

Week	Date	Teaching	Attended
9	Mar 2013	Lab 8: Azure Cloud Development	

**Aim:** The aim of this lab is to investigate the creation of a range of Cloud-based services within the Microsoft Azure Cloud

**Time to complete:**

4 hours (Two supervised hours in the lab, and two additional hours, unsupervised).

**Activities:**

- Complete Lab 8: Azure Cloud Development  
<http://www.dcs.napier.ac.uk/~cs342/CSN10102/Lab8.pdf>
- Complete the **End Of Unit Tutorial Questions** for this unit, online:  
<http://asecuritysite.com/>

**Learning activities:**

At the end of these activities, you should understand:

- How to setup cloud-based services.
- Create cloud-based infrastructures, and link to them.
- Create a cloud-based data infrastructure.
- IAAS and PAAS provided from the Azure cloud.

**Reflective statements (end-of-exercise):**

What are the key strengths of the Azure Cloud as opposed to other Cloud infrastructures?

What are the access permissions on TCP ports by default?

How are ports opened up for use, and how can then be locked-down?

# Lab 8: Microsoft Azure Cloud


Prof. Bill Buchanan, Rich Macfarlane 2013

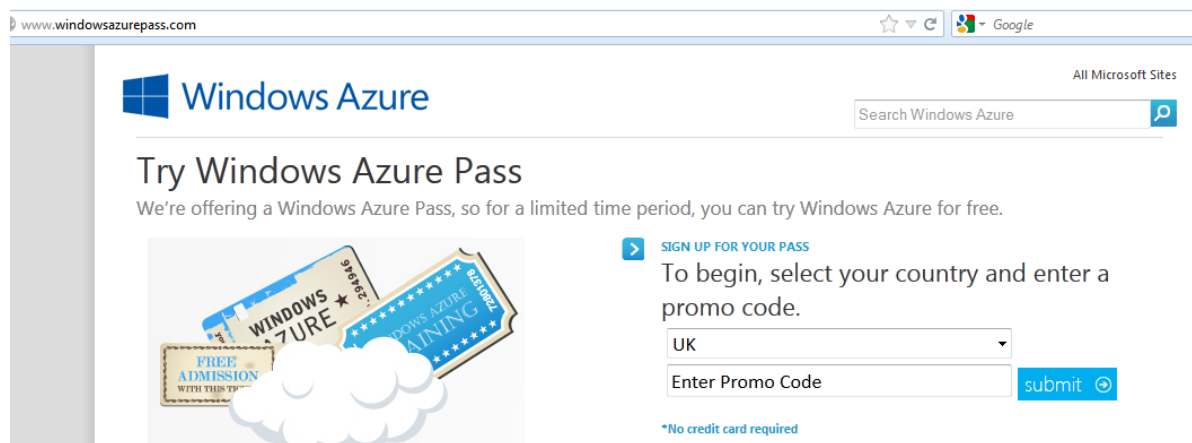
## 8.1 Details

Aim: The aim of this lab to setup an Azure Cloud infrastructure, and develop a deeper understanding of cloud-based services, with a specific focus on the security setup.


## 8.2 Register for Azure

You should have received an Azure promotional pass from your instructor, and previously registered it at the following link:


 Azure Educational 5 moth pass:  
<http://www.windowsazurepass.com/>

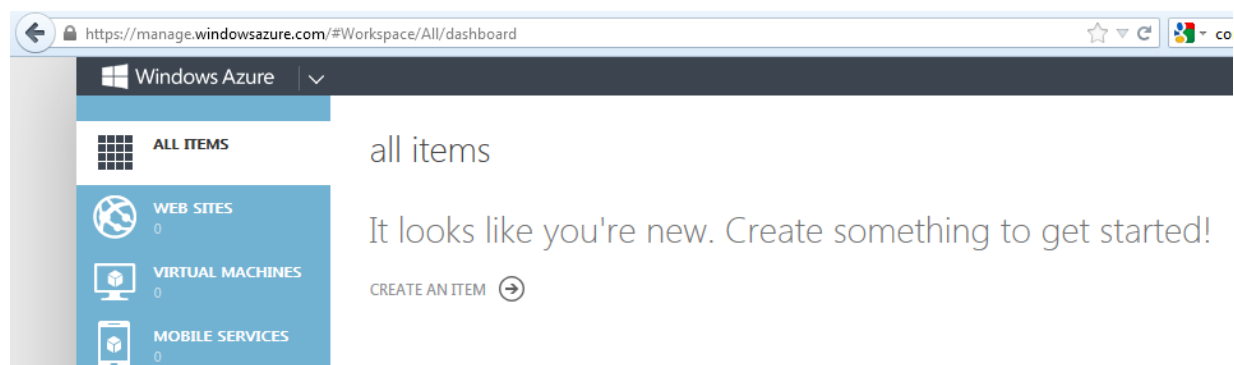


## 8.3 Create Cloud-based Web Service

 A video demo of this part the lab can be found at:  
<http://youtu.be/I4S8M0LgSwk/>

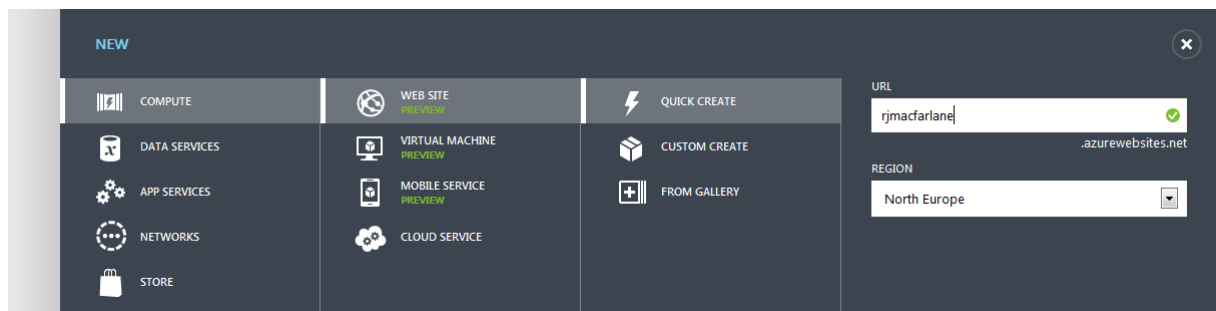
First login to the Azure Cloud:

 The **Azure cloud** development portal:  
[https:// www.windowsazure.com/](https://www.windowsazure.com/)

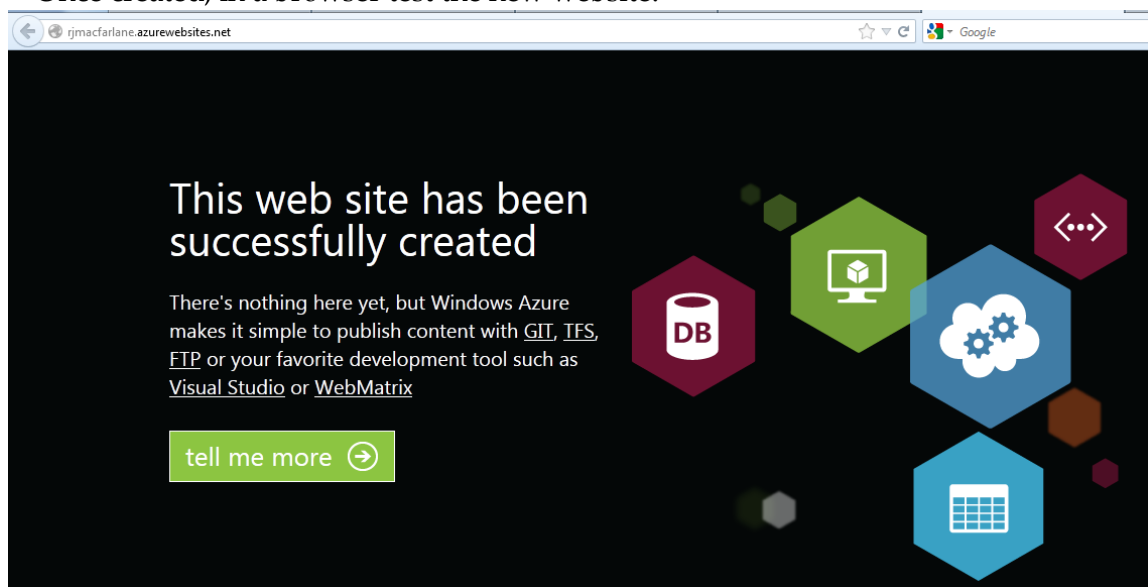


## Create a Web Site

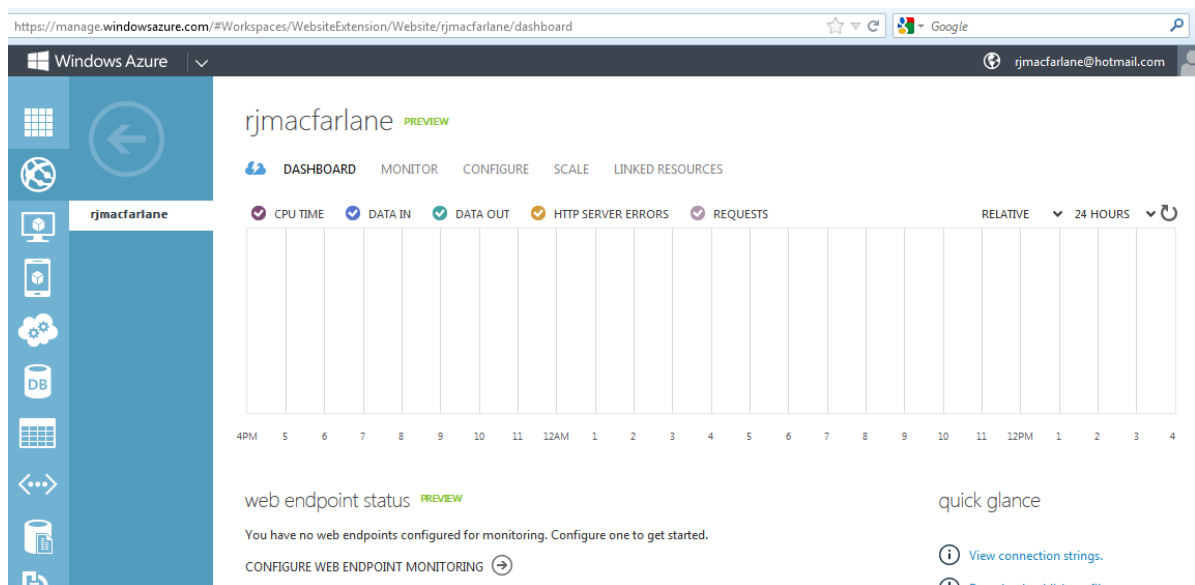
- Create a Web site named *matricno.azurewebsites.net*



- Once created, in a browser test the new website:



- In the Azure development portal, click on your new Website, and then click **Dashboard**:

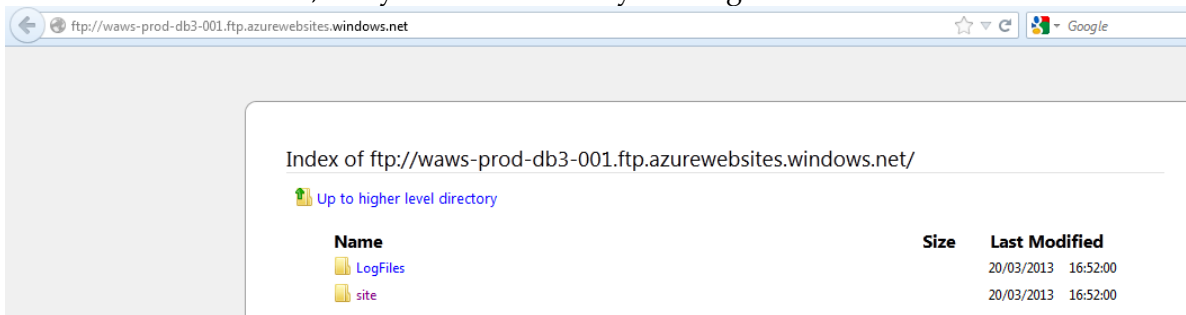


- From the Dashboard, set an FTP User/Password using **Reset deployment credentials**
- From the Dashboard, determine your FTP details.

#### Questions

- Q: What is your FTP User? (including the domain)
- Q: What is your FTP Hostname?

- From the Dashboard, test your FTP details by clicking the FTP Hostname.



### Create a Webpage

- Start-up Visual Studio, and add an **ASP.NET Web Application**. Note that the home file is named **Default.aspx**. Create your own home page (by modifying the HTML in the aspx page).

### Publish the Webpage on your Azure Website using FTP

- From Visual Studio publish your site by right clicking the web application and selecting publish. Use FTP and your FTP details from azure. Remember to publish to the *website/wwwroot* folder on your remote site.
- Test the webpage is working.

## 8.4 Create an Azure Cloud Virtual Machine



A video demo of this part the lab can be found at:

<http://youtu.be/6WSw3qlwIc4>

Now we will do the same, but this time we will create a Virtual Machine to perform the same thing:

- Create a Virtual Machine using Window 2008 for *initials\_matricno*.cloudapp.net
- Connect to your VM with Remote Desktop.

- In your VM, add a Role of Web Server (IIS).

**Note that this part of the lab will take some time to setup, so go onto the next section on SQL Data, and return once the IIS Role has been added.**

- In your VM, test HTTP is working locally, by connecting to the FTP Server from the command line FTP Client, and via a browser with the URL ftp:localhost.
- In your VM, add a Role Service of FTP.
  - Test FTP Server locally
- In your VM, setup the FTP firewall with the Public IP address of your site, and for the data ports of 7000-7002.
  - From Server Mgr IIS>FTP Firewall Support
    - Set Data Channel Port Range 7000-7002
    - External IP Address to the Server Public IP Address such as 168.63.x.x
  - From IIS>FTP Authentication
    - Basic Authentication>Enable
- Back in the Azure Development portal, select Endpoints
  - Add Endpoint for FTP of ports 21 and 20
  - Add Endpoints for Passive FTP data channels for ports 7000, 7001 and 7002
- From your local machine, Test the remote FTP Server using FTP command line client, and from a web browser.
- *On the Azure Management Page, add end points of 80, 20, 21, 7000, 7001, and 7002 to your Virtual Machine. Note that 7000, 7001 and 7002 will be used for the passive FTP ports.*

### Create a Webpage

- Start-up Visual Studio, and add an **ASP.NET Web Application**. Note that the home file is named **Default.aspx**. Create your own home page (by modifying the HTML in the aspx page).

### Publish the Webpage on your Azure Website using FTP

- From Visual Studio, using your FTP details, publish your site
- From the local machine, test the new webpage works.

## 8.5 Create SQL Data Infrastructure



A video demo of this part the lab can be found at:  
[http://youtu.be/vuEsRM\\_kh7I](http://youtu.be/vuEsRM_kh7I)

Connect to the SQL Data infrastructure:

- On the Azure Management Page, add a New SQL database.

- On the Azure Management Page, select Manage allowed IP addresses, and add the range 0.0.0.0 to 255.255.255.255.
- On the Azure Management Page, go to Manage URL, to administer your database.
- On the Database Admin Page, create you database schema for the following:

Item	Description	Price
XT311	CISSP Certification	10
XG312	CCNA Security	22
OT821	CCNP ONT	33
XP411	CCNP Route	44

- On the Database Admin Page, next add the required rows to populate the database.
- Start-up Visual Studio, and add an **ASP.NET Web Application**. Note that the home file is named **Default.aspx**. Add the following code to the page:

```
<asp:Button ID="Button1" runat="server" Text="Button"
OnClick="Button1_Click" />
<asp:GridView ID="GridView1" runat="server" >
</asp:GridView>
```

- Next click on the Button and add the following in the code behind:

```
System.Data.SqlClient.SqlConnection _SqlConnection = new
System.Data.SqlClient.SqlConnection();
    _SqlConnection.ConnectionString = "PASTE HERE>>>>";

System.Data.SqlClient.SqlCommand _SqlCommand =
    new System.Data.SqlClient.SqlCommand("SELECT * FROM Table1", _SqlConnection);

    System.Data.SqlClient.SqlDataAdapter _SqlDataAdapter
        = new System.Data.SqlClient.SqlDataAdapter();
    _SqlDataAdapter.SelectCommand = _SqlCommand;

    DataTable _DataTable = new DataTable();
    _DataTable.Locale = System.Globalization.CultureInfo.InvariantCulture;
    _SqlDataAdapter.Fill(_DataTable);
    GridView1.DataSource = _DataTable;
    GridView1.DataBind();
```

- On the Azure Management Page, get your connection string, and paste it into the code given above.
- Run the page, and make sure it works.
- Create a Web Site, and upload your code to it, and make sure it works in the Cloud.
- Now add a new table to your database (Table2), with the following:

ID	FirstName	Surname	FullAddress	Test 1	Test 2	Gender	Age
1	Fred	Smith	10 Fake Street	10	20	M	30

2	Bert	Smith	1 Round Lane	30	40	M	40
3	Bob	Malcolm	5 Square Road	100	30	M	22
4	Eve	Almond	11 Full Lane	45	40	F	56
5	Freddy	Smith	111 Edinburgh Road	50	50	M	43

- On the Azure Management Page, lock down your SQL connection, so that only your IP address is allowed to manage the database. Check this by accessing it on your host, and on another one.

## 8.6 Upload database



A video demo of this part the lab can be found at:  
<http://youtu.be/FRV6eIEQC1c>

1. On Microsoft Azure create a new SQL database.
2. Next install **Microsoft SQL Server Migration Assistant for Access**, and run the 32-bit version.
3. Next import the following database into your Cloud-based infrastructure:

<http://billatnapier.com/db1.zip>

## 8.7 Integrating with database

Once successful, go back to Microsoft Azure, and use the Manage URL link to connect to your database, and run the command of (Figure 1):

```
SELECT * from db1
```

One at a time, run the following SQL commands, and now how they operate (and any problems you have with them):

```
SELECT * FROM db1 ORDER BY Surname
SELECT * FROM db1 ORDER BY Surname DESC
SELECT * FROM db1 ORDER BY Age
SELECT * FROM db1 ORDER BY Gender
SELECT FirstName FROM db1 WHERE (Gender='M')
SELECT FirstName FROM db1 WHERE (Gender='F')
SELECT Surname FROM db1 WHERE (Gender='M')
SELECT Surname FROM db1 WHERE (Gender='F')
SELECT First(Surname) FROM db1 WHERE (Gender='M')
SELECT Last(Surname) FROM db1 WHERE (Gender='M')
SELECT Max(Age) FROM db1
SELECT Min(Age) FROM db1
SELECT FirstName FROM db1 WHERE (Surname='Smith' OR Surname='Almond')
SELECT Avg([Test 1]) FROM db1
SELECT Avg([Test 1]) FROM db1 WHERE (Age>30)
SELECT Sum([Test 1]) FROM db1
SELECT Sum([Test 1]) FROM db1 WHERE (Age>30)
```

```

SELECT Count(FirstName ) FROM db1 WHERE (Age<30)
SELECT Count(FirstName ) FROM db1 WHERE (Age>30)
SELECT Count(FirstName ) FROM db1 WHERE (Age=30)
SELECT FirstName,Surname FROM db1
SELECT DISTINCT Surname FROM db1
SELECT FirstName,Surname,Age,[Test 1] FROM db1 WHERE (Gender='M')
SELECT FirstName,Surname,[Test 1],[Test 2] FROM db1 WHERE (Gender<>'M')
SELECT FirstName,Surname,[Test 1],Age FROM db1 WHERE Age BETWEEN 10 AND 50
SELECT FirstName,Surname,[Test 1],Age FROM db1 WHERE Age IN (22,56,33)
SELECT FirstName,Surname,[Test 1],Age FROM db1 WHERE Surname LIKE 'Sm%'
SELECT FirstName,Surname,[Test 1],Age FROM db1 WHERE Surname LIKE '[AaSuu]%'
SELECT FirstName,Surname,[Test 1],Age FROM db1 WHERE Surname NOT LIKE 'Sm%'
SELECT Gender,AVG([Test 1]) FROM db1 GROUP BY Gender
SELECT FirstName,Surname,[Test 1] from db1 where Surname in ( 'Smith', 'Almond') ORDER BY
Surname

```

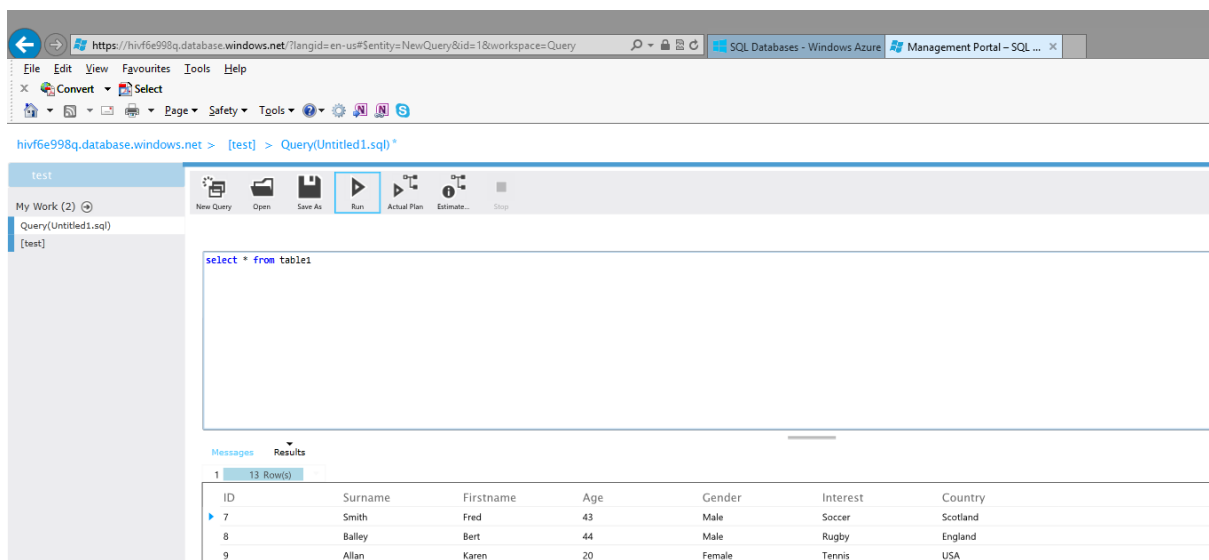


Figure 1: Sample

Hint: Sample queries are at: <http://www.asecuritysite.com/database/db>

## 8.8 ASPX Code

Using the code developed in the previous lab, create a Web site which integrates with your code. The button (or link) should implement the SQL command and the table should show the result.

## 8.9 Northwinds

Now create a new database and upload the following:

	Database: <a href="http://billatnapier.com/nwind.zip">http://billatnapier.com/nwind.zip</a>
---	--



Perform the following queries:

- One at a time, view all the tables (Categories, Customers, [Order Details], and so on.
- Find customers in Mexico
- Find customers in US or UK
- Find the number of customers in US or UK
- Find the number of customers in the database
- Find orders where France is the shipping country
- Find orders where it freight weights over 40
- Find suppliers who in the UK but not in London
- Find customers whose company name begins with A
- Find the number of customers whose company name begins with B
- What is the name of product 18?
- What is the name of products with an ID greater than 10?
- How were hired after Jan 17, 1993?
- The products for product ID of 4 (Dairy Products) which have less than 10 units left?
- Order products by unit price (ID, Price, and Product Name)
- Order products by unit price (ID, Price, and Product Name) - highest first
- Order products by unit price (ID, Price, and Product Name) with each category
- Show the number of condiments as MyCondiments for Category ID of 2

Hint: samples are at : <http://www.asecuritysite.com/database/db5>

## 8.10 AdventureWorks

---

Now integrate the following database:

<http://billatnapier.com/AdventureWorks.zip>

and do the following:

- Find productID, name and price for products
- Find ID the sales information from the state of Washington
- Find productmodelID and name from productsmodel

Hint: samples are at: <http://www.asecuritysite.com/database/db6>