Lab 14: Virtual Linux Server (LAMP) in the Amazon Cloud

14.1 Details

Aim: To create a useable virtual Linux Server, using open source LAMP (Linux, Apache, MySQL and PHP) software within the Amazon Elastic Cloud (EC2) IAAS infrastructure.

14.2 Activities

This part of the lab has three elements: the host machine (**DESKTOP**), a local Linux virtual machine image (**UBUNTU**), and a remote virtual Linux Server (**AWS_UBUNTU**) which has been created in the Amazon Cloud as shown below.

LAMP

The L in LAMP is the Linux Operating System. You will be given your own Linux **AWS_UBUNTU** instance details from your lab instructor.

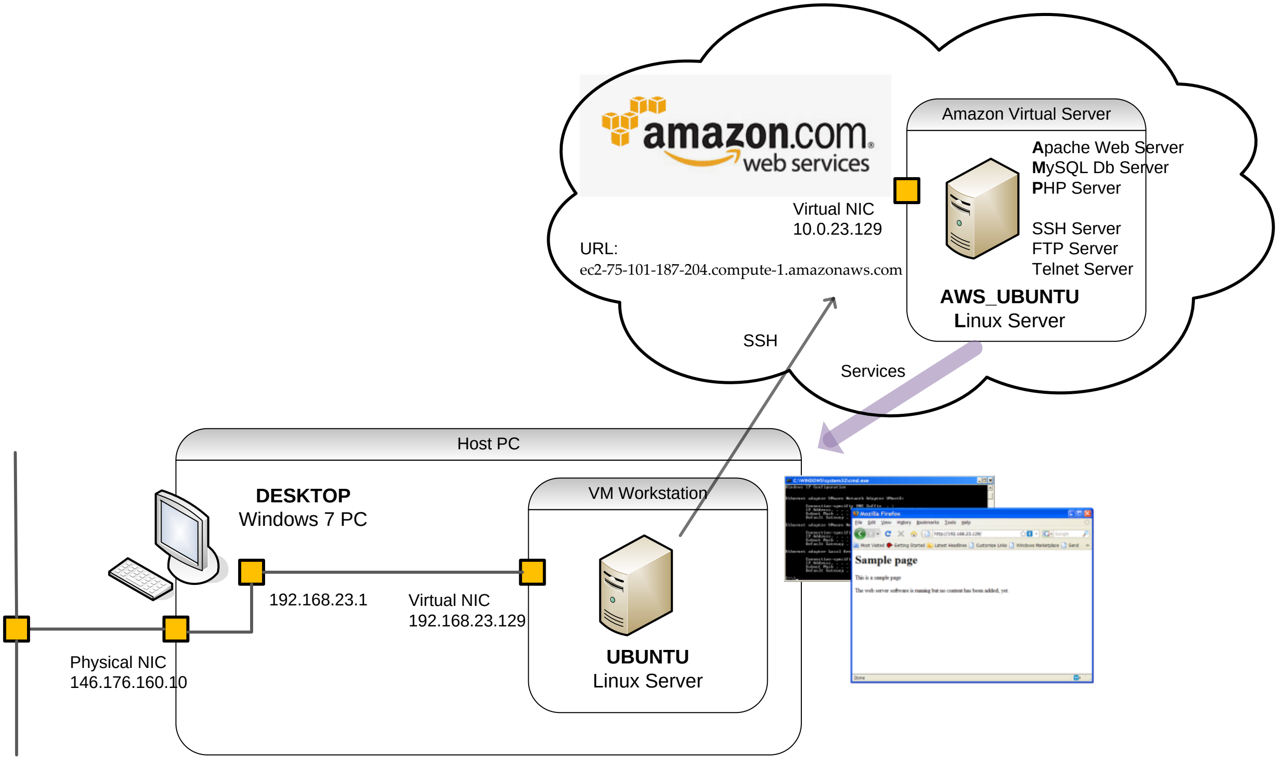


Figure 1 - Lab Architecture

L14.1 Start the Linux virtual machine (VM) **UBUNTU** on the local host system. Log in to the VM using **napier** with a password of **napier123**.

Details of an example **AWS_UBUNTU** instance are shown below in Table 1. You will be given details of your own instance from your lab instructor, which you can complete the table with.

Table 1 - Virtual Server Instance Example

AWS_UBUNTU Amazon EC2 Public URL	**AWS_UBUNTU** Password
ec2- ________________ .compute-1.amazonaws.com	____________

An overview of Linux commands, to assist with the lab, can be found at:
http://www.computerhope.com/unix/overview.htm

L14.2 Remote Administration of the AWS Server using Secure SHell (SSH)

There is a SSH server running on AWS_UBUNTU, which we can connect to, using public key authentication. The server has our public key, and we can connect using the associated private key to authenticate.

On the local UBUNTU machine, create the **private key** file for SSH Authentication such as **ssh_private_key.pem (**using a text editor), and copy and paste the following text into it, such as with the **`vi ssh_private_key.pem`** command, or using the **gedit** GUI-based text editor.

An overview of basic vi commands, can be found at:
http://www.tuxfiles.org/linuxhelp/vimcheat.html

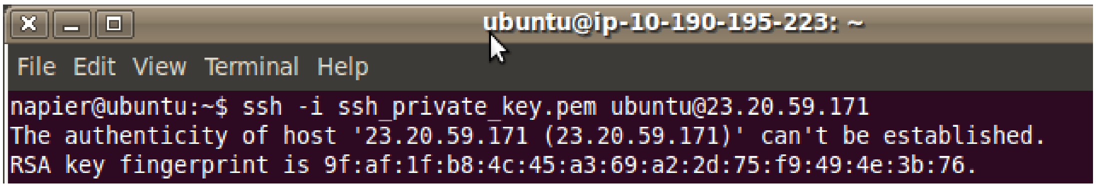
```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAtZ7T4sPiQVGtQ74ZEFPhIr6ayDj7w0cBZ8JN0NXuAFvVPMST1VLTUcdki+sx
GieNYvnUFSoTOzRvkWcn+n12TxpmyiTggt04cj6zHeMeCXqiYJCejvCTGVxIgjCTifBQg1/ZLkQ4
2FGH0YD6bFJJStuJ73KeGBb5yPy/AjlzaF0+1d+i/wS+oWT80ftT5w7SmLczWcYROtbg9z2t25zL
65L9vn97++OWwF7rCgiytWnBTDr4NlIfukdjco0MZDpczkxgEAmwlqgNYl6bWryVI7JbZm8mIfOR
MNsdlv+0v5GBBaoZdotRy8+hixrGZbfnR06Ct26isVA+HaOqWCb00QIDAQABAoIBAGagIMp6NVcD
eAxXVoJLY2PmoD+TM2/cp4ah3Kasu5eoTI3R5lccPhxvtvj5JP3Ka7IJyTVMWSGTN5fJ2mVIj5mT
KZH/1H0d0896bmvs8gQZL7exAGd0uSoTY0VtfXDsQhJ6DpZ+JkDVkRFo+BhNRuztsLuE4KBsGyje
6G+xVu0ZD15K5bbCPjB9Y4mSQH0jSFBeBQnzvTziMI9+qit/wKe6F9ZljO5uci1HsAIhwBfXqiyh
GBohbqaht6IRrKwVh21v+aDbNk9zHtc9V0AzPdsIcSSklpIkYCcIIoARpVAKZvQwWU2CwPk9iuA4
AafoVzbFDnxiN3s74xN06pKng6ECgYEA5BpqRxICQz6tItYJhcAFjTB+IvlCl1Xt09NvVA4wubm/
WE7titaDug2/UONsYF8c153K3OC5tC5lsBSQc7SnRgLtoJ9PHGROB0kSydSZr+08gYl1sKblPTxf
3Q9PQ/sf4DboGTIAg1AAzZSmeC5Hy4+7TFByMUkeLzwuRRw35J8CgYEAy9UahePGOORTK9cu2Jvr
+lXDVziUPRx+je4cmAEBzeX8kuffkKtL1oiGSrjufL5wSwqNDZjHT+Tq8sYrZFQyq7TPdn0/z5MD
J2AORqx/7ehF377Zhsyp4YIqxlz3W5MPTjGlNItbpKWEXGl3OUHSOQi7KR6WxLY0nSA6ZnQpwI8C
gYEAxphk/XmldKyGCzV01vlHHBUjTQndu2r6BJoqbHFqZNleyeD2GhIIYB2F+0P2YLWOo/9i1wnl
RQW8TSCqX8LifCcigt0RALQn51nsvMKYCD6xlkp6qBG2tqjxAcNJjPOAXjMPwpfxMGKgQFzyzuHk
aku6k4fa1CRH3nI0CsG1vBsCgYAe4lk0dECymhXoDKCyfV2R/rCsZIxBfjHGu49lkLF/Sxur/qon
N200QldmtwKbsBc/p4ebycxZVKwswM4OU5uh3LGamHu0refKbYl8N+LiX1FHB5naDOSooJpul+N6
4pttE6Seg4cNOW8GuYwxBsIdli8VmebqQ1lRfXKb13oEDQKBgCN5NOwKAsJz+zoLLM0UBQWDE3k7
PBVrCYNpO1nFTal9g/6PD/P+eZUfZjWOKNqOr6enQV+5nOtpCDhVzgXR9hFdkYhlpNQ5WD5uj/Z/
VgwM9yNgWSb5nKEgn56QfutlUvbv6lEqRqd0RUcHqvPvoIajQvQBnDBuCHdtZeGkC113
-----END RSA PRIVATE KEY-----
```

Change the permissions of the RSA Private Key file to be read only, using the `chmod` command, such as:

```
chmod 400 ssh_private_key.pem
```

From **UBUNTU**, use the **ssh** client to log into the **remote** Amazon server **AWS_UBUNTU,** with the username **ubuntu,** using a command such as the following (where you need to specify your own **AWS_UBUNTU Public URL**)

```
ssh -i ssh_private_key.pem ubuntu@ec2-xx-xx-xx-xx.compute-1.amazonaws.com
```



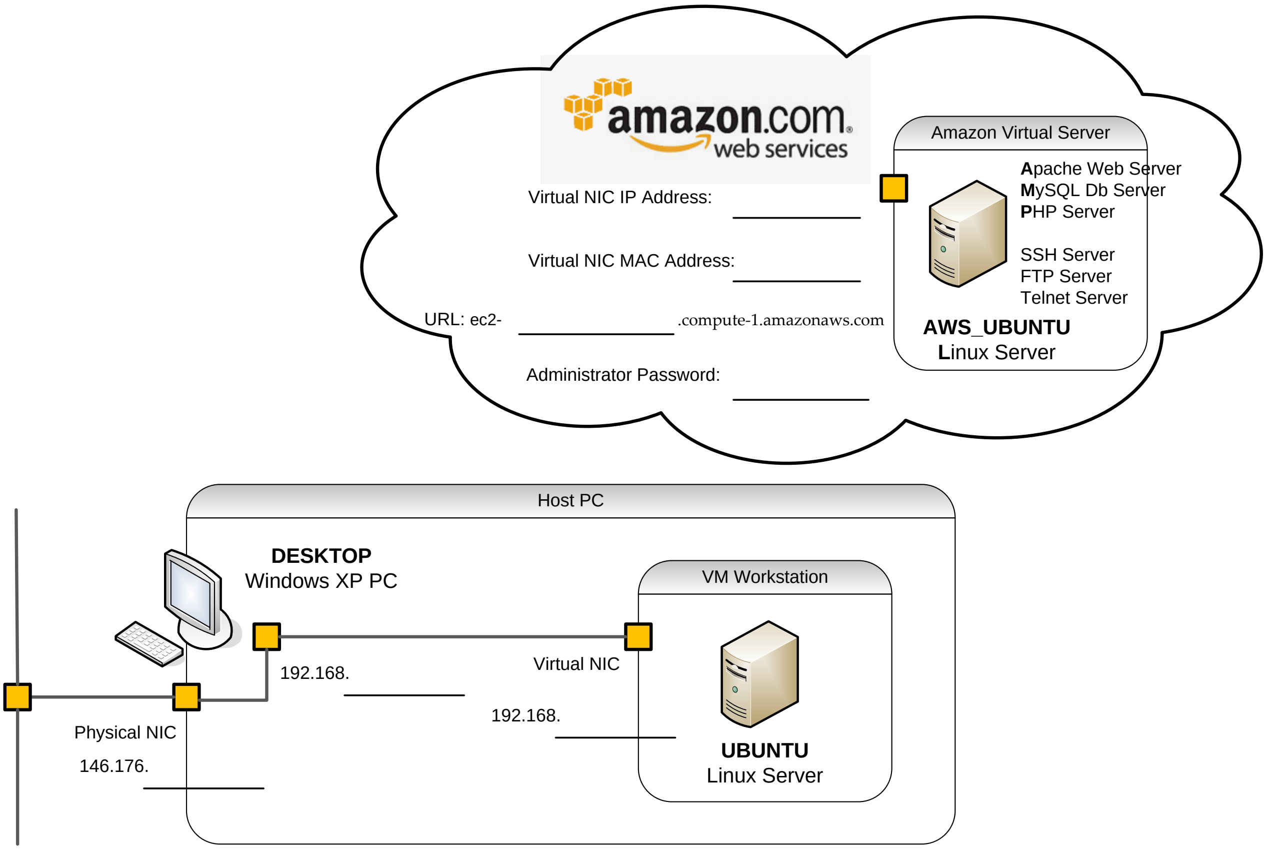
```
napier@ubuntu:~$ ssh -i ssh_private_key.pem ubuntu@23.20.59.171
The authenticity of host '23.20.59.171 (23.20.59.171)' can't be established.
RSA key fingerprint is 9f:af:1f:b8:4c:45:a3:69:a2:2d:75:f9:49:4e:3b:76.
```

You should now be remotely logged into the virtual Linux server **AWS_UBUNTU** in the Amazon Cloud. Use the `ifconfig` , `pwd`, `ls` commands to check the details of the current virtual machine.

☞ Have you successfully logged in to the remote Server? YES/NO

(If NO, then see your instructor for help)

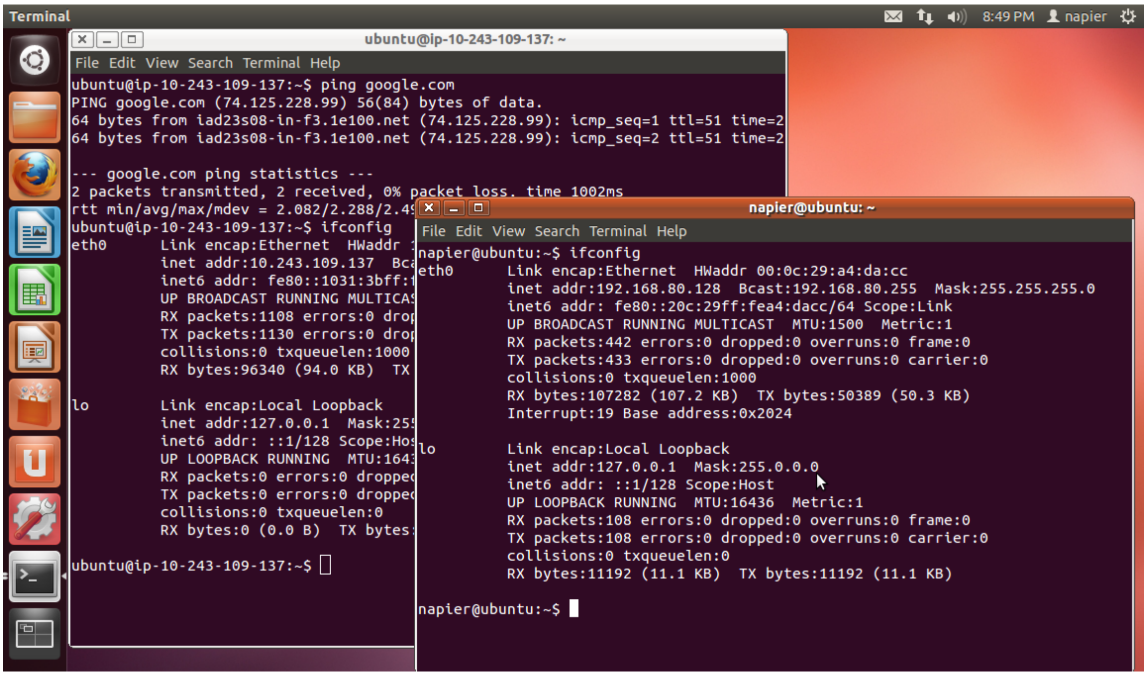
☞ What is the Linux prompt for **AWS_UBUNTU**?



Check connectivity between AWS_UBUNTU and the Internet, and open a second Terminal in local UBUNTU, and check connectivity between local UBUNTU and AWS_UBUNTU, using **`ping`**.

☞ Can AWS_UBUNTU ping an Internet server?

YES/NO


Terminal
8:49 PM napier
ubuntu@ip-10-243-109-137: ~
File Edit View Search Terminal Help
napier@ubuntu: ~
File Edit View Search Terminal Help

```
ubuntu@ip-10-243-109-137:~$ ping google.com
PING google.com (74.125.228.99) 56(84) bytes of data.
64 bytes from iad23s08-in-f3.1e100.net (74.125.228.99): icmp_seq=1 ttl=51 time=2
64 bytes from iad23s08-in-f3.1e100.net (74.125.228.99): icmp_seq=2 ttl=51 time=2

--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 2.082/2.288/2.49
ubuntu@ip-10-243-109-137:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr
          inet addr:10.243.109.137  Bca
          inet6 addr: fe80::1031:3bff:
          UP BROADCAST RUNNING MULTICAS
          RX packets:1108 errors:0 drop
          TX packets:1130 errors:0 drop
          collisions:0 txqueuelen:1000
          RX bytes:96340 (94.0 KB)  TX

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255
          inet6 addr: ::1/128 Scope:Hos
          UP LOOPBACK RUNNING  MTU:1643
          RX packets:0 errors:0 droppe
          TX packets:0 errors:0 droppe
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:

ubuntu@ip-10-243-109-137:~$
```

```
napier@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:a4:da:cc
          inet addr:192.168.80.128  Bcast:192.168.80.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea4:dacc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:442 errors:0 dropped:0 overruns:0 frame:0
          TX packets:433 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:107282 (107.2 KB)  TX bytes:50389 (50.3 KB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:108 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11192 (11.1 KB)  TX bytes:11192 (11.1 KB)

napier@ubuntu:~$
```

☞ Can DESKTOP/UBUNTU ping AWS_UBUNTU?

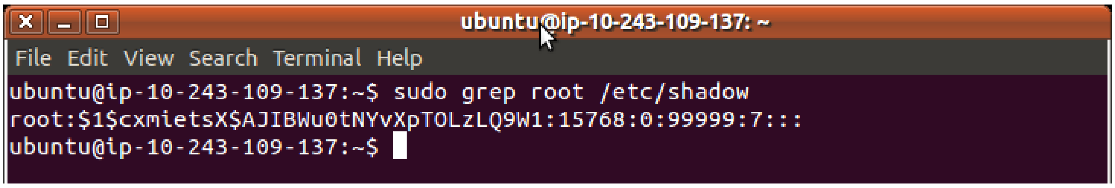
YES/NO

L14.3 User Accounts and Passwords

1. Creating your own user accounts, and changing passwords is the first thing you should do on your server, as a basic security measure. The passwords should only be readable by an administration level user.

On **AWS_UBUNTU** check the **root** user password with:

```
sudo grep root /etc/shadow
```


ubuntu@ip-10-243-109-137: ~
File Edit View Search Terminal Help

```
ubuntu@ip-10-243-109-137:~$ sudo grep root /etc/shadow
root:$1$cxmietsX$AJIBWu0tNYvXpTOLzLQ9W1:15768:0:99999:7:::
ubuntu@ip-10-243-109-137:~$
```

2. Passwords should always be stored in a well encrypted/hashed format.

In this case, the **root** password is stored as a **salted hash** in the format `$hash_algorithm$salt$hash_signature`

The first $ and the char after define the hash algorithm used, $1 for MD5, $2 for Blowfish etc. The chars after the second $ is the salt value, and the chars after the third $ is the salted hash signature.

☞ What hash algorithm is being used to store the password on AWS_UBUNTU?

☞ What is the hash signature for the root password?

On **AWS_UBUNTU** Change the **root** user password to **napier123** with:

```
sudo passwd
```

Change the **ubuntu** user password to **napier123** using:

```
sudo passwd ubuntu
```

On **AWS_UBUNTU** check the **root** and **napier** user passwords with `grep` commands on the shadow file.

☞ What is the hash signature for the root password?

☞ What is the hash signature for the ubuntu password?

☞ Why are the Hash Signatures different?

The Salt which the Linux passwd command adds, means every Hash will be different; even for the same password.

To replicate the **root** password hash try: (with the salt string from your shadow file)

```
openssl passwd -1 –salt SALTSTING “napier123”
```

☞ Did you successfully produce the same hash signature?

YES/NO

3. Password Changes

Administration level passwords should be set to expire within a fairly short period such as 60 days, and so have to be changed at frequent intervals.

Check the options for the passwd command using:

```
sudo passwd --help
```

☞ Which argument could be used to enforce periodic password changes?

Enforce changing of the root passwd every 60 days with:

```
sudo passwd --maxdays 60 root
```

4. Administrative accounts should not be shared between users, and passwords should be different from user non-administrative accounts.

Now, enforce changing of the Ubuntu passwd every 30 days, and then change the password to **napier**.

☞ What was the commands used?

Linux Services

L14.4 Check services running and the associated processes on **AWS_UBUNTU** with commands such as:

```
netstat -au
netstat -at
```

or

```
netstat -a | grep udp
netstat -a | grep tcp
```

☞ Are there any established connections to services?

YES/NO

☞ Which service is connected to a client?

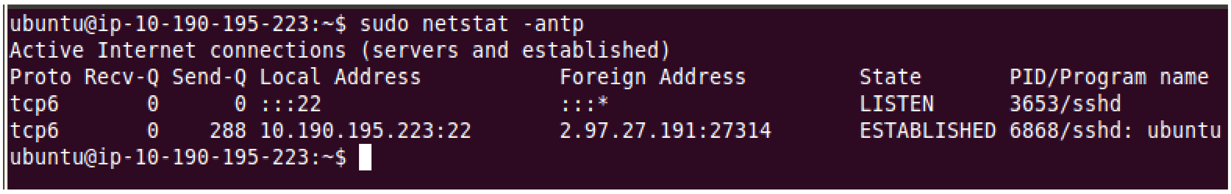
☞ List the IP Address of the client?

To view the associated processes for services and the port numbers they are on, on **AWS_UBUNTU** use netstat command with the –p flag such as:

```
sudo netstat -antp
```

☞ List the services, port numbers and the processes?

You should get output similar to the following:



```
ubuntu@ip-10-190-195-223:~$ sudo netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp6       0      0 :::22                   :::*                    LISTEN      3653/sshd
tcp6       0    288 10.190.195.223:22       2.97.27.191:27314       ESTABLISHED 6868/sshd: ubuntu
ubuntu@ip-10-190-195-223:~$ 
```

L14.5 Update System

You should be running a fully up to date and patched OS and software, and anything which is out-dated should be removed from the system, to prevent vulnerabilities.

Update your **AWS_UBUNTU** image, so you can see the latest **packages** available to install, using the Advanced Packaging Tool (APT) **`apt-get update`** command:

```
sudo apt-get update
```

> 🖱 An overview of Linux apt-get command can be found at:
> https://help.ubuntu.com/8.04/serverguide/apt-get.html

Update software packages, including security updates, with:

```
sudo apt-get upgrade -y
```

Use the **`man`** command to find out what the –y flag is used for.

> ☞ What does the –y switch do?

The **`dpkg`** command can be used to get information about packages:

```
dpkg –help
```

The `dpkg -l` lists information about all installed packages:

```
dpkg –l
```

List information about specific installed packages:

```
dpkg –l dpkg
```

LAMP – **Apache Web Server**

L14.6 The A in LAMP is the **Apache Web Server**. Install the Apache Web server using the **`apt-get install`** command (updates if already installed):

```
sudo apt-get install -y apache2
```

> 🖱 An overview of the important Apache files can be found at:
> http://wiki.apache.org/httpd/DistrosDefaultLayout#Debian.2C_Ubuntu_.28Apache_httpd_2.x.29:

To start/stop/restart the Apache web server, use the server control script in /etc/init.d/ such as:

```
sudo /etc/init.d/apache2 start
sudo /etc/init.d/apache2 start
sudo /etc/init.d/apache2 restart
```

☞ Start/restart the Apache server, then using netstat, check it is now running? YES/NO

From the **AWS_UBUNTU** command line, test that you can access the Web server from inside the Linux remote VM, using telnet:

```
telnet localhost 80
```

Get **information** on the default web page:

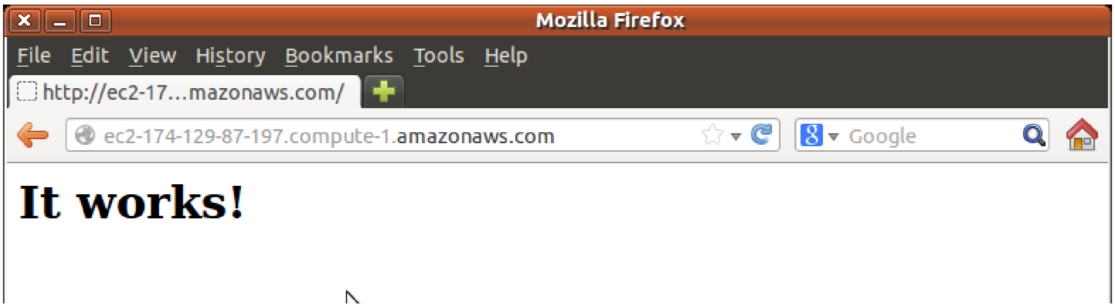
```
HEAD / HTTP1.1
```

Then get the **page contents** for the index.html page:

```
GET /index.html
```

☞ What message does the web page contain?

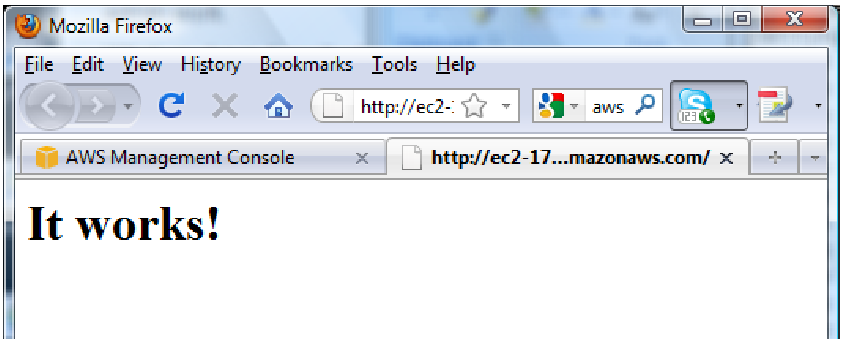
Test the access to the remote Web Server from your local Linux VM **UBUNTU**, using a web browser, such as shown below


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://ec2-17...mazonaws.com/
ec2-174-129-87-197.compute-1.amazonaws.com
Google

It works!

Figure 2 - UBUNTU to AWS_UBUNTU Apache Server

Test the access to the remote Web Server from your local system **DESKTOP**, using a web browser, such as shown in Figure 3**.**


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://ec2-
aws
AWS Management Console
http://ec2-17...mazonaws.com/

It works!

Figure 3 – DESKTOP to remote Web Server

L14.7 Go to the `/var/www` folder, and modify the default web page, so that it has a new greeting.

L14.8 Apache Logging

Go to the `/var/log/apache2` folder, and monitor the Apache log file, for the last 5 Web Server accesses, using the **`tail`** command:

```
tail -n 5 -f access.log
```

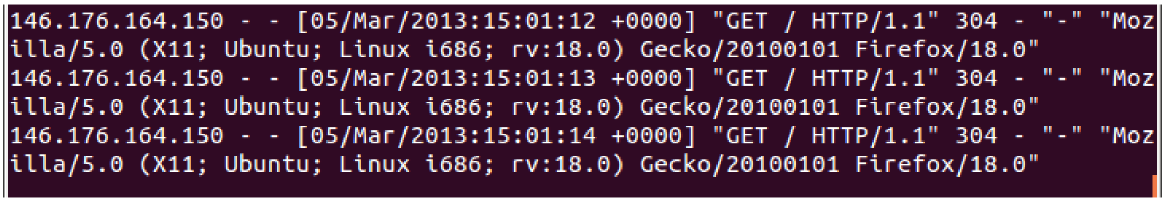
While watching the window running the tail command, test the new greeting, using a Web browser from **DESKTOP**.

☞ Can you access the updated web page, with the new message?
YES/NO

☞ Can you see the web page accesses being added to the access log file?
YES/NO

☞ What Information can you determine from the accesses, and what is different for the two accesses from different machines?

Tail can be stopped with CTRL+C



```
146.176.164.150 - - [05/Mar/2013:15:01:12 +0000] "GET / HTTP/1.1" 304 - "-" "Moz
illa/5.0 (X11; Ubuntu; Linux i686; rv:18.0) Gecko/20100101 Firefox/18.0"
146.176.164.150 - - [05/Mar/2013:15:01:13 +0000] "GET / HTTP/1.1" 304 - "-" "Moz
illa/5.0 (X11; Ubuntu; Linux i686; rv:18.0) Gecko/20100101 Firefox/18.0"
146.176.164.150 - - [05/Mar/2013:15:01:14 +0000] "GET / HTTP/1.1" 304 - "-" "Moz
illa/5.0 (X11; Ubuntu; Linux i686; rv:18.0) Gecko/20100101 Firefox/18.0"
```

LAMP - PHP

L14.9 The P in LAMP is typically the **PHP** scripting language (can also be **Perl** or **Python**).

Install PHP and the Apache module with the **`apt-get install`** command:

```
sudo apt-get install php5
```

Enable the PHP Apache module, and restart the Apache Web Server using:

```
sudo a2enmod php5
sudo /etc/init.d/apache2 restart
```

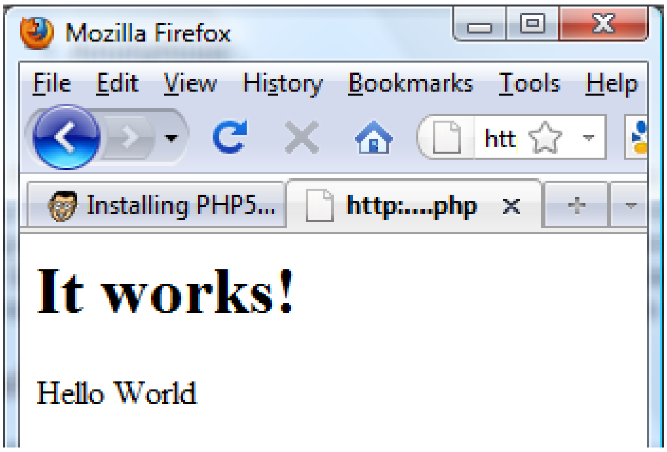
Create a new PHP file in the `/var/www` folder using a text editor such as with:

```
sudo vi webpage.php
```

and add:

```
<html><body><h1>It works!</h1>
<?php
print("Hello World");
?>
</body></html>
```

Test the access from your host system **DESKTOP,** as shown below



It works!

Hello World

Figure 4 - Browsing new.php on remote Web Server

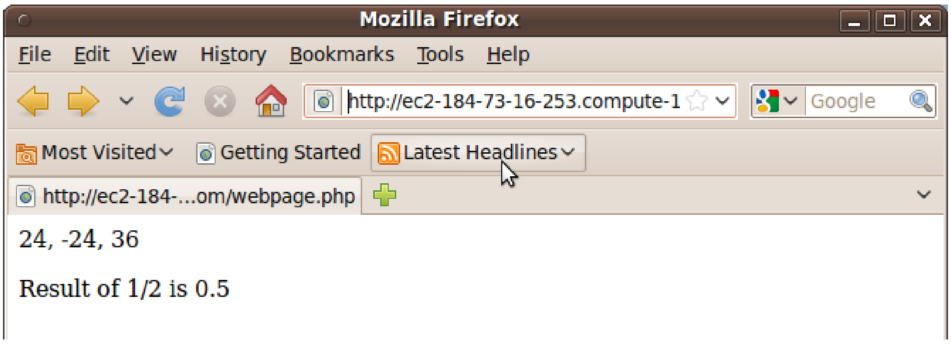
PHP Documentation, to assist with the lab, can be found at: http://uk3.php.net/manual/en/index.php

Next create web pages to investigate the following PHP code:

```
<?php
/* hyperlink */
print("<a href='index.html'>Index Page</a>");  // ; terminates php statements
print("<br>This is an <b>important</b> example");
?>
```

```
<?php
$a = 24;   // decimal number
$b = -24;  // a negative number
$c = 0x24; // hexadecimal number
print "<P>$a, $b, $c";
$val1=1;
$val2=2;
$result=$val1/$val2;
print "<P>Result of $val1/$val2 is $result";
?>
```

The PHP code can be tested from a Web Browser on the local UBUNTU VM, as shown below:


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://ec2-184-73-16-253.compute-1
Google
Most Visited
Getting Started
Latest Headlines
http://ec2-184-...om/webpage.php

24, -24, 36

Result of 1/2 is 0.5

```
<?php
/* Strings and String Operators */
$name1 = "Fred ";
$name2 = "Smith";
print "Hello \"Fred\" . How are you?";         // . Concatenates strings
print "<br>Your full name is " . $name1 . " " . $name2;
$name1 .= $name2;                 // .= Appends strings
print "<br>Your full name is " . $name1
?>
```

```
<?php
/* Dates */
$today_us = date("Y-m-d");
$today_uk = date("d-m-Y");
$d_and_t = date("Y-m-d H:i:s (T)");
PRINT "Date (UK): $today_uk";
PRINT "<br>Date (US): $today_us";
PRINT "<br>Date and time: $d_and_t";
$today = getdate();
$month = $today['month'];
$mday = $today['mday'];
$year = $today['year'];
echo "<br>US Date is: $month $mday, $year";
?>
```

```
<?php
/* Loops */
for ($i=0;$i<11;$i++)
{
 $sqr_value=$i*$i;
 print "<BR>$i $sqr_value";
}

$i=0;
do
{
 $sqr_value=$i*$i;
 print "<BR>$i $sqr_value";
 $i++;
} while ($i<11);

$i=0;
while ($i<11)
{
 $sqr_value=$i*$i;
 print "<BR>$i $sqr_value";
 $i++;
}
?>
```

```
<?php
/*  Arrays */
$colors = array('red','blue','green','yellow');
for ($i=0;$i<4;$i++)
{
  print "<BR>Color : $colors[$i]";
}

print "<BR>\n";
print "<BR>\n";
$mths = array(1=>'January', 'February', 'March');
print_r($mths);
?>
```

```
<?php
/* Calling Std Funtions */
```

```
print "<table border='1'>";
print "<tr><td>Value</td><td>Hex</td><td>Binary</td></tr>";
for ($val=0; $val<=16; $val++)
{
  print "<tr><td>" . $val . "</td><td>" . dechex($val);
  print "</td><td>" . decbin($val) . "</td></tr>";
}
print "</table>";
?>


<?php
/* Calling Funtions & Global Var’s */
$val1 = 11;
$val2 = 21;

function Add ()
{
  global $val1, $val2;
  $val2 = $val1 + $val2;
}

function add_with_args($a,$b)
{
  return($a+$b);
}

$result=add_with_args($val1,$val2);
print"<BR>Result is $result";
Add ();
print "<BR>Result is $val2";
?>


<?php
print("Browser: $HTTP_USER_AGENT <br />\n");
print("IP Address: " . gethostbyname("amazon.co.uk"));
?>
```

```
<?php
/* File System Functions */
<?php
$fname = "./webpage.php";
$dname=".";
print "<BR>Real path: " . realpath($fname);
print "<BR>File name: " . basename($fname);
print "<BR>Dirname: " . dirname(realpath($fname));
print "<BR>Disk free space (MB): " .
round(disk_free_space($dname)/1024/1024);
print "<BR>Total disk space (GB): " .
round(disk_total_space($dname)/1024/1024/1024);
print "<BR>File group: " . filegroup($fname);
print "<BR>File mode: " . fileinode($fname);
print "<BR>File owner: " . fileowner($fname);
print "<BR>File permissions: " . decoct(fileperms($fname));
print "<BR>File size (Bytes): " . filesize($fname);
print "<BR>File type: " . filetype($fname);

$str= strftime("%H:%M, Date: %d-%m-%Y ", fileatime($fname));
print "<BR>File last accessed: " . $str ;
$str= strftime("%H:%M, Date: %d-%m-%Y ", filemtime($fname));
print "<BR>File last modified: " . $str ;

if (is_readable($fname))
{
  print "<BR>File is readable";
}
```

```
else
{
  print "<BR>File is not readable";
}
if (is_writeable($fname))
{
  print "<BR>File is writeable";
}
else
{
  print "<BR>File is not writeable";
}
if (is_executable($fname))
{
  print "<BR>File is executable";
}
else
{
  print "<BR>File is not executable";
}
?>
```

LA**M**P - **MySQL**

L14.10 The M in LAMP is the **MySQL** database. Install MySQL using:

```
sudo apt-get install mysql-server
sudo apt-get install php5-mysql
```

and then restart the Apache Web Server with:

```
sudo /etc/init.d/apache2 stop
sudo /etc/init.d/apache2 start
```

L14.11 Next login to the MySQL Server, using the command line MySQL client application `mysql`. This allows us to connect to the MySQL Server, create and modify databases, and execute SQL queries and view the results.

Connect to the MySQL Server as the **root** user (use the root password from earlier):

```
root@ip-10-212-230-15:/var/www# sudo mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.0.51a-3ubuntu5.5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

Mysql>
```

You are now connected to the MySQL Server as **root**. Next enter the following commands to create a database, and then add a table, and some data:

```
create database mydatabase1;
use mydatabase1;
```

```
create table mynames (id INTEGER NOT NULL, name VARCHAR(50) NOT NULL);
insert into mynames values(1, 'fred smith');
select * from mynames;
```

and show that the output is in the form of:

```
+----+------------+
| id | name       |
+----+------------+
|  1 | fred smith |
+----+------------+
2 rows in set (0.00 sec)
```

L14.12 Create a new table named **Products**, and add the following data:

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

L14.13 Next modify **`webpage.php`** to access your database (from the mynames table) with:

```
<html><body>
<?
$user="root";
$password="napier123";
mysql_connect(localhost,$user,$password);
mysql_select_db('mydatabase1')
    or die('Could not select a database.');

$sql = "SELECT name FROM mynames";

$result=mysql_query($sql);
$row = mysql_fetch_array($result);
print "Showing $row[name]<hr/>";

mysql_close();

?>
</body></html>
```

L14.14 Next modify **`webpage.php`** so that it displays the Products table.

TELNET Service

L14.15 Next install a **Telnet** server, and start the service using:

```
sudo apt-get install telnetd
sudo apt-get install inetutils-inetd

sudo /etc/init.d/inetutils-inetd restart
```

L14.16 Next test that you can login in with Telnet from **DESKTOP** / **UBUNTU**, such as with:

```
ip-10-212-230-15 login: ubuntu
Password: napier123
Last login: Tue Mar 23 10:12:11 UTC 2010 from 5ac77477.bb.sky.com on pts/0
Linux ip-10-212-230-15 2.6.24-10-xen #1 SMP Tue Sep 8 19:06:53 UTC 2009 i686

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/

  System information as of Tue Mar 23 10:30:01 UTC 2010

  System load: 0.0               Memory usage: 5%   Processes:       54
  Usage of /:  3.7% of 14.76GB   Swap usage:   0%   Users logged in: 1

  Graph this data and manage this system at https://landscape.canonical.com/
---------------------------------------------------------------------
At the moment, only the core of the system is installed. To tune the
system to your needs, you can choose to install one or more
predefined collections of software by running the following
command:

   sudo tasksel
---------------------------------------------------------------------
3 failures since last login.
Last was Tue 23 Mar 2010 10:32:21 AM UTC on pts/1.
```

☞ Have you successfully logged in to the remote Telnet Server from both **DESKTOP** and **UBUNTU**?

YES/NO

FTP Service

L14.17 Install an **FTP** server with:

```
sudo apt-get install vsftpd
```

and connect to the FTP server from **DESKTOP / UBUNTU**:

```
ftp ec2-xx-xx-xx-xx.compute-1.amazonaws.com
```

Login with an anonymous login:

```
220 (vsFTPd 2.0.6)

530 Please login with USER and PASS.
USER ubuntu
530 This FTP server is anonymous only.
USER anonymous
331 Please specify the password.
PASS w.buchanan@napier.ac.uk
230 Login successful.
```

```
PWD
257 "/"
```

☞ Have you successfully logged in to the remote FTP Server from **UBUNTU** and **DESKTOP**?

YES/NO

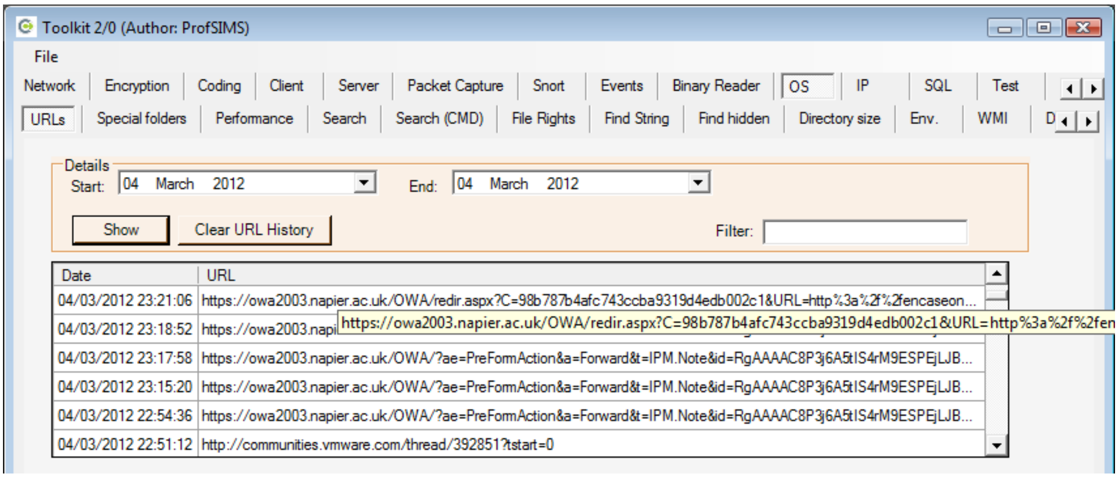
Challenge: By referring to the following, setup the server to allow other user logins.

Ubuntu FTP Server help:
https://help.ubuntu.com/6.06/ubuntu/serverguide/C/ftp-server.html

14.3 Toolkit Development – Web Cache

L14.18 This toolkit lab shows how to integrate more functionality into the security and forensics toolkit software. The finished toolkit can be used for reference.

The finished toolkit application can be downloaded from:
http://buchananweb.co.uk/dotnetclientserver.zip



Start: 04 March 2012
End: 04 March 2012
Filter:

Date	URL
04/03/2012 23:21:06	https://owa2003.napier.ac.uk/OWA/redir.aspx?C=98b787b4afc743ccba9319d4edb002c1&URL=http%3a%2f%2fencaseon...
04/03/2012 23:18:52	https://owa2003.napi
04/03/2012 23:17:58	https://owa2003.napier.ac.uk/OWA/?ae=PreFormAction&a=Forward&t=IPM.Note&id=RgAAAAC8P3j6A5tIS4rM9ESPEjLJB...
04/03/2012 23:15:20	https://owa2003.napier.ac.uk/OWA/?ae=PreFormAction&a=Forward&t=IPM.Note&id=RgAAAAC8P3j6A5tIS4rM9ESPEjLJB...
04/03/2012 22:54:36	https://owa2003.napier.ac.uk/OWA/?ae=PreFormAction&a=Forward&t=IPM.Note&id=RgAAAAC8P3j6A5tIS4rM9ESPEjLJB...
04/03/2012 22:51:12	http://communities.vmware.com/thread/392851?tstart=0

https://owa2003.napier.ac.uk/OWA/redir.aspx?C=98b787b4afc743ccba9319d4edb002c1&URL=http%3a%2f%2fen

For this lab, download the partially finished toolkit application source code (a Visual Studio C# Solution) from the link below:

Toolkit source code:
http://buchananweb.co.uk/toolkit.zip

Extract the source code for the C# Windows Application to a local folder. Next open the toolkit application with Visual Studio (VS) (double click the VS solution file **toolkit.sln**). You should see the **Solution Explorer panel** on the right of the VS Window.

Open the Toolkit Windows Form by double clicking the **client.cs** module, from the Solution Explorer panel. The Toolkit form should now be shown in the panel on the left. The Network tab should be displayed, as shown below.

Select the **OS tab,** and within that the **URLs tab.** Double click the **Show button**, and add the following code to the Showhistory() method:

```
           this.dgURLCache.Rows.Clear();
            this.dgFileCache.Rows.Clear();
            urlHistory = new UrlHistoryWrapperClass();
            enumerator = urlHistory.GetEnumerator();
            list = new ArrayList();


            GetHistoryItems();

            list.Reverse();


            if (textBoxFilter.Text != "")
            {
                enumerator.SetFilter(textBoxFilter.Text, STAT-
URLFLAGS.STATURLFLAG_ISTOPLEVEL);
            }
            foreach (STATURL u in list)
            {
                string[] url = new string[2];

                url[0] = Convert.ToString(u.LastVisited);
                url[1] = u.URL;
                STATURL u1 = (STATURL)list[0];

                if (u.LastVisited >= dtStart.Value && u.LastVisited <=
dtEnd.Value)
                {
                    u1 = (STATURL)list[list.Count - 1];

                    if (url[1].StartsWith("http"))
this.dgURLCache.Rows.Add(url);
                    else if (url[1].StartsWith("file"))
this.dgFileCache.Rows.Add(url);
                }
            }

            GC.Collect();
```

Test the new code, and show that recent URL's are displayed.

L14.19 Select the **Find Hidden** tab, and double click on the **Find/Start** button. Next add the following code to the **showFiles3** method:

```
try {
            if (rbHidden.Checked == true) DirSearch(listBox7, currentDrive,
textBox23.Text, this.tbSF.Text, this.textBox22, FileAttributes.Hidden);
            if (this.rbReadOnly.Checked == true) DirSearch(listBox7, cur-
rentDrive, textBox23.Text, this.tbSF.Text, this.textBox22,
FileAttributes.ReadOnly);
            if (this.rbSystem.Checked == true) DirSearch(listBox7, current-
Drive, textBox23.Text, this.tbSF.Text, this.textBox22,
FileAttributes.System);
            }
            catch (Exception ex)
            {
                CreateMessageForStatusAppend(this.lbError, ex.Message);

            }
```

And then add the following method:

```
void DirSearch(ListBox lb, string sDir, string search, TextBox tb)
        {

            try
            {
                foreach (string d in Directory.GetDirectories(sDir))
                {
                    System.Threading.Thread.Sleep(20);
                    CreateMessageForStatus(tb,"Searching " + d + "....");
                    if (searchFlag ==false) return;
                    try
                    {
                        foreach (string f in Directory.GetFiles(d, search))
                        {
                            CreateMessageForStatusAppend(lb, f);
                        }
                        DirSearch(lb, d, search,tb);
                    }
                    catch { }
                }
            }

            catch (Exception ex)
            {
                CreateMessageForStatusAppend(this.lbError, ex.Message);

            }
        }
```

Test, and show that it finds hidden files on the disk.

L14.20 Select the **Performance** tab, and double click on the **Start** button. Next add the following code on the timer3 tick event (timer3_Tick):

```
try
            {
                dgPerf.Rows.Clear();

                string[] s = new string[2];
                s[0] = "CPU"; s[1] = String.Format("{0:f2} %",
pc1.NextValue());

                CreateMessageForStatusAppend(this.dgPerf, s);

                float f = pc2.NextValue() / (float)1e9;
                s[0] = "Available Memory"; s[1] = String.Format("{0:f2} GB", f);
                CreateMessageForStatusAppend(this.dgPerf, s);

                f = pc9.NextValue();
                s[0] = "Processes"; s[1] = String.Format("{0:f0}", f);
                CreateMessageForStatusAppend(this.dgPerf, s);

                f = pc11.NextValue();
                s[0] = "IPv4 Datagrams"; s[1] = String.Format("{0:F0} data-
grams/sec", f);
                CreateMessageForStatusAppend(this.dgPerf, s);

                f = pc3.NextValue();
                s[0] = "ICMP"; s[1] = String.Format("{0:f2} messages/sec", f);
                CreateMessageForStatusAppend(this.dgPerf, s);
                f = pc4.NextValue();
                s[0] = "Active TCP connections (established)"; s[1] =
String.Format("{0:F0}", f);
                CreateMessageForStatusAppend(this.dgPerf, s);

                f = pc5.NextValue();
                s[0] = "TCP Segments "; s[1] = String.Format("{0:F2} Re-
ceived/sec", f);
                CreateMessageForStatusAppend(this.dgPerf, s);

                f = pc6.NextValue();


                s[0] = "TCP "; s[1] = String.Format("{0:F2} Segments/sec", f);
                CreateMessageForStatusAppend(this.dgPerf, s);

                f = pc7.NextValue();
                s[0] = "UDP "; s[1] = String.Format("{0:F2} Datagrams/sec", f);
                CreateMessageForStatusAppend(this.dgPerf, s);



            f = pc10.NextValue();
                s[0] = "Disk time"; s[1] = String.Format("{0:F0} %", f);
                CreateMessageForStatusAppend(this.dgPerf, s);

                                    f = pc8.NextValue();
                s[0] = "CurrentUrisCached"; s[1] = String.Format("{0:F0}", f);
                CreateMessageForStatusAppend(this.dgPerf, s);


            }
            catch (Exception ex)
            {
                CreateMessageForStatusAppend(this.lbError, ex.Message);

            }
```

Test the new code, and show that the buttons displays some key performance metrics.

Challenge: Add three more performance monitors.

Appendix

Linux packages/commands:

```
        sudo apt-get update && sudo apt-get upgrade -y
        sudo apt-get install -y apache2
        sudo apt-get install php5
        sudo apt-get install libapache2-mod-php5
        sudo apt-get install vsftpd
        sudo apt-get install telnetd
        sudo apt-get install inetutils-inetd

        sudo apt-get remove -y apache2
        sudo apt-get remove php5
        sudo apt-get remove libapache2-mod-php5
        sudo apt-get remove vsftpd
        sudo apt-get remove telnetd
        sudo apt-get remove inetutils-inetd
```

Lab 14a: Microsoft Azure Cloud

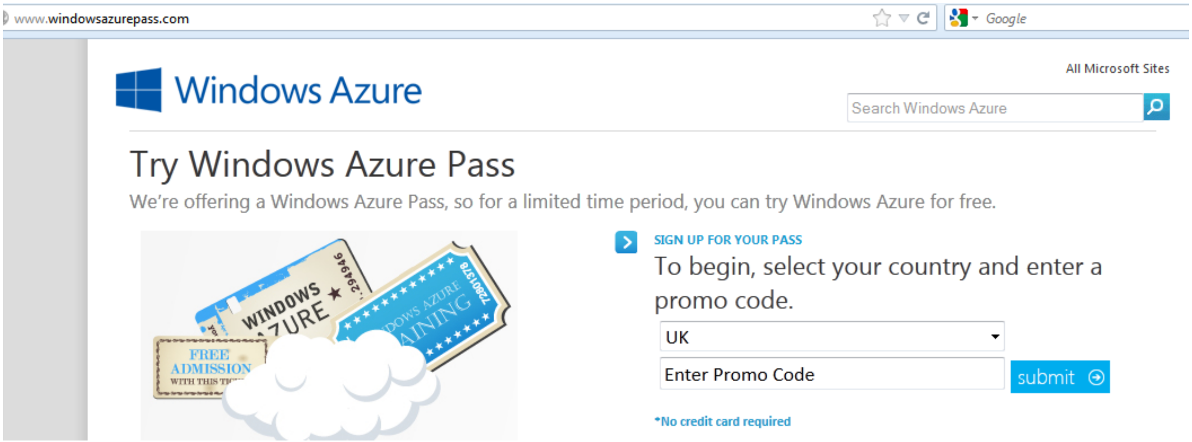
14a.1 Details

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.

14a.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/



Try Windows Azure Pass

We're offering a Windows Azure Pass, so for a limited time period, you can try Windows Azure for free.

SIGN UP FOR YOUR PASS

To begin, select your country and enter a promo code.

UK

Enter Promo Code

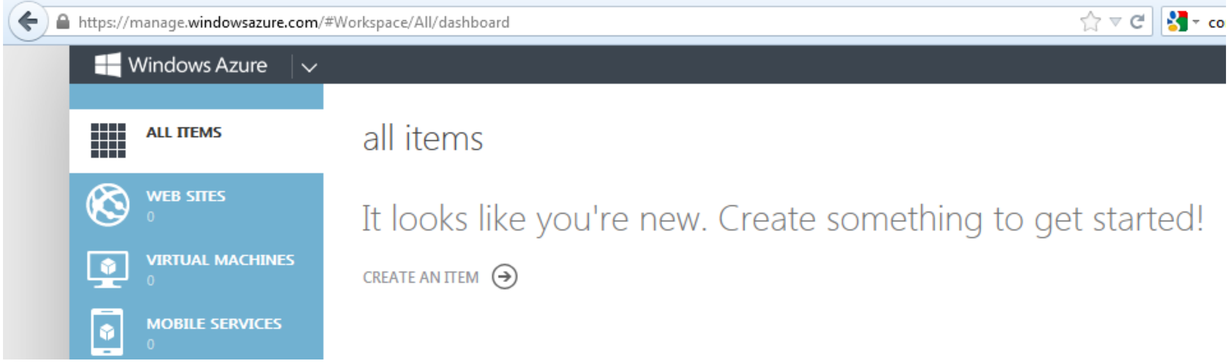
*No credit card required

14a.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:

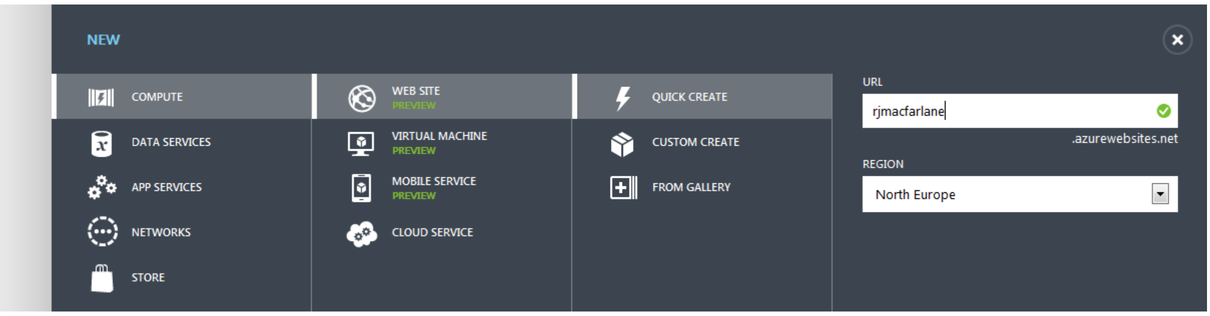
The **Azure cloud** development portal:
https:// www.windowsazure.com/



It looks like you're new. Create something to get started!

Create a Web Site

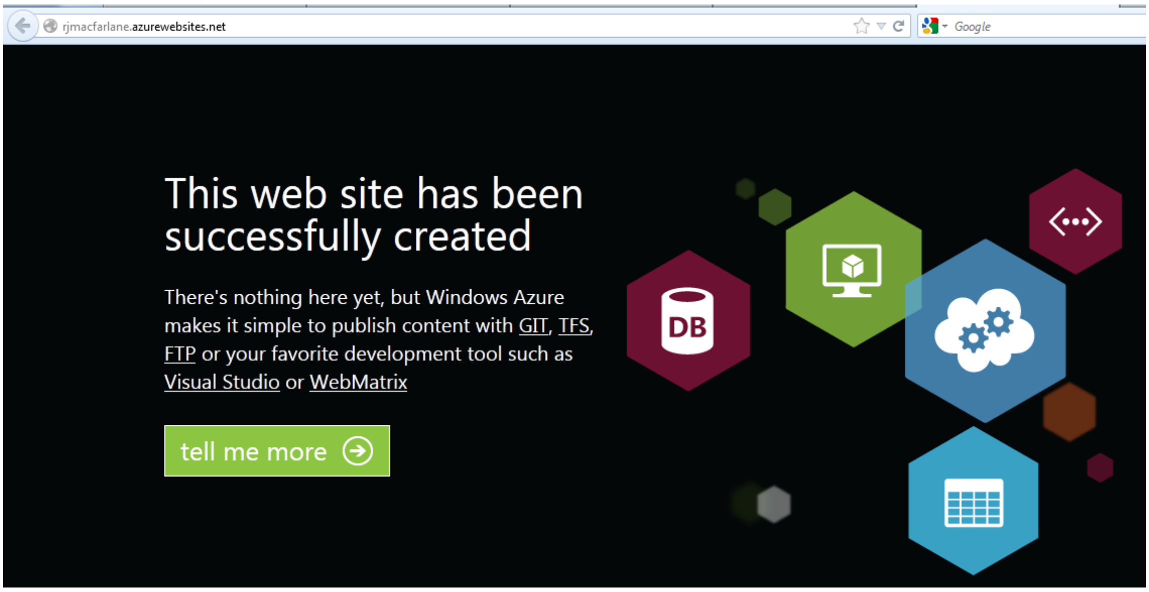
- Create a Web site named ***matricno***.azurewebsites.net



URL
rjmacfarlane
.azurewebsites.net

REGION
North Europe

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

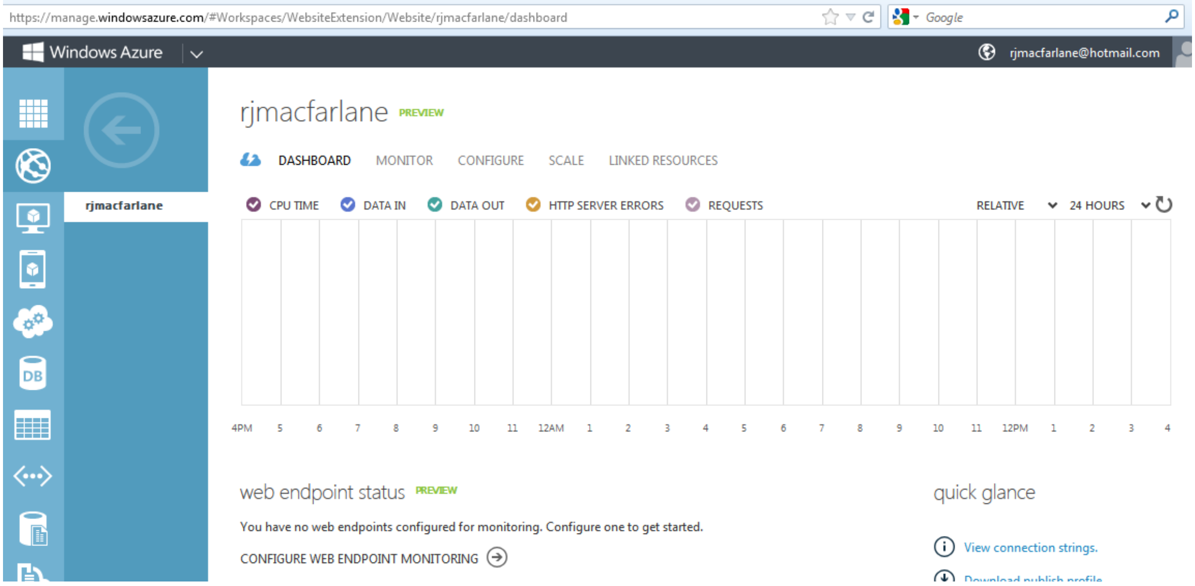


This web site has been successfully created

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

tell me more

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



You have no web endpoints configured for monitoring. Configure one to get started.

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

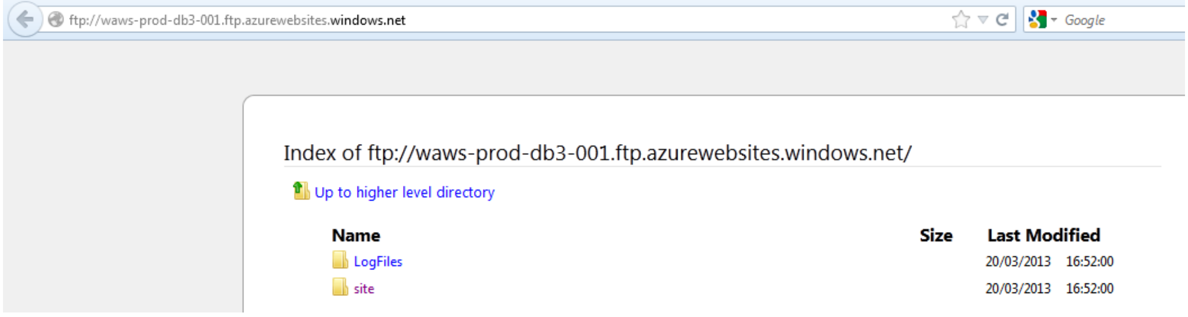
- From the Dashboard, determine your FTP details.

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.



Index of ftp://waws-prod-db3-001.ftp.azurewebsites.windows.net/

Up to higher level directory

Name	Size	Last Modified
LogFiles		20/03/2013 16:52:00
site		20/03/2013 16:52:00

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.

14a.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.

14a.5 Create SQL Data Infrastructure

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

14a.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

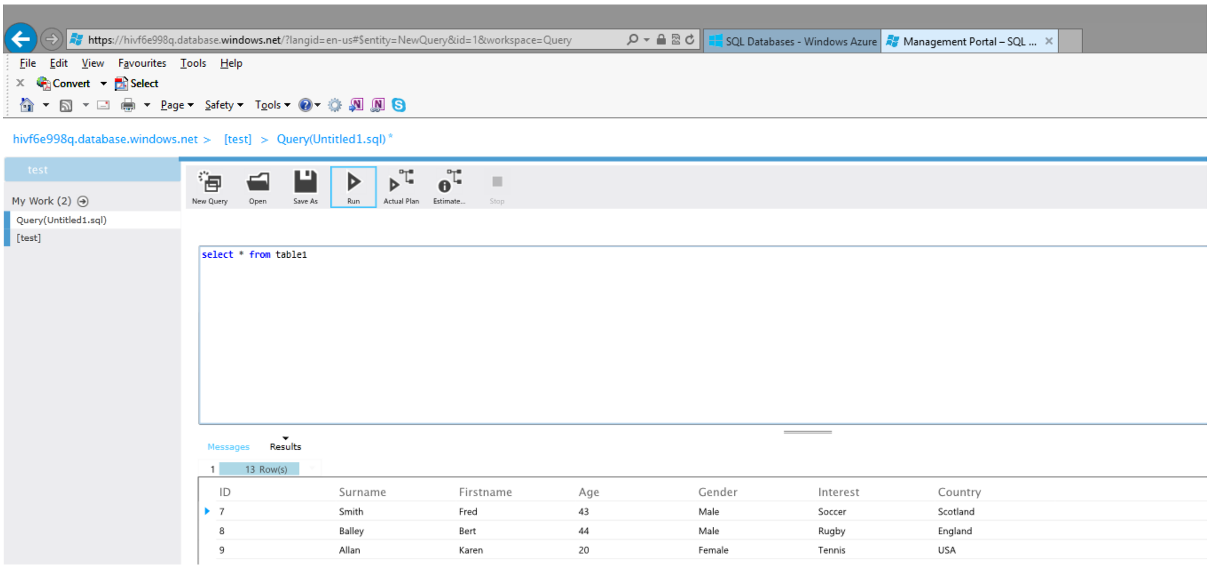
14a.7 Integrating with database

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 '[AaSsUu]%'
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
```


https://hivf6e998q.database.windows.net/?langid=en-us#$entity=NewQuery&id=1&workspace=Query
SQL Databases - Windows Azure
Management Portal – SQL ...
File
Edit
View
Favourites
Tools
Help
Convert
Select
Page
Safety
Tools
hivf6e998q.database.windows.net > [test] > Query(Untitled1.sql) *
test
My Work (2)
Query(Untitled1.sql)
[test]
New Query
Open
Save As
Run
Actual Plan
Estimate...
Stop
Messages
Results
1
13 Row(s)

```
select * from table1
```

ID	Surname	Firstname	Age	Gender	Interest	Country
7	Smith	Fred	43	Male	Soccer	Scotland
8	Balley	Bert	44	Male	Rugby	England
9	Allan	Karen	20	Female	Tennis	USA

Figure 1: Sample

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

14a.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.

14a.9 Northwinds

Now create a new database and upload the following:

Database:
http://billatnapier.com/nwind.zip

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

14a.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