Praktické cvičenie

Asymetrické šifrovanie dát pre SQL Azure

Spracované 30.1.2012

Ciele praktického cvičenia

Praktické cvičenie vás prevedie krokmi vytvorenia servera SQL Azure, nastavenia jeho zabezpečenia a šifrovaním dát pre perzistentné ukladanie.

Po dokončení cvičenia získate znalosti z týchto oblastí:

- Vytvorenie a zabezpečenie servera SQL Azure
- Vytvorenie databázy SQL Azure
- Nasadenie certifikátu na Windows Azure
- Asymetrické šifrovanie dát pre ukladanie do databázy SQL Azure

Stručný popis

SQL Azure umožňuje nasadiť do dátového centra relačnú databázu pre použitie v typických scenároch – databáza uceleného cloud riešenia, mobilne dostupná databáza, databáza vyžadujúca výkonnosť, geograficky distribuovaná databáza atď. Koncepcia SQL Azure vychádza z produktovej rady Microsoft SQL Server, čím sa zjednodušuje migrovanie dát a kódu z lokálneho "SQL Server riešenia" do cloudu. SQL Azure v porovnaní s SQL Server-om zavádza sadu nových technológií požadovaných konceptom cloudu (napr. SQL Azure Federation), zároveň však neposkytuje niektoré funkcie/komponenty známe z SQL Server-a. (Zoznam funkcií, ktoré nie sú v súčasnosti implementované na SQL Azure, nájdete na MSDN.)

SQL Azure poskytuje rovnaké TDS (tabular data stream – tabuľkový tok dát) rozhranie pre komunikáciu medzi klientom a serverom ako SQL Server. Preto pre tvorbu klientských aplikácií pre cloud dáta môžete použiť zaužívané nástroje a knižnice. Požiadavky klientských aplikácií sú presmerovávané balancermi záťaže na TDS Gateway pre optimálne využitie fyzických serverov a služieb v dátových centrách. Úlohou TDS Gateway je premostenie medzi vašou aplikáciou a podkladovou platformou, v ktorej sídlia dáta. Poskytuje funkcie izolácie, „provisioningu" , účtovania a merania spotreby, presmerovávania spojenia. Podkladová platforma pozostáva z viacerých inštancií SQL Server-a, z ktorých každá je spravovaná distribuovaným výpočtovým systémom pozostávajúcim z integrovaných sietí, serverov a úložiska. (Takéto systémy sa často označujú termínom "fabric".) Takáto topológia umožňuje automatickú ochranu proti výpadkom, rozklad záťaže a automatickú replikáciu medzi fyzickými servermi.

Predpoklady zvládnutia praktického cvičenia

- Základné znalosti Microsoft SQL Server-a a vývoja nad .NET Frameworkom
- Visual Studio 2010 ale Visual Web Developer Express 2010
- Windows Azure Tools z balíčka Windows Azure SDK
- Prístup na portal Windows Azure Platform (v ČR v rámci benefitov predplatného MSDN, programu BizSpark, komerčne zakúpených služieb Windows Azure, na Slovensku získaním dočasného Azure Pass prístupu.)

Viac informácií

Príručka **“**Technický popis Windows Azure Platform**”**

Scenár

V praktickom cvičení vytvoríte SQL Azure server a následne SQL Azure databázu. Potom vytvoríte hostovanú službu na Windows Azure, do ktorej nasadíte jednoduchú webovú aplikáciu ukladajúcu dáta do databázy SQL Azure asymetricky šifrované algoritmom RSA použitím X.509 certifikátu nasadeného do inštancie webovej role Windows Azure.

Predpokladaný čas potrebný pre dokončenie praktického cvičenia

60 minút

Pomôcky

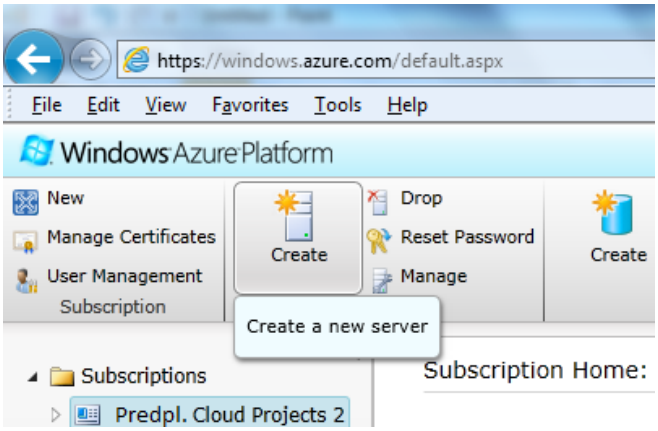
- Windows Azure Platform Training Kit

Cvičenie č.1: Vytvorenie servera a databázy SQL Azure

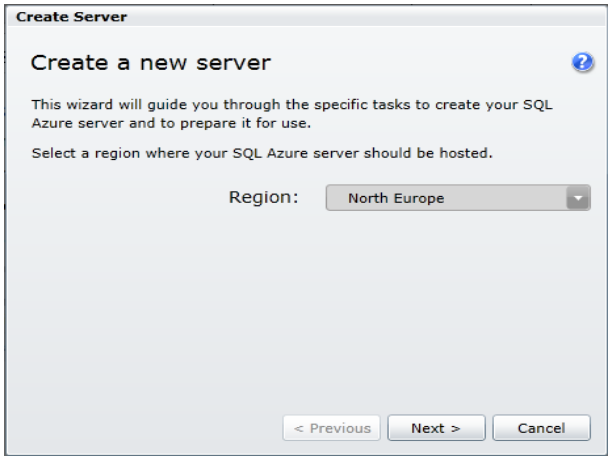
V tomto cvičení sa prihlásite na portál Windows Azure Platform, vytvoríte databázový server SQL Azure a následne databázu.

Úloha č.1 – Vytvorenie servera SQL Azure

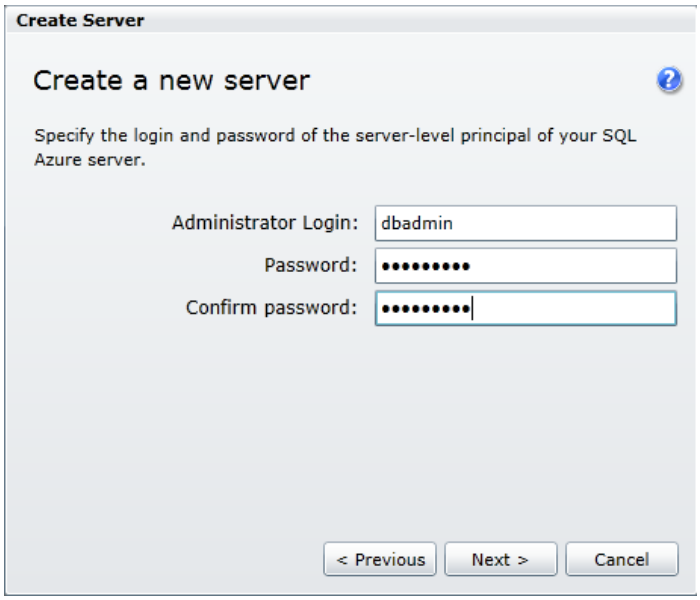
1. Spustite **Internet Explorer** a prejdite na URL adresu portálu Windows Azure Platform, https://windows.azure.com. Portál Windows Azure Platform je miestom správy vašich služieb prevádzkovaných v cloude.
2. Prihláste sa použitím LiveID vášho Azure predplatného alebo prístupu "Azure Pass".
3. V ľavom dolnom menu portálu vyberte "**Database"**.
4. Z ľavého horného menu vyberte zo stromu "Subscriptions" predplatné, pod ktorým chcete vytvoriť databázový server.
5. Zo zobrazenej ponuky horného "ribbon" menu vyberte "**Create"** v ribbon skupine "Server"**.**



6. Vyberte región dátového centra, v ktorom chcete databázový server vytvoriť a potvrďte tlačidlo "Next".

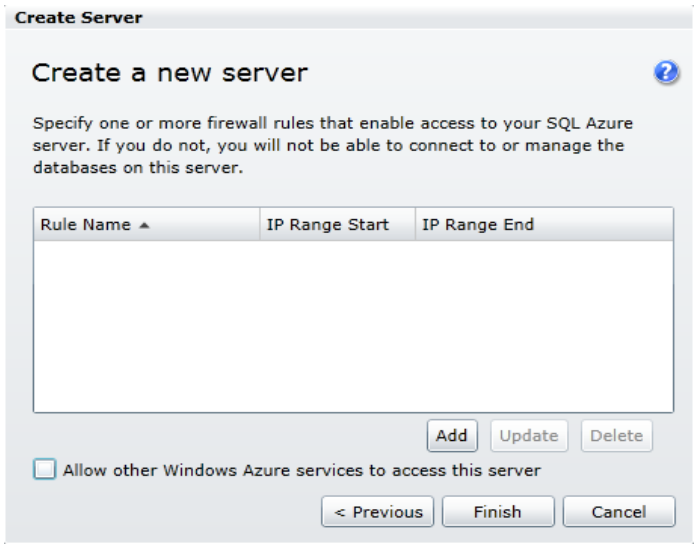


7. Navrhnite a zadajte meno a heslo správcovského účtu nového databázového servera. Potvrďte tlačidlo “Next”.

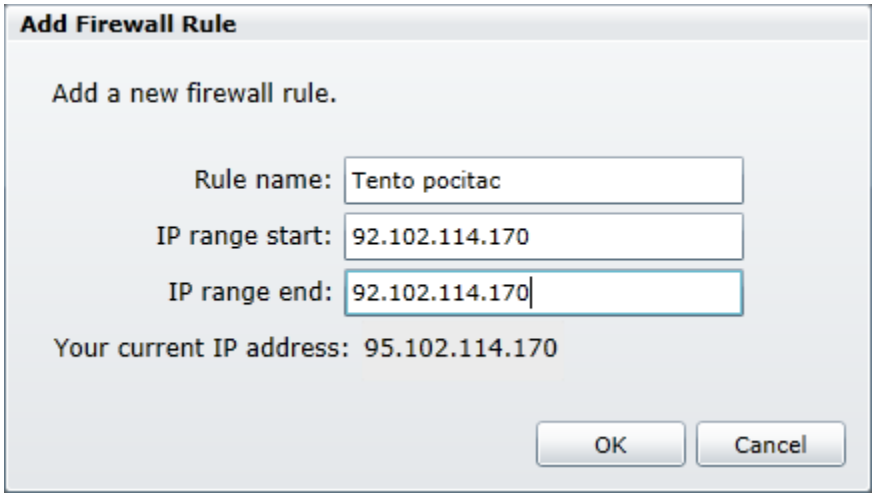


Poznámka: Správcovský účet je účet s najväčšími právami na správu databázového servera. Z bezpečnostných dôvodov sa vyhnite používaniu tohto účtu v aplikačných reťazcoch spojenia (“connection string-och”).

8. Zobrazí sa dialógové okno, v ktorom môžete nastaviť pravidlá firewallu pre prístup k databázovému serveru. Okno “**Firewall Rules”** vám umožní špecifikovať zoznam IP adries, ktoré budú môcť pristupovať k novému serveru SQL Azure. Firewall je prednastavený tak, že zamieta všetky pokusy o spojenie, takže aby ste klientským aplikáciam umožnili pristupovať k serveru, musíte do pravidiel firewallu pridať povolené IP adresy alebo rozsah povolených IP adries.



9. Vyberte nastavenie “Allow other Windows Azure services to access this server”, čím povolíte prístup k databázovému serveru z iných služieb dátového centra. Toto nastavenie je pre náš scenár potrebný, pretože k databáze SQL Azure bude pristupovať webová rola Windows Azure z identického dátového centra.
10. Potvrďte tlačidlo “Add” pre pridanie nového pravidla firewallu.
11. Pre konfiguráciu budeme potrebovať aj povolenie prístupu z lokálneho počítača. Preto musíme do zoznamu povolených IP adries pridať minimálne aktuálnu adresu nášho počítača. **Opíšte z položky “Your current IP address” (“Vaša aktuálna IP adresa”) hodnotu IP adresy do položiek “IP range start” a “IP range end”, tak ako sa zobrazuje na vašom počítači.** (Vytvoríte teda pravidlo s rozsahom jednej IP adresy. Ak by ste potrebovali povoliť k serveru SQL Azure prístup z všetkých možných IP adries, stačí zadať rozsah od IP adresy 0.0.0.0 po 255.255.255.255.)

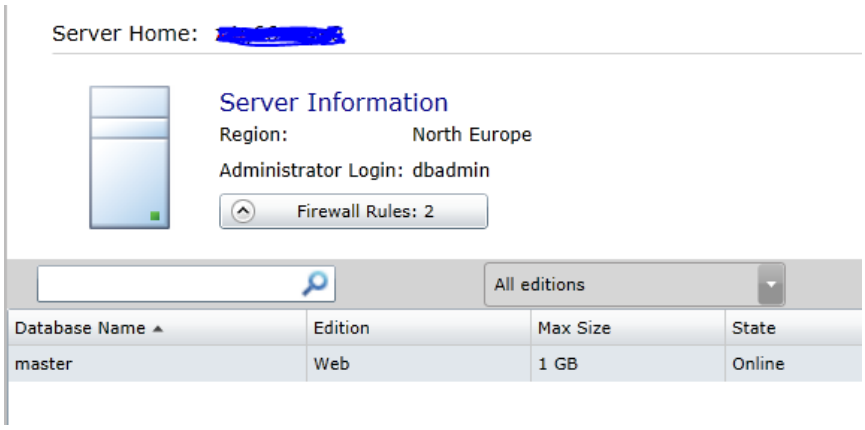


> **Poznámka:** Nastavenie firewallu sa dá konfigurovať nielen cez portál Windows Azure Platform, ale aj pomocou PowerShell skriptov, ktoré používajú SQL Azure Management REST API.

12. Potvrďte tlačidlo “OK” a potom “Finish”.
13. Do zoznamu databázových serverov sa pridá nová položka. Úplne meno servera, pod ktorým je dostupný aplikáciam z internetu aj z dátového centra, je v stĺpci “Fully Qualified Server Name”.

> **Poznámka:** Úplný doménový názov servera má nasledujúci formát:
>
> *<NazovServera>.database.windows.net*
>
> kde *<NazovServera>* identifikuje server.

14. Rozbaľte uzol “Subscriptions” v ľavom hornom menu až na úroveň serverov a vyberte v ňom nový databázový server. V hlavnom okne portálu sa zobrazia informácie o vybranom databázovom servri.



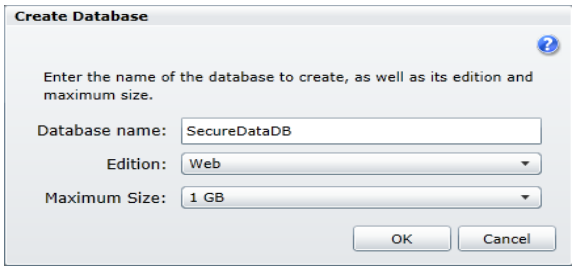
V dátovom centre sme v tejto úlohe vytvorili databázový server, na ktorom v ďalšom kroku vytvoríme databázu.

Úloha č.2 – Vytvorenie aplikačného prístupu a databázy na servri SQL Azure

Na správu databáz SQL Azure môžete použiť viacero dostupných nástrojov. Doporučovaným klientským nástrojom je SQ L Server Management Studio 2012, doporučovaným online nástrojom je „Database Manager for SQL Azure“ , ktorý je dostupný priamo z portálu Windows Azure Platform . V tejto časti praktického cvičenia vytvoríte použitím nástroja „Database Manager for SQL Azure“ SQL účet na prístup do SQL Azure a databázu, ktorú neskôr použijete ako perzistentné úložisko šifrovaných dát.

1. Vyberte v ľavom dolnom menu portálu Windows Azure Platform položku “Database”.
2. V ľavom hornom menu rozbaľte uzol “**Subscriptions”** až do úrovne mena servera a vyberte SQL Azure server, na ktorom chcete vytvoriť databázu.
3. V ribbon menu vyberte zo sekcie “Database” položku “Create”.

4. V dialógu “Create Database” zadajte do položky “Database name” názov databázy “SecureDataDB”, zvoľte edíciu “Web”, nastavte v položke “Maximum size” maximálnu veľkosť databázy na 1 GB. Potvrďte tlačidlo “OK”.



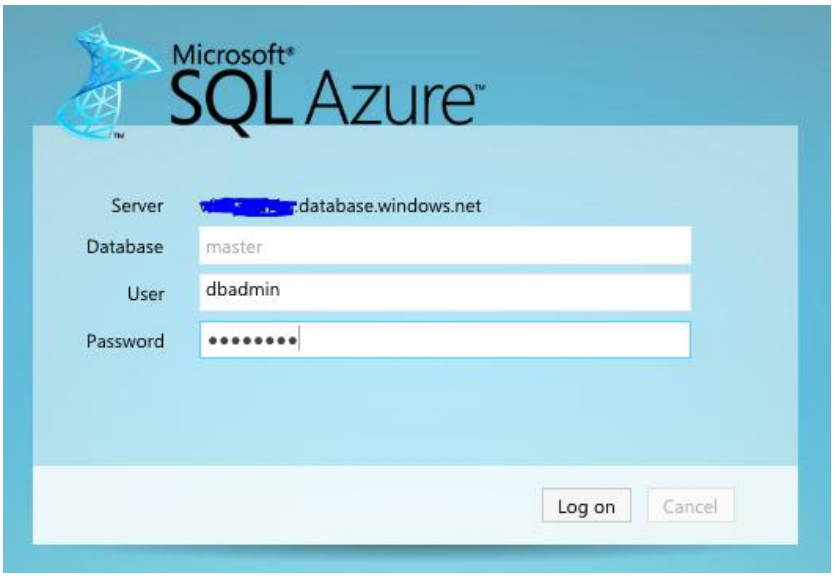
Poznámka: V prípade, že chcete databázu vytvoriť použitím nástroja SQL Server Management Studio, môžete použiť T-Sql príkaz "Create database", v ktorom špecifikujete databázovú edíciu (Web alebo Business) spolu s maximálnou veľkosťou databázy. Napr. pre vytvorenie databázy Business Edition s maximálnou veľkosťou of 30GB použijete T-Sql príkaz:

CREATE DATABASE SecureDataDB (MAXSIZE = 30GB)

Parameter maximálnej veľkosti databázy určuje strop, nad ktorý už nemôžete vložiť dáta bez explicitného zväčšenia databázy.

5. V ľavom hornom menu rozbaľte uzol "**Subscriptions"** až do úrovne názvu databázy. Vyberte databázu "master".
6. V ribbon menu vyberte zo sekcie "Database" položku "Manage".

7. Budete presmerovaný na **SQL Azure Management Portal**. Zadajte meno administrátora servera SQL Azure, jeho heslo a potvrďte tlačidlo "**Log on".**

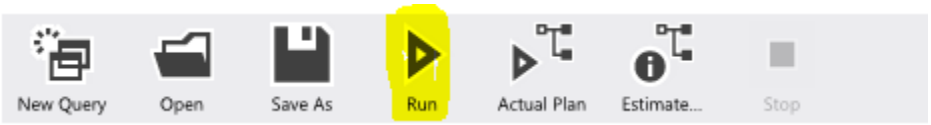


8. Počkajte, pokiaľ sa nenadviaže spojenie s databázou a nezobrazí sa hlavná stránka pre správu databázy. Z bezpečnostných dôvodov sa nedoporučuje používať prístupový účet správcu servera SQL Azure na iné účely, ako je konfigurácia servera a databáz. Preto teraz vytvoríme nový prihlasovací účet do SQL Azure, ktorý budeme používať z aplikácie. (Použijeme pritom zjednodušený model, kedy aplikácia bude bez ohľadu na aplikačného používateľa pristupovať do databázy vždy rovnakým menom/heslom.)
9. Vyberte z ribbon menu položku "New Query".

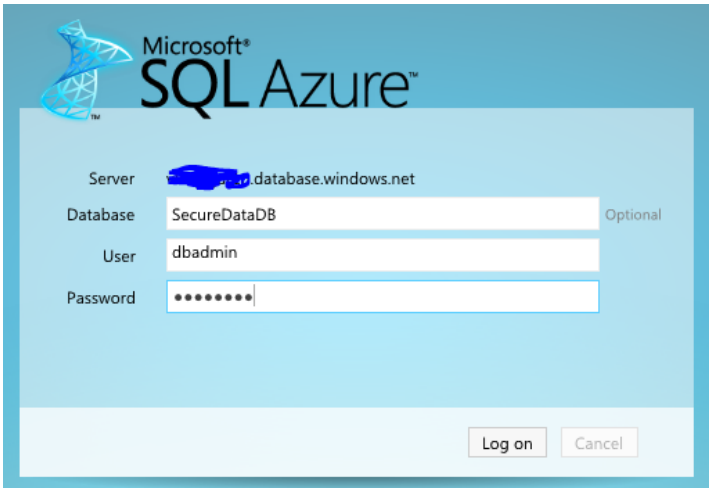
10. Vložte do plochy T-Sql príkazov nasledujúci príkaz na vytvorenie SQL účtu na úrovni servera:

```
CREATE LOGIN aplikuser WITH PASSWORD ='Heslo123'
```

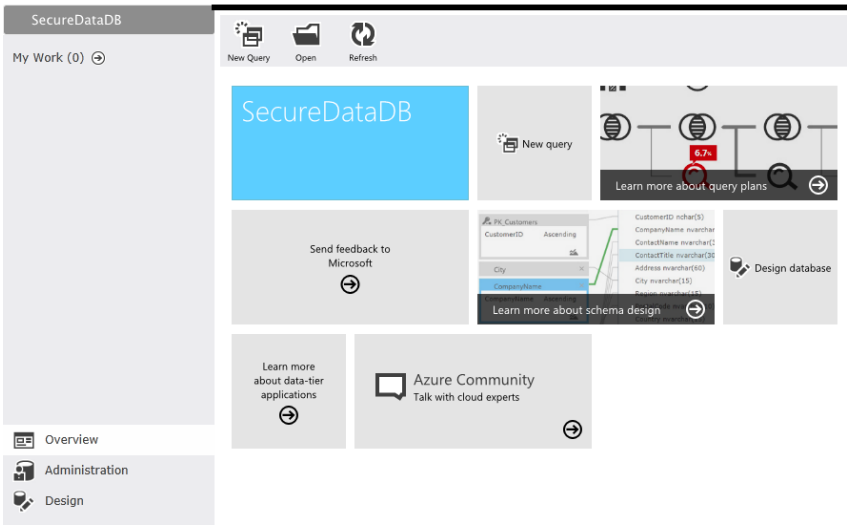
11. Povrďte “Run”.



12. Po spracovaní príkazu ukončite spojenie s databázou “master” potvrdením “Log off” v pravom hornom rohu. Budete opäť presmerovaný na prihlasovací formulár správcu databáz.
13. Zadajte databázu “SecureDataDB”, meno administrátora servera SQL Azure, jeho heslo a potvrďte tlačidlo “**Log on”.**



14. Počkajte, pokiaľ sa nenadviaže spojenie s databázou a nezobrazí sa hlavná stránka pre správu databázy.



15. Vytvorte novú tabuľku v databáze, ktorú v webovej aplikácii použijete na ukladanie “citlivých” dát. Môžete ju vytvoriť pomocou dizajnéra tabuliek dostupného cez dlaždicu “New Table”, alebo potvrdením T-Sql príkazu na vytvorenie tabuľky v okne “New Query”. Uprednostníme druhý spôsob, preto vyberte z ribbon menu položku “New Query”.

16. Vložte do plochy T-Sql príkazov nasledujúce príkazy na vytvorenie prístupu, tabuľky a nastavenia práv na ňu:

```
/*vytvorenie pouzivatela na urovni databazy*/
CREATE USER aplikuser FROM LOGIN aplikuser;
GO

/*vytvorenie tabulky, v ktorej budeme ukladat citlive data*/
CREATE TABLE KritickeData
(
  ID int identity(1,1) PRIMARY KEY,   -- id klienta
  Klient nvarchar(255) NOT NULL,      --  meno klienta
  Hodnota varbinary(max) NOT NULL, -- zasifrovana hodnota napr. heslo, bank.ucet
  PosllednaZmena datetime NOT NULL,  -- datum poslednej modifikacie
  HashToken binary(32) NOT NULL      -- hash zaznamu
)
GO

/*vytvorenie indexu nad stlpcom ID*/
CREATE UNIQUE INDEX IX_idklienta
   ON KritickeData (ID);
GO

/*vytvorenie indexu nad stlpcom Klient*/
CREATE NONCLUSTERED INDEX IX_klient
   ON KritickeData (Klient);
GO

/*pridelenie prav pouzivatelovi aplikuser na tabulku KritickeData*/
GRANT SELECT, INSERT, UPDATE, DELETE ON KritickeData TO aplikuser;
GO
```

Štruktúra tabuľky zodpovedá entite jedného používateľa vašej aplikácie uloženej v jednom riadku tabuľky s citlivými dátami. Položka “ID” je jedinečným identifikátorom používateľa vhodným na prípadný “join” s ostatnými tabuľkami (napr. s adresami, záujmami atď.). Položka “Klient” obsahuje meno a priezvisko používateľa. Položka “Hodnota” obsahuje binárnu

zašifrovanú hodnotu citlivej informácie. Položka “HashToken” je odtlačkom dôležitých hodnôt entity, ktorá nám poslúži na kontrolu, či niekto nezmenil neoprávnene zašifrovanú hodnotu priamo v databáze.

17. Povrďte “Run” .
18. Odstráňte príkazy z T-Sql plochy a vložte do nej nasledujúci príkaz na vytvorenie uloženej procedúry, ktorá zjednoduší aplikačný kód pre napĺňanie údajov tabuľky “KritickeData”:

```
CREATE PROC proc_UlozenieKritickychDat
  @klient nvarchar(255),
  @hodnota varbinary(max),
  @posledna Zmena datetime,
  @hash binary(32) AS

IF (Exists(SELECT * FROM KritickeData WHERE klient = @klient))
 BEGIN
  UPDATE KritickeData SET
    Hodnota = @hodnota,
    PoslednaZmena = @posledna Zmena,
    HashToken = @hash
  WHERE
     klient = @klient
 END
ELSE
 BEGIN
  INSERT INTO KritickeData
      (Klient, Hodnota, PoslednaZmena, HashToken)
  VALUES (
    @klient,
    @hodnota,
    @posledna Zmena,
    @hash )
 END
GO

/*pridelenie prav pouzivatelovi aplikuser na spustanie ulozenej procedury*/
GRANT EXECUTE ON proc_UlozenieKritickychDat TO aplikuser;
GO
```

Jednoduchá uložená procedúra prijíma zašifrované dáta a ich hash z aplikácie a v prípade, že aplikácia vkladá nového klienta, vloží dáta do tabuľky “KritickeData” príkazom “INSERT”. Ak sa klient v tabuľke už nachádza, aktualizuje jeho údaje príkazom “UPDATE”.

19. Povrďte “Run” .
20. Zatvorte okno databázového správcu.

Cvičenie č.2: Vytvorenie hostovanej služby na Windows Azure

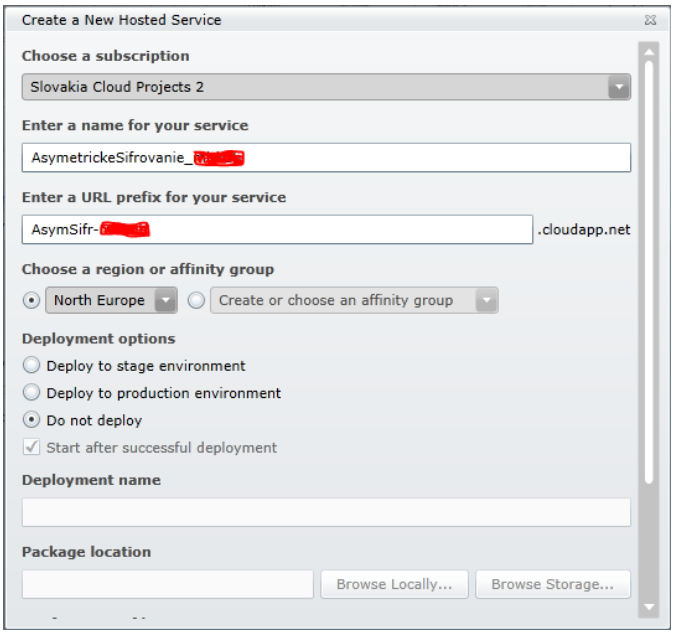
V tomto cvičení vytvoríte hostovanú službu na Windows Azure, do ktorej neskôr nasadíte jednoduchú webovú rolu.

Úloha č.1 – Vytvorenie "hosted service" na Windows Azure

1. Prepnite sa do okna portálu Windows Azure Platform a v ľavom dolnom menu portálu vyberte "**Hosted Services, Storage Accounts & CDN"**.
2. Zo zobrazenej ponuky horného ribbon menu vyberte "**New Hosted Service"**.
3. Vyberte Azure predplatné ("subscription"), ktoré chcete použiť pre aplikáciu.
4. Zadajte názov (napr. "AsymetrickeSifrovanie_*meno*") , ktorý bude identifikovať Azure služby používané aplikáciou v rámci Azure predplatného.
5. Zadajte URL vašej aplikácie. Windows Azure management portál skontroluje, či je URL adresa jedinečná v rámci dátových centier Windows Azure (a či ju už niekto nepoužíva).
6. Vyberte z ponuky región dátového centra, v ktorom chcete "hosted service" vytvoriť.

> **Poznámka:** Ak vyberiete identické dátové centrum, aké ste vybrali pri vytváraní servera SQL Azure, znížite tým latenciu sieťových odoziev a zároveň vám nebudú účtované poplatky za prenos dát medzi aplikáciou nasadenou v hostovanej službe a serverom SQL Azure.

7. Vyberte voľbu "**Do not deploy"**, aby sa v tomto kroku vytvorila iba deklarácia hostovanej služby (bez nasadenie aplikácie do nej).

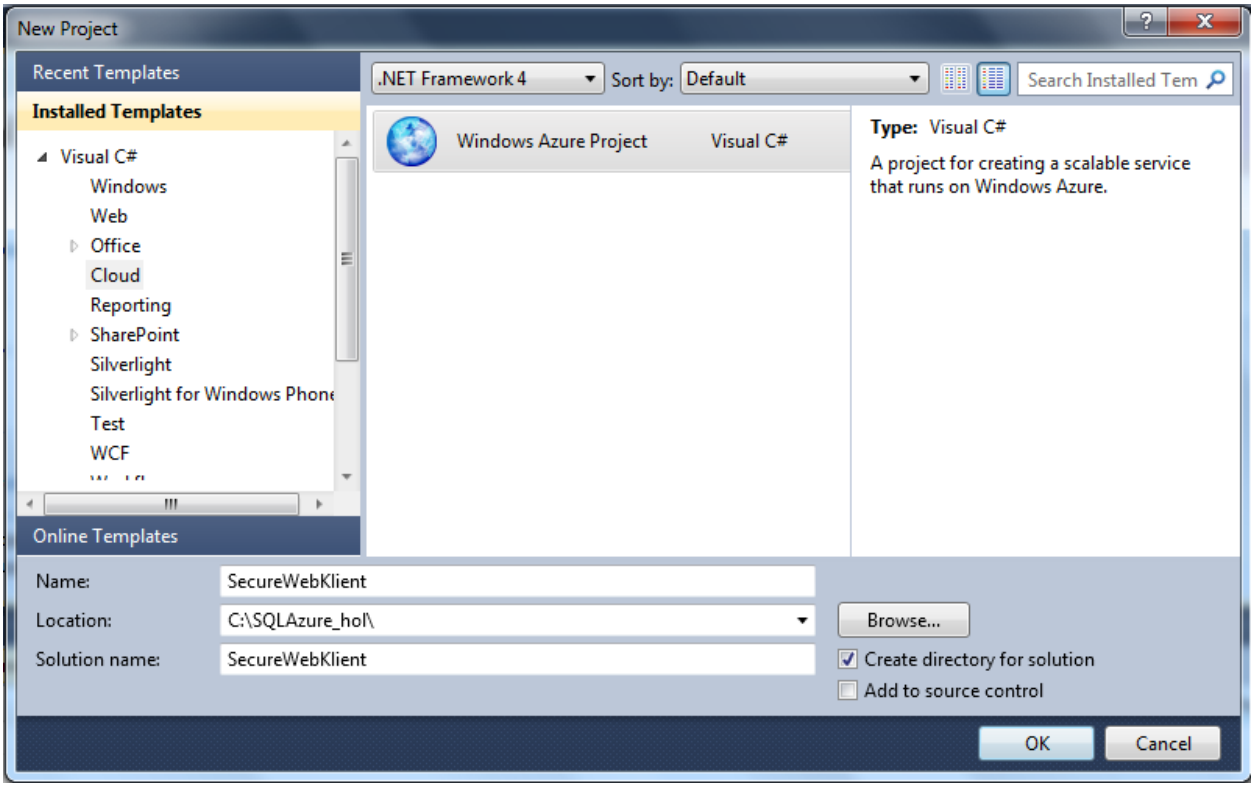


8. Potvrďte **OK**.

Úloha č.2 – Vytvorenie webovej aplikácie a certifikátu pre asymetrické šifrovanie

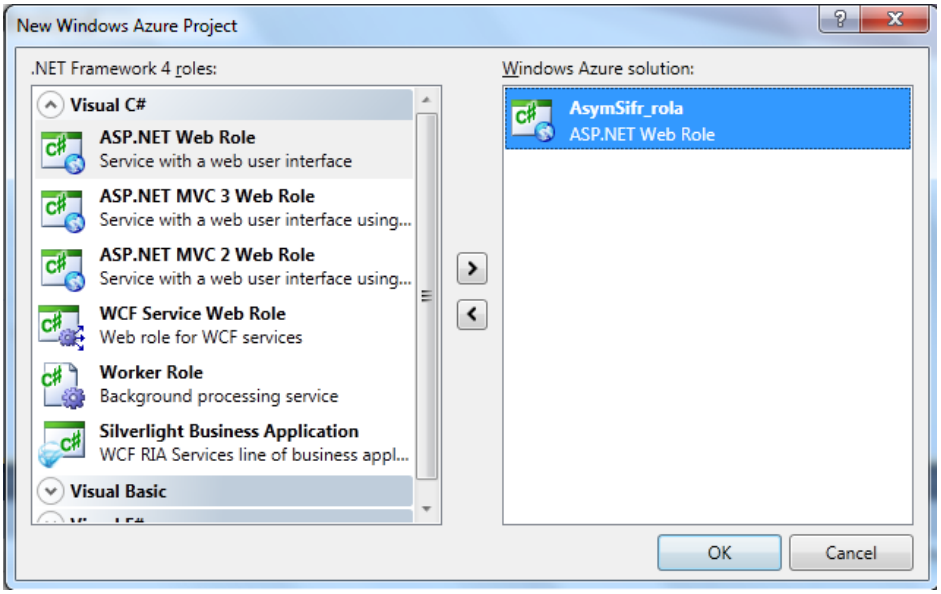
V tejto úlohe vytvoríte projekt webovej aplikácie a testovací certifikát pomocou Visual Studia .

1. Spustite Visual Studio 2010 v kontexte administrátora. Rozbaľte menu "**Start | All Programs | Microsoft Visual Studio 2010",** pravým tlačidlom myši nad "**Microsoft Visual Studio 2010"** zobrazte kontextové menu a vyberte z neho položku "**Run as administrator"**.
2. Cez menu "**File/New"** vyberte submenu **"Project"**.
3. V dialógovom okne "**New Project"** v paneli "**Installed Templates"** vyberte strom **"Visual_C#/Cloud"**.
4. Vyberte z ponuky šablónu „**Windows Azure Project"** a zadajte názov projektu "SecureWebKlient" (v položke "Name"):



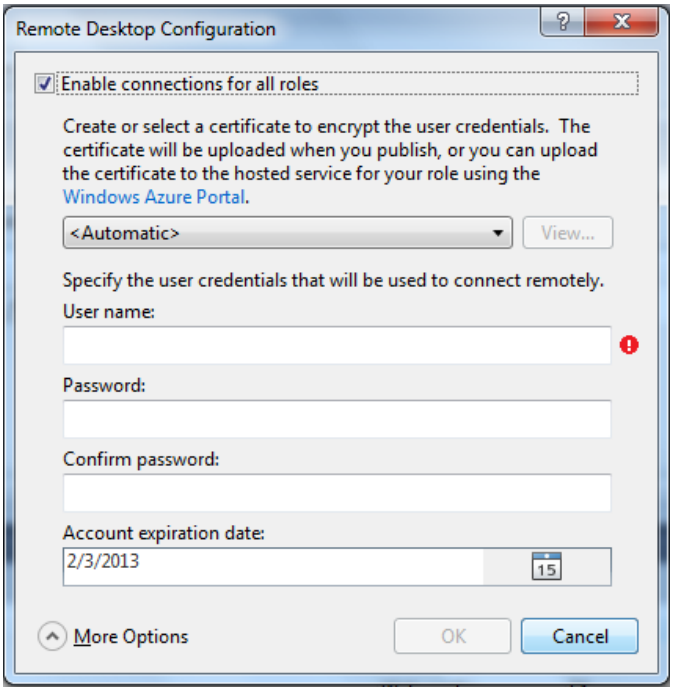
Potvrďte tlačidlo "**OK"**.

5. V dialógu "**New Windows Azure Project"** vyberte "**ASP.NET Web Role"** a kliknutím na symbol **">"** presuňte rolu medzi roly vybrané pre aktuálny projekt. Webová rola poskytuje prostredie pre prevádzku webových sajtov a aplikácií na infraštruktúre Internet Information Services 7.x.
6. Zobrazte pravým tlačidlom myši nad "WebRole1" v paneli "Windows Azure Solution" kontextové menu. Vyberte položku "Rename" a premenujte webovú rolu na "AsymSifr_rola".



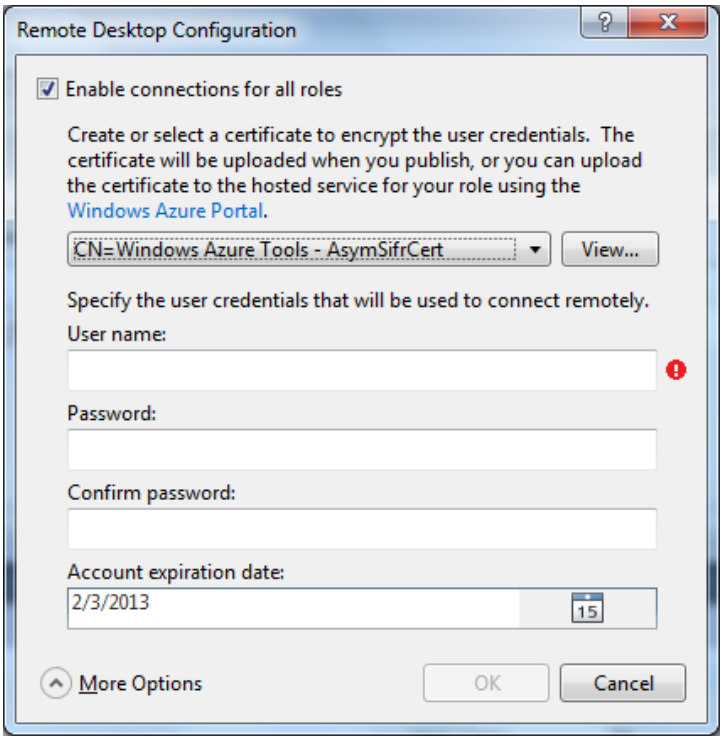
Potvrďte tlačidlo **OK**.

7. V okne “Solution Explorer” zobrazte pravým tlačidlom myši nad názvom projektu “SecureWebKlient” kontextové menu a vyberte z neho položku “Configure Remote Desktop”.
8. Vyberte položku „Enable Connections for all roles“ a kliknite na „dropdown“ položku s textom „<Automatic>”.



9. Vyberte poslednú položku zoznamu s názvom „<Create...>“.
10. V dialógu “**Create Certificate”** zadajte názov certifikátu pre evidenciu v Visual Studiu (napr. “AsymSifrCert”) a potvrďte tlačidlo “**OK”.**

11. V dialógu “Remote Desktop Configuration” potvrďte tlačidlo “View”.



12. V okne “Certificate” sa prepnite do záložky “Details”.
13. Vaša hostovaná služba Windows Azure musí obsahovať certifikát, aby ho pri vytváraní inštancií aplikácie v dátovom centre mohla nasadiť do každého virtuálneho servera. Potvrďte tlačidlo “**Copy to File**”.
14. V okne asistenta pre export certikátu potvrďte “Next”, vyberte export privátneho kľúča (voľba “Yes, export the private key”) a potvrďte “Next”.
15. Ponechajte vybranú voľbu “Personal Information Exchange – PKCS#12 (.PFX)“ a potvrďte „Next“.
16. Zadajte heslo pre export certifikátu a potvrďte „Next“.
17. Zadajte názov súboru, do ktorého chcete certifikát exportovať (napr.”AsymSifrCert”) a potvrďte “Next” a “Finish”.
18. Po úspešnom exporte certifikátu zatvorte okná s detailami o certifikáte a celý strom okien naviazaný na konfiguráciu „Remote Desktop Connection“.

Poznámka: Predchádzajúce kroky nám poslúžili na to, aby sme vytvorili „self-signed“ certifikát, ktorý pracuje s providerom typu CSP (**Cryptographic Service Provider**). Ak by ste použili na vytvorenie „self-signed“ certifikátu v Visual Studiu „Publish Configuration Wizard“, získate certifikát, ktorý pracuje s providerom typu KSP (Key Storage Provider), ktorý nie je možné v Windows Azure bez väčších úprav použiť na dešifrovanie algoritmom RSA.

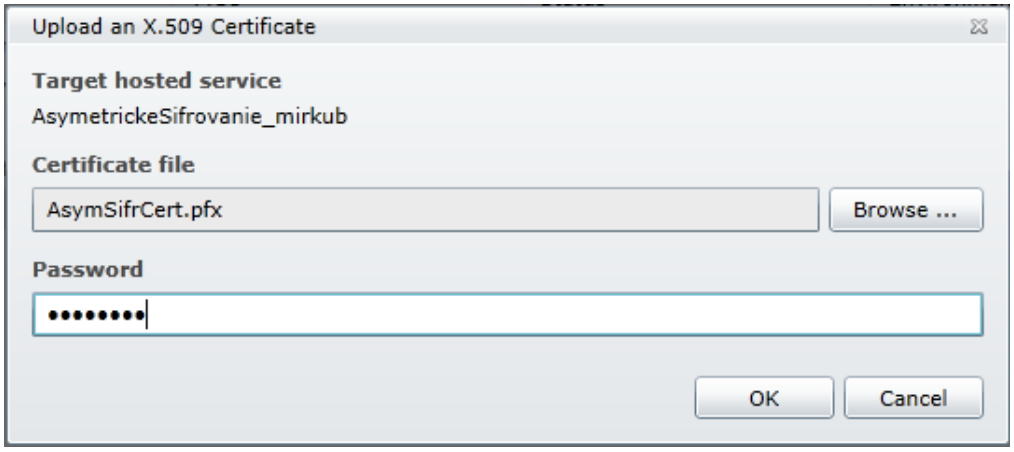
Úloha č.3 – Umiestnenie testovacieho certifikátu do hostovanej služby na Windows Azure

V tejto úlohe nasadíte testovací certifikát do hostovanej služby na Windows Azure. Tento postup platí aj pre import SSL certifikátu do hostovanej služby pre nutnosť šifrovania komunikácie protokolom “https”.

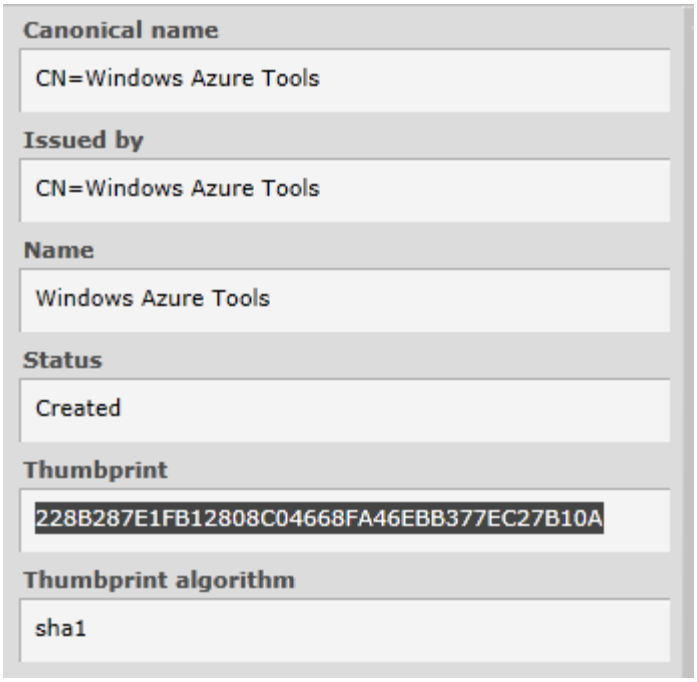
1. Prepnite sa na **Windows Azure Management Portal**.
2. V ľavom hornom menu vyberte “**Hosted Services**”.
3. V hlavnom panele správcovského portálu sa zobrazí zoznam aktívnych hostovaných služieb. Rozbaľte strom služby “AsymetrickeSifrovanie_*nazov*” a vyberte z neho položku “Certficates”.

4. Z ribbon menu vyberte “Add certificate”.

5. Po stlačení tlačidla “Browse” vyhľadajte súbor “AsymSifrCert .pfx” vytvorený v predošlej úlohe, zadajte heslo, ktoré ste použili na export certifikátu a potvrďte tlačidlo “OK”.



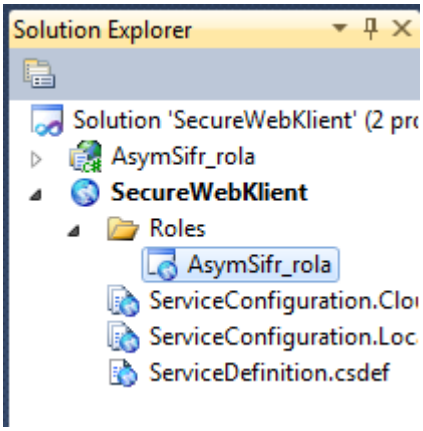
6. Do stromu hostovanej služby sa pridá nový certifikát. Všimnite si, že Visual Studio použilo pri generovaní certifikátu názov “Windows Azure Tools”. Dôležitou informáciou, na základe ktorej bude vedieť naša aplikácia spárovať aplikačný kód s certifikátom uloženým v dátovom centre, je “thumbprint”:



7. Uložte hodnotu thumbprintu do schránky (clipboard-u).

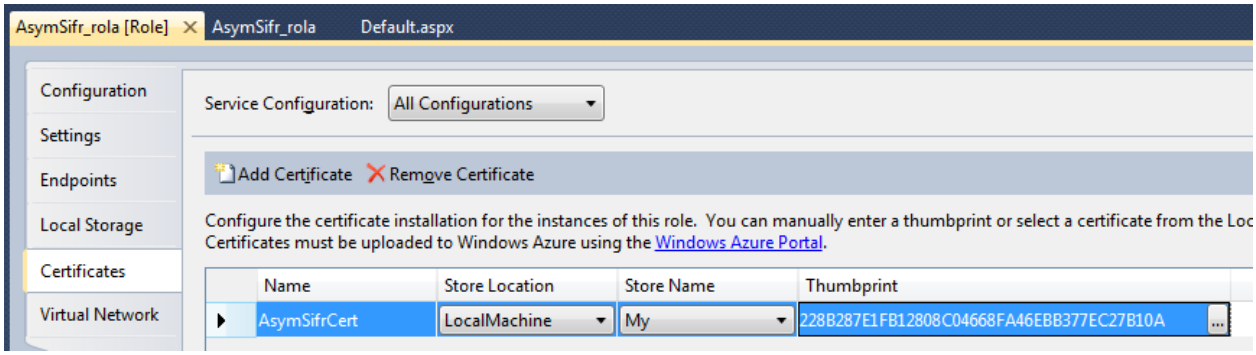
Úloha č.4 – Umiestnenie referencie na šifrovací certifikát (thumbprintu) do konfigurácie webovej role

1. Prepnite sa do okna **Visual Studio**.
2. V okne “Solution Explorer” v projekte “SecureWebKlient” zobrazte dvojklikom myši nad “AsymSifr_rola” nastavenia webovej role Windows Azure:



3. Z panela nastavení vyberte záložku “Certificates” a potvrďte tlačidlo “Add Certificate”.
4. Zadajte do položky “Name” názov certifikátu “AsymSifrCert”, na ktorý sa budete odkazovať v aplikačnom kóde.

5. Nastavenie úložiska certifikátu, “Store Location”, ponechajte na hodnote “LocalMachine”, “Store Name” na hodnote “My”. Do položky “Thumbprint” vložte zo schránky (clipboard-u) hodnotu “thumbprint”, ktorú ste načítali z Azure Management portálu.



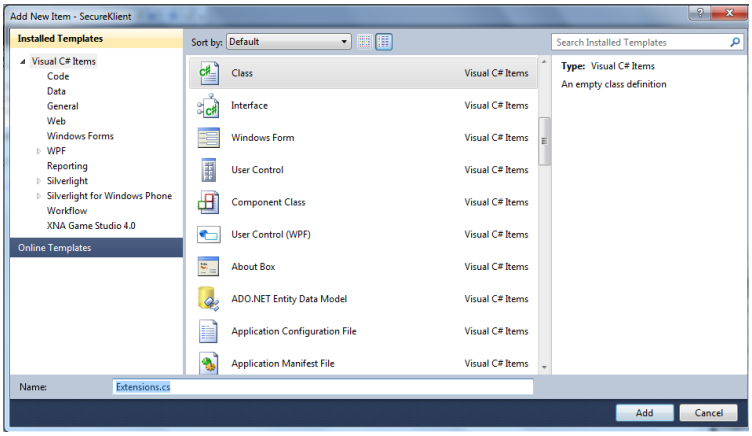
6. Uložte vytvorené nastavenie.

Cvičenie č.3: Rozšírenie webovej aplikácie o triedy asymetrického šifrovania

V tomto cvičení pridáte do webovej aplikácie aplikačnú logiku, v ktorej asymetricky zašifrujete dáta a následne ich uložíte do tabuľky na servri SQL Azure.

Úloha č.1 – Vytvorenie triedy na šifrovanie/dešifrovanie

1. Implementácie šifrovacích a hashovacích algoritmov narábajú s poľami bajtov. Preto skôr ako sa pustíme do kódovania zabezpečenia dát, predpripravíme si triedu, ktorá sprehľadní kód a postará sa o konverziu typov „string“ a „DateTime“ na pole bajtov. V okne „**Solution Explorer**“ vyberte pravým tlačidlo myši nad názvom projektu „AsymSifr_rola“ kontextové menu „**Add/New Item**“. Vyberte šablónu „Class“, do položky „Name“ vložte názov novej triedy „Extensions.cs“ a potvrďte tlačidlo „Add“.



2. Pridajte do zoznamu referencovaných menných priestorov

```
using System.Text;
```

3. Rozšírte deklaráciu triedy “Extensions” na

```
public static class Extensions
```

, pretože pridávanie rozširujúcich metód dátových typov vyžaduje “static” triedu.

4. Vložte do triedy „Extensions“ dve implementácie metódy „GetBytes“ pre dátové typy „string“ a „DateTime“, ktoré zabezpečia konverziu týchto dátových typov na pole bajtov. Výsledný kód triedy „Extensions“ bude nasledujúci:

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

namespace AsymSifr_rola
{
   public static class Extensions
   {
      //metoda GetBytes() pre datovy typ string
      public static byte[] GetBytes(this string value)
      {
         byte[] buffer = UTF8Encoding.UTF8.GetBytes(value);
         return buffer;
      }
      //metoda GetBytes() pre datovy typ DateTime
      public static byte[] GetBytes(this DateTime value)
      {
         return value.ToString().GetBytes();
      }
   }
}
```

Úloha č.2 – Vytvorenie triedy na šifrovanie/dešifrovanie

1. Vytvorme triedu, v ktorej si pripravíme “helper-y” pre šifrovanie a dešifrovanie asymetrickým algoritmom RSA tak, aby sme ich mohli použiť aj v iných typoch aplikácií.

V okne „**Solution Explorer**“ vyberte pravým tlačidlo myši nad názvom projektu „AsymSifr_rola“ z kontextového menu položku „Add/New Item“. Vyberte šablónu „Class“, do položky „Name“ vložte názov novej triedy „Sifrovanie.cs“ a potvrďte tlačidlo „Add“.

2. Do otvoreného okna s kódom súboru „Sifrovanie.cs“ vložte hneď za príkaz „using System.Web;“ príkazy na referencovanie hierarchie tried na šifrovanie, kódovanie textových hodnôt a vstupno/výstupné operácie.

```
//referencie potrebne pre sifrovanie a encoding
using System.Text;
using System.IO;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
```

3. Pre asymetrické šifrovanie certifikátom musíme aplikačnému kódu povedať, ktorý certifikát nasadený do inštancie webovej role (virtuálneho servera) má použiť. Referenciou je “thumbprint” certifikátu, ktorý ste použili pri doplnení konfigurácie webovej role. Vložte do vnútra triedy “Sifrovanie” deklaráciu pre “_THUMBPRINT_”:

```
  public class Sifrovanie
  {
    //thumbprint certifikatu pouziteho na sifrovanie
    private string _THUMBPRINT_ = " 228B287E1FB12808C04668FA46EBB377EC27B10A";

…
```

Poznámka: Pre jednoduchosť sme “thumbprint” vložili priamo do aplikačného kódu. Ak by sme chceli “thumbprint” načítať z konfigurácie webovej roly (t.j. súboru “csfcg”), ktorý je možné editovať aj na Windows Azure Management portále počas spustenia aplikácie, môžeme použiť metódu “RoleEnvironment.GetConfigurationSettingValue”.

4. Vytvorme v triede “Sifrovanie” metódu, ktorá zašifruje použitím certifikátu RSA algoritmom vstupný textový argument a vráti zašifrované pole bajtov. Vložte do vnútra triedy “Sifrovanie” definíciu metódy “ZasifrujCertifikatom”:

```
    .....
    //thumbprint certifikatu pouziteho na sifrovanie
    private string _THUMBPRINT_ = " 228B287E1FB12808C04668FA46EBB377EC27B10A";


    public byte[] ZasifrujCertifikatom(string hodnota)
    {
      byte[] buffer = UTF8Encoding.UTF8.GetBytes(hodnota);
      //nacitanie uloziska certifikatov
```

```
        X509Store store = new X509Store(StoreName.My,
                                StoreLocation.LocalMachine);
        store.Open(OpenFlags.ReadOnly);
        //vyhladanie certifikatu podla thumbprintu
        X509Certificate2 x509 = store.Certificates.Find(
                        X509FindType.FindByThumbprint,
                        _THUMBPRINT_, false)[0];
        store.Close();
        // RSA sifrovanie
        RSACryptoServiceProvider rsaSifrovanie = null;
        rsaSifrovanie = (RSACryptoServiceProvider)x509.PublicKey.Key;
        byte[] zasifrovaneBajty = rsaSifrovanie.Encrypt(buffer, false);
        return zasifrovaneBajty;
    }
    ....
```

5. Vytvorme v triede “Sifrovanie” metódu, ktorá dešifruje obsah zašifrovaný RSA algoritmom na základe vstupného poľa bajtov. Vložte do vnútra triedy “Sifrovanie” definíciu metódy “DesifrujCertifikatom”:

```
...
        byte[] zasifrovaneBajty = rsaSifrovanie.Encrypt(buffer, false);
        return zasifrovaneBajty;
    }

    public string DesifrujCertifikatom(byte[] zt)
    {
        if (zt == null)
            return "Chyba";
        //nacitanie uloziska certifikatov
        X509Store store = new X509Store(StoreName.My,
                                StoreLocation.LocalMachine);
        store.Open(OpenFlags.ReadOnly);
        //vyhladanie certifikatu podla thumbprintu
        X509Certificate2 x509 = store.Certificates.Find(
                        X509FindType.FindByThumbprint,
                        _THUMBPRINT_, false)[0];
        store.Close();
        // RSA desifrovanie
        RSACryptoServiceProvider rsaSifrovanie = null;
        rsaSifrovanie = (RSACryptoServiceProvider)x509.PrivateKey;
        byte[] desifrovaneBajty = rsaSifrovanie.Decrypt(zt, false);
        return UTF8Encoding.UTF8.GetString(desifrovaneBajty);
    }
...
```

6. Ďalším stupňom zabezpečenia dát uložených v perzistentnom úložisku (, akým je napr. SQL Azure), je detekcia zmien hodnôt priamo na databázovom servri obídením aplikačnej vrstvy. K tomu slúži generovanie a ukladanie hash reťazcov odvodených z ukladaných hodnôt spôsobom, ktorý je známy iba aplikačnej vrstve. Vytvorme v triede "Sifrovanie" metódu, ktorá vytvorí hash na základe vstupného poľa bajtov:

```
        ....
          byte[] desifrovaneBajty = rsaSifrovanie.Decrypt(zt, false);
          return UTF8Encoding.UTF8.GetString(desifrovaneBajty);
        }

        public static byte[] VypocitajHash(params byte[][] bajty)
           {
             SHA256 sha = SHA256Managed.Create();
             MemoryStream ms = new MemoryStream();
                    for (int i = 0; i < bajty.Length; i++)
                      ms.Write(bajty[i], 0, bajty[i].Length);
             ms.Flush();
             ms.Position = 0;
             return sha.ComputeHash(ms);
            }
...
```

Poznámka: Hash algoritmus sme použili pre jednoduchosť v základnej forme. Pre lepšiu ochranu hashovania sa doporučuje pridávať k hashovanému reťazcu aj „salt" reťazec.

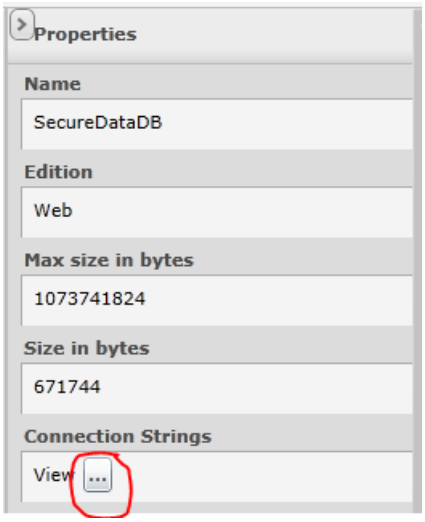
7. Uložte zmeny v kóde.

Úloha č.3 – Vytvorenie triedy na zápis do databázy SQL Azure

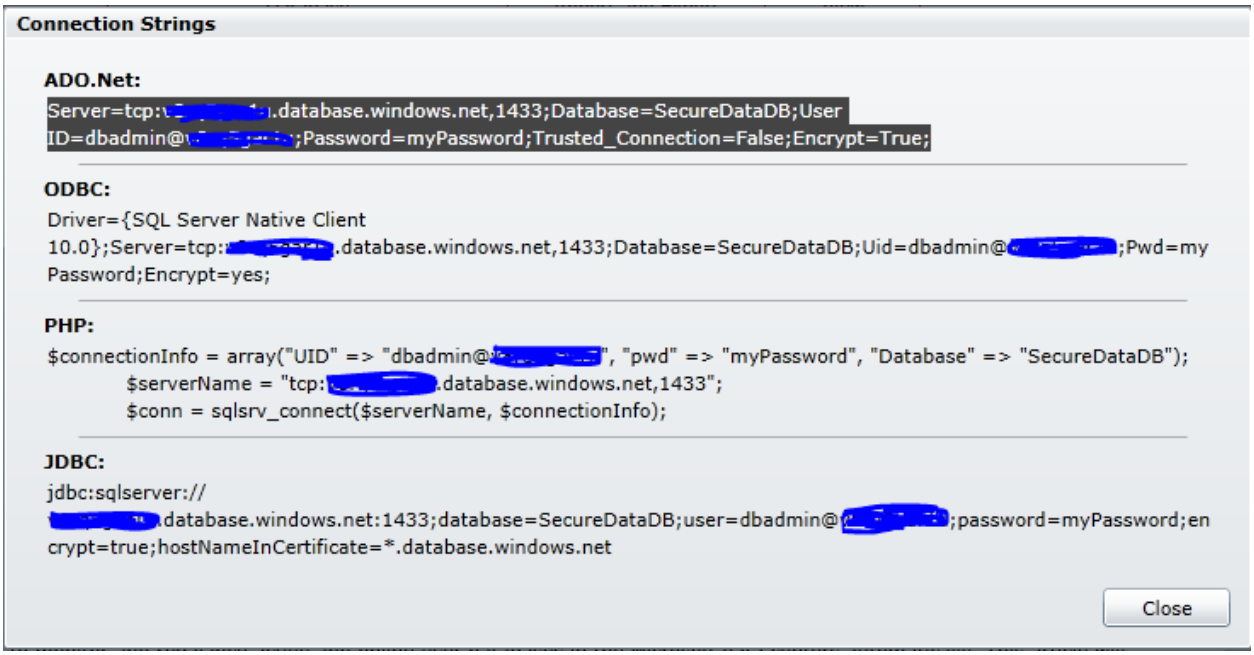
1. Dáta zašifrované na strane klienta budeme ukladať do databázy SQL Azure vytvorenej v prvom cvičení. Preto do konfiguračného súboru uložíme potrebný „connection string".

   Prepnite sa do okna portálu **Windows Azure Platform** a v ľavom hornom menu rozbaľte uzol "**Subscriptions"** až do úrovne názvu databázy. Vyberte z tohto menu databázu "SecureDataDB".

2. V pravom paneli portálu kliknite v sekcii "Connection Strings" na ikonu "…" vpravo od popisky "View".

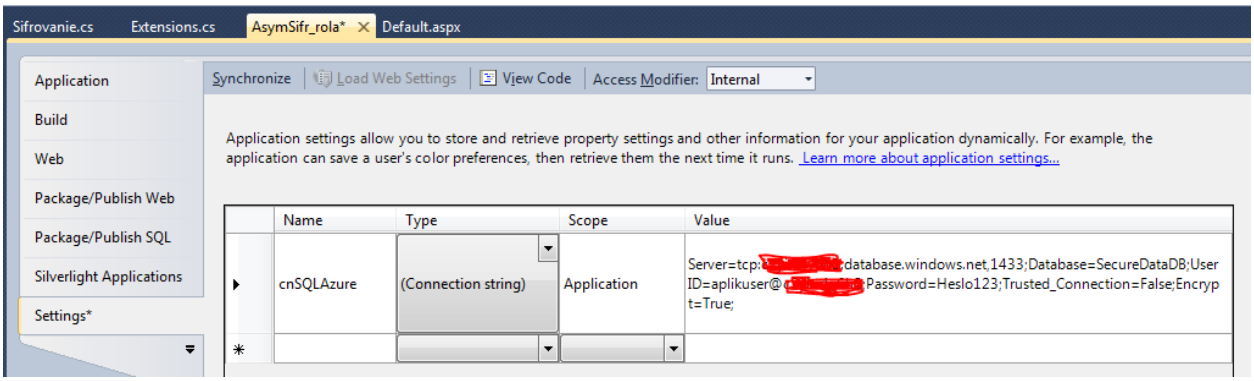


3. Skopírujte si schránky reťazec nachádzajúci sa pod “Ado.NET:”



4. Zatvorte okno tlačidlom “**Close**”.
5. Prepnite sa do okna **Visual Studio** s rozpracovaným oknom webovej aplikácie.
6. V okne „**Solution Explorer**“ vyberte pravým tlačidlo myši nad názvom projektu „AsymSifr_rola“ z kontextového menu položku „Properties“.
7. Vyberte záložku „**Settings**“.
8. Prepíšte predpripravený text v stĺpci „Name“ na „cnSQLAzure“. Vyberte v stĺpci „Type“ hodnotu „(Connection String)“ a vložte do posledného stĺpca „Value“ reťazec pripojenia na databázu SecureDataDB uložený v schránke (clipboard-e).
9. Reťazec pripojenia na databázu je predpripravený na pripájanie sa v kontexte správcu servera SQL Azure. Pre aplikácie sme vytvorili SQL účet „aplikuser“, ktorý má výrazne nižšie práva. Prepíšte teda v „connection string-u“ hodnotu parametra „UserID“ z „dbadmin@.....“ na

„aplikuser@.....“ . Upravte hodnotu parametra „Password“ podľa zadaného hesla pre SQL účet „aplikuser“.



Poznámka: SQL Azure podporuje v súčasnosti iba autentizáciu “SQL Authentication”, teda meno/heslo. Heslo nemusí byť uložené v čitateľnej podobe v “connection string-u”. Môžete ho na klientskej strane generovať podľa Vami zvoleného algoritmu, ukladať do konfiguračného súboru zašifrované aplikačným kľúčom atď. Komunikácia medzi klientom a SQL Azure server prebieha v SSL šifrovanom toku dát.

10. Uložte vytvorený konfiguračný súbor.
11. Vytvorme triedu, v ktorej si pripravíme “helper-y” pre zápis a čítanie dát z databázy SQL Azure. V okne „**Solution Explorer**“ vyberte pravým tlačidlo myši nad názvom projektu „AsymSifr_rola“ z kontextového menu položku „Add/New Item“. Vyberte šablónu „**Class**“, do položky „Name“ vložte názov novej triedy „KritickeData.cs“ a potvrďte tlačidlo „Add“.
12. Do otvoreného okna s kódom súboru „KritickeData.cs“ vložte hneď za príkaz „using System.Web;” príkazy na referencovanie hierarchie tried na prácu s dátami v ADO.NET, kódovanie textu a prácu s konfiguráciou aplikácie:

```
using System.Text;
using System.Data;
using System.Data.SqlTypes;
using System.Data.SqlClient;
using System.Configuration;
```

13. Pre ukladanie dát na SQL Azure použijeme jednoduchý koncept volania uloženej procedúry po zašifrovaní dát. Vytvorte v triede “KritickeData” metódu, ktorá v inštancii SQLCommand parametrizovane zavolá uloženú procedúru v databáze SQL Azure:

```
class KritickeData
{
    public static void Ulozit(string Klient, byte[] zt)
    {
        //vytvorenie spojenia na zaklade connection stringu ulozeneho v konfig.subore
        using (SqlConnection sqlConn =
```

```
            new SqlConnection( AsymSifr_rola.Properties.Settings.Default.cnSQLAzure ))
        {
            sqlConn.Open();
            using (SqlCommand sqlCmd = new SqlCommand())
            {
                //zavolanie ulozenej procedury na ulozenie dat do databazy
                DateTime datumAktualizacie = DateTime.Now;
                sqlCmd.Connection = sqlConn;
                sqlCmd.CommandType = System.Data.CommandType.StoredProcedure;
                sqlCmd.CommandText = "proc_UlozenieKritickychDat";
                sqlCmd.Parameters.Add("klient", SqlDbType.NVarChar, 255);
                sqlCmd.Parameters.Add("hodnota", SqlDbType.VarBinary, int.MaxValue);
                sqlCmd.Parameters.Add("poslednaZmena", SqlDbType.DateTime);
                sqlCmd.Parameters.Add("hash", SqlDbType.VarBinary, 32);
                sqlCmd.Parameters[0].Value = Klient;
                sqlCmd.Parameters[1].Value = zt;
                sqlCmd.Parameters[2].Value = datumAktualizacie;
                // kalkulacia hashu zaznamu. Zmena kritickych hodnot = zmena hashu
                byte[] hash = Sifrovanie.VypocitajHash(
                    Klient.GetBytes(),
                    zt,
                    datumAktualizacie.GetBytes());
                sqlCmd.Parameters[3].Value = hash;
                int res = sqlCmd.ExecuteNonQuery();
            }
            sqlConn.Close();
        }
    }
...
```

Poznámka: Hash reťazec sme pre jednoduchosť uložili do tej istej tabuľky, do ktorej ukladáme zašifrovaný obsah. Pre zvýšenie bezpečnosti a hodnoty hash odtlačku sa doporučuje ukladať hash reťazec mimo dát, z ktorých bol generovaný.

14. Pre načítanie dát z databázy SQL Azure a následné dešifrovanie použijeme parametrizovaný "SELECT" príkaz. Vytvorte preto v triede "KritickeData" metódu "Nacitat":

```
...
            sqlConn.Close();
        }
    }
    public static byte[] Nacitat(string klient)
```

```
    {
      //vytvorenie spojenia na zaklade connection stringu ulozenom konfig.subore
      byte[] zasifrText = null;
      using (SqlConnection sqlConn =
          new SqlConnection(AsymSifr_rola.Properties.Settings.Default.cnSQLAzure))
      {
        sqlConn.Open();
        using (SqlCommand sqlCmd = new SqlCommand())
        {
          //vyber dat pre vybraneho klienta
          sqlCmd.Connection = sqlConn;
          sqlCmd.CommandType = System.Data.CommandType.Text;
          sqlCmd.CommandText = "SELECT * from KritickeData where klient=@klient";
          sqlCmd.Parameters.Add("@klient", SqlDbType.NVarChar, 255);
          sqlCmd.Parameters[0].Value = klient.Trim();
          SqlDataReader dr = sqlCmd.ExecuteReader();
          try
          {
            dr.Read();
            zasifrText = (byte[])dr.GetValue(2);
            DateTime posledna Zmena_v_databaze = (DateTime)dr.GetValue(3);
            byte[] hash_v_databaze = (byte[])dr.GetValue(4);
            byte[] hash = Sifrovanie.VypocitajHash(
                  klient.GetBytes(),
                  zasifrText ,
                  posledna Zmena_v_databaze.GetBytes());
            //kontrola, ci vypocitany hash a hash z databazy sedia
            if (hash.SequenceEqual<byte>(hash_v_databaze) == false)
            {
              zasifrText = null;
            }
          }
          catch (Exception e)
          {
            HttpContext.Current.Response.Write(e.Message.ToString());
          };
        }
        sqlConn.Close();
      }
      return zasifrText;
  }
...
```

15. Uložte zmeny v kóde.

Úloha č.4 – Kompletizácia webovej aplikácie

Projekt webovej aplikácie po dokončení predchádzajúcich úloh obsahuje všetky potrebné triedy na kompletizáciu kódu v prednastavenej stránke “default.aspx”.

1. Otvorte stránku “**default.aspx**” v projekte “AsymSifr_rola”.
2. Nahraďte vnútro bloku “<asp:Content ID="BodyContent"...>…</ asp:Content>” kódom, v ktorom deklarujete tlačidlá na uloženie a načítanie zašifrovanej hodnoty:

```
    <br />
    <asp:Label ID="Label1" runat="server"
        Text="Zadajte text na zašifrovanie a uloženie: "></asp:Label>
    <asp:TextBox ID="txtZasifrovat" runat="server" MaxLength="50"
Width="288px"></asp:TextBox>
    <br />
    <br />
    <asp:Button ID="btnZasifruj" runat="server" onclick="Button1_Click"
        Text="Zašifrovať a uložiť na SQL Azure" />
    <br />
    <br />
    ----------------------------------------------------------------------------------------------------<br />
    <asp:Button ID="btnDesifruj" runat="server"
        Text="Načítať z SQL Azure a dešifrovať" /> <br /> <br /> 
    <asp:Label ID="lblDesifrovanyText" runat="server" Text=""></asp:Label>
    <br />
```

3. Prepnite sa do módu dizajnovania stránky (záložkou “**Design”**).
4. Dvojklikom na tlačidlo “btnZasifruj” vytvorte “handler” metódu pre spracovanie udalosti kliknutia.
5. Vložte do tela metódy “btnZasifruj_Click” kód na šifrovanie a uloženie textu zadaného do položky “txtZasifrovat”:

```
        Sifrovanie objSifrovanie = new Sifrovanie();
        byte[] zt = objSifrovanie.ZasifrujCertifikatom (txtZasifrovat.Text.Trim());
        //ulozenie zasifrovanych dat pre zadaneho klienta na SQL Azure
        KritickeData.Ulozit("Peter Majetný", zt);
```

6. Dvojklikom na tlačidlo “btnDesifruj” vytvorte “handler” metódu pre spracovanie udalosti kliknutia na druhom tlačidle.
7. Vložte do tela metódy “btnDesifruj_Click” kód na načítanie zašifrovaného textu z databázy, jeho dešifrovanie a zobrazenie v „label“ prvku „lblDesifrovanyText“:

```
        Sifrovanie objSifrovanie = new Sifrovanie();
        //nacitanie dat z databazy pre zadaneho klienta
        byte[] zt = KritickeData.Nacitat("Peter Majetný");
        //kontrola, ci pole nie je prazdne (nenacitane, alebo "vycistene" pre neplatny hash)
```

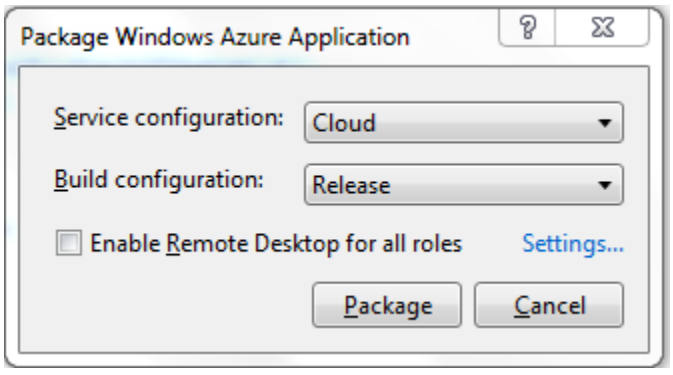
```
if ((object)zt != null)
{
   //desifrovanie pola bajtov a zobrazenie
   string desifrovane_data = objSifrovanie.DesifrujCertifikatom (zt);
   lblDesifrovanyText.Text = desifrovane_data;
}
else
{
  lblDesifrovanyText.Text = "Chyba pripojenia k databáze alebo neplatný hash kód.";
}
```

8. Uložte zmeny v kóde.

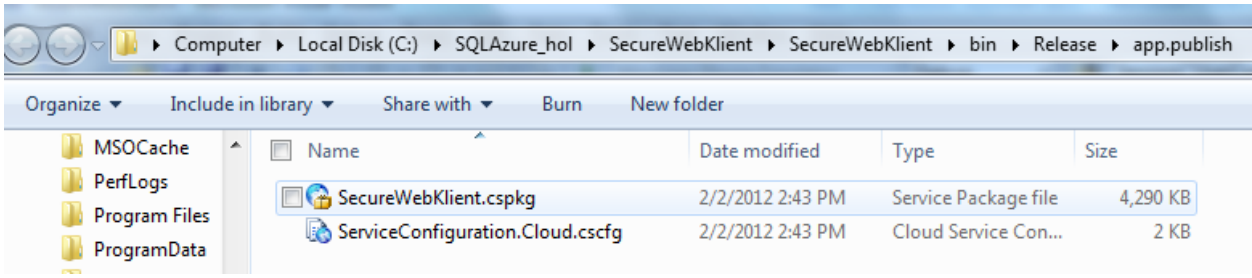
Úloha č.4 – Nasadenie riešenia do dátového centra Windows Azure

Aplikáciu je potrebné zbaliť do distribučného balíčka tak, aby dátové centrum po jeho prevzatí a rozbalení malo k dispozícii skompilovanú aplikáciu a konfiguračný predpis prevádzky v dátovom centre.

1. V paneli “**Solution Explorer**” pravým tlačidlom myši nad názvom projektu “SecureWebKlient”
2. Zobrazte kontextové menu a vyberte z neho položku “**Package**”.
3. V dialógu “**Package Windows Azure Application**” ponechajte prednastavené hodnoty a potvrďte tlačidlo “Package”:

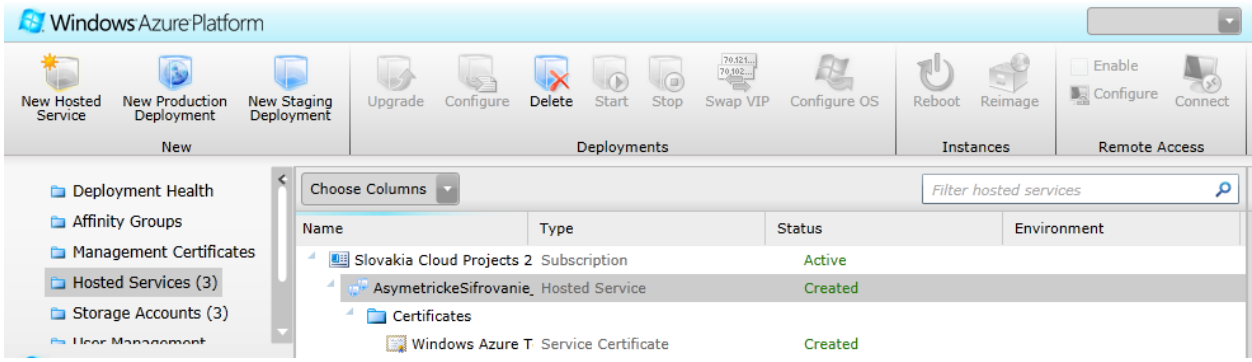


4. Visual Studio vytvorí inštalačný balíček a konfiguračný súbor a nakoniec otvorí priečinok na disku, kde sú potrebné súbory vygenerované.

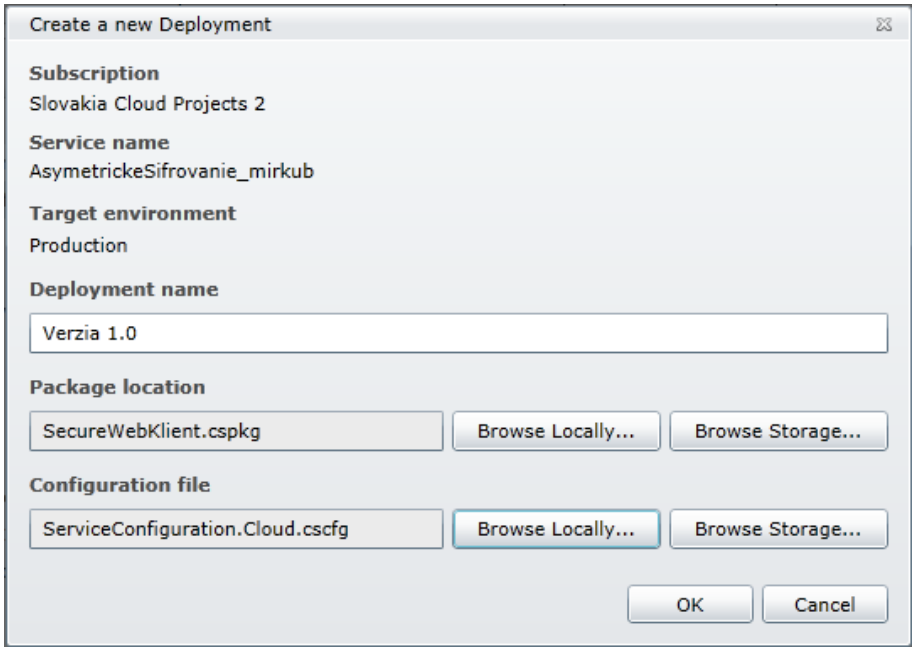


Súbor "ServiceConfiguration.Cloud.cscfg" obsahuje konfiguračný predpis webovej role, súbor "SecureWebKlient.cspkg" obsahuje definíciu služby, binárne súbory aplikácie a ďalšie položky nasadzované ako súčasť aplikácie. Zaznačte si cestu k súborom "cscfg" a "cspkg", budete ju potrebovať pri nasadzovaní aplikácie cez Windows Azure management portál.

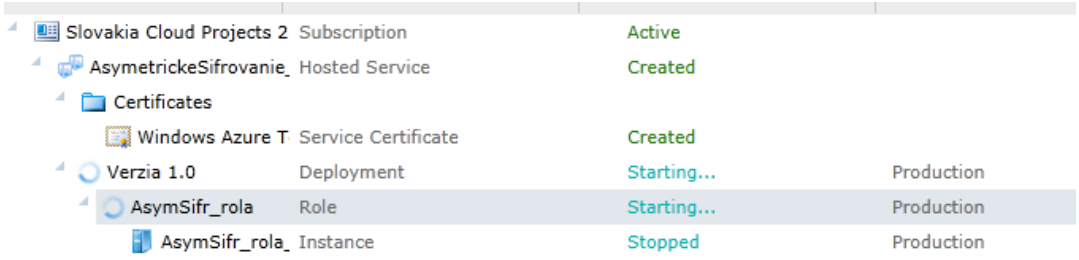
5. Prepnite sa do okna **Windows Azure Management Portal**.
6. V ľavom dolnom menu portálu vyberte "**Hosted Services, Storage Accounts & CDN**".
7. Z ľavého horného menu vyberte "**Hosted Service**".
8. V hlavnom paneli vyberte hostovanú službu, ktorú ste pripravili v predchádzajúcom cvičení.



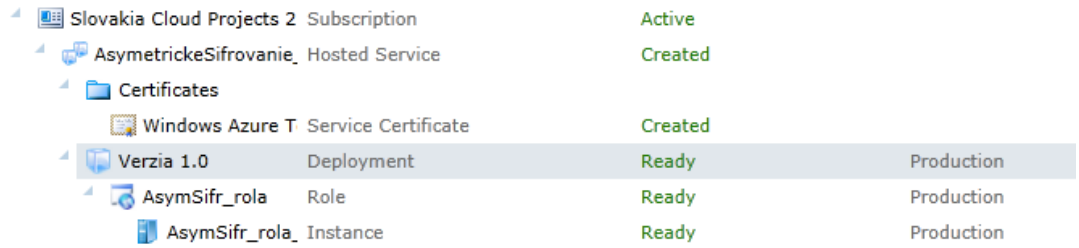
9. Vyberte z horného "ribbon" menu položku "**New Production Deployment**".
10. Zadajte názov nasadenia , aby ste neskôr ľahšie identifikovali, aká verzia aplikácie beží v cloude.
11. Pri položke "Package location" kliknite na zodpovedajúce tlačidlo "Browse Locally…" , prejdite do priečinka na lokálnom disku, v ktorom máte súbor "SecureWebKlient.cspkg" a vyberte ho.
12. Pri položke "Configuration file" kliknite na zodpovedajúce tlačidlo "Browse Locally…" , prejdite do priečinka na lokálnom disku, v ktorom máte súbor "ServiceConfiguration.Cloud.cscfg" a vyberte ho.



13. Potvrďte “OK”. Zobrazí sa upozornenie, že aplikáciu chcete spustiť iba v jednej inštancii (virtuálnom počítači), tak ako je prednastavené v definované v súbore “ServiceConfiguration.cscfg”. Pre účely jednoduchej ukážky to nie je problém, pri prevádzke produkčnej aplikácie by vzniklo riziko pri zaručení dostupnosti aplikácie 24x7. Potvrďte tlačidlo “**Yes**”.
14. Stav procesu nasadenia môžete monitorovať na Windows Azure management portále v sekcii Hosted Services .



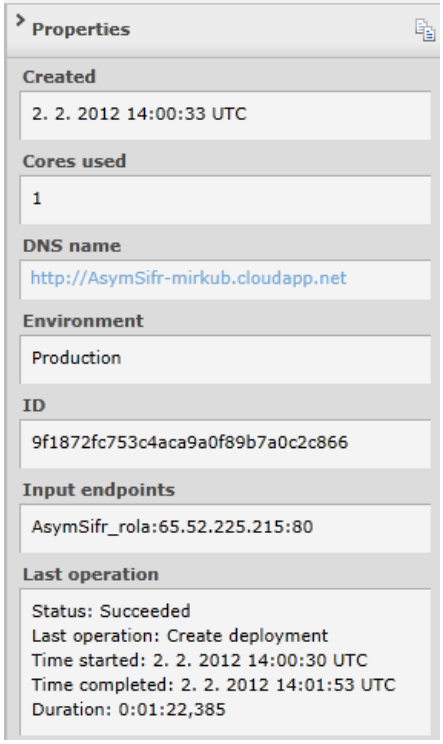
15. Keď sa proces nasadenia dostane do stavu “Ready”, môžete aplikáciu spustiť z dátového centra zadaním DNS mena (URL adresy) do internetového prehliadača, alebo priamo “kliknúť” na adresu v položke “DNS name” v pravom paneli portálu.



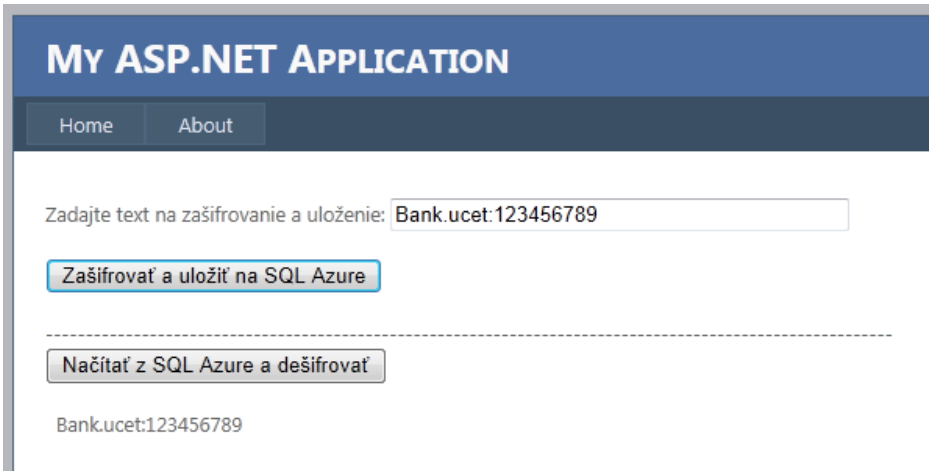
Poznámka: Aplikáciu ste sprevádzkovali v “production” prostredí a je dostupná cez URL adresu. Podobným postupom by ste ju mohli nasadiť do testovacieho prostredia a po otestovaní jednoducho presunúť do produkčného prostredia. Na Windows Azure Management portále si všimnite, že po nasadení balíčka môžete cez horné menu zastaviť a naštartovať aplikačné služby, odstrániť a “upgradovať nasadenie aplikácie v dátovom centre.

Upozornenie: Ak ste aplikáciu nasadili do dátového centra Windows Azure, vždy berte na zreteľ, že nasadená aplikácia, či už je spustená alebo nie, je spoplatňovaná v rámci vášho Azure predplatného. Preto je dôležité, aby ste nepotrebné nasadenia aplikácie vždy z portálu odstránili. Prejdete do sekcie “Hosted Service”, vyberiete nepotrebné nasadenie, zastavíte jeho služby cez ikonu “Stop” v hornej nástrojovej lište a potom ho odstránite cez ikonu “Delete”. Ak nepotrebné nasadenie neodstránite, alokuje definovaný počet virtuálnych serverov (aj keď je beh aplikácie zastavený), ktoré sa spoplatňujú podľa stanoveného predplatného Windows Azure.

16. Zo stromu hostovanej služby vyberte riadok, v ktorom je typ “**Deployment**”.
17. V pravom paneli “Properties” sa zobrazia detailné informácie o nasadenej konfigurácii.



18. Potvrďte URL linku aplikácie v položke „DNS name“.
19. Zobrazí sa okno webovej aplikácie prevádzkovanej v cloude.
20. Do textovej položky zadajte text, ktorý chcete zašifrovať a potvrďte tlačidlo “Zašifrovať a uložiť na SQL Azure”. Potvrďte tlačidlo “Načítať z SQL Azure a dešifrovať”. Zobrazí sa text, ktorý bol načítaný z SQL Azure a následne dešifrovaný v inštancii webovej roly.



Úloha č.5 – Overenie uloženia šifrovaných dát na SQL Azure

1. Prepnite sa do okna portálu Windows Azure Management a v ľavom hornom menu rozbaľte uzol “**Subscriptions”** až do úrovne názvu databázy. Vyberte z tohto menu databázu “SecureDataDB”.
2. V ribbon menu vyberte zo sekcie “Database” položku “**Manage**”.
3. Budete presmerovaný na **SQL Azure Management Portal**. Zadajte meno administrátora servera SQL Azure, jeho heslo a potvrďte tlačidlo “**Log on”.**
4. Počkajte, pokiaľ sa nenadviaže spojenie s databázou a pokým sa nezobrazí hlavná stránka pre správu databázy.
5. Vyberte z ribbon menu položku “**New Query**”.
6. Vložte do plochy T-Sql príkaz na načítanie obsahu tabuľky KritickeData:

```
SELECT convert(varchar(max),hodnota) as zasifr_hodnota from kritickedata
```

7. Povrďte “**Run**”.

Všimnite si, že dáta sú v databáze uložené v zašifrovanej forme.

Zhrnutie

V praktickom cvičení ste vytvorili server SQL Azure a následne databázu SQL Azure. Vytvorili ste hostovanú službu na Windows Azure. Potom ste vygenerovali testovací X.509 certifikát a nasadili ste ho do hostovanej služby. Nakoniec ste vytvorili webovú aplikáciu, v ktorej ste utajované dáta najskôr zašifrovali asymetrickým algoritmom RSA a potom ste ich uložili do databázy SQL Azure.