**

**Software downloads**  
Over 11 terabytes of Microsoft Products are available to you

[Go to Subscriber Downloads](#) [Go to My Product Keys](#)

---

**Technical support incidents**  
 1. [Set up your support contract](#) (required for the first time use)  
 2. Submit a new incident [online](#) or by phone

---

**Microsoft E-Learning course collections**   
[Register Now!](#)

---

**Windows Azure**   
 1. [Activate Windows Azure](#)  
 2. Go to the [Windows Azure Developer Portal](#)

---

**Priority Support in MSDN Forums**   
[Go to supported MSDN Forums](#)

Windows Azure MSDN môžete aktivovať zo stránky [msdn.microsoft.com](http://msdn.microsoft.com)

**Pri vývoji a testovaní aplikácie na reálnom účte v cloude je výhodné nastaviť „veľkosť“ virtuálneho počítača v ktorom bude vaša aplikácia zatiaľ fungovať na najlacnejšiu verziu „Extra small“ a až pri nasadení do ostrej prevádzky nastaviť projektovanú konfiguráciu virtuálneho stroja.**

[CREATE AN ACCOUNT](#)

### Windows Azure MSDN - Visual Studio Ultimate

[Details](#)

WHAT YOU'LL GET EVERY MONTH		WHAT WE'LL NEED
<b>1500 HRS</b>	Compute 1500 small compute hours	A credit card Required for proof of identity. No obligation to purchase and no charge unless you explicitly remove the spending limit.
<b>45 GB</b>	Storage 45 GB with 100M transactions	
<b>5 DU</b>	SQL Azure Database 5 DU of Web Edition	
<b>35 GB</b>	Data Transfers 35 GB outbound, unlimited inbound	

(1) (2) (3)

Úvodný dialóg pre aktiváciu Windows Azure MSDN

Pre kvalifikovaný odhad nákladov máte v záložke Pricing grafický kalkulátor pre výpočet ceny služby. Na stránke sú štyri grafické „posuvné potenciometre“ pre zadanie požadovaného objemu výpočtovej kapacity (Compute), úložnej kapacity (Database a Blob Storage) a prenosovej kapacity (Bandwidth). Nastavením grafického ukazovateľa zadáte požadovanú kapacitu a vpravo vám kalkulátor zobrazí odhad mesačnej platby. Pri položke Compute zadávate nielen počet inštancií, ale aj veľkosť servera, ktorá sa udáva ako XS, S, M L,XL.

No upfront costs. Pay only for what you use.

**Windows Virtual Machines**

	1	\$9.36	\$0.03/mo
Virtual Machine Type	Small VM (1GHz CPU, 768MB RAM)		

**Linux Virtual Machines**

	0	\$0.00	\$0.00/mo
Virtual Machine Type	Medium VM (1GHz CPU, 768MB RAM)		

**Storage (Geo Redundant)**

	1150GB	\$141.86	
Storage Size	1150GB		

**Bandwidth**

	106GB	\$12.72	
Bandwidth Usage	106GB		

**Full Calculator**

**\$163.95/mo**

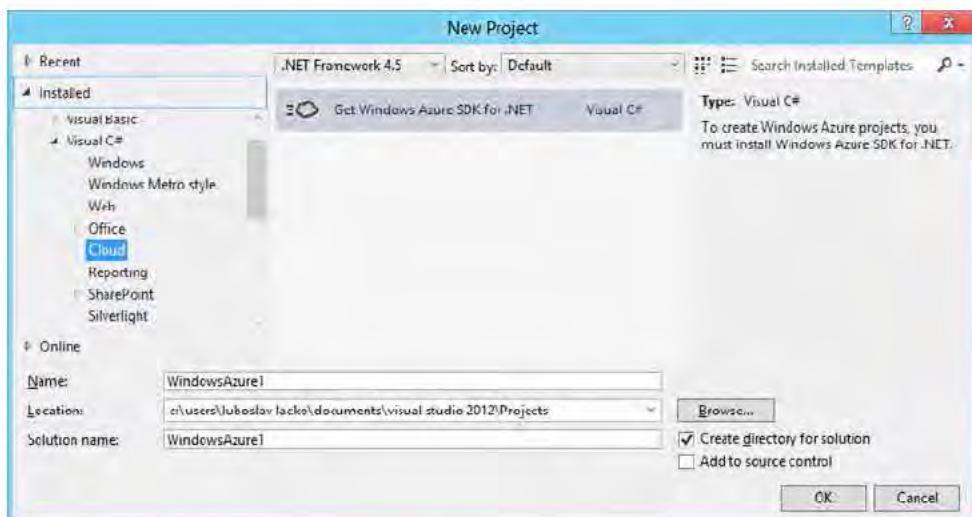
### Grafický kalkulátor pre výpočet ceny služby

Tarifuje sa doba, počas ktorej je inštancia nasadená (nemusí bežať).

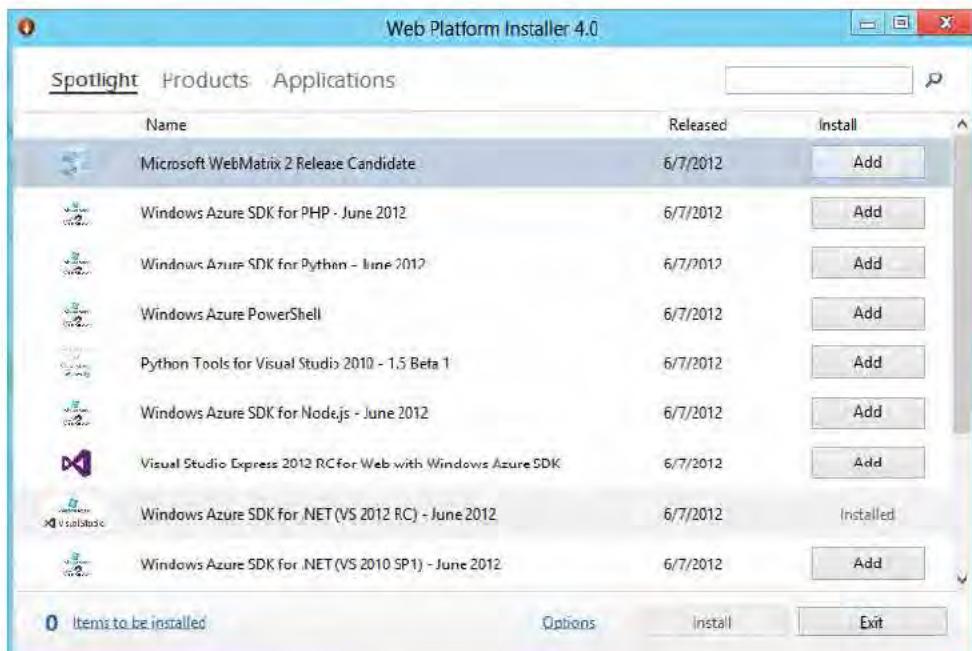
Vo vývojárskej príručke ako vstupný bod budeme predpokladať, že máte zriadené konto a pridelené prístupové privilégia. Po vytvorení konta je vstupný bod do azúrového sveta na adrese <https://windows.azure.com>.

### Konfigurácia vývojárskeho PC

Na vývojárskom počítači (môže to byť aj virtuálny stroj) je potrebné mať nainštalovaný operačný systém Windows 7, Windows Server 2008, alebo Windows Vista SP1 a vývojové prostredie Visual Studio 2012 (v dobe písania publikácie bol tento produkt v štádiu Release Candidate), prípadne staršiu verziu. Do vývojového prostredia je potrebné doinštalovať doplnok Windows Azure SDK. Doplnok vyžaduje funkčnosť IIS, preto je najskôr potrebné nainštalovať (presnejšie povoliť) službu Internet Information Services, a v nej funkčnosť ASP.NET.



*VS 2012 RC – pri prvom prístupe do priečinka clouдовých projektov sa zobrazí ponuka na stiahnutie a nainštalovanie Windows Azure SDK for .NET.*



*Široká paleta SDK pre rôzne vývojárske platformy*

## Praktické príklady

Po aspoň letomom preštudovaní základných informácií o platforme Windows Azure, odporúčame vyskúšať si čo najviac čŕt v praxi.

### Vytvorenie nového virtuálneho serveru

Pre vytvorenie nového virtuálneho serveru použite Windows Azure Portal (<https://manage.windowsazure.com>). Aktivujte položku menu New | Virtual Machine | From Gallery.



### Vytvorenie nového virtuálneho servera z galérie

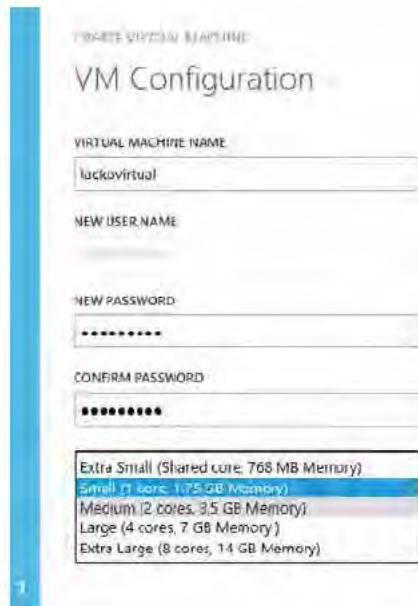
V galérii je bohatá ponuka konfigurácií virtuálnych serverov nielen s operačným systémom Windows Server 2008 R2, či Windows Server 2012, ale v ponuke sú aj serverové platformy na báze Linuxu (SUSE, Ubuntu).



### Ponuka „image“ pre vytvorenie virtuálnych strojov

Vytvorte napríklad virtuálny server s operačným systémom Windows Server 2008 R2.

V nasledujúcim kroku je potrebné zadat názov virtuálneho servera a prihlásovacie heslo. V dolnej časti dialógu si všimnite pravok pre výber výkonových parametrov, teda frekvencie CPU a kapacity pamäte.



### Konfigurácia virtuálneho stroja

V nasledujúcej záložke konfiguračného dialógu ponechajte označenú voľbu **Standalone Virtual Machine**, zadajte DNS, ponechajte voľbu **Use Automatically Generated Storage Account**, a vyberte región kde bude v dátovom centre váš virtuálny stroj fyzicky umiestnený.



### Konfigurácia DNS a výber regiónu globálneho dátového centra, kde váš virtuálny stroj pobeží

Vytvorenie virtuálneho stroja bude trvať nejaký čas (v našom prípade približne 10 minút). Po vytvorení sa zobrazí v zozname virtuálnych strojov Windows Azure portálu. Na aplikačnej liště v dolnej časti obrazovky je ponuka funkcií pre prácu s aktuálne vybraným virtuálnym serverom.

virtual machines PREVIEW

VM INSTANCES IMAGES DISKS

NAME	STATUS	SUBSCRIPTION	LOCATION	DNS NAME
lackovirtual	<span style="color: green;">Running</span>	Azure Trial Account	West Europe	http://lackovirtual.cloudapp.net

**LINKS:** ALL ITEMS WEBSITES VIRTUAL MACHINES CLOUD SERVICES SQL DATABASES STORAGE NETWORKS

**Actions:** NEW CONNECT RESTART SHUTDOWN ATTACH DETACH DISK CAPTURE DELETE ?

### Zoznam virtuálnych strojov

**lackovirtual**

DASHBOARD ENDPOINTS CONFIGURE

CPU PROVISIONING: 0.25 vcores | DATA IN: 0.000000B | DATA OUT: 0.000000B | TASK STATE THRESHOLD: 0% | DISK WRITE THRESHOLD: 0%

1 HOUR

Usage Overview

LACCOVIRTUAL OTHER ROLE AVAILABLE

LACCOVIRTUAL 67% at 11:00:00

DISKS

DISK TYPE VHD

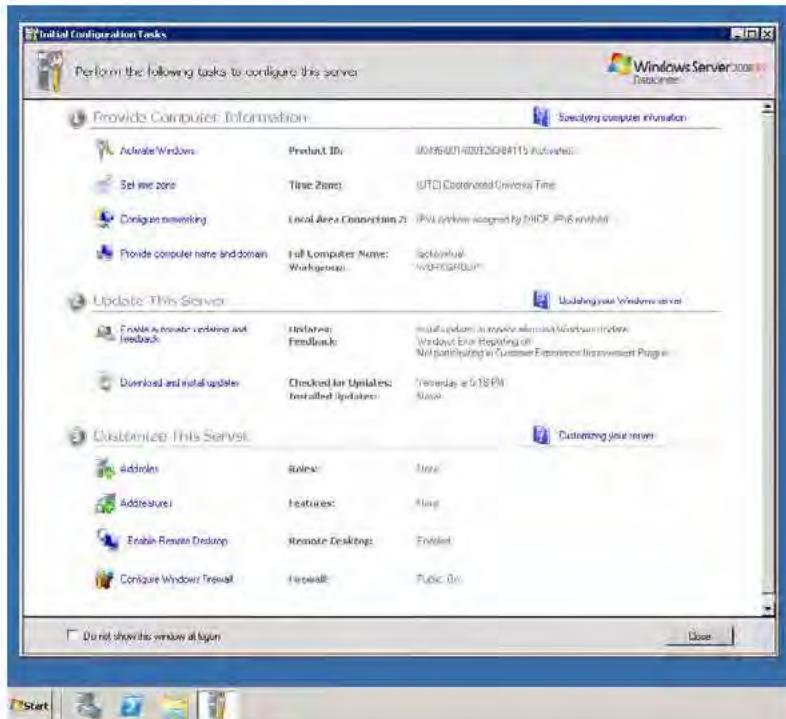
lackovirtual.vhd (0.25 vcores) Q32016

STATUS: Running  
URI: http://lackovirtual.cloudapp.net/  
HOST NAME: lackovirtual  
PUBLIC VIRTUAL IP ADDRESS (VIP): 160.63.15.157  
ROLE: Standard  
OS: Windows Server 2012 R2 Datacenter

**Actions:** NEW CONNECT RESTART SHUTDOWN ATTACH DETACH DISK CAPTURE DELETE ?

### Dashboard pre zobrazenie parametrov a stavu virtuálneho stroja

Pre pripojenie sa k virtuálnemu stroju je potrebné aktivovať funkciu CONNECT na aplikačnej lište. Týmto krokom sa na lokálny počítač stiahne súbor s príponou RDP pre pripojenie sa cez Remote Desktop Protocol.



Pripojenie sa k virtuálnemu počítaču cez RDP.

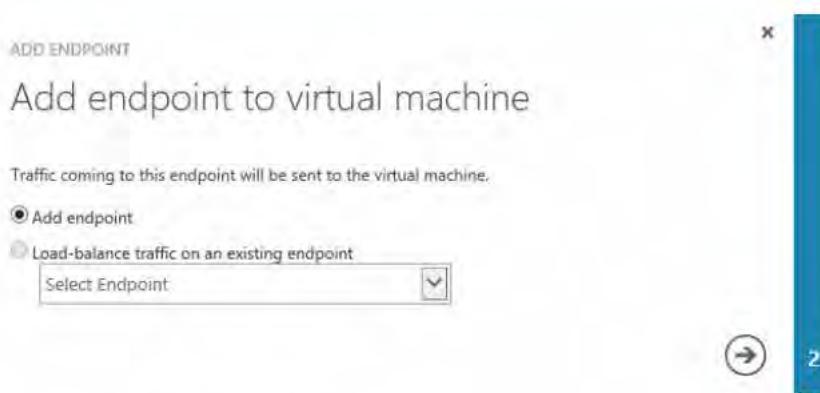
## Rozdelenie záťaze medzi viaceré virtuálne stroje

Ak si chcete vyskúšať prerozdelenie záťaze medzi viaceré virtuálne servery, napríklad pre IIS, je potrebné vytvoriť ďalšie servery rovnakým postupom, s jedinou výnimkou. V treťom konfiguračnom dialógu označte voľbu Connect to existing Virtual Machine a vyberte predtým vytvorený virtuálny stroj. V poli Storage Account vyberte voľbu Use Automatically Generated Storage Account. Zvyšok konfiguračného procesu je rovnaký ako v predchádzajúcim príklade.



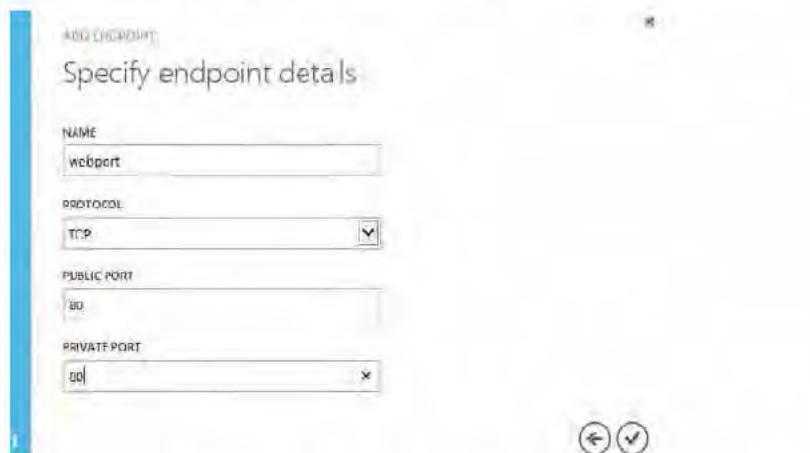
Definovanie parametrov pripojenia k inému virtuálnemu stroju

Po vytvorení druhého virtuálneho stroja je potrebné vytvoriť endpoint na port 80. Vyberte prvý virtuálny stroj, kliknite na položku Dashboard a aktivujte voľbu Endpoints. Vytvorte nový Endpoint.



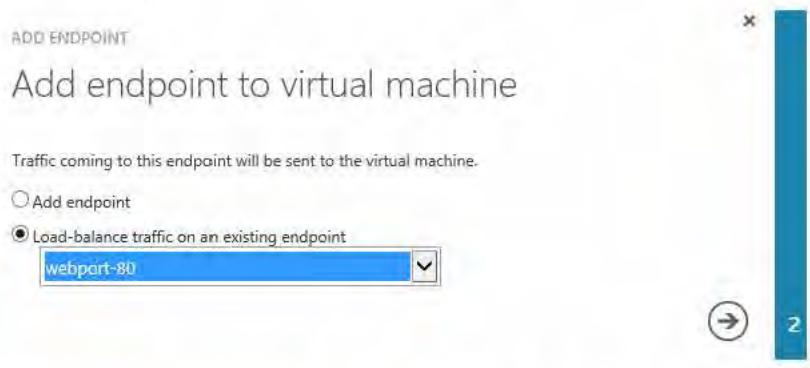
### Vytvorenie nového Endpointu

Zadajte názov endpointu webport, zvoľte protokol TCP a nastavte obidva porty, teda Public Port aj Private Port na hodnotu 80. Po potvrdení sa vytvorí nový Endpoint. Podobne ako pri vytváraní virtuálneho stroja, aj táto akcia trvá niekoľko minút.



### Konfigurácia parametrov nového Endpointu pre prvý virtuálny stroj

Aby fungovalo rozdeľovanie záťaze, je potrebné vytvoriť Endpoint aj na druhom virtuálnom stroji. Vyberte druhý virtuálny stroj, chodťte na stránku Dashboard a aktivujte voľbu Endpoints. Označte voľbu Load Balance Traffic On An Existing Endpoint. V combo boxe vyberte Endpoint vytvorený pre prvý virtuálny stroj.



### Konfigurácia parametrov Endpointu pre druhý virtuálny stroj

Aj druhý endpoint pomenujte webport a nastavte Private Port na hodnotu 80.

## Edit endpoint

NAME  
webport

PROTOCOL  
TCP

PUBLIC PORT  
80

PRIVATE PORT  
80

LOAD-BALANCED MACHINE(S)  
lackovvirtual2, lackovvirtual1

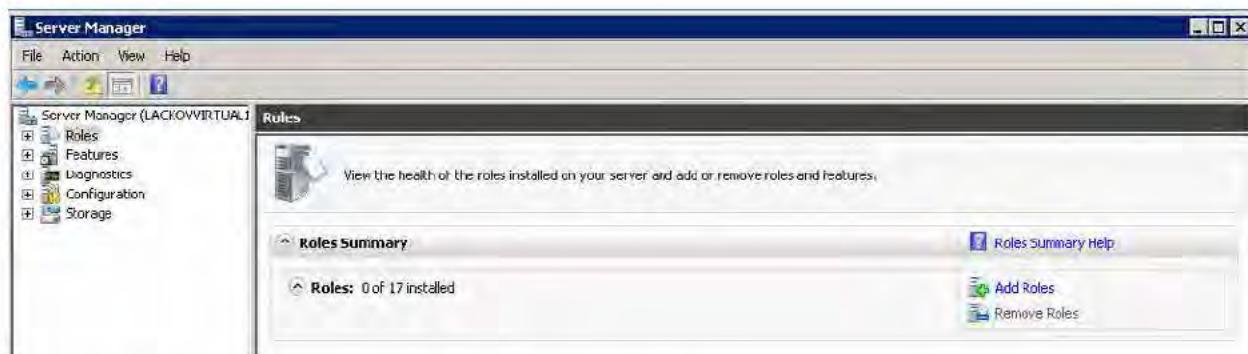
+ ADD ENDPOINT   
 ✎ EDIT ENDPOINT   
 ✖ DELETE ENDPOINT

*Po vytvorení druhého endpointu môžete pomocou položky Edit Endpoint na aplikačnej lište skontrolovať, či je nastavené vyrovnanie záťaže*

### Konfigurácia IIS na virtuálnych serveroch pre vyrovnanie záťaže

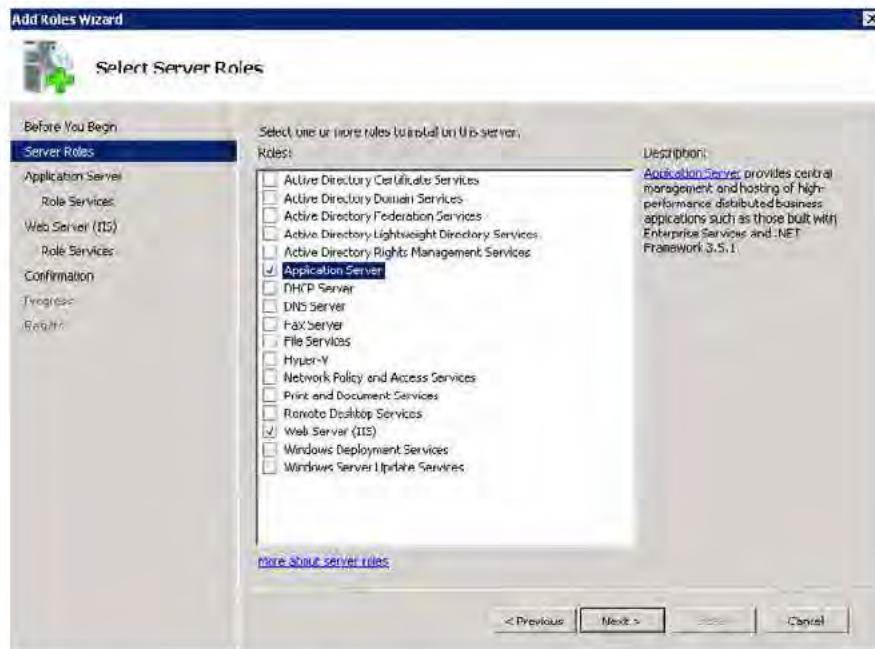
Pripojte sa prostredníctvom RDP na prvý virtuálny server. Rozmenené na drobné, zobrazte zoznam virtuálnych strojov, označte prvý z nich a na aplikačnej lište v spodnej časti obrazovky aktivujte funkciu Connect. Prihláste sa pomocou administrátorského konta.

Najskôr pridáme rolu IIS (Internet Information Server). Vo virtuálnom stroji spustite Server Manager (menu Start | All Programs | Administrative Tools). V priečinku Roles aktivujte funkciu Add Roles.



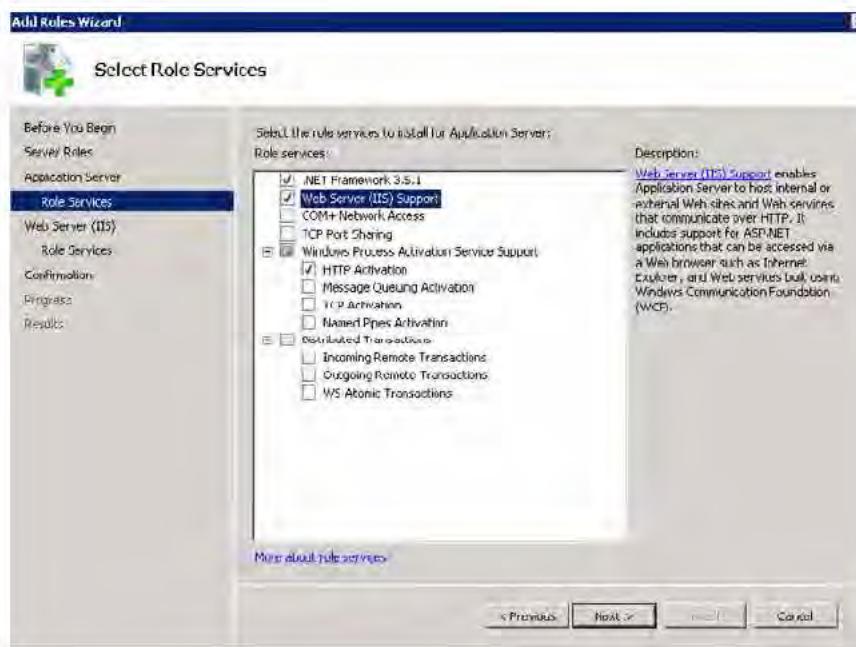
### Pridávanie rolí prostredníctvom aplikácie Server Manager

V dialógu Roles page pridajte dve role – Application Server a Web Server (IIS). Zobrazí sa upozornenie na nutnosť doinštalovania potrebných komponentov, v tomto prípade .NET Framework 3.5.1. Tento návrh samozrejme akceptujte.



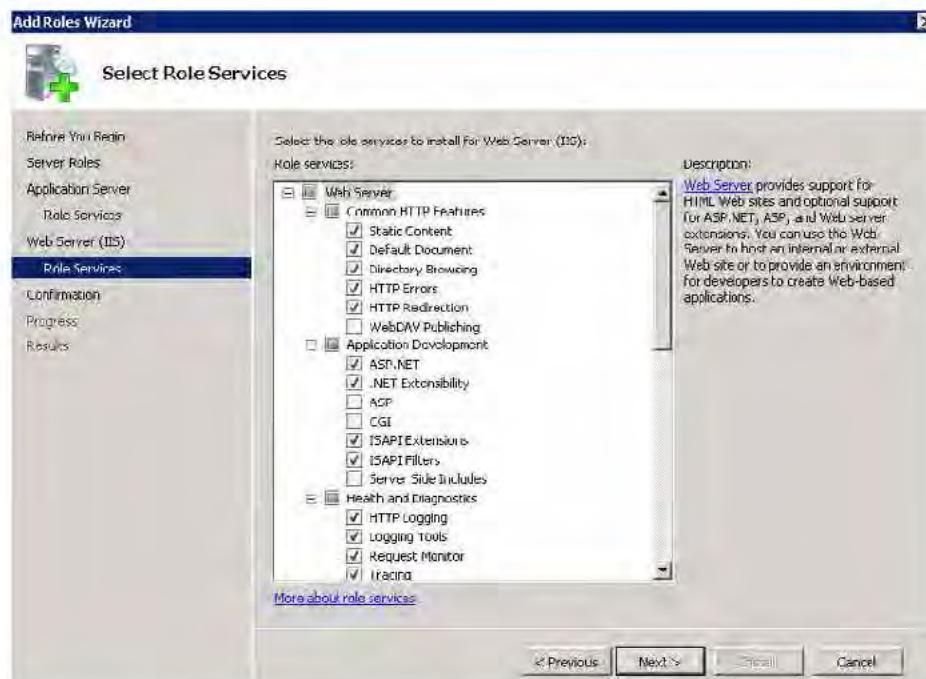
### Pridanie rolí Application Server a Web Server

Okrem technologickej platformy .NET Framework 3.5.1 je potrebné označiť aj položku Web Server (IIS) Support. Znovu je potrebné súhlasiť s inštaláciou doplnkových komponentov.



### Výber inštalácie potrebných komponentov

V nasledujúcom dialógu sa zobrazí zoznam inštalovaných služieb. Tento zoznam potvrďte tlačidlom Next.



### Výber inštalácie potrebných komponentov

Týmto krokom sme skončili prípravnú fázu pridania role aplikačného a webového servera do prvého virtuálneho servera. Môžete rozpojiť RDP pripojenie.

Rovnaký postup zopakujte aj na druhom virtuálnom serveri

Webovú aplikáciu musíte pochopiteľne publikovať na obidve inštancie virtuálnych serverov. Pri behu aplikácie sa o rovnomerné rozdelenie záťaže automaticky stará modul Network Load Balancer.



### Princíp rozdeľovania záťaže na jednotlivé serverové inštancie

#### **Vytvorenie virtuálneho servera s nainštalovaným serverom SQL Server**

Táto úloha bude v mnohých prípadoch logickým pokračovaním predchádzajúceho príkladu, v ktorom sme vytvorili dva virtuálne servery a nakonfigurovali sme vyrovnanie záťaže medzi IIS Servermi. Väčšina webových aplikácií však pracuje s údajmi a tak budeme potrebovať vytvoriť databázový server.

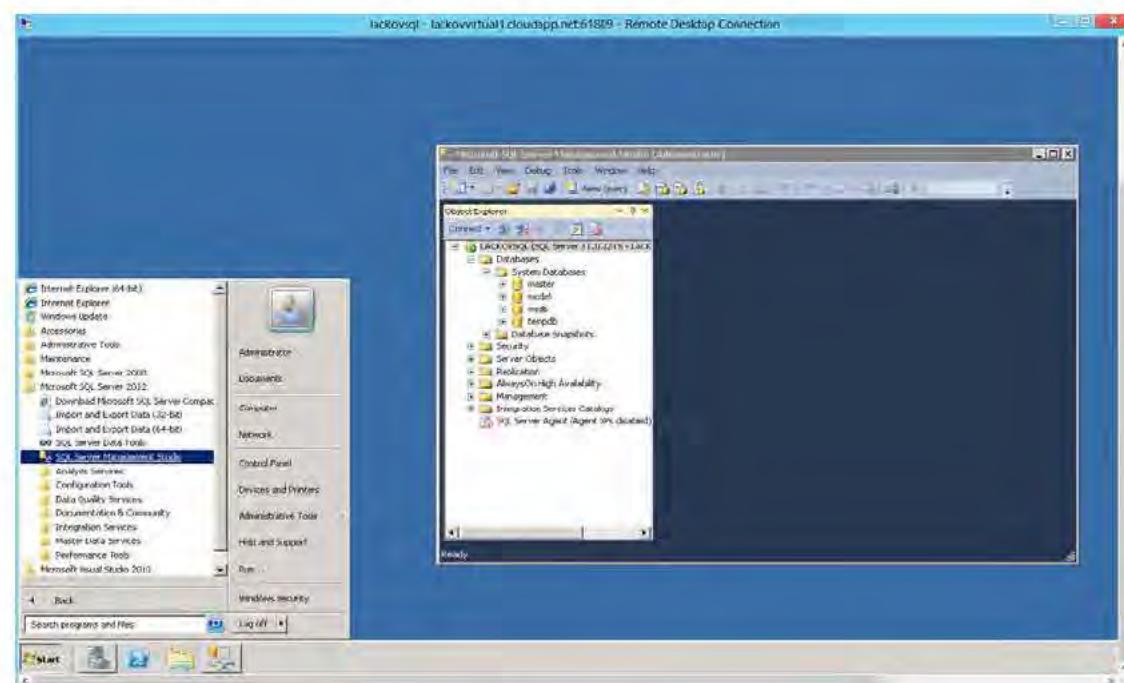
Postupujte rovnako ako pri vytváraní virtuálnych serverov v predchádzajúcich statiach. Na webe Windows Azure Portal (<https://manage.windowsazure.com>) aktivujte položku menu New | Virtual

**Machine | From Gallery a vyberte image virtuálneho serveru s predinštalovaným serverom SQL Server 2012.**



#### Výber image virtuálneho serveru s predinštalovaným serverom SQL Server 2012

Ďalej postupujte ako pri vytváraní klasického virtuálneho servera. V treťom dialógu predefinujte pripojenie na prvý z dvojice predtým vytvorených virtuálnych serverov s IIS. o vytvorení virtuálneho servera sa môžete k nemu pripojiť cez RDP a spustiť nástroj SQL Server Management Studio pre správu databázy a vytváranie databázových objektov.



*Spustenie nástroja SQL Server Management Studio*

Prvým krokom by bolo vytvorenie novej databázy pre webovú aplikáciu. Skôr než tak urobíte, je potrebné zamyslieť sa nad tým, kam databázu umiestnite. V našom prípade sme vytvorili konfiguráciu typu Small, ktorá disponuje diskom C s kapacitou 30 GB a temporary diskom D s kapacitou 70 GB. Pre stredné a väčšie objemy údajov je potrebné vytvoriť pre databázu nový disk. Odporúčame vytvoriť dva disky vhodnej veľkosti, jeden pre údaje a druhý pre redo log súbory. Postup je popísaný v ďalšej stati.

### Pripojenie diskov k virtuálnemu stroju

Vyberte v zozname virtuálny stroj, ku ktorému chcete pripojiť diskový image, na aplikačnej lište v dolnej časti obrazovky aktivujte tlačidlo **Attach** a vyberte možnosť **Attach Empty Disk**. V dialógu pre pripojenie virtuálneho disku je potrebné definovať jeho kapacitu.

Attach empty disk to virtual machine

VIRTUAL MACHINE NAME  
lackovsql

STORAGE LOCATION  
http://portalvhdsqcdffsbdy8qv3.blob.core.windows.net

FILE NAME  
lackovvirtual1-lackovsql-0704-1

SIZE (GB)  
50



*Pripojenie disku k virtuálnemu serveru*



## usage overview



## disks

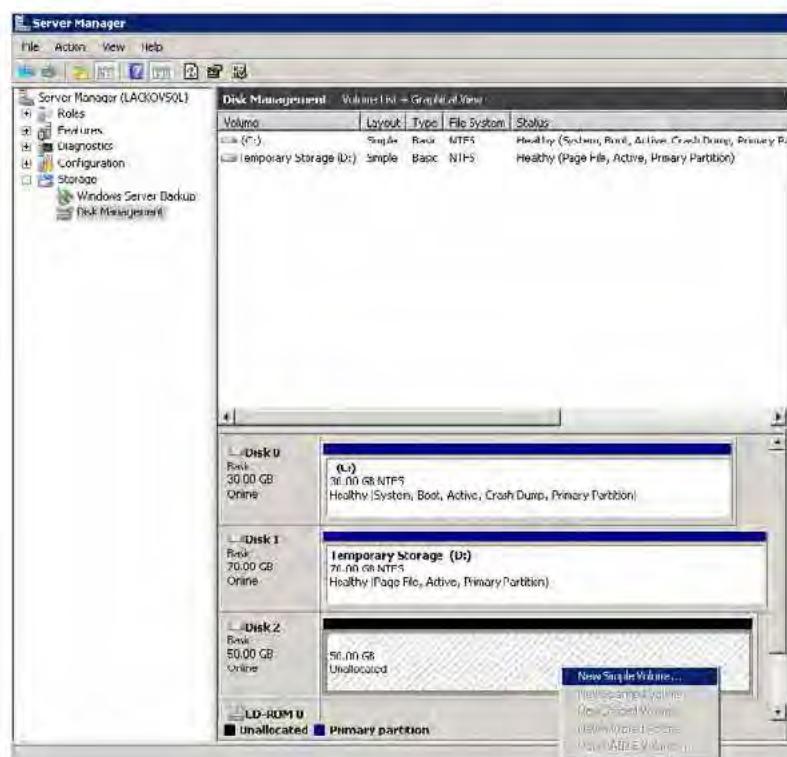
DISK	TYPE	VHD
lackovvirtual1-lackovsql-0-201207042...	OSDisk	http://contalvhdsqcrifftsboy8qv3.blob.c...
lackovvirtual1-lackovsql-0-201207042...	DataDisk	https://partehvhdsqcrifftsboy8qv3.blob.c...

## quick glance

STATUS	Running
URI	http://lackovvirtual1.cloudapp.net/
HOST NAME	lackovsql
PUBLIC VIRTUAL IP ADDRESS (VIP)	168.63.10.252
SIZE	Small
DRIVES	1

Pripojený disk sa zobrazí v zozname diskov

Posledným krokom je pripojenie sa k virtuálnemu serveru cez RDP a v nástroji Server Manager pridanie pripojeného disku do konfigurácie servera.



Pridanie pripojeného disku do konfigurácie virtuálneho servera

## Webová aplikácia vo Windows Azure – praktický príklad

Námetom príkladu bude vytvorenie jednoduchej webovej aplikácie, ktorá bude pracovať s údajmi. Je potrebné rozlišovať, v tomto príklade ide o model nasadenia WEB SITE. V nasledujúcom príklade budeme migrovať existujúcu webovú aplikáciu do cloubovej služby.

### Vytvorenie ASP.NET MVC 4 webovej aplikácie

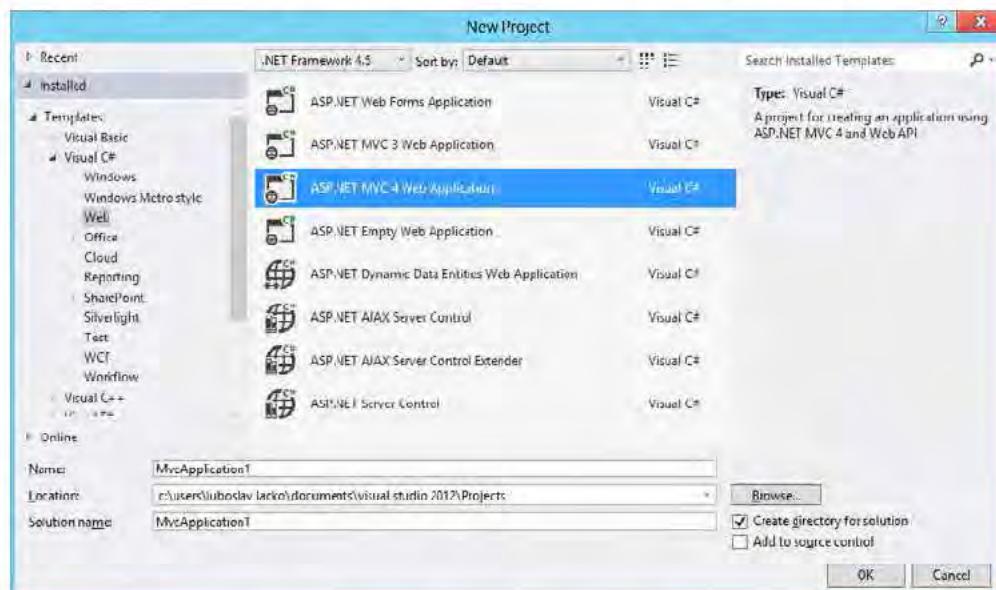
V nástroji Visual Studio 2012 vytvorte projekt typu ASP.NET MVC 4 Application. Šablónu pre príslušný programovací jazyk (C#, prípadne Visual Basic) nájdete v priečinku Web.

*Pre zaujímavosť MVC (Model View Controller) je relativne nový typ aplikačného rámcu pre ASP.NET aplikácie, ktorý umožňuje dôslednejšie oddelenie aplikačnej logiky od prezentácej vrstvy.*

*Architektúra MVC pozostáva z troch komponentov, pričom výhodou je, že zmena ktoréhokoľvek z nich má len minimálny vplyv na ostatné, preto sa využíva hlavne pri rozsiahlych a zložitých systémoch.*

- *Model – slúži na uchovávanie a aktualizáciu stavu aplikácie, poskytuje prístup k dátam, a umožňuje ich ukladanie a aktualizáciu.*
- *View – ako vyplýva z názvu je to „pohľad“ na údaje. Jeho hlavnou úlohou je prezentácia a formátovanie údajov z modelu. Pre jeden model môže existovať viaceré pohľadov.*
- *Controller – alebo po našom „riadiaca jednotka“ tvorí akési prepojenie medzi modelom a riadiacou jednotkou, má na starosti riadenie aplikácie, spracováva vstupy od používateľov, vyžiada si údaje od modelu a pošle aktuálny View späť užívateľovi.*

V našom príklade sme vytvorili v jazyku c# projekt s názvom AzurovyWeb.



Vytvorenie projektu typu ASP.NET MVC 4 Application

Poznámka. V dobe písania bola podporovaná len verzia .NET Framework 4.0

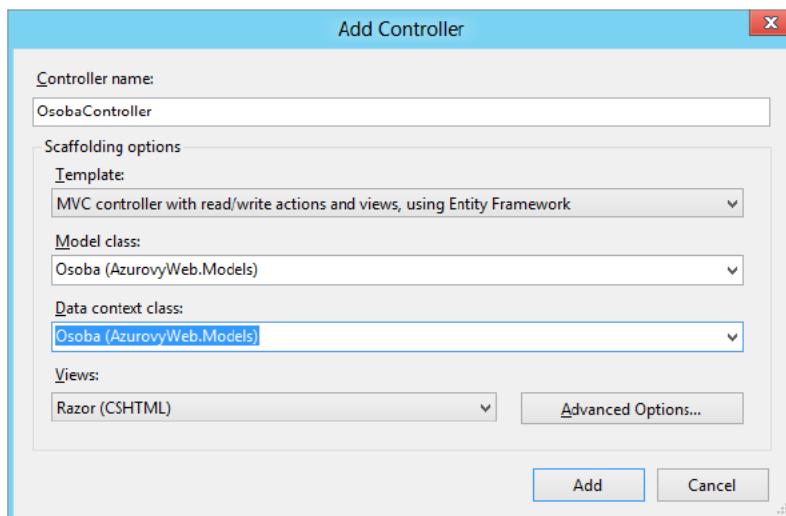
V nasledujúcom dialógu vyberte voľbu Internet Application.

V okne Solution Explorer kliknite pravým tlačidlom myši na priečinok Models a aktivujte voľbu Add | Class. Vytvorte napríklad triedu Osoba.cs. Pre novovytvorenú triedu doplňte niekoľko základných atribútov, napríklad:

```
public class Osoba
{
    public int ID { get; set; }
    public string meno { get; set; }
    public string priezvisko { get; set; }
}
```

Aktivujte menu Build | Build Solutions, prípadne použite klávesovú skratku CTRL+SHIFT+B.

V okne Solution Explorer kliknite pravým tlačidlom myši na priečinok Controllers a aktivujte voľbu Add | Controller. Controller pomenujte. Zo zoznamu vyberte voľbu MVC Controller with read/write actions and views, using Entity Framework option. V rozbaľovacom zozname vyberte triedu, ktorú ste vytvorili v predchádzajúcom kroku. V poli Data context class list vyberte <New data context...> a zadajte vhodný názov. Uistite sa, či je vybratá voľba Razor.



#### *Vytvorenie projektu typu ASP.NET MVC 4 Application*

V okne Solution Explorer otvorte controller OsobaController.cs. Metódy pre CRUD (Create, Read, Update Delete) sa vygenerovali automaticky:

```
// POST: /osoba/Create

[HttpPost]
public ActionResult Create(Osoba osoba)
{
    if (ModelState.IsValid)
    {
        db.Osobas.Add(osoba);
        db.SaveChanges();
    }
    return RedirectToAction("Index");
}

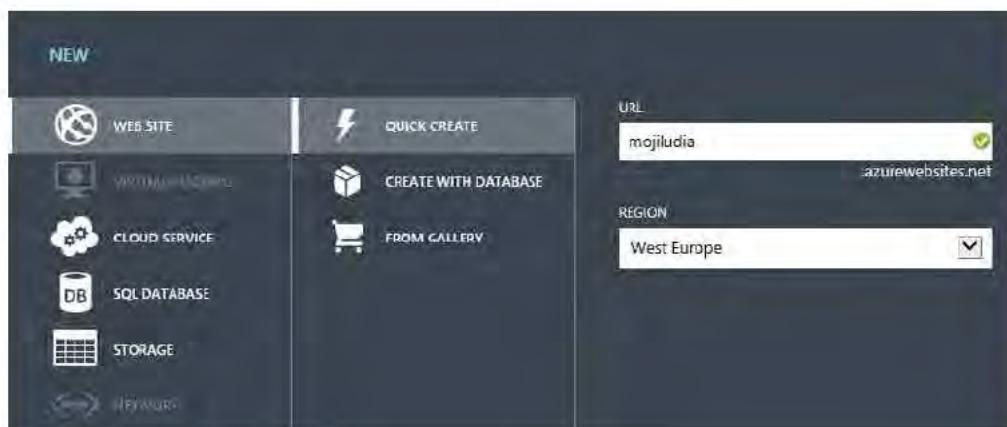
return View(osoba);
}

// GET: /Osoba/Edit/5
```

```
public ActionResult Edit(int id = 0)
{
    Osoba osoba = db.Osobas.Find(id);
    if (osoba == null)
    {
        return HttpNotFound();
    }
    return View(osoba);
}
```

### Vytvorenie Azure Web Site

Teraz sa dejstvo presunie na Management Portal Windows Azure. Vyberte záložku **Web Site**, kliknite na tlačidlo **New** a aktivujte voľbu **Quick Create**. Vyberte vhodné a hlavne jedinečné URL a potvrďte tlačidlom **Create Web Site**.



### Vytvorenie novej webovej stránky

NAME	STATUS	SUBSCRIPTION	LOCATION	URL
mojiludia		Microsoft Free Trial	West Europe	<a href="http://mojiludia.azurewebsites.net">http://mojiludia.azurewebsites.net</a>

Nová stránka v zozname webových stránok Management portálu

Ak kliknete na adresu URL, v našom prípade <http://mojiludia.azurewebsites.net> zobrazí sa obsah práve vytvorenej stránky.

# This web site has been successfully created

There's nothing here yet, but Windows Azure makes it simple to publish content with GIT, TFS, FTP or your favorite development tool.

[tell me more](#)



*Kliknutím na adresu URL sa dostanete na novovytvorenú webovú stránku*

Kliknutím na názov stránky v zozname sa zobrazí dashboard pre danú webovú stránku. Aktivujte voľbu **Download the publish profile**. Týmto úkonom sa na lokálny vývojársky počítač stiahne informácia (vo formáte XML) pre vypublikovanie webovej aplikácie.

*Dashboard novo vytvorenej stránky*

## Konfigurácia clouдовého úložiska SQL Database

Webová aplikácia bude ukladať údaje do relačnej databázy v úložisku SQL Database. Ak ešte nemáte vytvorenú vhodnú inštanciu cloudovej databázy, je potrebné ju vytvoriť. Aby ste mohli vytvoriť inštanciu databázy, musíte mať vytvorený databázový server. Všimnite si v hornej časti dva odkazy DATABASES a SERVERS. Prepnite sa do záložky SERVERS.

The screenshot shows the Windows Azure Management Portal interface. On the left, there's a sidebar with icons for All Items, Web Sites, Cloud Services, SQL Databases (which has 0 items), and Storage. The main area is titled "sql databases". Below it, tabs for "DATABASES" and "SERVERS" are shown, with "SERVERS" being the active tab and highlighted with a red box. A message says "You have no SQL servers. Create one to get started!" followed by a "CREATE A SQL SERVER" button with a red arrow pointing to it.

*Management portal – sql databases*

CREATE SERVER

### Database server settings

LOGIN NAME

lubolacko



LOGIN PASSWORD

\*\*\*\*\*

LOGIN PASSWORD CONFIRMATION

\*\*\*\*\*

REGION

West Europe



Allow Windows Azure services to access the server

### Dialóg pre vytvorenie databázového servera

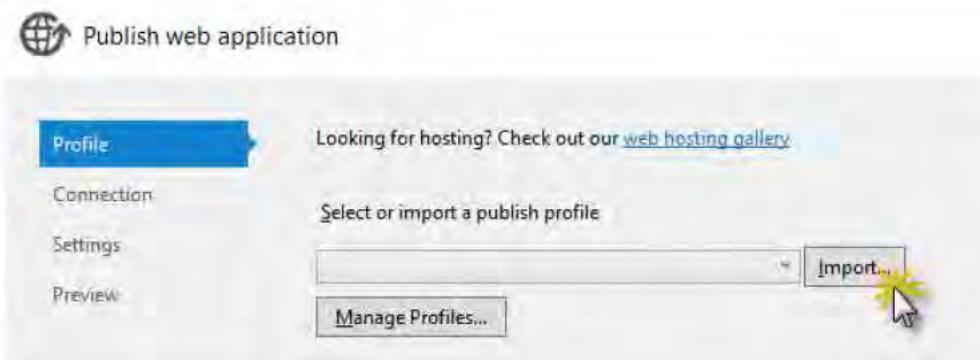
Zatiaľ nebudem vytvárať databázu, sústredíme pozornosť na pripojenie sa k databáze z prostredia Visual Studio. Musíte zahrnúť svoju lokálnu IP adresu do zoznamu povolených IP adries servera. Skopírujte v záložke Configure IP adresu z poľa Current Client IP Address do polí Start IP Address a End IP Address a toto pravidlo nejako pomenujte. Nezabudnite nastavenie uložiť tlačidlom Save na aplikačnej lište.

The screenshot shows the Windows Azure portal interface. At the top, there's a blue header bar with the Windows Azure logo and the word "Windows Azure". Below it, the main navigation bar includes "PREVIEW", "DASHBOARDS", "DATABASES", and "CONFIGURE". On the left, a sidebar titled "Servers" shows a single entry: "jaew4swvs7". The main content area is titled "jaew4swvs7" and contains two sections: "allowed ip addresses" and "allowed services". Under "allowed ip addresses", there is one entry: "pravidlo\_1" with IP address "87.197.98.123" and a "Remove" button. Below this, the "CURRENT CLIENT IP ADDRESS" is listed as "87.197.98.123". Under "allowed services", there are two options: "WINDOWS AZURE SERVICES" (selected) and "NO".

*Konfigurácia IP adries pre prístup k databázovému serveru*

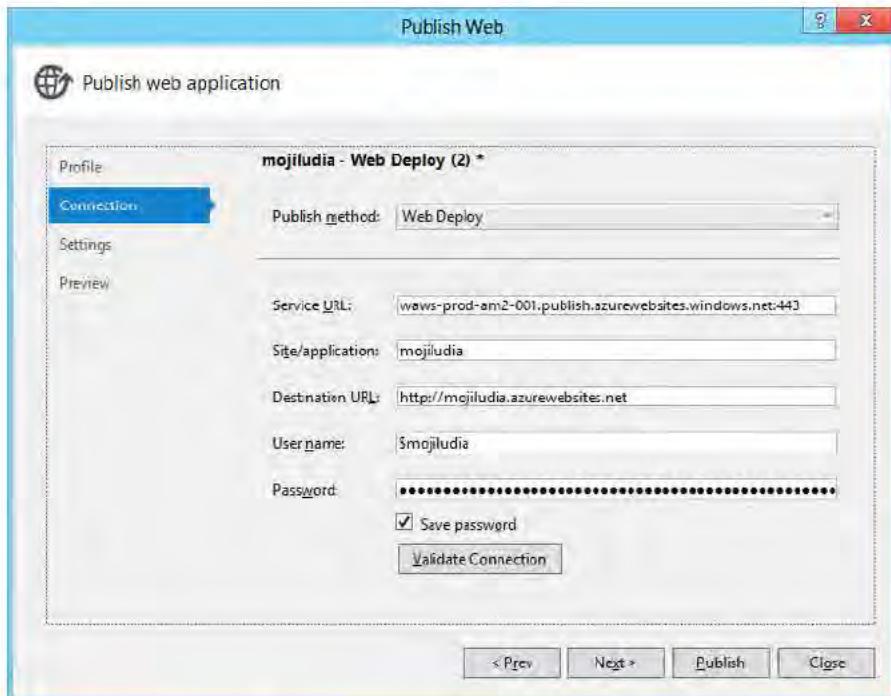
### **Publikovanie ASP.NET MVC 4 aplikácie prostredníctvom Web Deploy**

Vráťme sa do vývojového prostredia Visual Studio 2012 a v okne **Solution Explorer** aktivujte položku kontextového menu **Publish**. V dialógu nahrajte profil, ktorý ste si v jednom z predchádzajúcich krokov stiahli na lokálny vývojársky počítač.



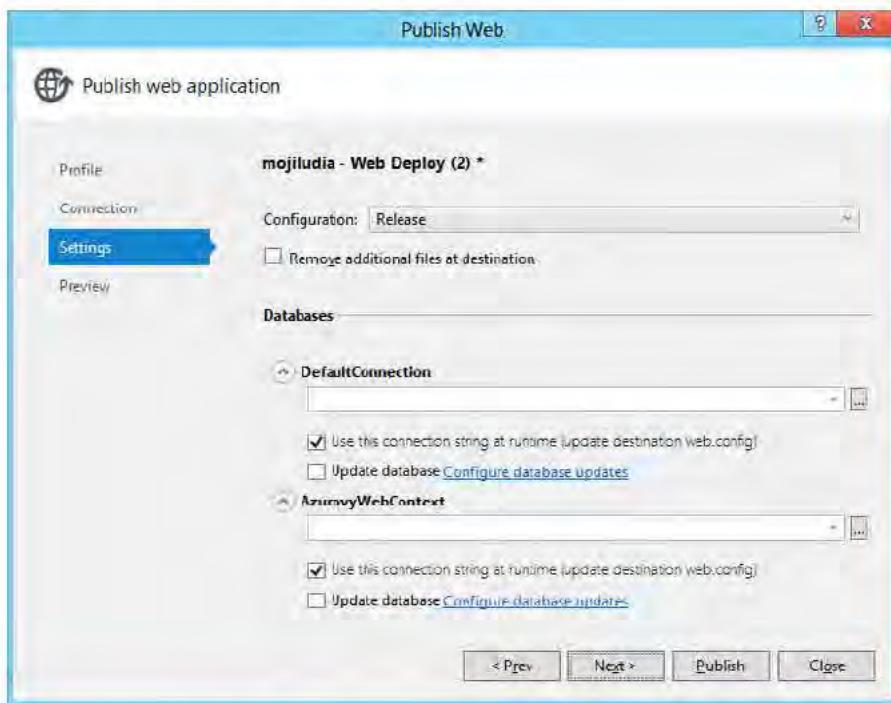
*Import profilu pre publikovanie*

Následne sa zobrazia informácie z profilu. Informácie potvrďte tlačidlom **Validate Connection**.



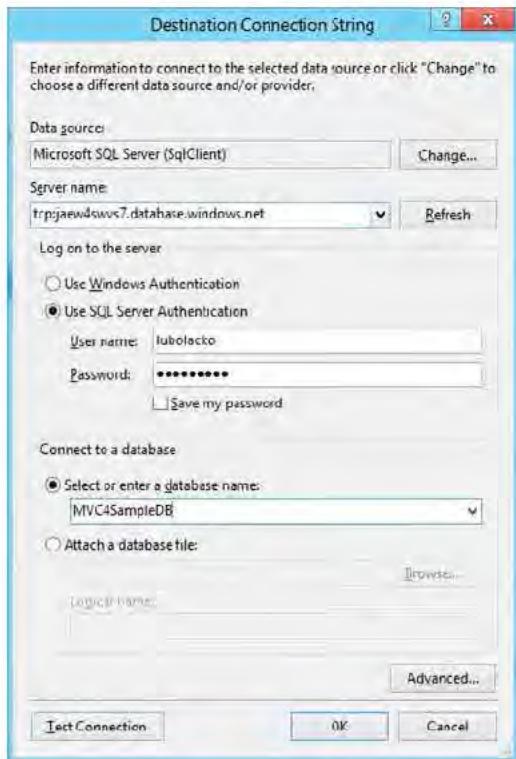
### *Profil pre publikovanie*

V ďalšom kroku nastavíme reťazec na pripojenie k databázovému serveru. Kliknite na symbol tlačidla s tromi bodkami vedľa názvu xxxWebContext.



### *Nastavenie profilu – databáza*

Názov SQL servera zadajte s prefixom tcp, v tvare: `tcp:xxxxxx.database.windows.net`.

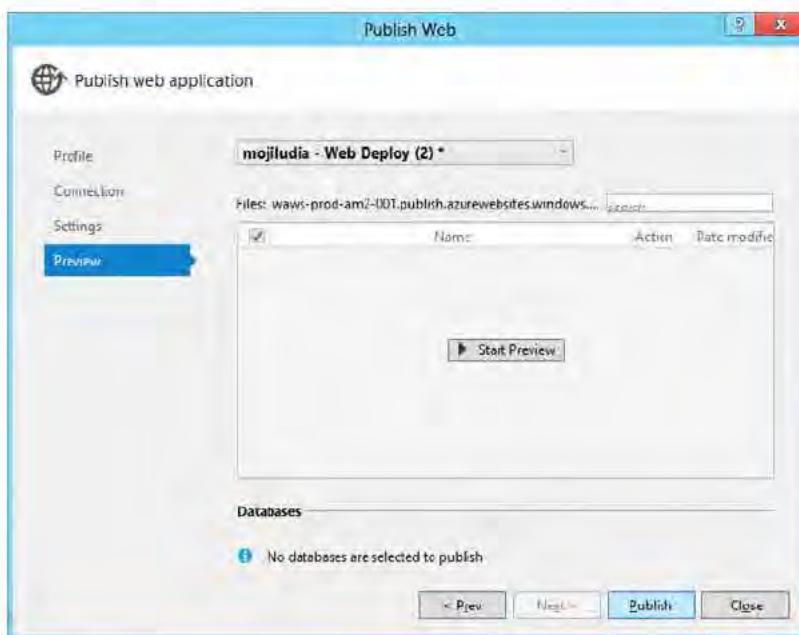


#### *Parametre pre pripojenie sa k databázovému serveru v cloude*

Po úspešnom pripojení databázový server zistí, že databáza s vami zadaným názvom nie je vytvorená a ponúkne vám jej vytvorenie.

**Poznámka:** Skopírujte si pripojovací reťazec pre ďalšie použitie.

V záložke Preview vypublikujte aplikáciu a môžete ju spustiť.



#### *Publikovanie webovej aplikácie do cloutu*

Your logo here

[Register](#) [Log in](#)[Home](#) [About](#) [Contact](#)

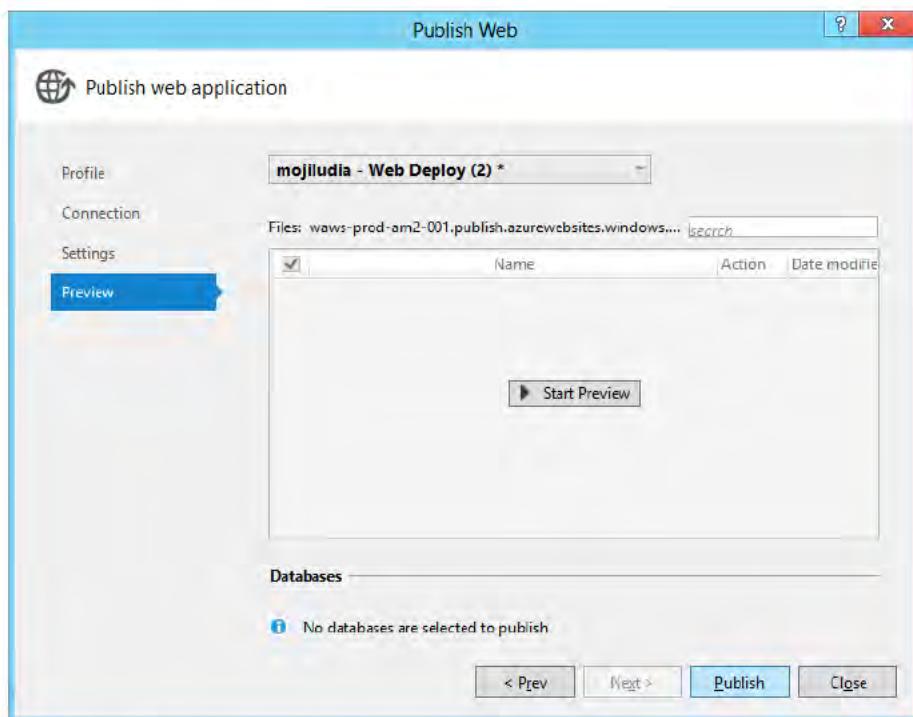
## Index

[Create New](#)**Meno Priezvisko**

© 2012 - My ASP.NET MVC Application



*Spustenie aplikácie. Zobrazí sa úvodná stránka. Následne doplňte za adresy URL /Osoba.*



**Záložka Preview**

### Migrácia webovej ASP.NET aplikácie do clodu

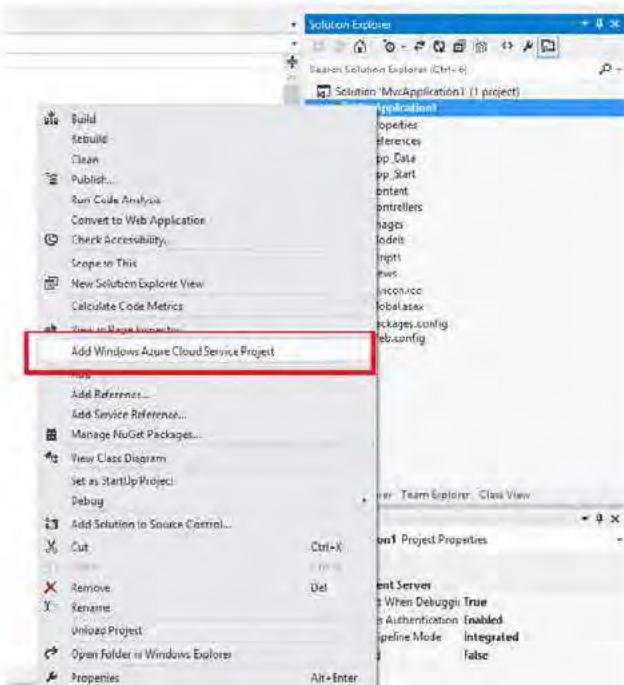
Na rozdiel od predchádzajúcej aplikácie, ktorá fungovala v móde WEB SITE, tento projekt webovej aplikácie bude migrovaný do CLOUD SERVICE. Ak nemáte vhodnú ASP.NET aplikáciu, s ktorou by ste to sami skúsili vytvorte si novú. Vo prostredí Visual Studio 2012, prípadne v staršej verzii 2010 vytvorte projekt podľa šablóny **ASP.NET MVC 4 Web application**, ktorú pre príslušný programovací jazyk (C#, prípadne Visual Basic) nájdete v priečinku **Web**.

Aplikáciu pre zaujímavosť zostavte a spusťte. Znovu zdôrazňujeme, že ste vytvorili webovú aplikáciu, ktorá zatiaľ nemá s cloudom nič spoločné.



### *Test webovej aplikácie*

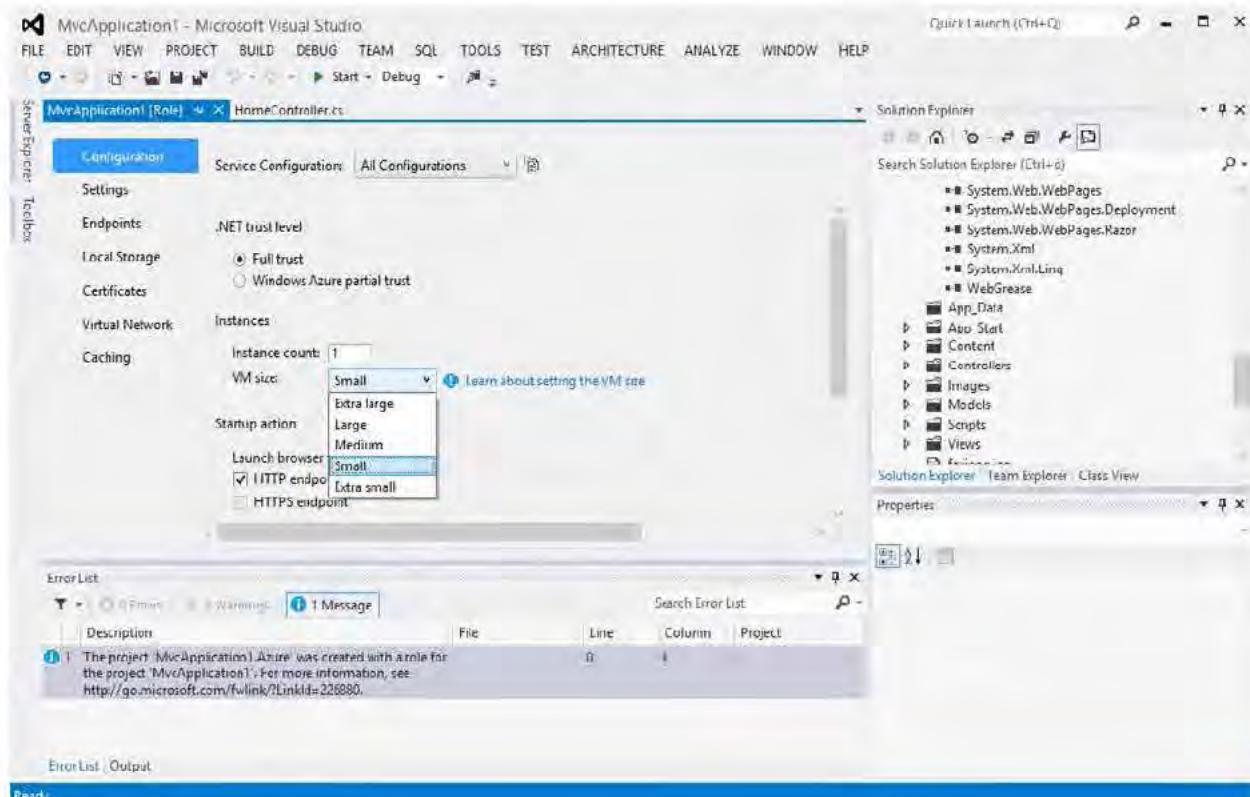
V druhom kroku budeme túto webovú aplikáciu migrovať do clodu. V kontextovom menu projektu aktivujte voľbu **Add Windows Azure Cloud Service Project**.



### *Pridanie projektu pre nasadenie webovej aplikácie do clodu*

Do vášho riešenia bude doplnený druhý projekt s názvom <nazov projektu>.Azure, v našom prípade MvcApplication1.Azure. V priečinku **Roles** nájdete webovú rolu, ktorej názov je zhodný s názvom webovej aplikácie.

Kliknutím na rolu sa na pracovnej obrazovke vývojového prostredia zobrazí konfiguračný dialóg pre nastavenie parametrov príslušnej role. Nastavenie je prehľadne rozdelené do niekoľkých záložiek. V záložke Configuration môžete definovať zdroje, ktoré budú použité pre beh aplikácie, čiže počet inštancií serverov, ich veľkosť a podobne.



### Konfigurácia webovej role

Toto nastavenie sa použije pre vytvorenie takzvaného distribučného balíčka, ktorý obsahuje informácie o konfigurácii pre konkrétnu aplikáciu. Tieto informácie sa odovzdajú do dátového centra a použijú sa pre vytváranie a konfiguráciu požadovaného počtu virtuálnych strojov.

### Spustenie aplikácie v emulátori platformy Azure

Ak by ste sa teraz pokúšali projekt zostaviť a spustiť, pravdepodobne by sa vám to nepodarilo. Aby ste mohli využívať emulátor cloubovej platformy Azure, musíte vývojové prostredie musíte spustiť v režime „Spustiť ako správca“. Ak sa vám zobrazilo chybové hlásenie, že sa nedá spustiť emulátor cloubovej platformy, projekt uložte, ukončite vývojové prostredie a znova ho spustite v režime „Správca“ (túto voľbu ponúka kontextové menu aplikácie Prieskumník Windows, pripadne kontextové menu ponuky aktivované tlačidlom Štart).

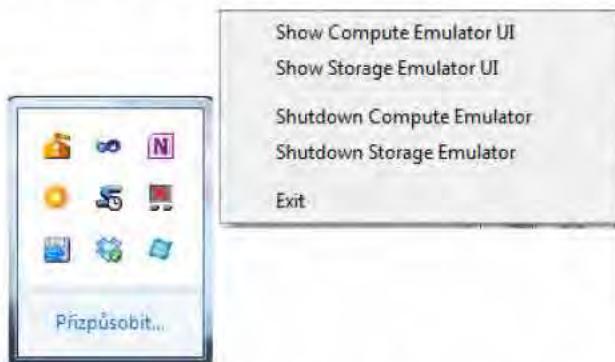
Po spustení aplikácie iniciuje Windows Azure emulator, ktorý emuluje prostredie cloubovej platformy a umožňuje odladiť aplikáciu ešte pred jej nasadením do reálneho cloubového prostredia.

*Emulátor vám môže ušetriť aj náklady na ladenie, nakoľko po uplynutí troch mesiacov je Windows Azure platenou službou, takže sem budete nasadzovať až fungujúcu aplikáciu.*

Proces spustenia trvá niekoľko desiatok sekúnd a v dialógu môžete sledovať, ako postupne nabiehajú jednotlivé zložky emulátora (storage emulator, compute emulator). Nakoniec sa do emulátora

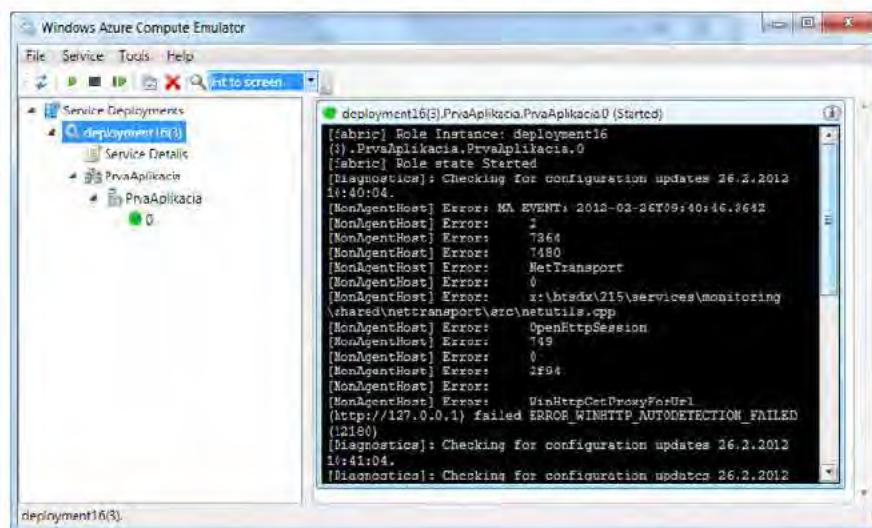
zavedie aplikácia a spustí sa v okne prehliadača webového obsahu. Po prvom spustení máte istotu, že všetko je správne nainštalované a nakonfigurované.

Ak vás zaujímajú podrobnosti ohľadne behu aplikácie v prostredí emulátora, v pravej časti spodnej lišty operačného systému Windows, teda v oblasti kde sa zobrazuje čas, prípadne jazyk klávesnice nájdete ikonku s azúrovým logom. V jej kontextovom menu sú položky pre zobrazenie používateľského rozhrania compute a storage emulátora.

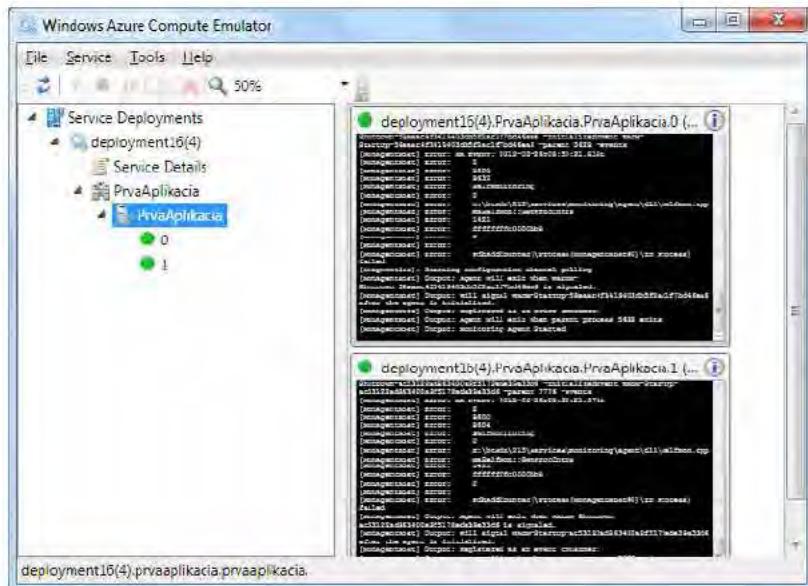


*Menu pre zobrazenie používateľského rozhrania compute a storage emulátora*

V priečinku Service Deployments je hierarchická štruktúra vyjadrujúca konfiguráciu nasadenia vašej aplikácie, teda počet inštancií. Po kliknutí na symbol inštancie sa v pravom okne (v konzole s čiernym pozadím) zobrazí výpis diagnostických údajov pre danú inštanciu.



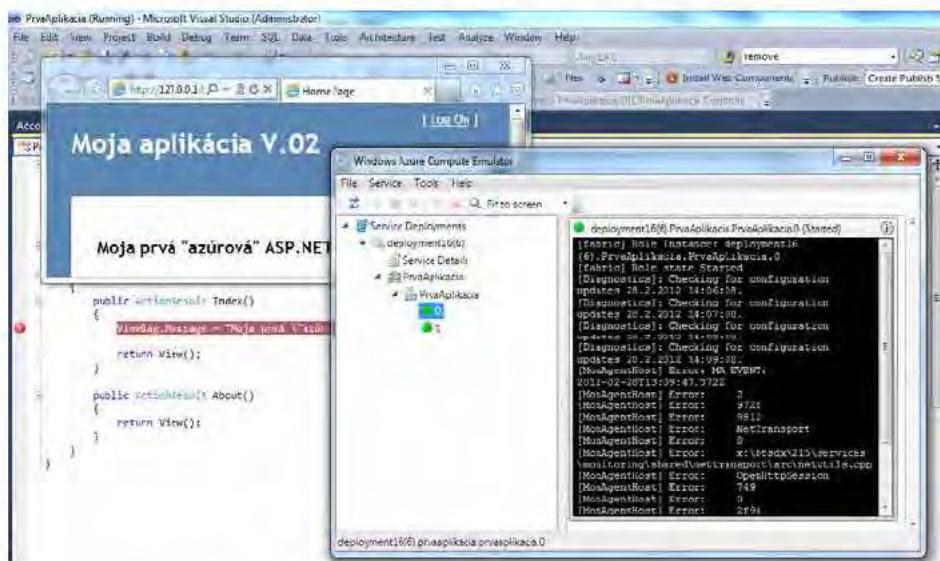
*Používateľské rozhranie Windows Azure Compute Emulator*



Ak v konfiguračnom dialógu pre nastavenie parametrov webovej role pomocou parametra *Instance Count* zadáte, že aplikácia bude bežať na viacerých inštanciach virtuálnych serverov (v našom prípade na dvoch), v rozhraní Windows Azure Compute Emulator sa zobrazia okná s priebehom deploymentu na obidva servery.

### Ladenie aplikácie v lokálном emulátori

Pri ladení postupujete rovnako ako pri klasickej ASP.NET aplikácii. Umiestnite v prostredí Visual Studio do kódu na požadované miesto zarázku. Pri vykonávaní kódu sa aplikácia zastaví na požadovanom mieste. Pri ladení aplikácie v emulátori platformy Azure môžete využiť všetky dostupné prostriedky nástroja Visual Studio na tento účel.



### Ladenie aplikácie

### Nasadenie projektu do reálneho clouдовého prostredia

Pre vývojára, ktorý sa nechce hlbšie zaujímať o to akým spôsobom bude jeho aplikácia nasadená, postačí pre popis toho čo ideme robiť veta z nadpisu tejto témy. Pre hlbavejšieho záujemcu uvedieme, že v procese nasadenia projektu do reálneho clouдовého prostredia sa z aplikácie

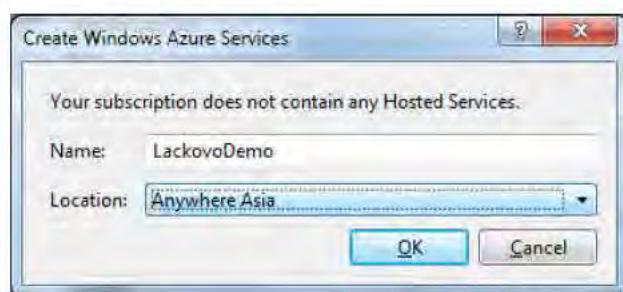
a pridruženého „clouдовého“ projektu vytvorí distribučný balíček pre nasadenie aplikácie do dátového centra. Balíček obsahuje jednak aplikáciu samotnú a taktiež aj predpis ako má byť aplikácia nasadená (na koľko inštancií virtuálnych strojov) a prevádzkovaná v dátovom centre. Distribučný balík vytvorte v nástroji Visual Studio.

V okne Solution Explorer v kontextovom menu projektu pre nasadenie aplikácie do clodu (projekt s názvom <nazov projektu>.Azure, v našom prípade PrvaApplikacia.Azure) aktivujte položku kontextového menu Publish. Zobrazí sa dialóg pre nasadenie aplikácie. Kliknite na odkaz Publish (v staršej verzii Visual Studio 2010 na odkaz Sign in to download credentials). Po prihlásení k svojmu Azure kontu si môžete do lokálneho vývojárskeho počítača stiahnuť XML súbor s oprávneniami.



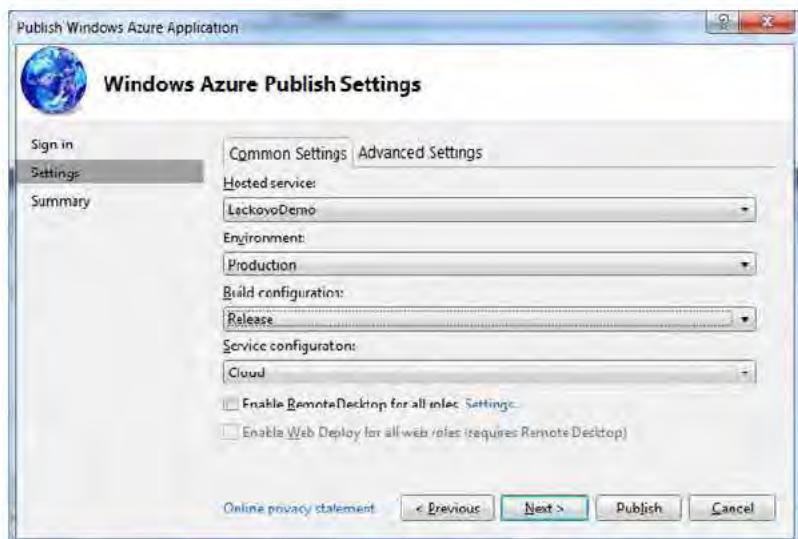
#### *Dialóg pre nasadenie aplikácie*

Pomocou tlačidla Import... tento súbor s certifikátom načítajte. V zozname Choose your subscription by sa mal zobraziť názov vášho Azure konta. Kliknutím na tlačidlo Next prejdete do fázy nastavenia deploymentu.



#### *Názov služby a výber lokality*

Dialóg má dve záložky. V záložke Common Settings sa v položke Hosted service nastavujú parametre hostovania služby (viď predchádzajúci dialóg). V položke Enviroment môžete nastaviť typ prostredia buď Production, alebo Staging. V položke Build configuration nastavujete konfiguráciu Release, prípadne Debug a v položke Service configuration máte na výber z dvoch možností: Cloud, prípadne Local.



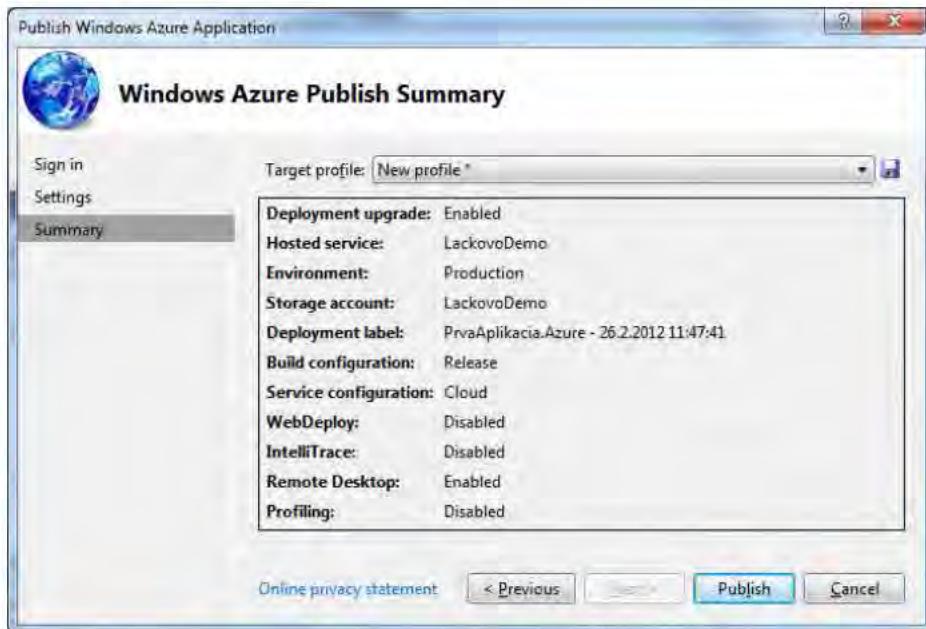
### *Nastavenie parametrov pre publikovanie aplikácie do cloutu*

V tejto fáze je potrebné aktivovať voľbu Enable Remote Desktop for all roles. Po označení začiarkavacieho boxu sa vám zobrazí dialóg pre nastavenie autorizácie diaľkového prístupu.



### *Nastavenie parametrov autorizácie diaľkového prístupu*

Následne sa pomocou tlačidla Next presuniete do záložky Summary, kde sú zobrazené sumárne údaje ohľadne nasadenia aplikácie do cloutu. Proces zavŕšite tlačidlom Publish.



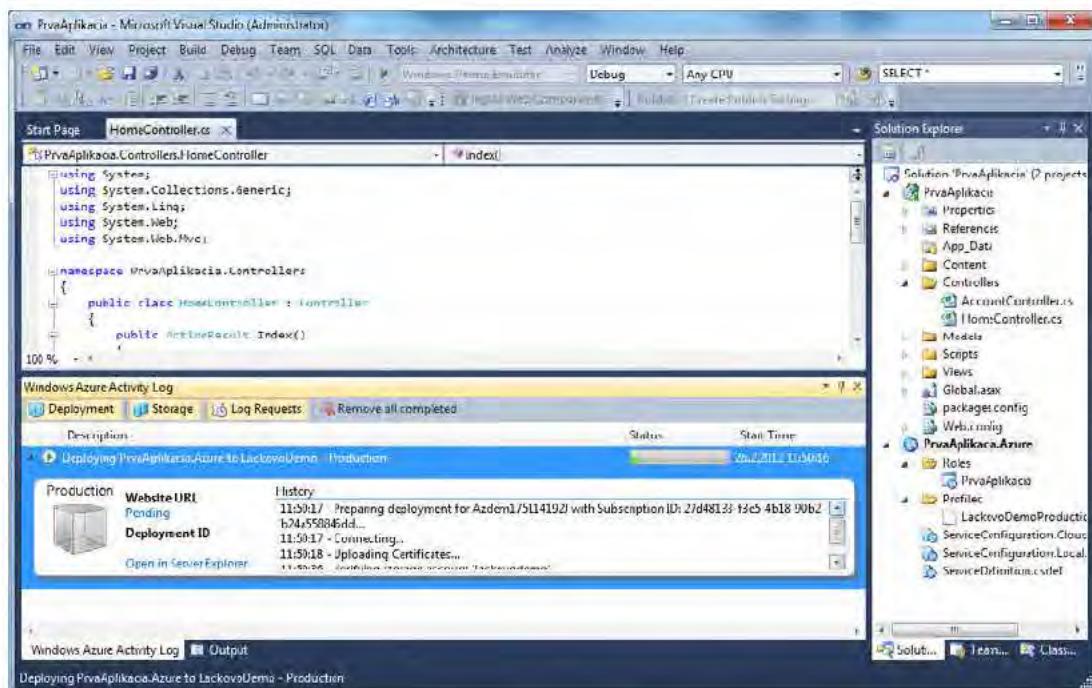
### Záložka Summary

V okne Windows Azure Activity Log situovanom v dolnej časti pracovnej obrazovky vývojového prostredia môžete sledovať priebeh nasadenia. Proces trvá niekoľko minút a v našom prípade bol jeho časový priebeh nasledovný. Všimnite si inicializáciu dvoch inštancií.

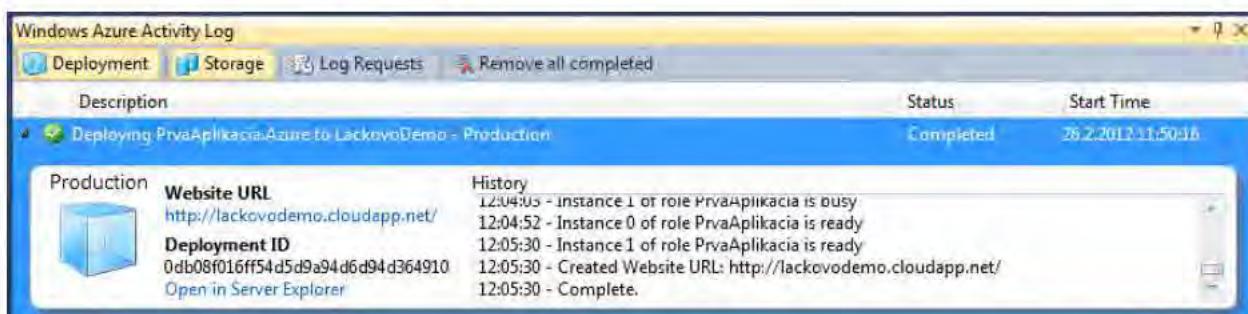
```

11:50:17 - Preparing deployment for Azdem175L14192J with Subscription ID: xxx...
11:50:17 - Connecting...
11:50:18 - Uploading Certificates...
11:50:36 - Verifying storage account 'lackovodemo',...
11:50:38 - Uploading Package...
11:57:38 - Creating...
11:59:52 - Created Deployment ID: 0db08f016ff54d5d9a94d6d94d364910.
11:59:52 - Starting...
12:00:50 - Initializing...
12:00:50 - Instance 0 of role PrvaApplikacia is creating the virtual machine
12:00:50 - Instance 1 of role PrvaApplikacia is creating the virtual machine
12:01:33 - Instance 0 of role PrvaApplikacia is starting the virtual machine
12:02:11 - Instance 1 of role PrvaApplikacia is starting the virtual machine
12:02:48 - Instance 0 of role PrvaApplikacia is in an unknown state
12:03:26 - Instance 0 of role PrvaApplikacia is busy
12:03:26 - Instance 1 of role PrvaApplikacia is in an unknown state
12:04:03 - Instance 1 of role PrvaApplikacia is busy
12:04:52 - Instance 0 of role PrvaApplikacia is ready
12:05:30 - Instance 1 of role PrvaApplikacia is ready
12:05:30 - Created Website URL: http://lackovodemo.cloudapp.net/
12:05:30 - Complete.

```



### Priebeh nasadenia aplikácie do clodu



Po úspešnom nasadení aplikácie sa zobrazí adresa URL pre prístup

Po úspešnom nasadení aplikácie sa zobrazí adresa URL pre prístup v tvare

<http://<NazovAplikacie>.cloudapp.net/> v našom prípade konkrétnie

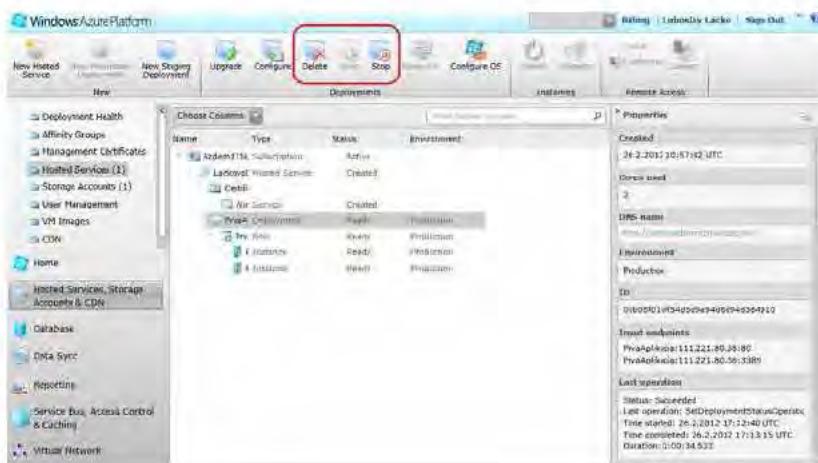
<http://lackovodemo.cloudapp.net/>.

### Odstránenie aplikácie z clodu

Ak ste aplikáciu nasadili do dátového centra Windows Azure, aplikácia je bez ohľadu na to, či je spustená alebo nie, spoplatňovaná v rámci vášho predplatného. Preto je dôležité, aby ste nepotrebné nasadenia aplikácie odstránili.

V priečinku Hosted Service management portálu vyberte položky, ktoré už nepotrebuje, tlačidlom Stop na nástrojovej lište zastavte služby a následne ich tlačidlom Delete odstráňte.

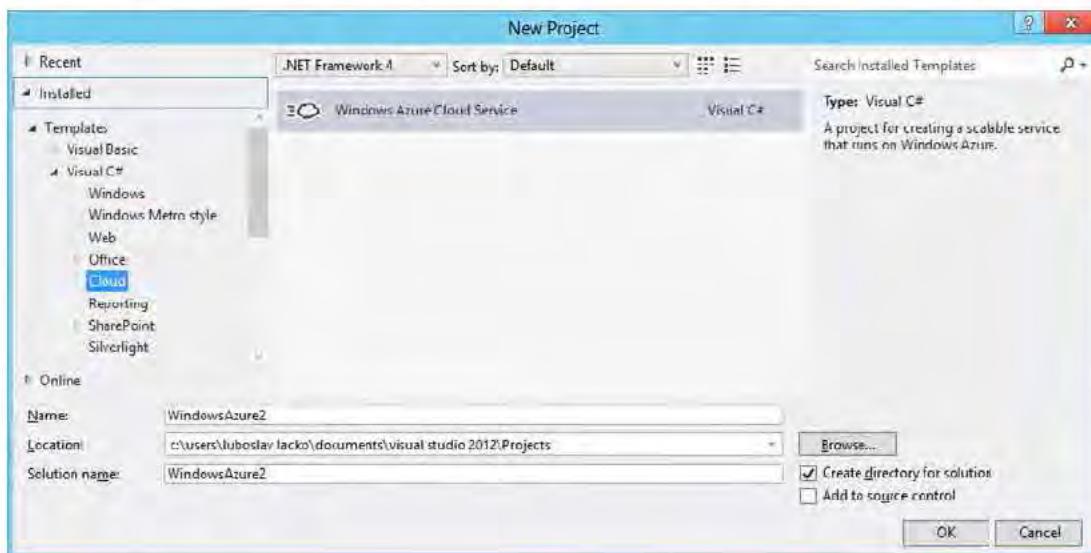
**Ak služby len zastavíte, virtuálne servery zostanú alokované a spoplatňované.**



Ovládacie prvky pre zastavenie a odstránenie aplikácie

## Vytvorenie clouдовého projektu

V predchádzajúcim príklade sme ukázali možnosť migrácie už hotovej ASP.NET webovej aplikácie do clodu. Ak vytvárate nový projekt „na zelenej lúke“, v nástroji Visual Studio 2012, prípadne v staršej verzii 2010, vytvorte nový projekt typu Windows AzureCloud Service. Táto šablóna vygeneruje dva súvisiace projekty. Vzorový projekt ASP.NET MVC webovej aplikácie a k nemu naviazaný projekt pre nasadenie aplikácie do clodu. Pripomíname, že vývojové prostredie musíte spustiť v režime „Spustiť ako správca“, inak nebudeť môcť pri lokálnom ladení spustiť emulátor platformy cloud. Môžete sa rozhodnúť pre programovací jazyk Visual Basic alebo Visual C#. Šablónu nájdete v priečinku Cloud.



Vytvorenie nového projektu

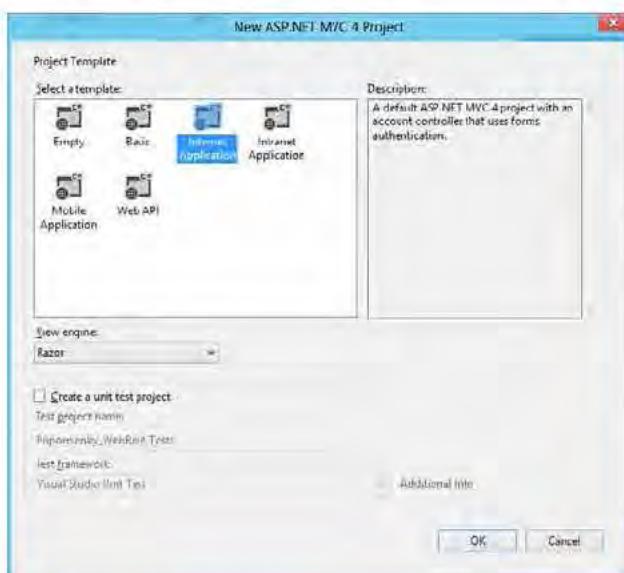
Sprievodca vytvorením „cloud“ projektu následne ponúkne dialóg New Cloud Service Project pre výber rolí, ktoré je potrebné pre projekt vytvoriť. Vývoj aplikácií pre Azure je principiálne podobný ako pri ASP.NET aplikáciách, novinkou, ktorú si nutne musíte osvojiť sú role. Inštancia role zapuzdruje kód, konfiguračné parametre a lokálne údaje. Rola je v dátovom centre zavedená na dedikovaný virtuálny stroj.

Pre tento príklad je potrebné vytvoriť webovú rolu, konkrétnie jej MVC variáciu ASP.NET MVC 4 Web Role. V dialógu New Windows Azure kliknutím na symbol „>“ presuňte túto rolu do okna rolí pre aktuálny projekt. Rolu vhodne premenujte, v našom prípade Priponienky\_WebRole.



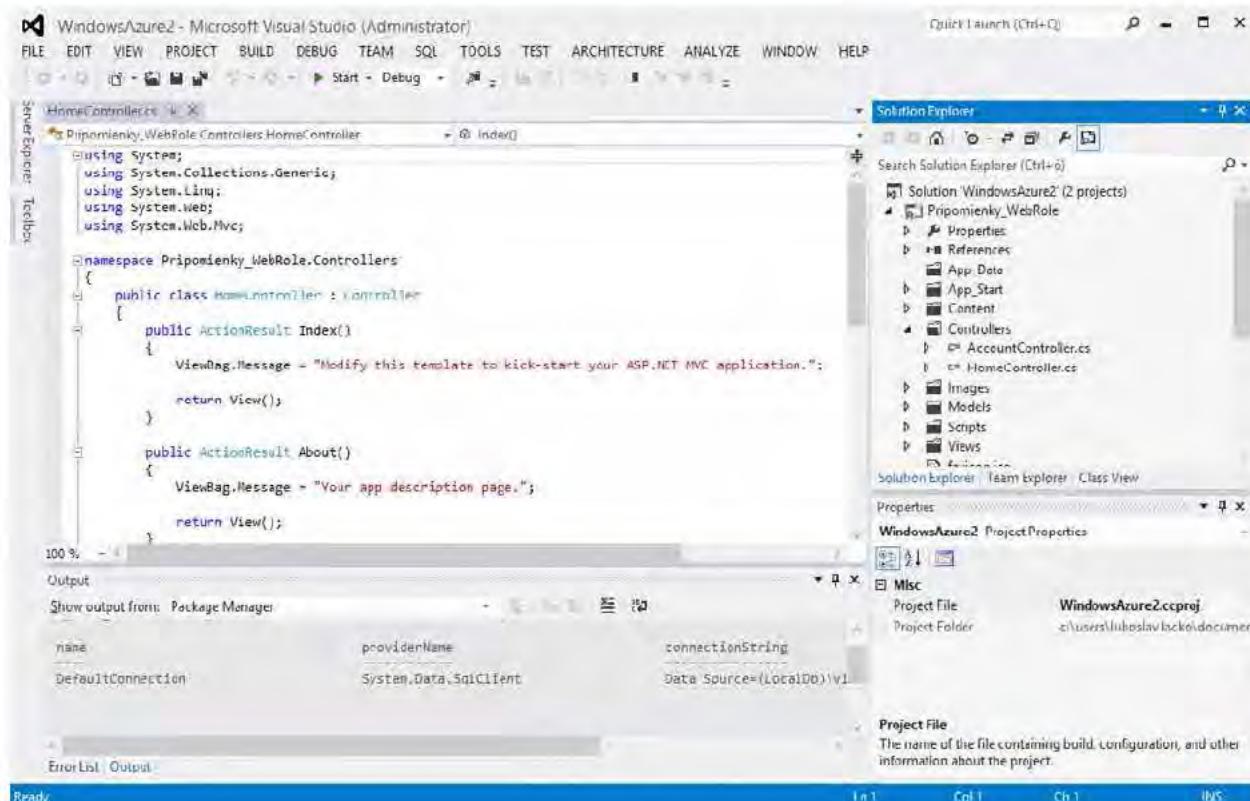
#### Vytvorenie ASP.NET WebRole

Následne špecifikujete typ šablóny pre ASP.NET MVC 4 projekt. Z ponúkaných šablón vyberte Internet Application. Všimnite si, že si môžete vybrať engine pre ASP.NET aplikáciu. Bud’ nový Razor, ktorý je implicitne preddefinovaný, alebo klasický ASPX engine.



#### Výber šablóny pre ASP.NET MVC 3 projekt

V okne Solution Explorer môžete preskúmať z akých súčasti riešenie pozostáva. Obsahuje dva projekty: Windows Azure projekt a MVC 4 projekt. Medzi nimi je definovaná súvislosť: **MVC 4 projekt vystupuje ako web rola Azure projektu**.



### Súčasti projektu v okne Solution Explorer

Informácie o roliach Windows Azure a informácie potrebné pre nasadenie aplikácie do clodu sú v definičnom súbore **ServiceDefinition.csdef** a v dvojici konfiguračných súborov

**ServiceConfiguration.Cloud.cscfg** a **ServiceConfiguration.Local.cscfg**. V súbore s príponou **csdef** sú metadáta potrebné pre hostovanie vašej aplikácie v dátovom centre. Konfiguračné súbory obsahujú údaje o počte požadovaných inštancií virtuálnych počítačov pre príslušnú rolu.

V úvodnej kapitole sme vám odporúčali nastaviť počas ladenia a betatestovania aplikácie v dátovom centre nastaviť „veľkosť“ virtuálneho počítača na najlacnejšiu verziu „Extra small“. Pre ostré nasadenie je potrebné nastaviť projektovanú konfiguráciu virtuálneho stroja v definičnom súbore **ServiceDefinition.csdef** nastavením parametra „**vmsize**“ v elemente „**WebRole**“.

```
<ServiceDefinition name="WindowsAzureProject" >
<WebRole name="MvcWebRole1" vmsize="Small">
<Sites>
<Site name="Web">
<Endpoints>
<InputEndpoint name="Endpoint1" protocol="http" port="80" />
</Endpoints>
<Imports>
<Import moduleName="Diagnostics" />
</Imports>
</WebRole>
</ServiceDefinition>
```

Pri inicializácii aplikácie sa hľadá trieda odvodená od triedy `RoleEntryPoint`. Pre webové role je táto trieda voliteľná, pre worker role je povinná. Môže obsahovať metódy:

- `OnStart()` – metóda je volaná po spustení, obsahuje potrebné inicializačné úkony. Ak chceme nazrieť trochu do zákulisia, táto metóda hlási load balanceru stav „BUSY“ až kým nie je vrátená hodnota `true`.
- `OnStop()` – metóda sa volá pri ukončovaní role a má na starosti „upratanie“.
- `Run()` – metóda obsahuje aplikačnú logiku Worker role. Tu je potrebné vytvoriť nekonečný cyklus. Len čo sa metóda ukončí, worker role bude recyklovaná.

V MVC aplikácii je trieda `WebRole` odvodená z triedy `RoleEntryPoint`, ktorá obsahuje metódu `OnStart`.

#### WebRole.cs

```
namespace MvcWebRole1
{
    public class WebRole : RoleEntryPoint
    {
        public override bool OnStart()
        {
            // For information on handling configuration changes
            // see the MSDN topic at http://go.microsoft.com/fwlink/?LinkId=166357.

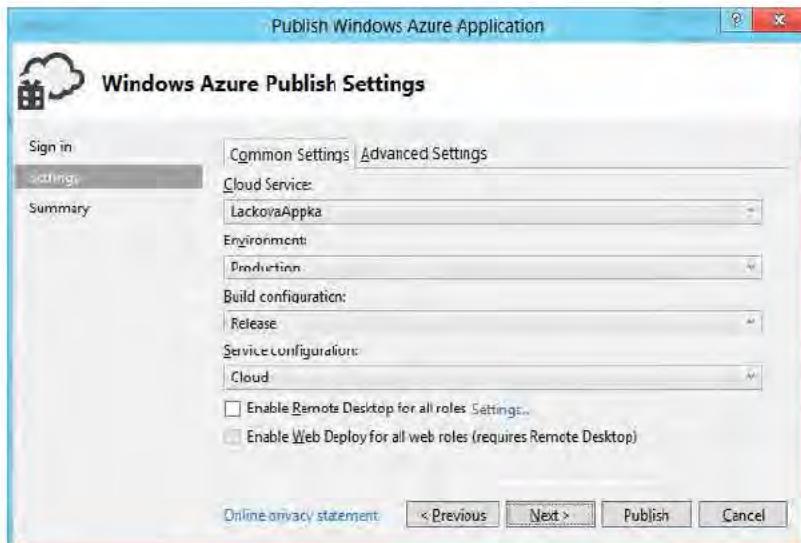
            return base.OnStart();
        }
    }
}
```

**Tip pre vývojárov:** Nepoužívajte konfiguračný súbor `web.config` pre hodnoty, ktoré sa budú meniť počas behu, nakoľko zmeny v tomto súbore vyžadujú nové nasadenie aplikácie. Použite súbor `ServiceConfiguration.cscfg`, kde po zmenách nie je nutná rekompilácia. Aplikácia je na zmeny upozornená prostredníctvom udalostí `RoleEnvironment.Changing` a `RoleEnvironment.Changed`.

```
public class WebRole : RoleEntryPoint
{
    public override bool OnStart()
    {
        RoleEnvironment.Changing += RoleEnvironmentChanging;
        return base.OnStart();
    }

    private void RoleEnvironmentChanging(object sender, RoleEnvironmentChangedEventArgs e)
    {
        if (e.Changes.Any(change => change is
RoleEnvironmentConfigurationSettingChange))
        {
            e.Cancel = true; // restart role
        }
    }
}
```

Pre publikovanie aktivujte v okne Solution Explorer položku kontextového menu **Publish To Windows Azure**.



### **Publikovanie aplikácie do clodu**

### **Úpravy a ladenie aplikácie**

Aj v prvej „Hello world“ aplikácii by mal vývojár aspoň trochu priložiť ruku k dielu a napísať nejaký text, aby získal pocit, že je to jeho vlastná aplikácia. Preto v súbore HomeController.cs nahraďte text „Welcome to ASP.NET MVC!“ svojím vlastným.

```
namespace MvcWebRole1.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewBag.Message = "Moja prvá \"azúrová\" ASP.NET MVC aplikácia!";
            return View();
        }

        public ActionResult About()
        {
            return View();
        }
    }
}
```

Pre začiatočníkov nezaškodí malé zoznámenie s projektom. Stránka, ktorú vidíte je skonštruovaná čiastočne v súbore \_Layout.cshtml, v priečinku Views – Shared, kde je „vizuál“ aplikácií.

```
<!DOCTYPE html>
<html>
<head>
<title>@ViewBag.Title</title>
<link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
<script src="@Url.Content("~/Scripts/jquery-1.5.1.min.js")"
type="text/javascript"></script>
</head>
<body>
<div class="page">
<div id="header">
```

```
<div id="title">
<h1>My MVC Application</h1>
</div>
<div id="logindisplay">
    @Html.Partial("_LogOnPartial")
</div>
<div id="menucontainer">
<ul id="menu">
<li>@Html.ActionLink("Home", "Index", "Home")</li>
<li>@Html.ActionLink("About", "About", "Home")</li>
</ul>
</div>
</div>
<div id="main">
    @RenderBody()
</div>
<div id="footer">
</div>
</div>
</body>
```

a čiastočne v súbore Index.cshtml, v priečinku Views – Shared, kde je časť obsahu úvodnej stránky.

```
@{
    ViewBag.Title = "Home Page";
}

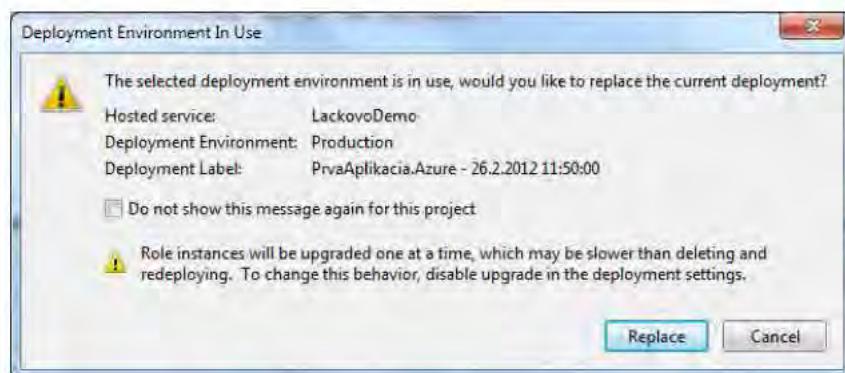
<h2>@ViewBag.Message</h2>
<p>
    To learn more about ASP.NET MVC visit <a href="http://asp.net/mvc" title="ASP.NET MVC Website">http://asp.net/mvc</a>.
</p>
```

Na tejto stránke je dobrá rada – viac podrobností o MVC aplikáciách sa dozviete na <http://asp.net/mvc>

Všimnite si, že šablóna MVC aplikácie obsahuje kompletnú správu prístupových účtov. Táto funkcia je implementovaná v priečinku Views – Account.

V súbore \_Layout.cshtml môžete zmeniť názov aplikácie a v súbore Index.cshtml pomocný text.

Po úpravách môžete aplikáciu znova vypublikovať. Ak ste „produkčnú“ aplikáciu už predtým niekedy spustili, zobrazí sa upozornenie, že aplikácia ktorá sa ide znova nasadiť sa používa a že súhlasíte s jej nahradením novou verzou.



*Upozornenie, že aplikácia ktorá sa ide znova nasadiť sa používa.*

Your logo here

Home Page. Moja prva "azúrová" ASP.NET MVC aplikácia.

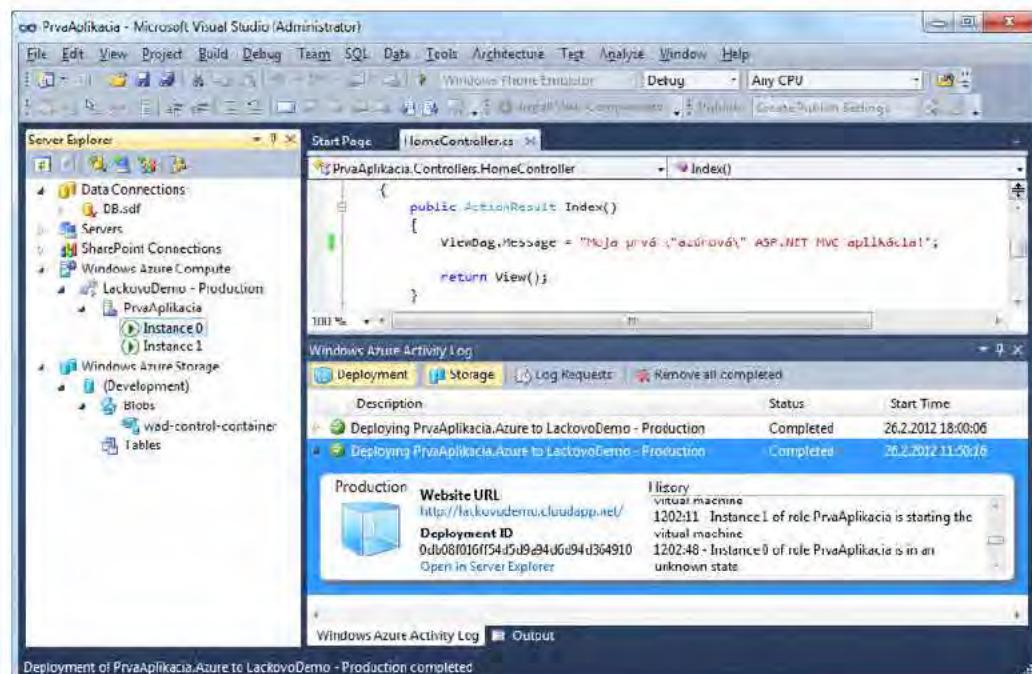
We suggest the following:

- 1 Getting Started  
ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and that gives you full control over markup for enjoyable, agile development. ASP.NET MVC includes many features that enable fast, TDD-friendly development for creating sophisticated applications that use the latest web standards. [Learn more](#)
- 2 Add NuGet packages and jump start your coding  
NuGet makes it easy to install and update free libraries and tools. [Learn more](#)
- 3 Find Web Hosting

### *Spustenie upravenej aplikácie priamo z clodu*

Všimnite si v grafickom zobrazení o priebehu deploymentu aplikácie odkaz **Open in Server Explorer**. Po aktivovaní odkazu si môžete pozrieť detaľy nasadenia aplikácie v okne Server Explorer v ľavej časti pracovnej obrazovky. K dispozícii sú dva priečinky:

- Windows Azure Compute,
- Windows Azure Storage.

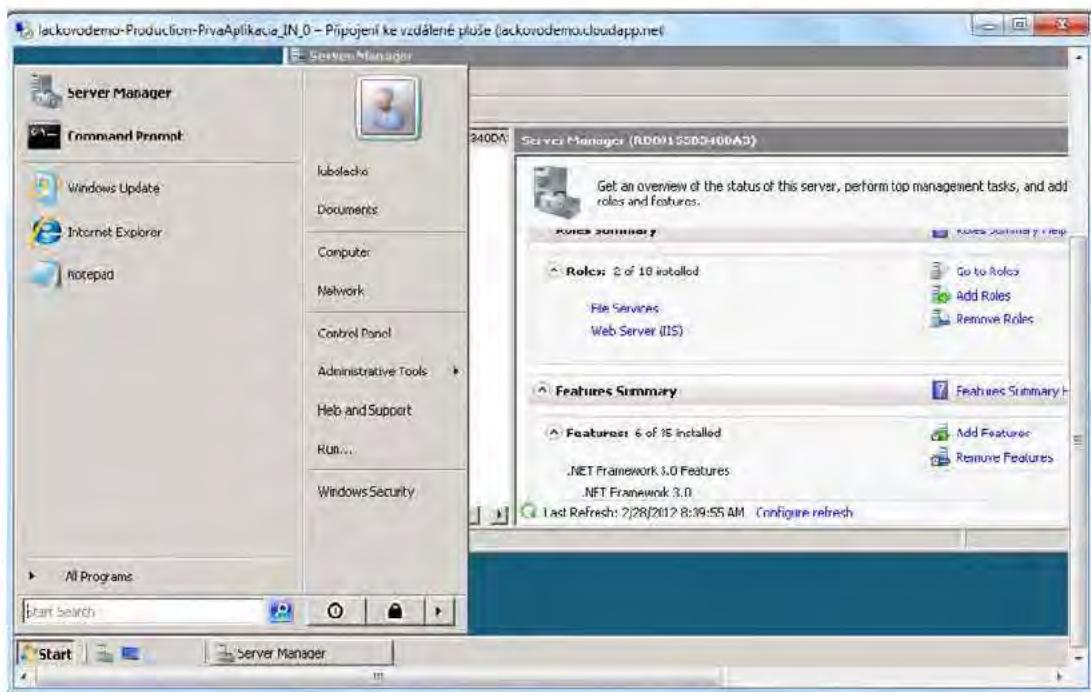


### *Spustenie upravenej aplikácie priamo z clodu*

Ak si chcete skontrolovať, ako to s využívaním inštancií a zdrojov vyzerá v „azúrovom“ cloude, otvorte v prehliadači stránku Management portálu <https://windows.azure.com>. Prehľad zdrojov nájdete v priečinku Hosted Services, Storage Account.

### *Spustenie upravenej aplikácie priamo z clodu*

Ak kliknete na symbol niektoréj z inštancií virtuálnych strojov, všimnite si, že sa na paneli nástrojov sprístupný ikona Connect. Pomocou tohto ovládacieho prvku sa pomocou prístupových privilégii, ktoré ste definovali v jednom z predchádzajúcich krokov, môžete pripojiť cez RDP protokol k vybranej inštancii a nechať si zobraziť jej pracovnú plochu.



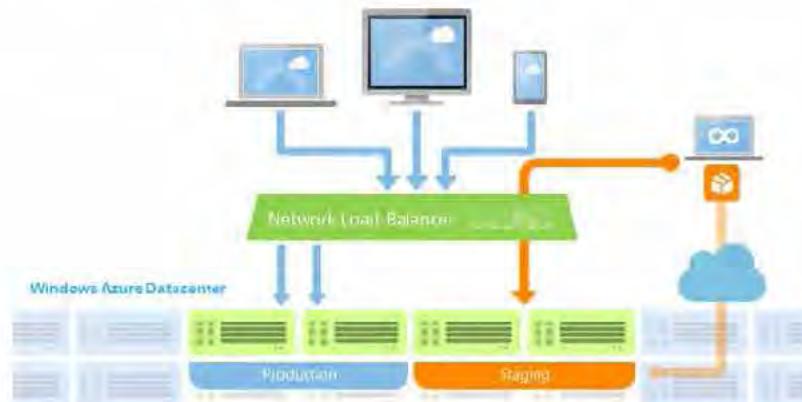
### *Vzdialé pripojenie k virtuálnej inštancii cez RDP protokol*

#### **Staging versus Production**

Platforma Windows Azure umožňuje updatovanie aplikácie bez prerušenia služieb, z toho vyplýva aj postup jej zavedenia. Najskôr je potrebné aplikáciu zaviesť do testovacieho úložiska (Staging), a tam

ju otestovať. Až keď budete s aplikáciou spokojní, môžete ju presunúť do produkčnej oblasti, kde je k dispozícii pre komerčné použitie.

Preto počas testovania novej verzie nezavádzame túto do „produkčného“ prostredia, ale do stagingu kde sa dôkladne otestuje.



#### *Nasadenie aplikácie do stagingu*

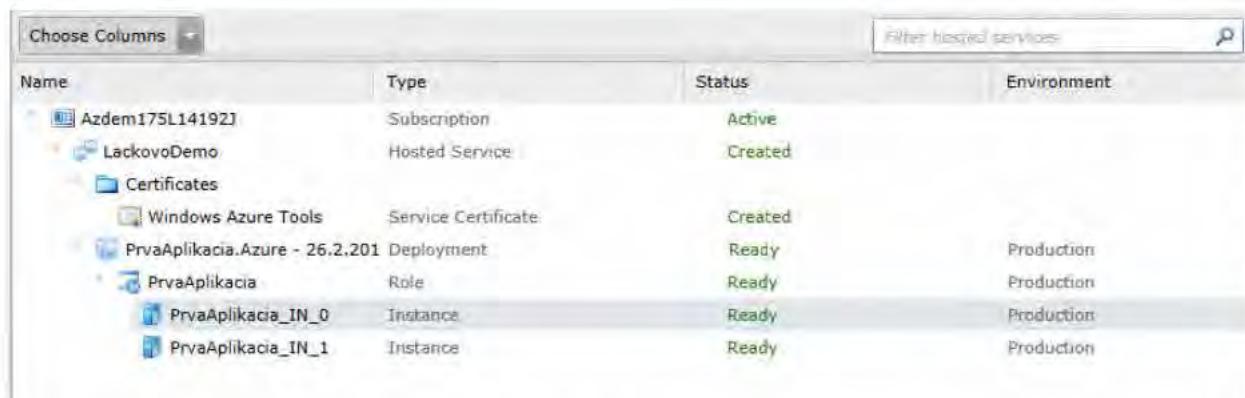
Spôsob nasadenia sa nastavuje v záložke Settings dialógu pre publikovanie Azure aplikácie, pomocou parametra Environment.



#### *Prepnutie parametra Environment*

Adresu URL aplikácie potom sprístupníte všetkým, čo budú zainteresovaní do testovania novej verzie aplikácie.

Typ nasadenia sa dá zistiť cez Management portál <https://windows.azure.com> v stĺpci Environment.



Name	Type	Status	Environment
Azdem175L14192J	Subscription	Active	
LackovoDemo	Hosted Service	Created	
Certificates			
Windows Azure Tools	Service Certificate	Created	
PrvaApplikacia.Azure - 26.2.201	Deployment	Ready	Production
PrvaApplikacia	Role	Ready	Production
PrvaApplikacia_IN_0	Instance	Ready	Production
PrvaApplikacia_IN_1	Instance	Ready	Production

*Typ nasadenia sa dá zobraziť cez Management portál*

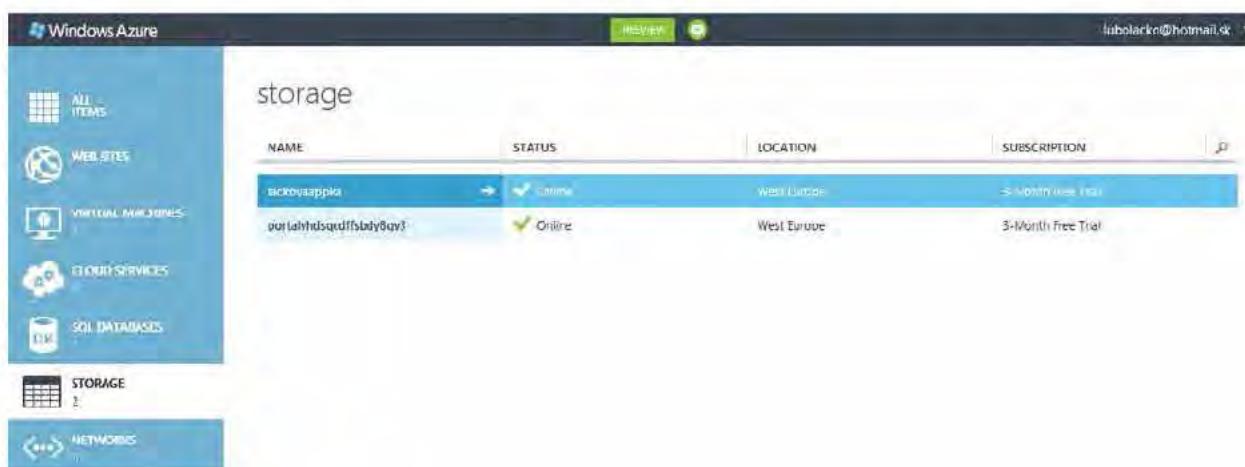
## Ukladanie údajov

Ako resumé za prvým príkladom môžeme skonštatovať jednoduchú migráciu webovej aplikácie, či už existujúcej, alebo nového projektu do „azúrového“ clodu. Skúsme si zrekapitulovať, čo zatiaľ dokážeme. Dá sa to vyjadriť jednou vetou – **dokážeme vypublikovať webovú aplikáciu do clodu a to bez akejkoľvek konfigurácie**. Stačí si objednať službu v požadovanom objeme. Optimista by povedal, že dokážeme veľa, pesimista by oponoval, že dokážeme zatiaľ pomerne málo. Objektivita je v tomto prípade skôr na strane pesimistu. Ak zrekapitulujeme veci ktoré zatiaľ nedokážeme, na prvom mieste určite bude to, že nedokážeme ukladať údaje. Takže sice môžete prevádzkovať modernú dynamickú webovú aplikáciu, no táto dokáže komunikovať zatiaľ len jedným smerom – od aplikácie k používateľovi. Používateľ nemôže robiť niektoré úkony, ktoré sa v ére Webu 2.0 zdajú úplne samozrejmé – nedokáže nechať komentár k obsahu na webovej stránke, nedokáže si nič objednať nedokáže... summa summarum aplikácia bez možnosti ukladania údajov má veľmi, veľmi obmedzené možnosti.

### Vytvorenie a konfigurácia úložiska

Konfiguráciu úložísk môžete zistiť cez Management portál

<https://manage.windowsazure.com> https://windowsazure.com v priečinku Storage.



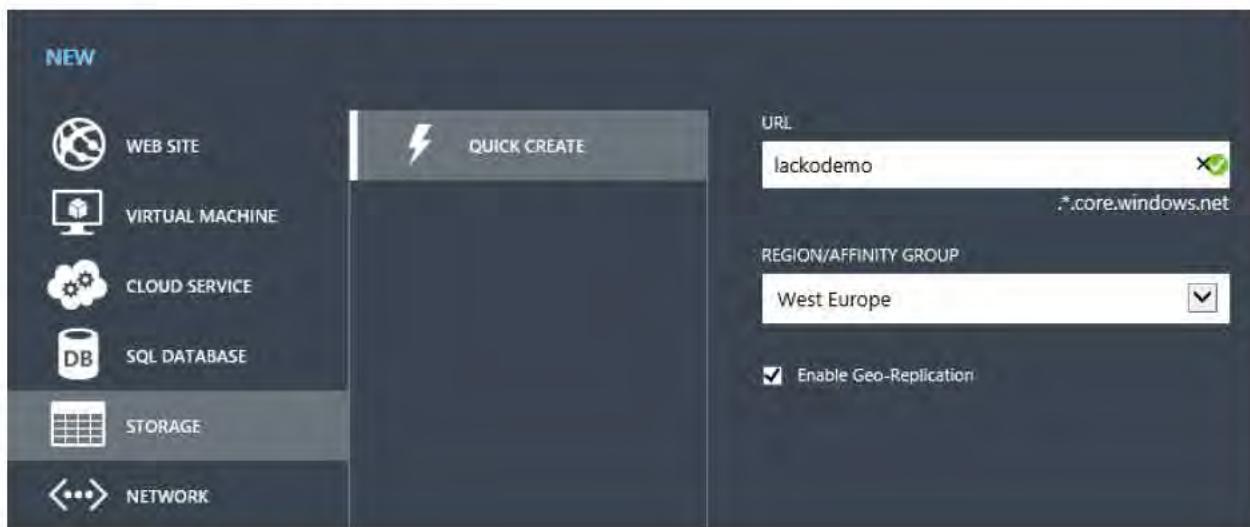
NAME	STATUS	LOCATION	SUBSCRIPTION
lackovoapp01	Online	West Europe	3-Month Free Trial
lackovoapp02	Online	West Europe	3-Month Free Trial

*Konfigurácia úložísk*

Ak chcete vytvoriť nové úložisko, aktivujte na aplikačnej lište New pred ktorým je ikona „+“. Pri vytváraní je potrebné zadať adresu URL v tvarе

<http://<StorageAccountLabel>.blob.core.windows.net>,

napríklad `lackovodemo.blob.core.windows.net`. Pri vytváraní úložiska si môžete vybrať lokalitu (kontinent) kde budú údaje uložené.

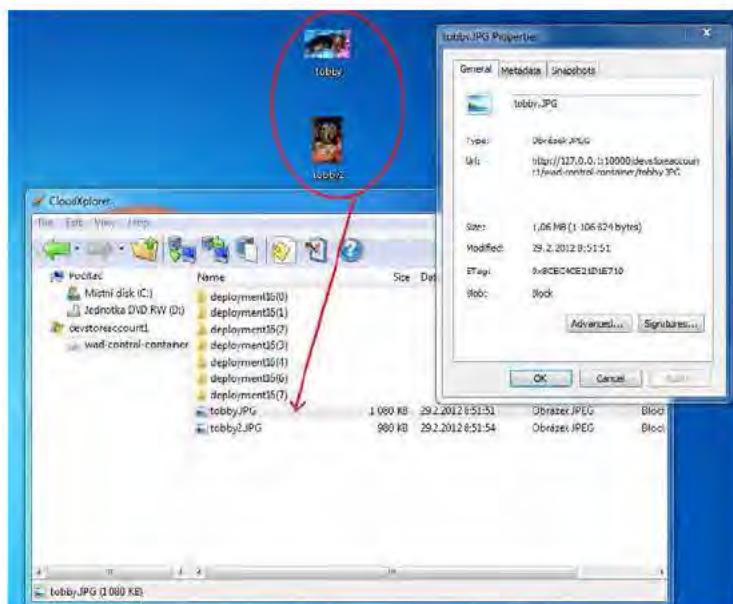


### Vytvorenie nového úložiska

### Možnosti nastavenia monitorovania úložiska

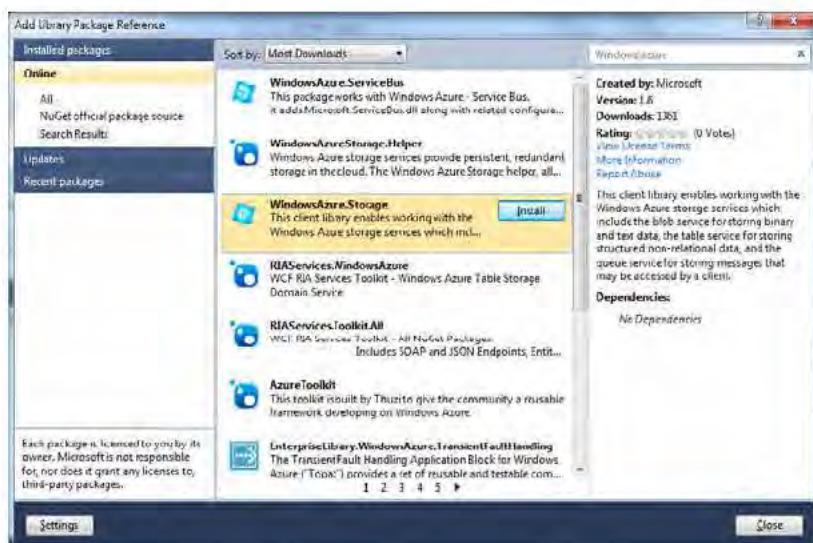
Pre manipuláciu s položkami (súbormi) v úložisku môžete použiť produkty tretích strán, napríklad CloudXplorer, či AzureXplorer for Visual Studio (<http://clumsyleaf.com>).

Môžete si vyskúšať kopírovanie súborov, napríklad obrázkov do Blob úložiska pomocou aplikácie CloudXplorer. V kontextovom menu, položka Properties získate adresu URL pre prístup ku konkrétnemu súboru, v tomto prípade obrázku.



### Kopírovanie súborov do Blob úložiska pomocou aplikácie CloudXplorer

Pre pohodlný vývoj aplikácií využívajúcich Blob úložisko si môžete nainštalovať balíček WindowsAzure.Storage z NuGet Gallery (<http://nuget.org>). Inštaláciu balíčka je možné urobiť priamo z prostredia Visual Studio 2010 z kontextového menu Azure projektu, položka Add Library package reference.



### Inštalácia balíčka WindowsAzure.Storage

V aplikácii využívajúcej clouдовé úložisko je potrebné najskôr nastaviť pripojenie k nemu.

```
public static CloudStorageAccount DataStorageAccount {get; private set; }

public static class ConfigurationHelper
{
    //...
    DataStorageAccount =
        cloudStorageAccount.FromConfigurationSetting("MojStorageConnectionString");
}
```

Pre vytvorenie referencie na konkrétny kontajner využijete fragment kódu

```
public static CloudBlobContainer GetContainer(string containerName)
{
    return ConfigurationHelper.DataStorageAccount.
        CreateCloudBlobClient().GetContainerReference(containerName);
}
```

Vytvorenie kontajnera

```
public static void VytvorContainer(string containerName)
{
    CloudBlobContainer container = GetContainer(containerName);
    container.CreateIfNotExist();
    BlobContainerPermissions permissions = container.GetPermissions();
    permissions.PublicAccess = BlobContainerPublicAccessType.Container;
    container.SetPermissions(permissions);
}
```

Do mozaiky kódových fragmentov nám chýba ešte ukladanie a čítanie údajov

```
public static CloudBlob PutBlob(string blobName, string containerName, string contentType,
byte[] data, NameValueCollection metadata)
{
    VytvorContainer (containerName);
    CloudBlob blob = GetContainer(containerName).GetBlobReference(blobName);
    blob.DeleteIfExists();
    blob.Properties.ContentType = contentType;
    blob.Metadata.Add(metadata);
    blob.UploadByteArray(data);
    return blob;
}

public static CloudBlob GetBlob(string blobAddress)
{
    return ConfigurationHelper.DataStorageAccount.CreateCloudBlobClient().
        GetBlobReference(blobAddress);
}
```

## Cvičný projekt pre ukladanie údajov do Table úložiska

Námetom projektu bude v konečnom dôsledku jednoduché diskusné fórum, umožňujúce písanie príponiek a komentárov k danej téme.

Vytvorte aplikáciu typu Windows Azure Cloud Service a do riešenia zahrňte rolu typu ASP.NET Web Role.



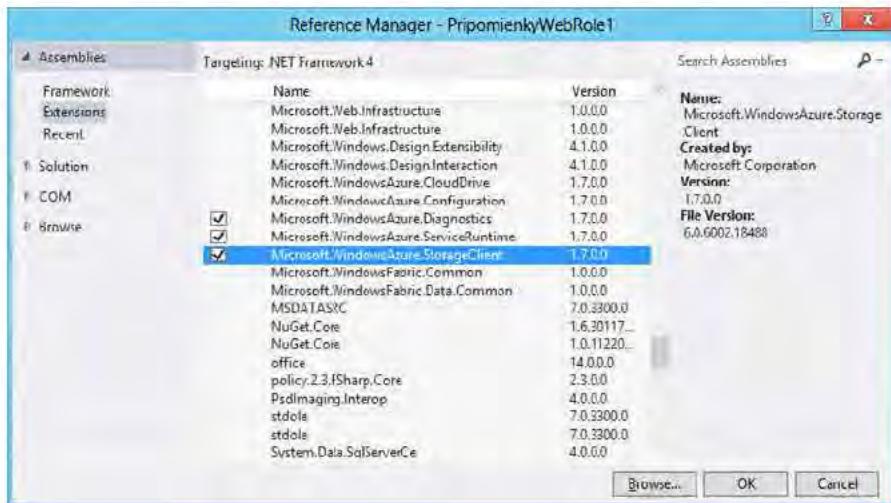
### Pridanie Web role do projektu

Šablóna úvodnej stránky aplikácie je na adrese Default.aspx.

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="Priponienky_WebRole._Default" %>

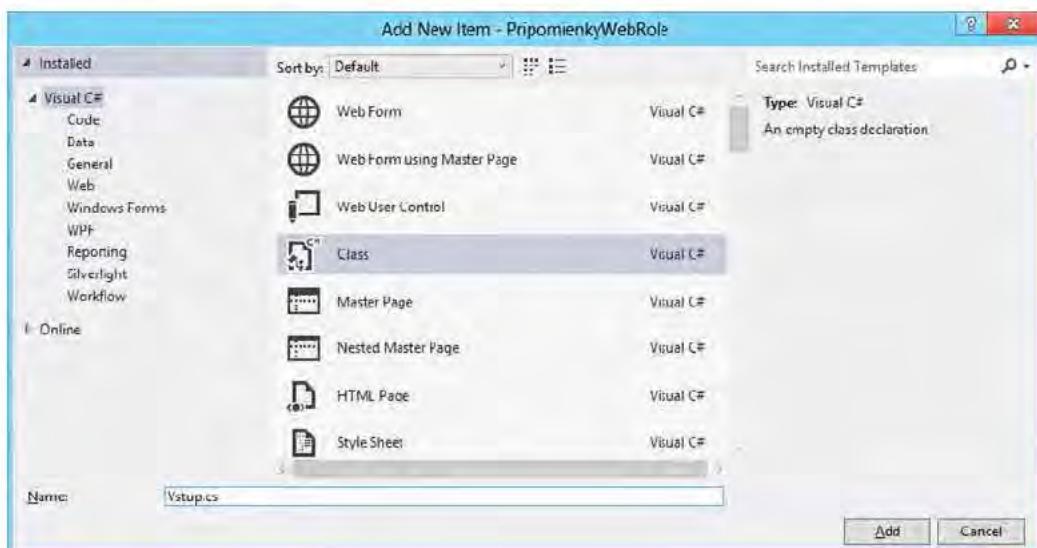
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
<h2>
    Welcome to ASP.NET!
</h2>
<p>
    To learn more about ASP.NET visit <a href="http://www.asp.net" title="ASP.NET Website">www.asp.net</a>.
</p>
<p>
    You can also find <a href="http://go.microsoft.com/fwlink/?LinkId=152368&clcid=0x409"
        title="MSDN ASP.NET Docs">documentation on MSDN</a>.
</p>
</asp:Content>
```

Ak projekt vytvárate v prostredí Visual Studio 2010 je potrebné zahrnúť do projektu aj referencie **Microsoft.WindowsAzure.StorageClient** a **System.Data.Service.Client**.



### **Knižnice tried pre ukladanie údajov**

Pomocou položky Add New Item kontextového menu v okne Solution Explorer aplikovaného na rolu Pripomienky\_WebRole vytvorte novú triedu napríklad s názvom Vstup.



### **Dialóg pre pridanie triedy**

Do kódu triedy v súbore Vstup.cs je potrebné pridať referenciu na namespace.

```
using Microsoft.WindowsAzure.StorageClient;
```

Pripomienky sa budú ukladať v dvoch členských premenných typu string. V prvej je meno prispievateľa a v druhej text pripomienky.

Pomocou parametra PartitionKey sa v konštruktori triedy vytvorí separátna partícia údajov pre každý deň. Každý záznam bude obsahovať unikátny identifikátor, odvodený od času. Pre tento účel je v konštruktori parameter RowKey.

```
using Microsoft.WindowsAzure.StorageClient;

namespace Namety_Data
{
    public class Vstup : Microsoft.WindowsAzure.StorageClient.TableServiceEntity
    {
        public string GuestName { get; set; }
        public string Message { get; set; }

        public Vstup()
        {
            PartitionKey = DateTime.UtcNow.ToString("MMddyyyy");
            RowKey = string.Format("{0:10}_{1}", DateTime.MaxValue.Ticks -
DateTime.Now.Ticks, Guid.NewGuid());
        }
    }
}
```

**Podobným spôsobom vytvorte triedu s názvom napríklad DataKontext, ktorá bude odvodená od triedy TableServiceContext.**

```
using Microsoft.WindowsAzure;
using Microsoft.WindowsAzure.StorageClient;

namespace Namety_Data
{
    public class DataKontext : TableServiceContext
    {
        public IQueryable<Vstup> Vstup
        {
            get { return this.CreateQuery<Vstup>("Vstup"); }
        }

        public DataKontext(string baseAddress, StorageCredentials credentials)
            : base(baseAddress, credentials)
        { }
    }
}
```

**Následne vytvorte triedu ktorá bude zapuzdrovať dátový zdroj napríklad s názvom NametyDataSource.**

```
using Microsoft.WindowsAzure;
using Microsoft.WindowsAzure.StorageClient;

namespace Namety_Data
{
    public class NametyDataSource
    {
        private static CloudStorageAccount storageAccount;
        private DataKontext context;

        static NametyDataSource()
        {
            storageAccount =
CloudStorageAccount.FromConfigurationSetting("DataConnectionString");

            CloudTableClient.CreateTablesFromModel(
                typeof(DataKontext),
                storageAccount.TableEndpoint.AbsoluteUri,

```

```

        storageAccount.Credentials);
    }

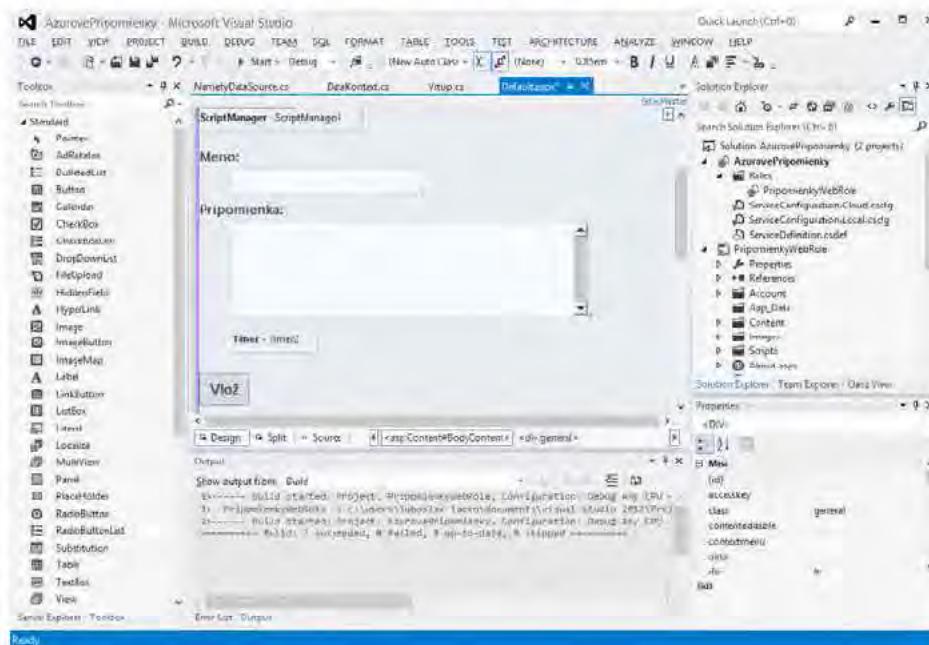
    public NametyDataSource()
    {
        this.context = new DataKontext(storageAccount.TableEndpoint.AbsoluteUri,
storageAccount.Credentials);
        this.context.RetryPolicy = RetryPolicies.Retry(3, TimeSpan.FromSeconds(1));
    }

    public IEnumerable<Vstup> Select()
    {
        var results = from g in this.context.Vstup
                      where g.PartitionKey == DateTime.UtcNow.ToString("MMddyyyy")
                      select g;
        return results;
    }

    public void NovaPripomienka(Vstup newItem)
    {
        this.context.AddObject("Vstup", newItem);
        this.context.SaveChanges();
    }
}

```

Tieto tri triedy tvoria akýsi jednoduchý dátový model pre ukladanie údajov a prístup k nim. Vo finálnej fáze vytvoríme používateľské rozhranie na stránke Default.aspx.



### Návrh používateľského rozhrania

Pre obnovu údajov pridajte do projektu časovač. Kompletný kód stránky Default.aspx:

```

<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true"
CodeBehind="Default.aspx.cs" Inherits="PripomienkyWebRole._Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

```

```

</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">


68


```

```

        "<%# Eval("Message") %>
</div>
<div class="signatureSays">
</div>
</div>
</ItemTemplate>
</asp:DataList>
<asp:Timer
    ID="Timer1"
    runat="server"
    Interval="15000"
    OnTick="Timer1_Tick">
</asp:Timer>
</ContentTemplate>
</asp:UpdatePanel>
<asp:ObjectDataSource
    ID="ObjectDataSource1"
    runat="server"
    DataObjectType="PriponienkyWebRole.Vstup"
    InsertMethod="NovaPriponienka"
    SelectMethod="Select"
    TypeName="PriponienkyWebRole.NametyDataSource">
</asp:ObjectDataSource>
</div>
</asp:Content>
```

Jadro kódu aplikácie, ktoré využíva dosiaľ definované triedy je v súbore Default.aspx.cs.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using PriponienkyWebRole;
using System.Net;

using Microsoft.WindowsAzure;
using Microsoft.WindowsAzure.ServiceRuntime;
using Microsoft.WindowsAzure.StorageClient;

namespace PriponienkyWebRole
{
    public partial class _Default : System.Web.UI.Page
    {
        private static bool storageInitialized = false;
        private static object gate = new Object();

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {
                Timer2.Enabled = true;
            }
        }

        protected void Timer1_Tick(object sender, EventArgs e)
        {
            DataList1.DataBind();
```

```
}

private void InitializeStorage()
{
    if (storageInitialized) { return; }

    lock (gate)
    {
        if (storageInitialized) { return; }

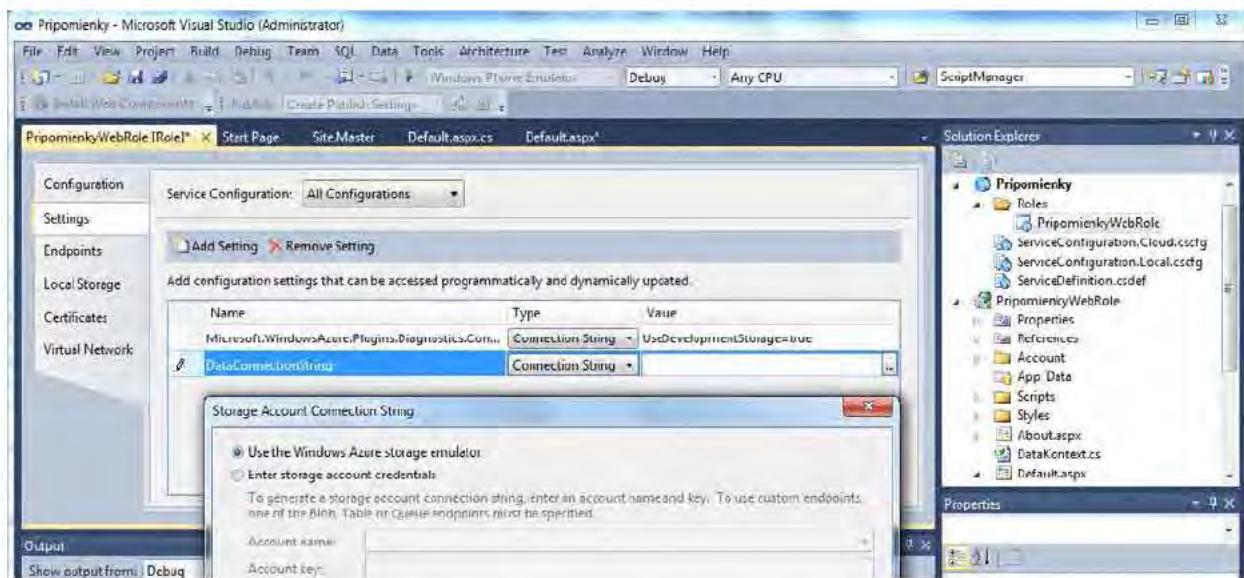
        try
        {
            var storageAccount =
CloudStorageAccount.FromConfigurationSetting("DataConnectionString");
        }
        catch (WebException)
        { throw new WebException("Zlyhala inicializácia služby Storage."); }

        storageInitialized = true;
    }
}

protected void SignButton_Click(object sender, EventArgs e)
{
    InitializeStorage();
    Vstup entry = new Vstup() { GuestName = NameTextBox.Text, Message =
MessageTextBox.Text };
    NametyDataSource ds = new NametyDataSource();
    ds.NovaPripomienka(entry);
    System.Diagnostics.Trace.Information("Added entry {0}-{1} in table storage
for guest '{2}'", entry.PartitionKey, entry.RowKey, entry.GuestName);
    NameTextBox.Text = "";
    MessageTextBox.Text = "";
    DataList1.DataBind();
}
}
```

Pred spustením aplikácie, či už v emulátore, alebo v reálnom prostredí platformy Windows Azure je potrebné nakonfigurovať prístup k dátovému úložisku.

#### **DataConnectionString Use development storage**



### Konfigurácia dátového úložiska

V reálnej aplikácii v prostredí clodu je potrebné vytvoriť úložisko (vygenerujú sa endpointy pre príslušné typy údajových štruktúr, primárny a sekundárny klúč) a zmeniť reťazec pre pripojenie k údajom v dátovom centre. Teraz môžete aplikáciu otestovať.

Zavedenie aplikácie do testovacej oblasti môže trvať dosť dlho, v prípade zložitejšej aplikácie, ktorá využíva aj dátové úložiská to môže byť aj niekoľko minút.

### Vytvorenie aplikácie využívajúcej SQL Database

Jednoduchú aplikáciu tohto typu sme už v tejto publikácii riešili v aplikácii Visual Studio 2012. Aby sme tému využitia SQL Database rozvinuli, pripravili sme ešte jeden príklad, tentoraz vo vývojovom prostredí Visual Studio 2010. Námetom aplikácie bude jednoduchý priponienkovač, čiže aplikácia typu „To Do List“. Aplikácia bude využívať databázovú službu Windows Azure SQL Database. Aby ste mohli využívať emulátor platformy Azure na lokálnom PC, spustite Visual Studio 2010 v režime „Spustiť ako správca“. Vytvorte nový projekt typu Windows AzureCloud Service. Do zoznamu roli pridajte rolu ASP.NET MVC3 Web role a vhodne ju premenujte, napríklad na ToDoWebRole.

V nasledujúcim dialógu ponechajte vybrané voľby Internet Application template a Razor view engine.

Najskôr prispôsobíme šablónu používateľského rozhrania potrebám aplikácie. V priečinku Views\Shared otvorte súbor \_Layout.cshtml. Zmeníte názov aplikácie, prípadne môžete lokalizovať položky používateľského rozhrania „Home“, „Index“, „About“

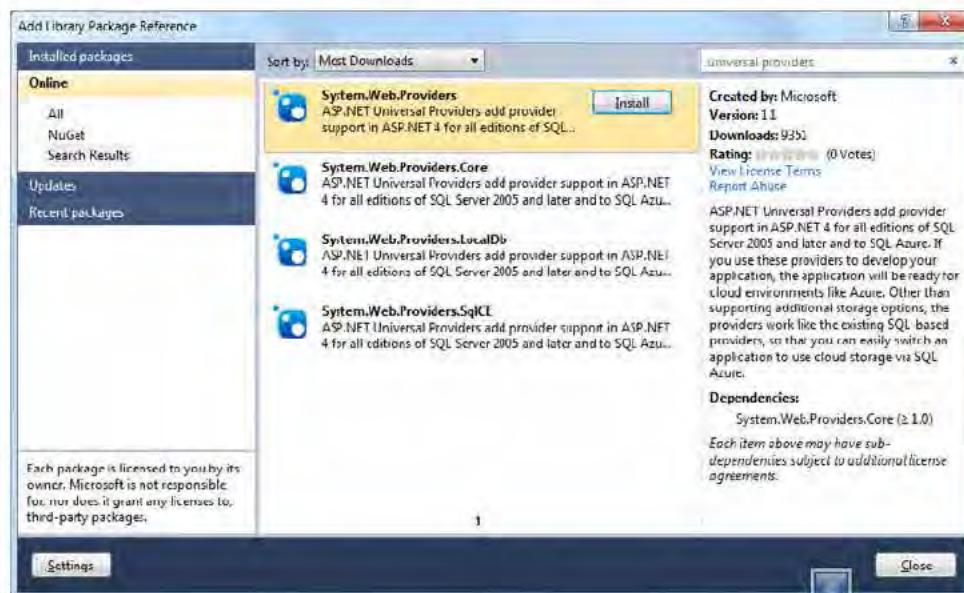
```
<!DOCTYPE html>
<html>
<head>
<title>@ViewBag.Title</title>
<link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
<script src="@Url.Content("~/Scripts/jquery-1.5.1.min.js")"
type="text/javascript"></script>
</head>
<body>
<div class="page">
<div id="header">
<div id="title">
```

```

<h1>Zoznam úloh</h1>
</div>
<div id="logindisplay">
    @Html.Partial("_LogOnPartial")
</div>
<div id="menucontainer">
<ul id="menu">
<li>@Html.ActionLink("Home", "Index", "Home")</li>
<li>@Html.ActionLink("About", "About", "Home")</li>
</ul>
</div>
</div>
<div id="main">
    @RenderBody()
</div>
<div id="footer">
</div>
</div>
</body>
</html>

```

Aplikujte na rolu v okne Solution Explorer položku kontextového menu Add Library Package Reference. V rovnomennom dialógu v priečinku ONLINE nechajte vyhľadávať „universal providers“:



#### Inštalácia providera pre SQL Server a Windows Azure SQL Database

Vyberte položku **System.WebProviders** a tlačidlom **Install** nainštalujte provider **ASP.NET Universal Providers** pre SQL Server a Windows Azure SQL Database.

Otvorte súbor **Web.config** a upravte sekciu **<configuration> / <connectionStrings>**. Implicitný obsah:

```

<add name="DefaultConnection" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=aspnetdb;Integrated Security=True" providerName="System.Data.SqlClient" />
upravte takto

```

```

<add name="DefaultConnection" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=aspnet_ToDoWebRole;Integrated Security=True;MultipleActiveResultSets=True"
    providerName="System.Data.SqlClient" />

```

Teraz môžete aplikáciu prvý krát spustiť v emulátore platformy Azure (klávesom F alebo ikonou zelenej šípky na paneli nástrojov).



### *Prvé spustenie rozpracovanej aplikácie*

Obsah zobrazenej stránky je definovaný v súbore HomeController.cs. Šablóna implicitne obsahuje kód pre stránku, ktorú vidíte na obrázku.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MvcApplication1.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewBag.Message = "Welcome to ASP.NET MVC!";
            return View();
        }

        public ActionResult About()
        {
            return View();
        }
    }
};
```

Tento controller o chvíľu zmeníme, presnejšie zmení sa sám na základe definovaných tried a údajov. Preto začneme najskôr pracovať na databázovej vrstve.

### Vytvorenie dátového modelu

Aktivujte kontextové menu na priečinok Models a vyberte položku Add – Class. Triedu pomenujte napríklad ToDoModels.cs.

### Obsah súboru nahradte kódom

```
using System;
using System.Collections.Generic;
```

```

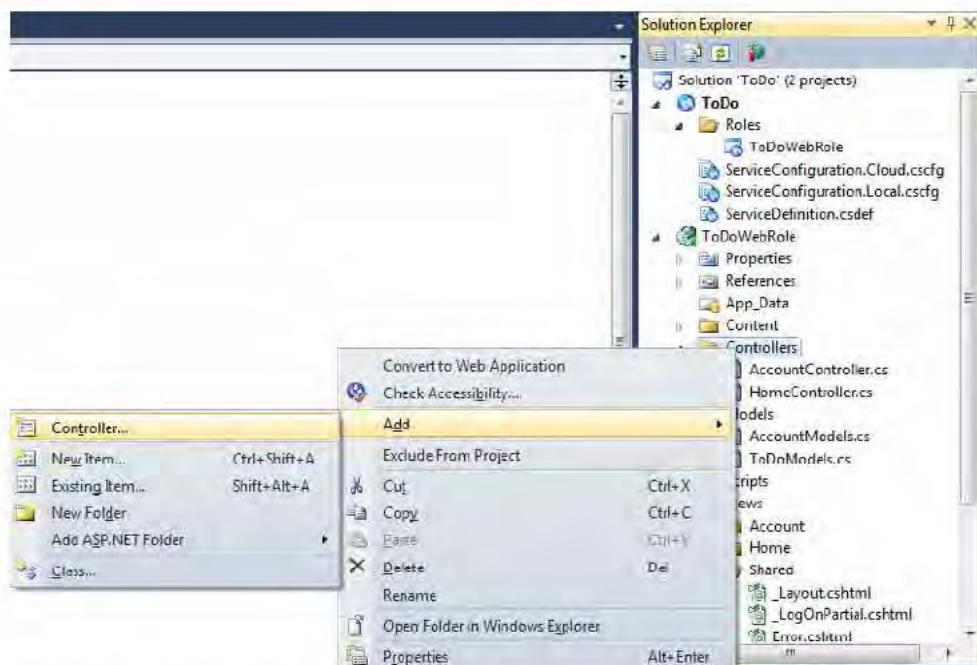
using System.Linq;
using System.Web;
using System.Data.Entity;

namespace ToDoWebRole.Models
{
    public class ToDoItem
    {
        public int ToDoItemId { get; set; }
        public string Name { get; set; }
        public bool IsComplete { get; set; }
    }

    public class ToDoDb : DbContext
    {
        public DbSet<ToDoItem> ToDoItemEntries { get; set; }
    }
}

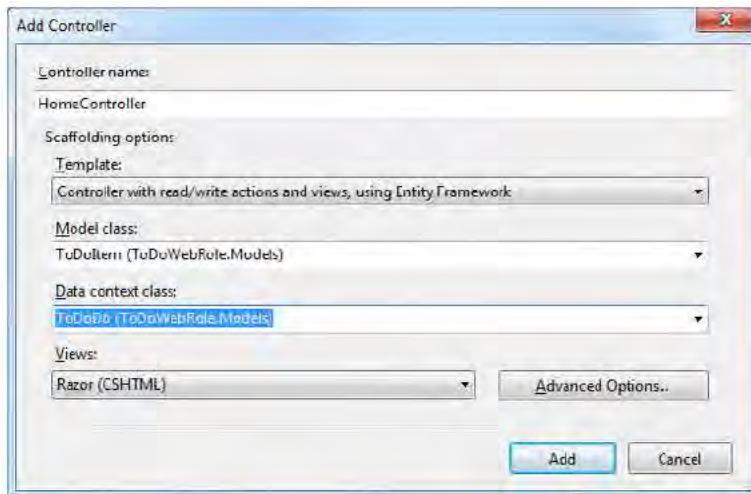
```

Aktivujte kontextové menu na priečinok **Controllers** a pomocou položky **Add** pridajte objekt typu **Controller**.



#### *Pridanie objektu typu Controller*

V dialógovom okne Add Controller zadajte názov controllera **HomeController** (takýto controller už existuje, preto po potvrdení musíte súhlasiť s jeho prepísaním). V položke **Template** vyberte **Controller with read/write actions and views, using Entity Framework template**. Ako model vyberte triedu **ToDoItem** a ako data context class vyberte **ToDoDb**. Situáciu názorne ilustruje obrázok:



*Dialóg Add Controller*

Po potvrdení tlačidlom **Add** budete upozornení, že už existuje. Potvrďte obidve voľby: **Overwrite HomeController.cs** aj **Overwrite associated views**.

V súbore **HomeController.cs** budú doplnené metódy pre databázové operácie pre objekty **ToDoItem** (**create, read, update, delete**).

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using ToDoWebRole.Models;

namespace ToDoWebRole.Controllers
{
    public class HomeController : Controller
    {
        private ToDoDb db = new ToDoDb();

        //
        // GET: /Home/

        public ViewResult Index()
        {
            return View(db.ToDoItemEntries.ToList());
        }

        //
        // GET: /Home/Details/5

        public ViewResult Details(int id)
        {
            ToDoItem todoitem = db.ToDoItemEntries.Find(id);
            return View(todoitem);
        }

        //
        // GET: /Home/Create

        public ActionResult Create()
        {
```

```

        return View();
    }

    //
    // POST: /Home/Create

    [HttpPost]
    public ActionResult Create(ToDoItem todoitem)
    {
        if (ModelState.IsValid)
        {
            db.ToDoItemEntries.Add(todoitem);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        return View(todoitem);
    }

    //
    // GET: /Home/Edit/5

    public ActionResult Edit(int id)
    {
        ToDoItem todoitem = db.ToDoItemEntries.Find(id);
        return View(todoitem);
    }

    //
    // POST: /Home/Edit/5

    [HttpPost]
    public ActionResult Edit(ToDoItem todoitem)
    {
        if (ModelState.IsValid)
        {
            db.Entry(todoitem).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        return View(todoitem);
    }

    //
    // GET: /Home/Delete/5

    public ActionResult Delete(int id)
    {
        ToDoItem todoitem = db.ToDoItemEntries.Find(id);
        return View(todoitem);
    }

    //
    // POST: /Home/Delete/5

    [HttpPost, ActionName("Delete")]
    public ActionResult DeleteConfirmed(int id)
    {
        ToDoItem todoitem = db.ToDoItemEntries.Find(id);
        db.ToDoItemEntries.Remove(todoitem);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
{

```

```

        db.Dispose();
        base.Dispose(disposing);
    }
}

```

Do súboru Web.config do sekcie <configuration> / <connectionStrings> je potrebné pridať connection string.

```
<add name="ToDoDb" connectionString="data source=.\SQLEXPRESS;Integrated Security=SSPI;
Initial Catalog=ToDoDb;User Instance=true;MultipleActiveResultSets=True"
providerName="System.Data.SqlClient" />
```

Súbor HomeController.cs máme zmenený podľa databázovej vrstvy, takže môžeme aplikáciu znova zostaviť a spustiť, aby sme sa presvedčili ako sa zmeny prejavia vizuálne.



*Spustenie upravenej aplikácie.*

Pripomíname, že kód v súbore HomeController.cs bol vygenerovaný automaticky na základe údajov o triedach typu Model class a Data context class.

Kliknutím na tlačidlo Create New sa zobrazí formulár pre zadanie novej položky.

*Formulár pre zadanie novej položky.*

Aby ste si uvedomili súvislosti, pripomenieme, ako bola navrhnutá trieda ToDoItem.

```
public class ToDoItem
{
    public int ToDoItemId { get; set; }
    public string Name { get; set; }
    public bool IsComplete { get; set; }
}
```

Na základe členských premenných tejto triedy (atribútov databázovej tabuľky) bol automaticky vygenerovaný formulár pre zadanie novej položky.

Podľa popisu v stati Nasadenie projektu do reálneho cloudového prostredia vypublikujte aplikáciu do clodu. V okne Solution Explorer v kontextovom menu aktivujte položku Publish.

### Nastavenie aplikácie pre spoluprácu s databázou SQL Database

V súbore web.ReleaseConfig nastavte v sekcií <configuration> / <connectionStrings> reťazec parametrov pre pripojenie ku clodovej databáze.

```
<connectionstrings>
<add name="ToDoDb" connectionString="data source=<serverName>.database.windows.net;Initial Catalog=ToDoDb;User ID=<user>@<serverName>;Password=<password>;Encrypt=true;Trusted_Connection=false;MultipleActiveResultSets=True" providerName="System.Data.SqlClient" xdt:Transform="SetAttributes" xdt:Locator="Match(name)" />
<add name="DefaultConnection" connectionString="data source=<serverName>.database.windows.net;Initial Catalog=ToDoDb;User ID=<user>@<serverName>;Password=<password>;Encrypt=true;Trusted_Connection=false;MultipleActiveResultSets=True" providerName="System.Data.SqlClient" xdt:Transform="SetAttributes" xdt:Locator="Match(name)" />
</connectionstrings>
```

Teraz môžete aplikáciu otestovať priamo v cloude.



Testovanie aplikácie v cloude

## Príklad komunikácie cez Service Bus

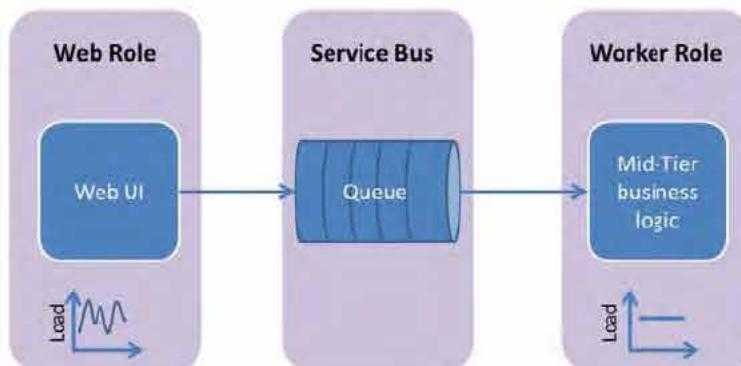
Námetom príkladu je komunikácia medzi používateľským rozhraním aplikácie, ktoré zastrešuje web rola a middlewarovej vrstvy aplikačnej logiky, ktorá je riešená ako worker rola. V príklade sa na sprostredkovanie komunikácie medzi týmito vrstvami (rolami) využíva Service Bus. Komunikácia prebieha prostredníctvom správ. Sprostredkovávaná komunikácia oddeľuje obidve vrstvy. Na rozdiel od priameho posielania správ cez TCP, alebo HTTP, webová vrstva sa nemôže pripojiť priamo k aplikačnej vrstve, ale posielala správy cez Service Bus. Táto sprostredkovateľská vrstva správy spoľahlivo udrží, kým nie je aplikačná vrstva na pozadí pripravená ich prijať a spracovať.

Pre podporu sprostredkovania správ na úrovni Service Bus slúžia dva subjekty fronty a témy (v originálnej terminológii queues a topics). Témy fungujú na princípe publish/subscribe modelu, v ktorom je každá zverejnená správa k dispozícii pre odber pre každú registrovanú tému. Každý odber preto logicky udržuje vlastnú frontu správ. Môže byť vybavený filtrom, ktoré obmedzujú nepotrebné správy.

Tento mechanizmus komunikácie má v porovnaní s priamym posielaním správ niekoľko výhod, hlavne:

**Oddelenie v čase.** Nakoľko správy sú posielané asynchronne, odosielateľ a konzument správy nemusia byť online v rovnakom čase. Služba Busreliably ukladá správy do doby, kedy príjemca bude pripravený ich prijať. To umožňuje, aby jednotlivé zložky distribuovanej aplikácie mohli byť odpojené, či už pre údržbu, alebo v dôsledku havárie komponentov, bez dopadu na systém ako celok. Oddelenie v čase môže preklenúť aj časové pásma, kedy na rôznych kontinentoch sú aplikácie online v určitých denných dobách.

**Vyrovnávanie záťaže.** V mnohých aplikáciach sa zaťaženie systému v priebehu času mení. Sprostredkovanie správ umožní vyrovnávať záťaž worker rolí. To umožní úsporu nákladov, nakoľko sa dá vystačiť s menším množstvom alokovej infraštruktúry.



Komunikácia medzi rolami cez Service Bus

### Nastavenie Service Bus Namespace

Ďalším krokom je vytvorenie menného priestoru služby a získanie zdieľaného tajného kľúča pre prístup. Prihláste sa na Windows Azure Platform Management Portal (<http://windows.azure.com/>). V ľavej časti, v dolnej polovici navigačného panelu portálu kliknite na tlačidlo Service Bus, Access Control & Caching.

The screenshot shows the Windows Azure Platform Management Portal. In the left sidebar, under the 'Services' category, the 'Service Bus' item is highlighted with a red box. The main content area is titled 'About Service Bus, Access Control & Caching'. It contains sections for 'Service Bus', 'Access Control', and 'Caching', each with a brief description and links to further resources like 'SDKs', 'Windows Identity Foundation', and 'Documentation'.

### Windows Azure Platform Management Portal vstup do sekcie Service Bus

V hornej časti ľavého panelu kliknite na položku Service Bus a následne aktivujte funkciu New. Zobrází sa dialóg pre zadanie unikátneho namespace. Tlačidlom Check Availability skontrolujte, či je namespace dostupný. Následne vyberte región v ktorom dátovom centre chcete namespace prevádzkovať.

Pre optimálny výkon aplikácie vyberte rovnaký regón v ktorom prevádzkujete službu Windows Azure a prípadné databázy.

Vytvorenie potvrdte tlačidlom Create Namespace. Vytvorenie môže trvať niekoľko minút.

This dialog box is used to create a new Service Namespace. It has three main sections: 'Available Services' (checkboxes for Access Control, Service Bus, Cache), 'General Properties' (Namespace: 'lackovalsuzba', Country/Region: 'Europe (North)', Subscription: 'Azdem175L141923'), and 'Service Properties' (Cache – Cache Size Quota: '128MB Cache'). At the bottom are buttons for 'Create Namespace' and 'Cancel'.

### Dialóg pre vytvorenie namespace

Po vytvorení sa namespace zobrazí v strednom okne portálu Management. Vpravo na paneli Properties potrebujete hodnotu parametra Default Key. Parameter je skrytý a zobrazíte ho tlačidlom View. Odporučame skopírovať ho do schránky a niekam uložiť.

jMJwDkHp2SA4Z4UVmjooL+SQt78A6/V11UAz18PHGxc=

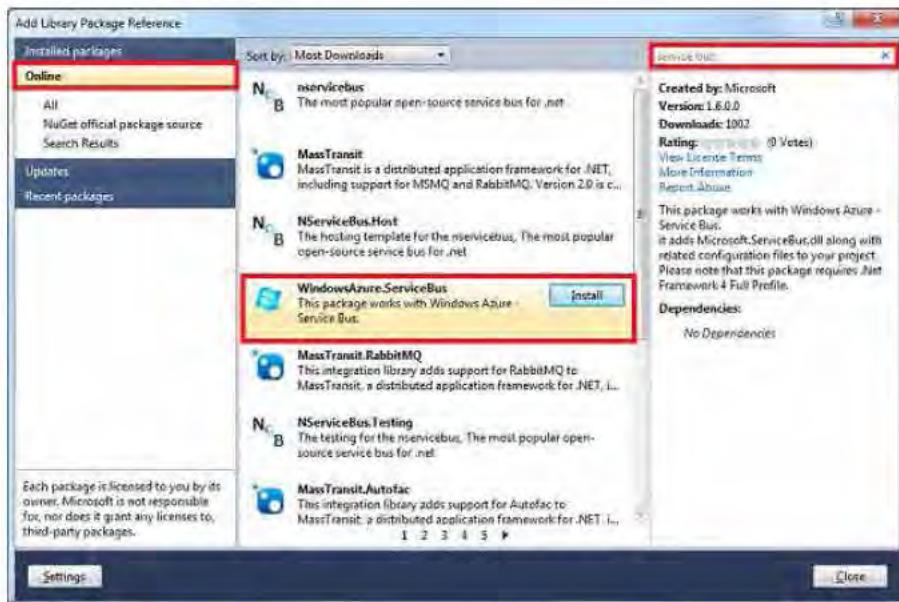
The screenshot shows the Windows Azure Platform interface. On the left, there's a sidebar with links like 'New', 'Modify', 'Delete', 'Refresh', 'Access Control Service', 'Manage Service Bus', 'New Queue', 'New Topic', 'Manage Entities', 'Home', 'Hosted Services, Storage Accounts & CDN', 'Database', 'Data Sync', 'Reporting', 'Service Bus, Access Control & Caching', and 'Virtual Network'. The main area shows a table titled 'Choose Column' with columns 'Name', 'Type', and 'Status'. One row is selected: 'Azdem172.1419 Subscriptions' (Type: Subscription, Status: Active). Below this table, it says 'Queues: 25 ( newest Queues)' and 'Topics: 25 ( newest Topics)'. To the right, a 'Properties' pane displays information such as 'Created on: 3/4/2012 7:14:03 AM UTC', 'Subscription ID: 27d48133f3e54h189b2b74a558846dc', 'Project ID: 3746065e79f046fb70e06a0c8587c9a', 'Service Gateway: https://lackovbus.servicebus.windows.net/', 'Management Endpoint: https://lackovbus-sb.accesscontrol.windows.r', 'ACS Version: ACSV2', and a 'Default Key' section. The 'Default Key' section is highlighted with a red box.

Zobrazenie parametra Default Key

## Vytvorenie projektu

V nástroji Visual Studio 2010 vytvorte aplikácie projekt typu **Windows Azure Project**. Zo zoznamu rolí vyberte **ASP.NET MVC 3 Web Role**. Postup je podrobne popísaný v predchádzajúcich kapitolách.

V okne **Solution Explorer** aktivujte položku kontextového menu **References – Manage NuGet Packages...**. V priečinku **Online** nechajte vyhľadávať „**Service Bus**“ a z nájdených položiek inštalujte **Windows Azure.Service Bus**.



Inštalácia balíčka Windows Azure.Service Bus.

Do projektu v priečinku **Models** pridajte triedu (Solution Explorer – pravé tlačidlo myši na priečinok **Models**, aktivácia položky kontextového menu **Add – Class**). Triedu vhodne pomenujte, napríklad **Sprostredkovatel.cs**. V súbore s rovnakým názvom upravte kód triedy:

```
namespace FrontendWebRole.Models
{
    public class Sprostredkovateľ
    {
        public string Zakazník { get; set; }
        public string Produkt { get; set; }
    }
}
```

V prečinku Controllers otvorte súbor HomeController.cs. Do kódu pridajte namespace pre Service Bus:

```
using FrontendWebRole.Models;
using Microsoft.ServiceBus.Messaging;
using Microsoft.ServiceBus;
```

**Zmenťte kód controlera:**

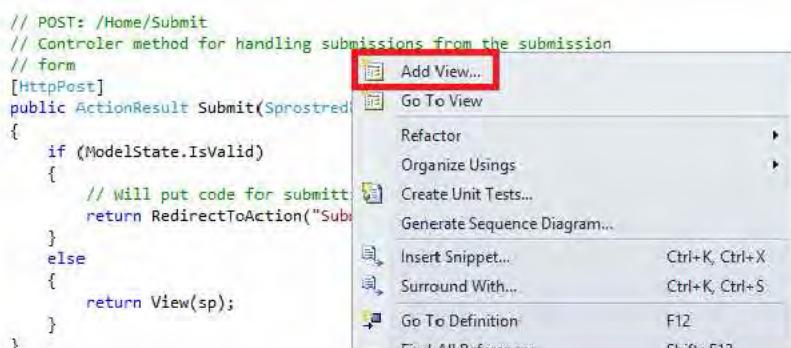
```
namespace FrontendWebRole.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            // Presmerovanie Submit ako vstupného bodu aplikácie
            return RedirectToAction("Submit");
        }

        public ActionResult About()
        {
            return View();
        }

        // GET: /Home/Submit
        public ActionResult Submit()
        {
            // tu neskôr zobrazíme počet správ vo frontu.
            return View();
        }

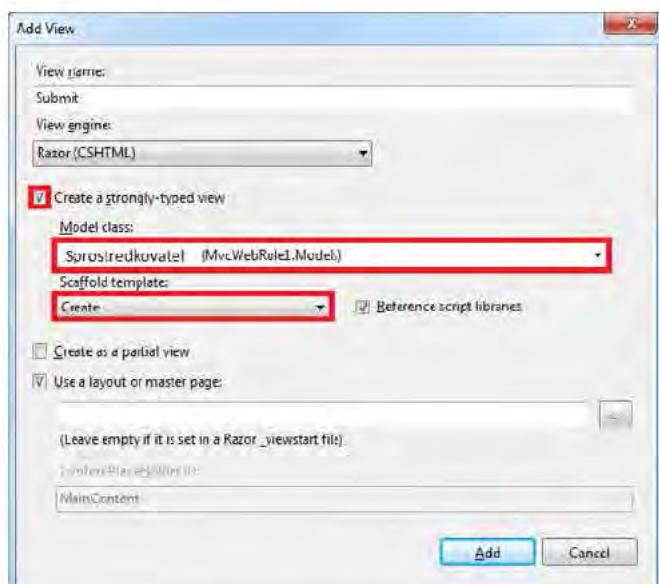
        // POST: /Home/Submit
        [HttpPost]
        public ActionResult Submit(Sprostredkovateľsp)
        {
            if (ModelState.IsValid)
            {
                // tu bude kód pre posielanie sprav do frontu.
                return RedirectToAction("Submit");
            }
            else
            {
                return View(sp);
            }
        }
    }
}
```

Ďalej je potrebné vytvoriť typový pohľad pre metódu controllera Submit(). Kliknite dovnútra kódu tejto metódy a v kontextovom menu aktivujte položku AddView.



#### Aktivovanie položky kontextového menu AddView

V položke Model Class vyberte vami vytvorenú triedu Sprostredkovateľ a v položke Scaffold template vyberte voľbu Create.



#### Dialóg pre pridanie pohľadu

Na základe triedy Sprostredkovateľ sa V priečinku Views\Home\ automaticky vytvorí súbor Submit.cshtml.

```
model Objednavky.Models.Sprostredkovateľ

@{
    ViewBag.Title = "Submit";
}

<h2>Submit</h2>

<script src="@Url.Content("~/Scripts/jquery.validate.min.js")"
type="text/javascript"></script>
```

```

<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>

@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)
<fieldset>
<legend>Sprostredkovateľ</legend>

<div class="editor-label">
    @Html.LabelFor(model => model.Zakaznik)
</div>
<div class="editor-field">
    @Html.EditorFor(model => model.Zakaznik)
    @Html.ValidationMessageFor(model => model.Zakaznik)
</div>

<div class="editor-label">
    @Html.LabelFor(model => model.Produkt)
</div>
<div class="editor-field">
    @Html.EditorFor(model => model.Produkt)
    @Html.ValidationMessageFor(model => model.Produkt)
</div>

<p>
<input type="submit" value="Create" />
</p>
</fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

```

## Za tagom

```
<h2>Submit</h2>
```

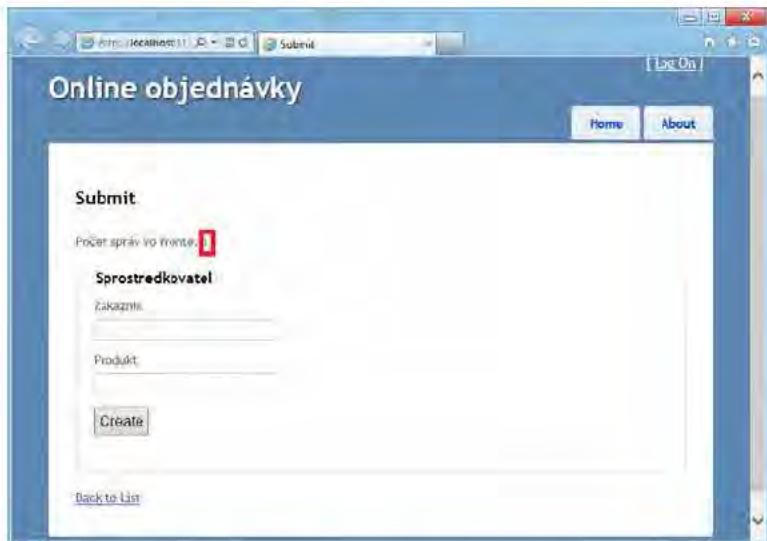
**pridajte zobrazenie počtu správ čakajúcich vo frontu.**

```
<p>Počet správ vo fronte: @ViewBag.MessageCount</p>
```

A napokon jedna kozmetická úprava používateľského rozhrania. V priečinku Views\Shared\ otvorite súbor \_Layout.cshtml a v tagu

```
<h1>My MVC Application</h1>
```

**zmeňte názov aplikácie napríklad na Online objednávky.**



*Používateľské rozhranie front-endovej vrstvy aplikácie (web rola)*

### Posielanie správ do fronty Service Bus

V okne Solution Explorer, zvoľte položku Add kontextového menu aplikovaného na FrontendWebRole. Budete pridávať triedu (položka Class) QueueConnector.cs.

**Do kódu triedy vložte položky Namespace, IssuerName a IssuerKey, ktoré ste získali vo fáze nastavenia Service Bus cez Management Portal.**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Microsoft.ServiceBus.Messaging;
using Microsoft.ServiceBus;

namespace FrontendWebRole
{
    public static class QueueConnector
    {
        // Kód musí byť Thread-safe
        public static QueueClient OrdersQueueClient;

        // Hodnoty získané cez Management Portal
        public const string Namespace = "<VASA HODNOTA>";
        public const string IssuerName = "<VASA HODNOTA>";
        public const string IssuerKey = "<VASA HODNOTA>";

        // Nazov fronty
        public const string QueueName = "OrdersQueue";

        public static NamespaceManager CreateNamespaceManager()
        {
            var uri = ServiceBusEnvironment.CreateServiceUri(
                "sb", Namespace, String.Empty);
            var tP = TokenProvider.CreateSharedSecretTokenProvider(
                IssuerName, IssuerKey);
            return new NamespaceManager(uri, tP);
        }
    }
}
```

```
public static void Initialize()
{
    // Http by malo prejst cez firewally
    ServiceBusEnvironment.SystemConnectivity.Mode =
        ConnectivityMode.Http;
    var namespaceManager = CreateNamespaceManager();

    // Ak fronta neexistuje vytvoríme ju
    if (!namespaceManager.QueueExists(QueueName))
    {
        namespaceManager.CreateQueue(QueueName);
    }

    // Klient
    var messagingFactory = MessagingFactory.Create(
        namespaceManager.Address,
        namespaceManager.Settings.TokenProvider);
    OrdersQueueClient = messagingFactory.CreateQueueClient(
        "OrdersQueue");
}
}
```

Aby sa pri inicializácii aplikácie zavolala vaša metóda `Initialize()` otvorte súbor `Global.asax.cs` v priečinku `Global.asax` a do kódu metódy `Application_Start()` doplňte podčiarknutý riadok.

```
protected void Application_Start()
{
    AreaRegistration.RegisterAllAreas();

    RegisterGlobalFilters(GlobalFilters.Filters);
    RegisterRoutes(RouteTable.Routes);
QueueConnector.Initialize();
}
```

Finálnym krokom pre web rolu bude doplnenie pripojenia cez konektor fronty do kódu metód `Submit()` a `Submit(Sprostredkovatel sp)` v súbore `Controllers\HomeController.cs`.

```
public ActionResult Submit()
{
    var namespaceManager = QueueConnector.CreateNamespaceManager();
    var queue = namespaceManager.GetQueue(QueueConnector.QueueName);
    ViewBag.MessageCount = queue.MessageCount;

    return View();
}

// POST: /Home/Submit
// Controller method for handling submissions from the submission
// form
[HttpPost]
public ActionResult Submit(Sprostredkovatel sp)
{
```

```

if (ModelState.IsValid)
{
    var message = new BrokeredMessage(sp);
    QueueConnector.OrdersQueueClient.Send(message);
    return RedirectToAction("Submit");
}
else
{
    return View(order);
}
}

```

## Vytvorenie Worker Role

Finálnym krokom vytvorenia našej dvojvrstvovej aplikácie bude vytvorenie vrstvy na pozadí, ktorá bude fungovať ako worker role a bude spracovávať požiadavky, v našom prípade objednávky, ktoré budú posielané cez Service bus z web role.

Aktivujte kontextové menu na projekt a kliknite na položku **New Worker Role Project**. Projekt vhodne pomenujte, napríklad RolaSpracovaniaObjednavok. Pridajte referencie na System.ServiceModel.dll, Microsoft.ServiceBus.dll a System.Runtime.Serialization.dll.

Aj v tejto roli vytvorte triedu QueueConnector a pomocou položky menu **Add – ExistingItem** skopírujte do nej kód z rovnomennej triedy vo web role v súbore QueueConnector.cs.

Rovnako postupujte aj v prípade triedy Sprostredkovateľ.

Nakoniec upravte kód v súbore WorkerRole.cs. V reálnej aplikácii by worker role „pracovala“, teda reálne spracovala objednávku, uložila niečo do databázy, niečo vypočítala... Nakoľko ide o cvičnú aplikáciu jej úlohou bude len vypisovať správy, ktoré dostáva zo Service Bus. Aby sme simulovali zaneprázdnenosť použijeme Sleep().

```

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Net;
using System.Threading;
using Microsoft.WindowsAzure;
using Microsoft.WindowsAzure.Diagnostics;
using Microsoft.WindowsAzure.ServiceRuntime;
using Microsoft.WindowsAzure.StorageClient;
using Microsoft.ServiceBus;
using Microsoft.ServiceBus.Messaging;
using FrontendWebRole;
using FrontendWebRole.Models;

namespace OrderProcessingRole
{
    public class WorkerRole : RoleEntryPoint
    {
        public override void Run()
        {
            while (true)
            {

```

```

        // Receive the message
        BrokeredMessage receivedMessage =
            QueueConnector.OrdersQueueClient.Receive();
        if (receivedMessage != null)
        {
            try
            {
                // zobrazenie spravy
                Sprostredkovatelsp = receivedMessage.GetBody();
                Trace.WriteLine(
sp.Zakaznik + ":" + sp.Produkt,
                    "Sprava je spracovávaná");

                // odstránenie správy z fronty
                receivedMessage.Complete();
            }
            catch (MessagingException)
            {
                receivedMessage.Abandon();
            }
        }
        else
        {
            Thread.Sleep(10000);
        }
    }

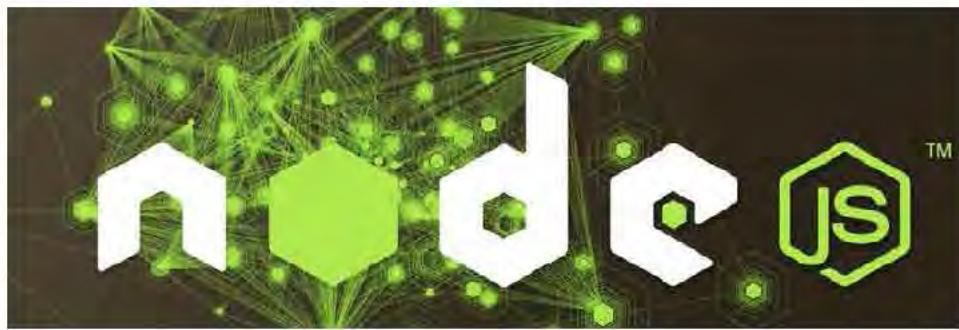
    public override bool OnStart()
    {
        // MAX počet paralelných pripojení
        ServicePointManager.DefaultConnectionLimit = 12;
        QueueConnector.Initialize();
        return base.OnStart();
    }
}
}
}

```

Teraz môžete aplikáciu spustiť v emulátore a cez Windows Azure Compute Emulator sledovať ako sú posielané správy.

## Vytvorenie aplikácie pre platformu Node.js a jej publikovanie na Azure

Popri ASP.NET a PHP je veľmi populárnochou platformou pre webové aplikácie framework node.js. Ako vyplýva z názvu, základným princípom je použitie populárneho JavaScriptu na serveri. Tento server-side javascript framework je vybudovaný nad V8 JavaScript engine od Google a vyznačuje sa jednoduchosťou. Začínajúci používateľ sa vôbec nemusí zaoberať jeho vnútorným fungovaním ani implementačnými detailmi. Z hľadiska používateľa sa javí ako „jednovláknový“ JavaScript. Podobne ako klasické frameworky je aj node.js „poháňaný“ udalosťami.



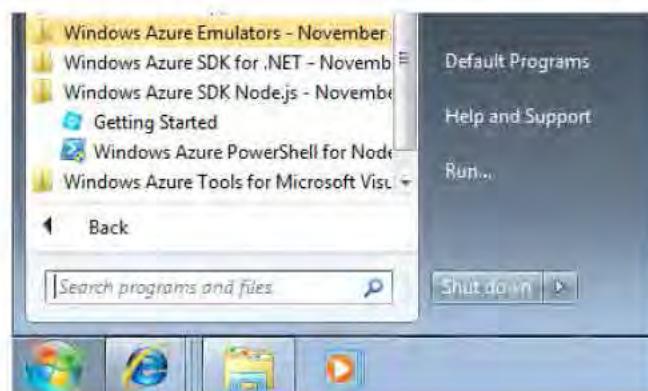
*Logo Node.js*

Pre nasadenie aplikácie na platforme Node.js je potrebné z [www.windowsazure.com](http://www.windowsazure.com) priečinok DEVELOP stiahnuť a nainštalovať Windows Azure SDK for Node.js.



#### *Inštalácia Windows Azure SDK for Node.js*

Po nainštalovaní SDK pribudne v menu operačného systému priečinok Windows Azure SDK for Node.js a v nej položka Windows Azure PowerShell for Node.js.



#### *Windows Azure PowerShell for Node.js v menu operačného systému Windows*

Sprístupnite Windows Azure PowerShell for Node.js, v režime „Spustiť ako správca“. Vytvorte nový priečinok NODE na vašom lokálnom disku a prepnite sa do novo vytvoreného priečinka.

```
mkdir C:\node
cd node
```

Následne vytvorte novú službu s vhodným názvom, napríklad:

```
New-AzureService priponienky
```

```
Administrator: Windows Azure PowerShell for Node.js
PS C:\> mkdir C:\node
    Directory: C:\

Mode                LastWriteTime     Length Name
----                -----          ---- 
d---        27.2.2012      11:06      node

PS C:\> cd node
PS C:\node> New-AzureService priponienky

Service has been created at C:\node\priponienky

PS C:\node\priponienky>
```

*Vytvorenie novej služby v prostredí „Power Shell konzoly“ Windows Azure PowerShell for Node.js*

Pričom DIR si môžete nechať vypísať obsah priečinka služby.

```
PS C:\node\priponienky> dir
    Directory: C:\node\priponienky

Mode                LastWriteTime     Length Name
----                -----          ---- 
-a---        27.2.2012      11:08      67 deploymentSettings.json
-a---        27.2.2012      11:08      280 ServiceConfiguration.Cloud.cscfg
-a---        27.2.2012      11:08      280 ServiceConfiguration.Local.cscfg
-a---        27.2.2012      11:08      240 ServiceDefinition.csdef
```

Nasleduje pridanie webovej role prostredníctvom „cmdletu“ Add-AzureNodeWebRole za ktorým môže nasledovať názov role. Ak názov role nezadáte, bude implicitne pomenovaná WebRole1.

```
PS C:\node\priponienky> Add-AzureNodeWebRole
```

```
Role has been created at C:\node\priponienky\WebRole1. For easy access to Windows
Azure services from your application code, install the Windows Azure client
library for Node.js by running 'npm install azure'.
```

Môžete preskúmať jej priečinok.

```
PS C:\node\priponienky\WebRole1> dir
    Directory: C:\node\priponienky\WebRole1

Mode                LastWriteTime     Length Name
----                -----          ---- 
d---        27.2.2012      12:12      bin
-a---        8.12.2011      11:54      214 server.js
```

```
-a--- 8.12.2011 11:54 1187 Web.cloud.config
-a--- 8.12.2011 11:54 1006 Web.config
```

V súbore server.js je vzorový kód aplikácie typu Hello World.

```
var http = require('http');
var port = process.env.port || 1337;
http.createServer(function (req, res) {
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    res.end('Hello World\n');
}).listen(port);
```

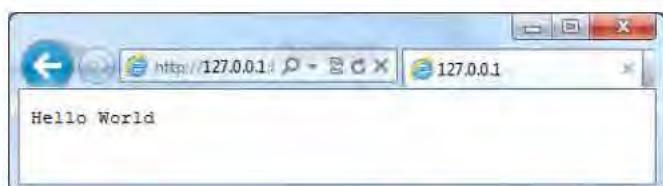
## Test aplikácie v emulátoru

Spuštite emulátor platformy Azure.

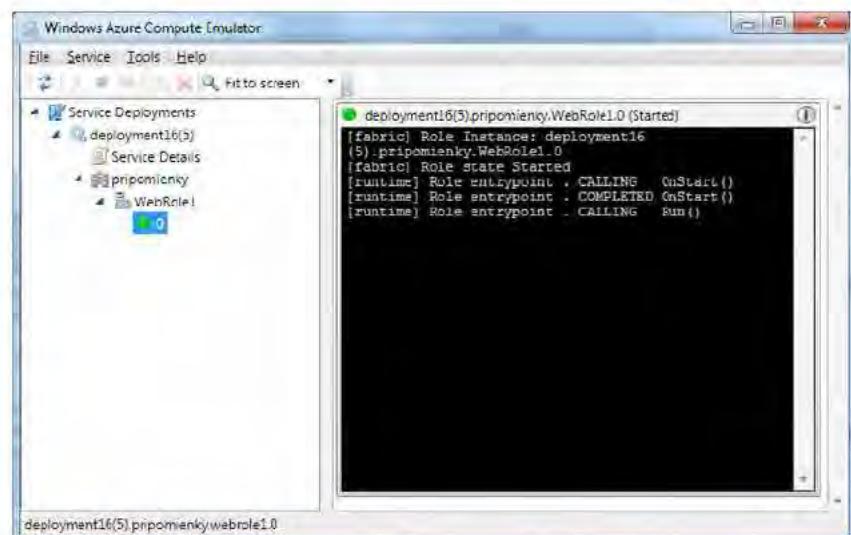
```
PS C:\node\priponienky\WebRole1> Start-AzureEmulator -launch

Creating local package...
Starting Emulator...
Role is running at http://127.0.0.1:81
Started
```

Po nábehu emulátora sa zobrazí adresa URL na ktorej je možné pristupovať k aplikácii cez prehliadač webového obsahu.



*Spustenie node.js aplikácie prostredníctvom emulátora platformy Azure*



*Podobne ako v príkladoch na platorme .NET si môžete nechať zobraziť stav emulátora, či už pre Compute Emulator, alebo pre Storage Emulator.*

## Nasadenie aplikácie do cloutu

Po odladení aplikácie v emulátore ju môžete nasadiť do reálneho prostredia cloudovej platformy Windows Azure. Ako prvý krok je potrebné nastaviť parametre pre nasadenie. V našom prípade nastavíme, že aplikácia sa nasadí na dve inštancie virtuálnych serverov.

```
PS C:\node\priponienky\WebRole1>Set-AzureInstances WebRole1 2
```

Následne je potrebné stiahnuť nastavenie pre publikovanie.

```
PS C:\node\priponienky\WebRole1> Get-AzurePublishSettings
```

Do lokálneho vývojárskeho počítača si stiahnite ponúknutý XML súbor s oprávneniami a príkazom Import ho spárujte so svojim riešením príkazom Import-AzurePublishSettings <publish-settings-file>, v našom príklade to bolo konkrétnie:

```
PS C:\node\priponienky\WebRole1>Import-AzurePublishSettings  
c:\node\Azdem175L14192J-2-27-2012-credentials.publishsettings
```

**Vo finálnej fáze aplikáciu vypublikujete do cloutu pomocou cmdletu Publish-AzureService.**

```
PS C:\node\priponienky\WebRole1>Publish-AzureService -name WebRole1
```

Táto akcia trvá niekoľko minút a potom sa aplikácia spustí priamo z cloutu.

## Záver

Cloud computing stal vlajkovým trendom súčasného IT. Firmám prináša výhody najmä vďaka širokým možnostiam škálovateľnosti a cene, a umožňuje tak vytvárať moderné zákaznícke riešenia. Okrem štandardizovaných procesov v cloude umožňuje používať úložiská s obrovskou kapacitou, či rýchlo a lacno vytvárať, testovať a distribuovať webové služby.

Cloudová platforma Windows Azure poskytuje širokú paletu služieb:

- **Infraštruktúra ako služba (IaaS)**, vrátane služby Windows Azure Virtual Machines, ktorá umožní zákazníkom vytvoriť a manažovať aplikácie bežiace na operačnom systéme Windows a Linux, ako aj Windows Azure Virtual Network, vďaka ktorému bude možné riadiť virtuálne súkromné siete (VPNs) vo Windows Azure a vytvoriť bezpečné prelinkovanie s lokálnou infraštruktúrou.
- **Webové lokality Windows Azure**, ktoré umožnia vývojárom rýchlo a jednoducho spustiť weby s podporou rôznych platform a oblúbených open source aplikácií vrátane ASP.NET, PHP a „Node.js“.
- **Azure Cloud Services (PaaS)** umožňuje spúšťanie aplikácií vo virtuálnom prostredí, ktoré sa spravuje plne v rézii poskytovateľa, vrátane monitorovania a obnovy po prípadnom zlyhaní. Pri tomto modeli poskytovania služby zákazník nemusí nič spravovať.

Novinky týkajúce sa Windows Azure zahŕňajú aj vylepšenia už existujúcich služieb:

- **Aktualizácia interoperability** – možnosť tvorby a spustenia aplikácie na Windows Azure v jazyku a na platforme podľa výberu vývojára.
- **Vylepšená vývojárska produktivita** s aktualizáciami pre Windows Azure SDK a podporou Visual Studio Team Foundation Services.
- **Zlepšené aplikačné služby** vrátane aktualizácie Windows Azure Active Directory, Content Delivery Network (CDN), distribuovaného caechovania a podpory prenosu mediálneho obsahu v službe Media Services.

Hlavnou výhodou platformy Windows Azure je filozofia: **platíte len za to, čo práve potrebujete a to bez dlhodobých záväzkov alebo viazanosti.** Medzi ďalšie výhody je možné zaradiť potenciálne zníženie nákladov na personál, softvér, hardvér, nehnuteľnosti, chladenie, energiu, licencie a údržbu. Toto zníženie nákladov v praxi znamená, že kapitál možno uvoľniť na akékoľvek iné použitie na splnenie strategických firemných cieľov. Riešením problémov sa bude zaoberať poskytovateľ služby, nie vnútropodnikový personál.

